SQL-Ajax Client a Tool Managing the Database of the Google Web Toolkit Framework

Chayaporn Kaensar

Abstract—**Database management system (DBMS) is widely used today and more are being developed. We have found that these DBMS tools provide a variety of formats on command lines, Window GUI, Web-based format and so on. However, most of them have several problems. For example, users must have a deep understanding of the code before generating commands within a database. Most tools are tied to one specific company and their functions cannot interact with other systems. This creates a lack of flexibility when working within various DBMS types. Sometimes, they do not have a standardized design, are slow refresh the output screen and are also not open-source. Thus, we are presenting a tool for translating actions and displaying SQL commands and data reports in real time simultaneously while providing functions to connect to more databases. We developed a tool, called SQL-Ajax, based on the Google Web Toolkit Framework (GWT). We include Java Servlet, Java Server Page (JSP), AJAX and GWT-Remote Procedure Call (GWT-RPC) libraries to create rich internet applications and to make web pages more dynamic. This tool enhances performance because it supports client-server communication without reloading an entire page. In summary, we list issues to improve system capability and discuss some future task.**

Index Terms—**SQL tool, google web toolkit (GWT), GWT-RPC (remote procedure call), AJAX, java**

I. Introduction

Relational Databases are widely used in today's applications. We have found that business in much of the world depends on database technology such as Transport, Stock Control System, Production Engineering, Tourism, Education, Healthcare, and Government Administration and so on. Many businesses regularly make database transactions to store, retrieve, sort, and display information as well as keeping records though DBMS Software. And often, most software comes with certain tools to help users manage data within a database such as SQLDeveloper, phpMyadmin, phpPgadmin, phpOracleAdmin etc.

Although there are many available applications to support Database Management, These tools have limitations. For example they usually are not open-source and they provide interface only with a specified kind of database, called platform-dependent. And some tools use Synchronous protocols that would make user wait for result.

Moreover, there are also problem with creating a SQL, which is short for Structured Query Language. SQL is a semi-declarative language used by applications to access information stored in relational database systems. Sometimes, problems are created for novice users with limited experience in SQL. They have trouble creating the right command for retrieving specific rows or joining data from tables they want because there are many common rules that must be considered while writing any SQL statement.

Thus, we'd like to propose the SQL-Ajax Tool based on the Google Web Toolkit Framework (GWT), which is a open source Ajax development tool for creating high-performance web applications using Java Programming Languages. We provided a simple graphical interface and mapped each action to a specified command for the user. It can also teach and guide users to understand the semantics of the SQL languages. That is to say, users can view the database or table easily by clicking on the dropdown menu list. At the same time, they can also view the results or the data tables in a simple format simultaneously on a local web page to facilitate selecting, filtering, sorting, grouping and joining data across web services without knowing SQL.

To do so, we used Java Servlet, Java Server Page (JSP), GWT Libraries and Java Script to handle the presentation issues of the SQL Input page and the output screen for front-end design. We also provide helpful information to guide and explain each command, which may help the users understand the command's meaning and definition. GWT-RPC (Remote Procedure Call) mechanisms are employed to make a call to the server-side code and share data between the client and the server, resulting in an application with greatly improved performance and reduced web server load [1], [2].

In addition to managing the database, we imported JDBC API Libraries to connect to a wide variety of relational databases and perform database transactions. That is, the user is able to access database table data from this tool using database specific JDBC drivers and connection pools. [3]

This paper is organized as the following form follows: Section 2 presents the literature and history of software technology; Section 3 present main concept and technology Section 4 describes the system architecture; Section 5 explains the main aspects of the construction of the system framework; Section 6 presents implementation issues to improve system development; and, finally, Section 7 presents a brief conclusion and discusses specific issues of pertaining to this research project.

II. Related Work

Although SQL is simple syntax, it is difficult for users to learn SQL and its underlying concepts, especially when

Manuscript received July 20, 2012; revised September 5, 2012.

Chayaporn Kaensar is with the Faculty of Science, Ubon Ratchathani University, Thailand. She is now with the Department of Mathematic, Statistics and Computer Science, Faculty of Science, Ubon Ratchathani University. Warinchamrabb District, Ubonratchathani Province Thailand. (e-mail: scchayka@ubu.ac.th).

considering its more advanced functionalities. A lot of research has been conducted in an attempt to identify the reasons for these difficulties. For example, [4] discovered its declaration nature is very complex. While [5] noted that there are no steps to sequentially create a command and users are not able to easily follow SQL. And [6] identified that the notification or error messages from RDBMS are not insightful and do not provide thorough descriptions.

