

CONFSYS: The Cindi Conference Support System

Zhengwei Gu

A Major Report

in

Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada

March 2002

© Zhengwei Gu, 2002

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the major report prepared

By: Zhengwei Gu

Entitled: Electronic Conference System

Master of Computer Science

compiles with the regulations of the University and meets the accepted standards with respect to originality and quality

Signed by the final examining committee:

D. Goswami Examiner

Bipin C Desai Supervisor

Approved by _____
Chair of Department or Graduate Program Director

20 _____
Faculty of Engineering and Computer Sciences

Acknowledgement

I would like to take this opportunity to give my true appreciation to my supervisor, Professor Bipin C. Desai, for his helpful advice in system architecture, his experience in conference logistics and his technical support in the computing and real environment. With his help, I was able to get the additional package from Professor Michael Ley, which is a key component in this project. Thanks for his time, most often during late evenings which usually were the only times I could meet him: he was more than generous with this unpaid and unappreciated “overtime”.

I also want to give my thanks to the Professors and staffs in Computer Science department at Concordia University for the wonderful courses and service. Through these courses I got the fundamental knowledge and skills to develop this project

Thanks also should go to Hui Guan, who was the first developer for this project and some functions are based on her development.

Finally, I have to say thanks to my wife, Joy Jin for her understanding, continuous support and encouragement. I should apologize to Ziyan and Candia, my lovely daughters for not able to spend enough time to play with and guide them, especially during the weekends and holidays. I am looking forward to make up for this, now that this is behind me.

Abstract

The Cindi Conference Support System

Zhengwei Gu

This project describes the design and the implementation of a conference support system. By using this system, an author can submit her paper; the program committee members can download and to review these papers; and finally the program chair can make a decision for there papers, all on-line.

This project is a Java based application. Most of functions of this project are developed by using Java servlet technology. Through the Java servlet, a client can communicate with MySQL, an easily available Open Source database server.

This application also uses Linux as the operating system, Apache as web server and Tomcat as servlet engine. Additional third party components are also used in this application.

Finally, the major important interfaces and codes for key algorithms are presented.

Contents

1.	Introduction	1
2.	Architecture and Configuration	3
2.1	Users of System	3
2.2	Overall Architecture	4
2.3	Software Components	5
2.4	System configuration	7
2.4.1	Install and configure Tomcat in Linux system	7
2.4.2	Configure Tomcat into Apache web server	9
2.4.3	Setting servlet classes correctly	10
2.4.4	Setting up JDBC driver	11
2.4.5	Setting Java Mail	11
3.	Implementation	13
3.1	Key algorithms	13
3.1.1	Paper Uploading	13
3.1.2	DBLP system and its convert to MySql table	15
3.1.3	Paper allocation algorithm	16
3.1.4	Database connection strategy	18
3.2	Major functions explanation	19
3.2.1	Topic control	19
3.2.2	PC member control	20
3.2.3	Setting up parameters for automatic allocation.....	21
3.2.4	PC member registration	22
3.2.5	Paper uploading function	24
3.2.6	Paper allocation	26
3.2.7	Setting up the Start Time for PC review	28
3.2.8	PC member download file	29
3.2.9	PC paper review	30
3.2.10	Paper decision	32
3.3	Application Security	34
4.	Database Schema and Table Description	36

5.	Testing Plan	42
5.1	Test from user point of view	42
5.2	Test from technical point of view	43
6.	Conclusion	44
6.1	The Conference Support System, a useful system	44
6.2	Reliable Open Source software	44
6.3	Java, a good programming language	44
6.4	MySQL, a good database server	44
6.5	Tomcat, a good Servlet Engine	45
7.	Future Improvement	46
	Reference	48

1. Introduction

The Conference Support System(Confsys) is an electronic paper collecting and allocating system which allows an author of a conference to submit paper to the system via the Internet and General Chair (GC) to allocate the submitted papers to Program Committee (PC) members via the Internet and finally Program Committee members to download and review the allocated papers via the Internet. The major part of this system is a web-based application, which uses a 3- tier framework. In client side, the web browser is the only tool to allow all kinds of users including paper authors, PC members and GC to communicate with server. It represents the visual aspects of this system. It also provides all the functions with human interfaces. Through the interface a user can use a number of the functions of the system. The server side stores the business logics for the system. By reviewing the client's request, the server will give client an appropriate response. In another word, the web server and servlet engine works together to handle both URL (Uniform Resource Locator) and URI (Uniform Resource Identifier) from client side. Finally the database supports the server and keeps all information sent by clients. In additions, Confsys has a DBLP sub-system, which contains a huge number of XML files and is maintained by professor Michael Ley of University Trier in Germany. Each of these files represents a paper in conference or journal, or a book. Each file has the author name, paper title and so on. By using the author information, Confsys has a function to assist in the assigning papers to PC members. During that procedure, the system will avoid allocating a paper to a PC member who was a co-author of one of the paper's authors. In another word, Confsys will ensure that a PC member and paper author were not co-authors in the past based on the information in DBLP. The objective of this is to

let PC members give a fair and impartial review to each paper and avoid conflict of interests.

2. Architecture and Configuration

2.1 Users of System

The system supports three groups of users: authors, Program Committee members and program (co) chair (General Chair). We abbreviate these three groups by the term author, PC and GC. For the clear and easy notation, we assume that authors and GC are female and the PC members are male. Below, we indicate the needs of each group and how these are implemented.

First group of users is the paper author who can submit paper through Internet and can receive review result and final decision via an email. An author normally uses the function of “Paper Submit” in the system to do related operation. Second group of users is PC members who can download papers, evaluate paper and submit the result to GC via the Internet. A PC member normally uses the functions of “PC Member” in the system.

The last kind of users is GC member who can download files, allocate papers, and make final decision on papers and a number of other operations. First user group is the largest user group, normally 100 to 400 people for most average conferences. Second user group is about 30 – 60 people. There are only a few people in last group, say 1 or 2.

For the functions of the first and second user group, we need to consider concurrency, synchronization, database lock, etc. Functions for last user group have very complex in business logic, such as allocation algorithm, but requires minimal multi-user consideration.

2.2 Overall Architecture

Confsys is a three-tire web application as shown in Figure 1. It includes presentation layer, business layer and database layer.

Following is brief explanation of this architecture.

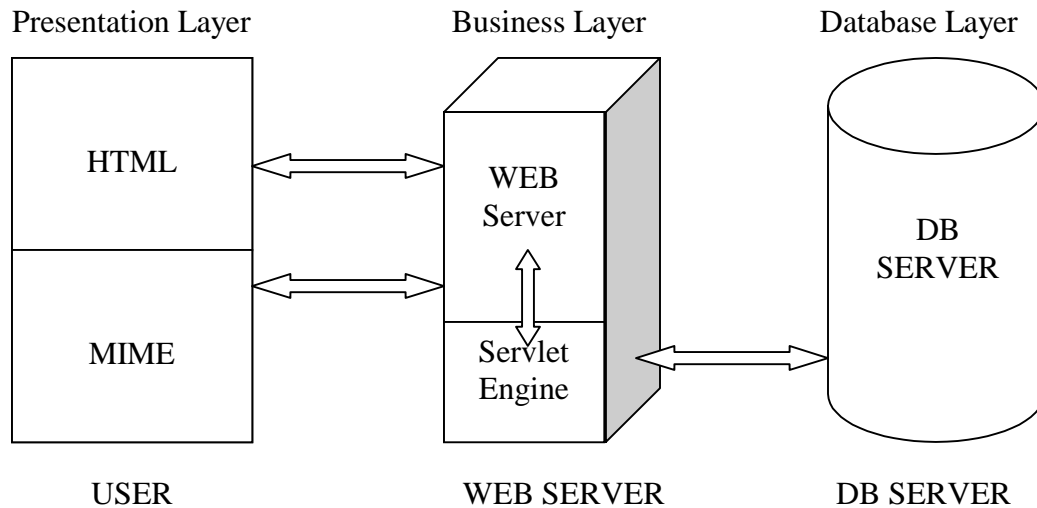


Figure 1

Since this is a web-based application, it makes sense for users to use browsers (IE or Netscape) to communicate with the server. Other functions to be added are the ability to upload papers by authors, and the ability of the program committee members to download the papers they are to review and submit their reviews on them. They also need to debate the reviews and if needed modify some scores in case of conflicts in the reviews of a given paper.

The application allows the users to upload a file from the client side to server side. In the presentation layer, it includes not only HTML format but also MIME format. In order to handle non-HTML package sent by clients using browser, it needs an additional tool in the server side. Normally this kind tool is called a multi request handler. In this system

two java classes, MultiRequest and MultiRequestServlet written by Jason Pell[12], are added.

Since this application uses Java Servlet as middle ware, a servlet engine needs to be added in server side. This application uses the Tomcat (an Open Source Software) as the servlet engine. Though Tomcat has certain function of a web server, it is not advisable to use Tomcat as both a web server and servlet engine because its performance as a web server is not as good as that of a dedicated web server, such as Apache. This application adopts Apache as the web server and Tomcat servlet engine to be an extension of Apache to support Java servlet classes. If users try to connect to a JSP/Servlet URI, then this request will be passed to the Servlet Engine. Servlet Engine will execute one or more servlet classes and return the result to Web Server; the Web Server returns the result to client's browser.

The back end is the database server. MySQL is used as the DB server in this application simply because it is Open Source free system and is suitable for middle size applications. A user query of the database is handled as a request, which is passed to Servlet Engine; a Java class will perform this task for the query. By the way it is fairly easy to connect MySQL with Java or C/C++.

2.3 Software Components

As mentioned above, this application includes web, database, middle ware and some third party components to support the functions of the system. Besides these, the application also uses a specific support database, DBLP, in XML format. All these are based on a Linux OS. All these products are sponsored by the Free Software Foundations. No software cost is needed in this system. Following is the details about these products.

