

Integrating SuperSQL in Software Development Environment Systems

Ria Mae BORROMEO and Motomichi TOYAMA

† Department of Information and Computer Science, Keio University,

3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522 Japan

E-mail: ria@db.ics.keio.ac.jp, toyama@ics.keio.ac.jp

Abstract

SuperSQL is an extension of SQL that allows various application data to be directly generated from a query. The query syntax is similar to SQL with additional data formatting capabilities. A SuperSQL query can generate different kinds of output such as a HTML files, LaTeX files, Flash animations, MS-Excel worksheets, PDF, etc. While SuperSQL may be very useful for media publishing and presentations, its use could be complicated. Complex queries may result from the inclusion of formatting data and when errors occur in such queries, debugging could be difficult. To address these problems, we designed a tool that allows syntax highlighting, execution, HTML output viewing and error reporting of SuperSQL queries and integrated it to the Eclipse development environment. We then conducted a study to evaluate its usefulness.

Keywords SQL extension, web publishing

1. INTRODUCTION

SuperSQL is a database query language designed to translate information obtained from a database management system (DBMS) into a specific application data. The current implementation is capable of generating web pages in HTML, printed matters in LaTeX sources and PDF files, spread sheet data in Excel, animations in Flash, 3D computer data graphics in X3D sources, and web applications adhering to Ajax.

The syntax of SuperSQL is similar to SQL. The major difference is that instead of `SELECT <target list>` clause in SQL, SuperSQL uses the `GENERATE <medium> <TFE>` clause. TFE (Target Form Expression) is a generalization of target list of SQL. However, 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. [2]

The SuperSQL system is invoked in the command prompt with the SuperSQL query file as input. Upon execution, some log messages are printed in the command prompt. This is not a problem when there is no error. However, when errors are encountered, the normal log messages mix with the error messages. SQL exceptions and SuperSQL errors also mix thus making the messages confusing to the debugger. Moreover, queries that contain formatting data may be hard to read in plain text editors.

The statements above affect the usability of SuperSQL negatively. Thus, to enhance the usability of SuperSQL, we realized the need to provide SuperSQL syntax highlighting, organized error reporting, viewing of output, and integrate these functionalities in one development environment. In this study, the Eclipse plug-in system was used to address these concerns.

Although SuperSQL supports many application data output, our tool is still limited to viewing only HTML outputs within the Eclipse environment. We decided to implement this first as it is the most commonly used output format.

Two surveys were conducted with potential SuperSQL programmers as respondents. The surveys show that integrating SuperSQL in Eclipse is preferred by the respondents over the way it is originally used.

2. SUPERSQL DEVELOPMENT ENVIRONMENT

The original SuperSQL development environment makes use of at least three tools: the editor to create queries, the command prompt to execute SuperSQL and the application data program to view the output. Figure 1 shows a sample SuperSQL development environment. In our study, we use a web browser to view the query output.

Since SuperSQL is an extension of SQL, some editors provide syntax highlighting for SQL elements. However, elements that are unique to SuperSQL are not highlighted.

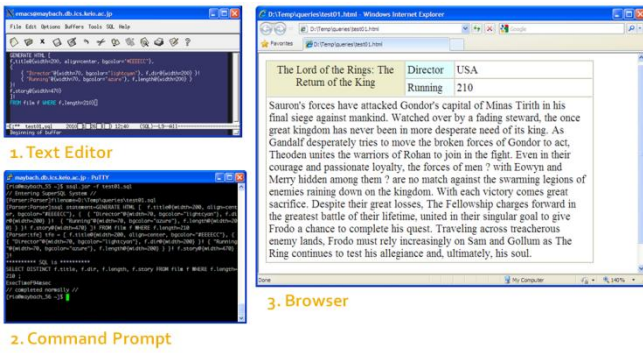


Figure 1. SuperSQL Development Environment

While this environment provides the programmer freedom to take advantage of all command prompt and browser functionalities, this setup can get disorganized as more queries are created.

3. OUR APPROACH

3.1 The Eclipse Development Environment

Eclipse is a multi-language integrated development environment that easily extensible because of its plug-in system. The extensible plug-in system allows Eclipse to work with many languages including typesetting languages like LaTeX, network applications such as telnet, and database management systems. Because of its extensible properties and user population, we decided to integrate SuperSQL into it.

