CONFSYS: The Cindi Conference Support System

Xin Jin

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 2003

Abstract

CONFSYS: The CINDI Conference Support System

Xin Jin

This report describes a web-based conference management system (ConfSys) to be used by three groups of users: authors, reviewers and program chair(s). Authors can submit their papers on-line; reviewers (program committee members) can assign their preference for papers to review in an on-line auction and later download and review papers assigned to them on-line; finally the program chair(s) can use the system to automatically perform an initial assignment of papers. This assignment is based on the reviewers' preference from the auction process as well as certain guidelines used in avoiding conflicts of interests. The system also facilitates the program committee in debating the review and converging to a decision about the papers. The authors can directly get the results of the review and the blind comments from the reviewers, online.

This report presents the design and implementation of the ConfSys. It focuses on the important new features and automatic allocation algorithm based on the preference of reviewers. The system is implemented using Java Servlet communicating with a MySQL database server on Linux platform. The project uses an Apache web server integrated with the Tomcat servlet engine.

Acknowledgements

First, I would like to express my gratitude to my supervisor Dr. Bipin C. Desai. He gave me valuable suggestions and encouragement and the chance to put this software to real use in IDEAS'03. Thanks for his tireless efforts to this project, his system environment support and valuable suggestions.

I also want to give my thanks to Zhengwei Gu. His clear explanations and valuable discussions enabled me to understand the existing conference system and start my work in a short time.

Finally, my thanks go to my family, my parents and husband Xiaoyu Zhang, for their encouragement and support.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Survey on Conference Management Systems

In managing an academic conference, the program chair (PC) is required to deal with many administrative tasks such as: interaction with authors and program committee members (reviewers), paper collection, paper allocation, distributing the paper to the reviewers, collating, sorting and tabulating the reviews, orchestrating the debate on reviews of controversial papers, making the final tabulation and preparing the notification and comments to the reviewers and authors. Such work, while being trivial and repetitious, is time-consuming.

A number of Web-based systems for automating the above tasks have been reviewed in [1]. Here, we summarize the features of the following recent systems: CyberChair [2], START [3], and WitanWeb [4]. As in many of the systems reviewed in [1], they are implemented using CGI script languages: CyberChair uses Python, while START and WitanWeb use Perl. All three systems have the most of needed features for conferences such as electronic paper submission, assignment of papers to reviewers, submissions of reviews by the reviewers, mailings to reviewers and authors, etc.

CyberChair system implements the champion identification scheme presented in [5]. The system supports four groups of users: the maintainer, the authors, the PC and the reviewers. The following are its major features: author registration; upload of abstract, paper (either in PS or PDF format), and camera-ready version of paper via web, paper

allocation based on reviewers' preferences and expertise, conflict of reviews detection. The reviewers are asked to fill in the classification of papers on the review form. The classifications are as follows: A. Advocate/Accept; B. Accept, but could reject; C. Reject, but could accept; and D. Detractor. CyberChair only considers the lowest and highest classification given to a paper, and shows the review conflicts by a coloring scheme. For example, red means serious classification conflict – this occurs when a paper gets both review of types A and D; green means no conflict – for B and C type of scores.

START [3] uses Perl as its scripting language. It doesn't provide author registration, a feature that has been added independently by some users. It has the following features: paper submission and paper format conversion from Postscript to PDF; assign papers to reviewers using a round robin scheme taking into account their preferences; provides the reviewers facility to review papers on-line, including viewing the abstract and downloading of papers, and submit reviews; generate review reports; notify authors if their papers are accepted or rejected.

WitanWeb [4] has one more special user named (external) referees, who are chosen by reviewers to review some papers. WitanWeb includes the following major features: it supports paper submission in multiple formats; authors create and update their personal information as well as paper information. They can update and delete their papers. WitanWeb can assign papers to reviewers; reviewers or referees can review papers, generate reports; and notify authors if their papers are accepted or rejected.

The existing conference systems reviewed in [1] have following problems:

- Some of the systems reviewed in [1] are no longer supported.

- Most of these systems use the scripting language and CGI technology. However, CGI program is not efficient in handling concurrent user requests. Java Servlet is famous for security and has better performance for high-traffic web application (see details in section 3.1).

- The user interface is non-existent especially for administration functions and requires the administrator or user to run a number of different scripts to set up the system.

1.2 ConfSys System

The Conference Management System (ConfSys) presented in this paper is a web-based system, which provides support to three groups of users to perform their specific tasks that occur during a conference's flows of events. The flow is illustrated in Figure 2.2. It allows authors to register and submit papers to the system as well as to view the comments and final decisions of their own submitted paper on-line. ConfSys records the topics of expertise of the reviewers; the reviewers also have a facility to bid in an auction for papers to review and later to download and review the assigned paper via the Internet. During the debate period, reviewers can view blind comments from other reviewers so that they may modify their comments accordingly. It helps the PC by performing an automatic allocation of the submitted papers to the reviewers. The automatic allocation algorithm of Confsys assigns papers to reviewers based on their expertise and bids of

preference. It uses the DBLP database [6] in the automatic assignment of papers to avoid any conflict of interest and thus helps in the allocation of papers to the reviewer for a fair and impartial review of each paper.

ConfSys uses a three-tier client/server framework, and is implemented with Java Servlet, MySQL database on an Apache web server. As such, the system is independent of the operating system and is easily scalable to any other system environment. It has a user-friendly interface and many powerful functions. Since all development tools and environment are based on open source software, our system would be released for use to the academic community.

1.3 Organization of the Report

The report is organized into the following sections: Section 1 is the introduction. Section 2 presents the 3-tier architecture and flow of events in ConfSys. The software used in developing ConfSys is addressed in section 3. Section 4 describes major functions of the system, and the new features of the system are highlighted. Section 5 details the implementation of the PC auction process and automatic reviewer-paper allocation algorithm. Section 6 gives the data schema and description. Section 7 describes lessons we learned during the real test. Finally, the conclusions and future work are given in section 8.

2. Architecture and Sequence of Events

2.1 Architecture of ConfSys

This system adopts the three-tier client/server architecture. The three tiers are presentation tier, business logical tier and data storage tier.

Figure 2.1 shows the three-tier architecture used by ConfSys. The presentation layer provides the front-end to the user. The user uses a web browser to access the system via the Internet. To use ConfSys, the web browser (IE or Netscape) is the only tool installed on the client computer. The browser sends a request to the web server, and the web server processes the request, and sends the response to the browser. The communication protocol between web browser and web server can be HTTP or Secure Sockets Layer (SSL). If the SSL protocol is used, the user should type https:// rather than http:// in the URL of the browser. The SSL protocol includes measures to protect the confidentiality of the information. The web browser is unable to access the database server directly. The client should first send request to the web server, and the web server communicates with the MySQL server through JDBC API to further process the request, and returns the resulting data to the browser.

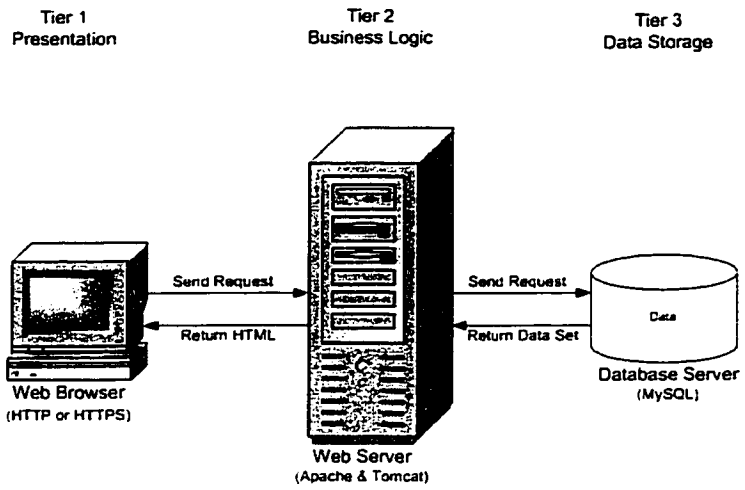

Figure 2.1 Three-tier Client/Server Architecture in ConfSys

In two-tier client/server architecture, the client processes and presents the entire application, whereas the server is only responsible for storing data. The inherent feature of this architecture results in heavy traffic in the network. In contrast, the three-tier client/server architecture adds one middle tier named business logic tier between clients and database server. Three-tier client/server architecture has many advantages over the two-tier client/server architecture.

First of all, it is a thin client solution. Client computers aren't required to install any application programs except a web browser program. As long as they have a web browser and Internet connection, they can access the system. So there are no special hardware or software requirements for the client.

Management and maintenance work is also reduced. When upgrading the system, all programs can be updated on the server, and no further work needs to be done on the

client computers. Users can easily upgrade their system for taking full advantage of new technology. The upgrade and any modifications to the application are transparent to all users.

Moreover, it reduces the load on the network and is able to balance the load. Once a program runs on the server, many clients can share the same result. Since the server usually has a better hardware configuration, the response speed is increased and performance is highly improved. If heavy traffic is detected, the server can transfer the access to other servers on the network.

The three-tier architecture has better security than the two-tier architecture. The web server always has many features to assure the security of the server. The administrator has more experience to make sure the server runs well at all times. The users of the system on the client-end sometimes don't have much knowledge of the system, and the operating system used on the personal computer has poor security features.

2.2 Flows of Events

In Figure 2.2, below, we outline the flow of events, the actors involved and the actions that are orchestrated by ConfSys; it thus indicates the needs of each group. It indicates the manual and the automatic actions supported by it.

The first group of users is the authors who, once the call for papers (CFP) is announced, the authors register their user IDs (UID) and receive their personal password (PW) via e-

mail to enable them to access ConfSys. Using the user id and the PW they can modify their personal profile, submit their papers and later view the results of the review directly through ConfSys. The second group of users is the reviewers who are emailed their user ID and password; using these, they register the topics of interests that would be used in selecting a subset of papers for the auction and automatic paper assignment processes. The reviewers can download papers, evaluate them, debate on-line, submit the result to ConfSys and finally view the comments from other reviewers as well as final decisions from the program chair (PC). The third group is the PC who is responsible for setting up the system for convocation of a meeting by inserting a program committee, the topics of the meeting and the important dates. PC can also initiate the automatic allocation of the papers to the reviewers and do the fine-tuning. In addition, the PC can make final decision on papers and initiates the notification to the authors and reviewers by email.

