TECNICA DEL NORTE UNIVERSITY

FACULTY OF APPLIED SCIENCE ENGINEERING

RACE COMPUTER SYSTEMS ENGINEERING

THEME

ESTUDIO DE LA HERRAMIENTA DE DESARROLLO JASPERREPORTS EN APLICACIONES WEB CON TECNOLOGÍA JSP

APPLICATION

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE RECURSOS TECNOLÓGICOS PARA LA COOPERATIVA DE AHORRO Y CRÉDITO ATUNTAQUI LTDA.

Technical Report

ENGLISH

AUTHOR

WILSON ANÍBAL CÁRDENAS HERNÁNDEZ

DIRECTOR

ING. MARCELO JURADO

"INVESTIGATION OF THE DEVELOPMENT TOOL JASPERREPORTS IN WEB APPLICATIONS WITH JSP TECHNOLOGY"

(November 2011)

Ibarra - Ecuador

I. JAVA TECHNOLOGY

Java is a programming language object-oriented, developed by *Sun Microsystems* in the early 90's. The language itself takes much of its syntax from C and C++, but has a simpler object model and eliminates lowlevel tools, which often lead to many mistakes.

Java applications are typically compiled into *bytecode*, although compilation to native machine code is also possible. At runtime, *bytecode* is usually interpreted or compiled to native code for execution, although direct hardware execution of *bytecode* by a Java processor is also possible.

Between December 2006 and May 2007, Sun Microsystems released the bulk of their Java technologies under the GNU GPL, according to the specifications of the *Java Community Process*, so that virtually all of the Java software is now free.

On April 20, 2009, *Oracle Corporation* announced the purchase of *Sun Microsystems*, the amount of $ 7.400 million dollars; following the acquisition of Sun, Oracle becomes the guardian of all of some of the most important jewels of open source, including popular MySQL database, the Solaris operating system and development platform for Java.

1.1. Characteristics

The main characteristics that Java technology provides us with respect to any other programming language are:

- Simple
- Object-Oriented
- Distributed
- Robust
- Architecture neutral
- Secure
- Portable
- Interpreted
- Multithreading
- Dynamic

1.2. Java Platform

The Java platform computing environment is originally from *Sun Microsystems*, able to run applications developed using the Java programming language or other languages that compile to *bytecode* and a set of development tools.

In this case, the platform is not a specific hardware or operating system, but rather a virtual machine responsible for the execution of applications, and a set of standard libraries that provide common functionality.

Java Virtual Machine

The Java Virtual Machine is a native program, executable on a specific platform, able to interpret and execute instructions given in a special binary code, which is generated by the Java compiler. Java Virtual Machine is one of the fundamental pieces of the Java platform. Basically it is at a higher level to system hardware on which it intends to run the application, and this acts as a bridge to understand both the *bytecode*, and the system on which you intend to run.

Java API

The set of libraries of language is known as *Aplication Programming Interfaceof Java* and is a set of components that provide different tools for development. The Java API consists of a set of packages of classes that provide great functionality. The core of the API comes with each of the implementations of the Java Virtual Machine.

Java language

Java is a language of pure object-oriented programming, in the sense that no variable, function or constant that is not within a class. You access the data members and member functions through objects and classes.

1.3. Java Platform Editions

There are three basic distributions of the Java platform in an attempt to cover different application environments.

- Java Platform, Standard Edition of Java SE (before J2SE).
- Java Platform, Enterprise Edition of Java EE (before J2EE).
- Java Platform, Micro Edition of Java ME (before J2ME).

II. JAVA SERVER PAGES

Java Server Pages is a Java technology that generates dynamic content for web, as HTML, XML or otherwise.

This technology is a development of the company *Sun Microsystems*. The JSP 1.2 specification was the first to be released and currently available JSP 2.1 Specification.

The JSP's allow the use of Java Script code. It is also possible to use some predefined actions by JSP tags. These tags can be enriched by using tag libraries (TagLibs or Tag Libraries) or even custom external.

2.1. Operation of JSP

JSP's are really servlets, JSP compiles a Java program to the first time it is invoked and the Java program creates a class that starts running on the server as a servlet.

