

A web application serving queries on renewable energy sources and energy management topics database, built on JSP technology

P.N. Christias

Department of Electronic & Computer Engineering, Technical University of Crete, Greece

D. Kolokotsa

Department of Natural Resources and Environment, Technological Educational Institute of Crete, Greece

ABSTRACT

The present paper describes the development and the specifications of a software package that delivers useful information on scientific topics, bibliography, achievements and activities related to the area of renewable energy sources and energy management. The queries and results are transmitted over the Internet through a simple graphical interface. The package consists of a database designed to host this information and a web-based software application, which serves the dynamic content. With the proper interconnection to the database, the latter becomes a suitable interface to satisfy user queries on this scientific area, as faster and easier as possible.

The database is designed in SQL-92. The web application is a fully graphical dynamic website where the technology of Java Server Pages™ (JSPs) is used to connect, search, and retrieve information from the database. In addition, JSP technology is used in order to transform user input into query language and query results into a comprehensive representation to users.

1. INTRODUCTION

The area of renewable energy sources and energy management is constantly growing during the last decades. Therefore, an information bank on relevant scientific topics, bibliography, achievements and activities, becomes necessary. The scope of the present paper is to describe a dynamic web-based application that attains the above information in a database, offering querying capabilities through a simple graphical interface and features: (a) quick response, and (b) easy-to-use query interface with uncomplicated

criteria requirements, while maintaining the ability to produce the wider possible set of results.

The graphical interface is embedded in a website, since access and search operations can be performed via the Internet. The database is designed in SQL-92 language and implemented in the Microsoft's SQL Server™ 2000, running in a computer separate from the web server computer. A software tool is required to bridge the database and the web application, using the Java Server Pages (JSP) technology. Although Active Server Pages (ASP) can be considered as an attractive approach for the dynamic content development since ASP and SQL Server have a common research background (Microsoft) and operating logic and there are numerous software tools which simplify ASP programming, JSP technology is selected for the following reasons:

- JSP features cross-platform and cross-Web-server support and is platform independent (Sun Microsystems, JSP Manual; Sun's JSP official site). It runs on all Java-enabled platforms and can be enabled within most popular Web Servers. The website of the current application resides in an individual Web Server machine, apart from the Database Server. The Web Server is Apache's Tomcat™, running on a Linux operating environment providing a long-term stable host platform. The Apache's Tomcat web server is selected because it is the servlet container used in the official Reference Implementation for the Java Servlet and JSP technologies (The Apache Jakarta Project site). Tomcat supports the Servlet 2.2 and JSP 1.1 specifications offering compatibility on compiling

and execution (Sun Microsystems, JSP Manual). ASP is fully supported under Windows, but deployment on other platforms is cumbersome due to reliance on the Win32-based component model (Sun Microsystems, JSP Manual).

- JSP uses Java Database Connectivity (JDBC) for data access while ASP uses Microsoft's Open Database Connectivity (ODBC) through Active Data Objects for data access. This interface architecture is also supported by JSP with a special type of JDBC driver – called *JDBC-ODBC bridge* – that connects to databases through an intermediate ODBC driver. The driver in the ODBC approach is single-threaded and may, therefore, perform poorly under a heavy load (Hanna, 2003). Performance drawbacks were not the major reason for rejecting the ODBC approaches either using ASP or a *JDBC-ODBC bridge*. A JDBC driver of type *pure Java direct to database* – provided only by JSP – is used, primarily because it implements a call-level interface. The JSP program uses direct function calls to access the Database Management Systems (DBMS), as opposed to embedded SQL statements. If the use of CallableStatement (see Section 5) is also considered, the automated query becomes more efficient. Secondly, the choice of the ODBC approach would require the configuration of an ODBC data source on the operating system of the Database Server. This would be added to the administrator's maintenance responsibilities because damage to the data source causes loss of communication with the DBMS.
- In JSP there is separation of static from dynamic content. Static presentation templates are encapsulated in external JavaBeans components. News, announcements and graphical add-ons are possible to be added to the website. The changes of the static content do not require knowledge of JSP technology. Only HTML or XML knowledge is necessary. When a designer makes any changes to the presentation template, the JSP page is automatically recompiled and reloaded into the Web Server by the JSP engine.
- JSP can make use of the full Java 2 Enterprise Edition environment. This provides functional compatibility with external code,

ideal for sophisticated and robust applications (Hanna, 2003).

The search criteria are distilled by the JSP software and then forwarded to the database server (as valid SQL variables), to retrieve the optimum results.