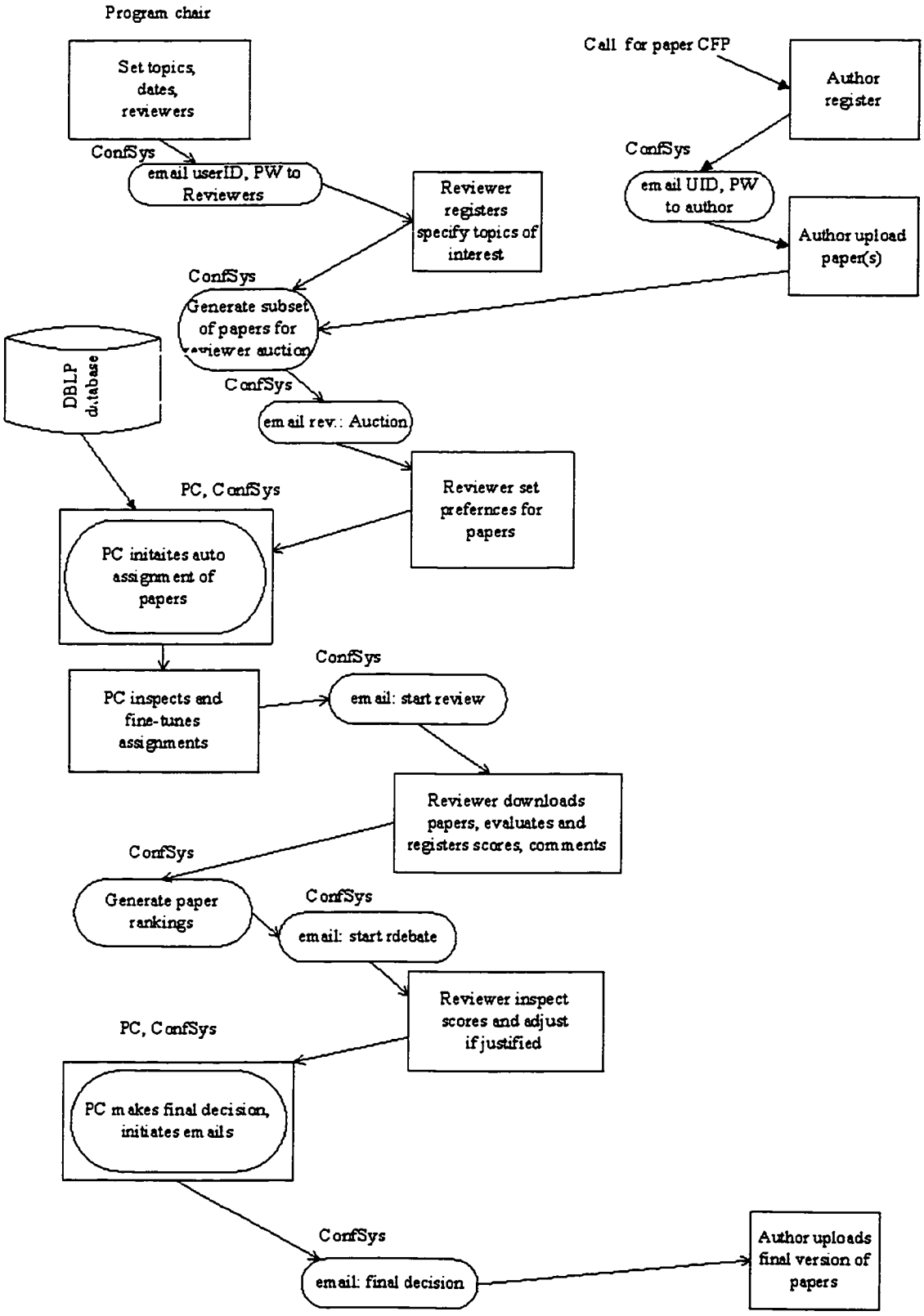

Figure 2.2 ConfSys Sequence of Events

3. Software Used in ConfSys

3.1 Java Servlet

ConfSys is implemented using Java Servlet. In the survey of conference management systems [1], most of them were developed using CGI script language such as Perl or Python. Java Servlet is becoming an increasingly popular alternative to the traditional CGI script languages. Script language is easy to learn, and very flexible, but compared to Java Servlet, it's not very powerful, portable, safe and efficient.

- **Powerful**

Servlet programmers can take full advantage of the core Java API. Because servlets follow a standardized framework, it's possible to use third-party Java classes and JavaBeans components. Java servlets allow programmers to easily handle cookies, track sessions and set HTTP headers, etc by providing many convenient classes.

- **Portable**

Java Servlet is actually a special kind of Java class running on the server-side. Since it is written with Java, they are portable across various operating systems and web servers.

- **Efficient**

Compared to CGI programming, Java Servlet has superior advantages to be used in developing web-based database application, particularly in supporting a large number of concurrent users to access the web site.

The web server is required to create a new process to invoke CGI scripts for each http request from the client. Of course, it costs server resources, and is hard to handle multiple concurrent users' requests for high demanding web site. We call it heavyweight processing since it involves a lot of overhead time and system resources. Each client request needs another different process. Using Java Servlet, the web server uses lightweight thread to handle multiple requests to one web page. During the first access to a web page, there is a time overhead. After the process is loaded to the server memory, the upcoming request will only invoke another thread to handle it. Therefore, no more overhead for other requests to the same Servlet class is required. Servlets can also provide persistent database connection to significantly reduce the overhead caused by database connection between each request to database server.

- **Safe**

Java is well known for its security. Java Servlet inherits security features from Java. First, Java Servlet forbids any strong type violation. Second, Servlets are free from memory management due to Java's garbage collection and lack of pointers. In addition, servlets can catch and handle all error exceptions safely; otherwise, the errors may crash the server.

3.2 MySQL

The back-end database of ConfSys is MySQL, the most popular open source database server. It is released under the GNU general public license. Users can download MySQL from their web site (http://www.mysql.com) for free. Moreover, source code is also

provided for any modifications to fit their special needs. One does not need to purchase the license except when it is embedded into commercial applications, but the cost is still very low compared to other commercial DBMS software.

According to [7], by January 2003, MySQL had become one of the most widely used databases in the world with over 4 million installations. ConfSys uses MySQL version 3.23.54, which has been declared stable. MySQL is a very fast speed database server. It can handle unlimited number of simultaneous users and 50 million records [8]. MySQL has implemented many features of other commercial software and continuously adds more new features without compromising the speed and stability.

Some features, such as stored procedures, triggers, and subselect, etc are still not supported in the version of MySQL we used in ConfSys. The newest release Version 4.1 supports subselect now. However, there are always workarounds available to satisfy your purpose. Other features are planned in the future versions of MySQL.

4. Major Functions and New Features

4.1 Overview of ConfSys

The system can be separated into three major subsystems according to three types of users. These subsystems are accessible via author, reviewer and Program Chair (PC) workstations.

4.1.1 Author workstation

As a new user, the author is required to register in our system. Once they establish their profile, the system will automatically generate a password to the author and send it to them by e-mail. The author is able to use their username and password to log into the system. The author's menu is shown in Figure 4.1.

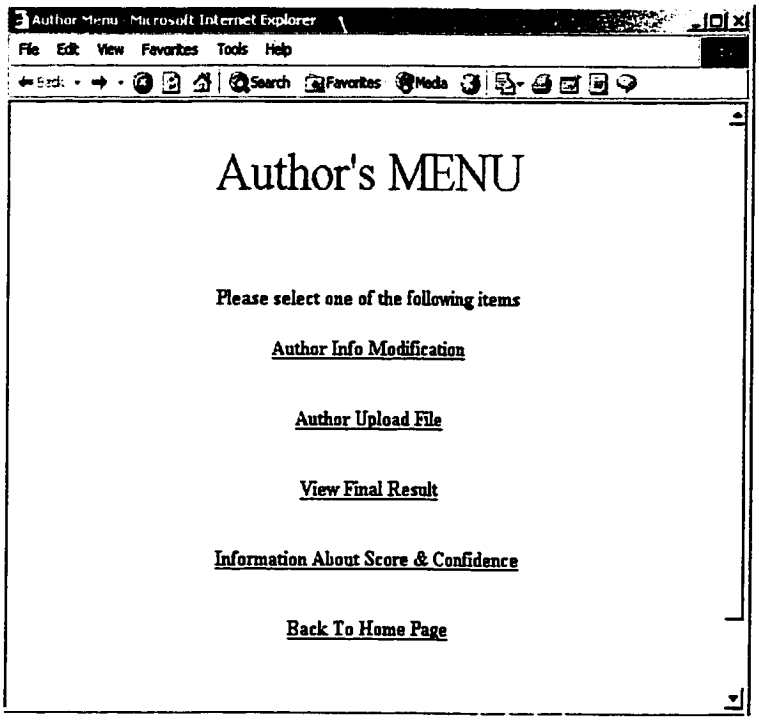

Figure 4.1 Author's Menu

The system allows the author to modify their profile. The system supports upload of files for paper in many formats, such as PDF, Doc, PS, TXT, etc. After the review process ends, the PC will send an e-mail to authors to inform them to log into the system to check the final result for their papers.

4.1.2 Reviewer workstation

The PC is responsible for choosing the program committee and hence knows all these reviewers. It is, thus, possible for the PC to choose the reviewer id, and passwords for them and communicate with them via e-mail. After reviewers get their id and password, they are able to log into the system. Figure 4.2 shows the reviewer's menu. They can update their personal profile and indicate their topics of interests for review. We use an algorithm to generate a number of papers for an auction process where the reviewer indicates their preferences. The automatic allocation algorithm for papers will consider the reviewers' preference as well as their expertise. In addition, the algorithm uses DBLP to make sure the review assignment is fair for all papers. For details about the algorithm refer to the section 5.2. Before the PC makes the final decision for the papers, ConfSys sends an e-mail to all reviewers to view other reviewers' comments and scores, and reviewers could modify their own comments if it's necessary in a debate process.

After the debate process is over, the reviewers can view the comments and final decision of all papers they reviewed. The information provided to the reviewer contains other reviewers' opinions, scores and the final decision made by PC.

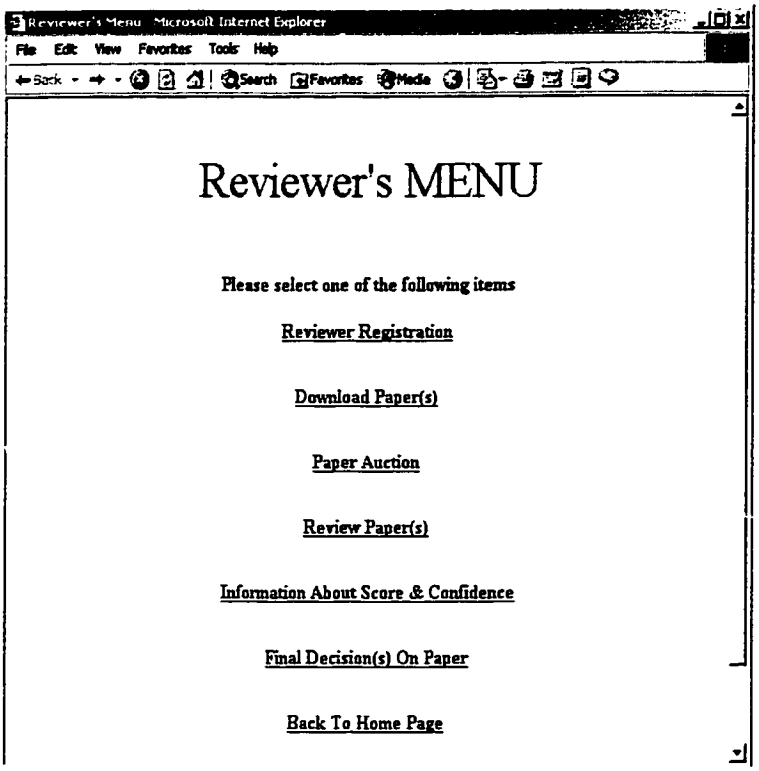

Figure 4.2 Reviewer's Menu

4.1.3 PC workstation

The PC has full control over the system. Figure 4.3 shows PC's menu. The PC can add, edit and delete authors and reviewers profile including their username, password and other information with Author Control and Reviewer Control functions.