The main difference between servlets and JSP's is the approach to programming. A JSP is a web page with special tags and embedded Java code, while a servlet is a program that receives requests and generates from them a website.

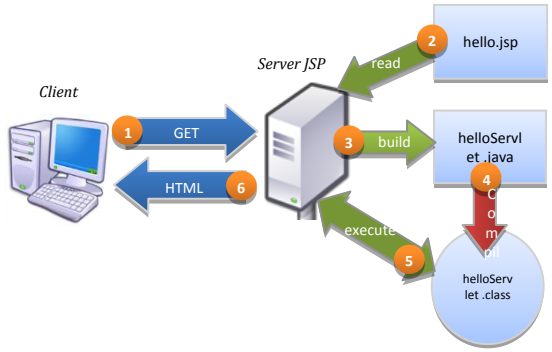

Figure 1.1. Operation of de JSP.

2.2. Components JSP

The components can be used in a JSP page to make dynamic are:

- Expressions
- Statements
- Scriptlets
- Directives
- Labels
- Comments

2.3. Implicit objects

In JSP we have some implicit objects, which allow us access to different information and perform various actions. In JSP we have the following implicit objects: *request, response, out, session, application, config, pageContext and page.*

III. XML (eXtensibleMarkupLanguage)

XML (eXtensible Markup Language) is an extensible metalanguage of tags developed by the World Wide Web Consortium (W3C). It is a simplification and adaptation of SGML and allows defining the grammar of specific languages. So XML is not really a particular language, but a way of defining languages for different needs. Some of these languages that use XML for its definition are XHTML, SVG, MathML and Android.

3.1. Characteristics of the XML standard

The XML standard is defined as "*the universal format for structured documents and data on the Internet*" and its main operating characteristics as described by the W3C itself, are:

- XML is a standard for writing structured data in a text file.
- XML seems to HTML but it is not.
- XML is text, but not to be read.
- XML consists of a family of technologies.
- XML is verbose, but that's not a problem.
- XML does not require licenses.
- Document Object Model (DOM).

3.2. Advantages of XML standard

The main advantages of XML are the following:

- It is extensible, after designed and put into production; it is possible to extend XML with the addition of new tags.
- The analyzer is a standard component is not necessary to create a parser-specific version of XML.
- If a third party decide to use a document created in XML, it is easy to understand its structure and process.
- Transforming data into information, because it adds a specific meaning and associate with a context.

3.3. Parts of an XML document

An XML document consists of the prologue and the body of the document.

Prologue

Although not required, XML documents can start with a few lines describing the XML version, the document type and other things. The prologue of an XML document contains:

- XML Declaration
- Document Type Declaration

Body

Unlike the introduction, the body is not optional in an XML document, the body must contain only one root element, and also indispensable feature for the document

is well formed. The body can contain other elements such as:

- Elements
- Attributes

IV. JASPERREPORTS

The JasperReports library is a very powerful and flexible report-generating tool that has the ability todeliver rich content onto the screen, to the printer or into PDF, XLS, CSV, HTML, XML, among others.

The library is entirely written in Java and can be used in a variety of Java enabled applications,including J2EE or Web applications, to generate dynamic content. Its main purpose is to help creatingpage oriented, ready to print documents in a simple and flexible manner.

JasperReports organizes data according to the report design defined in an XML file. This data maycome from various data sources including relational databases, collections or arrays of Java objects.Users can plug the reporting library to custom made data sources, by implementing a simple interface,as you will see later in this book.

In order to fill a report with data, the XML report design must be compiled first. Through compilation,a report design object is generated and then serialized in order to store it on disk or send it over thenetwork. This serialized object is then used when the application wants to fill the specified reportdesign with data. In fact, the compilation of a report design implies the compilation of all Javaexpressions defined in the XML file representing the report design. Various verifications are made atcompilation time, to check the report design consistency. The result is a ready to fill report design thatwill be then used to generate documents on different sets of data.

To fill a report design, the engine needs to receive the report data. This may come in various forms.Some of this data can be passed in as report parameters. But most of the data will be found in the reportdata source. The reporting engine can directly receive special data source objects from which to get theinformation to put on the report, or can deal itself with a supplied JDBC connection object, if that datais located in a relational database.

