

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

**An Integrative Approach to Online Conference
Review Management**

**A THESIS
SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE IN INFORMATION SYSTEM
IN
MASSEY UNIVERSITY
By
He Meng**

**MASSEY UNIVERSITY
2003**

Acknowledgements

I would like to thank Dr. Kinshuk, my supervisor, for his sincere and constant guidance. He has been a source of knowledge and inspiration all the way long. This thesis would have been impossible without his guidance and support. I have felt lucky to have him as a supervisor.

Content

CONTENT.....	I
CHAPTER 1 INTRODUCTION.....	1
1.1 CURRENT PROBLEMS IN CONFERENCE PAPER REVIEW MANAGEMENT	1
1.2 ROLES IN AN ACADEMIC CONFERENCE.....	2
1.3 PROCEDURE OF PAPER PROCESSING IN ACADEMIC CONFERENCE.....	3
1.4 ONLINE PAPER SUBMISSION SYSTEM	5
1.5 ADVANTAGES OF ONLINE PAPER SUBMISSION SYSTEM.....	5
1.6 OBJECTIVE OF THE PROJECT	6
1.7 OVERVIEW OF THE THESIS.....	7
CHAPTER 2 RESEARCH METHODOLOGY.....	9
2.1 INTRODUCTION	9
2.2 PICTURE COMPARISON METHOD.....	9
2.3 SYSTEM SELECTION	9
2.4 FRAMEWORK DEVELOPMENT	10
2.5 SOURCE OF INFORMATION.....	13
2.6 SUMMARY.....	13
CHAPTER 3 REVIEW OF EXISTING CONFERENCE MANAGEMENT SYSTEMS.....	15
3.1 INTRODUCTION	15
3.2 CYBERCHAIR.....	15
System full name.....	15
System short name.....	15
System web site.....	15
User manual/Technical manual/System description.....	15
System demonstration available.....	15
3.2.1 <i>Functionality</i>	16
Author registration.....	16
Paper submission.....	16
Reviewer registration.....	16
Reviewer obtaining paper.....	16
Submitting review	17
Discussion	17
Assignment to reviewers	17
Paper distribution.....	18
Monitoring the review process	18
Control the review process	18
Make final decision	18
Summary of time.....	19
Summary of skills needed for each type of users.....	19

3.2.2 <i>Development</i>	20
Language	20
Database.....	20
Interface	20
3.2.3 <i>Limitations</i>	20
3.3 CONFMAN.....	20
System full name	21
System short name	21
System web site	21
User manual/Technical manual/System description	21
System demonstration available.....	21
3.3.1 <i>Functionality</i>	21
Author registration.....	21
Paper submission	21
Reviewer registration.....	22
Reviewer obtaining paper	22
Submitting review.....	22
Discussion.....	23
Assignment to reviewers.....	23
Paper distribution.....	23
Monitoring the review process.....	24
Control the review process.....	24
Make Final decision.....	24
Summary of time	25
Summary of skills needed for each type of users	25
3.3.2 <i>Development</i>	25
Language	25
Database.....	26
Interface	26
3.3.3 <i>Advantage and limitations</i>	26
3.4 EDAS	26
System full name	27
System short name	27
System web site	27
User manual/Technical manual/System description	27
System demonstration available.....	27
3.4.1 <i>Functionality</i>	27
Author registration.....	27
Paper submission	27
Reviewer registration.....	28
Reviewer obtaining paper	28
Submitting review.....	28
Discussion.....	28
Assignment to reviewers.....	28

Paper distribution.....	29
Monitoring the review process	29
Control the review process	30
Make final decision	30
Summary of time	30
Summary of skills needed for each type of users.....	31
3.4.2 <i>Development</i>	31
Language	31
Database	31
Interface.....	32
3.4.3 <i>Features and limitations</i>	32
3.5 THE ELECTRONIC SUBMISSION AND REVIEWING SYSTEM USED FOR POS8	33
System full name	33
System short name.....	33
System web site	33
User manual/Technical manual/System description.....	33
System demonstration available	33
3.5.1 <i>Functionality</i>	34
Author registration.....	34
Paper submission.....	34
Reviewer registration.....	34
Reviewer obtaining paper.....	35
Submitting review	35
Discussion	35
Assignment to reviewers	35
Paper distribution.....	36
Monitoring the review process	37
Control the review process	37
Make final decision:	37
Summary of time	37
Summary of skills needed for each type of users.....	38
3.5.2 <i>Development</i>	38
Language:.....	38
Database	39
Interface.....	39
3.5.3 <i>Features and limitations</i>	39
3.7 OTHER CONFERENCE MANAGEMENT SYSTEMS.....	40
3.8 SUMMARY.....	44
CHAPTER 4 REQUIREMENT ANALYSIS OF CONFERENCE SUBMISSION SYSTEM.....	49
4.1 OVERVIEW	49
4.2 GENERAL REQUIREMENT OF A TYPICAL CONFERENCE SUBMISSION SYSTEM	50
4.2.1 <i>System functionality</i>	50
4.2.2 <i>User characteristics</i>	52

4.3 FUNCTIONAL REQUIREMENTS OF THE SYSTEM	54
4.3.1 <i>Common functionality for all types of registered users</i>	55
4.3.1.1 Registration.....	55
4.3.1.2 Login.....	57
4.3.1.3 Logout.....	57
4.3.1.4 Forget password or user name.....	57
4.3.1.5 Modify profile.....	58
4.3.1.6 Change password	58
4.3.2 <i>Specific functionality for paper contact person</i>	58
4.3.2.1 Submit paper.....	59
4.3.2.2 View submitted paper	60
4.3.2.3 Cancel submitted paper.....	60
4.3.2.4 Download paper.....	60
4.3.3 <i>Particular functionality for reviewers</i>	61
4.3.3.1 Download paper.....	61
4.3.3.2 Make or Modify review	62
4.3.4 <i>Particular tasks for committee members</i>	63
4.3.4.1 Assign paper to reviewer.....	63
4.3.4.2 Make and cancel recommendation on the paper	67
4.3.5 <i>Particular tasks for committee chair</i>	67
4.3.5.1 Assign papers to committee members.....	69
4.3.5.2 Assign paper to reviewers over limit.....	69
4.3.5.3 Make decision on papers and notify the author.....	70
4.3.5.4 Authorise committee members and reviewers	70
4.3.5.5 Set up system parameters	71
4.4 NON-FUNCTIONAL REQUIREMENT.....	71
4.4.1 Scalability.....	71
4.4.2 Compatibility.....	72
4.4.3 Security.....	72
4.4.4 Adaptability.....	72
4.4.5 Maintainability	73
4.5 SUMMARY.....	73
CHAPTER 5 WEB PROGRAMMING TECHNIQUES AND OUR CHOICE.....	75
5.1 INTRODUCTION	75
5.2 WEB-BASED APPLICATION	75
5.3 WEB PROGRAMMING TECHNIQUES	77
5.3.1 <i>The server side programming techniques</i>	77
5.3.1.1 Common Gateway Interface (CGI).....	77
5.3.1.2 Java servlet.....	78
5.3.1.3. ASP.....	78
5.3.1.4 JSP	78
5.3.1.5 PHP	79

5.3.2 <i>The client side programming technique</i>	79
5.3.2.1 JavaScript	79
5.3.2.2 VBScript.....	79
5.3.2.3 Applet	80
5.4 SELECTION OF THE PROGRAMMING TECHNIQUES FOR THE PROJECT---JAVA SERVLET, JSP,	80
JAVABEAN, JDBC AND MYSQL	80
5.5 JAVA SOLUTION	81
Tomcat.....	81
JavaBean.....	82
JDBC	82
MySQL.....	83
Web techniques selected for the project	83
The usage of applet.....	84
JavaBean vs EJB	84
Java IDE-JBUILDER4.....	85
5.6 SUMMARY.....	85
CHAPTER 6 SYSTEM DESIGN AND IMPLEMENTATION.....	87
6.1 INTRODUCTION	87
6.2 DESIGN OF THE THREE-TIER ARCHITECTURE	87
6.3 THE USER INTERFACE DESIGN	87
6.3.1 <i>Login interface</i>	87
6.3.2 <i>Interface design for authors</i>	88
6.3.3 <i>Interface design for reviewers</i>	89
6.3.3 <i>Interface design for reviewers</i>	90
6.3.4 <i>Interface design for committee members</i>	91
6.3.5 <i>Interface design for Program Committee Chair</i>	92
6.3.5.1 Set up system parameters.....	92
6.3.5.2 User management	95
6.3.5.2 User management	96
6.3.5.3 Paper management.....	96
6.3.6 <i>Interface design for user registration and modifying profile</i>	97
6.4 THE DATA STORAGE AND DATABASE DESIGN	100
6.5 IMPLEMENTATION DETAILS OF SYSTEM ARCHITECTURE.....	107
6.5.1 <i>Communication between browser and servlets</i>	107
6.5.2 <i>Request processing in servlets</i>	108
6.5.3 <i>JSP and JavaBeans implementing the user interfaces</i>	109
6.5.3.1 Locating or creating a JavaBean from JSP page with UseBean Tag.....	109
6.5.3.2 Representing HTML form controls with data form Java Beans.....	109
6.5.4 <i>Cooperation of Java components in implementing the system function</i>	111
6.6 SUMMARY.....	120
CHAPTER 7 CONCLUSION AND FUTURE WORK	121

7.1 THE USAGE OF THE CONFERENCE SUBMISSION SYSTEM.....	121
7.2 REVIEW OF THE SYSTEM FUNCTIONALITY	136
7.3 FUTURE WORKS.....	138
REFERENCES:	139

Chapter 1 Introduction

Academic conferences play an important role in modern science and technology development. They provide venues for people involved in particular research areas to communicate with each other, share research achievement, and understand the current leading edge of development in that area. They also provide opportunity to periodically summarise the past work and find new directions for future research. The peer reviewed conference papers become reference for other researchers.

There are thousands of conferences organised all over the world every year. For each conference there are hundreds of paper submissions. The management of submission, review and selection process is a hard and tedious task for conference committees.

This thesis discusses the design and implementation of a web-based application to facilitate an efficient and effective management of paper submission, review and selection process in academic conferences.

1.1 Current problems in conference paper review management

Current conference paper review management has the problem of inefficiency and ineffectiveness. In this section we will identify these problems in terms of paper submission, paper distribution and even the paper review process.

- Long submission cycle

For most conferences nowadays people no longer submit hard copy of papers by surface mail. Taking the advantage of Internet, they begin to submit papers by email. This greatly saves time and money comparing with submission by surface mail. But the subsequent processing of papers still takes a long time due to various manual tasks involved.

After the papers are submitted, they need to be archived. To avoid name conflict each submission needs to be given a unique name for later reference. In some conferences, the paper submission is divided into two steps. First the paper information including abstract,

topics, authors details and contact information is submitted. Then, the paper content is submitted by email attachment or by FTP.

- Paper assignment: A heavy task for Program Committee (PC) chair

For each paper PC chair or PC members are expected to find several appropriate reviewers (in some conferences, papers are reviewed by PC members only). They need to know the subject areas the papers address, and the reviewers' research interests and expertise. For a conference with a lot of papers and reviewers, the assignment task can be quite iterative and time consuming.

- PC decision making meetings could be long and difficult

(Nierstrasz, 1999) identified various difficulties in program committee meetings:

- ◆ PC meetings are expensive to organize.
- ◆ It is hard to know in advance which papers will generate the most discussion.
- ◆ Without strong guidelines, a PC meeting can quickly degenerate into chaos.

1.2 Roles in an academic conference

Many people are involved in a typical conference, some of them submit papers, some review papers, and some organise the review process and make final decision on papers. According to the tasks they performed in a conference we classify them into 4 categories, *paper contact person, PC chair, PC members and reviewers*.

With these roles defined, we can elaborate the procedure of conference, and identify the conference tasks therefor fulfil the task oriented requirement analysis of the conference submission system.

- Paper contact person

Paper contact person usually is one of the co-authors of the paper. But sometimes it could be a secretary in an organisation. Paper contact persons submit papers to the conference committee and take charge of the subsequent communication with conference committee on the paper-related issues.

- PC member

Program committee members (PC members) are members of conference program committee, they are usually experts in some of the conference's topics. They help the PC chairs to organise the conference, review papers and/or assign papers to reviewers, monitor the review process and summarise the review output. They make significant recommendation on paper's acceptance and rejection. Sometimes they hold the workshops on particular conference topics.

- PC chair

Program committee chairs (PC chairs) are committee members who organise the conference, receive papers from paper contact person, assign papers to PC members and sometimes directly to reviewers.

Based on the reviews and PC members' recommendation they make final decision on paper's acceptance and rejection and notify contact person of the decision. There are usually one or two committee chairs in a conference committee. They organise the conference committee, in other words they appoint and authorise other committee members.

- Reviewer

For some conference, reviewers are usually PC members who are assigned with papers for reviewing. Nowadays, besides PC members, most conferences let independent technical people to review the papers and make feed back to program committee. Reviewers are experts in some of the conference subjects and volunteer to review papers for the conference.

1.3 Procedure of paper processing in academic conference

The paper processing in an academic conference can be divided into six steps (refer to figure 1-1), *call for papers, paper submission, paper assignment, review submission, paper selection, notify authors and publish selected papers*. Although these steps could be

overlapped, the program committee needs to set up the timelines for finishing each step especially *for paper submission, assignment and review submission*.

1. Call for papers

Conference committee specifies the theme and topics of the conference, set up timelines and schedule, notify the potential authors of the conference via public media like newspaper and academic journals.

2. Paper submission

Authors submit their papers to the conference along with the contact person's contact information and all the co-authors' name and affiliation. They need to specify the topics the paper is about and the category as which they want the paper is accepted by the conference.

3. Paper assignment

The papers are assigned to appropriate PC members and reviewers. Their expertise and authority in the paper's topics decide the appropriateness of PC members and reviewers.

4. Review submission

The PC members and reviewers review the paper and submit their comments and recommendation to the PC chair.

5. Paper selection

PC chairs make final decision on whether accept or reject the paper.

6. Notify authors and publish selected papers

The PC chairs notify the author (contact person) of the acceptance or rejection of the paper. The accepted papers are to be presented in the conference and published as conference's paper selection.

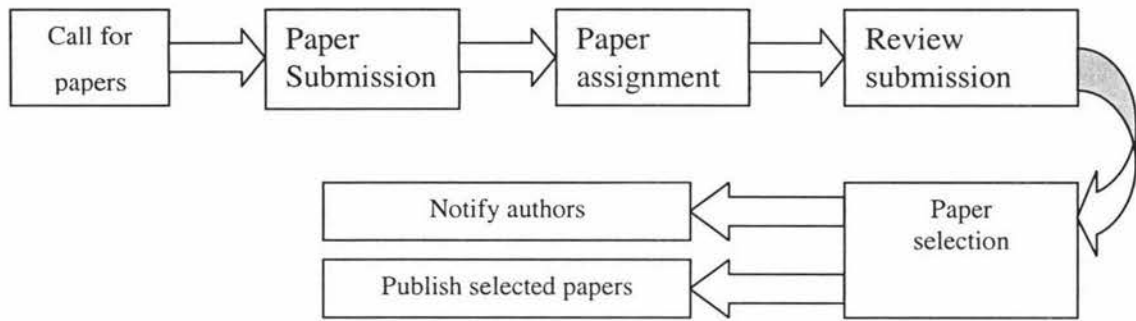


Figure 1-1 Procedure of paper processing

1.4 Online paper submission system

With the popularisation of World Wide Web, people begin to develop online paper submission systems to support the paper processing procedure mentioned above. Specifically, online paper submission systems are supposed to facilitate the conference tasks in the following aspects (Snodgrass, 1999):

- Submission of papers ;
- Assignment to reviewers;
- Mailings to PC members ;
- Submissions of reviews ;
- Reviewer comment threads;
- Distributed PC meeting;
- Letters to authors;
- Conference registration;
- Handles associated workshops; and
- Handles different tracks of submissions.

1.5 Advantages of online paper submission system

Online paper submission systems provide many advantages over traditional methods. Some of them are listed below.

1. Authors can submit paper information and abstract by filling HTML forms, upload papers content via Internet. This greatly reduces the time for submission. With electronic copy of paper information and paper content stored in the system, the subsequent paper processing like distribution and publishing becomes much easier.
2. Reviewers can submit reviews using HTML forms, no hard-copy forms are needed.
3. With reviews and recommendations stored in the system, it's very easy for PC chairs and PC members to track and monitor the review process. Actually there's no "collecting review" phase, because the moment the reviewer makes the review it is "collected" in the system.
4. The system provides a common workspace for every user in the conference, they store the information in the system, retrieve the information and update it. Taking advantage of computer and web technique, some useful features can be implemented, like
 - automatic email notification and confirmation
 - automatically sorting the reviewer list for a paper according to their keyword matching
 - automatically detecting the assignment conflict e.g.
 - ◆ a reviewer is assigned too many papers
 - ◆ a paper is assigned to one of it's authors to review
 - automatically limiting submissions against deadline
 - PC chairs and PC members can work at their own place for paper processing rather than having to gather at the conference location
 - PC chairs can configure the system remotely

1.6 Objective of the project

After a careful study of existing conference systems and their weakness, we aimed to take advantage of database and web technology, and design and implement an online paper submission system, which should be able to:

- shorten the paper submission cycle by filling online form and file uploading;
- facilitate the paper assignment by providing paper-reviewer keywords matching and automatically conflict of interest checking;
- enable the PC chair and PC member to monitor the review process by providing various paper searching methods;
- support “double blind” review;
- provide each user with a personalised interface and workspace;
- secure paper and user information access by password and privilege checking;
- be configured easily to meet different conferences’ requirement; and
- be installed and run on different OS platform.

Overall, the system should reduce the manual work to be done by every user, simplify the decision making, provide friendly user interface for various users, apply security mechanism for accessing papers and user profiles information, and be flexible to be applied to various conferences.

1.7 Overview of the thesis

Chapter 1 introduces roles and their tasks in an academic conference, describes the paper process, and discusses the advantage of online paper submission system and the objective of this project.

Chapter 2 describes the methodology adopted in the research and defines the framework for existing system evaluation.

Chapter 3 evaluates the existing online conference submission systems *CyberChair*, *ConfMan*, *EDAS* and *POS8* in terms of function, development and reviews some other systems in term of features and limitations.

Chapter 4 analyses the functional and non-functional requirement of the typical conference submission system.

Chapter 5 first introduces in general the popular web techniques including CGI, Java servlet, JSP, ASP, PHP, Java script and VB script and applet, which can be used to develop such systems. Then it discusses in detail the Java solution in web-base application development besides the rationale for its selection. Both server side and client side solutions are discussed, including techniques like Java servlet, JSP, Applet, JavaBean, and Java database package JDBC.

Chapter 6 describes the design and implementation of the web-based online submission system using Java servlet, JSP, JavaBean and JDBC. The overall system architecture design and database structure is narrated.

Chapter 7 summarises the whole thesis and discusses the future extension possibilities.

Chapter 2 Research Methodology

2.1 Introduction

As defined in Chapter 1 the object of the thesis is to design and implement a web based computer application, so currently existing applications should be evaluated in terms of advantages and limitations. This chapter will introduce the methodology we will use for the evaluation.

2.2 Picture comparison method

(Sharland, 1991) described a picture comparison method in evaluating a software package. The picture comparison method uses a set of blank templates which are filled in for each package under consideration. By forcing the information into a common format, the final comparison is somewhat more straightforward than when using the detailed evaluation technique.

Picture comparison method has three stages:

1. The first stage is to determine the requirements of the system-data and functionality supported
2. The second stage is the trawling and dissemination of information concerning potential packages.
3. The third stage is to evaluate, by means of comparison, package against packages and packages against requirements

We will adopt the picture comparison method to evaluate the existing conference management systems to identify the limitation, and our own system for comparison. Next we will select some existing systems and develop a framework (the template) for the evaluation.

2.3 System selection

There are very few articles and works on the evaluation of conference management system. (Snodgrass, 1999) uses the picture comparison method investigated a list of conference management systems. It concluded in a table for each system whether it provides a particular conference management function or not.

Snodgrass's survey gives out

- a list of conference management systems and
- a list of preliminary functionality requirements for a conference management system

We try to select some typical conference management systems from the list that

- support the most functional requirement of conference review management
- provide the user/technical manual or detailed functional description
- and preferably, provide demonstration of system functions

Meeting these criteria, Cyberchair, Confman, EDAS and POS8 are selected for the evaluation.

To cover more systems and features, CMT, AERA, CHI, START, AAI, and WIMPE for which incomplete functional description is available are also selected for feature and limitation investigation.

2.4 Framework development

Unlike the survey done by (Snodgrass, 1999), for each system under evaluation instead of whether a particular function is supported we focus on how well the function is supported. We will try to answer questions like

- How many (types of) users are involved in this function
- How many steps it takes for each user
- How long it is estimated to take
- Are there any special utilities implemented to make the function easier to complete

Table 2-1 lists the functions to analyse in the framework, for each function the items listed in table 2-2 should be filled if appropriate. Particularly for the Paper Assignment

function, how the reviewer's preference, expertise and conflict of interests are taken into account should be specified.