The PC can initialize some parameters for the system such as the starting date of review and some parameters used by the reviewer auction and allocation process. The functions provide flexibility for use in any academic conference. The PC simply sets the parameters via the graphic user interface provided by the system.

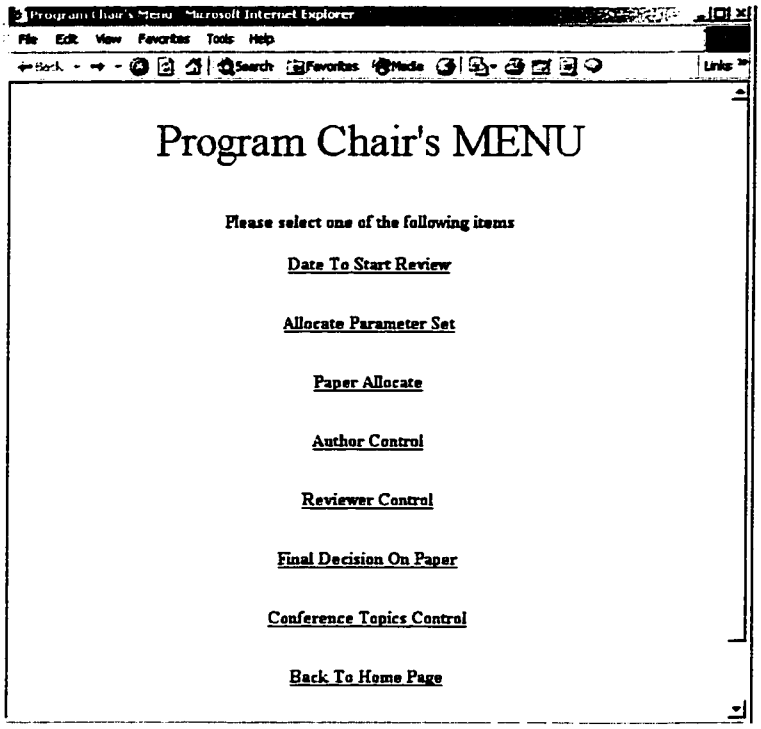

Figure 4.3 Program Chair's Menu

The system provides two methods, automatic and manual allocation, to assign papers to reviewers. The algorithm used in the system allows most reviewers to be assigned papers based on their preferences and other restrictions required by the allocation. The allocation will be fair for all reviewers and will not violate the allocation rules and other constraints. For details about the allocation algorithm, refer to the section 5.2. The PC can use the manual allocation function to make modifications and allocations without any restrictions.

Finally, the PC is able to view the review result of all reviewers, make the final decision (accept or reject) on the paper and initiate the system to send e-mails to authors and reviewers to check the final result.

4.2 New Features in Conference System

4.2.1 New Author Registration

As a new author of ConfSys, they should first register using the registration form (see Figure 4.4). After they fill out all the required information and click the "submit" button, an e-mail will be sent containing their user name, password and conference information to them. The password is generated by ConfSys.

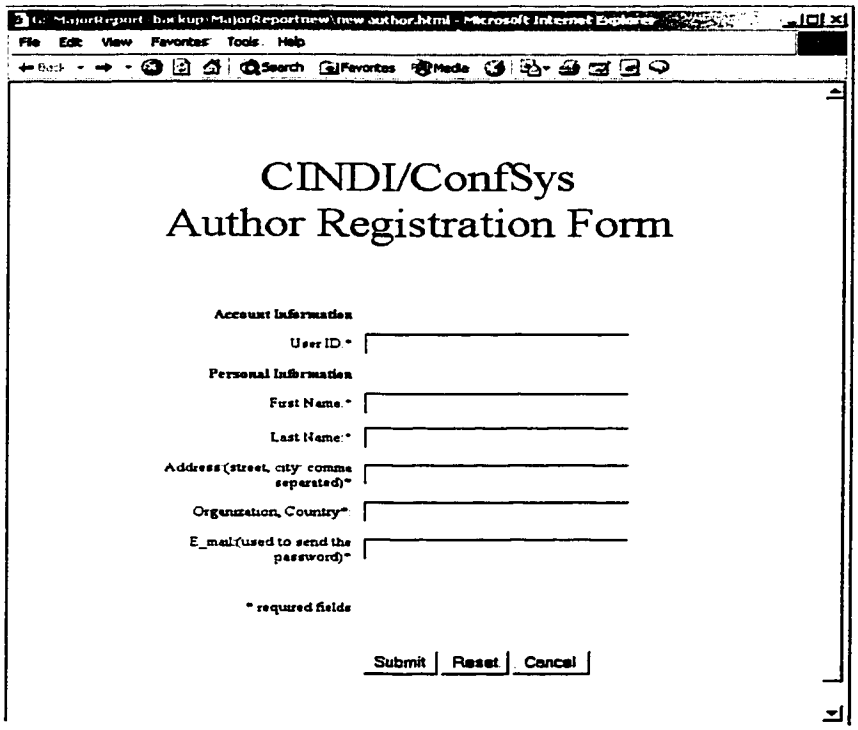

Figure 4.4 New Author Registration Form

4.2.2 Author Sign in Form

After the author receives the e-mail, he/she can log into the system using the username and password. Figure 4.5 shows the author sign in form.

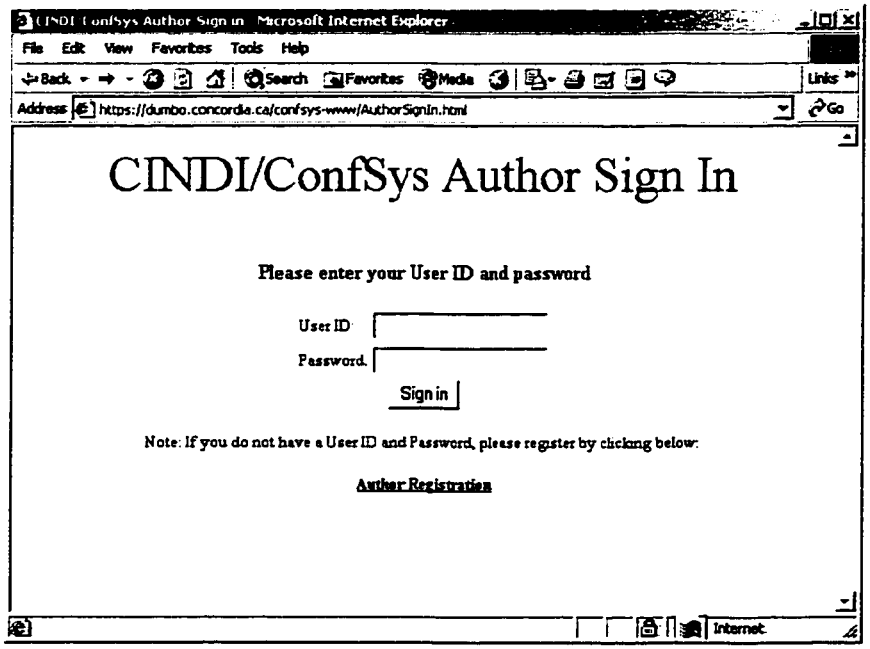

Figure 4.5 Author Sign in Form

4.2.3 Modify Personal Profile

Authors can modify their personal profile at any time (see Figure 4.6, 4.7). The following automatic paper allocation will consider the author's organization. The paper cannot be assigned to the reviewer who is working in the same organization as the author.