The result of the report filling operation is a new object that represents the ready to print document.This one is also serialized for storage on disk or network transfer. It can be viewed directly using theJasperReports built-in viewer or can be exported to other, more popular formats like PDF, HTML orXML.

4.1. API of JasperReports

The following describes the most important classes and interfaces that integrate this library.

- Class *dori.jasper.engine.design.JasperDesign*
- Class *dori.jasper.engine.JasperReport*
- Class *dori.jasper.engine.JasperCompileManager*
- Class *dori.jasper.engine.JasperPrint*
- Class *dori.jasper.engine.JRResultSetDataSource*
- Class *dori.jasper.engine.data.JRTableModelDataSource*
- Class *dori.jasper.engine.JREmptyDataSource*
- Class *dori.jasper.engine.JasperFillManager*
- Class *dori.jasper.engine.JRAbstractScriptlet*
- Class *dori.jasper.engine.JRDefaultScriptlet*
- Class *dori.jasper.engine.JasperPrintManager*
- Class *dori.jasper.engine.JasperExportManager*
- Class *dori.jasper.engine.JasperRunManager*
- Class *dori.jasper.view.JRViewer*
- Class *dori.jasper.view.JasperViewer*
- Class *dori.jasper.view.JasperDesignViewer*
- Class *dori.jasper.engine.util.JRLoader*
- Interface *dori.jasper.engine.JRDataSource*

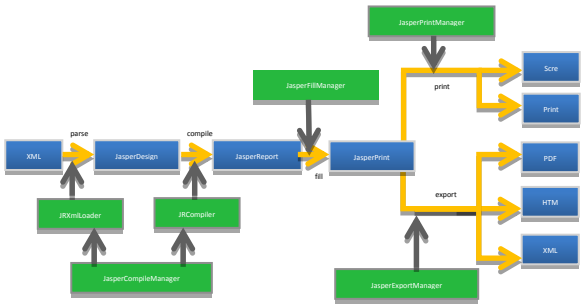

Figure 4.1. Main classes and interfaces of the library.

4.2. Tasks and processes

It describes what needs to know to analyze designs in XML; compile, fill, view, print or export to other formats.

Analysis XML

JasperReports uses the SAX 2.0 API to parse the XML files. However, it is not tied to a particularSAX 2.0 implementation, like Xerces for examples, but instead you are able to decide at runtime whatXML parser you are using.

To instantiate the parser class, JasperReports uses the *createXMLReader()* method of the *org.xml.sax.helpers. XMLReaderFactory class.* In this case, it will be necessary at runtime to set the *org.xml.sax.driver* Java system property tothe full class name of the SAX driver, as specified in the SAX 2.0 documentation.

Compiling the design

In order to generate a report, one has to create the report's design first; either by editing an XML file orby programmatically building a *dori.jasper.engine.design. JasperDesign object,* to pass the compilation process before it is ready to be presented with the data by the reporting engine.

The main objective of the compilation process is to produce and load the bytecode of the class containing all terms in the report. This dynamically created class will be used in completing the report to evaluate all the expressions.

Layout Preview

The JasperReports library is not supplied with an advanced GUI tool to assist in the design task reports. However, there are currently some development tools that together with this library provides a simple and friendly interface for designing reports. The library contains a very useful visual component that allows developers to preview designs reports and see how reports are left.

Filling the Report

The filling process of the report is the most important of all the functionality of the JasperReports library. This represents the main goal of this software component, since it is the process that manipulates the data sets to produce high quality documents. There are three things that must be provided to the process of filling the report:

- Design
- Parameters
- Data source

The result is always a document ready for viewing, printing or exporting to other formats.

Viewing reports

To view reports generated by the proprietary format or proprietary XML format produced by the exporter internal XML, JasperReports has a built-in viewer. This is a swing based component that can be integrated easily with other Java applications that wish to incorporate this functionality without the use of export documents in popular formats, so that they can be displayed. *Dori.jasper.view.JRViewer* class represents this visual component. This can be customized to meet particular application needs, so we could add or remove buttons from the existing toolbar that appears, or make other modifications.

Printing reports

The main objective of the JasperReports library and all the tools for the generation of reports is created print-ready documents. Most reports that are generated are supposed to end or end up in the paper.