First of all, the web server is Apache, which is a very common, stable and effective web server. Apache also has a Linux version in addition to Unix version. In this system, a Linux version Apache is used as the web server.

Second, the middle ware is Tomcat servlet engine, which is a well integrated with the Apache web server. Since more and more web applications are beyond only displaying static HTML text. A lot of web extensions are added to the web server directly as a function of web server or as a module, which can be included into the web server.

Tomcat belongs to the second type. With Tomcat servlet engine, a java class file can be executed on the server side when someone uses certain function of a web application. The idea is similar to CGI (Common Gateway Interface). The major difference is that CGI handles different client calls in different instances while servlet handles different client calls in a single instance using different threads. Once a servlet is executed, executable binary code is always in memory until the servlet engine shutdowns. So, it is more efficient than CGI.

Third, Mysql database is a very popular Open Source database. It uses the SQL92, a milestone in database query language, as a standard. Mysql has robust interface for C/C++ and Java. For C/C++, it has several library files, such as libmysqlclient.a. Once a C/C++ program includes these library files and points to correct socket file (mysqld.sock), it can communicate with the Mysql easily. For Java, Mysql has built JDBC classes, which support Java to connect to Mysql.

Fourth, DBLP is a XML database that contains author and title information for most computer related conferences, journals, Ph.D thesis and books. DBLP is maintained by Professor Michael Ley of University Trier in Germany. As of December 2001, DBLP contained about 250000 XML files. Each one represents a paper in conference or in

journal, or a book and contains the author name, paper title, publication year, etc. In our system, the algorithm for allocating a paper to a PC member uses this information to avoid the scenario where the paper's author and the PC member were the co-authors in any published paper, book or journal.

Since an author will upload her paper to this system through the web server, the information the author passes to the web server is not a simple HTML file. It is a MIME file. To detach an attached file from HTML, a third party component is required. This system selects a java class called MultiRequest which is provided by Jason Pell as a free software. To be used as a Java servlet another class called MultiRequestServlet is needed. With these two classes, Attached files in any format can be extracted. It solves the problem of MIME uploading

2.4 System configuration

2.4.1 Install and configure Tomcat in Linux system

To install Tomcat, first we need to download certain source files. Since Tomcat is Open Source free software, it was developed by a large group of developers. Hence there are a lot of versions of this product (ex version 3.2.x; 3.3.x; 4.0.x). In this project, we use version 3.2.x. For each version, there are also a number of installation files for different operating system. We use Linux installation files. Even for Linux or Unix, there are still three modes. First is a RPM mode that is a self-extracting file, which makes for an easy installation process. Second is a binary archive mode that has a small size. After it is unzipped, all executable files are there. The last one is an archive file with the source code. Once we unzipped the download file, it must be compiled and installed. Each mode has its advantage and shortcoming. First one is easy for installation, but less flexible. In

this mode, the executable files will be put into predetermined standard directories. In second mode, the executable files and support files are included, but not the source codes. In case we need to add something into Tomcat (ex an ssl security layer), this mode makes it difficult. Third mode is very flexible mode. We can add additional features to Tomcat just by compiling them with appropriate libraries. However, the difficulty of compilation varies with platform. The reason being that is different platforms may have different core libraries and also the libraries directory structure varies. So unless it is necessary, first or second mode will be a good choice for installation.

Tomcat itself is a Java application. To install Tomcat, we need to make sure JDK has been installed in the server. Without JDK, we can't install or run Tomcat. Normally, Tomcat will be installed in /usr/local/ directory in UNIX and Linux system. In our case, we use Linux binary mode. After Tomcat is installed, we can test it independently. We go to Tomcat /~/bin/ directory and run "startup.sh". Then we can use <http://server:8080>. If everything is correct, there will be a Tomcat homepage. From this homepage, we can test servlet and JSP pages. If every thing is OK, it means Tomcat now is running as an independent web server and servlet engine. Due to the performance problem, Tomcat is not very good as a web server. We must combine Tomcat with a web server, Apache web server in our case. Hence, we only use Tomcat as a servlet engine, a web server extension.

2.4.2 Configuration of Tomcat into Apache web server

Apache web server has a long history with UNIX system. It started as a UNIX application and was developed in C. Since Apache needs to use a lot of libraries, it can be compiled in either statically or dynamically. With static linking, an application can have

better performance than dynamic linking but makes application bigger in size. Hence the choice is always a compromise between performance and resource.

The choice also depends on the type of Apache web server used. We can use a command called “`httpd -l`” to determine this. After we run this command, we get the following result:

compiled-in modules:

`http_core.c`

`...`

`mod_so.c`

`...`

If we find that `mod_so.c` is here, it means this Apache is able to combine some `*.so` file as extended module of Apache server. In our case, we can download a file from Tomcat site called `mod_jserv.so` which is a bridge between Apache and Tomcat. After Apache includes this module, it can communicate with Tomcat without problem. How to include this file? We just need to modify Apache configuration file called “`httpd.conf`”. For the detail see the attached digest of “`httpd.conf`” file.

If we do not find `mod_so.c` file in the result, then we have to recompile Apache to include `mod_so.c`. In order to do so, we can use following command in Apache source directory:

```
./configure --prefix=APACHE_DIR --enable-rule=SHARED_CORE --enable-module=so.
```

Then use “`make`” and “`make install`” in Apache source directory.

In either case, we may find certain `.so` files do not work properly. We need to use `apxs` command in Apache bin directory to recompile these `.so` files. Following is an example:

`/~/bin/apxs -o mod_jserv.so -c *.c`. If this operation fails, it means that the *.o files have been generated but *.so file has not been created. In this case we execute following command:

`gcc -shared -o mod_jserv.so *.c`. Then we copy the generated mod_jserv.so file into `APACHE_HOME/libexec/` directory.

2.4.3 Setting servlet classes correctly

Servlet executable files are Java class files. Since servlet is an extension of Standard Java we must install an extended Java package for running servlet. In the case of Tomcat, they are a `servlet.jar` file and its help files. To make these files reachable when a servlet needs them, we can either set Java Classpath to include files or put these files into some special directory, example `JDK_HOME/jre/ext/`. For details, it is best to refer the JDK document because it changes from version to version [11].

Besides this, a servlet class file should be put into a special directory in which Tomcat servlet engine can execute it. Normally, it is in `WEB-INF/classes/` directory. This also depends on how we set it in the `httpd.conf` file. In the application, we give our directory `./home/WWW/webapps/confsys`, an alias, `confsys` and `mount /confsys/servlet as /servlet`. So a URL from a client of the form `~/confsys/servlet/~` will be passed to Tomcat servlet engine by Apache web server. Tomcat servlet engine will look for the `WEB-INF/classes/` directory in `/confsys/` virtual directory and execute certain class file in that folder. Why do we need to do this? By default, the servlet files should be put into `TOMCAT_HOME/webapps/WEB-INF/classes/` directory. Since we need to consolidate our application class files in a certain directory, we create one just for our project. To make web server to recognize the created directory, we need to set in `httpd.conf` as mentioned above. Finally,

we must set these files and directories to be executable for the system users otherwise Tomcat can't execute these class files.

2.4.4 Setting up JDBC driver

To connect MySQL database from Java servlet classes or any other Java classes, a JDBC driver is needed. MySQL has its own driver. It is necessary to download it from MySQL site, uncompress it and put it in any directory. This directory must be included in the Classpath. If the Classpath is to be available for all users, it can be set in profile file inside /etc/ directory. If Classpath should only be available for certain users, settings are needed in the users' shell initialization files. In our application, the user's shell file is the .cshrc files in the user's home directory.

2.4.5 Setting Java Mail

Since there is an automatic mail function in the application we need a Java Mail package for the system. Java Mail is an extension package of Java and is documented by Sun. Although standard Java already has some mail function under "org" package, it is not documented and not even supported by Sun. It is safe to use Java Mail if we need mail function in the system. Like other Java extensive packages, Java Mail needs to be included in CLASSPATH correctly. Once everything is set properly, we can use the functions of Java Mail to send and receive emails. Theoretically, we can use any external mail host to send email if that mail host support "reply" function, which means any anonymous email sent to the host will be resent to proper destination or as junk mail. Since Internet security is becoming more and more stringent, less and less mail hosts use "reply" function. In order to guarantee send mail function, we need at least one

SMTP(Simple Mail Transfer Protocol) running on the web server. In the system, we only have sending mail functions. Thus the POP3 (Post Office Protocol) that deals with receiving mail is optional.

3 Implementation

3.1 Key algorithms

3.1.1 Paper Uploading

To upload a file from client to server via Internet, we must rely on Hyper Text Transfer Protocol (HTTP 1.0, RFC1945 [14]). We know HTTP is designed for handling text transmission. But uploaded files can be any format, example pdf, doc, etc. Since HTTP 1.0, it includes the Multipurpose Internet Mail Extensions (MIME, RFC1952 [15]). With MIME protocol, a client can send multipart message in one request. Basically, the thing being sent and received between a web client and a web server via Internet is text. There are certain keywords, such as Content-Type and Multipart which may be included. If there are more than one data types in one HTTP stream, the keyword Multipart will appear and another keyword, Boundary, will appear too. “Multipart” indicates the information for the embedded part and “Boundary” marks the beginning and the end of the embedded part. Correspondingly, in the server side there must be a way to receive the embedded part in one HTTP request. In the project, we adopt a Java class called MultipartRequest, a freeware, under term of Free Software Foundation, which was developed by Jason Pell. By using this class, we can easily extract the embedded part in one HTTP request on the server side.