System full name	The complete name of the system
System short name	A short name used in the evaluation
User/technical manual	Listed as a web ULRs
System web site	Listed as URLs
Demonstration copy available?	Yes or No, if Yes provide URL
System functions	Users, Steps, Estimated time, Skill, Utilities
Author registration	
Paper submission	
Reviewer registration	
Reviewer obtaining paper	
Submitting review	
Discussion	
Register reviewer	
Paper assignment	
Reviewer preference	
Reviewer expertise	
Conflict of interests	
Paper distribution	
Monitor/control review process	
Make final decision	

Table 2-1 Definition of the framework

User involved	Type(s) of users involved in the function
Steps	Number of steps for the user to complete this function
Estimated time	The estimated time in minute for a user to perform the function, the unit of the item to deal with should be specified, for example for one paper, for one reviewer, for all papers, all reviewers, etc.
Knowledge/skill needed	Computer knowledge and skills required for the user to complete the function, like operating the HTML controls (HTML controls include HTML forms, buttons, check buttons, optional buttons, etc), command line interface (run UNIX scripts)
Utilities provided	Particular online or command line tools provided to facilitate the user's operation

Table 2-2 Items for each function in the framework

2.5 Source of information

(Sharland, 1991) listed the typical sources for gathering information for software package evaluation:

- Vendor's product development manager and/or technical support manager and /or training staff
- User group
- Books on the product
- User manual
- Technical manual
- Package videos, training films

Vendor's staff should have good knowledge of their products but their comment on the performance of the system may not be fair enough for assessment. The most reliable and persuasive information source for this project should be "user group". But limited by time we are not able to gather information from the users of each of the systems. So we choose to gather information mostly from user/technical manual and system function demonstration for the research.

2.6 Summary

This chapter illustrated the methodology we are using to carry out the research, with this methodology a framework is designed to evaluate the conference management system. Based on Snodgrass's work, some existing systems are selected for evaluation and review. In next chapter we will evaluate the selected systems with the framework and review some other systems in terms of features and limitations.

Chapter 3 Review of existing conference management systems

3.1 Introduction

There are various conference management systems available in the market to facilitate the management of various conferences tasks. Some of them, like CyberChair, ConfMan, EDAS and POS8 support every major phase of paper review process, while others, like CMT, AERA Electronic Proposal Submission System, START, AAAI conference management software and WIMPE, support part of the review process. In this chapter we shall evaluate CyberChair, ConfMan, EDAS and POS8 with the framework defined in the previous chapter in terms of functionality, development, features and limitations. Other systems' features and limitations will also be reviewed.

3.2 CyberChair

CyberChair is a web-based group-ware application that supports the review process for technical contributions to conferences. It was used for ECOOP 2000 Conference (Stadt, 2000).

System full name

CyberChair

System short name

CyberChair

System web site

<http://www.cyberchair.org>

User manual/Technical manual/System description

System description: <http://www.cyberchair.org>

System demonstration available

Yes

<http://www.cyberchair.org>

3.2.1 Functionality

Author registration

Authors (contact persons) register at phase 1 of submitting paper.

Paper submission

Users involved: author (contact person) and system maintainer

Steps: 3

- Submit paper abstract and contact information
- Receive login and password via email
- Login and upload paper via web browser

Estimated time: 40 min for one paper

Knowledge/Skill needed: Filling online form, uploading file in web browser

Reviewer registration

Users involved: system maintainer

Steps: 2

- System maintainer set up password protected personal web page for each reviewer
- Notify reviewer of the login and password (detail not available)

Estimated time: 20 min for one reviewer

Knowledge/Skill needed: HTML programming

Reviewer obtaining paper

Users involved: reviewer

Steps: 1

- Select paper and download to local drive

Estimated time: 5 min for one paper

Knowledge/Skill needed: downloading file in web browser

Submitting review

Users involved: reviewer

Steps: 1

- Fill review form and submit it

Estimated time: 20 min for one review

Knowledge/Skill needed: Filling on line form

Discussion

Utilities provided:

- Read other reviewers' reviews after submitting his/her own
- Send email to Committee chair and other reviewers of the paper (discuss by email)

Assignment to reviewers

User involved: Committee Chair

Steps: 2

- On paper assignment page add paper Ids to reviewer's papers field and submit the form
- Send emails to all reviewers notifying the papers they are assigned to

Estimated time: 5 min for one paper

Reviewers' preference: incorporated

Reviewers' expertise: incorporated

Conflict of interests

Author should specify if he/she is a reviewer

Utilities provided:

- "Bidding for papers" allows reviewers to express their preference on the papers
- A paper distribution proposal can be generated for Committee Chair taking into account of reviewers' preference on papers, willingness and expertise on topics

Paper distribution

User involved: maintainer

Steps: 3

- Create a directory on the web server for the reviewer
- Copy the files of papers assigned to the reviewer to that directory
- Create a link to each of the files in the reviewer's personal web page

Estimated time: 30 min for one reviewer

Knowledge/Skill needed: file directory management and HTML programming

Monitoring the review process

User involved: Committee Chair

Utilities provided:

- Four kinds of overviews of reviews are generated every hour for Committee Chair to monitor the review process
 - ◆ Overview of all reviews
 - ◆ Overview of categories
 - ◆ Overview of champions of papers ("champion" means that a reviewer claims in the review that he/she will support or oppose to accept the paper in the PC meeting)
 - ◆ Low expertise reviews for papers that have only been reviewed by reviewers who indicated that they are no expert on the topics the paper covers
- A coloring scheme is adopted to indicate the level of agreement

Control the review process

User involved: Committee Chair

Utilities provided: Not mentioned

Make final decision

User involved: system maintainer

Steps: 1

- Send final decisions on papers to contact persons

Estimated time: 2 min for all papers

Utilities provided:

- Overview of champions of papers

Summary of time

Time used by each user in a typical conference with CyberChair is summarised in table 3-1, unit of time is minute, all time used by system maintainer is considered as by Committee Chair.

Assume that for a typical conference there are 200 papers and 100 reviewers, each papers need 3 reviews, thus 600 reviews are to be made and each reviewer makes 6 reviews.

	Committee Chair	Reviewer	Author
Author registration	0	0	0
Paper submission	0	0	40
Reviewer registration	2000	0	0
Obtaining paper	0	30	0
Submitting review	0	120	0
Assignment to reviewers	1000	0	0
Paper distribution	3000	0	0
Make final decision	2	0	0
Total	6002	150	40

Table 3-1 CyberChair user time summary

Summary of skills needed for each type of users

To perform their tasks with CyberChair

Authors need the skills of:

- Filling online form, uploading file in web browser

Reviewers need following computer skills:

- Filling online form, downloading paper in web browser

Committee Chairs need the skills of:

- HTML programming, Unix file management

3.2.2 Development

Language

CGI scripts developed in the programming language of Python.

Some Unix scripts developed for use by the maintainer to perform administrative tasks from local machine.

Database

No database is adopted. All the information is stored in flat files.

Interface

HTML pages embedded in CGI scripts for on line users

Unix command line tools for maintainer

3.2.3 Limitations

There is no uniform user registration and management. Author's information is registered with papers. PC member information is entered by PC chair.

Maintainer generates reviewers' web pages. Actually the maintainer sets up a directory for each reviewer and copies the assigned papers to this directory. Maintainer also does author notification.

From these we can see that some of the PC chair's task were transferred to maintainer. The maintainer performs the tasks manually or with some pre-built UNIX scripts, this could also be tedious and error-prone. Since the system does not apply database the data inconsistency and redundancy seem inevitable.

3.3 ConfMan

ConfMan (Conference Manager) is web-based management software for conference organization. It includes features for paper registration, an online meeting for program committee members, registration of conference participators and some administration tools

for the conference organization committee. ConfMan stores all information from these pages into a MySQL database (Preuss et al, 1999).

System full name

Conference Manager

System short name

ConfMan

System web site

<http://confman.unik.no>

User manual/Technical manual/System description

User manuals:

<http://www.ifi.uio.no/confman/MANUAL/use.html>

<http://www.ifi.uio.no/confman/MANUAL/functionality.html>

System demonstration available

Yes.

<http://www.ifi.uio.no/confman/demo/>

3.3.1 Functionality

Author registration

Authors (contact persons) register along with first step of paper submission.

Paper submission

Users involved: author (contact person), Committee Chair

For contact person

Steps: 2

- Submit paper abstract and contact information
- Upload paper via web browser (No login required, the paper abstract and contact information displayed on web page as well as emailed to contact person.)

Estimated time: 25 min for one paper

Knowledge/Skill needed: filling online form, uploading file in web browser

For Committee Chair

Steps: 1

- Check if the paper printable and send confirmation email to contact person

Estimated time: 3 min

Knowledge/Skill needed: operating HTML controls

Reviewer registration

Users involved: Committee Chair

Steps: 3

- Fill reviewer's information into online form and submit the form
- Select login name and password for each reviewer
- Notify reviewers of their login names and passwords in group emails

Estimated time: 20 min for one reviewer registration, 2 min for notification of all reviewers

Knowledge/Skill needed: fill and submit online forms, execute UNIX scripts in command line interface.

Reviewer obtaining paper

Users involved: reviewer

Steps: 1

- Login in, select paper and download to local drive

Estimated time: 5 min for one paper

Knowledge/Skill needed: downloading file in web browser

Submitting review

Users involved: reviewer

Steps: 1

- Fill up review form and submit

Estimated time: 20 min for one review

Knowledge/Skill needed: Filling on line form

Discussion

Utilities provided:

- Read other reviewers' reviews after submitting his/her own
- Two rounds of ballots supported

Assignment to reviewers

User involved: Committee Chair

Steps: 2

- For each paper, select the paper, input reviewers Ids and click the "Save change button to save the assignment.
- After all papers have been assigned, use the "Send list of assigned papers to reviewers" function to notify reviewers of the assigned papers along with the login name and password.

Estimated time: 3 min for one paper, 2 min for notification of all reviewers

Reviewers' preference:

After all papers are submitted, Committee Chair sends list of all submitted papers to reviewers by email so that the reviewers can give preference on which papers they would like to review (no implementation details).

Reviewers' expertise: considered by Committee Chair personally

Conflict of interests: considered by Committee Chair personally

Utilities provided:

- Send list of all submitted papers to reviewers
- Send list of assigned papers to reviewers
- Disable a reviewer from voting for a paper

Knowledge/Skill needed: HTML controls operation

Paper distribution

User involved: Committee Chair

Steps: 2

- Run "htpasswd" command to generate a password file which allows the authorized reviewers to access papers by logging in.
- Generate list of papers to update the URL on the web page for reviewer to download the paper.

Utilities provided:

- On line function “Generate list of papers

Estimated time: 2 min for one reviewer

Knowledge/Skill needed: HTML control operation and UNIX command line operation

Monitoring the review process

User involved: Committee Chair

Utilities provided:

- See which papers have been uploaded
- See papers by paper types
- See list of assigned papers to reviewers
- See list of all reviews
- See list of papers regarding vote, average rating, etc

Control the review process

User involved: Committee Chair

Steps: 1

- Select phase number and submit the form

Utilities provided:

- Change phase from phase 1(review) to phase 2(ballot), the average rating is calculated for each paper

Make Final decision

Users involved: Committee Chair

Steps: 2

- For each paper set status to accepted or rejected regarding number of votes and average rating
- When all the decisions about the final status of the papers are finished, send an accept/reject notification to contact authors

Estimated time: 3 min for setting one paper, 2min for notification of all final decisions

Utilities provided:

- See list of papers (with number of votes and average rating)
- Send an accept/reject notification to contact authors

Summary of time

Time used by each user in a typical conference with ConfMan is summarised in table 3-2, unit of time is minute.

Assume that for a typical conference there are 200 papers and 100 reviewers, each papers need 3 reviews, thus 600 reviews are to be made and each reviewer makes 6 reviews.

	Committee Chair	Reviewer	Author
Author registration	0	0	0
Paper submission	0	0	25
Reviewer registration	2002	0	0
Obtaining paper	0	30	0
Submitting review	0	120	0
Assignment to reviewers	600	0	0
Paper distribution	200	0	0
Make final decision	600	0	0
Total	3402	150	25

Table 3-2 ConfMan user time summary

Summary of skills needed for each type of users

To perform their tasks with ConfMan

Authors need the skills of:

- Filling online form, uploading file in web browser

Reviewers need following computer skills:

- Filling online form, downloading paper in web browser

Committee Chairs need the skills of:

- Running scripts in Unix command line interface

3.3.2 Development

Language

CGI scripts Written in PERL.

Database

MSQL database server

Interface

HTML pages

3.3.3 Advantage and limitations

ConfMan provides good support in paper selection. It enables program committee members (reviewers) to review, discuss, and vote for papers, it summarises the review and ballot by automatically calculating the average rating and number of votes.

One limitation is that ConfMan provides limited support in paper assignment. The reviewers' preference and expertise are considered only by PC chair personally, the software doesn't provide any suggestion regarding them, to choose the right persons for a paper from a large number of reviewers maybe sometimes difficult for PC chairs.

Another limitation is that the paper search method is not good enough, with large number of papers the program chair or program committee members may require to list papers which meet some criteria, e.g. completely reviewed, accepted, etc.

The same problem as in CyberChair is that the system does not provide uniform user registration. For each reviewer PC chair needs to insert a record and set up login and password (Preuss *et al*, 1999). This is quite inefficient, especially when there are a lot of reviewers.

3.4 EDAS

EDAS is web-based software that supports editors, reviewers and technical program committee members in managing paper submission, review and acknowledgement for workshops, conferences and journals. Papers are assigned and reviewed through web pages. The program is currently designed primarily to handle conferences, with one or more levels of review (Schulzrinne, H. 2001).

System full name

EDAS (Editor's Assistant)-Conference Paper Processing Software

System short name

EDAS

System web site

<http://www.cs.columbia.edu/~hgs/edas>

User manual/Technical manual/System description

System description: <http://www.cs.columbia.edu/~hgs/edas/edas.html>

Technical manual: <http://www.cs.columbia.edu/~hgs/edas/doc/internals.html>

System demonstration available

No

3.4.1 Functionality**Author registration**

Authors (contact persons) register along with paper submission.

Paper submission

Users involved: author (contact person)

Steps: 3

- Submit paper abstract and contact information
- Receive a paper identifier via email
- Name the file with the paper identifier and send it as email attachment or transfer the paper file to the ftp server

Estimated time: 30 min

Knowledge/Skill needed: Filling online form, transfer files to ftp server

Utilities provided:

- Software can check the file format, then compress it and store in a designated local directory that is visible to the web server.

Reviewer registration

Users involved: Committee Chair

Steps: 1

- Create a list of reviewers by entering their name, affiliation, and email address into a web form.

Estimated time: 10 min for one reviewer

Knowledge/Skill needed: filling online form

Reviewer obtaining paper

Users involved: reviewer

This function is not mentioned in the manuals. Assume that the URLs the reviewer gets contain links to the files that can be opened on line

Steps: 2

- Open the paper file on line
- View paper online or print out the content

Estimated time: 5 min for one paper

Knowledge/Skill needed: Web browsing

Submitting review

Users involved: reviewer

Steps: 1

- Fill up review form and submit it

Estimated time: 20 min for one review

Knowledge/Skill needed: Filling on line form

Discussion

Utilities provided: None

Assignment to reviewers

User involved: Committee Chair or Committee Members

Steps: 2

- Select a paper from a list sorted by paper numbers by clicking on the title
- From the paper's display, assign TPC member (reviewer) to the paper from a pull-down list

- Send emails to all, the email message contains links to papers allows the reviewers to access the assigned papers

Estimated time: 3 min for one paper, 2 min for notification of all reviewers

Reviewers' preference: incorporated

Reviewers' expertise: considered by Committee Chair personally

Conflict of interests: considered by Committee Chair personally

Utilities provided:

- Send emails to all
- Reviewers may indicate a range of topics, ordered by preference, which are then also used by authors to classify their papers, assignment algorithm then attempts to match all papers to appropriate reviewers by working through the list of reviewers

Paper distribution

As long as the paper assignment finished, the paper distribution is finished

User involved: none

Steps: 0

Estimated time: 0

Utilities provided:

- System dynamically generates web page for each reviewer contains links to the paper he/she is assigned to.

Monitoring the review process

User involved: Committee Chair

Utilities provided:

- List papers sorted by paper number, title, topic area, summary score, status
- List papers filtered with status (active, withdrawn, rejected, accepted)
- Customise the listing to show only a subset of paper fields
- Color-coded paper listing
 - ◆ review complete-green
 - ◆ waiting to be reviewed-yellow
 - ◆ less than five days to review deadline-orange
 - ◆ overdue review-red

Control the review process

User involved: Committee Chair

Utilities provided: none

Make final decision

Users: Committee Chair

Steps: 3

- mark all paper as rejected first
 - then mark the accepted paper through a menu on the full-paper view page
- once all the decisions have been made, an email note is send to the author of the final decision

Estimated time: 30 min for all papers

Utilities provided:

- View papers sorted by summary score, as well as individual reviews

Summary of time

Time used by each user in a typical conference with EDAS is summarised in table 3-3, unit of time is minute.

Assume that for a typical conference there are 200 papers and 100 reviewers, each papers need 3 reviews, thus 600 reviews are to be made and each reviewer makes 6 reviews.

	Committee Chair	Reviewer	Author
Author registration	0	0	0
Paper submission	0	0	30
Reviewer registration	1000	0	0
Obtaining paper	0	30	0
Submitting review	0	120	0
Assignment to reviewers	602	0	0
Paper distribution	0	0	0
Make final decision	30	0	0
Total	1632	150	30

Table 3-3 EDAS user time summary

Summary of skills needed for each type of users

To perform their tasks with EDAS

Authors need the skills of:

- Filling online form, transferring file by email attachment or via FTP utility

Reviewers need following computer skills:

- Filling online form

Committee Chairs need the skills of:

- Running scripts in Unix command line interface

3.4.2 Development

Language

CGI scripts

TCL

Database

Gdbm hashed database engine

Interface

HTML pages embedded in CGI scripts for online users

3.4.3 Features and limitations

Features

Unlike the other systems discussed above, the assignment phase defined in EDAS does not contain “bidding for papers”. TPC members and reviewers indicate their expertise as a range of topics ordered by preference, which are then also used by authors to classify their papers. Therefore the assignment algorithm can be developed to match the papers to appropriate reviewers. There is no detail about the automatic assignment function for Committee Chair.

System dynamically generating web pages for each reviewer that contains links to the assigned paper, this avert Committee Chair’s effort to operate on the files and maintain reviewer’s personal web page which takes a lot of Committee Chair’s time in other systems.

EDAS supports more than one level review, each paper is assigned to one TPC member, who, in turn find 3 to 4 reviewers to review it.

Papers are submitted via email or ftp. To copy the file to appropriate directory and associate it with the paper’s administrative info, utilities are developed to capture the email and scan ftp directory for the file. Even more sophisticated, the utility can check the file’s format and even change it.

Only one record is kept for a user in the system. This avoids repetitive registration and data inconsistency.

Review forms can be customised. Review questions are edited as variables in the conference configuration file.

Templates of email messages for notification and confirmation to authors, reviewers, TPC members are edited as text files.

Limitations

Papers are not submitted by uploading. There's no uniform user registration and management. TPC chair enters record for TPC members and TPC member enters records for reviewers. The customisation cannot be done online. The reviewers' expertise and conflict of interests in assignment have to be considered by Committee Chair personally.

3.5 The electronic submission and reviewing system used for POS8

An Electronic Conference Submission System (Kirby, 1998) was designed and used for *Eighth International Workshop on Persistent Object Systems: Design, Implementation and Use (POS8)*.

System full name

Electronic Conference Submission Systems for Eighth International Workshop on Persistent Object Systems: Design, Implementation and Use

System short name

POS8

System web site

<http://www-ppg.dcs.st-and.ac.uk/Conferences/POS8/>

User manual/Technical manual/System description

Technical manual:

<http://www-ppg.dcs.st-and.ac.uk/Conferences/admin-system/Readme.html>

System demonstration available

Submission forms

Paper abstract submission form:

<http://www-ppg.dcs.st-and.ac.uk/Conferences/admin-system/online/web-pages-public/abstract-form.html>

Paper submission form:

<http://www-ppg.dcs.st-and.ac.uk/Conferences/admin-system/online/web-pages-public/submission.html>

3.5.1 Functionality

*Note that in this section all steps marked with * at the end of line are carried out by running a PERL script tool in UNIX command line interface.*

Author registration

Register contact person's email along with the paper abstract submission

Paper submission

Users involved: author (contact person) and Committee Chair

For contact person

Steps: 2

- Submit paper abstract along with contact person's email
- Submit the final paper (file) as email attachment

Estimated Time: 25 min

For Committee Chair

Steps: 3

- Rename received file regarding the submission Id
- Copy the file to a particular directory
- Send an acknowledgment of the successful receipt of the full paper*

Estimated time: 10 min for one paper

Knowledge/Skill needed: UNIX file management

Reviewer registration

Users involved: Committee Chair

Steps: 8

- Generate reviewers' password*
- Insert reviewers' record in a reviewer file
- Generate HTML directories for reviewers (all in one command)*
- Create the encrypted password file by running htpasswd command once for each reviewer
- Generate .htaccess file in the directory for each reviewer*
- Generate the pages listing the papers in each category*
- Generate the page listing all papers*

- Mail instructions and passwords to reviewers*

Estimated time: 30 min for each reviewer

Knowledge/Skill needed: edit text file and run UNIX scripts in command line interface

Reviewer obtaining paper

Users involved: reviewer

Steps: 1

- Download file in web browser

Estimated time: 5 min for one paper

Knowledge/Skill needed: download paper in web browser

Submitting review

Users involved: reviewer and Committee Chair

For reviewer:

Steps: 1

- Fill in on line review form and submit

Estimated time: 20 min for one review

Knowledge/Skill needed: fill on line form

For Committee Chair

Steps: 1

- Periodically transfer submitted reviews to the ACCEESS database. This could be done only once for all reviews, but in reality it could be done everyday.