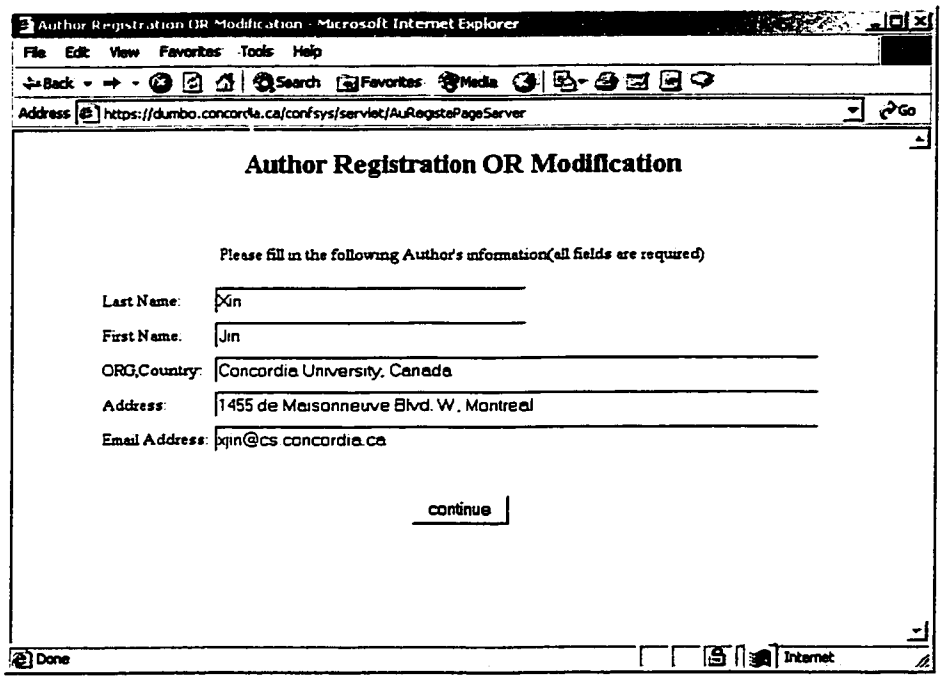

Figure 4.6 Editing Author's Personal Profile

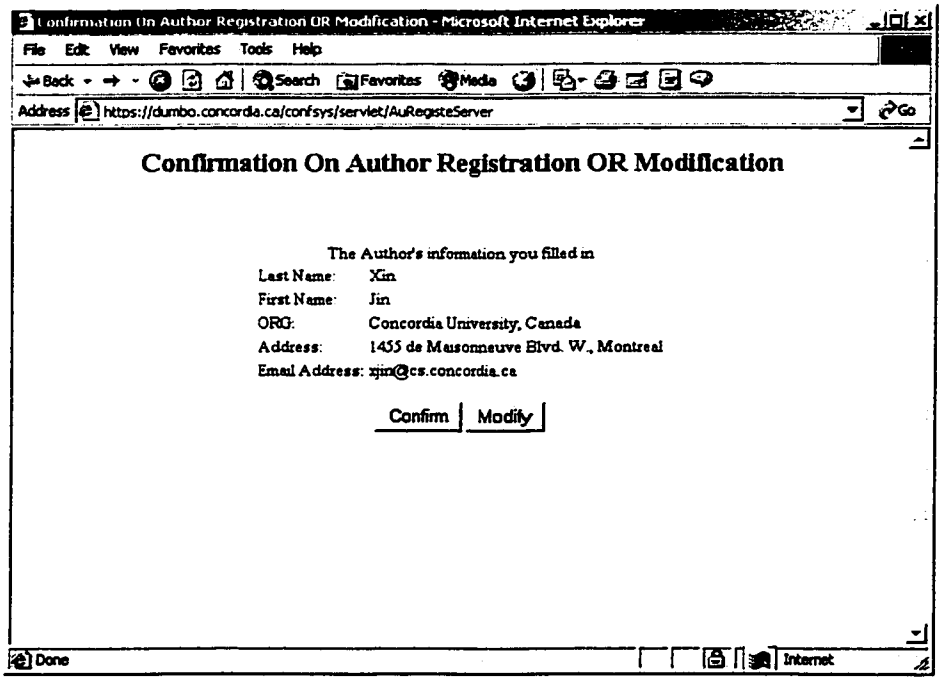

Figure 4.7 Confirmation on Modification

4.2.4 Author Control

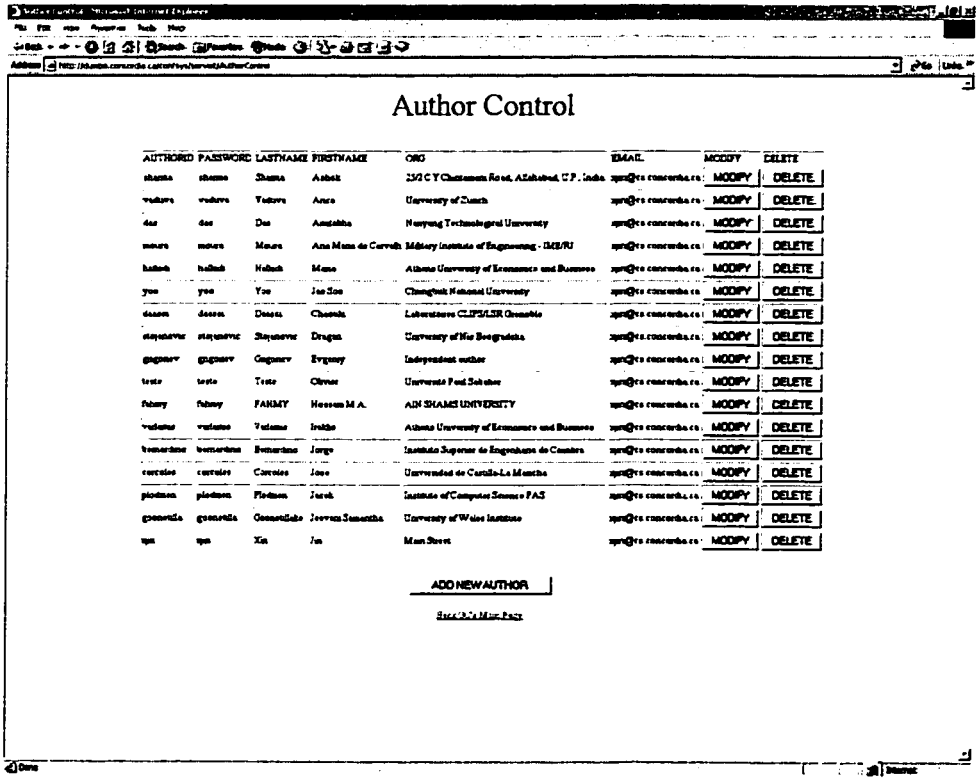

Figure 4.8 Author Control

In actual use, we found it's quite useful for the PC to view all the authors' information. In a special case, some authors might not be able to register on ConfSys. Author Control functions (see Figure 4.8) allow the PC to register such authors. The PC can use this function to modify and delete author's information. At the end of the uploading paper stage, the PC can recheck author's information to make sure the information is accurate. In case the author does not follow instructions, the PC may have to modify it.

4.2.5 Send e-mail to PC and Author to Check the Final Result

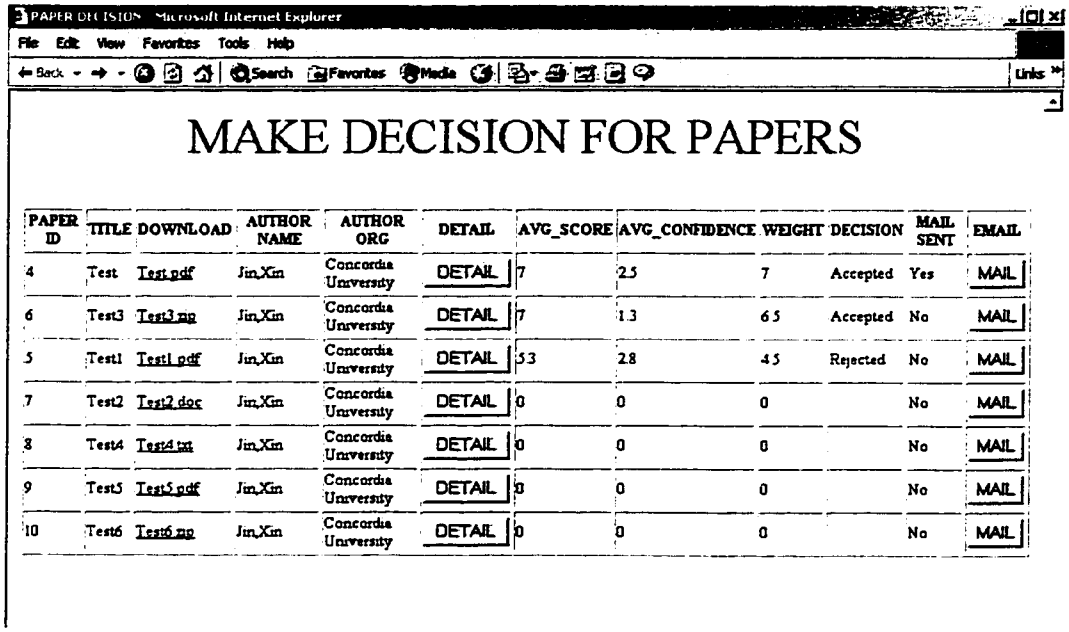

Figure 4.9 PC Sends E-mail to Authors and Reviewers

In this version of our system, after the PC makes the final decision, he/she will use this interface (Figure 4.9) to send e-mail to inform authors and reviewers the final results by clicking "MAIL" button. However, the e-mail won't show any detailed information about the final result. It directs the users to go to the web site to check the final result. This function is implemented using Java Mail.

4.2.6 Author, Reviewer and PC View Final Result

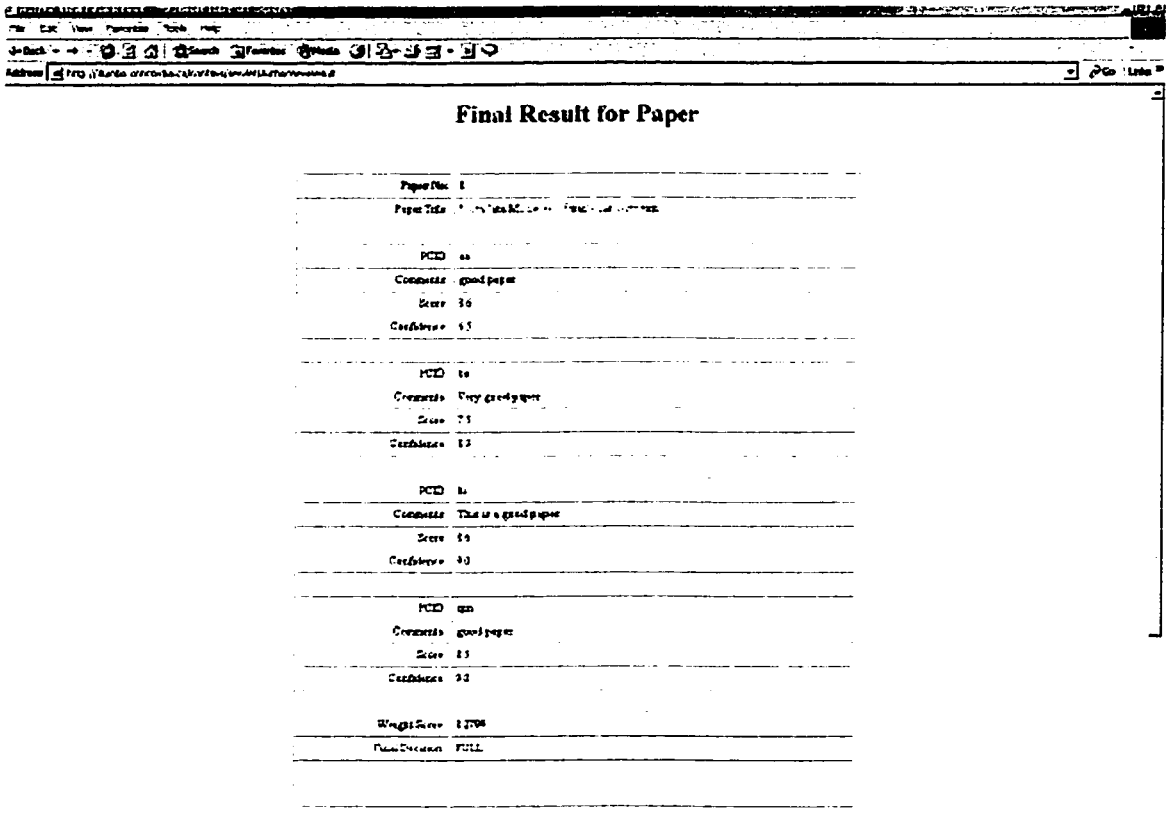

Figure 4.10 Author Views Final Result

For the author, as shown in Figure 4.10, they can only view their own submitted papers. They can see all the reviewer's ids, comments, scores, confidence and the weighted score as well as final decision for the paper. If they submit more than one paper, they can see results for all of them. The final result is available only after the PC makes the final decision and sends e-mail to them. Otherwise, authors cannot see any information submitted by the reviewer.

