

WISH QUERY COMPOSER

Vivien Liang, Greg Butler

*Department of Computer Science, Concordia University
1455 de Maisonneuve Blvd. West, Montreal, Quebec, H3G1M8, Canada*

Keywords: database, schema browser, form-based query, report generator, XML, Java, JAXB, Eclipse

Abstract: The WISH (With Intuitive Search Help) Query Composer is a software tool for composing form-based queries and their associated reports for relational databases. It incorporates the SQL and XML industry standards to generate user-friendly customizable queries and reports. It uses the very simple but flexible XML semantics to represent database schemas, SQL queries and result datasets, regardless of in which relational database management system (RDBMS) the data is stored. The tool is developed in the Eclipse development environment using the Java programming language with Swing components, and connects to the database through Java Database Connectivity (JDBC). The Java Architecture for XML Binding (JAXB) is used to automate the mapping between XML documents and Java objects.

1 INTRODUCTION

Relational databases today are essential to every business. The use of traditional query languages, such as SQL, are, in practice, restricted to Database Administrators (DBA) and software developers. Most end users access databases through applications with pre-programmed query statements. End users with no programming background or database knowledge do not have the flexibility to specify their own search criteria. Furthermore, the different DBMS vendors each have their own query interface and extensions to the SQL syntax.

The main motivation for constructing this tool was to support advanced users, who understood the notion of a query, to construct HTML forms which end-users could use to search a database and generate reports. The HTML forms provide either a single “canned” query, or a template for a family of queries. In the template, the end-user could have options to set the data attributes to be included in the report, and to set particular search criteria. The range of options is decided by the advanced user when creating the form.

The tool includes a browser for the database schema, so the advanced user has readily at hand all the information required to correctly create the query. The tool also allows the creation of the “specification” of the structure of the report that is generated from the

result set of the query.

The tool is called the WISH Query Composer, where WISH stands for “With Intuitive Search Help” since it provides an intuitive search tool for end-users, and provides the advanced user with an intuitive environment to create the forms and report specifications for the end-user.

One requirement was to use XML as the basis of all information transfer between the DBMS and the tool. This freed us from the particularities of any single DBMS platform, and gave us a powerful set of development tools in Java that greatly simplified our task.

This short paper discusses the system requirements, design, and components, as well as the technologies used in its development. The paper is organized to present the background on the technologies, then the requirements and design of WISH.

2 BACKGROUND

The power of databases comes from a body of knowledge and technology that has developed over several decades and is embodied in specialized software called a Database Management System (DBMS) (Garcia-Molina et al., 2000). One of the most common operations a DBMS supports is