In the following section, there is an overall description of the parts and functions of the entire application. In sections 3 and 4, implementation issues of the database and the website are covered. At the end, conclusions and future approaches are discussed.

2. FUNCTIONALITY ISSUES

The website resides in the Apache's Tomcat™ web server computer as mentioned before. The JSP specification is a standard extension of J2EE™'s Servlet Application Interface (API), for dynamic web content serving (Hanna, 2003).

The database server designed in SQL-92 language and implemented in the Microsoft's SQL Server™ 2000 is an independent information source, thus enhancing the overall response time. The JSP code - which handles the users' requests - acts as a client and interconnects to the database for querying, over the Internet.

The structural pattern of the application is shown in Figure 1.

The starting point is the web browser. The user enters the desired search criteria, which are forwarded to the JSP code for processing. De-

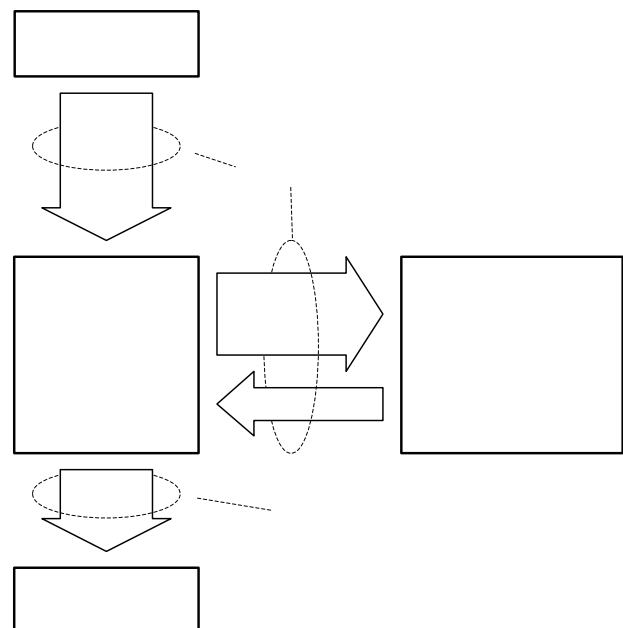


Figure 1: Diagram of the application.

pending on the entered criteria, the JSP code decides which part of the database will be scanned for results. As soon as this operation is complete, appropriate commands are sent to the database for execution. A result set is returned to the Servlet container. By means of the JSP code, the set is filtered and appropriately formatted, in order to be displayed at the browser.

The energy management and renewable energy sources data categories which are stored in the database are listed below:

- International bibliography and editions.
- Articles, scientific reports, dissertations and research studies published in press.
- Educational activities and courses offered by universities and organizations worldwide.

The main objective is to provide combined information from all the above categories, relevant to the query criteria. For optimum performance, the database is designed to manage the data categories based on the following structural grouping:

- Group 1: Bibliography and editions (e.g. title, ISBN), accompanied by publisher and author information.
- Group 2: Published releases (e.g. title, magazine, publishing date), accompanied by author and sponsoring university/organization information.
- Group 3: Courses and seminars features (e.g. name, level), accompanied by hosting university/organization information.

The three groups directives must be preserved during data handling because they characterize the way data are stored and represented to the user. The importance lies in the fact that in this way, the search criteria and the database results are suitably grouped when sent for processing to the JSP code. As a consequence, the JSP algorithm receives input and sends output to the user in the same fixed form.

Another functionality requirement is that the criteria declaration must have a general format while the query should be particularized in specific parameters. The user can insert any criteria without having to specify in which group this might belong.

Moreover, searching categories are available. The user can isolate the query among the following categories: "books", "authors", "pub-

lished releases", "courses" and "organizations". If category "ALL" is selected, the entire database will be scanned. Additionally, the insertion of up to three criteria combined with the logical operators "OR", "AND" is available. Depending on the choice of category, the query is executed on the "attributes" of the "entities". Results are presented along with their relative information, as described in the structural groups.

Bibliography (1st group) and published releases (2nd group) contain all the available printed material. Published releases (2nd group) and courses (3rd group) are considered as organizations activities. If a user is searching for courses conducted in Technical University of Crete, he/she will insert the term "Technical University of Crete" and will select the category "courses". If the category "ALL" is selected, the browser will return all the references to the searched term; either publications (2nd group) or courses (3rd group).

Furthermore, the user is entitled only with query rights. Altering or deleting data is not allowed. The SQL and JSP-based developed software routines are shaped by the functionality specifications stated above.