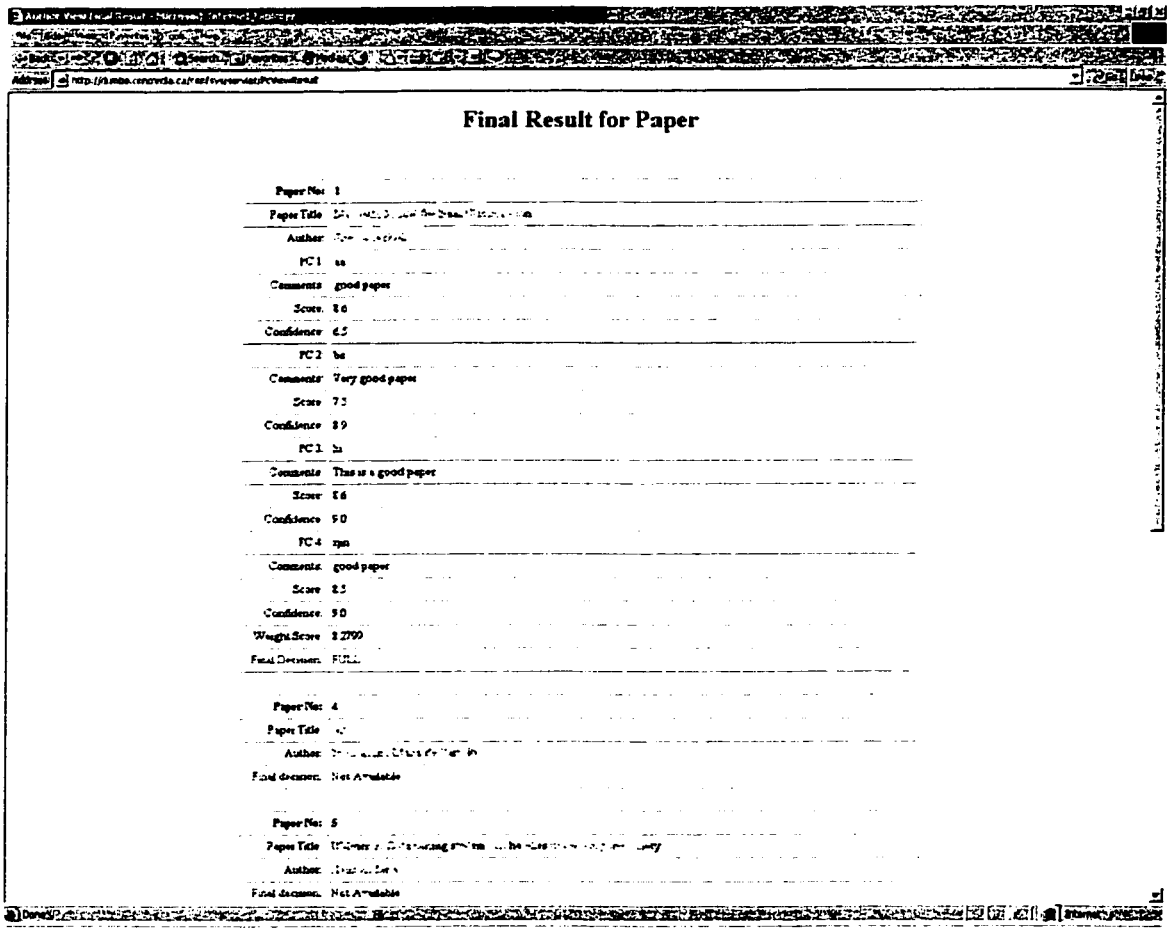

Figure 4.11 Reviewer Views Final Result

Reviewers can view final results of the papers that they reviewed (see Figure 4.11). Other reviewers' id, scores, confidence ratings, and weighted score, etc will be shown in the window. The final result is only available after the PC makes the final decision and sends e-mail to them. The interface can be used optionally for reviewers' debate before the final decision is made.

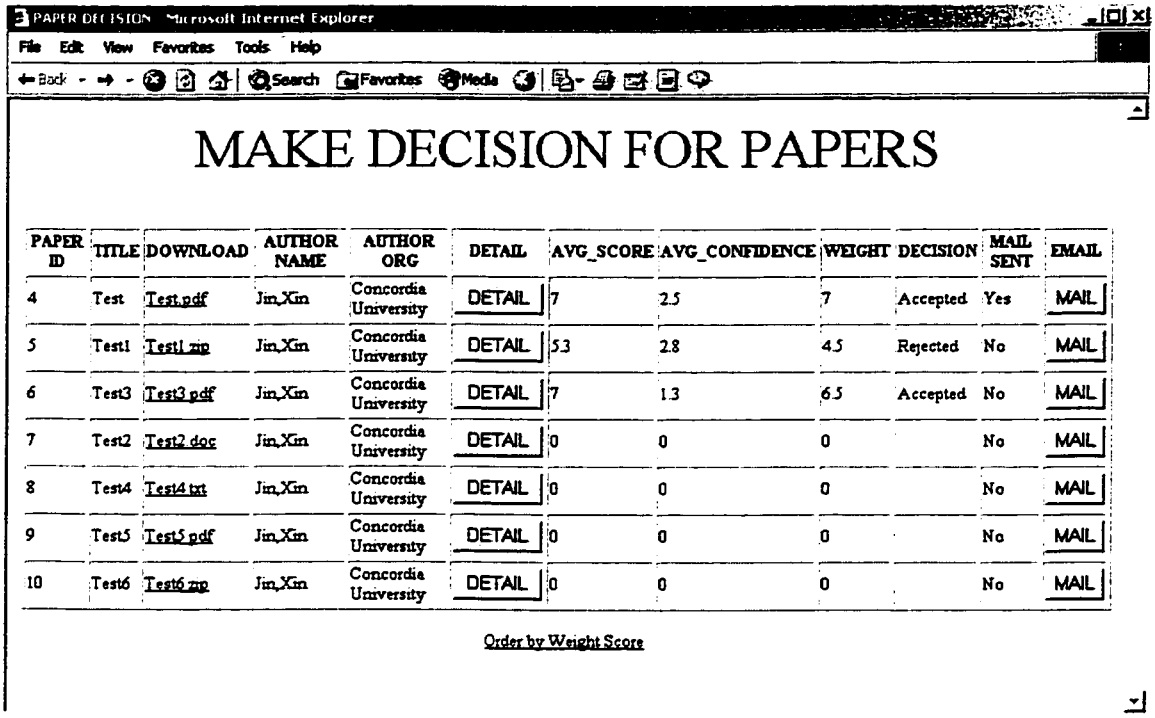

Figure 4.12 PC Views Final Result

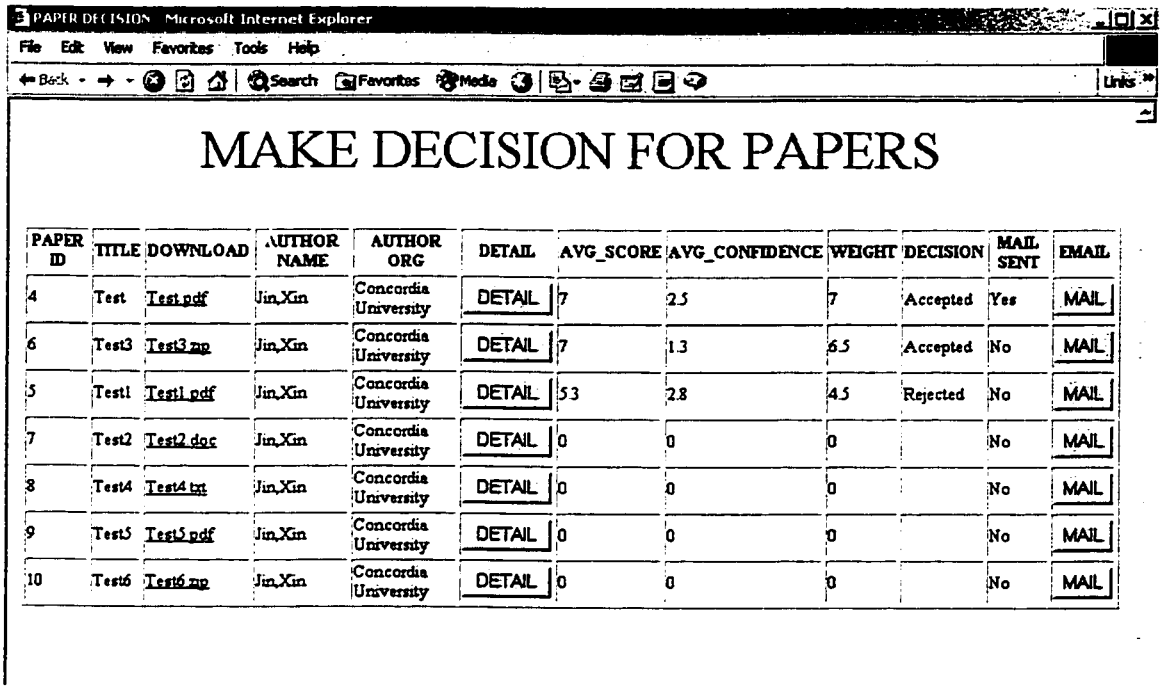

Figure 4.13 PC Views Final Result Ordered by Weighted Score

This functionality was implemented in the first version of ConfSys for the PC (see Figure 4.12). Now a text link called "Order by weighted score" is added, which allows the PC to see the final results ordered by weighted scores in a new window (shown in Figure 4.13). This feature will provide a different view to the PC, which will help in making the final decision for each paper.