The embedded part in a multipart request can only be from a web client to a web server. There is no way for a web server to send back the embedded part to a web browser. This causes a problem of carrying the uploaded file name from page to page. To solve this problem, we use a child browser in uploading page instead of using upload button directly. In the child browser, we upload the file to the web server, store the file in a temporary folder and write the file into the main browser. File name can be carried from

page to page and get to be verified. If the client makes the final confirmation to upload the file and to submit related information, then the uploaded file will be moved from a temporary folder to a regular folder. If the client decides to delete the uploaded file, the file in the temporary folder will be deleted. If the client decides to abort, the uploaded file will remain in the temporary folder. The next time, a file with the same name will overwrite this temporary 'garbage' file. Otherwise, it will be there until the super user cleans the temporary folder. Leaving a garbage file in a web server will not block this application, but just waste certain a mount of resource.

File upload function is a multi-user function. There is a possibility that two users use the same file name. To avoid the file being overwritten, the later user will be told to change the duplicate file name by the system. A more complex case is two users to upload files with the same name and at the same time. To avoid the file being overwritten, we use a synchronized mechanism to generate a number for each file. The different files even with the same name are given different numbers. In the temporary folder, the file name is the generated number. Since the number and the file name are sent back to the client's confirmation page, the relation is clear. After the client click confirm button, the server save the file into regular folder with the original name. If there is a name conflict at this level, one user will be asked to change the name of the file. This mechanism resolves the problem of two files with the same names at the same time by two clients. It can also resolve the problem of sending the same file repeatedly from the same client because we assign the original name for the file in the destination folder. If a client deletes or aborts the procedure, the file will not be in destination fold and the file in the temporary folder will have a numerical name. This will not block the client to resend a file with the same name as a file name in temporary fold is a number and always incremented. Thus, if a

client aborts a procedure and resends a file with the same name, the numerical file name in the temporary folder will be different.

3.1.2 DBLP system and its conversion to MySQL table

DBLP, Digital Bibliography & Library Project, is a Compute Science library project, which contains information of conference papers, journals papers, books and so. As we mentioned in the introduction, DBLP is maintained by professor Michael Ley of University Trier in Germany. Each DBLP file is an XML file, which represents a conference paper, a journal paper or a book. Currently there are about 250,000,000 XML files in DBLP (Dec 2001). These files are in tree structure directories. Since we need to avoid conflict of interest in the review of submitted papers, we use DBLP to help us. Is it possible to use Java XML package on XML files directly? Theoretically, yes. But the reality is that it takes too much time to search all tree structure directories to verify each pair of PC member and author. Finally, using Java XML package on XML files directly is not feasible. One quick solution is to convert all XML files into a database table once and use this table to verify the relationship for each pair of PC member and paper author. Since we only need to verify the relationship, we create a table with three columns. One is key, which is an incremental number. Another is the path of the file, in case we need to verify the correctness of converting. The last is a string, which contains all the authors of each document in DBLP. If the PC's name and paper author's name appear in the same row, it means they were co-authors and this PC member is not to be allocated the paper under consideration. This solution also has very good performance since querying a table in MySQL database is much faster than searching XML files in tree structure directories. In order to convert XML files into a DB table, we need to know about the structure of

DBLP files. Each DBLP file is an XML file, which contains author name. In an XML file from DBLP, there are tags called author tags. So between the <author> and </author> tags is the author name. It is possible that there are more than one pair of tags, <author> and </author>, in one XML file. So no matter how, we just collect all the strings between <author> and </author> for all pairs of such tags. Then we get all the authors' names for each document.

The next step is to find an algorithm to search all the files in one directory, including files in the subdirectories. Obviously, it is a recursive process. We can use both Java and C to do this. Again, there is a performance problem. It takes 8 hours by using Java to do the conversion but takes only 10 minutes by using C to do the same thing. Using C to connect MySQL database, a mysqlclient library must be included. Indeed, the C program connects MySQL database through a MySQL socket. So, the socket file must be indicated correctly in the program.

3.1.3 Paper allocation algorithm

In the system, a paper can be allocated to a PC member either by an automatic allocation function or manual allocation by the GC. In the case the GC uses the automatic allocation function, several constraints will be applied to the function. There is no restriction for manual allocation. GC may allocate any paper to any PC member. So, the paper allocation algorithm is mainly used for the automatic allocation function.

Following are the constrains for the automatic allocation

- i. A paper should not be allocated to the same PC member more than once.

- ii. A paper can be allocated to a PC member if neither the number of papers allocated to the PC member has been exceeded nor the number of reviewers for the paper is over its limit (typically each paper is to reviewed by three to four PC members).
- iii. A paper can be allocated to a PC member if the paper's topics and PC's interests fields are overlapping.
- iv. A paper can't be allocated to a PC member if the paper author and the PC member are in the same organization.
- v. A paper can't be allocated to a PC member if one of the paper's authors and the PC member were co-authors.

For the constraint i to iv, we can use the information from the system to make a decision.

For the constraint v, we use the information from DBLP system to make a decision.

For the constraint i, we can check the allocate table in the database. If the pair of PC id and paper id exists, allocation function will skip this pair of PC id and paper id. For the constraint ii, there are two parameters, which can be set by GC. One indicates that a paper can be reviewed by maximum number of PC members; another states that the maximum number of papers a PC member can review. By using these two parameters, automatic allocation process will keep the constrain ii. For the constraint iii, we can check a paper's topics, a required field when an author submits a paper, with the PC's areas of interest. If there is an overlap, then the paper can be allocated to the PC. For the constraint iv, we can use the information provided by paper author and PC members. The organization is required field for author to submit a paper and it also a required field for a PC member's registration. Hence the system has information about the organization for both the author and PC member. The problem is the same organization can be presented in different ways. In our system, if we find the organization has more than one word, we

pick up the first letters of each word as abbreviation. If the abbreviation of author's organization is a sub string of the abbreviation of PC member's organization or vice versa, we assume the two organizations are same. Example, University Quebec à Montreal can be University of Quebec at Montreal, UQAM in abbreviation, or even Univ. de Quebec à Montreal. Besides this, if every word in one organization appears in another organization, we assume these two organizations are same. Example, AT&T Research Lab and AT&T are assumed to be the same; University of McGill and McGill University are assumed to be the same. For the constraint v, the DBLP table in MySQL database is the solution. If we can find the author's name and the PC's name in the same row we assume they were the co-authors. As for the constraint iv, names could be abbreviated and in different order. For example, Hon Fen Li. can appear as H. F. Li. We can use the "like" operation in SQL language to resolve such a problem.

3.1.4 Database connection strategy

To connect to the MySQL database using Java, we must use a JDBC driver for MySQL. Since there is a lot of common code for initiating a JDBC connection, the better way to implement the DB connection is to use a centralized Java class to do all database operations. Thus setting database connection in each Java class where we need to communicate with database is avoided. The common Java class is called ConferenceSql. If we need a database connection in a class we need to first instance the ConferenceSql class then simply use its functions. This not only centralizes the connection functions in one class but also makes the functions of ConferenceSql reusable. One requirement is that all ConferenceSql's functions that are to be accessed from other classes must be made "static". This is because the JDBC driver is "static" class and we can't wrap a static class with a non-static class in Java.

3.2 Major functions of Confsys

3.2.1 Topic control

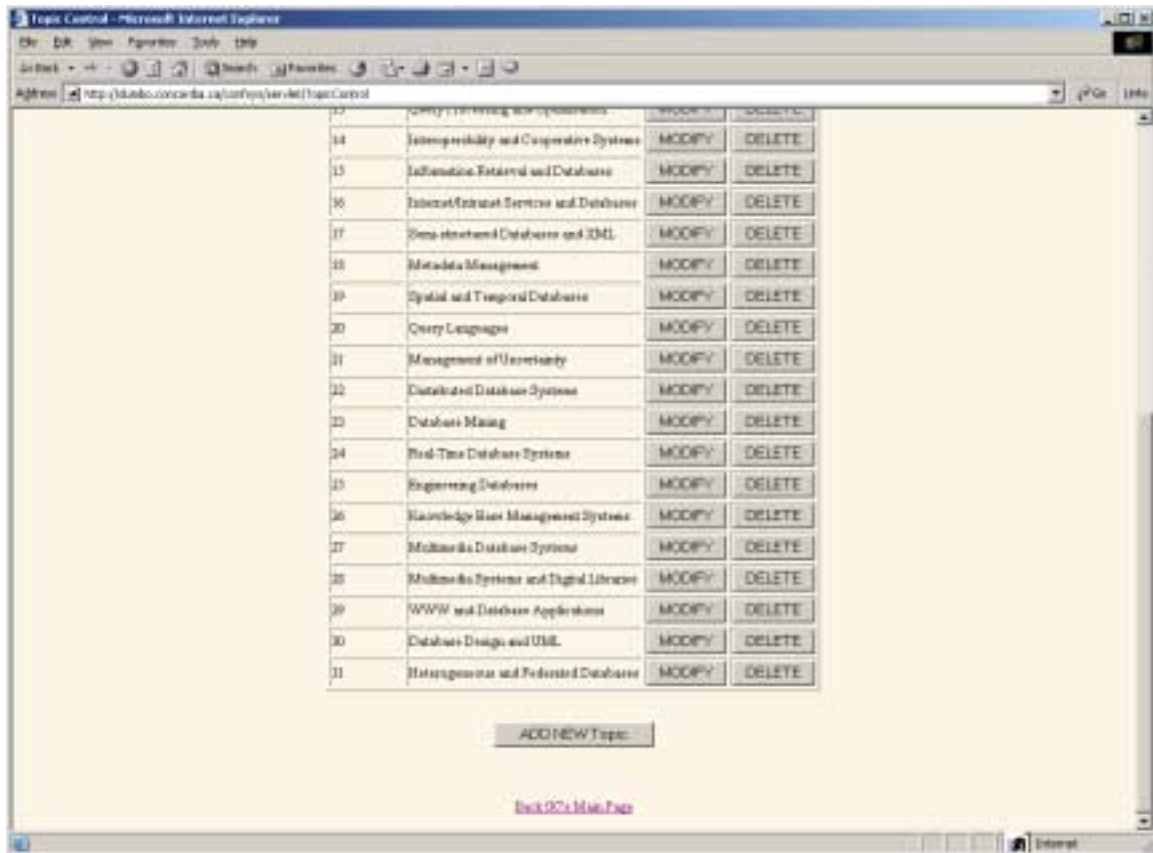


Figure 2

Since every conference has its own topics, the GC must have a way to insert the topics, the PC members and other parameters for the event. These functions give the GC a way to add, delete and modify the topics. The interface is given in Figure2. Using this window, when the GC click “ADD NEW Topic” a new sub window will popup and the GC can add a new topic there. There is validation for the “add” function. If the topic already exists a message will show up to indicate the new added topic is duplicate with an existing topic. Same validation rule is applied to “modify” function if the GC member tries to modify topic A into topic B and topic B already exists.