3.2 Syntax Highlighting

Syntax highlighting is a feature of text editors to improve the readability and context of the text. Moreover, the color difference in the text elements helps programmers find errors in their program. Currently, there is no other editor that supports syntax highlighting for SuperSQL queries.

We implemented syntax highlighting by creating an XML-based configuration file for SuperSQL syntax. We then added this configuration file in an existing Eclipse plug-in called Colorer. The Colorer plug-in takes syntax highlighting configuration as input then highlights text in Eclipse editor accordingly. The configuration file we created was based on the SQL configuration file with the addition of the following keywords and function names: GENERATE, HTML, XML, XLS, X3D, SWF, PDF, FOREACH, imagefile, link, invoke and embed.

3.3 HTML Preview and Syntax Error Reporting

We integrated the functionalities of the command prompt and the browser in Eclipse by creating a new plug-in. The plug-in creates a SuperSQL toolbar button. This allows execution of a SuperSQL query just by pressing a button rather than typing commands in the command prompt. This toolbar button can be seen in Figure 4. The plug-in also makes use of Eclipse's console to print the log and error messages that were previously shown in the command prompt. It also invokes Eclipse's built-in HTML browser to show the output of the query upon successful execution.

3.4 Implementation Details

The SuperSQL Plug-in communicates with Eclipse and the SuperSQL system. The following diagram shows an overview of how the whole system works.

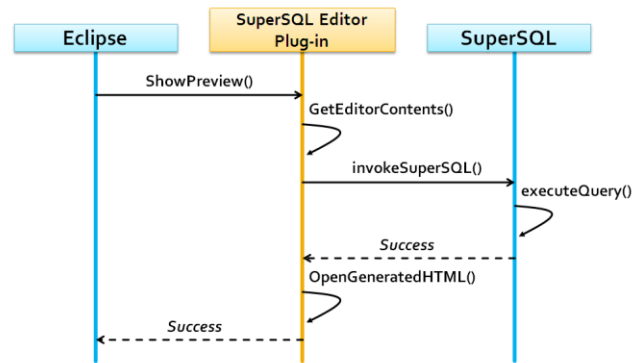


Figure 2. Sequence diagram for showing HTML Preview

The SuperSQL toolbar button is pressed to activate the SSQL plug-in and show the generated HTML page. The plug-in will read the SuperSQL query from the editor then call SuperSQL to execute the query. If the query is successful, the plug-in will open the generated HTML in the Eclipse browser.

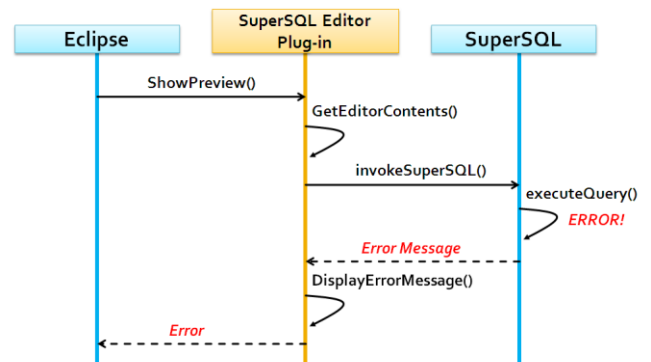


Figure 3. Sequence diagram for error reporting

As illustrated in Figure 3, if the query is not successful, the error message will be displayed in the Eclipse console.

In order to create a new plug-in, an Eclipse wizard was utilized. The wizard automatically generated the plug-in manifests and the plug-in classes. The plug-in manifests contain definition on how the plug-in relates to all the others in the system. In this project, the plug-in is set to be a new toolbar button.

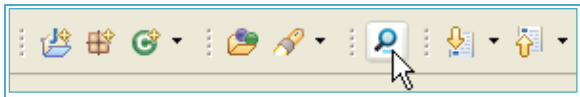


Figure 4. The Eclipse toolbar button of the SuperSQL plug-in

Two plug-in classes are automatically created. The first one is the Activator class which takes care of plug-in start up and the second one is the Handler class where the functionalities of the plug-in are programmed. Every time the toolbar button is pressed, the following algorithm is executed.