Thus, we will present solutions as to how to successfully manage the relevant DBMS tools: The first era of SQL tool is based on the command line. For example, [7] the stand-alone command line interfaces with and is installed with Oracle Database Installation. This tool can format, perform calculations on query, examine tables and perform database administration. But, it is important to note that this requires the user to have a full understanding of the unformatted SQL Command display output.

Next, most tools usually avoid prompts on the command line and the use Graphical User Interface (UI) instead. This simplifies the processes of the many functions that support the database tasks. The tool can browse objects, run statements, view results or even perform administrator tasks. Although, this could introduce risks on the backend because there might be unused functions that are unfamiliar to general users. Moreover, most of them are platform-dependent because they provide interface only with a specified kind of database such as [8]-[10].

Later, a new era for SQL Tools started. One of the added features supports different DBMS products and proposes a framework for flexible tools. For example, the Query By Tree Structure [11] helps users build SQL and retrieve information through the web interface on the net. The server manipulates data and the client provides query generation and interactive formatting though Java Applet. It is simple and fast because the restructuring of the tree happens as the query is taking place. Sometimes, the client may not have Plug-In and JRE Requirements for their browser and large applets may be especially slow when loading.

The PJ-SQL Browser [12] provides CMS Software Framework and python libraries to handle applications. It is open source and fast but there are still some problems with displaying simple SQL and Output Data because adequate functions are not provided to translate their code and display the output simultaneously.

Even so, a few tools use visualization to explain the operations and transform data. But, this is still under construction and the systems are being improved because some required plug ins or modules are not widely encouraged by the browser and they do not adequately support large data sets such as [13], [14]

We also survey tools for managing database systems and we have found that most of them run though browser and support tasks for relational databases such as SQL Developer [8], phpMyAdmin [9] and phpPgAdmin [10] but these tool provide functions for their platform only.

In summary, we present SQL-Ajax Tools; a system that uses a simplified interface for supporting learning and understanding of SQL Languages. There are many features. First, it use simple and more interactive with users and helps explain in detail how to operate the date query and convert them into SQL commands immediately due to both SQL command panel and output data panel can be shown simultaneously. The benefit is to help users eliminate common SQL mistakes. Second, it is faster and more dynamic because we implement a system based on java software development framework that makes writing AJAX applications easier. This is called the Google Web Toolkit (GWT). We also used GWT-RPC (Remote Procedure Call) to handle client-server communication. Its mechanism helps improve performance and reduce the use of the web server because UI appearance was moved to client while we remained business login on server. And third, it is platform-independent because it can access many databases running on any machine though pure JDBC APIs and XML configuration.

III. Concept and Technology

The construction of this system involved efficient concepts and technology related to SQL standards and technology for frameworks, as follows

A. Structural Query Language (SQL)

SQL stands for "Structured Query Language" which was first created by IBM in 1975. It is a query language used for accessing, modifying, inserting or deleting information in a database. However, it is also used for defining and manipulating databases or for establishing or revoking permission on database objects.

SQL is very important for web applications because it is commonly used for database development which is comprised of common commands including "insert," "update," and "delete." However, in our system, we decided to turn our attention to the selection query only.

In this section, we describe methods and rules of the SQL basic structure commands within the systems as (1) [15]

SELECT column
FROM tablename JOIN tablename
[WHERE condition]
[GROUP BY column]
[HAVING aggregatecolumn condition] (1)
[ORDER BY column]

1) SELZECT

The select list operator specifies the columns that users want to retrieve from the database. Users can specify and define the object names followed by a period and the asterisk (*) to select all columns from table. DBMS will then return a set of columns in the order in which the columns were specified when the object was created.

2) From join clause

The form operator specifies the table that will be used for the selection. In the case of Join, multiple tables may be combined, merging rows according to a user-provided join condition. In our system, we focus on “Equi Join” which uses only equal comparisons in the join-predicate statement. We can also make CROSS JOIN (aka Cartesian Product) in order to produce a combination of rows for each table as well as joining all tables. .

The system will automatically fill the schema name in

front of the related column and link the common key according to the user definition. The system will automatically create SQL commands and display all possible result sets

3) Where

This clause is used to extract only those records that fulfill a specified criterion. In this paper, we provide several comparison operators such as "=" (equal), ">" (greater), "<" (less), ">=" (greater or equal), "<=" (less or equal), != (Not Equal To) and % (Like). When a user selects the key, and if matches are found, the corresponding row is immediately selected and put into the output table.

