*2012 International Conference on Computer Technology and Science (ICCTS 2012)*
***IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore***
*DOI: 10.7763/IPCSIT.2012.V47.23*

# Text Editor with Database Interface

Shashidhara P Ithal[1], Sadanand Rudraiah[2], Omprakash Sringeri[3+]

OmS Patent Services Pvt. Ltd,

Bangalore, India

**Abstract.** Editors are generally used because of its extensive features that allow the users to modify and update text document as per their needs. Existing text editors provide just the ability to edit, manipulate text information but they do not provide the ability to connect to the database on fly and upload the text data to the database. In this paper, a text editor is proposed that enables user to open text document and upload selected content to the database. It also enables the user to view the database metadata for creating a workspace in which the contents are being uploaded.

**Keywords**: **Text Editor; Visual interface to relational database;**

## 1. Introduction

Text Editors are basically used to display, edit documents or text files. Editors are widely used because of their ready to use, easy to use features. In the current context, a text editor is defined to be an editor capable of at least selecting a text part in the document as text. Existing text editors allow the user to view the document in context, read the document and modify the document. Existing text editors do not interface to the relational database and do not provide the ability to save the contents of the document onto the relational database as and when the user is reading through the document.

This paves way to interface the relational database through the text editor itself making the user to perform a single action of viewing, selecting and by suitable menu items saving the necessary contents onto the relational database.

We have developed and proposed in this paper a text editor software package that enables display of text information from one or more document format (html, ASCII text file), selection of text portion from the document, mapping selected text to relevant fields of relational database table and saving them to the fields in a relational database table. Initially the user selects an appropriate relational database and a table. Certain portions of the displayed document that the user thinks to be saved as the information for respective fields of the relational database table can be selected from the document and just by right clicking on the selected portion, the user can map it to the relevant fields of the table. The user then selects the suitable menu items for uploading selected data into the respective relational database.

This text editor also serves as a mini web browser for opening HTML files, where user enters the URL of the HTML files to be opened. The contents of the html document opened can be mapped to relational database table fields and saved.

## 2. Background

### 2.1 Existing Text Editors:

A text editor is a program that allows user to open, view, and edit plain text files. Text editors are used by wide variety of people, for a wide variety of purposes.

Some of the salient features that the existing editors provided are as below:

[+] Corresponding author. Tel.: +91 94483 56142.
*E-mail address*: omprakash@omspatentservices.com

- Ability to cut, copy and paste which enables the duplication of the text within the file or between files.
- Ability to undo and redo, which enables user to change or bring back to the last edit.
- Ability to highlight a piece of text in the editor to read it for further reference.
- Ability to insert hyperlink, look up synonyms, etc. when user right clicks mouse button.

Interactive Text Editors [1]:

Interactive text editors allow the manipulation of a set of files stored on the host machine by means of a terminal device such as a teletypewriter or a display-keyboard combination. The files may contain natural language text, computer programs, or alphanumeric data.

The Editor’s role in creating documents [2]:

Editors serve a variety of roles in preparing documents, including helping to plan the document, coordinating the work of writers, and supervising production; however, their fundamental and defining role is to improve the document by marking change in the draft they receive from the author.

### 2.2 Database Interface:

The database interface (DBI) [4] separates the connectivity to the DBMS into a “front-end" and a “back-end". The elements of the DBI are:

- A set of classes and methods that define what operations are possible and how they are defined.
- Database connectivity drivers.

## 3. Proposed Text Editor Architecture

In this paper, proposed text editor provides the necessary menu options for selecting a relational database to connect to, selecting a table to upload information to, performing the basic actions of opening a document, editing, manipulating, mapping selected text portion to relational database table field and saving the contents of the mapped portions of the document to relational database table..

The Software Architecture of text editor for database interface is as shown in Fig. 1. Text editor serves as a medium for user to open a text file or a HTML document and also serves as database interface providing a link to relational database using JDBC [3] connectivity as shown in Fig. 1. Thus user can upload the contents from the document to the relational database available in the server with the help of suitable buttons like Upload and Update that are provided in the text editor.

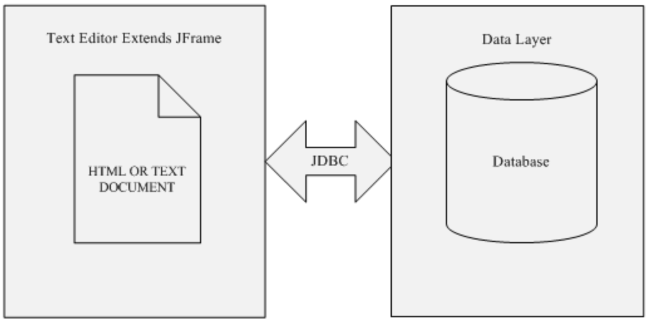


Fig. 1: Architecture of Text editor for Database Interface.