Estimated time: 5 min for all reviews

Knowledge/Skill needed: import ASCII files to ACCESS database table

Discussion

Utilities provided: None

Assignment to reviewers

User involved: Committee Chair

Steps: 2

- Insert a record for each assignment (paper-reviewer) in ACCESS table as well as an ASCII file
- Mail instructions on retrieving allocated papers to the reviewers*

Estimated time: 5 min for each assignment (review), 2 min for notification of all reviewers

Knowledge/Skill needed: Import ASCII files to ACCESS database table, run PERL scripts in UNIX command line interface

Reviewers' preference:

User involved: Committee Chair and Reviewer

For reviewer

Steps: 2

- Select paper they prefer to review
- Submit the preference on line

Estimated time: 10 min for one reviewer

Knowledge/Skill needed: fill and submit online form

For Committee Chair

Steps: 2

- Periodically transfer submitted review preference to the database*
- Generate ACCESS report of Review preference*

Estimated time: 10 min for the whole conference

Knowledge/Skill needed: Import ASCII files to ACCESS database table, generate report from ACCESS database

Reviewers' expertise: considered by Committee Chair personally

Conflict of interests: considered by Committee Chair personally

Utilities provided:

Paper distribution

User involved: Committee Chair

Steps: 3

- Generate a page for each reviewer containing links to the papers allocated to them*
- Generate symbolic links in each reviewer's directory to their allocated papers*
- Generate a page for each reviewer allowing reviews to be entered*

Estimated time: 5 min for one reviewer

Knowledge/Skill needed: run PERL scripts in UNIX command line interface

Monitoring the review process

User involved: Committee Chair

Utilities provided:

- Following reports can be generated from the ACCESS database
- Review-Preference report
- Reviewer-paper-assignment report
- Paper-reviewer-report
- All –Papers report lists the review marks for each paper
- Ratings-Summary-Paper-Order summarises the review marks for each paper sorted by paper No.
- Ratings-Summary-Rating-Order summarises the review marks for each paper sorted by marks
- PC average lists average marks given by each PC member (reviewer)
- Ranking-numbers lists the overall ranking of each paper

Control the review process

User involved: Committee Chair

Utilities provided: none

Make final decision:

User involved: Committee Chair

Steps: 2

- Inserted Ids of accepted papers into a file
- Mail acceptance/rejection notifications to authors*

Estimated time: 2 min for one paper, 2 min for notification of all papers

Utilities provided:

- The reports generated from ACCESS database

Summary of time

Time used by each user in a typical conference with POS8 is summarised in table 3-4, unit of time is minute.

Assume that for a typical conference there are 200 papers and 100 reviewers, each papers need 3 reviews, thus 600 reviews are to be made and each reviewer makes 6 reviews.

	Committee Chair	Reviewer	Author
Author registration	0	0	0
Paper submission	0	0	25
Reviewer registration	3000	0	0
Submit preference	10	10	
Obtaining paper	0	30	0
Submitting review	5	120	0
Assignment to reviewers	3002	0	0
Paper distribution	500	0	0
Make final decision	402	0	0
Total	6919	160	25

Table 3-4 POS8 user time summary

Summary of skills needed for each type of users

To perform their tasks with POS8

Authors need the skills of:

- Filling online form, transferring file by email attachment

Reviewers need following computer skills:

- Filling online form

Committee Chairs need the skills of:

- Running tools written in PERL scripts in Unix command line interface
- Edit text files
- Import data from ASCII file into ACCESS database
- Generate reports from ACCESS tables

3.5.2 Development

Language:

PERL scripts

Database

Microsoft ACCESS

Interface

Online HTML forms for authors and reviewers

UNIX command line for Committee Chair

3.5.3 Features and limitations

This system provides most of the functionality for paper submission and review. But it is not a real web based system. Only the paper abstract submission and review submission is through the web. All other functions are supported as PERL scripts edited and executed by PC chair directly in UNIX command line environment. The PC chair even needs to edit and compile some C program. Microsoft ACCESS database is used, but not as a client-server structure's back end. Instead, the PC chair manually imported data originally stored in text file in order to generate ACCESS reports used by conference committee.

3.6 Evaluation of the four systems

From the above analysis of the four existing systems we can get some conclusion:

- The online paper submission functions provided by the existing systems are fairly good for authors to submit their papers. Basically a paper can be submitted in 30 minutes. This time is similar for different submitting methods, e.g. by email attachment, by online uploading or by FTP.
- The online review submission functions supported by the existing systems are efficient. A reviewer can obtain a paper in about 5 minutes, and submit a review in 20 minutes.
- Reviewer registration, paper assignment and paper distribution takes most of the time of the Committee Chair. For a clear overview we summarise the Committee Chair's time in the four systems in table 3-5. We can see EDAS gives the best performance in term of Committee Chair's time. This is because:
 - ◆ With EDAS Committee Chair does not need to select login and password for each reviewer

- ◆ EDAS System dynamically generate web page for each reviewer that contains links to assigned papers

	CyberChair	ConfMan	EDAS	POS8
Author registration	0	0	0	0
Paper submission	0	0	0	0
Reviewer registration	2000	2002	1000	3000
Obtaining paper	0	0	0	10
Submitting review	0	0	0	5
Paper assignment	1000	600	602	3002
Paper distribution	3000	200	0	500
Make final decision	2	600	30	402
Total	6002	3402	1632	6919

Table 3-5 Committee Chair time summary

- POS8 got the worst performance since it provides a lot of single command line tools which do not work as an integrated system, also it requires the Committee Chair to be not only familiar with UNIX command line interface but also good at ACCESS database operation.

So in our system design we to need consider how to avoid the inefficiency in review registration, paper assignment and paper distribution. Also we should try to provide all user with online web browser interface which requires the least IT skills of the users.

3.7 Other Conference Management systems

There are some other systems or software modules that partially support the functionality in conference paper review process or only part of their functionality can be identified from the description on their web sites.

Microsoft's Conference Management Tool

The Conference Management Tool (CMT) is designed to be a complete solution for the submission and review process of conference papers. It is designed to be flexible enough to support a wide range of conference types, yet easy enough to use so that the program chair does not need to write any code. All functions are handled through the web browser, even setup and configuration (Microsoft Research, 2001a). The detailed description of CMT's function is not available.

Microsoft claims some features of its Conference Management Tool (Microsoft Research, 2001a):

- 100% web based interface
- Supports papers submitted in electronic format, hard copy format or both
- Provides confirmation screens and automatic emails throughout the submission and review process
- Pre-built email messages are easily customized
- Paper allocation is handled by an intuitive, market-based system
- Both criteria and display of the review form is customized by the PC chair with the browser
- Prebuilt sorting and aggregation tools to make paper selection efficient even with large numbers of reviewers

Among these features, "Paper allocation is handled by an intuitive, market-based system and Pre-built sorting and aggregation tools to make paper selection efficient" might be unique for this system. However, the details of these features are not available. Another good point is that "both criteria and display of the review form is customized by the program chair with the browser". ConfMan only allows program chair to customise the text in HTML page and CGI scripts, but to customise the criteria of review form will involve adding/deleting controls in HTML pages and probably changing the structure of database tables.

The major limitation of this system is the platforms it requires. Unlike Apache and MSyQL, the IIS server and SQL server are not free software packages.

Another limitation is similar to that of Cyberchair and ConfMan---user registration, PC chair needs to add each member's email to the system and set up login and password for him/her.

AERA Electronic Proposal Submission System

AERA Electronic Proposal Submission System is used for proposal submission and process in annual meeting of *The American Educational Research Association*. It is claimed to have some good features (AERA, 2000):

- The system provides all the functionality on-line: proposal submission, user registration (reviewer and PC chair), assigning reviewers to proposals, reviewing and making final decisions.
- Elegant interface---Functions for program chair are organised into menu and submenus,
- Excellent help--- on-line tutorial for authors, reviewers, program chairs.
- Reviewer and program chairs can register to the system and enter/change their profile. Program chair may not know the reviewers before they register.
- Each program chair can select a number of registered reviewers for his division and assign them to proposals.

Detail function demonstration, interface and system architecture is not available.

CHI Reviewer Volunteer Center

(<http://www.acm.org/sigchi/volunteers/rvcenter.html>)

The annual conference for Computer-Human Interaction(CHI) invites volunteer reviewers to review the conference papers.

The registration of reviewers includes two steps:

- Reviewer submits his/her email to the conference via web form. The conference system generates a user ID and password, email them to reviewer along with the instruction and link to login to registration web page.
- On registration web page reviewers can specify contact information, participation categories and reviewing background.

START: Submission Tracking and Review Tool set

START Conference Manager is another conference management system, which supports paper submission and review process.

In START, the papers are submitted with form-based uploading in postscript or PDF format.

A private directory is automatically created, which contains an indexed, hyperlinked list of papers; PC members can then download and print the papers. START tools set up exclusive login/password access for PC members. Another tool helps to assign papers to PC members, taking into account their preferences, as well as their conflicts of interest. START's report module produces multiple collations of the review reports - for the chair's use during the review period and then for the committee's use during the "vetting stage" (Gerber, et al., 1999).

START was developed with Perl5 and HTML, runs on Unix/Apache.

AAAI conference management software

AAAI is for American Association for Artificial Intelligence, it only supports the paper submission and assignment to reviewers. The paper abstract and full paper are submitted separately. The paper abstract is submitted via web form along with author and co-author information, and two classification selections. The full paper is submitted via ftp (AAAI, 2001).

WIMPE

WIMPE (Web Interface for Managing Programs Electronically) is a set of Perl and Tcl/Tk scripts designed to support the management of electronic paper submission and reviewing for a conference (Nicol, 2001).

The features of WIMPE:

- System architecture contains a web server httpd/APACHE and WIMPE server. The web server contains some CGI scripts that are executed via web pages, and in turn the cgi script trigger the commands or programs on the WIMPE server which is actually

the application server and database server. The connection between CGI scripts and WIMPE server is through a socket.

- System can be customised by editing a text file of keywords and editing a Perl script containing variables. The variables represent commands and path name of programs such as bring up a PDF viewer, or send email. There are also variables for conference info such as name, URL, chair password, and reviews due date, etc.
- Authors, PC members, referees interact with the system via web pages. They can register, submit paper, access reviewing load and submit reviews. The program chair locally work with a GUI developed with Tcl/Tk to set up the system, assign papers, generate various reports and web pages, and send email to users.
- The GUI provides some good functionality for program chair:
 - ◆ Check if the newly received paper is printable.
 - ◆ Conversion of paper between Postscript and PDF format.
 - ◆ Find if a paper was submitted by a program committee member. If so, the identity of reviewers should be protected from program committee author.
 - ◆ Generate reviews report for paper with average rating and overall recommendation
 - ◆ Generate Variant Reviews Report for identifying papers that have conflict review opinions.

The major limitation of WIMPE is that the program chair can only work locally with the GUI, since the GUI is not available on line.

3.8 Summary

In this chapter we reviewed some existing conference management systems/tools, focusing on their support in paper review process. Here we summarise the features of different systems in supporting the paper review process---paper submission, assignment to reviewer, review and discussion, selection and notification. This information will then become the basics for requirement analysis of the system developed in this project.

Paper submission

Most systems/tools provide online submission of papers' administrative information. For submission of full papers, ConfMan, EDAS, AAAI and WIMPE use FTP, POS8 uses email attachment, and CMT and START use direct form-based uploading.

Each submission should be assigned a unique ID for tracing the subsequent process. To associate the file of a paper with its administrative information (title, abstract, authors, and keywords), in most cases the file is given a name regarding the submission's unique ID. If papers are submitted with FTP or email attachment, the submission has to take two steps--- first, the author submit the paper information and receive the unique ID, then, referring to the ID, submit the full paper. With form-based uploading, paper file can be submitted together with its administrative information and software takes care of the association. Obviously, form-based uploading is accurate and efficient.

Almost in all the systems, author's contact information is included in the paper submission. If one author submits many papers, he/she needs to enter the contact information many times. In addition, if the author is also a PC member, the contact information is repeatedly entered and saved. This inefficiency could be avoided by separating the user registration from paper submission and applying database.

Paper assignment

For paper assignment there are four aspects to consider: reviewer registration, reviewers preference/expertise, paper assignment and paper distribution.

In some systems reviewers are only PC members (CyberChair, ConfMan, CMT), while in systems for some others conferences volunteer reviewers are invited to review papers (EDAS, AREA, CHI, ICALT2001).

There are three ways for reviewer registration:

- Program chair enters each reviewer's details and assign login and password to them (ConfMan, EDAS and POS8).
- Program chair first assign login and password to reviewer and let them to register other information online (CMT, CHI),

- Reviewers register by themselves online (AREA, ICALT2001 and WIMPE).

The first two methods require the program chair to generate unique login ID and password for each reviewer either manually or with the help of some tools. We believe that letting the reviewers to register by themselves is the most efficient way, but this requires software to check the uniqueness of ID.

There are two kinds of mechanism to take into account the reviewer's preference and expertise in assigning reviewers to papers:

- Bidding for papers, enable reviewers to download and read paper abstract and then bid for papers CyberChair, ConfMan, CMT, POS8, START.
- Registering keywords, enable reviewers to indicate the conference subject areas in which they are interested (EDAS, ICALT2001), EDAS even requires reviewers to rank the subject areas they choose so as to show their preference.

System like EDAS only provides this information to program chair and let him/her to consider it personally when assigning reviewers to papers. Systems like CyberChair, CMT, START incorporate assignment tools to generate an assignment proposal for program chair to modify and apply.

For paper distribution most system allows reviewers to download paper via web. To realise this, CyberChair requires program chair or maintainer to copy files manually to reviewers' directory, POS8 provides tool to do the copying, START and CMT automatically upload files to common directory and setup links to download them for each reviewer.

Reviewing and discussion

Most of the systems provide review form on the web so that reviewers can submit reviews on-line. In some conferences there are PC meeting to discuss and make final decisions on each paper. To prepare the committee members for the PC meetings CyberChair, CMT, EDAS can be set up to allow the reviewer to see others' review on the paper after he/she submit his/her review.

To enable the PC members and reviewers to discuss on papers, ConfMan provides on-line meeting on which reviewers/PC members can enter their own comments and view others' comments for a paper.

Paper selection

Most of the systems allow program chair to monitor the review process and view the reviews.

To support the PC meetings and paper selection, systems produce various reports to summarise the reviews, comment and voting (if applicable).

CyberChair generates overview of all reviews, overview of category, overview of champion and overview of low expertise reviews. ConfMan generates report summary of reviewing and ballots. EDAS and POS8 generate reports of summary score of review in different sorting criteria. WIMPE and START generate report for each paper regarding average rating and overall recommendation.

System like ConfMan enables online voting for papers--- every PC members votes for every paper, then software calculate the number of voting to facilitate program committee to make decision. The drawback of voting is that PC members have different levels of expertise on the subject areas the paper addresses but every of them makes one vote. To avoid that, some systems enable the program chair to adjust the weight for each PC member, which requires a lot of effort and is hardly precise.

Notification and confirmation

There always need for program committee to communicate with authors, reviewers, PC members. Most of the systems provide automatic email by program or triggered by program chair to send notification and confirmation messages with pre-built template.

Customisation

The level of customisability varied from system to system. Systems support customisation in the following aspects: conference information, email message template, text in HTML page and CGI script. Almost all systems require customisation done locally on web servers. We believe it will be more efficient if some of the customisation can be done on-line.

System development and implementation

Rather than a unified 2-tier or 3-tier client-server structure, Most of the systems are developed as a combination of specific-purpose tools, like generate login password, send emails, generate private directory for PC members, etc. The advantage is that it is easy to add independent functions to the system. The disadvantage is that new functions are not easily integrated into the web-based part of the system, so we can see quite a few systems have various sophisticated functions but can only be executed by administrator or program chair locally with commands or in GUI environment.

Many systems don't apply database, the result is that program chair have to do a lot of customisation and setup, like generate unique ID for submissions and reviewers, create private directory for PC members and copy files or create links in that directory. One system uses database independently, like POS8, use ACCESS database to generate report, data is manually imported in to database, and reports are only viewed with ACCESS or printed as hardcopies.

Based on the literature review above, in next chapter we shall present the requirement analysis for conference submission system.

Chapter 4 Requirement analysis of conference submission system

This chapter describes the functionality of the Conference Submission System. It also covers relevant non-functional requirement of the system. We apply the use case approach to gather the dynamic functional requirements of the system. A *use case* is a description of a complete course of events initiated by an actor and the interaction between the actor and the system (Jacobson, 1992). A use case approach requires creating use case diagrams and a use case description for each use case-actor relationship identified in the use case diagrams.

4.1 Overview

In past, for most conferences the authors used to send hardcopies of their papers to conference program committee chair by surface mail. Then the copies needed to be assigned to program members and reviewers along with the hard-copy review forms. When the forms were completed they were returned to committee chair who made final decision on whether to accept the paper or not. The chair then notified the author the final result by post. The submission and review process was rather tedious because of the time consuming surface mail and administrative tasks that come with the process.

Until 1996 the European Conference on Object Oriented Programming (ECOOP) still used hard-copy review forms.

Nowadays people usually use electronic submission for conference, workshops, and journals. Compared to surface mail, sending the submission with email is a big progress. But even so, there's a lot of administrative work for program committee chairs, such as collecting and classifying papers, assigning papers to committee members and reviewers, collecting reviews and recommendations and making decision to accept or reject papers, and so on.

The project aimed to simplify the conference submission and review process by taking advantage of database and web technology to:

- facilitate the work of PC Chairs;
- simplify the process of assigning the papers and in accepting /rejecting papers;
- reduce the time for submission and review ;
- provide friendly user interface for various users of the system; and
- set up security mechanism for storing papers' information and user profiles.

In order to meet the needs of various conferences, the system is aimed to be as flexible as possible, while maintaining its usability and user-friendliness.

4.2 General requirement of a typical conference submission system

A typical conference submission system provides a common workspace for all kinds of its users. The users record and update their work in the system-authors' papers, reviewer's reviews, committee members' recommendations and committee chairs' decisions. The system also provides the workflow control, user authentication, user communication, and information security.

4.2.1 System functionality

Considering the feature required for most conferences, the system should provide support for at least following aspects: system configuration, paper submission, paper assignment, paper review, making decision, notifying authors.

- **System configuration**

Before the system is put to use, the committee chair should configure the system for the specific conference, including setting up conference name, keywords, time lines, submission form and review form prompts and email message templates.

- **Paper submission:**

The system should provide online forms for authors to register and login as well as forms to enter paper information and upload paper. The back-end of the system should store the contact person's profile, every co-author's necessary information, paper information and paper content for further use.

- **Paper assignment:**

The assignment should be done in two steps, the first step is whenever a paper is submitted to PC chair by an author, the PC chair assigns it to a PC member. The particular PC member can be selected according to the keywords matching between the paper and the PC member. In turn, the PC member should assign the paper to a number of reviewers. Because during submission period the papers are subject to change or even canceled by their authors. The PC members can pre-assign papers to reviewers, after the deadline for paper submission the reviewers are notified by email with their assigned papers titles.

- **Paper review**

When papers are assigned to reviewers, the reviewer should be able to login to the system to read the paper abstract and download the content of papers and fill the review forms online.

After all the reviews are submitted for a specific paper, the PC member should be able to make recommendation to PC chair based on the review opinion from reviewers.

- **Making decisions**

For each paper the PC chair should then make final decision whether to accept or reject based on the opinion from PC members and reviewers.

- **Notifying Authors**

Just like the PC members and reviewers are notified of their assigned papers by email, the author should also be notified of the final decision on their papers by email.

To uncover the business logic and clearly identify the functionality of the system, first the users' characteristic should be studied. In terms of use case approach, the actors are to be

defined before the interaction between actors and use cases are to be studied. Therefore in the next section, we shall study the characteristics of the users.

4.2.2 User characteristics

There are six types of users that directly interact with the system, they are:

- paper contact persons,
- program committee chairs,
- program committee members,
- paper reviewers,
- system administrator, and
- unregistered users, who visit the conference website.

Because this is a web-based application, except the system administrator, all other users should be able to access the system using internet browser. Since internet technology is so extensively used, especially in academic area, we assume every author who intends to submit a paper to an academic conference will have the skills like surfing the Internet, filling online forms, sending and receiving emails.

Paper contact person

Paper contact person is usually, but not necessarily, one of the co-authors of the paper. He/she could be also be a secretary or helpdesk in an institute. For one paper there should be only one contact person whose contact information is kept in the system and who takes the responsibility to communicate with the program committee.

The international conferences generally have large number of submissions from all over the world. The system should be easy enough to use without any formal training requirements. Therefore we need to provide enough help information in the system to assure people can use the system correctly and efficiently.