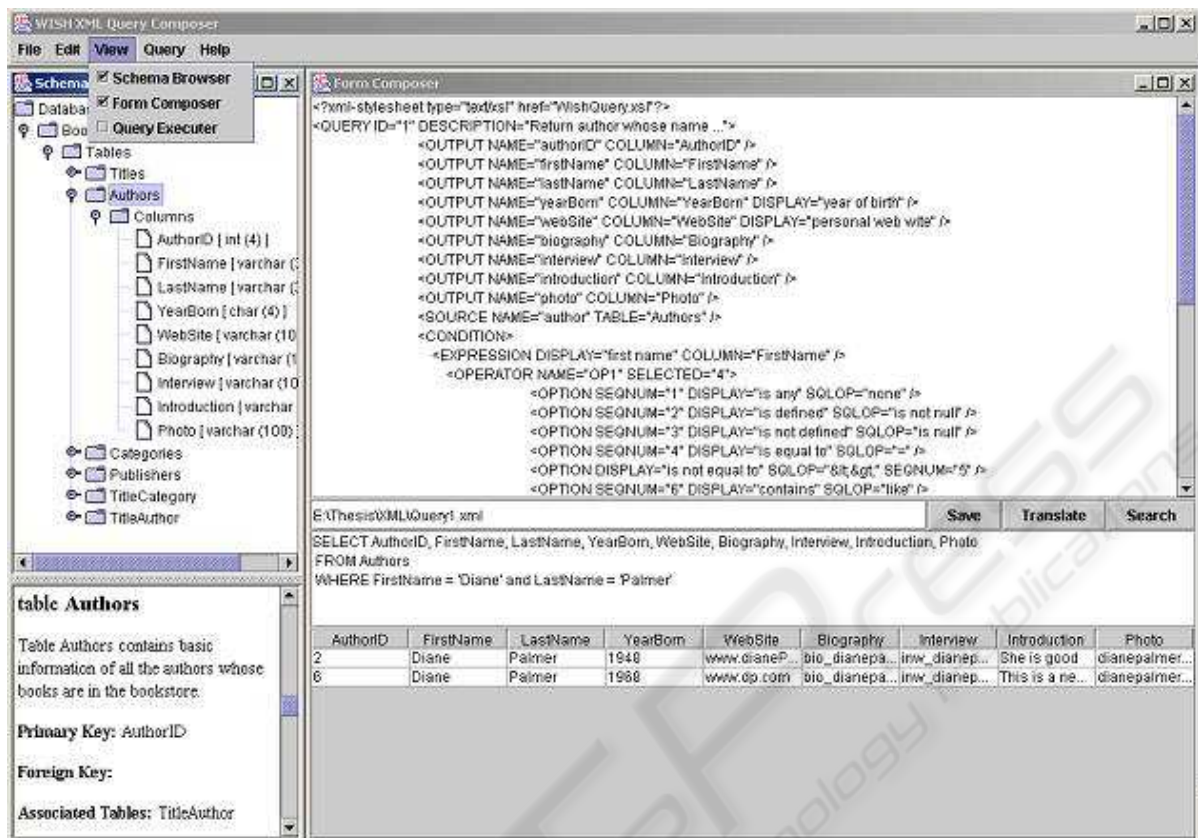


Figure 1: Schema Browser and Query Composer

database query, an operation to extract specified data from databases. For this purpose, database query languages, such as Structured Query Language (SQL), were developed. SQL has a large number of capabilities, including statements that query the database. Queries are generally expressed with a `select-from-where` statement.

In this version of WISH, relational DBMS's and SQL are the main objects of data queries. In WISH, the queries submitted to the relational databases are compatible with SQL standards, including supports for aggregation, duplicate names and sub-query.

For many people it is much easier to express a query visually than to write a set of statements in a language such as SQL. The advantage of form-based queries is that many users are comfortable with forms and tables (Zloof, 1977). When the schemas of the tables are visible on the screen, the users do not have to remember column names, semantics and other details of a query.

The eXtensible Markup Language (XML) is a meta-language defined by the World Wide Web Consortium (www.w3c.org) as a subset of SGML. An XML Schema is an XML document that defines the

valid format of an XML dataset. Schemas describe structural relationships and data types. The eXtensible Stylesheet Language (XSL) is a family of W3C recommendations for defining XML document transformation and presentation. An XSL style sheet is a file that describes how to display an XML document of a given type. A style sheet can be used to transform any instance of the XML Schema it is designed for.

There are at least four benefits to use XML as the format of the database schema, the database query and result dataset (Quin, 2000).

1. **Mobility of data:** The XML representations of the database schema, query and report provide a single format independent of the DBMS thus simplifying data exchange and portability of databases.
2. **Browser Views:** As more browsers support XML directly, the XML query or result can be displayed on the web with XSL transformation.
3. **Databases into Documents:** For interoperability with a document-based system, the XML documents from a relational database can be loaded into a document repository or an object-oriented database.

4. XML Tools: Expressing database metadata, queries and results as XML allows the processing of this information in standalone XML-based tools, such as formatting or statistics packages.

A new Java API called Java Architecture for XML Binding (JAXB) (Ort and Mehta, 2003) makes it easier to access XML documents from applications written in the Java programming language. This API is available in the Java Web Services Developer Pack (JWS DP) pass 1.1. JAXB is a Java technology that generates Java classes from XML schemas. As part of this process, the JAXB technology also provides methods for unmarshalling an XML instance document into a content tree of Java objects, and marshalling the content tree into an XML document. JAXB provides a fast and convenient way to bind an XML schema to a representation in Java code, making it easy for Java developers to incorporate XML data and processing functions in Java applications.