As an example for this application, Patent Database is used for uploading data. The patent database is a repository of patent information. The patent document in context is read, viewed and selected. Selected contents are then uploaded to the selected table and fields in the patent database.

This software package thus reduces the time that the user was taking to upload contents of the patent document to the patent database via sql inserts since only the upload or update button will connect and then save the mapped document contents to the table fields in the patent database.

Also, this text editor enables the user to open HTML files through a mini web browser, map content to relational database table field and upload the content to relational database.

## 4. Working of Proposed Text Editor

Technologies used for implementing the text editor are:

- Java Swing: Swing is used for creating the user interface.
- MySQL: MySQL is used as database.
- JDBC: Java Database Connectivity is used for connecting to the database.

Detailed description of the proposed Text Editor:

### Database List:

Databases available in database server are first listed when the text editor is opened for use by making use of the JDBC API:

```
getMetaData().getCatalogs();
```

The database list will be populated into DBList menu item provided in the menu bar of the text editor.

The databases available in the server can be viewed via DBList menu item provided in the text editor. User selects appropriate patent database for a particular session for uploading relevant patent data to respective fields of that patent database.

### Login dialog:

Once user selects a particular database, Login dialog appears where the text editor validates the user's credentials against the database user list. If login is unsuccessful, an error message will be displayed indicating that username and password is incorrect. If login is successful, database connection will be established between the selected database and text editor by making use of the JDBC API:

```
Conn = DBConnectionManager.getconnection(DBURL, username, password);
```

Where DBURL is the URL of the database selected, username is the name of the database user that is retrieved from the login form, password is the password retrieved from the login form. For example:

### Table list of selected Database:

After connection is established successfully, TableList menu item provided in the menu bar of the text editor is populated with the tables of the selected database by making use of the JDBC API:

```
getTables(DBname, null, "%", null);
```

User selects appropriate table from TableList menu item for a particular session.

### Field list of selected table:

After table is selected from the TableList menu by the user, FieldList menu item provided in the menu bar of the text editor is populated with the fields of the selected table by making use of the JDBC API:

```
getColumns(DBname, null, Tablename, "%");
```

The mandatory fields are identified by checking whether the database column is nullable. The mandatory fields in the selected table are marked with an icon (*) placed next to it in FieldList menu.

The mandatory fields are automatically loaded into popup menu item which appears when right click mouse event is triggered by user. The non mandatory fields may be selected by user from the FieldList menu if specific data is to be uploaded to those fields. All the selected non mandatory fields will be appended into popup menu item along with the mandatory fields.

Thus, user can easily map the patent data to the respective fields of the relational database table. Initially, user selects the patent data from the document. Selected patent data is mapped to relevant fields when the right button of mouse is fired.

### Opening a Document:

A dialog window is enabled when user selects the Open menu item in the text editor. User uses this dialog window to browse files. User selects either a text file or a HTML file and opens it.

### Mini Browser functionality in Text Editor:

The basic components needed for creating a mini browser are as follows:

- A text field for entering new HTTP URLs and three buttons namely Back, Forward and Go.
- A JTextPane for displaying the current page.

User enters the HTTP URL of the HTML files to be opened in the text field and selects Go button. The HTML file corresponding to that URL will be opened in the text editor's JTextPane. The user can navigate between the pages by just clicking on the Hyperlink in the document.

### Selecting appropriate content and adding to the respective field:

User selects certain portion by firing the left click mouse button and dragging the mouse till required data to be loaded into respective fields is highlighted. Once the right click mouse event is triggered by the user, popup menu containing the fields of the patent database appears. User then selects appropriate field from the popup menu to which the selected patent data is to be stored.

After loading all the mandatory field data and other non-mandatory field data in to a buffer, user can then upload all the patent data to the respective fields in the patent database via Upload button in toolbar of the text editor.