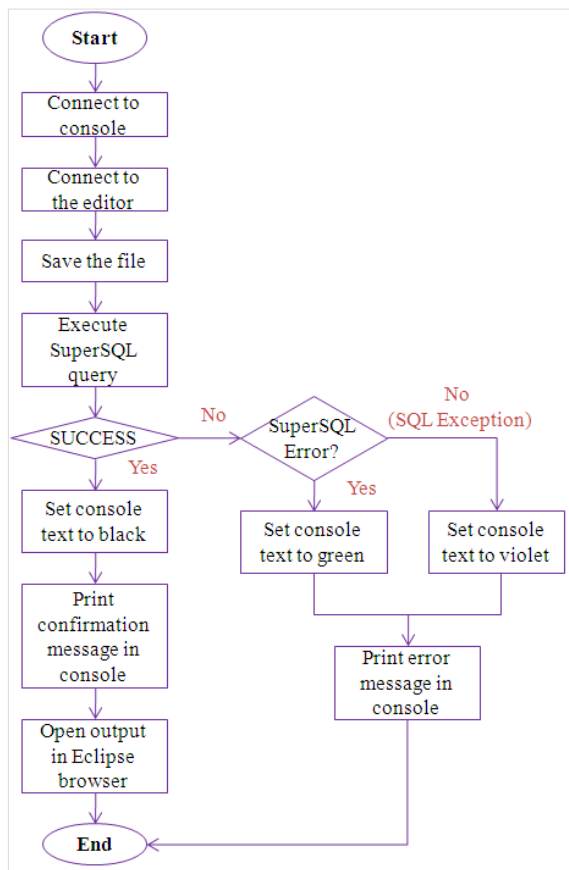


Figure 5. Flowchart of the Handler Class

Three GUI components are being used by the plug-in: the console, editor and the browser. The *IConsoleManager*

class was used to connect to the console. The *IEditorPart* class was used to connect to the editor and the *IWebBrowser* class was used to connect to the browser. These classes are all included in the Eclipse Platform API Specification.

The existing SuperSQL source code was also packaged as a plug-in in order for the algorithm above to work.

4. RESULTS

The configuration file for syntax highlighting was successfully created. The following result was obtained.

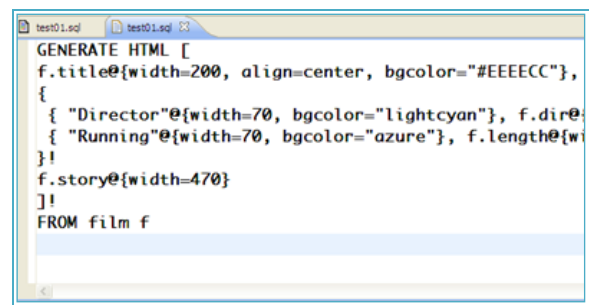


Figure 6. SuperSQL query without syntax highlighting

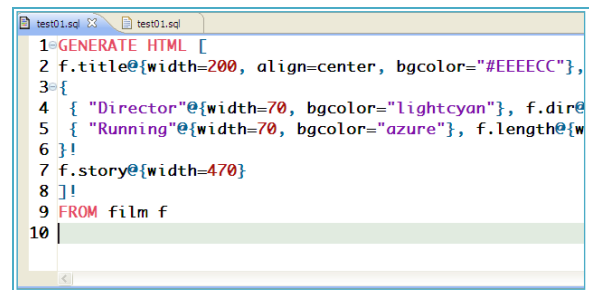


Figure 7. SuperSQL query with syntax highlighting

The new plug-in was also successfully created and integrated in the Eclipse development environment.