3.2.2 PC member control



Figure 3

The Program Committee members are entered in the Confsys as shown in Figure 3.

Through this interface, the GC can add a new PC member, delete an old PC member and modify an existing PC member.

3.2.4 PC member registration

PC Registration OR Modification - Microsoft Internet Explorer

Address: http://banks.ccsd.edu/~ccs/education/PCRegistration.htm

PC Registration OR Modification

Please fill in the following PC's information (all fields are required):

Last Name:

First Name:

ID#:

Address:

Email Address:

Telephone:

Please check the following subjects you are interested in. Press 'Ctrl' key when you select multiple items (optional):

- Energy Efficiency and Green Buildings
- Active Learning
- Introduction and Fundamentals
- Construction Control and Recovery
- Construction and Post-Occupancy and
- Green Architecture
- LEED Rating and Transition to Leadership

Figure 5

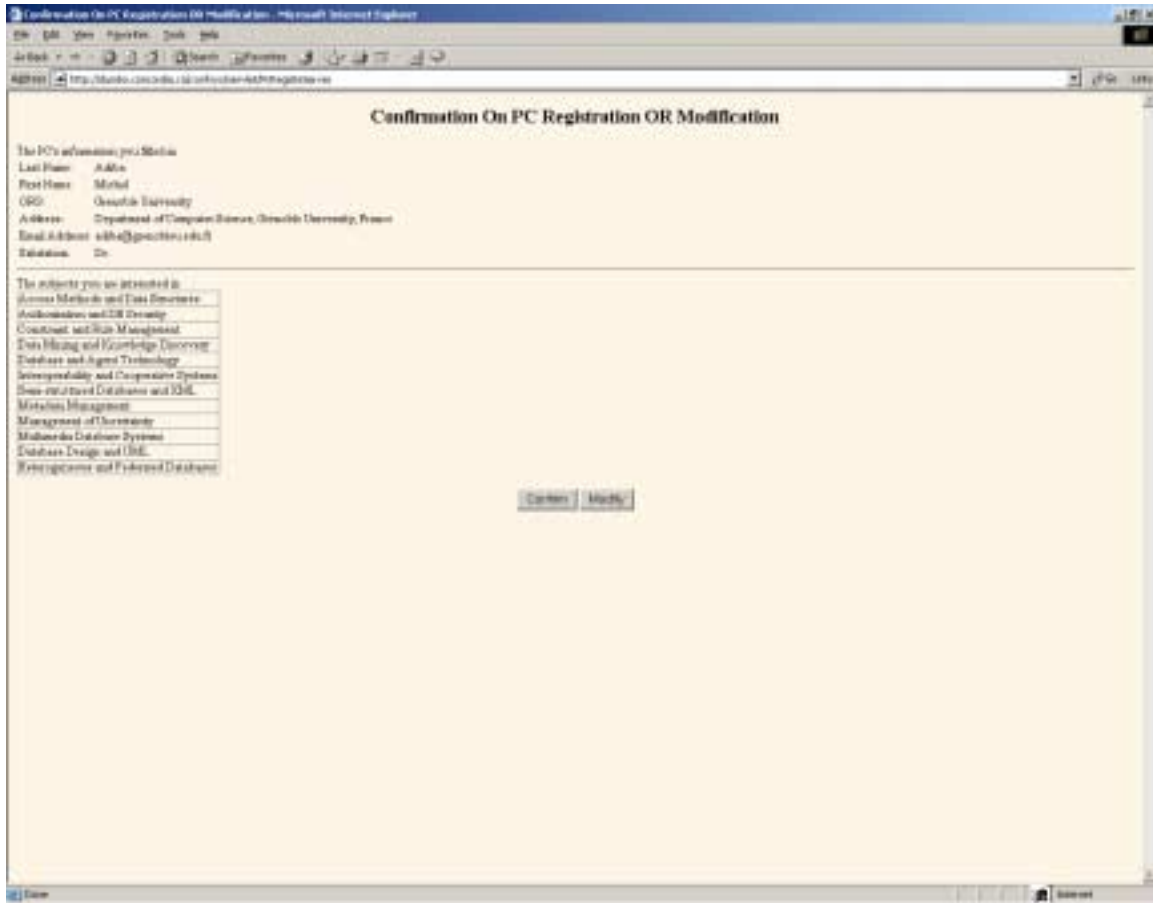


Figure 6

After a PC member received the user id and password from the GC, he can use the user id and password for accessing the PC's functions. The GC member will only give a PC member the user id and password and basic information. For the other information, PC member will use the registration function to fill them in. The web page in Figure 5 lets the PC members to fill their information and the next page in Figure 6 lets them to confirm the input or correct any errors. Once they confirm, the information will be saved in the database.