We can print the documents generated by JasperReports library using *dori.jasper.engine.JasperPrintManager* class.

Export reports

In some application environments, it is convenient to transform the JasperReports generated documents to other popular formats such as PDF, HTML or XLS. By doing this, you can do that others can view these reports without installing special displays in their systems, especially when sending documents over the network.

There is also a facade class JasperReports for this type of functionality. His name is *dori.jasper.engine.Jasper Exportmanager*and can be used for content PDF, HTML or XLS for documents produced by the process of filling out reports.

4.3. Design report

The report design represents a template to be used by the JasperReports report generator to deliver a document ready for the screen or printer.

Generally, a design contains all the information concerning the structure and appearance of the documents that are generated when the data provided. This information involves the location and contents of various elements of text or graphics to be displayed in the generated document, appearance, custom calculations, grouped data and data manipulation to be performed to generate the documents.

Typically, designs are defined in XML files with a special structure and are subject to the JasperReports compilation process before being filled with data. But they can also be built in memory, using the JasperReports API.

4.4. Report data

During the process of filling the report there are three things that must be provided: the design, the values of the parameters and data source. The parameters and data source are the only source of data that the report generator can be used for filling the report. This information is organized according to the template defined in the design and produce a document ready for printing.

4.5. Sections of the report

The sections are excerpts from the report template with a specific height and width and can contain report elements such as lines, rectangles, images or text fields. The sections are filled several times during the report generation time and make the final document being produced. By declaring the contents and characteristics of a section of the report in XML design, use the generic <band>.

The whole structure of the design is based on the following sections: *<title>, <pageHeader>, <columnHeader>, <groupHeader>, <detail>, <groupFooter>, <columnFoter>, and <summary><pageFooter>.*

Title

This is the first section of the report and this is generated only once during the filling process and will report at the beginning of the final document.

As the first section of the report means that will precede even the page header section.

Page Header

This section appears at the top of each page of the final document.

Column Header

This section appears at the top of each column of the final document.

Detail

For each record from the data source, the engine generate reports about this section.

Column Footer

This section appears at the bottom of each column of the final document. Never extends down to view the content of the text items it contains and will always be a fixed height.

Page Footer

This section appears at the bottom of each page of the final document. As the section of the column footer, footer never stretches down to display the contents of the elements it contains and will always be a fixed height.

Summary

This section is generated only once in the report and at the end of the final document, but not necessarily the final section is generated. That's because in some cases, the foot of the column and/or footer on the last page you can follow.

4.6. Elements of the report

Reports generated would be empty if there are no elements in the design. The elements of the report can show objects as static texts, text fields, images, lines or rectangles, which are placed in sections of the report design to appear on the generated document.

Text elements

Are the design elements of the report that displays static text and text generated dynamically.

Graphic Elements

Correspond to graphic objects inserted into the report design as lines, rectangles and images.

When you add an element of the report in a section of the report, you have to specify the relative position of the element in the particular section and size, along with other general properties of the elements.

4.7. Subreports

The subreports are an important feature for a report generation tool. These enable and facilitate the creation of complex reports and simplify the design work. The subreports are very useful when creating a master-detail report or the structure of a single report is not sufficient to describe the complexity of the final document.

A subreport is in fact a normal report which has been incorporated as part of another report. You can mount the subreports and make a subreport contain another subreport itself, the nesting level is limited.

On the other hand, a subreport is also a special type of report element that helps to introduce a teacher underreporting in the report. There is nothing more to say about subreports, because reports are seen as normal, because it is compiled and full reports as normal. In fact, any report design can be used as a subreport when it is incorporated into another design reports without changing anything inside.

V. DATABASE

A database or database is a set of data belonging to the same context and systematically stored for later use. At present, due to technological development in fields such as computer and electronics, most databases are in digital format, which offers a wide range of solutions to the problem of storing data.

There are programs called Management System or abbreviated Database DBMS to store and subsequently access data quickly and structured. The properties of these DBMSs, and their use and management, are studied within the field of computing.

5.1 Database Management Systems (DBMS)

Management Systems Database (*DBMS*) is a very specific type of software dedicated to provide an interface between the database, users and applications that use it.