3. THE DATABASE

The DBMS is implemented using the Microsoft's SQL Server™ 2000 that supports web-based applications on (a) connectivity through URLs, and (b) data exchange matters (Shapiro, 2001). For the proposed DBMS, 10 stored procedures are developed. The database goes through automatic querying, executed by the JSP code. The algorithm "intelligently" decides where to look for results inside the database, taking into account the input coming from the browser. The decision mechanism is analyzed in section 4. The "views" in collaboration with the 10 "stored procedures" satisfy every possible query that the algorithm will have to process. Each time a query is executed, the web browser has to wait for the SQL Server to parse and run the commands, thus increasing the pending time. The total response time T_{res} is the summation of the data propagation time over network from JSP to DBMS, the query expression parsing time, the command execution and data retrieval time and the data propagation time over network from DBMS to JSP. If stored pro-

cedures are used, the parsing is already done; therefore *Tres* is reduced (Shapiro, 2001). Each of the 10 “stored procedures” acts only on a specific part of the “view” to which is assigned to. The initial search is split into smaller ones. Instead of executing a large and complex query expression for each different search scenario, a collection of the available “stored procedures” is selected to do the same. Ultimately, the appropriate part of the database is queried by use of the suitable subset of “stored procedures”. The code is reused with different combinations, depending on the search. The initial query is equivalently performed by smaller specialized queries. This is done without multiple scans of the database since the target “attributes” of the selected “stored procedures” subset are different. As a result, the number of requests is relatively small. Although access to the database is performed via the Internet, the DBMS is chosen to reside in a database server, separated from the web server that hosts the website. This structure eases administration and improves overall performance. The computational burden goes to the database server, increasing the website navigation speed. The separation dictates communication and data exchange between the two servers (database & web), which is provided by the JSP code.

4. THE WEBSITE

4.1 JSP Operating Issues

JSP technology is used to serve database connectivity and dynamic content. Inside the HTML pages there are special code items, called “JSP tags”. The main advantage of the JSP technology is the separation of static and dynamic content (Sun’s JSP official site; Hanna, 2003). Besides a proper web server (as already mentioned), a Java Runtime Environment & Developer kit installation is necessary in order for the JSPs to operate. The database access model is depicted in Figure 2.

The processing is split into “controller” components and “view” components. The “Controllers” filter the HTML requests in order to produce objects used by the “views”. The important thing is that the “views” have no processing logic. They just embed the objects’ dynamic content into their static parts (Sun’s JSP official

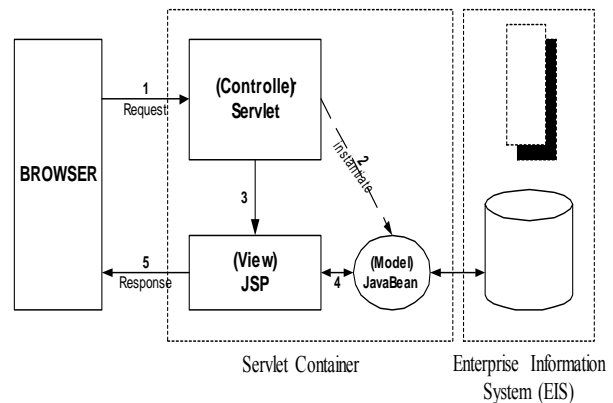


Figure 2: Access Model.

site). There is only one access point (web browser), simplifying maintenance and security issues.

In the proposed application, the benefit of the JSPs lie in the supported access functionality to the database systems, through JDBC. This is an API between Java and DBMS. It includes functions for database connectivity, SQL commands execution and result processing. To establish connection to the DBMS, a JDBC requires a small Java class, called JDBC driver. The steps in a JDBC operation are shown in Figure 3. In network terms, communication takes place through interprocess communication mechanisms (IPC), using the TCP/IP protocol. The JDBC software exploits the network domain as well as DBMS login accounts to gain access (Sun Microsystems, JSP Manual; Hanna, 2003).

4.2 The JSP Code

The JSP algorithm key features are examined in extension below. At the browser output (see Figure 2), there are seven string variables in

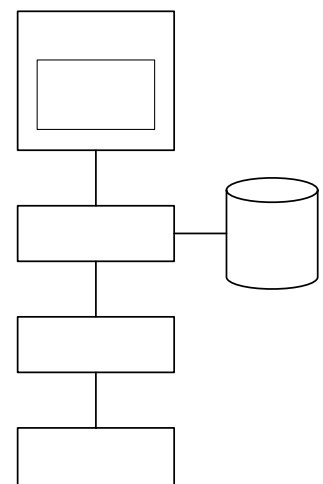


Figure 3: JDBC Steps.