3.2.5 Paper uploading function

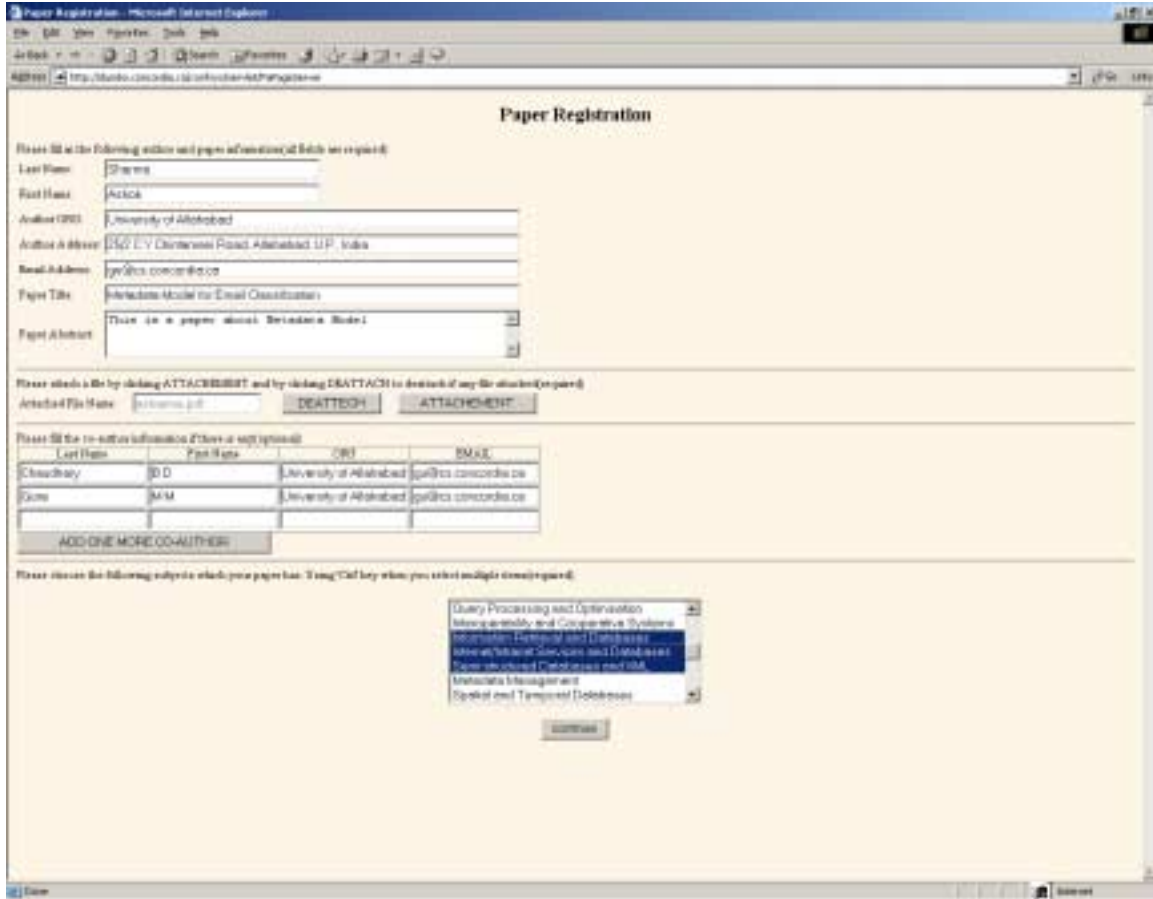


Figure 7

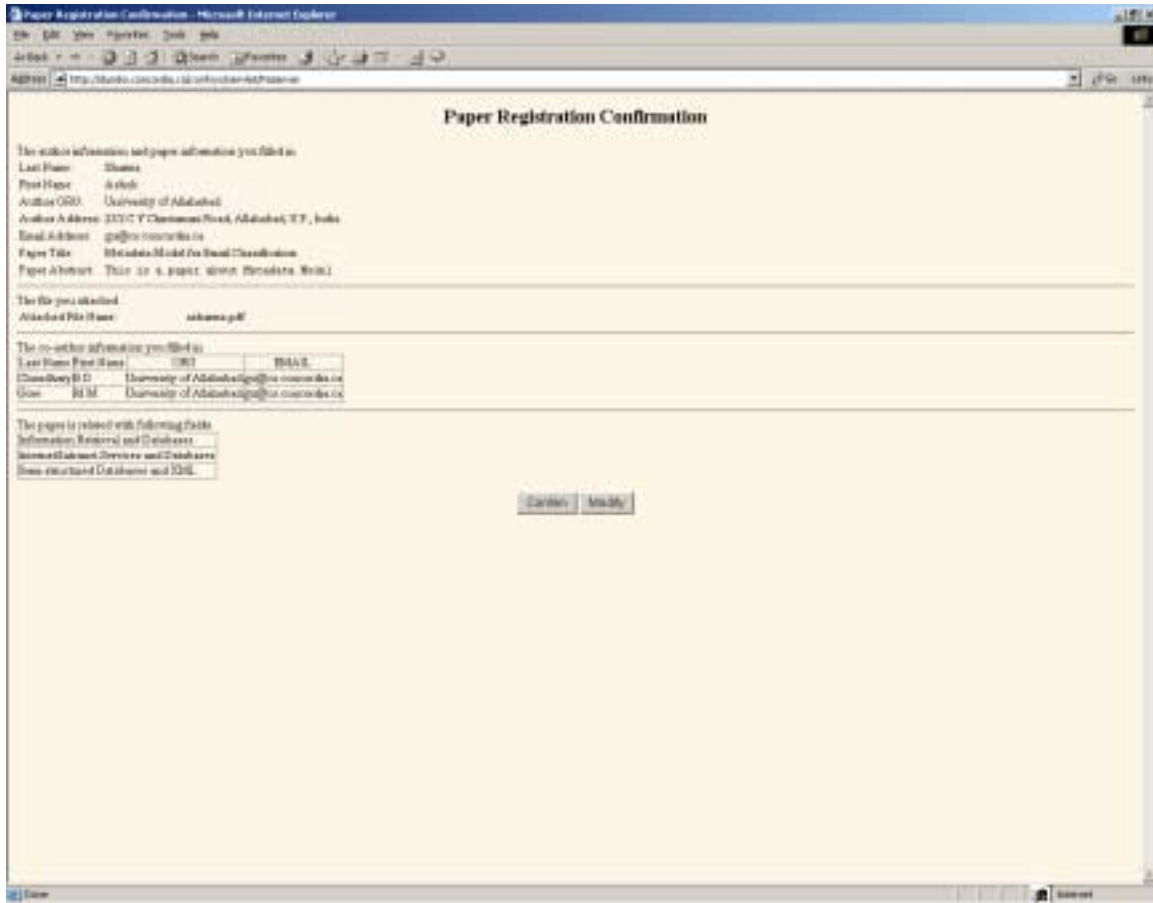


Figure 8

The page, as shown in Figure 7, is used by an author to fill in her information and those of co-author(s) and to upload the file for the paper. Once the author clicks on the “continue” button, she is presented with a confirmation page shown in Figure 8 if every required field in this page has been filled correctly or with a page similar to Figure 7 with a warning message if a required field is missing. The author can click on “confirm” button if everything is in order. If the author needs to modify something, she can click “modify” button to return to the page in Figure 7. All the information the author filled will be re-displayed. One problem can occur if we use “file” input box directly; the file name can’t be restored. The reason for this is that the value of a “file” input box couldn’t be filled in by a server side program. It can be filled in only by a dialog window in client side. Hence, instead of using “file” input box directly, we use a normal “text” input box.

The value in a “text” input box can be sent to the server and can be restored. The reason we need the file name to be restored is because we need to keep consistency of the restored page. If and only if when a user clicks the “confirm” button in the page of Figure 8, the file and all the other information can be saved to the server.