The general purpose systems database management is to manage in a clear, simple and ordered a set of data which is then converted into relevant information for an organization.

5.2 Objectives of the DBMS

There are several objectives to be met by the DBMS:

- Conduct pre-defined and complex consultations.
- Flexibility and independence.
- Eliminate redundancy.
- Data integrity.
- Concurrent users.
- Security.
- Abstraction of information.
- Response time.

5.3 SQL Language

Structured Query Language or SQLis a declarative language to access relational databases allowing you to specify various types of operations on them. One of its features is the management of relational algebra and calculus allowing consultations in order to recover, in simple, interesting information from a database, as well as make changes to it.

5.4. Management of DBMS

The database administrators are responsible for the proper functioning of management systems and databases to ensure data availability. Problems or involved in emergency situations, but their primary responsibility is to ensure that no incidents occur. The following are a list of typical tasks of the database administrator.

- Maintenance, management and control schemes.
- To ensure maximum availability of data.
- Resolution of emergencies.
- To ensure the integrity and quality of data.
- Physical design, access and restructuring strategy.
- Performance monitoring and decisions regarding changes.
- Advising end users about the use of the database.
- Control and Security Management.

5.5. Adaptive Server Enterprise

Adaptive Server Enterprise (ASE) is the database management systems (DBMS) Sybase company logo. ASE is a system of database management, highly scalable, high performance, with support for large volumes of data, transactions and users, and low cost, which allows:

- Store data safely.
- Process data intelligently.
- Mobilize data.

VI. SYSTEM DESIGN AND IMPLEMENTATION

The Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda.,is a financial institution located in the province of Imbabura and northern Pichincha, offering the community a variety of products and services. In order to provide adequate care for their partners and customers, the IT Department of Atuntaqui Cooperative Ltd., constantly updates its technology, causing a variety of changes and changes in technology resources.

During my time working and professional in the Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., I have noted that in the institution there is a considerable quantity and variety of technological resources to facilitate the performance of daily activities of their employees, the administration and manual control is tedious and complex, which is why the computer department does not have the clear, accurate and timely, thus complicating the management of resources.

For this reason, in collaboration with the computer areaof the Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda.,has decided to bring this project to implement a system that optimizes the process of administration and control of technological resources .

6.1 System Description

After a study of the processes involved in the administration and control of technological resources of the Savings and Credit Cooperative Ltd. Atuntaqui, has seen fit to classify them into five main modules. Same as were optimized for better management of technology.

Figure 6.1. General scheme of the system (SARTE).

Administration

The system is mostly based on parameters set by the administrator of the system, same as may be configurable via the information system.

Resources

Once the configuration parameters are essential for proper functioning of the system, these allow us to enter, edit and organize the information on technology resources, according to the categorization established by the administrator.

Maintenance

One of the main needs and motives by which proposed the implementation of the system of management of technology resources, was able to record and keep track of all maintenance and new developments arise during the lifetime of resources.

Transfers

During the study the process of resource control and the process of gathering information, we determine that there is lack of information regarding the transportation and equipment custodians, making the management process. This is why it was proposed to implement a module movement of equipment to help us with this activity.

Reports

The main reason that raised the implementation of this system in the Cooperative is able to have clear, accurate and timely submitted all the new technological resourcesin order to meet the institutional reality in regard to technology and to take prompt better decisions in the administration of these resources.

VII. CONCLUSIONS

- The implementation of a computer system at the Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., greatly simplifies the process of administration and control of technological resources by automating repetitive activities in an organized record of all the new products. And with the launch of the system in the Cooperativa de Ahorro y Crédito "Atuntaqui" Ltda., currently available accurate, timely and clear existing technological resources and their respective modifications and adaptations.
- The use of a computer tool for designing and reporting on web information systems, such as JasperReports, decreases the cost of the project since it allows to save large amounts of money for the purchase of licenses, while facilitates the design of reports that display information systems development by reducing the time of the project.
- JasperReports is an open source tool for producing powerful documents ready for print, display or export to other formats like PDF, XLS, HTML, among others, that can be incorporated into any information system developed in the Java platform.
- The development of applications using Open Source tools, it allows other programmers to modify the code of the systems, allowing suit your needs.