The contact persons should have access to the information of only those papers that they submitted to the conference and their personal account information.

Committee chairs

There are usually one or two committee chairs for a conference. They are main decision makers. They can assign papers to program committee members and reviewers. They make the final decision on accepting or rejecting a paper, design the schedule of the conference, setup and change important deadlines for the conference such as paper submission, review collection and final decision.

They have the highest propriety in the system, they have access to all the paper information and user accounts information.

The committee members including committee chair may or may not be computer expert. Since they are the most frequent users of the system, and use most of the system functions, they should have appropriate online help available. They will access the system through a web browser.

Committee members

Committee members are experts in the subject area of the conference, they will assist the committee chairs to organise the conference and contact with reviewers. In order to properly assign papers to suitable reviewers, they need access to both papers information and reviewers profile information. Since they are members of program committee, they should have access to all papers including those that are assigned to other members by the PC chair.

The number of committee members is limited. Usually they are invited by committee chairs, and their opinions on papers mostly determine the final decisions. The number of papers that the PC members can assign to a reviewer should be determined by the program committee chair.

Reviewers

Reviewers are experts in one or more subjects of the conference. They volunteer to review the papers for the conference. There could be a fairly large number of reviewers from all over the world. The interface for them should be informative and friendly with sufficient help information. Reviewers have access only to their own account information

and they should be able to read the abstract and download the content of papers that are assigned to them. They should not be able to access the papers' author information to maintain the blind review process.

Administrator

The conference submission system requires an administrator who can handle the technical details. This person is not supposed to be expert in the conference subject. But he/she must be computer expert, and should be able to set up and maintain web server and database server. Administrator should be familiar with every function of the system, be able to setup and customise the system, and operate directly on the data in database. The maintenance work could be done using commands provided by database products or via a GUI interface located on a computer connected to the server through local network, not necessary only via a browser.

The task often performed by the administrator is to setup new user name and password for the committee chairs and authorise them.

Sometimes a person may need to play more than one role in a conference, e.g. the PC chair may also be one of the PC members; PC members could also be reviewers and authors. The different interface and different proprieties should be provided to them depending on their login type.

Unregistered users

Unregistered users may be any Internet surfers visiting the web site. They should be able to view the general information of the conference such as schedule, organiser, subjects, etc. If the program committee decides to publish the papers after conference, all the web surfers should be able to view the papers' abstracts. We also need to provide them with the interface to retrieve papers by keywords.

4.3 Functional requirements of the system

We'll analyse the requirement for each type of users one by one. First we'll discuss the common tasks for all types of users, and then we'll discuss the tasks for particular type of users, i.e. contact person, reviewer, PC member and PC chair.

4.3.1 Common functionality for all types of registered users

The common tasks for all types of registered users include *Login*, *Logout*, *Register*, *Change password*, *Modify profile*, *Forget password and user name*, as described in Figure 4-1.

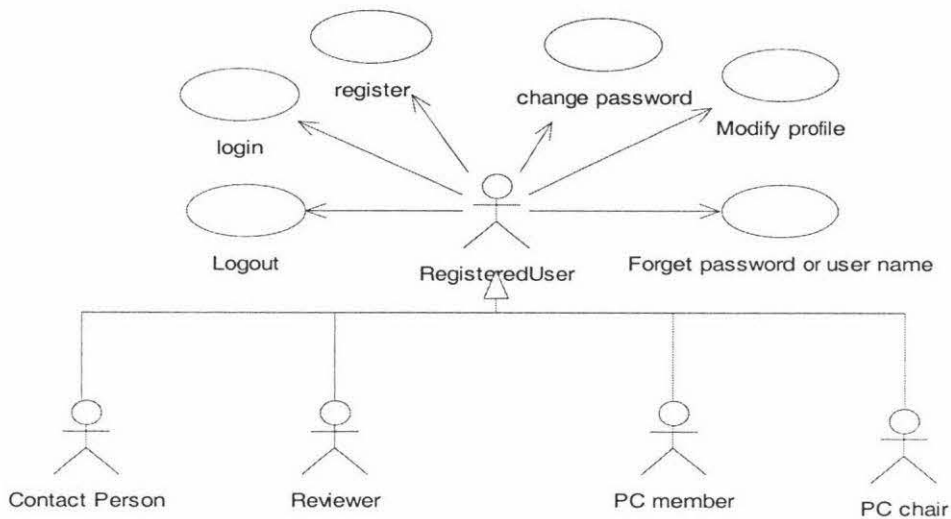


Figure 4-1 common functionality for all register users

4.3.1.1 Registration

The contact person, reviewer, committee member and committee chair need to register before they can use the system, we call them registered user.

On the home page the system need to provide buttons or links for user to go to register interface.

There are two cases a user need to register:

1. The first time the user want to set up an account in the system, the user need to input all the contact information such as address, email. If the user want to register as reviewer, committee member he also need to provide the brief CV and keywords which show his expertise and the kind of papers he wants to deal with.
2. After user has an account in the system e.g. contact person, he wants to register as another type of user e.g. reviewer. To avoid user entering the same information once

again, the system should provide the function for user to get his account by providing user name and password.

Registration information

- User name and password

Every registered user need to select a user name and a password for later logging in to the system. The user name should be unique in the conference, the system should check the uniqueness when user registering.

- User type(s)

User has to specify the user type(s) he wants to register as, this means a user can choose one or more user types to register. The *reviewer* user type does not need explicit authorisation, which means if a user registers as a *reviewer*, he will be able to login as a reviewer automatically. But the user types of committee member or committee chair need to be authorised, committee member need to be authorised by committee chair, committee chair need to be authorised by system administrator. Every registered user is able to login as a contact person.

- Basic user information

Every user need to provide contact information including: e-mail, first name, last name, title, address, affiliation, city, state/province, country, zip code, work phone and Fax.

The email should be unique in the conference. The system should check uniqueness when registering.

If a user registers as a paper contact person only, he only need to provide the above contact information, we call it *basic user information*.

- Special user information

If a user wants to register as a reviewer, a committee member or a committee chair, he need to provide more information to show his expertise and qualification. In addition to the *basic user information*, the user also need to provide: position in his own organisation,

brief CV, and up to five keywords selected from the conference keywords list. We call this *special user information*.

Usability

When users submit the registration form the system should check the validation of input field, and the uniqueness of user name and email. In case of an error occurs, the system should give feedback to the user and allow the user to re-enter some fields and submit again.

After the user succeeds in registration, the system should display confirmation message and e-mail the user name and password to the specified e-mail address.

4.3.1.2 Login

The system should provide login interface with user name and password input field and login user type selection. For each user type, the successful login will lead the user to the interface particular for that type of user. From there the user can start to perform the tasks.

If the user makes a mistake in login, such as input wrong password/user name or select unregistered or unauthorised user type, the system will display corresponding feedback and allow user to login again.

4.3.1.3 Logout

In the interfaces for all types of user, function “Logout” is provided. Once a user chooses to log out, the user’s session is terminated, all the function and links that requires user identification will be unavailable until user chooses to login again.

4.3.1.4 Forget password or user name

In case the user forgets user name or password, system can email them to the user’s registered e-mail address.

The user enters his/her email address and clicks a button “send me my account detail”, the system then checks user accounts, if email address exists, the user’s user name and password are sent to that address. Otherwise the system prompts that the email-address does not exist.

The users can use this function when they can not login or can not get existing account information for registration.

4.3.1.5 Modify profile

This function is for users to modify account information. Unlike user registration, with this function users can only change their personal information in the account, such as address, keywords, and so on.

When users submit the updated profile to the system, it will do the input validation and give error message in case an error occurs.

4.3.1.6 Change password

For the purpose of security, change password function is not included in modify profile function. User can only change password by providing correct user name and current password. The new password is needed to be entered twice to avoid typing mistakes.

System should be able to carry out the input validation, correctness of username and old password, and consistency of the newly entered password.

Interfaces for all major user types should provide links to change password function.

4.3.2 Specific functionality for paper contact person

Particular tasks for paper contact person include *Submit paper*, *View submitted paper*, *Cancel submitted paper* and *Download paper*, as described in figure 4-2.

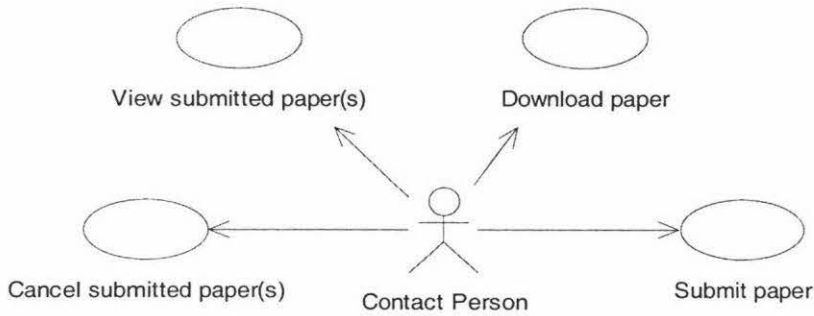


Figure 4-2 particular tasks for paper contact person

4.3.2.1 Submit paper

Paper submission is a heavily used function in conference submission system. This function enables user to enter paper information, co-author information and upload paper content.

Paper information includes paper title, abstract, keywords, category and whether the committee member is allowed to change the category.

One paper can have up to five co-authors. Co-authors' information includes first name, last name, affiliation and country.

Paper content should be in a Word file or PDF file stored on user's local computer. Paper is submitted in two steps, submit paper and co-authors' information, upload the paper content.

Submit paper and co-authors information

When user enters and submits the paper and co-author information, system will assign a unique ID for the paper, and record the contact person's user name with the paper information. In addition to the input field validation, system also needs to check the

uniqueness of contact person's user name and paper title (one contact person can not submit two papers with same title).

Upload paper

After the paper and co-author's information is submitted the system need to provide the user interface for uploading the paper file from the user's local machine. For uploading the file the system will

- generate the file name as paper x , x is a number representing the unique paper Id,
- save it in the contact person's folder (if not exists create one),
- record the file path as paper's URL,
- display confirmation message and
- send confirmation e-mail to user's e-mail address.

Submission deadline

The committee chair should set paper submission deadline. If the users attempt to use the "submit paper" function after the submission deadline, the system should forbid the submission and give prompt message.

4.3.2.2 View submitted paper

Logging in as paper contact person, the user can view the paper title list of the papers he submitted, and can select one of the papers to view its information.

4.3.2.3 Cancel submitted paper

Logging in as paper contact person, the user can select a paper from the list and choose to cancel it. The system should delete the paper information, co-author information and paper content.

4.3.2.4 Download paper

Contact person should be able to select a paper from the paper list and choose to download the paper file. This is mainly for contact person to test if his paper content is correctly uploaded.

The saving file interface should be provided.

4.3.3 Particular functionality for reviewers

Tasks for reviewer include download paper, make and modify reviews as described in Figure 4-3.

The reviewing process should be “double blind”, which means the reviewer should not know the paper’s author while the paper’s contact person should not know who reviews the paper. So what the reviewer can see of the paper is the paper number, title, abstract, category and content.

When the user login to the system as a reviewer, the system should display the paper list along with the function buttons to download the selected paper and review the selected paper.

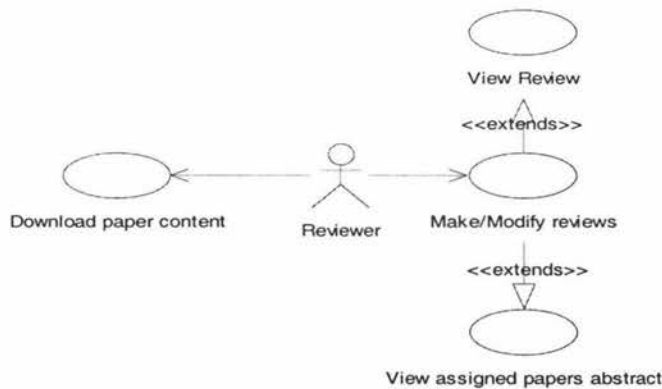


Figure 4-3 particular tasks for reviewer

4.3.3.1 Download paper

When user select to download a paper that is assigned to him/her, the system should provide the saving file interface to enable user to modify the filename and choose the path to save the file.

4.3.3.2 Make or Modify review

If user selects to review paper, system will display the paper's abstract and the review form.

Within the review form user is able to:

- select the relevant level from (very, moderately, not relevant), and enter explanation of the rating;
- select the significant level from (very, moderately, not significant), and enter explanation of the rating;
- select the significant level from (very, moderately, not significant), and enter explanation of the rating;
- select the original level from (very, moderately, not original), and enter explanation of the rating;
- comment on the quality, clarity and appropriateness.

For overall summary of review, user needs to provide:

- overall recommendation selected from (accept, reject, accept as new category), if accept as new category, user need to select a new category from the category list;
- recommended revision if user have any suggestion;
- the confidence level in his the appropriateness for being the referee of the paper, selected from (very confident, confident, fairly confident and not confident).

In terms of usability, the system needs to check input field validation, give feedback for any mistake. In addition, if the paper being reviewed marked as “not allow committee member to change the category”, the system should disable the “accept as new category” option in overall recommendation and disable the “new category” selection.

Before the committee chair makes the recommendation on the paper, the reviewers can modify the review any time. The process to modify review is just to fill or modify the review form and submit again.

Review submission deadline

Committee chair should set up a review submission deadline. The reviewers are supposed to submit their reviews before this date. But in reality, some reviewers may not submit the review in time or simply do not review the paper. In this case the committee member in charge of the paper has to remind the reviewer or has to assign the paper to some other reviewer(s), so even after the review submission deadline, the “review paper” function is still available for reviewers. But after the deadline for committee members’ recommendation, no review could be made or modified.

4.3.4 Particular tasks for committee members

Committee members’ major tasks are *assigning papers to reviewers and making recommendation on papers to committee chairs*. Figure 4-4 shows these tasks

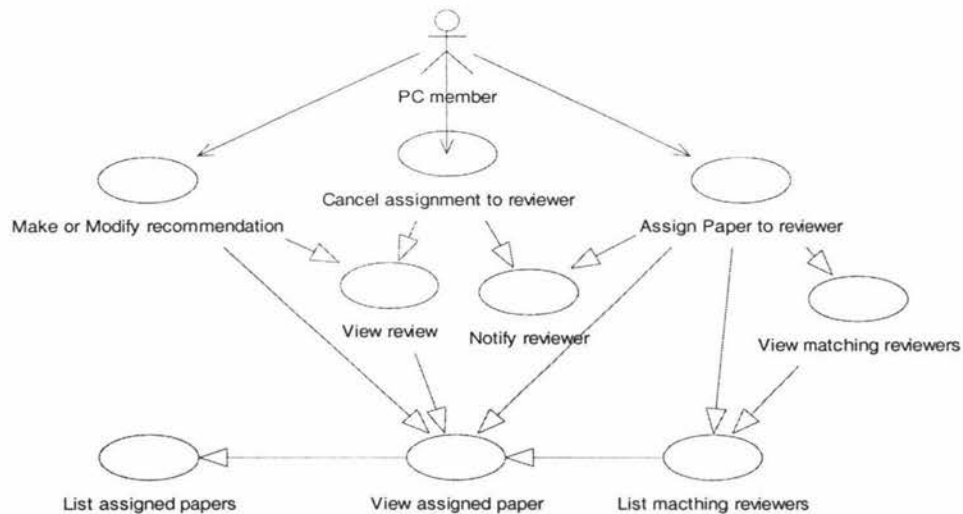


Figure 4-4 Particular tasks for PC member

4.3.4.1 Assign paper to reviewer

To assign papers to reviewers two issues need to be considered:

- **assign paper to the right persons**

This means the reviewer assigned to should be expert on the topics the paper talks about. We take the approach of keyword matching for this purpose, the basic idea is that the more and better matching between the paper's and the reviewer's registered keywords means the better candidate the reviewer is for this paper. The matching extent is represented by a number that we call matching-coefficient. Based on this number, the list of relevant reviewers is sorted for each paper so to facilitate the committee member's selection.

- **avoid conflict assignment**

This includes: a reviewer is assigned with too many papers; a paper is assigned to one of it's co-authors or to the person with same affiliation as authors, e.g. same department.

When the committee members login to the system, the committee members' interface is shown, on which the function to search paper(s) assigned to him is provided. The system should enable user to:

- search paper by paper information;
 - ◆ paper number range;
 - ◆ keywords in paper title;
 - ◆ paper's contact person ID.
- search papers by reviewer's ID, which means search papers that are assigned to a particular reviewer.
- search papers by assignment status:
 - ◆ not assigned to any reviewer;
 - ◆ completely assigned-the papers has been assigned to numbers of reviewers required by conference committee, e.g. 3;
 - ◆ not completely assigned-the number of reviewers to whom the paper is assigned is less than that required by conference committee.
- search papers by review status:

- ◆ completely reviewed-required number of reviews have been submitted for the papers;
- ◆ not completely reviewed-not enough reviews has been made as required by the conference committee.
- search papers by recommendation status:
 - ◆ not made;
 - ◆ accept;
 - ◆ reject;
 - ◆ accept as new category.

The searching criteria are to limit the range of papers, i.e. if the criterion is specified/selected only the papers meeting the criteria are listed, otherwise all papers assigned to the committee member are listed.

From the paper list, the user should be able to select one paper and perform the following tasks on it:

- view paper information, download paper content;
- view the paper's reviews;
- select a reviewer and view his user profile;
- select a reviewer and assign the paper to him;
- cancel the assignment;
- make recommendation on the paper;
- cancel recommendation.

View paper information and download paper content

The paper information, contact person Id and the co-authors information are displayed on the screen once the user select a paper and chooses to work on it. The function “download paper content” is provided, see 4.3.2.4.

Select a reviewer and view his/her user profile

As described above, when committee member selects a paper, the keyword-matching reviewers' user names are listed in descending order of matching-coefficient, along with the number of assigned papers to that reviewer.

Committee member should be able to select one reviewer and choose to view the reviewer's profile, the profile could be displayed on another screen.

View the paper's reviews

Besides the paper information, the existing review list is displayed on the committee member's paper screen, the list contains reviewer Id, review made (true, false). Committee member could select one of the reviews and choose to view the review. The review is shown on another screen or window.

On this screen the committee member should also be able to send e-mail to the reviewer in case the committee member wants to remind the reviewer.

Select a reviewer and assign the paper to him

Committee member can select a reviewer and choose the function to assign the paper to the reviewer.

To ensure the quality of reviews, there should be a limit number of papers assigned to one reviewer, this should be specified as system parameter.

The system should check if the reviewer is assigned with papers over the number limit, if true, give error message to the user.

If the selected reviewer happens to be the author of the paper or has same affiliation with one of the co-author, the system should give error message to the user.

If assignment succeeds, the system will add a line in the **review** list of this paper, and specify the "review made" as "false". The system also sends e-mail to the reviewer with the paper title and paper number.

Cancel the assignment

If the "review made" column in the review line is still "false", the committee member can select the line and choose to cancel the assignment. The system will delete the line in the review list. But if the review has been made, committee member can not cancel it.

4.3.4.2 Make and cancel recommendation on the paper

Make recommendation on paper

All the actions above are to help the committee member to perform his real task---make recommendation, on the committee member's paper screen there should be a section for recommendation.

The recommendation could be "not made", "accept", "reject" or "accept as new category". If committee member chooses the last option, the new category could also be selected.

If the paper's authors do not allow committee members to change the category, "accept as new category" function is disabled.

Cancel recommendation

Before the PC chair makes the final decision on the paper, the committee member can choose to cancel the recommendation, the system will change the recommendation to "not made".

4.3.5 Particular tasks for committee chair

Major tasks for committee chair are *paper management*, *user management* and *set up system parameter* as described in Figure 4-5.

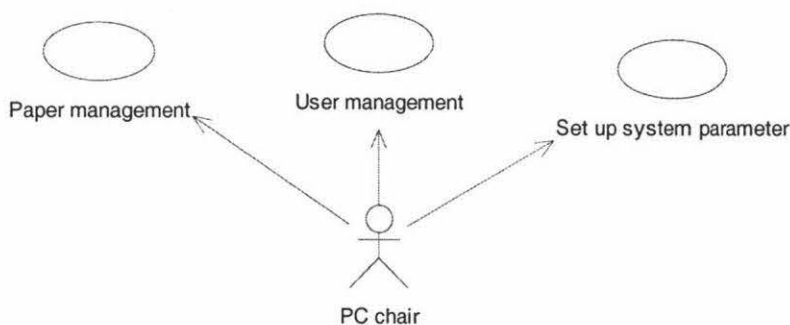


Figure 4-5 The main tasks for PC chair

- paper management
 - ◆ whatever the committee members can do with a paper
 - ◆ assign papers to reviewers over the number of papers per reviewer limit
 - ◆ make final decision and notify the contact person

Figure 4-6 shows the tasks in paper management

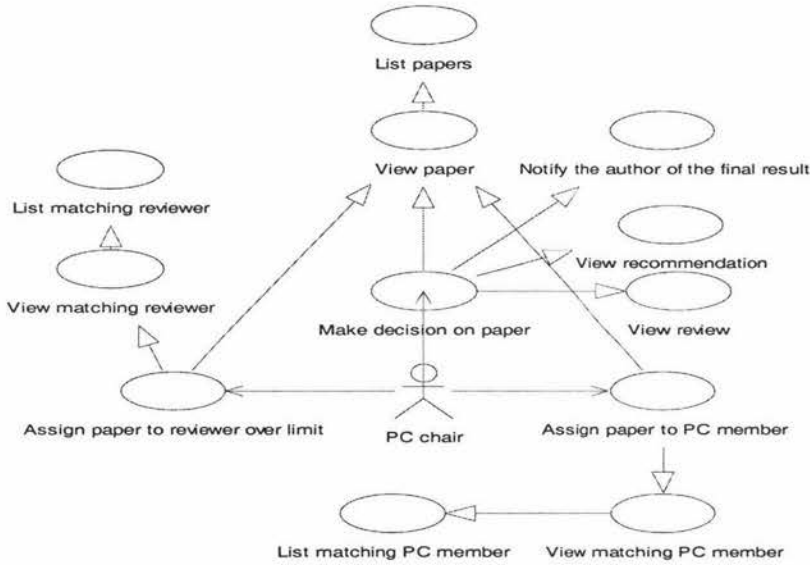


Figure 4-6 functionality for paper management

- user management
 - ◆ search user by user types and authorisation status, e.g. authorised reviewer
 - ◆ authorise committee member/reviewer
 - ◆ cancel committee member and reviewer's authorisation

Use cases in user management are described in Figure 4-7.