3 WISH TOOL

The development of the WISH Query Composer adopted an object-oriented approach using the Unified Modeling Language (UML). Requirements were captured as use cases, and the design mainly consisted of developing several XML schemas (for representing database schemas, query forms, SQL queries, and result sets) as the JAXB binding of the corresponding XML documents provided our Java objects in the design.

The design of WISH is an object-oriented approach that follows the Unified Modeling Language (UML) methodology. The Use Case Diagram in Figure 2 shows the relationships between actors and use cases of WISH.

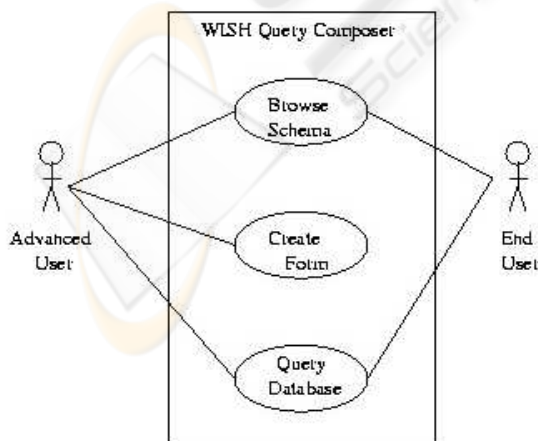


Figure 2: Use Case Diagram

The Use Case Diagram in Figure 2 shows the relationships between actors and use cases of WISH. There are two types of actors:

1. End-users who may be very unsophisticated in their knowledge of relational database technology. They do not wish to write SQL queries nor to be shown a database schema as a collection of SQL table definitions.
2. Advanced-users who may not be as expert as a Database Administrator (DBA) in relational database technology, but they do know SQL, XML, HTML, and the application domain so they can comprehend the database schema for the domain.

The use cases are

1. Browse a Database Schema, where the advanced user consults the existing schema for a database created by the DBA, in order to compose the query correctly. An end-user may consult the database schema in order to select which attributes to include in the report (provided the query form has that option).
2. Create a Query Form, where the advanced user composes a form and report specification for a family of queries.
3. Pose a Query to the Database, where the end user completes the query form, submits it to the DBMS, and receives the report based on the result set.

When debugging the query form, an advanced user may also pose the query to the database, and see the result set displayed in the default tabular representation. When debugging the report specification, an advanced user may also pose the query to the database, and see the result set displayed as a report.

WISH is designed as three subsystems:

1. a Schema Browser;
2. a Form Composer; and
3. a Query Executer.

The Schema Browser, which is shown at the left of both Figure 1 and fig:screen2, provides a tree view of the information about tables, attributes, keys and types; as well as documentation (bottom left of the Figures) on them.

The Form Composer, which is shown at the top right of Figure 1, allows the advanced user to design an XML/HTML form for query, an XML report for the database results, and a style sheet for presentation. The advanced user needs to decide on the SQL query and its parameters. In this version of WISH, the XML itself is created in an editor (as shown); however, more graphical “builders” are planned future work. To aid debugging, the advanced user may click on the Translate button (see Figure 1) to see the equivalent SQL statement displayed in the pane