3.2.6 Paper allocation

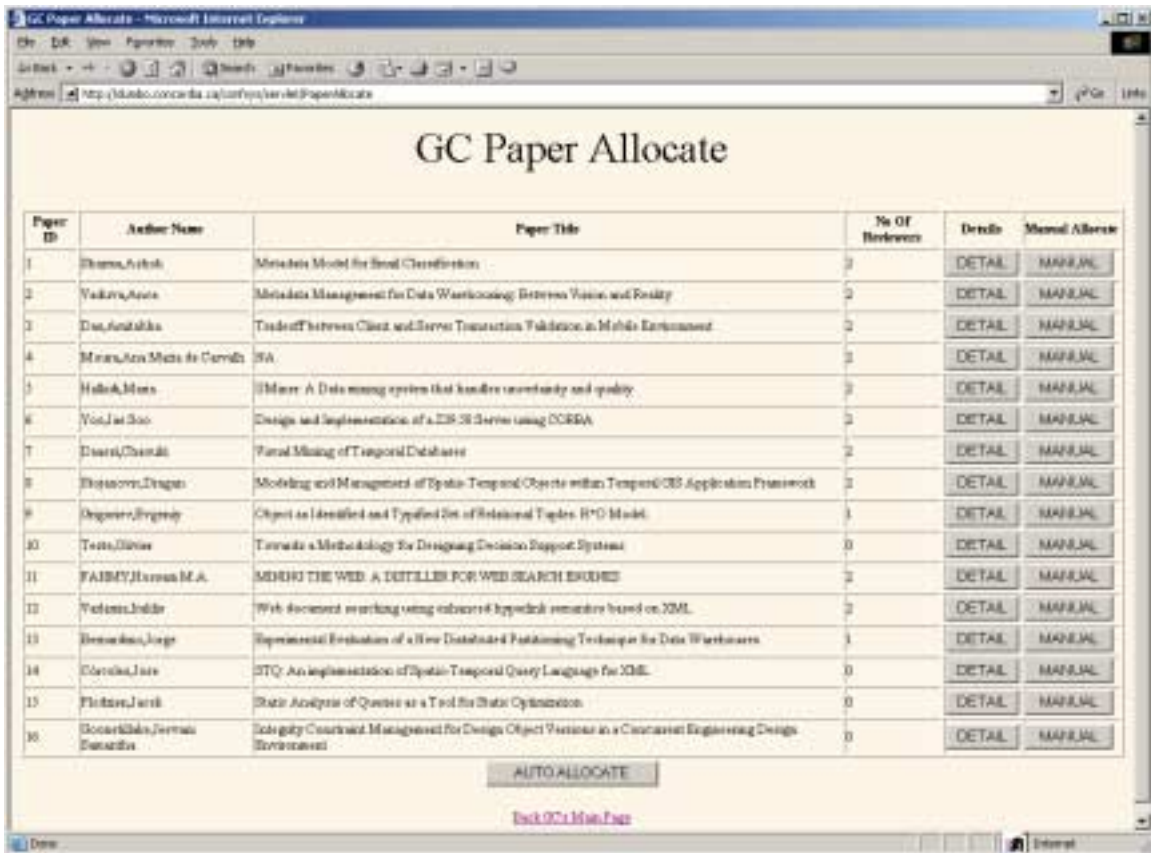


Figure 9

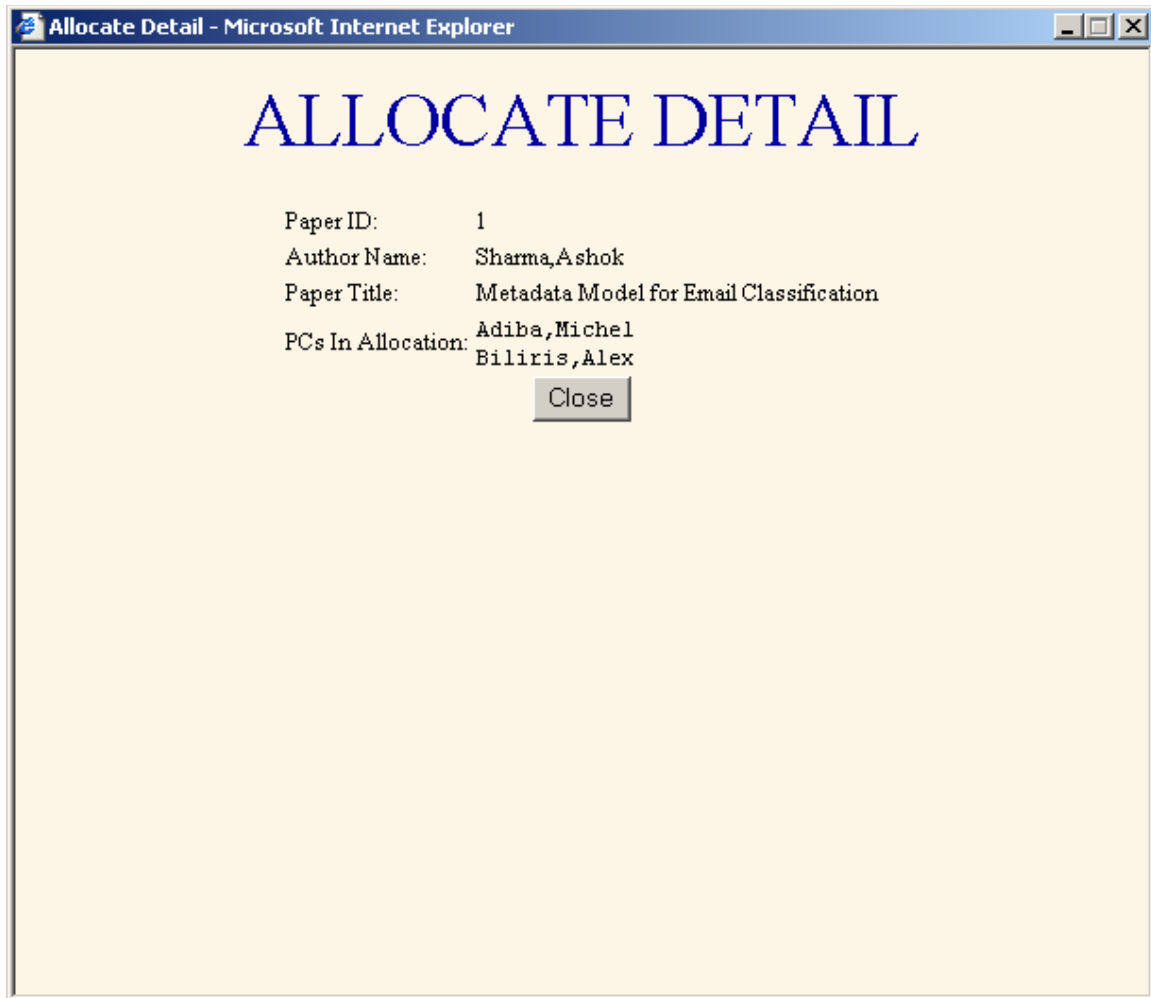


Figure 10

Once the paper submission is over, the GC can use auto allocate function, which we discussed in 3.1.3. At the start of the process, the number of reviewers for each paper is 0. The auto-allocation function would attempt to assign reviewers for the most of the papers and the most of reviewers would be assigned some of the papers. If the GC clicks the detail button for each paper she will see the names of reviewers for the paper as shown in Figure 10. Notice that there are a few papers which have no reviewers shown in Figure 9. This is because these papers do not match the interests of PC members or those who can review these papers have already been allocated other papers or they may have conflicts. In this case, the GC must use the manual allocate function to assign these papers to PC

members regardless of the allocation rules used for automatic allocation. Also the GC may not be satisfied with the result of automatic allocation. So the GC can use the manual allocation function to do some adjustment. Once the GC member feels comfortable with the allocate result, she can set the start review date to let PC members review the papers.

3.2.7 Setting up the Start Time for PC review

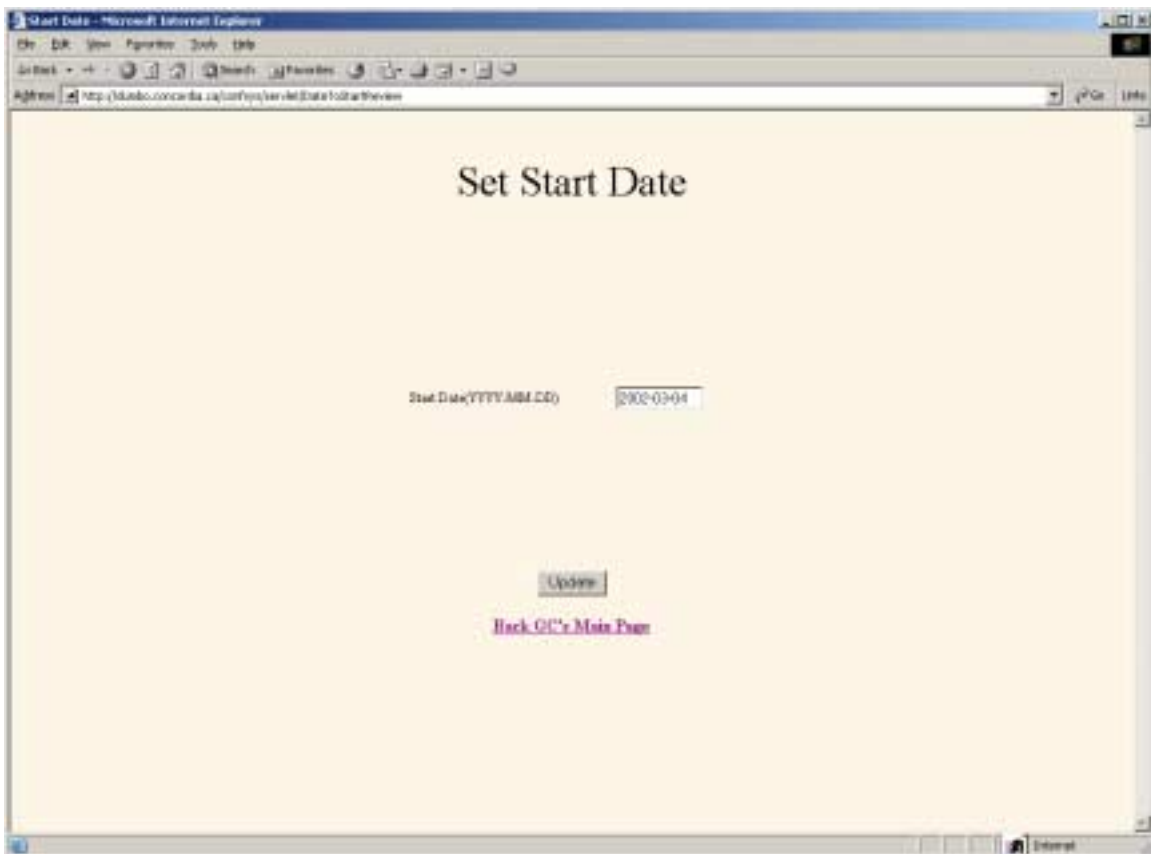


Figure 11

Once GC allocates papers to the PC members by using automatic allocation function in the system, she needs certain time to do the adjustment manually with the “manual allocate” function in the Confsys. If the PC members are allowed to review the papers right after automatic allocation there will be an inconsistency if there is a manual modification. It also makes PC members unhappy if they make some comments on some

to download the paper(Figure 12). Depending on the file format, the browser may either display the paper or ask permission to save the file. The PC member also can right click the file to just save the file right away. There is a security concern for downloading the files for papers. If the files are put in a folder which is accessible by the web server it is possible for any one, who knows the name of the directory and file name, to download it. To avoid this drawback, we don't put the files in a web server accessible folder but let the servlet read the files from a non-web directory and return the stream for downloading by the client. Since the download servlet can only be accessed by PC member, only they can download the assigned files. This is an application level security.

3.2.9 PC paper review

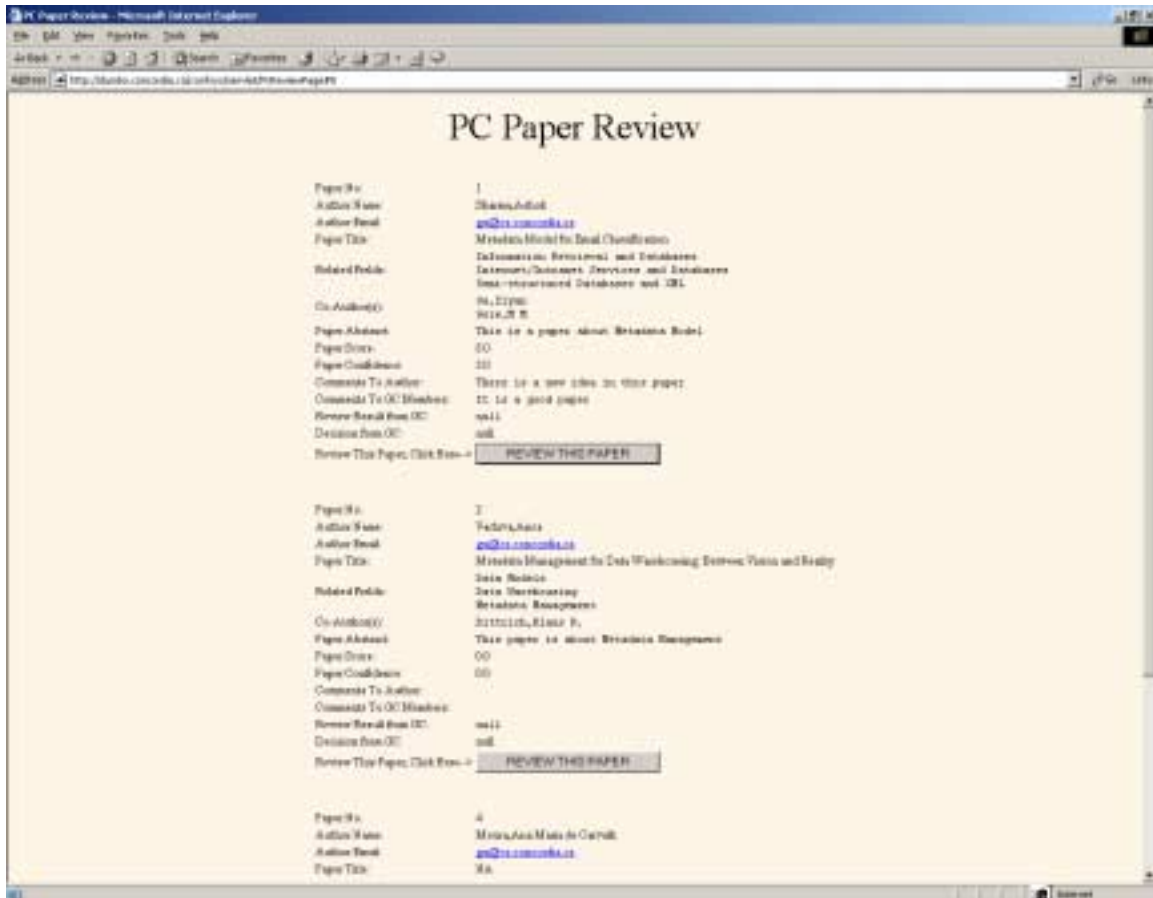


Figure 13

PAPER REVIEW - Microsoft Internet Explorer

PAPER REVIEW

Paper No: 1
Author Name: Sharma, Ashok
Paper Title: Metadata Model for Email Classification
Paper Score: 8.0
Paper Confidence: 3.0

Comments To Author: There is a new idea in this paper

Comments To GC Members: It is a good paper

Submit Reset

Figure 14

Once the PC member downloaded the papers allocated to him and reviewed them, he can use the evaluation function in Confsys shown in Figure 13. Once the reviewer clicks the REVIEW THIS PAPER button, the sub window shown in Figure 14 will popup to let him fill in the evaluation information including score, comment, etc. If the reviewer makes a confirmation in the sub window by clicking the “submit” button, the result will be sent to the server and the values will be refreshed in the main evaluation window (Figure 13).