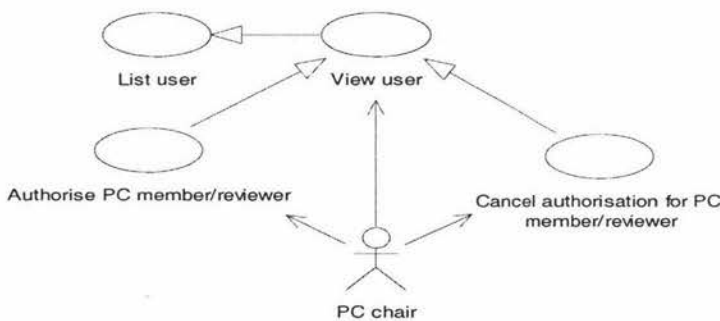


Figure 4-7 functionality for user management

- set up system parameters

system parameters include conference timelines, conference name, keywords, etc

- ◆ deadline for paper submission
- ◆ deadline for review submission
- ◆ deadline for committee members recommendation made
- ◆ deadline for committee chair's decision made
- ◆ whether to publish the selected papers
- ◆ if the selected paper to be published, the date of publishing
- ◆ the maximum number of papers committee member can assign to one reviewer
- ◆ conference name
- ◆ the root directory for storing conference paper files
- ◆ conference keywords

when special users (reviewer, PC member and PC chair) registering they need to select 1 to 5 of the keywords as their keywords to represent their expertise. When contact person submitting a paper, he need to select 1 to 5 of the keywords as the paper's keywords.

When the committee chair login, on the main committee chair's screen, the three major function above are provided, each leads to a sub screen.

4.3.5.1 Assign papers to committee members

After the PC chair selects a paper, the system will display a PC chair's paper page. It's a regular task for PC chair. Just like PC member can assign papers to reviewers, the PC chair should assign each paper to a PC members based on the keyword matching. The PC chairs can view the paper information, download paper content, view PC members' user profile and assign paper to PC member.

One paper can only be assigned to one PC member.

4.3.5.2 Assign paper to reviewers over limit

The process to assign paper to reviewer is similar to what is described in 2.3.4.1. The difference is that the PC chair can assign over the limit of papers per reviewer for PC member. This is not a regular task for PC chair, it happens when one paper can not find enough “free” reviewers.

4.3.5.3 Make decision on papers and notify the author

Make decision

Working on a paper, the PC chair can view the paper’s reviews and the recommendation of the PC member who is in charge of this paper, if he want he can also view the paper’s information and download content. Based on these, the PC chair can make decision whether to accept or to reject the paper. The decision could be “not made”, “accept”, “reject”. The PC chair should make decision on every paper, so this is a regular task.

Cancel decision

The PC chair can cancel the decision that is already made on a paper, the system will change the recommendation to “not made”.

Notify the author of the final result.

After making the decision, the PC chair can choose the function to notify the contact person of the final result. The system will automatically generate a message from the templates depending on the decision(accept/reject) and send it to the contact person’s email address.

4.3.5.4 Authorise committee members and reviewers

Authorise PC members is a regular task for PC chair. Every user can register as PC member, but can not login as PC member before authorised.

PC chair usually does not need to authorise reviewers, because once a user registers as a reviewer he can login as a reviewer. To keep some malicious user from being assigned with papers, the PC chair should be able to cancel authorisation of a reviewer.

In order to authorise users, the PC chair can search users by authorisation status (“authorised”, “not authorised”) and user types (“contact person”, “reviewer”, “PC member”, “PC chair”).

PC chair can select a user and view the profile.

4.3.5.5 Set up system parameters

The system should provide the interface for PC chair to set up system parameters such as timelines, keywords, etc.

The parameter should be set up in two steps:

1. update the parameters, which means to save the new parameters;
2. apply parameters, which means load the parameters to the currently running application.

4.4 Non-functional requirement

4.4.1 Scalability

The system is for online paper submission, it should support number of users to access system and perform their tasks simultaneously.

At a typical conference, there are usually 1 or 2 PC chairs, 10 to 20 PC members, 100 to 200 reviewers and 200 to 500 contact persons. Contact persons will login and use the system for paper submission during the submission period. The submission period is a fairly long period---several months, and during this period the PC members and reviewers don't have much to do with the system except registering. The review period is relatively short and with intensive operations by reviewers, PC members and PC chairs. But during the review period the contact persons seldom access the system.

Therefore, it is assured that on average, the system should accommodate 50 concurrent users' connections and the response time for a user request should be less than 10 seconds.

The system needs to store all the paper information, review information and user profiles in database server, in addition, all the paper content should be stored on server in the form of Word or PDF document. On the basis of above assumptions about the size of the conference web server and database server for one conference can reside on a computer with 20 GB free disk space.

4.4.2 Compatibility

Since we want the system to be applied on various conferences, we need the server side program of the system can be installed and run on different platforms, such as windows NT or Linux.

As a web-based system, the user must access the system through web browsers. We expect users can use the system via Internet Explorer 4.0 and above or Netscape 4.0 and above without any special configuration, e.g. plugins.

4.4.3 Security

All the registered users should present user name, user type and password to login to the system. Contact persons can only access their own paper information, reviewers can only view papers that are assigned to them. PC chairs and PC members can view other users' profiles but can not change them.

The privilege and login status should be checked for every user request.

4.4.4 Adaptability

To be adapted to different conferences, the system should be configured with conference parameters:

- Conference name, keywords, location, time lines, and root directory to store paper files.
- Message templates for notification and confirmation.
- Prompts in paper submission form and review form.

4.4.5 Maintainability

Exceptions will occur due to imperfect system design and resource conflicts, all the exceptions should be recorded in a log file so that the system administrator can detect and correct the problems in time.

4.5 Summary

This chapter analysed the requirement of conference submission system. First the user types of the system are defined, then the functional requirement are discussed in the form of groups of tasks for each type of user and finally the non-functional requirements are specified in terms of scalability, compatibility, security, adaptability and maintainability.

In the next chapter, we will study some popular web programming techniques and decide what techniques we shall use for implementing our system.

Chapter 5 Web programming techniques and our choice

5.1 Introduction

This chapter will analyse the features of current web programming techniques and describe our choice---the Java solution.

First the web application and web programming are introduced. Then the web programming techniques of CGI, Java Servlet, JSP and others are briefly discussed. Finally the Java solution is illustrated in detail.

5.2 Web-based application

Generally speaking, web-based application is a software application running on World Wide Web. Web-based application can be classified as a kind of client/server application. Comparing with applications with ordinary client/server architecture, web based application has some special features, see Figure 5-1

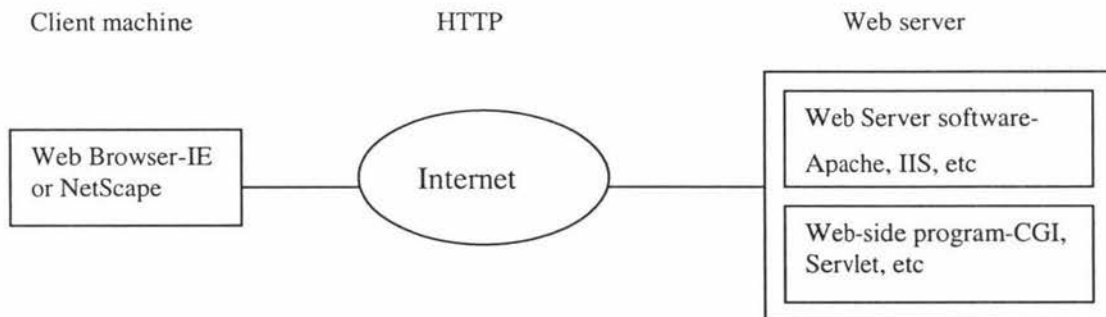


Figure 5-1 Architecture of web-based application

- *Web-based applications need support of web server.*

Web server is a computer that delivers (*serves up*) web pages. Every web server has an IP address. Any computer can be turned into a web server by installing server software and connecting the machine to the Internet. There are many web server software applications, including Apache, NCSA, Microsoft IIS, Netscape and others.

By “web server” here we only mean http server.

- *Web-based applications need support of web browser.*

Web browser is a software application used to locate and display web pages. The two most popular browsers are Netscape Navigator and Microsoft Internet Explorer. In terms of client-server architecture, web browser is the client end application, which can take the user input and send request to web servers, receive the response from web server and display the output to user in the form of HTML pages.

- *Web-based applications require communication between web server and browser.*

The communication complies with the protocol of HTTP. HTTP stands for ***Hypertext Transfer Protocol***. It's the network protocol used to deliver files and other data (called *resources*) on the World Wide Web.

HTTP uses the client-server model: A Web browser is an *HTTP client* because it sends requests to an *HTTP server* (Web server), which then sends responses back to the client.

An *HTTP client* opens a connection and sends a *request message* to an *HTTP server*; the server then returns a *response message*, containing the resource that was requested. After delivering the response, the server closes the connection. Since HTTP does not maintain any connection information between transactions, it is described as *stateless* protocol. In the next chapter we'll see that this feature brings the user-tracing problem in the web programming.

There three types of HTTP requests--- GET, POST and HEAD :

- ◆ GET is the most common HTTP method; it says "give me this resource". And send the local path for the requested resource (URI)
- ◆ A POST request is used to send data to the server to be processed. There's a block of data sent with the request, in the message body. The request URI is usually a program to handle the data in the request.

- ◆ A HEAD request is just like a GET request, except it asks the server to return the response headers only, and not the actual resource (i.e. no message body). This is useful to check characteristics of a resource without actually downloading it, thus saving bandwidth.

- *Web-based applications are supposed to support multi users.*

Web-based applications are always exposed to requests from multi users simultaneously. Although the Web servers tends to take advantage of multi-threading or multi-processing to handle the concurrent user requests (Hughes *et al*, 2002), as a resource residing on the web servers, the program of Web-based applications are expected to support multi-threading or multi-processing.

5.3 Web programming techniques

HTML (HyperText Markup Language) is the basic language understood by all WWW (World Wide Web) clients. But it is proving insufficient by itself to develop the myriad Web-based applications envisioned (Ford *et al*, 1996). To develop sophisticated Web-based applications, web programming is extended by server and client programs. Server side programs are run on Web servers, client side program are run by Web browsers.

5.3.1 The server side programming techniques

5.3.1.1 Common Gateway Interface (CGI)

Common Gateway Interface is a specification for transferring information between a Web server and a CGI program. A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any programming language, including C, Perl, Java, or Visual Basic. (Webopedia)

CGI programs are the most common way for Web servers to interact dynamically with users. Many HTML pages that contain forms, for example, use a CGI program to process the form's data once it's submitted.

5.3.1.2 Java servlet

The name “servlet” is derived from “server and applet”, which means an applet that runs on a web server. Unlike applet, servlet is Java small program running on the web server. Once a servlet is loaded into memory, it stays there and deals with multiple requests from browser simultaneously taking advantage of Java’s multithreading feature. Comparing with CGI program, which is reloaded into memory for each request, the servlets are much faster. (Webopedia)

Because the result of server side program has to be presented as HTML pages, CGI or servlet program have to output a lot of HTML code which makes the program unnecessarily long and cluttered. Even worse, when you want to change the look and feel of interface (HTML page) you have to modify code and recompile it. That means the presentation logic and business logic is not separated.

Several techniques have been invented to overcome this shortcoming, they are ASP, PHP and JSP, etc. In the following section these techniques are briefly introduced.

5.3.1.3. ASP

Active Server Page (ASP) provides an server side environment which can integrate scripting language into HTML. The scripting language could be VBScript or Jscript(MS supported subset of JavaScript). The embedded scripting programs are executed on the server side in place of original CGI programs. ASPs are usually run on Microsoft platforms, such as WindowsNT Server4.0 with Microsoft IIS3.0/4.0 or Windows95/98 with Microsoft Personal Web Server1.0a/4.0. (Webopedia)

5.3.1.4 JSP

JavaServer Page is a server-side technology, Java server pages are an extension to the Java servlet technology. It was developed by Sun as an alternative to Microsoft's ASPs. JSPs can integrate Java source code with HTML code, separating the page logic from the static elements -- the actual design and display of the page. Embedded in the HTML page,

the Java source code and its extensions help make the HTML more. JSPs are not restricted to any specific platform or server. (Webopedia)

5.3.1.5 PHP

PERSONAL/PROFESSIONAL Hypertext Preprocessor is a server-side, HTML embedded scripting language used to create dynamic Web pages. In an HTML document, PHP script (similar syntax to that of Perl or C) is enclosed within special PHP tags. PHP can perform any task any CGI program can do. (Webopedia)

5.3.2 The client side programming technique

Another increasingly common way to provide dynamic feedback for Web users is to include scripts or programs that run on the user's machine rather than the Web server. These programs can be Java applets, Java scripts, or VBScript. These technologies are known as client-side solutions.

5.3.2.1 JavaScript

A scripting language developed by Netscape to enable Web authors to design interactive sites. Although it shares many of the features and structures of the full Java language, it was developed independently. Javascript can be embedded in HTML source code, enabling web pages to display dynamic content. (Webopedia)

5.3.2.2 VBScript

Visual Basic Scripting Edition, a scripting language developed by Microsoft and supported by Microsoft's Internet Explorer Web browser. VBScript is based on the Visual Basic programming language, but is much simpler. In many ways, it is similar to JavaScript. It enables Web authors to include interactive controls, such as buttons and scrollbars, on their Web pages. (Webopedia)

Unlike server side ASP, client side VBScript and JavaScript don't need support from web server, they are directly interpreted in Web browsers. Both Netscape and Internet Explorer support JavaScript. VBScript is only supported in Internet Explorer.

5.3.2.3 Applet

Applet is a Java program designed to be executed from within another application. Unlike an application, applets cannot be executed directly from the operating system. Web browsers, which are often equipped with Java virtual machines, can interpret applets from Web servers. Because applets are small in file size, cross-platform compatible, and highly secure (can't be used to access users' hard drives), they are ideal for small Internet applications accessible from a browser. (Webopedia)

5.4 Selection of the programming techniques for the project---Java servlet, JSP, JavaBean, JDBC and MySQL

To satisfy the requirement analysed in chapter 2 and avoid the limitations of most existing conference paper submission systems addressed in chapter 3, our system is supposed to be an interactive Web-based application with database integrated as back-end.

Client side VBScript and JavaScript do not support database connection independently, they are usually embedded in HTML code to do the tasks like input data validation, error message display. We consider incorporating them in the system but not as a major choice.

The major choice is among the server side techniques. CGI is a traditional way to handle web-to-database applications. The performance of CGI is stable. The main drawback of CGI technique is overall inefficiency in handling concurrent client requests, which is due to the heavy weight context switch of processes between client requests, thus it is not suitable for high-traffic applications that need to process a large number of client requests (WU *et al*, 2000).

Comparing with CGI program, ASP programs have the advantages of inherently multi-threaded, quick to write, and fast to run and allowing a greater number of concurrent users.

The limitation of ASP is that it is mostly restricted to Windows platform, the Web server-- IIS and database server---SQL Server, are commercial software, which may cause extra cost to users.

PHP provides a programming approach similar to ASP, it has broad support of databases rather than being restricted to Windows platform. Combining with Apache web server and MySQL, PHP becomes a popular web project solution. According to Amanda W.Wu's benchmark experiment (WU, 2000) Java Servlet has better performance over PHP3.

Now we have Java servlet, JSP on server side and Java Applet on client side. These techniques together with Java database package JDBC and Java Bean form the complete Java solution for Web-Based projects. The Java solution is discussed in details in the following section.

5.5 Java solution

Servlets provide a component-based, platform-independent method for building web-based applications, without the performance limitations of CGI programs. Servlets have access to the entire family of Java APIs, including the JDBC API to access databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the Java language, including portability, performance, reusability, and crash protection.

JavaServer Pages (JSP) is an extension of the Java Servlet technology. It separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content.

JSP is quite powerful. On the one hand, it can include Java code within HTML pages to implement any flow control logic, on the other hand, JSP can apply JavaBean and custom tag library to solely serve as a dynamic and interactive web page.

Tomcat

Both Servlet and JSP need to run within Servlet container, which is the software that creates, loads, and executes servlets as needed. The message passing between Web server and servlet container is shown in Figure 5-2. Web browser can send requests to Web server, if it's a request for a HTML page, the Web server fetch the HTML page and send back to Web browser, if it's a servlet request (request service from a servlet), the Web

server pass the request to servlet container. The servlet container will create/invoke the servlet as necessary.

Tomcat is chosen as our servlet container, it contains the Java packages for servlets and JSPs to deal with HTTP objects- request, response, session, etc. Tomcat can either work together with Apache or work in the stand-alone mode, in stand-alone mode it acts as a web server as well. This feature makes it possible to run and debug servlet/JSP on a single machine in development.

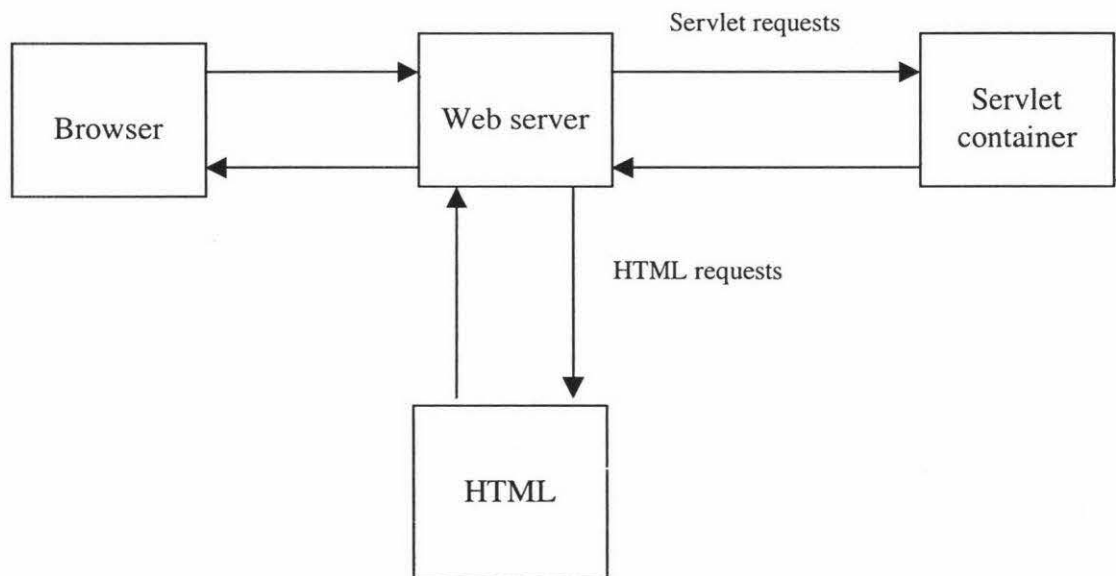


Figure 5-2 Servlet requests are passed to servlet container.

JavaBean

As Java's component architecture, JavaBean is extensively used in Java programming. In cooperation with JSP, JavaBean provides a way to separate Java code from HTML pages. JavaBean can provide storage for dynamic information in JSP page presentation, it can also reflect database record in the programming.

JDBC

In terms of database connection, Java programs including Java servlet have advantage over PHP and ASP. ASP relies primarily on Microsoft's products such as SQL Server. PHP uses different database functions and methods for handling different databases (WU, A. W. et al. 2000). In contrast, Java programs use JDBC to provide uniform functions and

methods for data retrieval and manipulation in different databases if only the database vendor provides JDBC driver.

JDBC stands for Java Database Connectivity API. JDBC 2.0 Core API refers to the Java package `java.sql`, which is included in Java2 Platform Standard Edition (J2SE) since version 1.2. (Sun MicroSystem, 1998) This package provides *Connection* interface for searching database driver and setting up connection with specific database, *Statement* interface for building and executing SQL queries to retrieve and manipulate database, *ResultSet* interface to return rows of database data for reading and even updating. (Sun MicroSystem, 1998)

MySQL

MySQL is a SQL database server. It's very popular in Web-sites design because it's reliable, fast and capable of managing very large databases. As speed is a critical requirement for Web-based application, MySQL does not support triggers and foreign key because they may cause performance penalty. Another limitation of MySQL is that it does not support subselects.