4) Group by

These statements summarize data over a group of rows from the database. It also can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns, where each subset contains all the rows of the original table that have identical values on one or more target columns chosen at input. The system will provide simple aggregate functions, such as *SUM(), MIN(), MAX(), AVG()* and *COUNT()* in the select list and in the HAVING clauses

5) Having

This operator is used to perform an action on groups created by GROUP BY. It specifies a search condition for a group or an aggregate function and can be used only with the SELECT statement. According to Standard SQL's Rules, it is typically used in a GROUP BY clause. So we do not permit the HAVING clause to name any column not found in the GROUP BY clause unless it is enclosed in an aggregate function

B. Google Web Toolkit (GWT)

GWT (Google Web Toolkits) is a new technology related to the creation of rich AJAX applications, which is a widely used technique to create such web pages and to support more flexible content presentation and thus enhance page performance.

They consist of class libraries (ex. JRE emulation libraries, GWT Web UI) and development tools (ex. GWT-Java to Java Script Compiler, GWT Host Web Browser) as shown in Fig 1. [17]

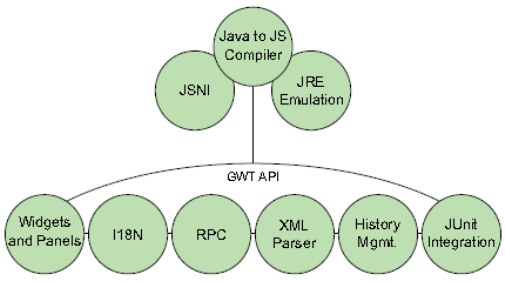

Fig. 1. GWT component overview

There are many benefits of these frameworks. They improve system performance significantly. That is, they do not need to obtain new HTML pages from the server when they want to make UI changes. They can obtain the data from the servers while executing and updating specific parts of the UI.

To do so, one of the techniques that can be used is the GWT-RPC (Remote Procedure Call Mechanism). It makes it easier for the client and server to pass Java objects back and forth over HTTP. Furthermore, it is easy to maintain the system because RPCs give you the opportunity to move all of your UI logic to the client and leave business logic on the server. [1], [2], [16]

IV. SQL-AJAX TOOL ARCHITECTURE

In Fig 2, when a user visits the application though a web browser, they have to know which database address to use. That is, they will fill connection form. From there, the data will be sent to the server to import the database and analyze the client request

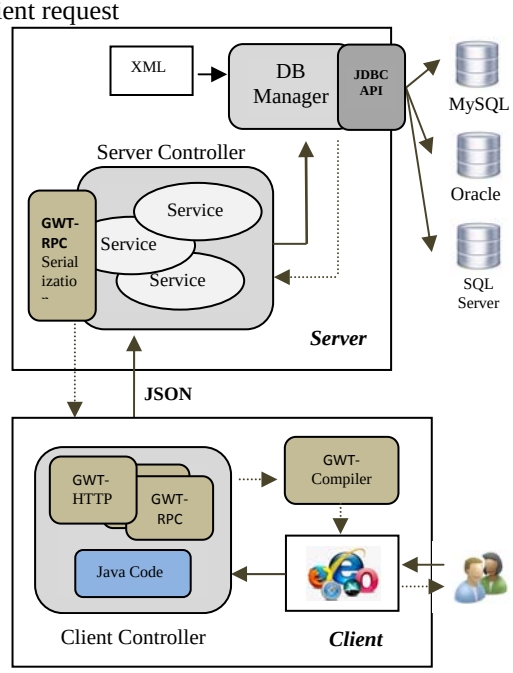

Fig. 2. System architecture

And when users try to make an operation via browser, the client controller (Java Code) will convert action and create the proper SQL command and send them as JSON format (JavaScript Object Notation) to the server side.

To handle communication, we employed the GWT-RPC mechanism to automatically invoke callback methods to the client side and generate the data format and customize the scripts to serialize and transmit via GWT-HTTP.

That is, when the server controller receives any request and encode data in JSON format via GWT-RPC Mechanism, A relevant service (java servlet) will be called. It next performed task and retrieved data in database according SQL command using relevant JDBC APIs.

To make the connection, we first load DBMS driver data from an entire XML file into memory when system was started. From there, we open a connection when user log in success and close a connection in case of logging off.

At last, if the RPC is successful, then asynchronous method on client (Java Code) is called; otherwise an exception will be thrown. And the GWT-Compiler will automatically called to convert the responded data formats to java script in the browser, using JavaScript Libraries and GWT Interface in a form of HTML documents before displaying page to the user.