These values are modifiable by the PC member until a final decision on the paper is made by the GC. Once the GC makes a final decision on a paper that paper could not be

evaluated by any other PC members or the review modified. A message will be displayed to indicate that a final decision on this paper has been made.

3.2.10 Paper decision

PAPER ID	TITLE	DOWNLOAD	AUTHOR NAME	AUTHOR ID	DETAIL	API_SCORE	API_CONFIDENCE	WEIGHT	DECISION	MAIL COUNT	EMAIL
1	Miner's Model for Social Classification	download.pdf	Shamsulhadi	University of Alabaha	DETAIL	0.1	0.3	0.4		00	MAIL
2	Miner's Management for Data Warehousing Between Space and Time	download.pdf	Fahma, Agus	University of Jambi	DETAIL	0.1	0.0	0.0		00	MAIL
3	Tradeoff between Client and Server Transactions Volume in Mobile Environment	download.pdf	Das, Anandika	Heppang Technological University	DETAIL	0.1	0.0	0.0		00	MAIL
4	MA	download.pdf	Motera, Ana Maria de Castro	Ministry Institute of Engineering - IMEN	DETAIL	0.1	0.0	0.0		00	MAIL
5	Online A Data Mining System that Handles Incompleteness and Quality	download.pdf	Wahid, Mawati	A Bera University of Economics and Business	DETAIL	0.1	0.0	0.0		00	MAIL
6	Design and Implementation of a ERP II Server using COBRA	download.pdf	Yon, Ju-Boo	Chungbuk National University	DETAIL	0.1	0.0	0.0		00	MAIL
7	Virtual Mining of Temporal Database	download.pdf	Tanaka, Naomichi	Laboratoire CLIRIS/IRIT University	DETAIL	0.1	0.0	0.0		00	MAIL
8	Modeling and Management of Spatio-Temporal Objects within Temporal DB application Framework	download.pdf	Dejagovic, Dejan	University of the Republic	DETAIL	0.1	0.0	0.0		00	MAIL
9	Object as Identified and Typification of Relational Tables: FPO Model	download.pdf	Nguyen, Nguyen	Independent author	DETAIL	0.1	0.0	0.0		00	MAIL
10	Towards a Methodology for Designing Decision Support Systems	download.pdf	Tanaka, Naomichi	University of the Republic	DETAIL	0.1	0.0	0.0		00	MAIL
11	SEARCHING THE WEB: A DISTILLER FOR WEB SEARCH RESULTS	download.pdf	FANMY, Huyen M. A.	KHONAM UNIVERSITY	DETAIL	0.1	0.0	0.0		00	MAIL
12	Web document searching using relevance algorithms: a case study on XML	download.pdf	Fahma, Agus	A Bera University of Economics and Business	DETAIL	0.1	0.0	0.0		00	MAIL
13	Relevance Evaluation of a Data Warehouse Partitioning Technique for Data Warehouses	download.pdf	Benabib, Joao	Faculdade Superior de Engenharia de Coimbra	DETAIL	0.1	0.0	0.0		00	MAIL
14	SQL: An Implementation of Open Temporal Query Language for XML	download.pdf	Giannakopoulos, Christos	University of Central Florida	DETAIL	0.1	0.0	0.0		00	MAIL
15	Static Analysis of Context as a Tool for Static Optimizations	download.pdf	Fahma, Agus	University of Economics and Business	DETAIL	0.1	0.0	0.0		00	MAIL
16	Integrating Contract Management for Design Object Version as a Document Engineering Design Representation	download.pdf	Benabib, Joao	University of Coimbra	DETAIL	0.1	0.0	0.0		00	MAIL

Figure 15

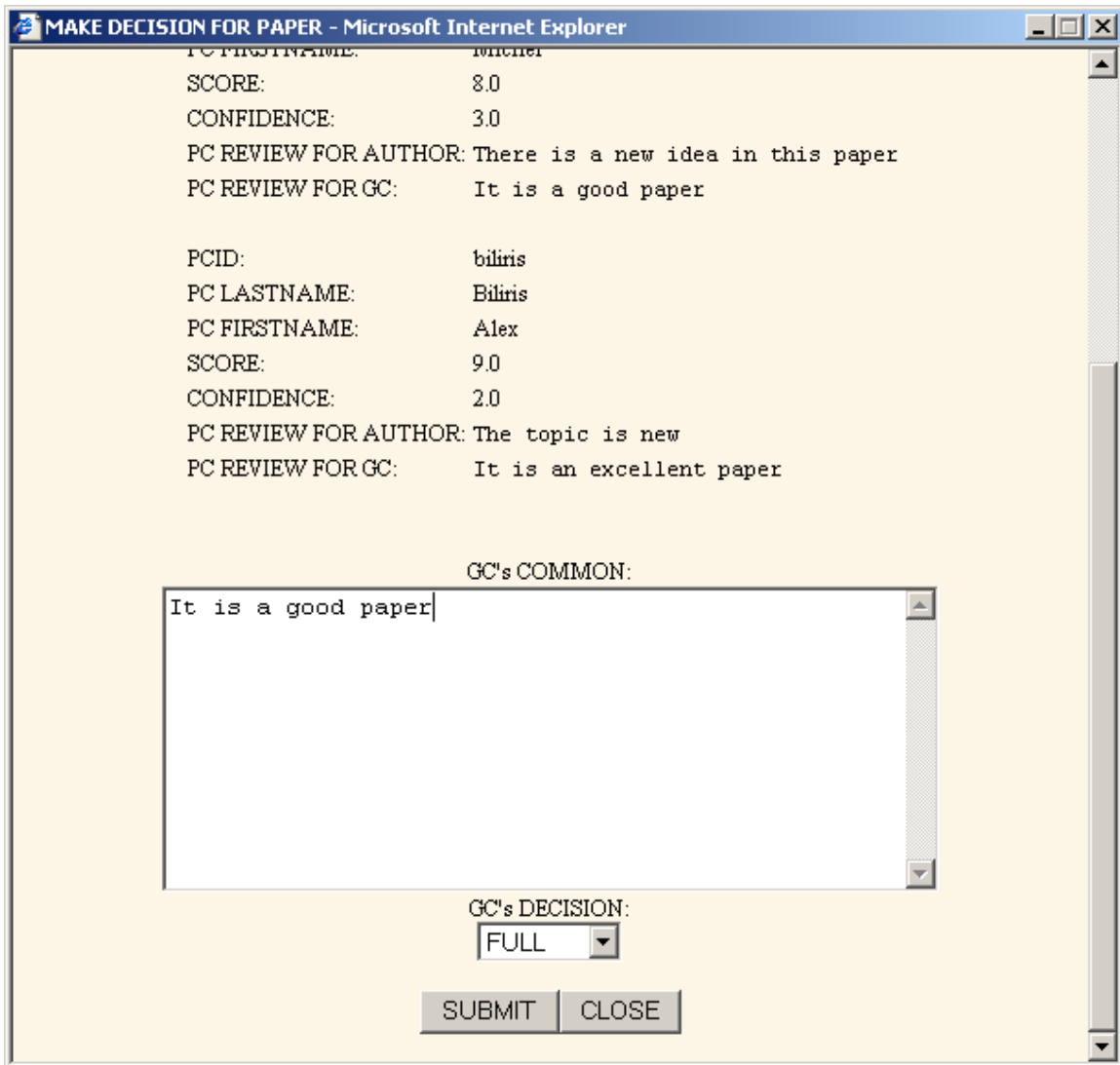


Figure 16

Once the majority of the reviews have been made, the GC needs to make a decision on each paper. Based on the Program Committee's review including score, weight and other information the GC member can either accept a paper as full paper or short paper or simply reject the paper. In the web page given in Figure 15, the GC member can click detail button to see the detail review from each PC member on each paper in a pop-up window shown in Figure 16 and can make final decision for the paper in that window. Once a decision has been made, the result will appear in the decision column. For the paper for which a decision is made, the GC can use an automatic function to send the

emails to both paper's author and the PC members who reviewed the paper to notify them of the final decision.

3.3 Application Security

Internet security is becoming more and more important. The main reason for the need of security is due to the number of hackers and abnormal users of the internet. Since the internet is an open system, anyone has the potential to attempt to access a system even through it has security check, such as user name and password. A hacker may attempt to break in by trying many user names and passwords. There is no one hundred percent secure system. The more secure a system, the less likelihood an unexpected user to break into it.

For a web application, there are two kinds of securities. One is system level security and another is application level security. The system level security is concerned with securing server resource and only allowing permitted users to access. For example, we can set the Apache configure file to only allow permitted users access application site. In this solution, an application administrator must ask the web server administrator to make appropriate settings. This will weaken the application administrator's control power on the application and cause the migration problem when an application is transferred from one server to another server. Alternative solution is application level security, in which the application program controls the security. In this application, we adopt the application level security. The system uses user name and password for PC members and GC. The user name and password will be saved in a session object to be carried to all the following pages. In all the functions for PC members or GC, the user name and password

checking will be applied internally. So, any user who tries to access a page which is not supposed to be accessed by that user will be denied by the system. This is the key security concept in this system.

4. Database Schema and Table Description

The database in this application is called confsys. There are total of 12 tables.

Followings are these tables and their schemas.