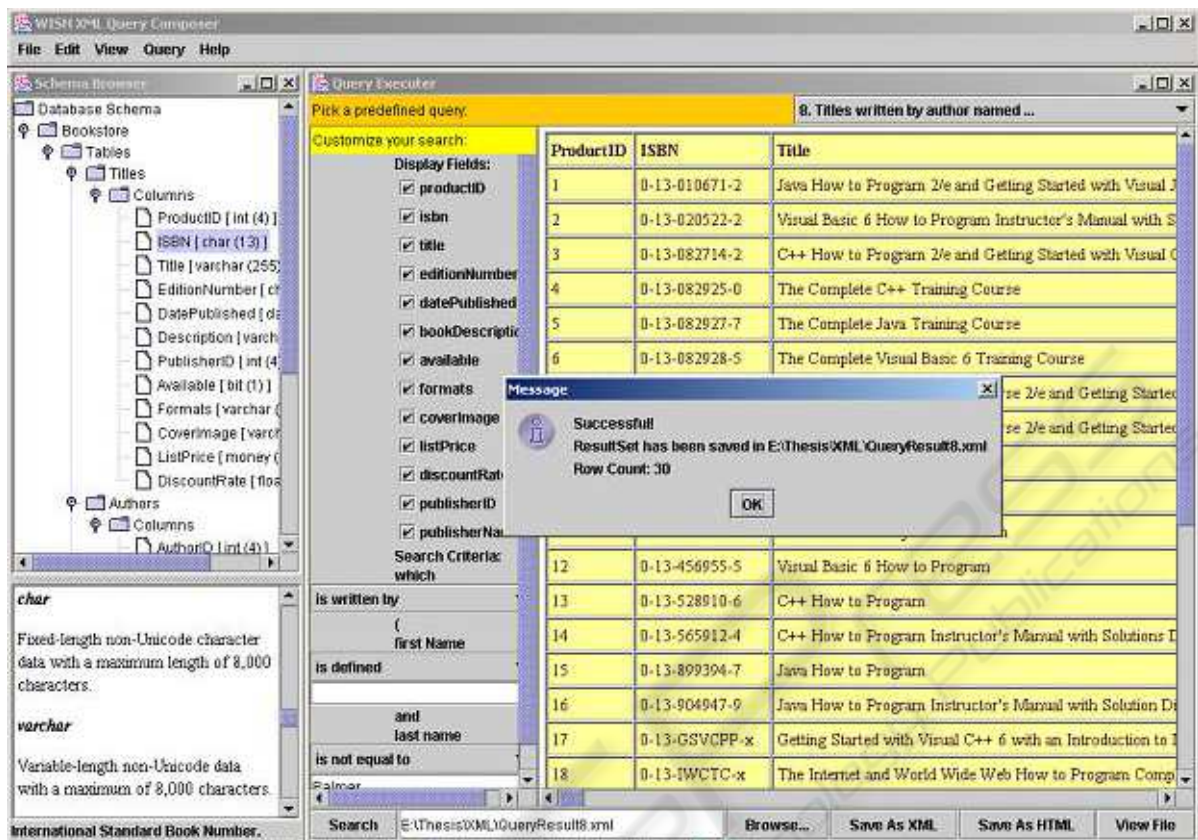


Figure 3: Example of Report

below. This validates the XML text complies with its schema, and translates it to SQL for perusal by the advanced user. By clicking on the *Search* button, the query is submitted to the database, and the result set if displayed in tabular form in the pane below the SQL query.

A highly customizable query form is shown in the centre of Figure 3: it has checkboxes for selecting which attributes to include in the report, and slots for entering attribute values (or ranges). The query forms are saved in a repository. They are accessed by the end-user as "canned" queries from a pull-down list in the Query Executer.

The Query Executer, which is shown at the top right of Figure 3, allows the end-users to customize a query (using checkboxes and slots as shown in the centre of the Figure) and to submit it to the database. The corresponding SQL query is sent to the database. Results can be seen immediately in tabular form, as in Figure 3. The query results can then be saved into an XML document in the report format designed by the advanced user, so it may be rendered by a browser using the specified stylesheet.

4 CONCLUSION

The WISH Query Composer has proven to be a highly useful tool. The use of XML, Java, and JAXB meant a very short development time (less than two months by one person). Our future plans are to construct graphical builders for the XML forms and report specification as alternatives to the existing text editors.

REFERENCES

- Garcia-Molina, H., Ullman, J. D., and Widom, J. (2000). *Database System Implementation*. Prentice-Hall.
- Ort, E. and Mehta, B. (2003). Java architecture for xml binding(jaxb). Technical Article, java.sun.com.
- Quin, L. (2000). *Open Source XML Database Toolkit*. Wiley.
- Zloof, M. (1977). Query by example: A database language. *IBM System Journal*, 16(4):324–343.