V. System Implementation

User interface is built using libraries and visual components provided by Java and GWT Libraries which are more complex and attractive [16].

In this section, we will explain the three main functions as follows:

Fig. 3. User interface

A. Client Manager

This module includes three main tasks; First, it converts UI action to the proper SQL from the interface that the user interacts with (see Fig. 4).

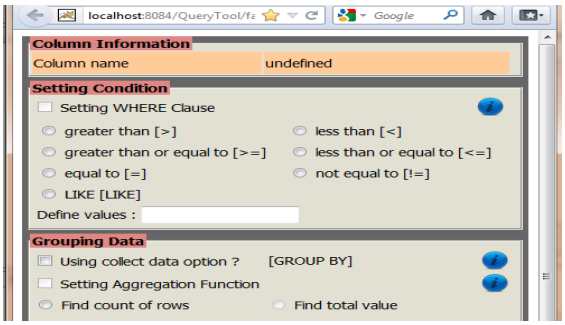

Fig. 4. Action screen (Input)

Then, we used JSON to store the command structure in (2). Java Code will call and get this pattern to validate the right command before sending it to the server side.

```
var statement = {
        "SELECTION":[{
    "select":"SELECT ", "col":col+" ",
    "from":"FROM ", "table":table1,
        "optWhere":optWhere,
"compareValWhere":"'"+compareValWhere+"'",
        "end":";"
        } ]  };
```

(2)

Second, it communicates and responds by pasting data though the network. To do so, we used GWT and JSON integration to achieve client-server communication via RPC calls. The GWT- RPC service is based on Servlet Architecture which is optimized and simplified [1]. Moreover, this mechanism makes it easy for clients and servers to pass Java objects back and forth over GWT-HTTP during a service call.

To do so, we define RPC service interface that extends the GWT RemoteService interface. Within code, we will accept the SQL command and return an array as a String[ ][ ] for GWT object serialization purposes which contain key value pairs of data table to be displayed as shown below

```
…
public interface TableServiceAsync {

void getDataTable (String cmd, AsyncCallback callback);

}
…
public interface TableService extends RemoteService {
public abstract String[][] getDataTable (String cmd);
}
```

(3)

Similarly, we also created a client-side asynchronous interface (TableServiceAsync) which is used to pass in callback objects (AsyncCallback) and notify the caller when an asynchronous call is completed. The client's two stub interfaces example can be shown in (3)

Third, it also manages events on the user interface such as verifying input, responding to user events and handling those events. To do so, we import declarations for ClickHandler and ClickEvent. In detail, we instantiate the TableService using GWT.create() to specify a service entry point and create an asynchronous callback object to be notified (overwrite the onSuccess() method) when the RPC has completed and makes a call as shown in (4).

```
public class SetRule_UIScreen implements EntryPoint {
public void onModuleLoad() {
…..
final Button
checkNum=
new Button ("Submit", new
ClickListener()) {

public void onClick(Widget sender) {

AsyncCallback callback = new AsyncCallback() {

public void onSuccess(Object result) {

// Get the service call result
String[][] reverse = (String[][]) result;
…
}
public void onFailure(Throwable caught) {
// Handle error ….

} ……
```

(4)

In this system, we choose to pass primitive types like String[][] that can be serialized and transmitted via the GWT-RPC mechanism to client side because it is a simple, but robust way to make objects persistent. [1], [16]

B. Server Controller

On the server side, we follow with the implementation of the service methods which are executed and hosted in the servlet container as a service class that extends the GWT's *RemoteServiceServlet class*

TableServiceImpl will take the user-entered string that was passed and perform the task. Once finished, the result string

data is returned

And to enable the client to map the right class in the server, we define GWT modules using XML descriptor files ending with the *.gwt.xml* extension to map web application deployment descriptor and point to the implementation class (ex. *TableServiceImpl*). This file contains inherited modules, an entry point application class name, path entries and so on

The service java file with functionality to handle data can be shown in (5)

```
public class TableServiceImpl extends
RemoteServiceServlet implements TableService {
  ….
 public String[][] getDataTable (String cmd) {
      ….
     if  (database connection = success) {
      // convert resultSet to ArrayList to dynamically
increase element
      ArrayList<ArrayList<String>> table;

      // get a number of column
      int columnCount
=                                  resultSet.getMetaData
().getColumnCount();

      table = new ArrayList<ArrayList<String>>();

      // store result in ArrayList                    (5)
      For (ArrayList<String> row; resultSet.next();
        table.add(row)) {
            …
        row = new ArrayList<String>(columnCount);

        for(int c = 1; c <= columnCount; ++ c)

            row.add(resultSet.getString(c).intern());
      }
      //convert ArrayList to String[][]
      ...
      String[][] output = new String[table.size()][];
      ...
      return  output;
```

C. Database Manager

We let users establish a connection to various database. To do so, user first inform data source they used such as database name, host, port, user, password and other of data pertaining to a corresponding JDBC driver (see Fig 5.)