### Viewing uploaded data through MySQL editor:

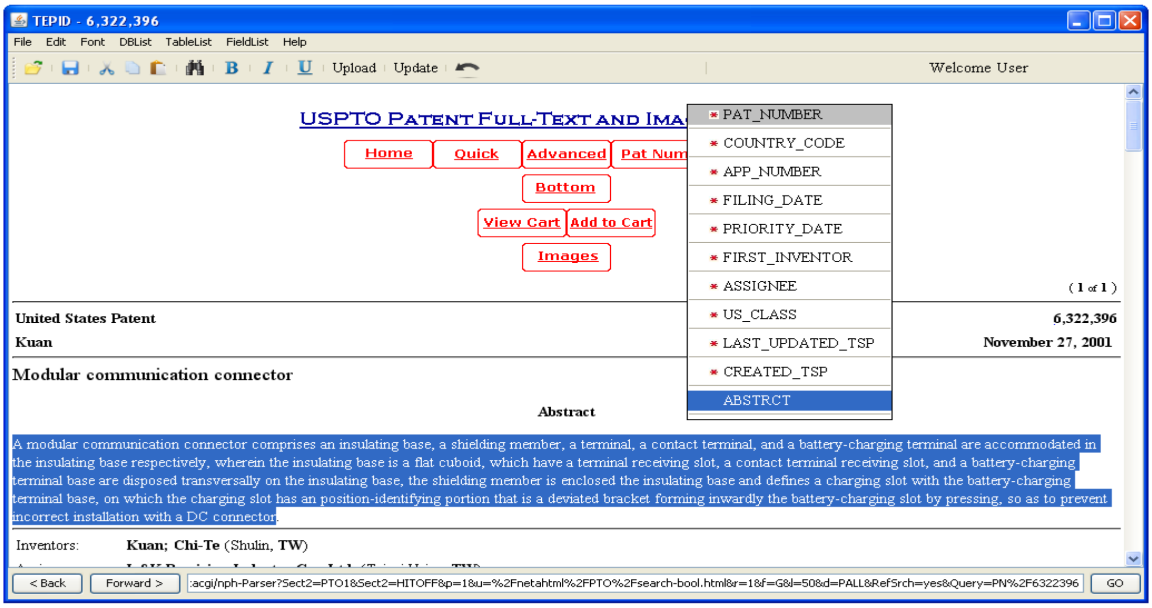
After information message indicating that the data is uploaded successfully is displayed, user can log into MySQL console and can check the uploaded data in the respective database.

If user wants to update patent data for the non mandatory fields from the text editor, user short lists the non mandatory fields from the FieldList menu item for a particular session to which the patent data is to be updated. These short listed menu items will appear when user selects the right click mouse button. User selects certain portion of the patent data to be loaded into specific fields by firing left click mouse button and dragging the mouse till required data is highlighted. Once the right click mouse event is triggered by user, popup menu containing user selected fields of the patent database appears. User then selects appropriate field from the popup menu to which the selected data is to be stored. After loading all the data in to a buffer, user can then update all the data to the respective fields in the patent database via Update button in toolbar of the text editor.

### Screenshot of Text Editor's Graphical User Interface:

The user interface of Text Editor is shown in Fig. 2. The text editor supports the following features:

- Opening and saving a text file or a HTML file via menu bar items or toolbar buttons.
- Searching for a word in the document.
- Mini web browser functionality for opening HTML files, where user enters the URL of the HTML files to be opened.
- Ability to view database metadata via menu bar items.
- Ability to upload or update patent data to respective fields of the patent database.



( 1 of 1 )

United States Patent 6,322,396

Kuan November 27, 2001

Modular communication connector

Abstract

A modular communication connector comprises an insulating base, a shielding member, a terminal, a contact terminal, and a battery-charging terminal are accommodated in the insulating base respectively, wherein the insulating base is a flat cuboid, which have a terminal receiving slot, a contact terminal receiving slot, and a battery-charging terminal base are disposed transversally on the insulating base, the shielding member is enclosed the insulating base and defines a charging slot with the battery-charging terminal base, on which the charging slot has an position-identifying portion that is a deviated bracket forming inwardly the battery-charging slot by pressing, so as to prevent incorrect installation with a DC connector.

Inventors: Kuan; Chi-Te (Shulin, TW)

:acgi/nph-Parser?Sect2=PTO1&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&d=PALL&RefSrch=yes&Query=PN%2F6322396

Fig. 2: Screenshot of Text Editor's User Interface.

## 5. Conclusion

In this paper, text editor with database interface is developed and proposed. This text editor enables the user to view the database metadata through the text editor itself. It also enables the user to select the necessary contents from the document and upload the contents onto the database by providing database interface which is unique of its kind. This application may be used in areas where extensive document analyses are performed.

## 6. Future Scope

To enable the text editor to open different file formats, automatically highlight the patent text information in the text editor when user maps the text information to the fields of the relational database. To enable the text editor to open pictures, other piece of data for uploading the relevant content into the fields of the relational database table.

## 7. References

[1] Embley. D. W. and Nagy. G., “Behavioral Aspects of Text Editors,” Computing Surveys 13, March 1981.

[2] David K. Farkas and Steven E. Poltrock, “Online Editing, Mark-up Models, and the Workplace Lives of Editors and Writers”, IEEE Trans. Prof. Comm., VOL 38(2), JUNE 1995.

[3] George Reese, “Database Programming with JDBC and Java”, O'Reilly, June 1997.

[4] R-Databases Special Interest Group, “A Common Database Interface (DBI)”, 26 August 2002.