Web techniques selected for the project

Now we summarise our choice of Web techniques in a diagram--- Figure 5-3, Apache and JDBC driver reside on the same server machine, MySQL server can either reside on the same machine or another machine.

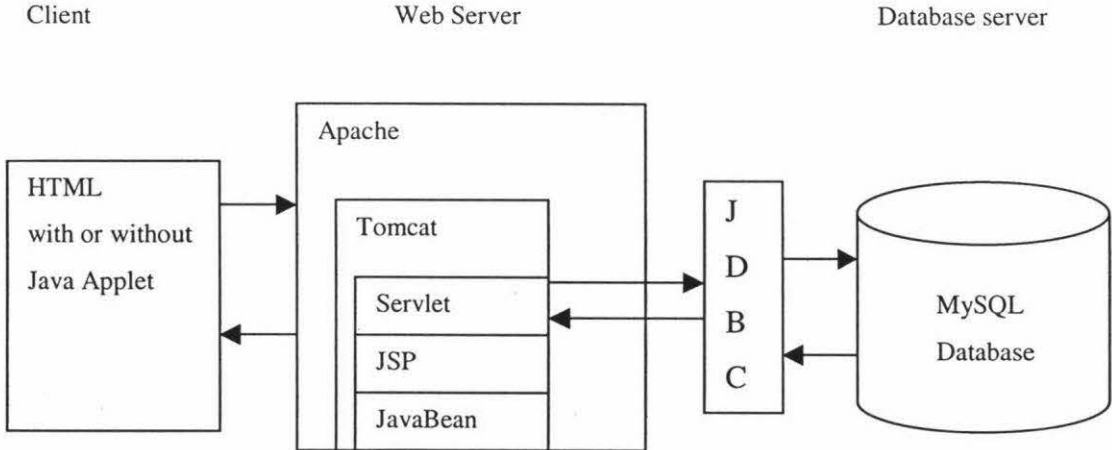


Figure 5-3 Web techniques selected for the project

The usage of applet

Figure 5-3 implies Java applet can be embedded in HTML pages (JSP page as well) to act as client side program. Applet has the strength to build sophisticated windows-style user interface taking the advantage of Java AWT components (Java platform 1.1) and Swing components (Java 2 platform). However, the communication between applet and servlet is not so straightforward as clicking a hyperlink or submitting a HTML form. For a GET method, the applet need to URL encode the parameters as name/value pairs and open a URL connection to receive the input stream from servlet and process the data. For a POST method the applet need to open the URL connection and specify the content type of data to be transferred, get input stream and output stream and send data to the servlet (Darby, 1998)

Chad Darby present the way of transmitting serialised Java objects between applets and servlets (Darby, 1998). With this method we can transmit database records as serialised Java objects rather than name/value pairs. So the applet is suitable to build the online database maintenance interface if needed.

JavaBean vs EJB

Although JSP and Java Servlet are key components in Java 2 platform Enterprise Edition (J2EE), our solution is not a typical one that is based on J2EE platform. In the typical J2EE model (Sun MicroSystem, 2001a), the Enterprise JavaBeans (EJB) are used to encapsulate the entire business logic. Whilst in our solution the ordinary JavaBeans are just simple Java components with properties and their access methods, and the major business logic is realised in servlets.

Regarding development, EJB servers provide automatic support of middleware services such as transactions, security, database connectivity, and more (Sun Microsystems, 2001b). So it surely increases the development productivity.

Considering the scale of our project, all the server side Java component, Servlets, JSPs, Beans and JDBC should reside on the same machine. While EJBs support a distributed transactional multitier architecture. Regarding performance, because of its distributed natural, an EJB approach does add overhead.

Java IDE-JBuilder4

We choose Jbuilder 4 as IDE for our system development. JBuilder 4 is a very helpful Java IDE. It has friendly interface, abundant online help information. It provides templates for developing JSP, servlets, Java Bean, applets, etc. It contains Java SDK1.3. It even incorporate a Tomcat container so that the server side components Java servlets and JSPs can be tested and debugged within the IDE.

5.6 Summary

Based on the analysis of different Web techniques, Java components-JSP, Servlet and JavaBean are selected as programming technique for our project. MySQL, as extensively used in Web applications, is chosen as our database server. The next chapter will present the system design and implementation with the selected techniques.

Chapter 6 System design and implementation

6.1 Introduction

In the previous chapter, we discussed the reasons of selecting the Java solution for our project, This chapter presents the design of the system architecture, design of the interface and database structure. The implementation with the Java solution is described in detail. The chapter also introduces some special techniques used in the system.

6.2 Design of the three-tier architecture

The system was implemented with a three-tier client/server architecture. The client tier provides interfaces for various users to interact with the system. It is implemented with JavaServer pages using JavaBeans and presented as interactive, dynamic HTML pages running within a Web browser. The conference logic for paper submission and review, paper management, user management and system configuration resides in the middle tier, the business logic is mainly implemented with Java servlets. The Java servlets on the middle tier communicate with client tier through http protocol and manipulate the backend database through JDBC APIs. MySQL serves as back end tier to store the paper, review and user information. Figure 5-3 in chapter 5 reflects the 3-tier architecture.

6.3 The user interface design

We design the user interfaces for authors, reviewers, committee members and committee chairs. As MySQL AB(MySQL AB 2002) now provides the graphical maintenance interface, we leave the on-line interface for administrator for future version. To be easily customised by PC chair the interfaces will only use HTML form controls.

6.3.1 Login interface

Login interface provides an entry point to the system functions. Every user needs to login in order to perform the tasks with the system. On the login page a user needs to fill in user name and password, and specify the user type to login as, e.g. Reviewer. The system will

check the user against database records, and if authenticated, will present the major interface for the particular user type---*author page*, *reviewer page*, *committee member page* or *committee chair page*, illustrated in figure 6-1.

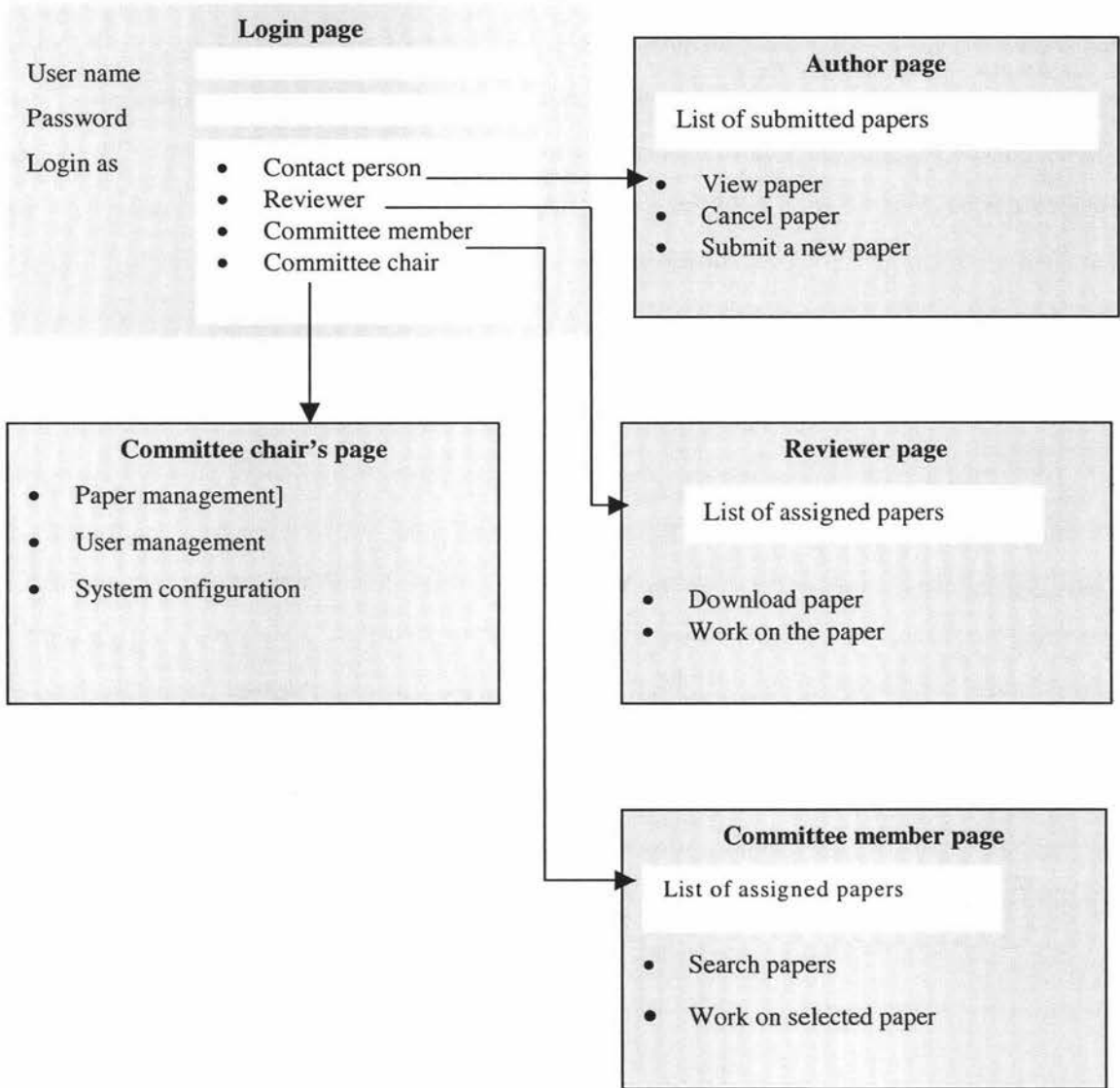


Figure 6-1 Login interface and main pages for different kind of users.

6.3.2 Interface design for authors

On the *Author page* user can see the title list of submitted papers. The user can select one paper and choose to view it or cancel it. If “cancel paper” is chosen, the paper title will be eliminated from the title list. If “view paper” is chosen, the paper’s information is displayed in a new page-*View paper page*. The selection of “submit a new paper” will lead the author to *Paper submission page* followed by the “upload file” page. The

submission will be confirmed with “*Submit paper success*” page. The navigation is described in Figure 6-2. Since the keywords in the system are to be used for matching papers with committee members and matching papers with committee chairs, the keywords should be selected from a predefined list rather than directly typed in.

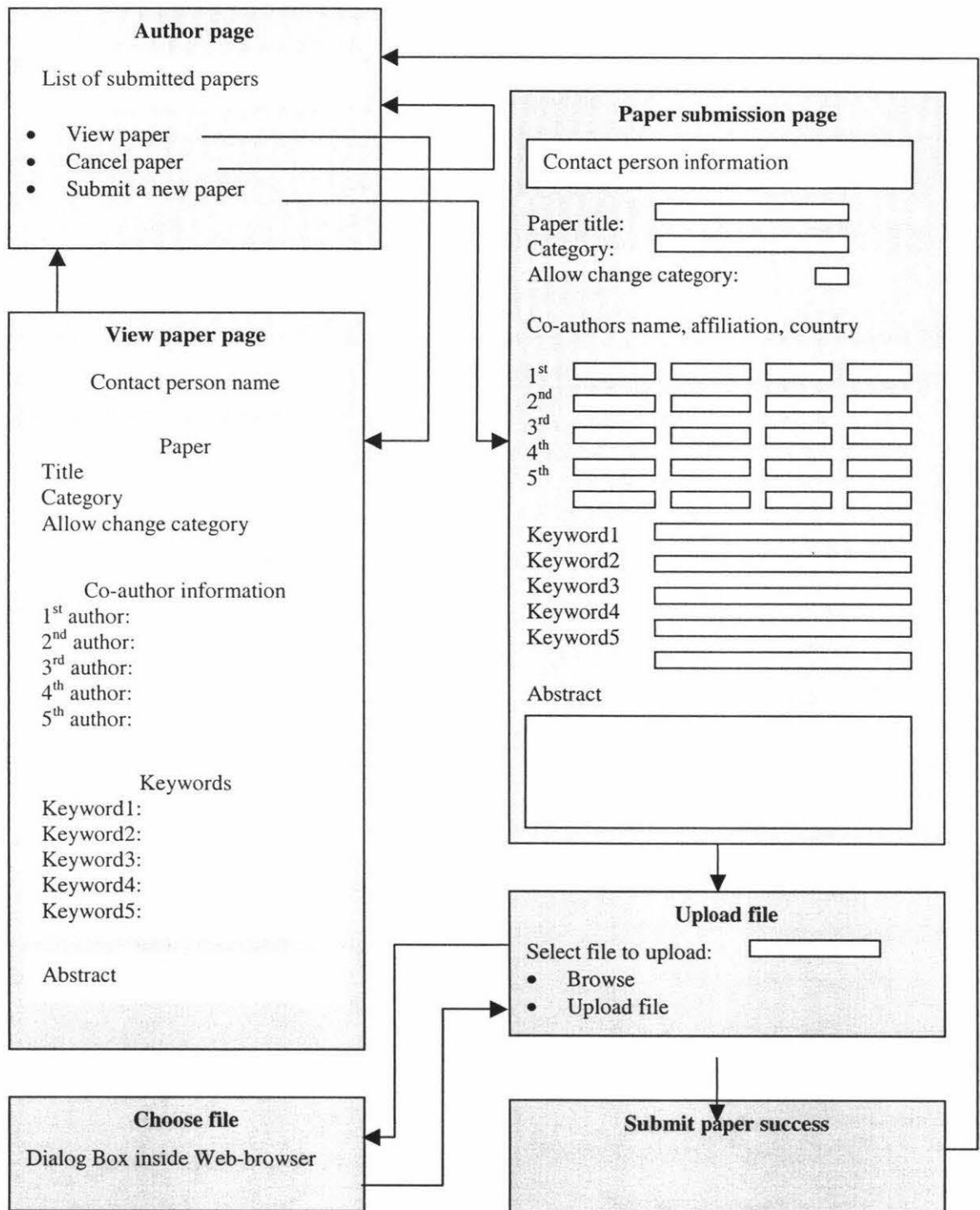


Figure 6-2 Interfaces for authors

6.3.3 Interface design for reviewers

From the *Reviewer page*, reviewers can see the title list of assigned papers. Selecting one paper, a reviewer can choose to download it, this will be followed by *Download file* and *Save file* dialogue boxes invoked by browser. If the reviewer chooses “Work on the paper”, the review form along with the paper information is presented in *Review page*, on which the user can submit, cancel or modify the review. The review submission is confirmed with “*Review confirmation*” page. See Figure 6-3.

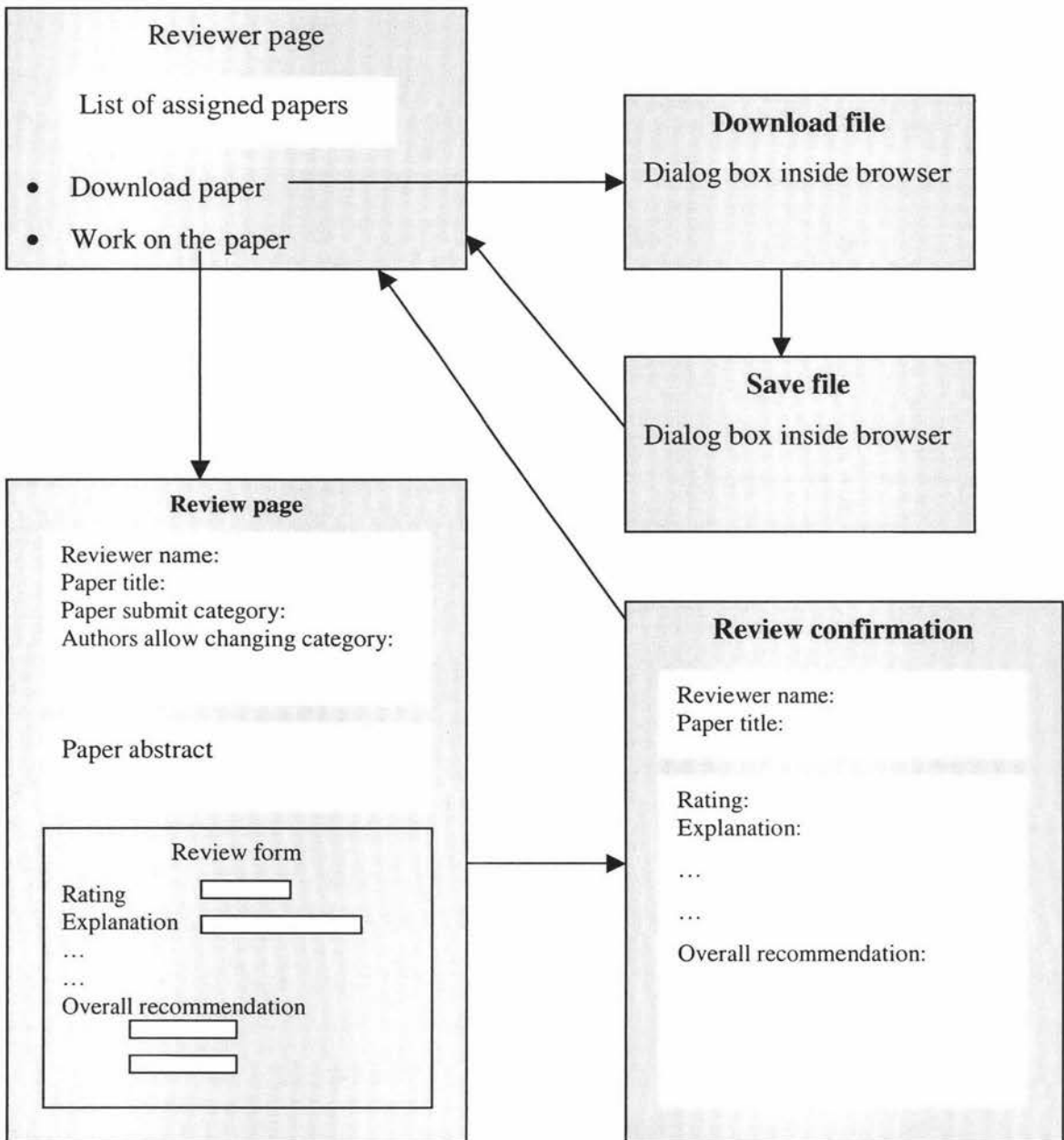


Figure 6-3 Interface for reviewers

6.3.4 Interface design for committee members

On the main page of committee member, a committee member can search papers with various searching criteria including:

- paper number range, keyword in paper title, contact person ID,
- reviewer ID,
- assigned to reviewers status--- fully assigned/not fully assigned,
- review status---completely reviewed/not completely reviewed,
- recommendation status- not made/accept/reject/accept as new category.

The searching result is presented in the paper title list, from which committee member can select one paper and choose to “work on the selected paper”. This will lead to the *Committee member’s paper page*.

The *Committee member’s paper page* is constructed with three sections, paper info, reviewer & review, committee member’s recommendation.

The paper info section includes the paper’s information and provides the “download paper” function.

The reviewer & review section presents the list of keyword-matching reviewers for this paper, sorted by the extent of keyword matching. From the list, the committee member can select one reviewer and choose to “view reviewer’s profile” to display the reviewer’s information in a separate page-*User profile*. Or the committee member can choose to “assign paper to the selected reviewer”, which will result in the addition of a new line in the paper’s review list below.

Below the reviewer part, the reviewer & review section presents the list of reviews for this paper. Each line contains the reviewer ID, “review made” status-true/false. Selecting one line the committee member can choose to “View selected review” so to display the particular review in a separate page of *Review*. Since one line of review happens to

represent an assignment, the committee member can also choose to "Cancel the selected assignment", which will result in the elimination of the line in the review list.

The last section-committee member's recommendation, presents the committee member's recommendation and provides the function of canceling recommendation and making recommendation-accept, reject, accept as new category. With committee member choosing each function, the recommendation accordingly changes to "not done", "accept", "reject" and "accept as new category". If the author specifies "not to allow the committee member to change the category", the new category selection and "accept as new category" function are disabled. Figure 6-4 illustrates the interface and navigation for committee member.

6.3.5 Interface design for Program Committee Chair

The main page for committee chair is *Committee chair 's page*. It provides three functions, "user management", "set up system parameters" and "paper management", each leads to a sub screen.