We employed Java Database Connectivity (JDBC) to bridge the application with the DBMS because they are able to support multiple DBMS drivers while connecting to the various databases [3]

We used an XML file to create system flexibility while creating and maintaining the DBMS drivers. To begin with, we initialized the data and configured all of the files. One of them is connectDB.xml which contains custom tags of database configurations such as vendor name, driver class name, class path, version and description. This xml document was first loaded using xml parser. And the servlet class will next read the relevant target driver to open the connection. In turn, the client send message to server to close the connection when logging out as well.

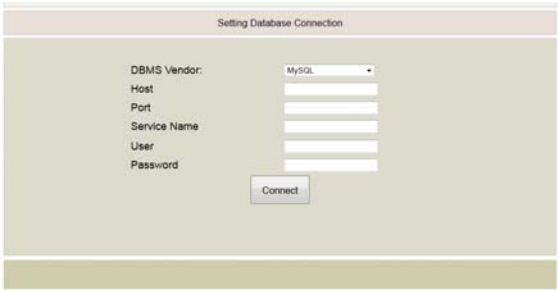

Fig. 5. Log in screen

VI. Open Issue

A. Polling Technique

In this paper, we used Ajax to make the data display real-time. This technique is a simple method for polling and retrieves the latest data and events from the server for a web application.

We essentially poll the server on a regular basis, based on setTimeout Technique. In detail, we first call script to runs automatically, sets up the second interval time and makes the asynchronous Ajax call to the server. Then, finally, sets up the next poll recursively. However, the cost of polling frequently is very high because a new connection to the server must be opened each time when the Ajax method is called. It make server need to buffer the data that it will send to the client until the next request comes from the client.

To solve this problem, we will employ some techniques such as Long Polling and Ajax Push Techniques (for HTML5 WebSockets). It helps simply keep the connection open and make faster response times and appear more responsive [18]. But since these specifications are not yet supported in all browsers, (only Google Chrome and Safari implement these APIs). Thus, we have not yet put into and test with this version.

B. Thread Management

We have found that there was some testing code for checking performance of GWT RPC and Tomcat 6. They found that Tomcat is serving about 1600 concurrent connections using 500 threads with no noticeable degradation of performance.

While these techniques with GWT's pluggable RPC and Ajax. There may be 1600 connections but they don't tie up the limited supply of 500 threads. This means the server saves resources and reduces thread switching - performing better and safer than a server using 1800 threads.

However, this issue is not probably spectacular for a traditional web page but it's a serious situation for application that holds connections, that are constantly transmitting data, open for long periods. In addition, many of these connections use hanging HTTP requests to receive events from the server.

Thus, Using GWT RPC can increase correctness and speed but it is not great for interoperating with the web. [19] However, we will continue to push the server to see and find an upper limit.

VII. Conclusion and Future Work

In this study, we aimed to develop SQL-Ajax Tools for online database management which is capable of supporting the teaching and understanding of the SQL languages. Graphical user interfaces can be used to explain, in detail, query select operations and data transformation from input databases. It helps users to overcome misunderstandings with regards to different operations of a SQL and their functionalities because it can both create SQL command and display data sets simultaneously. To do so, we integrated the Google Web Toolkit (GWT) that allows us to easily develop Ajax based on web applications in the Java language. We also used GWT-RPC (Remote Procedure Call) for handling client server communication. This mechanism allows our system to make server calls and retrieve new data web pages without reloading all of the contents, which speeds up web applications' performance, responsiveness and interactivity. In summary, we conducted a user satisfaction survey and the satisfaction level was found to be good.

As a next step, we are planning to perform an experiment evaluating of system performance and comparing them with the other DBMS tools and also integrating some framework in order to manage databases more efficiently.

Additionally, we will consider how to further develop more intelligent tools and attempt to move this model to the visualization system that can create SQL commands easily.

Acknowledgment

This research was supported by the Faculty of Science, Ubon Ratchathani University, Thailand.