HTML format:

- Three criteria entered in the text boxes of the search form.
- Two logical operators selected from the drop-down boxes.
- The selected search category.
- A hidden string variable, defining whether the search is performed through the advanced search form or the simple search form. It is assigned the "advanced", "simple" values respectively.

Inside the JSP code, the HTML variables values are stored in Java-type string variables by the use of the `request.getParameter().Java` method. Then, they are transformed into SQL variables inside the JSP code. The algorithm decides which part of the database to scan for results by choosing the appropriate "views". For instance, if the selected category is "Authors", the "views" that will be used are those who represent the 1st (books) and 2nd (published press releases) structural groups, because it is impossible to find authoring information in the 3rd group. In consequence, it is decided which of the stored procedures – among those operating on the selected "views" – will be used. For the above example, only the procedures querying on "AUTHORS" "attributes" shall be called for execution. After this operation is complete, the names of the appropriate stored procedures, along with the SQL input arguments, are sent to the DBMS. The algorithm decisions on which "view"(s) and "stored procedure"(s) to use depends upon the chosen search category. The user input is text and is stored in variables of three different formats during the application's execution: HTML, Java and SQL. The choice of the variable type on each case is important because the text must remain consistent while it is exchanged over the parts of the application. If one or two criteria insertion text boxes are left blank, the corresponding string variables are set to *null* and "stored procedures" are executed with the remaining variable(s) as valid input argument(s). If the simple search form is used the second and third search criteria are set to *null* and search category is set to "ALL". The application is built in such a way to find matching results, regardless how general or few the query terms are.

The database returns a result set, comprised

of raw SQL data. Inside the JSP code, the raw data are filtered, stored into string variables and appropriately grouped, in order to be properly displayed at the browser. Depending on which "view(s)" the results come from, the program adopts different HTML format for their display.

5. SPECIAL IMPLEMENTATION FEATURES

Usually, the SQL statements are sent as commands to the DBMS, from the JSP code. Problems arise when the search terms contain escape characters, i.e. apostrophes. While the SQL Server™ successfully compiles such expressions (Shapiro, 2001), the JSP "engine" returns false in a form of an exception report. The reason is that the query expression is cut at the point of the escape character and a nasty error message is shown at the user's screen. Such case cannot be predicted and controlled because SQL statements are constructed on the fly based on the user entered criteria. In the proposed application, as already described, only the names of the stored procedures to be executed are sent to the DBMS. Moreover, the search terms are sent in a form of input arguments, thus avoiding dynamic syntax errors. This feature is based on a special interface of the JSP 2.0 standard, referred as "callable statements" (Hanna, 2003). This interface is used to directly invoke "stored procedures" if the DBMS supports them. By incorporating this feature into the code, the SQL statements execution is passed on to the SQL Server™.

6. CONCLUSIONS

The value of the developed software application lies not only to the functions it offers. Easy maintenance and smooth operation are important parameters implemented.

The functional parts of the application are designed to depend on each other at the level of input-output exchange only. Therefore, a malfunction occurred on one of them does not cause collapse of the rest.

The application is destined to provide any interested Internet user with information concerning renewable energy sources management and smart buildings technology. By adding a login function into the website and a set of learning

rules into the JSP code, an e-learning platform can be produced. The database is predestined to handle this kind of operation level, since special “attributes” - capable of disposing full-text files storage - are incorporated into the database “entity-relationship” architecture.

The software application is available for browsing at the following URL:
<http://ebuilding.serveftp.org>.

It serves successfully the Electric Circuits & Renewable Energy Sources Laboratory of the Technical University of Crete educational purposes with an easy-to-use, reliable and quick responding online information bank.

REFERENCES

- Shapiro, J.R., 2001. SQL Server 2000: The complete reference. New York: Mc Graw-Hill.
- Sun Microsystems, JSP Manual, [Online] Available [http://developer.java.sun.com/developer/online Training/JSP intro/](http://developer.java.sun.com/developer/online%20Training/JSP%20intro/)
- Sun's JSP official site, <http://java.sun.com/products/jsp/index.html>
- Hanna, P., 2003. JSP 2.0: The complete reference. New York: Osborne/Mc Graw-Hill.
- The Apache Jakarta Project site, <http://jakarta.apache.org/tomcat/>