6.3.5.1 Set up system parameters

This page presents the system parameters:

- timelines,
 - ◆ for paper submission
 - ◆ for review submission
 - ◆ for making recommendation
 - ◆ for making decision
 - ◆ for publishing selected papers (if applicable)
- conference info,
 - ◆ conference name
 - ◆ maximum number of papers per reviewer
 - ◆ minimum number of reviews per paper
 - ◆ root directory for uploading files

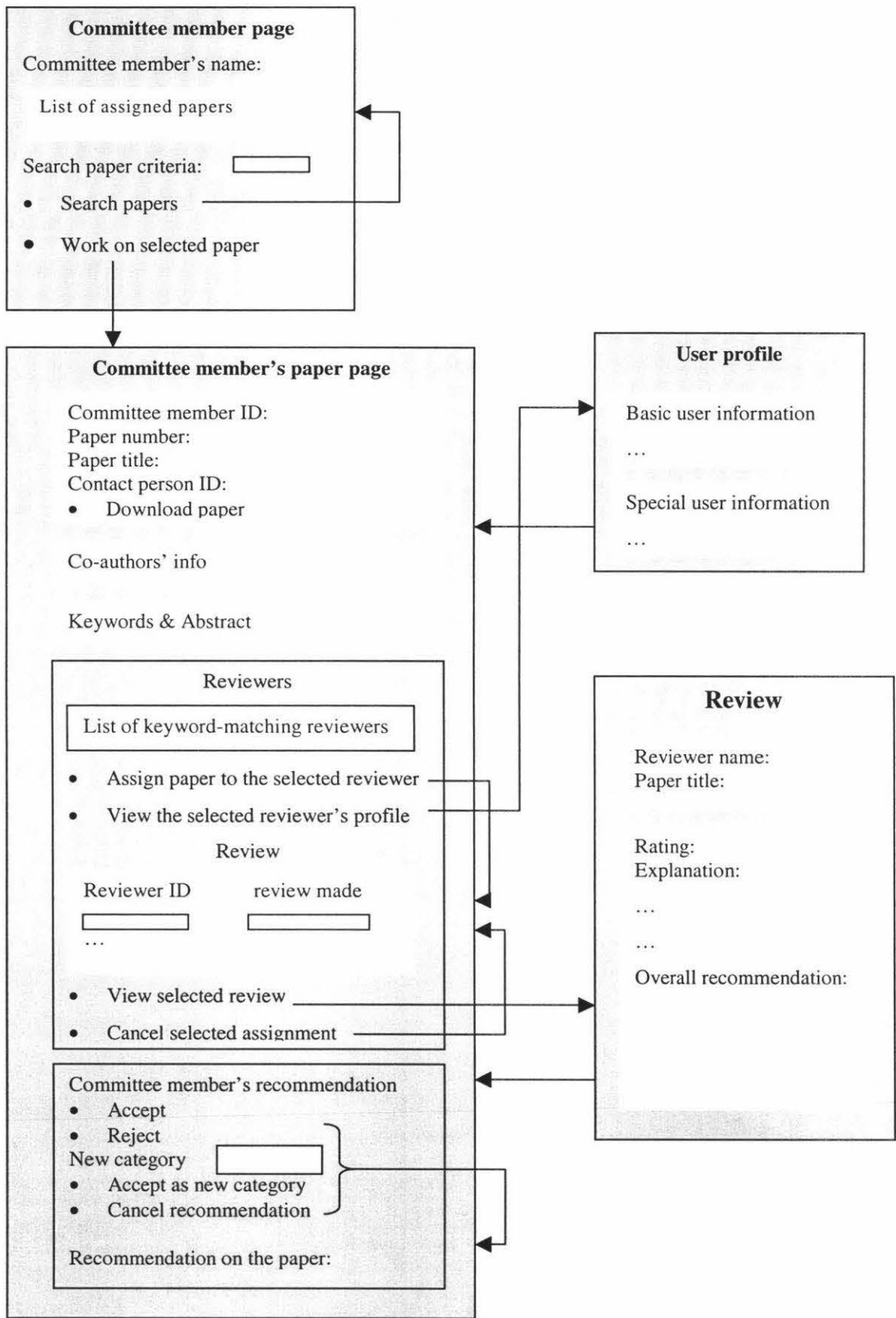


Figure 6-4 Interface and navigation for committee member

- keywords, that reflect the subject areas of the conference, can be selected by authors to describe their papers and be selected by reviewers and committee members to specify their expertise and preference.

The committee chair not only can view the current system parameters, but also can edit and update these parameters, it enables the online customisation of the system.

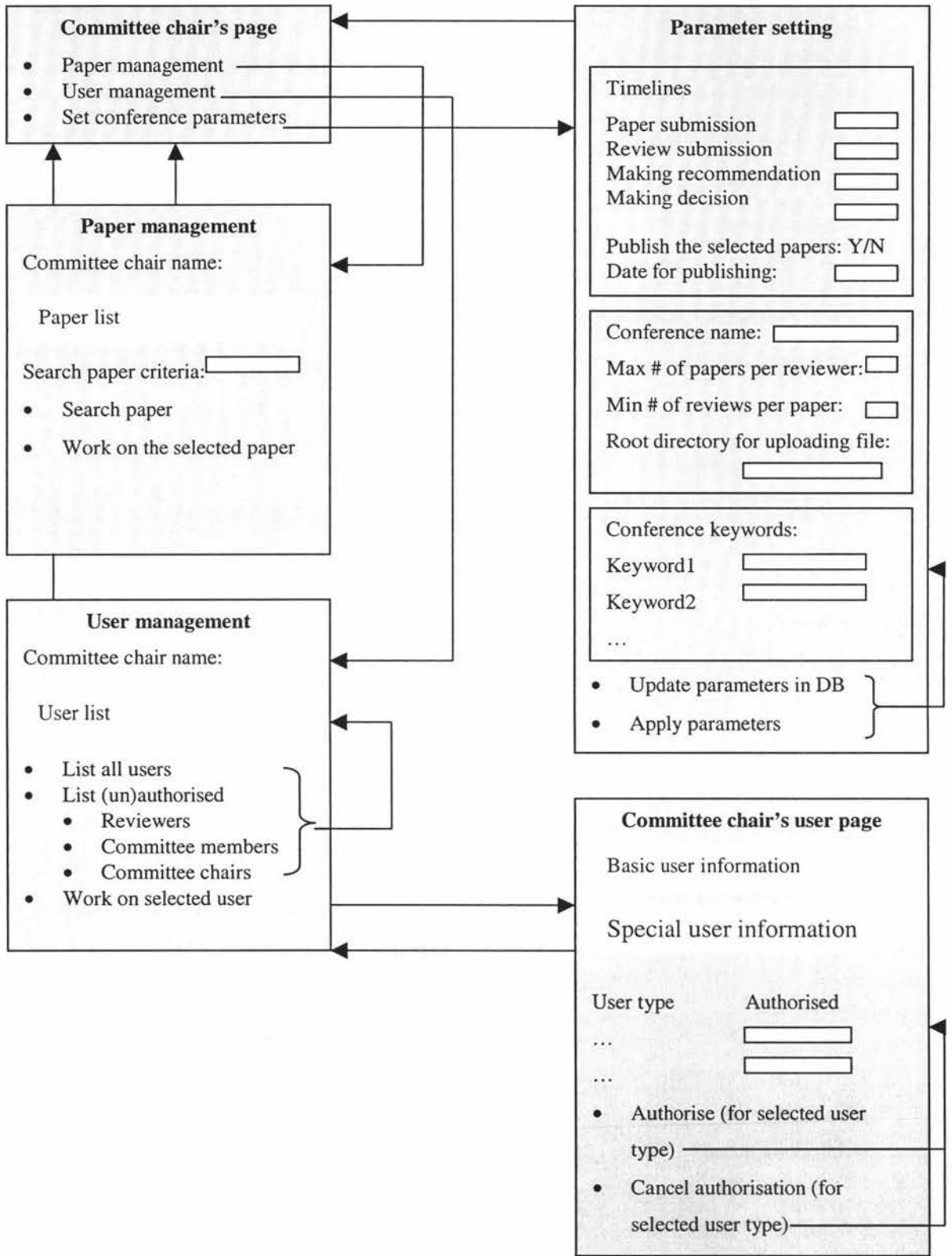


Figure 6-5 Interface for committee chair

6.3.5.2 User management

The *User Management* page displays a list of users, committee chair can choose to list “all users”, “(un)authorised reviewers”, “(un)authorised PC members”, or (un)authorised PC chairs. From the list, committee chair can select a user and choose to “work on the selected user”, then the user’s profile and all the registered user types and their authorisation status are displayed in another page-*Committee chair’s user page*.

On the *Committee chair’s user page*, the committee chair can authorise the user for a specific user type or cancel the authorisation for it.

The design of *committee chair’s page*, *Set up system parameters page* and *User management page* are shown in figure 6-5.

6.3.5.3 Paper management

The *paper management* page is similar to the *committee member’s page* (see Figure6-4). The committee chair can list the papers meeting the criteria entered in the form, in addition to the criteria listed in 6.3.4, there is another criterion, “decision status”, which can be “not made”, “accept”, or “reject”.

From the list of papers, the committee chair can select one and choose to “work on the paper”.

This will invoke the *Committee chair’s paper page*, which is similar to the Committee member’s paper page in Figure 6-4 but has some significant differences:

- The “Assign paper to reviewer” function changes to “Assign paper to reviewer over limit”. Referring to section 6.3.5.1, one system parameter is “maximum number of papers per reviewer”, when committee member assigns a paper to reviewer, the system will check if the reviewer has been assigned with papers more than this number, if so, the system decline the assignment and give an error message. But when the committee chair assign a paper to a reviewer, there isn’t such limit.
- The “Committee member” section takes the place of “Committee member’s recommendation” section in Figure 6-4. In the “Committee member” section the

committee chair can see the list of keyword-matching committee members sorted by the matching extent. The committee chair can choose to view the selected committee member's profile or assign the paper to this committee member. Once assigned, the committee member's ID is shown in this section.

- Additional section "Decision" presents the current decision status and enables the committee chair to make or cancel decision on the paper. After the final decision is made, the committee chair can choose to "notify author of the final decision", this will trigger an automatic email sent to the contact author referring the final result.

The paper management interface and Committee chair's paper page are illustrated in Figure 6-6.

6.3.6 Interface design for user registration and modifying profile

The *Registration page* and *Modify profile page* provide a uniform user registration interface for contact person, reviewer, committee member and committee chair. One user only registers his/her personal information once even if he/she plays more than one role in the system. One user uses the same user name and password for logging in as any authorised user type.

The *Registration page* enables a new user to select user name and password, and input the contact information (Basic user information) and the expertise information (special user information). If an existing user (already registered at least one user type) wants to register a new user type he/she should provide user name and password and use the function "Get my account information" to fill the form with existing user information.

Once user submits the registration form, the system will check if it's a new user or an existing user. For a new user, the user name and the email address are checked to avoid duplicate user name or duplicate email address in the database. For an existing user, the user name can't be changed, the email address is checked against duplication.

Modify profile page can only be accessed after user logging in, on this page user can only change the contact information and expertise information, while the register type, user

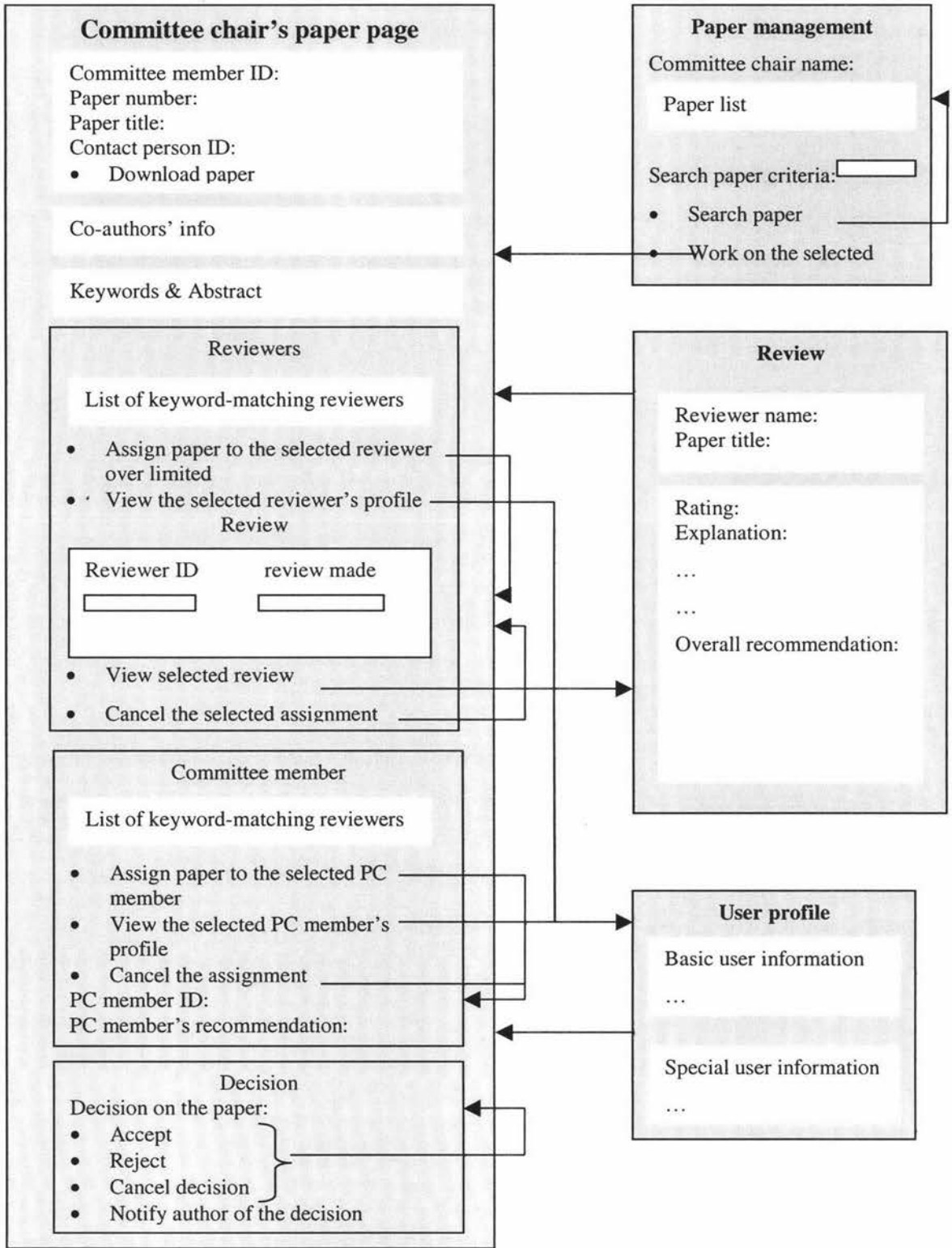


Figure 6-6 Paper management interface for committee chair

name, and password can not be changed. *Change password* page is for user to change password separately. All the main user pages-*Author page*, *Reviewer page*, *Committee*

member page and Committee chair page, provide links to Modify profile page and Change password page.

The interface regarding user registration and user profile is described in Figure 6-7.

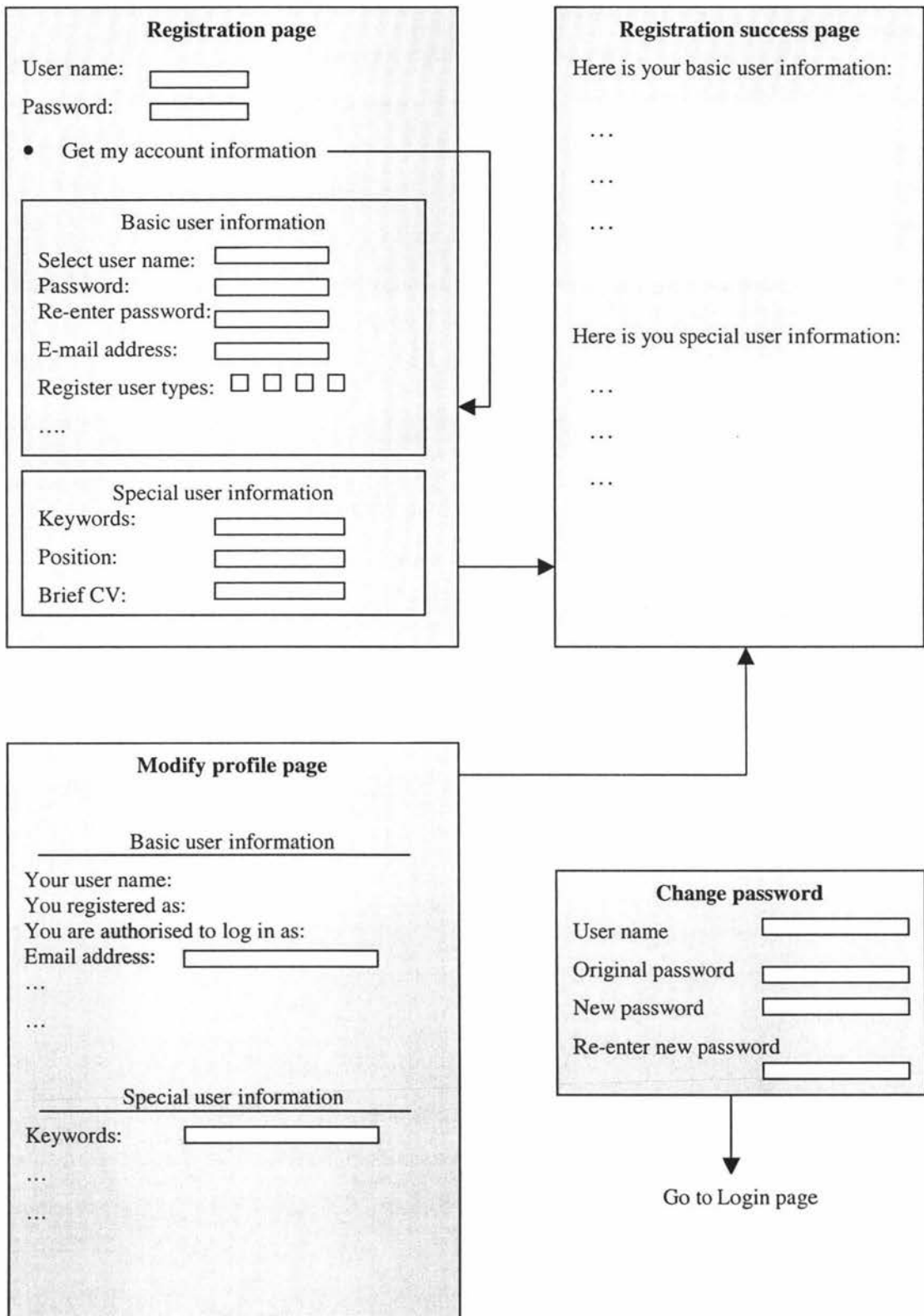


Figure 6-7 Interface for user registration and modifying profile

6.4 The data storage and database design

In general, what we need to store in the system are paper- related information, user-related information and some system parameters.

Paper-related information

Paper-related information includes the file uploaded by the author and the paper information submitted by the author. Reviewers' reviews on the paper, committee member's recommendation on the paper and program chair's decision on the paper should also be considered as paper related information.

The uploaded files are saved in the directory structure on hard drive, the root directory for a conference's papers is specified as a system parameter edited by program chair like "d:/conference/". Under the root directory each paper contact person has a subdirectory named in his/her user name, all the papers that are submitted by this contact person are saved under this subdirectory (One contact person may submit more than one paper). In our design, the paper information and file are submitted simultaneously. On reception of a paper, system generates an auto-incremented integer as the paper's unique ID, and changes the original file name of the uploaded file to "P"+ID+original extension, then saves it under the paper's contact person's sub directory (if not existed, create one).

For example, the original filename is *document1.doc*, the paper ID is 12, then, the new file name will be *P12.doc*. Keeping the filename extension enables the committee members and reviewers to open the file with suitable editor. An example of file directory structure is shown in Figure 6-8. To associate the file with paper information, the full path of the file is written back into the paper's record in database as value of field *URL*.

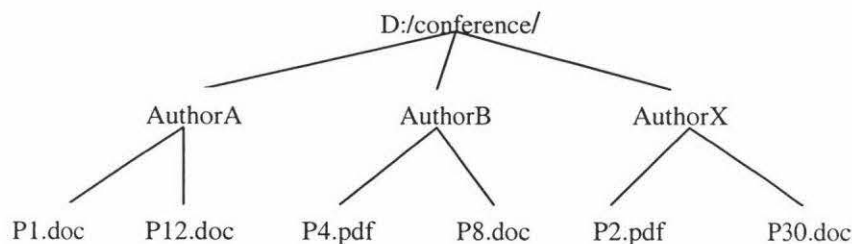


Figure 6-8 directory structure for storing uploaded files

One paper should be assigned to only one committee member, and the committee member's recommendation and committee chair's decision take the form of a text string like "Accept", so the Committee member's ID, recommendation, and decision are stored together with paper information.

One paper has up to five co-authors, one author may be a co-author of more than one papers, so we store the authors information independent of papers and use a "paper-author" entity to associate them.

One paper requires several reviews, one reviewer may be assigned to several papers so reviewer information is stored independent of papers, the entity "review" associates them. Here we see that a review represents a reviewer-to-paper assignment.

User related information

Every user should have their contact information stored in the system, reviewers, committee members and committee chairs should also have expertise information to store. To avoid user entering user information for every user types that he/she wants to register as, the entity "Contact Person" will keep both "basic user information" (contact information) and "special user information" (expertise information) for each user.

For each special type of users (reviewers, committee members, and committee chairs) there should be authorisation status to keep. So we define entity "Reviewer", "PCMember", and "PCChair" to store the users' ID and their authorisation status.

System parameters

- **Keywords**

Keywords are text strings probably with up to 20 English words each. On the user interface the strings form the list for user to select when submitting papers or registering expertise. In paper assignment keywords are compared for matching the appropriate reviewers or committee members. Comparing long text strings can cause serious inefficiency in system performance, so the entity “Keyword” is designed to represent “keyword ID” and “keyword”. “Keyword” is the text string that the users see, “keyword ID” is an integer that the users actually select and the programs deal with.

- **Other parameters**

Conference timelines, conference name, file root directory, and some numeral limits are stored as entity “Parameter”, there is only one record of parameter for one conference.