References

[1] R. Hanson and A. Tacy, "GWT in Action: Easy Ajax with the Google Web Toolkit," *1st ed. Manning Publishing Co*., USA, 2007.
[2] D. Strok, "The Google Web Toolkit Shines a Light on Ajax Frameworks," *IEEE Software*, pp. 94-98, 2007.
[3] G. Reese, "Database Programming with JDBC and Java," 1st ed, O'Reilly Media, USA, 1997.
[4] G. Rudiq, M. Orlowska, W. Sadiq, and J. L. Sqlator "An online SQL Learning Workbench," in *Proceedings of the 9th annual SIGCSE conference on Innovation and Technology in Computer Science Education*, pp. 223-227, 2004.
[5] G. Russel and A. Cumming, "Improving the student learning experience for SQL using automatic marking," *The IADIS Cognition and Exploratory Learning in Digital Age conference*, pp. 281-288, 2004.
[6] A. Mitrovic. "A knowledge base teaching system for SQL," *In World* Conference *on Educational Multimedia, Hypermedia and Telecommunications*, pp.1027-1032, 1998
[7] SQL*Plus Quick Start (2005). [Online]. Available: http://docs.oracle.com/cd/B19306_01/server.102/b14357/qstart.htm
[8] SQL Developer (2012). [Online]. Available: www.oracle.com/technology/products/database/sql_developer/index.html
[9] phpMyAdmin (2012). [Online]. Available: www.phpmyadmin.net
[10] phpPgAdmin (2012). [Online]. Available: http://phppgadmin.sourceforge.net/doku.php?id=download.
[11] N. Rishe, K. Naboulsi, O. Wolfson, and B. Ehlmann, "An Efficient Web-based Semantic SQL Query Generator," In *Proceedings 19th IEEE International Conference on Distributed Computing Systems*, 2004.
[12] A. Pierro, F. Cavallari, S. D. Guida, V. Innocente, and A. Beinaravicius, "SQL-jQuery Client a Tool Managing the DB Backend of the CMS Software Framework through Web Browser," *The 17th Real Time Conference*, 2010.
[13] M. Cembalo, A. D. Santis, and F. P. Umberto, "SAVI: A new System for Advanced SQL Visualization," *The 12th Annual Conference on* Information *Technology Education*, New York, USA. 2011.
[14] B. Allenstein, A. Yost, P. Wagner, and J. Morrison, "A query simulation system to illustrate database query execution," In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, New York, USA, 2008.
[15] R. Stephens, "Beginning Database Design Solutions," 1st ed, Wiley Publishing, Inc, USA, 2009
[16] Official Google Web Toolkit (2012). [Online]. Available: https://developers.google.com/web-toolkit/
[17] Tutorial GWT: Installation and Analyze (2012). [Online]. Available: http://www.objis.fr/formation-java/tutoriel-gwt-installation-analyse-permutations.html.
[18] Network Section of the Whatwg HTML5. [Online]. Available: http://www.whatwg.org/specs/web-apps/current-work/#network.
[19] R. Dewsbury, "Google Web Toolkit Applications," 1st ed, Pearson Education, Inc, 2008.

C. Kaensar was born in Ubonratchathani Province, Thailand, on July 29th, 1979. She received a B.Sc. degree in computer science (2000) from Thammasat University, Bangkok, Thailand and in a M.Sc. degree in information technology (2007) from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. She is currently a lecturer of Department of Mathematics, Statistics and Computer Science at Ubon Ratchathani University. From 2000-2006, she worked for Software Development Company in Bangkok, Thailand. At Freewill Solution Co,Ltd., As a Senior Software Specialist, She designed and developed Internet Stock Trading System using Java, C Program and PHP. Moreover, she also responded to assign & track the tasks of writing software, setting up functions, entering data and processing information with development team. And from 2006-2008, she worked for True Corporation Public Company Limited where is a communication conglomerate in Thailand. At True, She is a Software Consultant. Her responsibilities are to create and develop the Mobile Virtual Private Network and implement payment system to support customer service system. Later, she joined Department of Mathematics, Statistics and Computer Science in 2008 as a Lecturer in Ubon Ratchathani University. She responded to teach Database System, Java Programming,VB.NET Programming, Information System Analysis & Design. Moreover, she also worked with the Community-Based Research Project for E-Learning Development, at empowering rural villagers in Samrong District, Ubon Ratchathani Province. Her research interests include intelligent systems, database technology, web application framework and programming and business intelligence field.