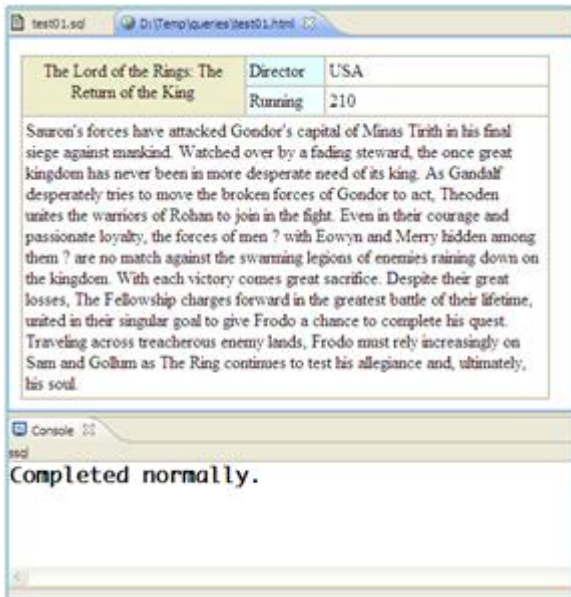


Figure 8. Preview of a query that completed normally

When a query completes normally, the result is shown in another tab in the Eclipse Environment.

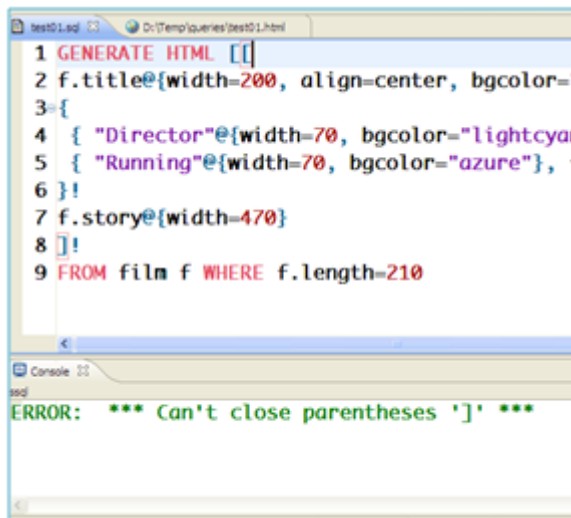


Figure 9. Query execution with SuperSQL syntax error

When errors are encountered, the error message is simply printed in the console.

5. EVALUATION

To evaluate the resulting tool, two surveys were conducted. The respondents were 15 programmers with at least three years work experience. They have never used SuperSQL but they were provided with a background about SuperSQL's basic functionalities. We consider them as potential SuperSQL programmers.

The first survey was intended to prove the usefulness

of syntax highlighting. We found that 14 out of 15 respondents preferred the editor with syntax highlighting. The respondent who preferred the editor without syntax highlighting reasoned that since a SuperSQL query has many different elements, a query can be very colorful and these colors may be distracting rather than helpful.

The second survey was intended to evaluate the overall appeal of the compact development environment to potential SuperSQL programmers. The respondents were asked which they would prefer, the traditional way of using SuperSQL which uses at least three windows, or the compact one which is integrated in Eclipse.

We found that 14 out of 15 respondents preferred the compact environment. The respondent who preferred the traditional way reasoned that in the traditional way, it will be easier to test the resulting web page against different browsers. Other respondents also pointed that a major disadvantage of Eclipse is its speed since the entire environment runs over the Java virtual machine.

Based on the comments of potential SuperSQL programmers, there are still things to be improved in our tool like speed, syntax highlighting preferences and browser support. However, we can conclude from the surveys that integrating SuperSQL in Eclipse is preferred by the respondents over the original development environment.

6. CONCLUSION AND FUTURE WORK

The SuperSQL system has been integrated to the Eclipse development environment thus providing an editor with syntax highlighting and allowing executing and viewing output of SuperSQL queries in one compact environment. Error reporting has also been improved by providing color-coding to the two different kinds of errors. This new development environment is preferred over the old one by potential SuperSQL users.

We plan to add more improvements to our tool such as specifying line numbers in error messages and viewing other output formats such as PDF.

REFERENCES

- [1] D. Rubel, E. Clayberg. Eclipse Plug-ins, 3rd Edition, 2009.
- [2] M. Toyama, SuperSQL: An Extended SQL for Database Publishing and Presentation in SIGMOD '98 Seattle, WA, USA, 584-586.
- [3] Colorer Plug-in <http://colorer.sourceforge.net/>
- [4] SuperSQL Website <http://ssql.db.ics.keio.ac.jp/>