4.2.7 Allocate Parameter Set

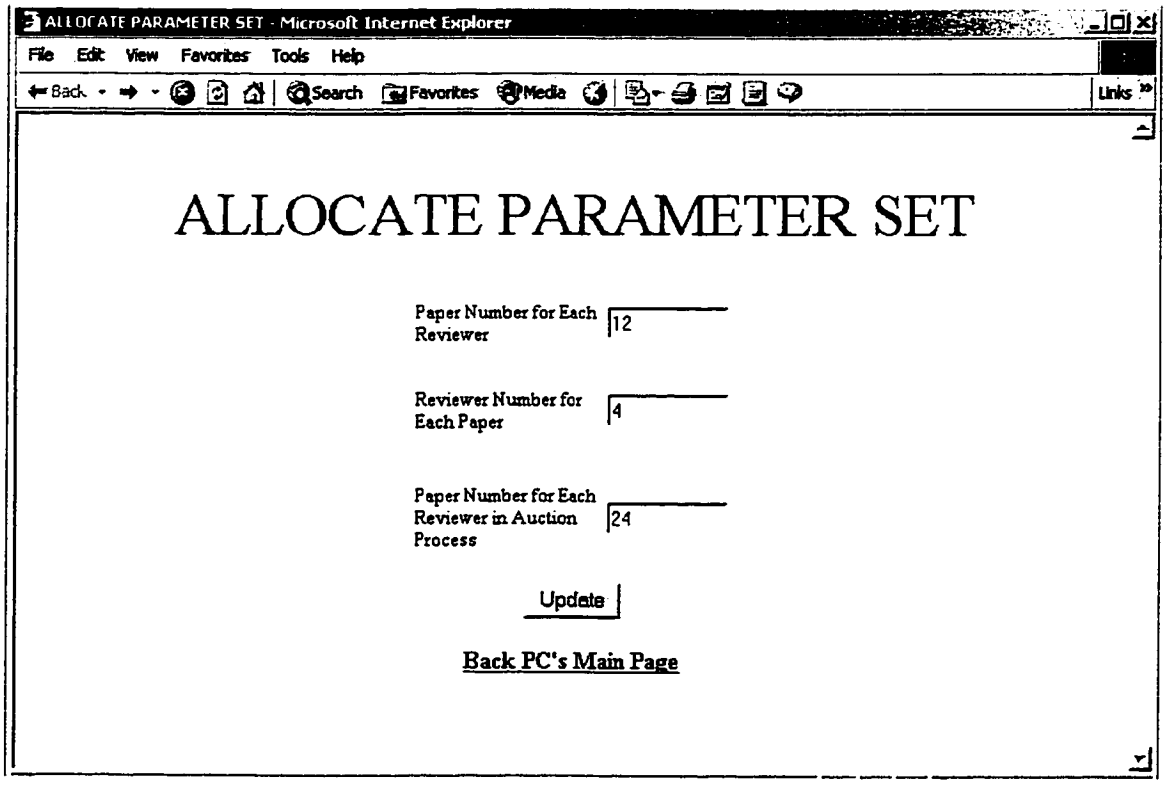

Figure 4.14 Allocate Parameter Set

In this version of ConfSys, an additional parameter set is provided; this is "paper number for each reviewer in auction process". This gives the number of papers to be shown to the reviewers in the auction process. Using this interface (see Figure 4.14), the PC can establish the total numbers of papers for reviewer's auction. For example, if the PC inputs

12, that means 12 papers will be generated for reviewers to choose from during the auction process. For this number of papers, some of them are retrieved based on the matching topics, and the remaining papers are randomly selected from the database. If more than 12 papers' topics match the reviewer's topics of interests, all papers will still be displayed, but no random paper will be generated. In this way, too many papers unrelated to the topics of interests of reviewers, are not shown, which would otherwise make the auction process a bit tedious. In addition to that it gives more options to those reviewers who don't have many papers matching topics of interests.

4.2.8 Reviewer Auction

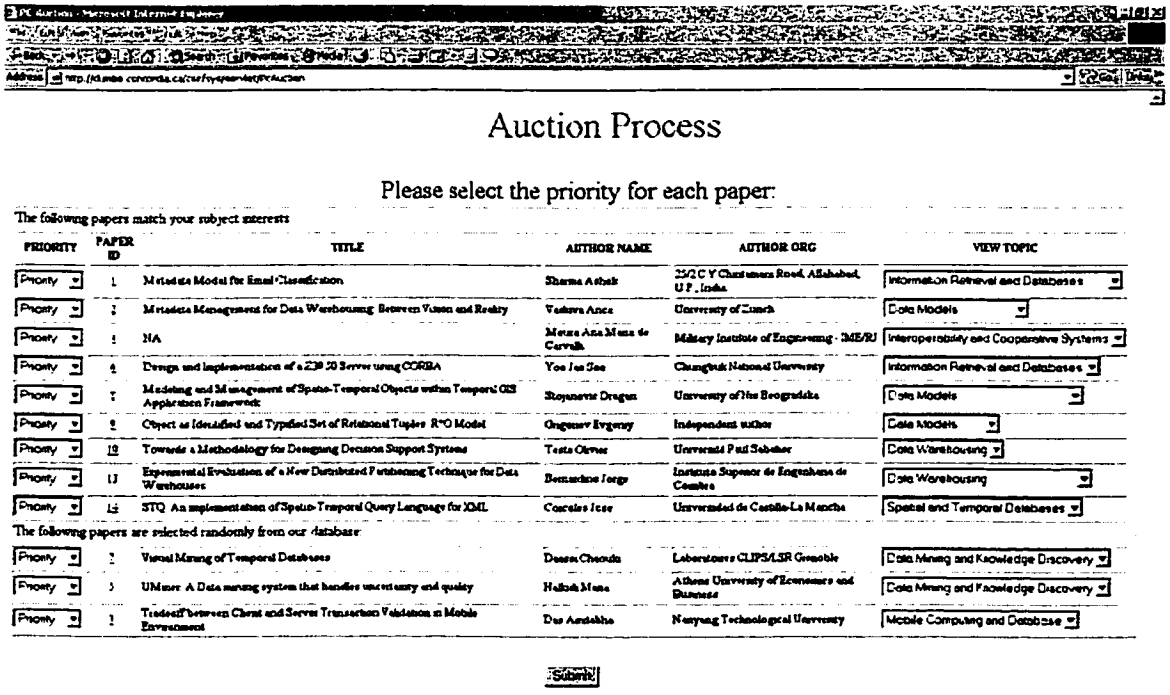

Figure 4.15 Reviewer Auction Process

The reviewers use this interface (Figure 4.15) to choose priorities for each paper. They can optionally choose *high*, *low* or *average*. If they have no interest in a paper, they do

not indicate any option. They can also view the paper's abstract and topics. The total number of papers displayed is based on the allocated parameter set by the PC. First, it will select those papers, which match the reviewer's interests. If the total number is still less than the parameter set, more papers will be generated randomly from the remaining submitted papers.

4.2.9 Automatic Paper Allocation

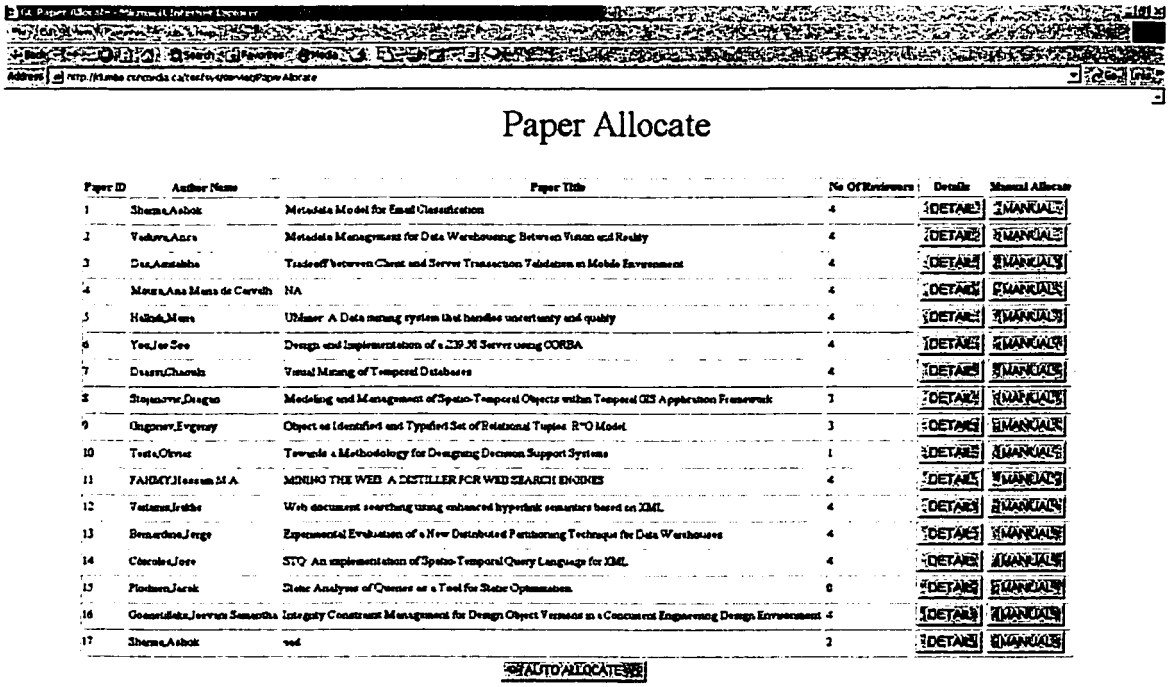

Figure 4.16 Automatic Paper Allocation

Here, a new algorithm for automatic allocation is introduced. In particular, reviewers' preferences for different papers are considered first. In section 5.2, the algorithm used in automatic allocation will be explained in detail. Figure 4.16 shows the automatic allocation result.

5. Implementation

5.1 Reviewer Auction

In the reviewer auction process, we consider whether this is the first time for the reviewer to view this web page, and handle it separately.

If this is the first time, we first show all papers that match the topic of interests selected by the reviewer. If the numbers of papers are still less than the papers limits set by the PC then more papers can be generated by the system randomly from the remaining unselected papers until the number of papers reach the limit. The choices made by the reviewer along with the papers pre-selected for the auction are stored in the database.

Suppose the reviewer returns for revising the choice of the auction process. The reviewer in this case wants to see the exact same papers and the priority as chosen the last time. The information relating to the previous auction visit which is stored in the database is retrieved, and displayed.

5.2 Automatic Paper Allocation

5.2.1 Overview

This system can be used either automatic or manual allocation. For manual allocation, the PC can allocate any papers to any reviewers without any restrictions. The other way to assign papers to reviewers is to use the automatic paper allocation. The following algorithm is used for the automatic allocation.

In this algorithm, following factors affect the assignment of papers:

- Reviewer's bid during auction: the information is collected by auction process.

- Reviewer's topics of interest: when reviewers register, they select their topics of interests.

- The maximum number of reviewers for each paper and the maximum number of papers that can be assigned to each reviewer. The PC can set the limits using allocation parameter set form.

If the author of the paper and the reviewer work in the same organization, or the author of the paper and the reviewer were co-authors in the past, the reviewer is not eligible to review the paper. DBLP database is used for checking these constraints. DBLP server maintained by Professor Michael Ley of University Trier in Germany provides bibliographic information such as the article's title, the author as well as co-author information on major Computer Science journals and proceedings. DBLP has collected 350,000 articles by February 2002. A subsystem, which converts the XML file to MySQL database, has been implemented in ConfSys.

CyberChair and START also implemented algorithms to allocate papers to reviewers based on expertise and preferences. However, they simplify the problem by assigning any number of papers to reviewers. To promise highest satisfaction and meet the mentioned constraint are two essential issues to be solved.

This allocation algorithm based on preferences could be very complex. The algorithm used in ConfSys is simpler to implement and the allocation constraints, in most cases, are satisfied without additional fine-tuning by the PC. The algorithm attempts to assign most reviewers paper that has been assigned higher priority by them in auction process.

5.2.2 Comparisons of Algorithm in Different Versions

In the previous version of this system, some constraints have been enforced. For example, the allocation process will check if the paper's author and reviewer work in the same organization, or if the paper's author and the reviewer were co-authors based on information from the DBLP system. The new algorithm is different from the older version as follows:

First, the old algorithm only assigns the paper to the reviewer if the paper's subject matches the reviewer's subject interests. However, the new algorithm will assign the paper to the reviewer if the reviewer gives preference to this paper, even though the paper and the reviewer's interests are not overlapping. So the algorithm actually first considers the reviewer's preference of papers for reviewing, and then considers if the topics of interests are same.