- allocate

Column Name	Description
paid	Foreign key for paper table
pcid	Foreign key for pc table
score	the score of paper
confidence	the confidence of one paper
revi_to_au	PC's comment to paper author
revi_to_gc	PC's comment to GC member

The “allocate” table keeps the information for allocated papers. The evaluation results are also saved in this table. Before the allocation, either automatic or manual, this table should be empty. After the allocation, the relation of a paper and a PC member is stored in this table. Once a PC member gives his score and comments on a paper, this score and/ comments will be stored in this table as well.

- allocate_limit

Column Name	Description
papers_pc	maximum number of papers for each PC
pcs_paper	maximum number of reviewers for each paper

The “allocate_limit” table just has one row, which contains the maximum number of papers for each PC member and the maximum number of reviewers for each paper. This information is used for auto allocation function.

- author

Column Name	Description
paid	Foreign key for paper table
lastname	paper author's last name
firstname	paper author's first name
org	paper author's organization
address	paper author's address
email	paper author's email address

The “author” table stores paper authors’ information, such as name, organization, etc.

When a paper author uploads a paper, her information is stored in this table.

•coauthor

Column Name	Description
paid	Foreign key for paper table
coid	co-author id number
lastname	co-author's last name
firstname	co-author's first name
org	co-author's organization
email	co-author's email address

The “coauthor” table stores the coauthor information for each papers. It has no relation to co-author in DBLP system.

•dblp

Column Name	Description
rowno	the Key of the dblp table

path	path and file name for corresponding XML file
author	all authors' name for corresponding XML file

The “dblp” table is a derived table from the conversion of DBLP database. There are three columns in this table. One is the key, an auto-incremental integer. One is path, which stores the file name with the path. With it, we can look for more information in the original XML file. The last is names of authors of the document from the XML file.

Obviously, these authors are co-authors. The author information is the major component for the automatic allocation function.

- gc

Column Name	Description
gcid	GC member's logon name
password	GC member's password
lastname	GC member's last name
firstname	GC member's first name
org	GC member's organization
address	GC member's address
email	GC member's email address
mailhost	GC member's mail host with reply feature

The “gc” table keeps the GC's information. There are only a limited number of rows in this table because there are only a limited number of GC members. There is no interface to initiate this table and needs a MySQL DBA to set up this data because the GC can't

input her own security and other information by herself. Once this information is inserted into this table, it should not be changed by any user except the MySQL DBA.

•pa_subject

Column Name	Description
paid	Foreign key for paper table
subjid	Foreign key for subject table

The “pa_subject” table is a relation table between “paper” table and “subject” table. Each paper can cover multiple areas, which are the paper subjects. This table just indicates a one to many relation between “paper” table and “subject” table.

•paper

Column Name	Description
paid	Key of the paper table
total_pc	total PC number a paper is assigned to
paperTitle	paper title
paperfilename	paper file name
paperabstract	paper abstract
review_result	GC’s review result
decision	GC’s final decision
weighted_score	N/A not used in this application
mailed	a flag sent an email or not for certain paper

The “paper” table stores the information about papers, such as titles and file names etc.

After an author uploads a paper to the system, the paper file would be stored in certain directory and the information related to paper would be stored in this table.

•pc

Column Name	Description
pcid	PC’s login name
password	PC’s password
total_pa	the number of papers a PC has
lastname	PC member’s last name
firstname	PC member’s first name
org	PC member’s organization
address	PC member’s address
email	PC member’s email address
salutation	PC member’s salutation, like Mr., Dr.,

The “pc” table stores the information for PC members, such as name, organization, email and interested topics, etc. The values inside this table can be manipulated by the GC and by PC members themselves. Once a PC member logs on the system, he can use the registration function in the system to modify the information for himself. The GC can also initiate and modify the PC’s information.

•pc_subject

Column Name	Description
pcid	Foreign key for pc table

subjid	Foreign key for subject table
--------	-------------------------------

The “pc_subject” table is a relation table, like “pa_subject” table. Once a PC member selects topics of interest, rows will be inserted into this table.

•start_date

Column Name	Description
start_date	the start date for PC members to reveiw

The “start_date” table only has one row, which stores a date. This date will determine if a PC member can download and review papers or not. The GC member can modify this date and thus control the start of using the system by PC members.

•subject

Column Name	Description
subjid	Key of subject table
subjname	subject description

The “subject” table stores the subjects for each conference. The subjects can be added, deleted and modified by the GC. Once the subjects or topics are decided by the GC, the authors can select one or more topics to be associated with their papers during the submission procedure. The PC member can likewise indicate the area of their interests using these topics.

5 Testing Confsys

5.1 Test from user point of view

Any application will be finally used by users. The user can verify the business logical to see if the application meets the user requirement. In Confsys, there are three kinds of user groups. They are authors, PC members and GC.

For the authors, we tested the paper uploading and information submit function and verified that the uploaded files are exact replicate of the original files. We also ensured that the information about the papers is correctly associated with the uploaded files. We also tested some exceptional cases. For example, we need to test how the system behaves if an author interrupts her submitting operation before it is completed. If the author submits the same file next time, the system allows her to do so.

For the PC members, we verified that functions work properly, like paper review, paper download and PC member registration.

For the GC, we performed test of all GC's functions to ensure that they work correctly.

Finally, we tested the interaction among authors, PC members and the GC. For example, once an author submits a paper, we got the result that the system receives it reliably.

Once the GC assigned a paper to a PC member, the PC member was able to access it.

Again, once a PC member gave his reviews on a paper, the information was correctly stored in the system. Finally, when the GC made a final decision on a paper, both the author and the relevant PC members received the notification. Besides this, we also made a check on the automatic allocation algorithm; the result met our original purpose. In order to test this, we need to run the automatic allocation first and test the result to see if there are any conflicts.

5.2 Tests from technical point of view

Confsys is a multi-user system. Concurrency is an issue to be tested. We tested different authors sending their papers at the same time and found that the system can handle them concurrently. One special case is different authors sending files with same name at the same time. In this case, the system rejects one of them.

Format of the uploaded files can vary. We made test to ensure that the system can handle all file formats.

For the DBLP converting, we made spot checks to verify that the information in the MySQL table is consistent with the information in the XML files.

6 Conclusion

6.1 Confsys, a useful system

Confsys can coordinate the authors, PC members and GC to process paper via internet. It also helps GC to reduce the paper allocation time. It is a secure system.

6.2 Reliable Open Source software

All the software packages used in this application are Open Source. There is no licence problem. These Open Source software are very stable and for a middle size application these software are reliable.

6.3 Java, a good programming language

Java can be used in both front end and back end. In this application, we use Java as back end tool, Java Servlet. Like Perl CGI (Common Gateway Interface) and ASP (Active Server Page), Java Servlet is a server side program. Since Java is a full programming language, it has more features than a script language.

6.4 MySQL, a good database server

MySQL is a good database server for middle or small size application. It is straight forward and efficient. It has an easy way to connect to application programs by using C/C++ and Java languages.

6.5 Tomcat, a good Servlet Engine

Tomcat supports the Java Servlet well. It also cooperates with Apache web server very well. Tomcat also supports the updated Java package, like JDK 1.3.

7. Future Improvement

There are three major improvements needed for future work.

First, in the current system, authors don't need to register. They can submit their papers without any check. Thus, it raises a potential risk that anyone can submit any file to the system. This is not our original design purpose. To overcome this problem, we must have function to allow potential authors register and be assigned a user name and password. Authors could use their user name and password to access the conference system. In order to do this, the GC must have an author control function, which is similar with PC member control function. An auto email sub function for sending confirmation to authors also needs to be added for this enhancement.

Second, in the system, the GC's final decision is sent to authors and PC members via email by the automatic email function. But this email function depends a lot on mail host. It may not function properly if SMTP (Simple Mail Transfer Protocol) server is set up with a lot of constrains. It is better to add an option where authors and PC members can view the final decision on the system.

Third, we need to add function to allow the PC members to select a number of papers from submitted pool. If there is no conflict, the requested papers will be assigned the PC member. We also need to add function to allow each PC to decide his own maximum number of papers for reviewing. This is more reasonable in real world because some PC member prefers more papers to review while others prefer less.

Besides these, we need to add an email function to let GC make announcement to all PC members that they can start to download and review papers. We may separate the hosts for database server and web server. Now we only use one host to act as both web server

and DB server. It is always good to set a DB server standalone. It can reduce the risk of conflict among various servers.

Reference

- [1] Patrick Chan, *The Java Class Libraries, Second Edition, Volume 1*(ISBN 0-201-31002-3), Addison Wesley Longman Inc, 1998
- [2] Kurt Wall, *Linux Programming* (ISBN 0-7897-2215-1), QUE, 1999
- [3] Arman Danesh, *Teach Yourself JavaScript 1.1 in a Week*, SAM, 1996
- [4] Laura Lemay, *Teach Yourself Web Publishing with HTML 4* (ISBN 0-672-31725), SAMS, 1999
- [5] Y.Danie Liang, *Java Programming Second Edition*, QUE, E&T, 1998
- [6] Bill Ball, David Pitts, *Red Hat Linux 7* (ISBN 0-672-31985-3), SAMS, 2000
- [7] MySQL reference, <http://www.mysql.com/documentation/mysql>
- [8] Clayton Walnum, *C++ Master Reference* (ISBN 0-7645-7524-4), IDG, 1999
- [9] Tomcat Reference Manual, <http://jakarta.apache.org/tomcat/tomcat-3.3-doc/index.html>
- [10] DBLP Reference Site, <http://dblp.uni-trier.de/db/>
- [11] JDK 1.3 Reference Site, <http://java.sun.com/j2se/1.3/>
- [12] MultiRequest and MultiRequestServlet, <http://www.geocities.com/jasonpell>
- [13] Java Mail Site, <http://java.sun.com/products/javamail/>
- [14] RFC 1945, <http://www.w3.org/Protocols/rfc1945/rfc1945>
- [15] RFC 1952, <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1952.html>