Second, the old algorithm assigns each paper according to reviewer's id order. The reviewer id's order decides who will be assigned first. The consequence of this algorithm is the reviewer's priority to get the paper based on their ID. Of course, it's not fair for all

reviewers. The new algorithm allows more reviewers to get papers; meanwhile, it tries to satisfy each reviewer's preference.

Finally, the papers are assigned in order by their paper id in the old algorithm. As a result, the last papers are usually not assigned. The new algorithm will categorize and sort all papers to assure most of papers are assigned.

5.2.3 Automatic Algorithm in ConfSys

The algorithm gives higher priority to the reviewer's preference before considering a match of topics. During the setup the PC would assign the number of reviewers for each paper (called paper limit, LP); and the maximum number of papers to assign to each reviewer (called reviewer limit, LR).

The automatic allocation algorithm has the following features:

- **Collect paper preference information from reviewer**

In the auction process, the reviewer can assign high, average, low preference or no interests, for each paper.

- **Categorize papers**

Once the auction process is over, the automatic allocation algorithm categorizes all submitted papers into four groups: High group (H), aVerage group (V), Low group (L) and No interests group (N). A paper must be in one of the above four groups and no paper will be in more than one group at the same time.

The rules used for categorization are as follows:

1. H group

For any paper, if at least one reviewer assigns a high priority, the paper will be categorized into a group, called H group. For a paper in H group, we allow other reviewers to assign lower levels of priority such as average, low or as no interests. In other words, for a given paper, as long as it gets one high priority, it is categorized in the H group.

2. V group

In this group, the highest priority assigned by at least one reviewer is "average" priority. Of course, the algorithm allows other reviewers to assign low priority or no interests. However, there cannot be a "high" priority assigned to any paper in this group. In such case, the paper should be categorized in the H group.

3. L group

Each paper in this group is assigned at least one "low" priority by a reviewer. Other reviewers can assign "low" or no interests for the paper.

4. N group

This group of papers is assigned no bids by any reviewers. Two possible scenarios to handle papers in this group are: (1) The paper's topic matches some reviewers' topic of interest. The automatic allocation can handle this case. (2) No such matches. The PC should assign these papers manually.

- **Sort papers**

The automatic allocation algorithm sorts the paper in the order: H group, V group, L group and N group. In each category, it sorts the paper based on their count of selection by reviewers.

Next, we take papers in H group as an example to explain how the algorithm to categorize and sort papers according to the rules.

If there are n papers (n_1 papers in H group) and m reviewers, table 5.1 shows the sorted papers in the H group. The notation "*" stands for 1 or 0. The value of "*" is 1 if the reviewer assigns "high" to a specific paper. Otherwise, the value of "*" is 0.

Table 5.1 Sort Papers in H Group

Category	Paper	Reviewer				Total Num of H assigned to P_i
		R_1	R_2	...	R_m	
H Group	P_1	*	*	...	*	$H(P_1)$
	P_2	*	*	...	*	$H(P_2)$
	...	...	...	...	...	...
	Pn_1	*	*	...	*	$H(Pn_1)$

For each row of table 5.1, at least one "high" is assigned to paper i (i = 1, 2, ..., n_1). If not, the paper will be categorized in one of other groups, but not H group. In other words, it is possible that some of the columns of this table (rounded by the bold lines) are all 0, but none of the rows would be all 0s.

The rule for sorting papers is $H(P_1) >= H(P_2) >= ... >= H(Pn_1)$. That means the algorithm sorts the paper by total number of "high" priorities assigned to the paper. The paper assigned maximum number of "high" priorities is allocated first.

- **Sort reviewers**

In our database, there is a table named "auction_chosensum", which is used to store summarized information about the bids of auction for each reviewer. It has three attributes: pcid (reviewer id), priorid (priority id), chosensum (total number of selection for each priority). For example, the records for a reviewer whose id is 100 in the table are shown in table 5.2.

Table 5.2 Records in auction_chosensum Table

Reviewer ID	Priority ID	ChosenSum
100	1	5
100	2	3
100	3	2

In the auction_chosensum table, the values of priority id can be 1, 2 or 3. 1 is the priority id for "High", 2 is for "Average" and 3 is for "Low". As long as a reviewer takes part in the auction, 3 records (because of three kinds of priorities) are inserted in the table for each reviewer. The value of column ChosenSum stores the total number of a priority assigned to all papers by a specific reviewer. The notation $H(R_j)$ is used later to show the sum of "high"s assigned by the reviewer whose id is j; similarly, $V(R_j)$ is for the sum of "average"s and $L(R_j)$ stands for the sum of "low"s. In Table 5.2, $H(R_j) = 5$, $V(R_j) = 3$, $L(R_j) = 2$ and $j = 100$.

For the assigned paper, its assignment is based on the following factor: paper limit LP and reviewer limit LR. At the beginning of allocation we should also choose the correct assignment order of reviewers. Again, we use the papers in H group as an example to illustrate the sorting of reviewers. We assume that there are 6 reviewers. Since the sorting process for all papers are the same, we show only the allocation for the paper P_1. The only relevant limit for sorting reviewers is paper limit LP. We assume at most 4 reviewers can review a paper. Table 5.3 shows the priority assigned to P_1 by the six reviewers. The data shown in table 5.3 is derived from the table pc_auction.

Table 5.3 The Priority Assigned to P_1 by Reviewers

Category	Paper	Reviewer					
		R_1	R_2	R_3	R_4	R_5	R_6
H Group	P_1	H	V	N	L	H	V
	...						
	...						
	...						

The sorted reviewers are shown in table 5.4. The algorithm attempts to assign papers to the reviewers who assigned a higher priority in the auction, so that the order of assigning a paper is always from the reviewers who bid "High" to the reviewers who bid "Low" to a specific paper. If a reviewer selects fewer papers for the preference in one priority level, the reviewer should be assigned paper earlier. Otherwise, the reviewer may have no chance to be assigned any papers. In our example, the reviewer who assigned the least number of "high" priorities is allocated first. Therefore, in this example, $H(R_1) <= H(R_5)$. If there is no more "high" priority assigned to the paper and the paper limit LP is not

reached yet, the algorithm will continue to allocate papers to the reviewers who assigned "average" to this paper. The reviewer who assigned the least number of "average" priorities is allocated first. Hence, $V(R_6) <= V(R_2)$. The values of $H(R_1)$, $H(R_5)$, $V(R_6)$, $V(R_2)$ and $L(R_4)$ can be retrieved from the auction_chosensum table. Each time a reviewer is assigned a paper, the total number of priorities at this priority level for this reviewer is incremented by 1, and the value is saved to the column chosensum in table auction_chosensum. Here, we take the reviewer R_6 as an example. R_6 is assigned the paper P_1 because the reviewer bids an "average" to the paper. So the value of chosensum field corresponding to R_6 and "average" priority will be incremented by 1. It attempts to give other reviewers' a chance so that each reviewer is assigned almost the same number of papers to review.

Table 5.4 Sorted Reviewers

Category	Paper	Reviewer					
		R_1	R_5	R_6	R_2	R_4	R_3
H Group	P_1	H	H	V	V	L	L
	...						
	...						
	...						
Total num of the priority assigned to papers by R_j (j= 1,...,6)		$H(R_1)$	$H(R_5)$	$V(R_6)$	$V(R_2)$	$L(R_4)$	$L(R_3)$

Papers in V group and Low group are similarly sorted.

- **Allocate Papers**

If the number of priorities assigned to a paper is ≥ LP, then the paper may be assigned based only on the priorities. If the number of priorities assigned to a paper is ≤ LP, then

the algorithm can assign paper directly to those reviewers with high priority followed by lower priority until the limit LP is reached. In the previous example, only two reviewers (R_1 and R_5) bid "High" to the paper P_1. Since the limit LP is 4, two more reviewers (R_6 and R_2) who bid "Average" are also assigned. Then the limit LP is reached, so no more reviewers are assigned. If the number of reviewers assigned priorities to a paper is < LP then matching of topic of the paper with the reviewer's interest is used.

For each assignment, the algorithm checks if one of the authors and the reviewer work in the same organization, or if paper's author and the reviewer were co-authors of a paper using DBLP database.

For each assignment, the algorithm checks both limit LP and limit LR.

After the first three groups have been allocated, the algorithm checks the remaining papers which may have the same topics as the reviewer's topics of interest. The algorithm assigns papers to those reviewers.

After this, the algorithm will not be able to assign the remaining papers. If any, the PC is required to allocate them manually during the fine-tuning stage.

5.2.4 Performance and Limitations

In the seventh International Database Engineering and Applications Symposium (IDEAS' 03), we used automatic allocation function to assign 75 papers to 48 reviewers. 42

reviewers had participated in the auction process. It takes about 10 minutes to complete the automatic allocation on a system with Athlon XP 1700 and 1G MB of memory. The parameter sets used by the automatic allocation are shown in table 5.5. For two papers the system was able to assign only two reviewers, so the PC had to manually assign two more reviewers for these papers. The performance was acceptable.

Table 5.5 Automatic Allocation Parameter Sets in IDEAS'03

Max. number of papers for each reviewer (LR)	Max. number of reviewers for each paper (LP)	Number of papers for auction process
12	4	24

The automatic algorithm discussed in this report provides a practical solution to the assignment problem. It doesn't violate the reviewer interests and paper constraints while trying to satisfy the reviewers' preference.

This algorithm is used on the basis of the auction process. Without the auction process, the algorithm will simply assign paper according to the topics of interests; therefore we cannot see any improvements from the older version. This algorithm doesn't enforce the minimum number of papers for each reviewer although it can be implemented if required. Our design is based on the idea that it's not reasonable to assign a paper to a reviewer automatically if the reviewer doesn't indicate preference for the paper in auction and the paper's topic doesn't match any topics of interest of the reviewer.

The implemented algorithm has a limitation: the algorithm cannot detect all conflicts of interests of assigned papers using DBLP database, for example if the author of a paper is an ex-student of the reviewer and they had no joint publications recorded in DBLP.

6. Database Schema and Table Description

The ConfSys database comprises 16 tables including existing 12 tables in the first version of this system.

In this section, some modifications to the data schema will be explained. In order to meet the new requirements some new tables are added. Attributes and descriptions of the data schema will be described.

6.1 Modifications

(1) Table author, coauthor, gc, and pc all have lastname and firstname attributes. The type is varchar(20). Now we change it to varchar(50) because some names may be larger than 20 characters.

(2) Add one attribute papers_auction in the table allocate_limit. This field is used to set the parameters of showing how many papers to reviewers during the auction process. So reviewers can choose different levels of interests from those numbers of papers. Table 6.1 and table 6.2 show different versions of table allocate_limit; the new attribute is in bold font.

Table 6.1 Table allocate_limit (previous version) Description

Attribute Name	Description
papers_pc	Maximum number of papers for each reviewer
pcs_paper	Maximum number of reviewers for each paper

Table 6.2 Table allocate_limit (new version) Description

Attribute Name	Description
papers_pc	maximum number of papers for each reviewer
pcs_paper	maximum number of reviewers for each paper
papers_auction	maximum number of papers in auction process for each reviewer.

(3) Since we didn't have author control in the first version of this system, table author only had paid field. Table 6.3 is the older version of the table author's description. The paid in the table is the primary key and foreign key in table paper. One problem in this design is that it is hard to handle the case when one author may submit many papers. Authors are required to register and upload files at the same time in the older version. If they want to upload another paper, they should re-type the registration information again. So the database stores author's information in the table author repeatedly. The redundancy problem makes keeping data consistency very difficult. Now we separate the registration and uploading paper functions. Authors can upload more than one paper without providing registration information many times.

Table 6.3 Table Author (previous version) Description

Attribute Name	Description
paid	paper id
lastname	last name
firstname	first name
org	organization
address	address
email	email

Below (table 6.4) is new version of table author. The added attributes authorid and password are shown in bold font. The attribute paid has been removed.

Table 6.4 Table author (new version) Description

Attribute Name	Description
authorid	author id, primary key
password	Password
lastname	last name
firstname	first name
org	Organization
address	Address
email	Email

Table 6.5 shows the new version of table paper. The attribute authorid is added. Authorid is the foreign key that refers to the attribute authorid in table author.

Table 6.5 Table Paper (new version) Description

Attribute Name	Description
paid	paper id, primary key
authorid	author id, foreign key
total_pc	number of reviews assigned to
paperTitle	paper title
paperfilename	paper file name
paperabstract	paper abstract
review_result	PC's review result
decision	PC's final decision
weighted_score	not used; this value is calculated in the program
mailsent	a flag to show if the PC sends e-mail or not to author and reviewer for the paper

6.2 New tables in the New Version

The system in this new version provides paper automatic allocation process based on the PC's preference. In order to implement it, the following tables have been added to the database.

- pc_auction

  Primary key: pcid and paid

Foreign key: pcid references table pc; Paid references table paper; Priorid references table priority

Table 6.6 Table pc_auction Description

Attribute Name	Description
pcid	reviewer id
paid	paper id
priorid	reviewer's levels of interests to review a specific paper; value can be 1 (high), 2(average), 3(low)

- priority

  Primary key: priorid

Table 6.7 Table priority Description

Attribute Name	Description
priorid	priority id
priorname	priority name, value can be high, average, low

- pa_random

  Notes: This table only stores random paper ids that are used for a specific reviewer who did an auction process to assure that on subsequent re-visits, the system will not generate new random paid for this reviewer. A reviewer can modify his/her previous auctions.

  Primary key: pcid and paid

  Foreign key: pcid references table pc, and paid references table paper

Table 6.8 Table pa_random Description

Attribute Name	Description
pcid	reviewer id
paid	paper id

- auction_chosensum

  Notes: This table is used to automatic paper allocation process.

  Primary key: pcid and priorid

  Foreign key: pcid references table pc, and priorid references table priority

Table 6.9 Table auction_chosensum Description

Attribute Name	Description
pcid	reviewer id
priorid	priority id
chosensum	the number of times the reviewer assigned a priority to a specific paper.

7. Test

ConfSys was used in the IDEAS'03 conference. Through this real-use testing, more bugs were found and fixed. The lessons we learnt from the testing are as follows:

- Concurrency issues

  Usually, a conference has a hundred and even a thousand authors. The program for authors should consider concurrency issues carefully. The uploading file form is a very important form for authors. We found that most of authors intended to upload their files just before the deadline, which makes such a problem more serious. The program of uploading the paper has to insert data into three tables, which are paper, coauthor and pa_subject by using multiple SQL statements when the author clicks the "Submit" button. We found some co-author's information was inserted into the table for the wrong paper id. MySQL automatically performs locking a table for record update in a table, but in this case, it involves several SQL statements used to access three tables, which is a transaction including a bunch of database operations, and should be executed together. MySQL supports the transaction by employing InnoDB table type. We can continue to use the non-transactional MyISAM table type, and change our code to prevent another user from simultaneously accessing the block of code, but if a part of transaction fails, we are unable to rollback the executed statements. Therefore, using the InnoDB is a better approach to eliminate the concurrency problem. This task will be completed as an addendum to the project. It is not reported here due to the deadline for submission of this report.

- Auto-increment attribute

  Auto-increment attribute is used for generating a unique id automatically for the id sequence. It should be used whenever it is possible. In the table paper we didn't use auto-increment paper id. In reality, some authors requested removing the older versions of their papers. We did it and had a problem when uploading papers afterwards. In our program we generated the paper id by incrementing 1 to the count number of papers. For example, if there are 50 papers, the last paper id should be 50. However, if we remove paper 45, the last paper id is still 50. But the total number of papers becomes 49. The new record should use 51 as the paper id and not 50 (49 + 1). So we should avoid using the count number of paper to generate id in the program. Instead, employ the auto-increment id in the table. Also, there is a need to provide a mechanism to the authors to replace their uploaded file with a new version. This is already been implemented and would be incorporated in the future.

- Internationalization

  Since this is a web-based application, the users come from different countries. European languages may cause some problems. The e-mail address from France and Germany is sometimes very long and has special characters. If it is not handled properly some problems appear. For example, an author cannot register because his e-mail address is too long to fit into the text field provided in the form. Here is another example. One author's e-mail address includes a special character "-" that our registration program regarded as an invalid character before. We have corrected this problem; however other such cases have to be investigated.

- Database initialization

  At the beginning of using a system, the database is initialized. The empty table should be given enough notice. For example, null pointer exceptions appear. There are two cases. First, because we know the sequence in which the events occur in the conference some information for sure will come in before the next event process happens. For example, the automatic allocation process must occur after the reviewer registration is finished. Reviewers are required to provide their organization information for the following allocation process in order to avoid conflict of interest. Before performing the paper allocation process, the PC should recheck if all the reviewers have been registered and they have filled in proper information. If not, the PC should modify their files by using the Reviewer Control function. So errors will never occur in real use.

  Second case may cause errors; such functions include set parameters for automatic allocation and set start review date, etc. If we simply delete records to empty the table from the back-end when initializing the database, the PC is unable to insert a new date or a new parameter from the user interface. In our program, we have prevented the user from leaving some fields blank by validating the data for certain fields. However, as a system administrator, who may neglect the development details when performing database initialization task, he/she can delete records from the database without using the user interface. In the future, we should provide a user interface to the PC or administrator to initialize the database to avoid causing this problem.

8. Conclusion and Future Work

8.1 Conclusion

In conclusion, ConfSys is a very useful system for the management of conferences. All of the software used in ConfSys are popular open source software and have been declared stable. We have carefully examined the architecture and the features provided by the software to make sure it can meet all our functional and non-functional requirements and possibly port it to any other operating systems.

ConfSys has implemented most of the important features required in an academic conference. ConfSys can save hundreds of hours for the PC in managing tasks. In IDEAS'03 conference, about one hundred authors and more than forty PC members have used our system. Users are satisfied with the functionality of ConfSys. Some bugs were found and have been fixed.

ConfSys has the following advantages compared to other conference management system:

- Can assure minimal development cost and good technical support since we use current popular open source software.

- ConfSys is implemented using Java Servlet. Java Servlet is written in Java, thus it possible to take full advantage of Java language, i.e. powerful, secure, operating system independent, etc.

- ConfSys is a secure system. In addition to login username and password for different users we also use http session on all web pages. Users cannot go to any page directly by typing the URL. Instead, they should log into ConfSys and follow the text link to any specific web page. ConfSys employs Secure Sockets Layer (SSL) communication protocol between client and web server to ensure confidentiality.

- ConfSys supports many file formats of the uploaded paper so that the author can use any preferred software to edit their paper.

- It has scalability and flexibility to meet different conference needs. Information of conferences such as the important dates, parameters for automatic allocation, etc for conference can be initialized using the graphic user interface provided by ConfSys.

- The automatic allocation algorithm ensures the assignments of papers based on the reviewer's expertise and review intention without violating constraints. In most cases, no further fine-tuning effort is necessary from the PC's side. The DBLP database is applied to check the conflicts of interests.

8.2 Future Work

The system can be improved in the following aspects:

- A feature is needed for authors to modify their papers' related information and re-upload their paper. Without this feature authors have to send e-mails to request the modifications.

- It needs an automatic e-mail system to remind of users to perform some tasks. For example, before the deadline of uploading papers for authors and the deadline of review for reviewers, the system can send e-mails to the users to ask them to do such work.

- The automatic allocation algorithm can detect some conflicts of interests automatically using DBLP database, such as: reviewer and author work in the same organization; they have together published a paper. But the system still needs to add one option as conflict of interests in reviewer auction because some conflicts of interests are not detected. For example, the author is an ex-student of the reviewer. Adding this option will improve the automatic allocation performance because the conflicts of interest papers are removed from the paper list for some reviewers.

- An interface needs to be implemented for the PC to view the list of papers and the status of the review information. It's very useful for the PC to monitor the review process, for example, the number of reviewers who have finished their review for a specific paper.

- Similarly, the above function also needs to be provided for the reviewers. But for the reviewer, he/she can only view the status of the assigned papers. In addition, the remaining number of reviews for the reviewer is also necessary because it can remind of the reviewer to review all assigned papers.

- Finally, an online help is needed. More clear instructions and explanations can assist users to finish their work correctly and easily. Such as salutation field on reviewer registration form is clear to our developer but may be not for some of the reviewers.

References

[1] R. Snodgrass, *Summary of Conference Management Software*, http://www.acm.org/sigs/sgb/summary.html

[2] CyberChair's homepage, http://cyberchair.cs.utwente.nl

[3] START's homepage, http://www.cs.umd.edu/~rich/START/README.html

[4] WitanWeb's homepage, http://witanweb.iit.nrc.ca/

[5] O. Nierstrasz, *Identify the Champion, An Organizational Pattern Language for Programme Committees*, In Pattern Languages of Program Design 4, vol. 4, Addison Wesley, 2000, pp. 539-556, http://iamwww.unibe.ch/~oscar/Champion/

[6] DBLP Reference Site, http://dblp.uni-trier.de/db/

[7] MySQL's home page, http://www.mysql.com

[8] Matthew Stucky, *MySQL: building user interface*, New Riders, 2002

[9] Dustin R. Callaway, *Inside Servlets, second edition*, Addison-Wesley, 2001

[10] Jason Hunter, William Crawford, *Java Servlet Programming, second edition*, O'Reilly, 2001

[11] Dan Chang, Dan Harkey, *Client/Server Data Access with Java and XML*, John Wiley & Sons, 1998

[12] Patrick Niemeyer, Jonathan Knudsen, *Learning Java*, O'Reilly, 2000

[13] Zhengwei Gu, *CONFSYS: The Cindi Conference Support System*, Concordia University, 2002