Based on the above analysis, the Entity Relationship Model of the system is designed in Figure6-9. The model is to be implemented in My SQL database, the tables and fields defined in the database are shown in Table 6-1

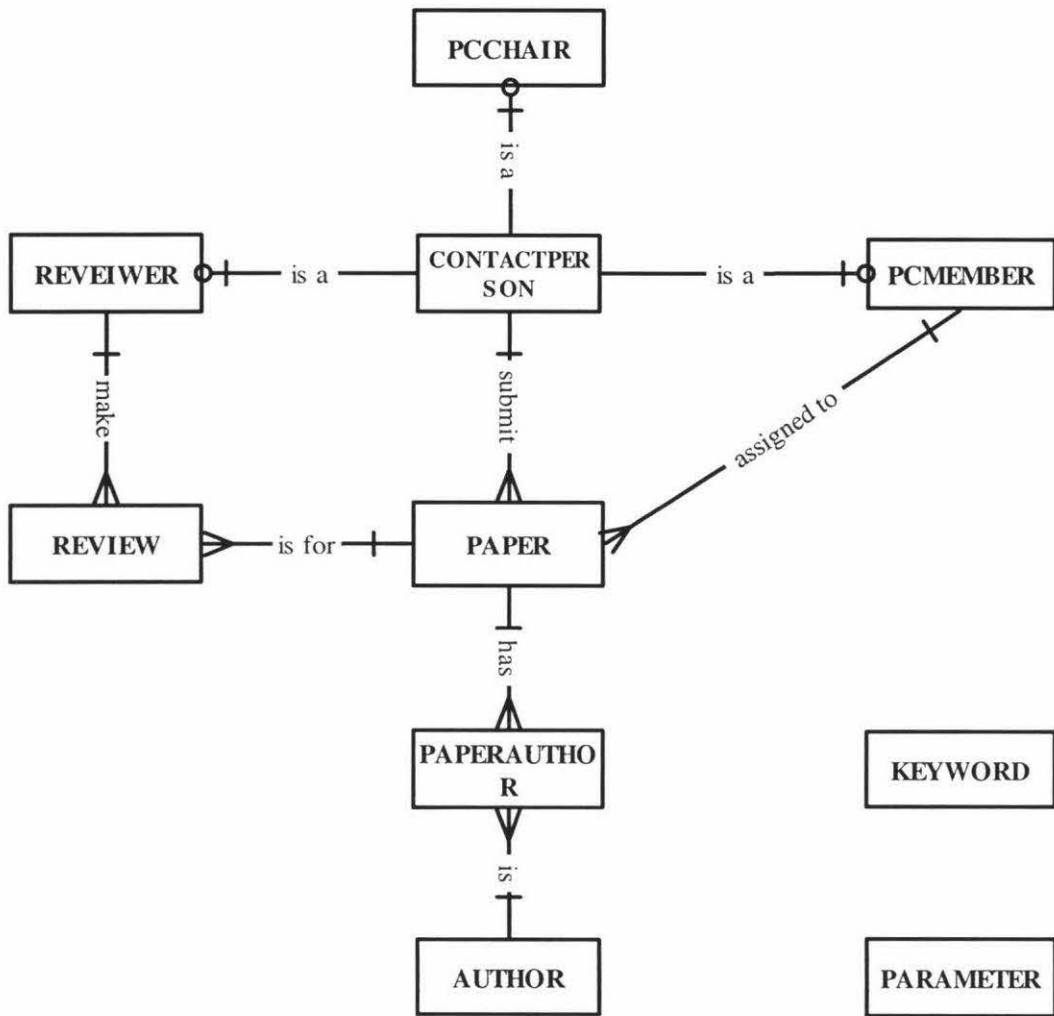


Figure 6-9 Entity Relationship Model

Table 6-1: database structure design

Table	Field	Type	Constraint	Null	Default
CONTACT PERSON	UserID	Varchar(20)	Primary key		
	Password	Varchar(20)			
	FirtstName	Varchar(20)			
	LastName	Varchar(20)			
	Title	Enum("Mr.", "Ms.")		Yes	Null
	Email	Varchar(50)			
	Affiliation	Varchar(50)			
	Address1	Varchar(50)		Yes	Null
	Address2	Varchar(50)		Yes	Null
	City	Varchar(50)			
	State-province	Varchar(50)		Yes	Null
	Country	Varchar(50)			
	Zip-postal-code	Varchar(20)		Yes	Null
	Work -phone	Varchar(20)			
	FAX	Varchar(20)		Yes	Null
	Position	Varchar(50)			
	Keyword1	Varchar(100)			
	Keyword2	Varchar(100)		Yes	"Not selected"
Keyword3	Varchar(100)	Yes	"Not selected"		
Keyword4	Varchar(100)	Yes	"Not selected"		
Keyword5	Varchar(100)	Yes	"Not selected"		
BriefCV	Text	Yes	Null		
AUTHOR	AuthorNo	Smallint(6)	Primary key		
	FirstName	Varchar(20)			
	LastName	Varchar(20)			
	Title	Enum(Mr,Ms)		Yes	Null
	Affiliation	Varchar(50)			
	Country	Varchar(50)			

Table	Field	Type	Constraint	Null	Default
PAPER	PaperNo Title Abstract URL Contact person ID Category AllowChangeCategory Keyword1 Keyword2 Keyword3 Keyword4 Keyword5 PCMemID Recommendation Comment Decision	Smallint(6) auto-increment Varchar(100) Text Varchar(100) Varchar(20) Enum("FULLPAPER", "SHORTPAPER", "POSTER") Enum("TRUE", "FALSE") Varchar(100) Varchar(100) Varchar(100) Varchar(100) VarChar(20) Enum("NOTDONE", "ACCEPT", "REJECT", "ACCEPTASNEWCATEGORY") text enum("NOTDONE", "ACCEPT", "REJECT")	Primary key Foreign key	 Yes Yes Yes Yes Yes Yes	 Null "NOTDONE"
PAPER AUTHOR	PaperNo AuthorNo Rank	Smallint(6) Smallint(6) Tinyint(3)	Primary key Primary key	 Yes	 Null
REVIEWER	UserID Authorised	Varchar(20) Enum("TRUE", "FALSE")		 Yes	 Null
PC MEMBER	UserID Authorised	Varchar(20) Enum("TRUE", "FALSE")		 Yes	 Null
PC CHAIR	UserID Authorised	Varchar(20) Enum("TRUE", "FALSE")		 Yes	 Null

Table	Field	Type	Constraint	Null	Default
REVIEW	PaperNo	Smallint(6)	Primary key		
	ReviewerID	Varchar(20)	Primary key		
	RelevantLevel	Enum("Not relevant", "Moderately relevant", "Very relevant")		Yes	Null
	RelevantExplain	Text			
	SignificantLevel	Enum("Not sinificant", "Moderately significant", "Very significant")		Yes Yes	Null Null
	SignificantExplain	Text			
	OriginalLevel	Enum("Not original", "Moderately original", "Very original")		Yes Yes	Null Null
	OriginalExplain	Text			
	CommentOnQuality	Text		Yes	Null
	CommentOnClarity	Text		Yes	Null
CommentOnAppropriateness	Enum("ACCEPT", "CHANGECATEGORY", "REJECT")		Yes	Null	
FurtherComment	Text		Yes	Null	
Recommend	Enum("FULLPAPER", "SHORTPAPER", "POSTER")				
ChangeToCategory	Enum("Not confident", "Moderately confident", "Very confident")		Yes	Null	
Confident	Text		Yes	Null	
RecommendationRevision			Yes	Null	
KEYWORD	KeywordID	Tinyint(3)	Primary key		
	Keyword	Varchar(100)			
PARAMETER	ConferenceName	Varchar(100)	Primary key		
	PaperSubmitDate	Date			
	ReviewSubmitDate	Date			
	RecommendationMadeDate	Date			
	DecisionMadeDate	Date			
	PublishSelectedPapers	Enum("TRUE", "FALSE")			
	PublishDate	Date			
	MaxNumOfPaperForOneReviewr	Tinyint(3)			
NumOfReviewsForOnePaper	Tinyint(3)				

Table	Field	Type	Constraint	Null	Default
	FileRootDirectory	Varchar(100)			

6.5 Implementation details of system architecture

JavaServer pages are used to generate the presentation layer, and servlets to perform process-intensive tasks. The servlets act as the controllers and are in charge of the request processing, creation of any beans used by the presentation JSP, instantiating the beans with dynamic data received from the request or retrieved from the database, deciding which JSP to forward the request to. The programming model is shown in Figure 6-10. This model falls into the category of Model-view design pattern.

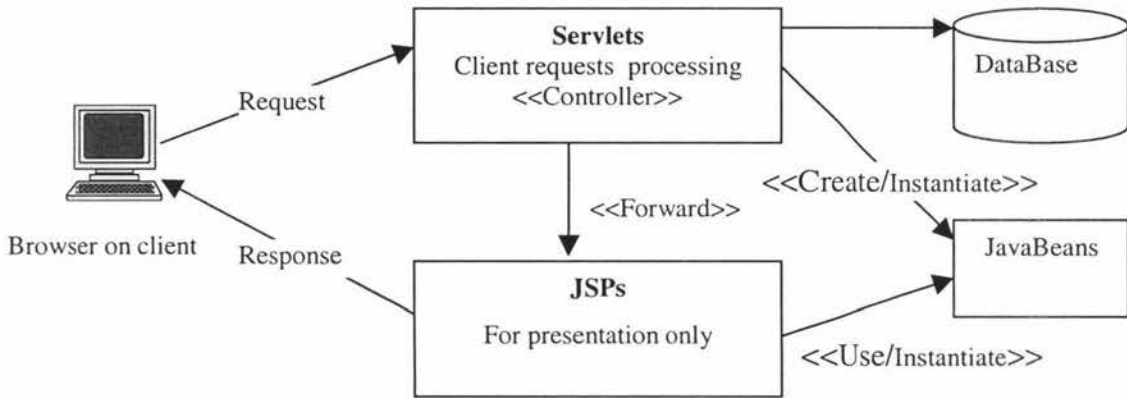


Figure 6-10 programming model with Java component

6.5.1 Communication between browser and servlets

An http request can be sent from browser to servlet through Internet by clicking a hyperlink like

```

<A href="/conference/servlet/
PaperManager?formname=workOnPaper&subscript=2">Work on paper</A>

```

or by submitting a HTML form like

```

<form method=post action="/conference/servlet/LoginHandler">
User name<input type=text name="username"><br>
Password <input type=password name="password"><br>
<input type=hidden name="formname" value="login">
<input type=submit name="Submit" value="Login">

```

</form>

The web server receives the request and passes it to servlet container (Tomcat), which invokes a thread of the servlet “Paper Manager” in the hyperlink example, or “LoginHandler” in the HTML form example, and passes the request to the servlet as a Java object *HttpServletRequest* (Coward, 2000, p32). The *HttpServletRequest* object provides method of *getParameter()* to extract the name-value pairs submitted along with the request.

For servlets to identify the request so to execute the corresponding processing, a parameter “formname” is defined to “name” each request. In case in a HTML form which has more than one “Submit” buttons the parameter “Submit” is also checked to identify the request.

6.5.2 Request processing in servlets

Generally, when a servlet is invoked with a request, it will extract the parameters, authenticate the user. Then, depending on the authentication result and the “name” and parameters of the request, the servlet will execute the corresponding piece of request processing code and deciding which JSP file should be presented to the browser. The data produced for output is put in the Javabeans associated with the JSP file. To enable the JSP file utilising the beans, servlet adds the bean(s) as an attribute to the *HttpSession* object, which is obtained from *HttpServletRequest* object with the *getSession()* method, then “Forward “ the request to the JSP file. The forwarding is actually performed by *RequestDispatcher* (Coward, 2000, p55) object. This will result in the HTML page derived from the JSP file and its dynamic data in the bean(s) to be sent to the browser.

Session tracking and user authentication

The *HttpSession* object is designed for session tracking. A session associates a series of different requests from a particular client (Coward, 2000, p48). The attributes added to the *HttpSession* object can be shared by all the servlets or JSPs invoked in the session. Taking advantage of this characteristic, the JavaBeans can be shared among the processing servlets and the presentation JSPs. Even further we can do the user authentication with *HttpSession*. Since each user should enter the system through Login page (actually controlled by servlet LoginHandler), LoginHandler will check the user

name, password and login type against user account in the database. If the user and login type is authenticated, LoginHandler will create a ContactPerson object containing the user account, add it to HttpSession as an attribute called "CurentUser" and also add the login type as an attribute called "LoginType". Other servlets invoked by subsequent requests in the session will check the "CurrentUser" and "LoginType", if the "CurrentUser" not null and "LoginType" has the privilege to execute the requested function the processing will continue, otherwise the user is redirected to the Login Page.

6.5.3 JSP and JavaBeans implementing the user interfaces

As mentioned in section 4.5, JSP can apply JavaBean to solely serve as a dynamic and interactive web page. Here we explain how JSPs use JavaBeans to include the dynamic data.

6.5.3.1 Locating or creating a JavaBean from JSP page with UseBean Tag

```
<jsp:useBean id="_loginJSPBean" class="LoginJSPBean" scope="session"/>
```

With this tag in JSP, the program compiled from the JSP first looks for an instance of the bean in the "session" scope. Recall that in last section, the servlet put the bean in HttpSession object as an attribute. Th program can find the instance of the bean if only the name of the attribute added by the servlet is the same as the id defined in the tag. In case the bean can't be found, the program will create one and add it to the "session" scope. The bean is referred with "id" within the page from this point forward.

6.5.3.2 Representing HTML form controls with data form Java Beans

Text Controls

JSP tag *GetProperty* can be used to display dynamic data in a text control in HTML form:

```
User name<input type=text name="username" value=<jsp:getProperty name="_loginJSPBean"
property="username"/>>
```

With the “getProperty” tag the value of the attribute “username“ in the bean is presented in the input box of the text control. The “getProperty” implicitly calls the “getter” method of the attribute in the bean.

It’s not so straightforward to display dynamic data in check box, radio button and pull-down menu, we have to include some code in the JSP file to realise this:

Check Boxes

```
<INPUT TYPE="CHECKBOX" NAME="registerType" value="CONTACTPERSON"
  <%=_registerJSPBean.isCbSelected("CONTACTPERSON")%>>Contact Person
<INPUT TYPE="CHECKBOX" NAME="registerType" value="REVIEWER"
  <%=_registerJSPBean.isCbSelected("REVIEWER")%>>Reviewer
<INPUT TYPE="CHECKBOX" NAME="registerType" value="PCMEMBER"
  <%=_registerJSPBean.isCbSelected("PCMEMBER")%>>Commitee member
<INPUT TYPE="CHECKBOX" NAME="registerType" value="PCCHAIR"
  <%=_registerJSPBean.isCbSelected("PCCHAIR")%>>Committee chair
```

Since Checkboxes have to be represented as a string array in the bean, the *registerType* is defined as an string array in the *_registerJSPBean*. The method *isCbSelected(String)* defined in the bean checks if the *String* parameter appears in the *registerType* array, if so, it returns the string "checked" , otherwise returns nothing. The check boxes with “checked” attribute are checked in HTML form (Powell, 1998, p423). The JSP tag “<%=expression %>” calculates the “expression” and put the result in the tag’s position.

Radio Buttons

```
<INPUT TYPE="Radio" NAME="recommend" VALUE="Accept"
<%=_review.isRadiochecked(_review.getRecmmd(),"Accept") %>>Accept <BR>
<INPUT TYPE="Radio" NAME="recommend" VALUE="Reject"
<%=_review.isRadiochecked(_review.getRecmmd(),"Reject") %>>Reject<BR>
<INPUT TYPE="Radio" NAME="recommend" VALUE="ChangeCategory"
<%=_review.isRadiochecked(_review.getRecmmd(),"ChangeCategory") %>
  >Accept but change category<BR>
```

recmmd is a property in the bean *_review*, the method *isRadiochecked (String,String)* returns “checked” if its two parameters are same. The radio button with attribute “checked” is preselected in HTML form (Powell, 1998, p425).

Pull-Down Menu

```
<SELECT NAME="changeToCategory"
  <OPTION>
  <OPTION <%=_review.isListSelected(_review.getChgToCtgy(),"FULLPAPER" )%>>FULLPAPER
  <OPTION <%=_review.isListSelected(_review.getChgToCtgy(),"SHORTPAPER" )%>>SHORTPAPER
  <OPTION <%=_review.isListSelected(_review.getChgToCtgy(),"POSTER" )%>>POSTER
</SELECT>
```

chgToCtgy is a property in the bean *_review*, the method *isListSelected(String,String)* returns “selected” if its two parameters are same. An occurrence of the attribute **selected** in the `<option>` element sets the form control to select this item by default (Powell, 1998, p419).

6.5.4 Cooperation of Java components in implementing the system function

The java components design is discussed in groups of system functions implementation, the Figures illustrate the message flow between components, tables explained the event(request) processing.

Based on the programming model in Figure 6-10, we designed 26 JSP pages implementing the user interface, 12 servlets as the working process controllers, 23 Java beans and other Java classes to model the dynamic data for the interface and business entities. In this section we describe these Java components in categories of functional groups. For sake of clarity, the creation and calling of Javabeans from the servlets are omitted in the diagrams. When describing the control logic of servlets, the user authentication is omitted since they are all the same as described in section 6.5.2.

Figure 6-11 illustrates the components participating in the processing of login/logout as well as the emailmyaccount requests. Table 6-2 elaborates the detailed action of the controlling servlet

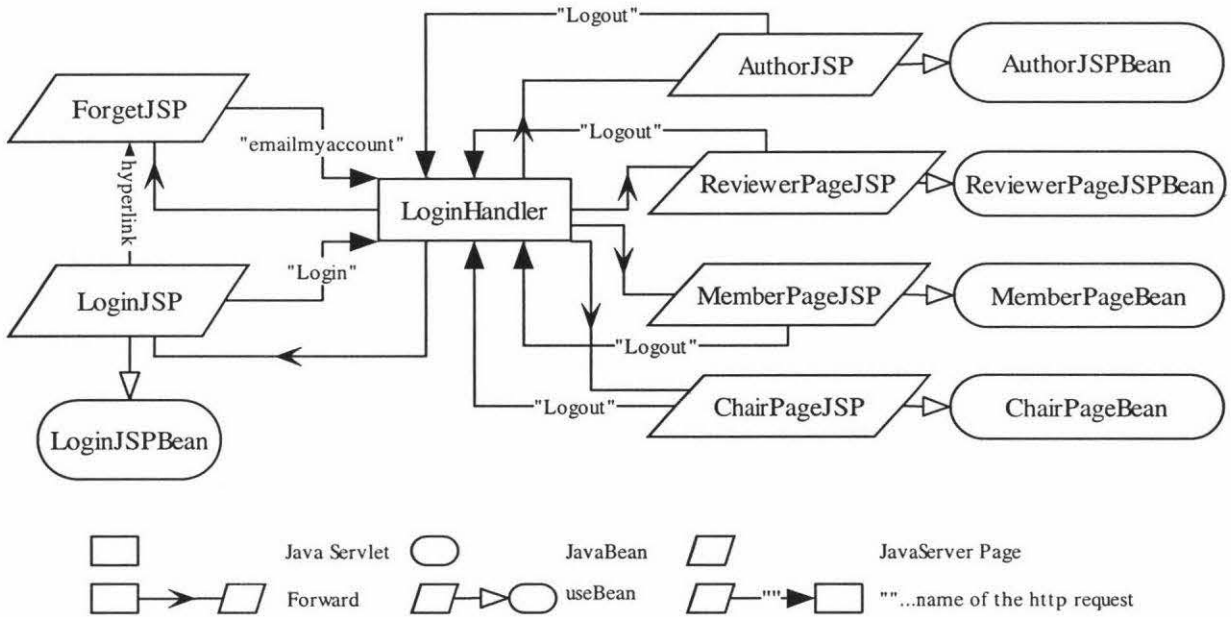


Figure 6-11 Java components involved in user Login and Logout

Table 6-2

Request	Controller	Action
Login	LoginHandler	<p>If login fails error message is added in Login JSPBean and request is forwarded to LoginJSP page. Otherwise “CurrentUser” and “Current UserType” are put in <i>session</i> for subsequent user authentication. Depending on particular user type, preparation is made and request is forwarded to corresponding user page:</p> <p>For login as “Contact person”, all papers submitted by this user are retrieved are put in the array of Paper object in AuthorJSPBean as well as the current user’s account information, forward the request to authorJSP.</p> <p>For login as “Reviewer”, all the reviews made or to be made by the user are retrieved and put in the array of Review object in ReviewerPageBean as well as the current user’s account information, forward the request to ReviewerPageJSP.</p> <p>For login as “PC member”, put the current user’s account information in MemberPageBean, forward the request to MemberPageJSP.</p> <p>For login as “PC chair”, put the current user’s account information in ChairPageBean, forward the request to ChairPageJSP</p>
Emailmyaccount	LoginHandler	Check the submitted e-mail in database, if exist, send the user name and password to this e-mail addresses.
Logout	LoginHandler	The current user session is explicitly terminated, request is forwarded to LoginJSP page.

Figure 6-12 illustrates the components and message flow in user account related processing. Table 6-3 explains the controllers behavior in details.

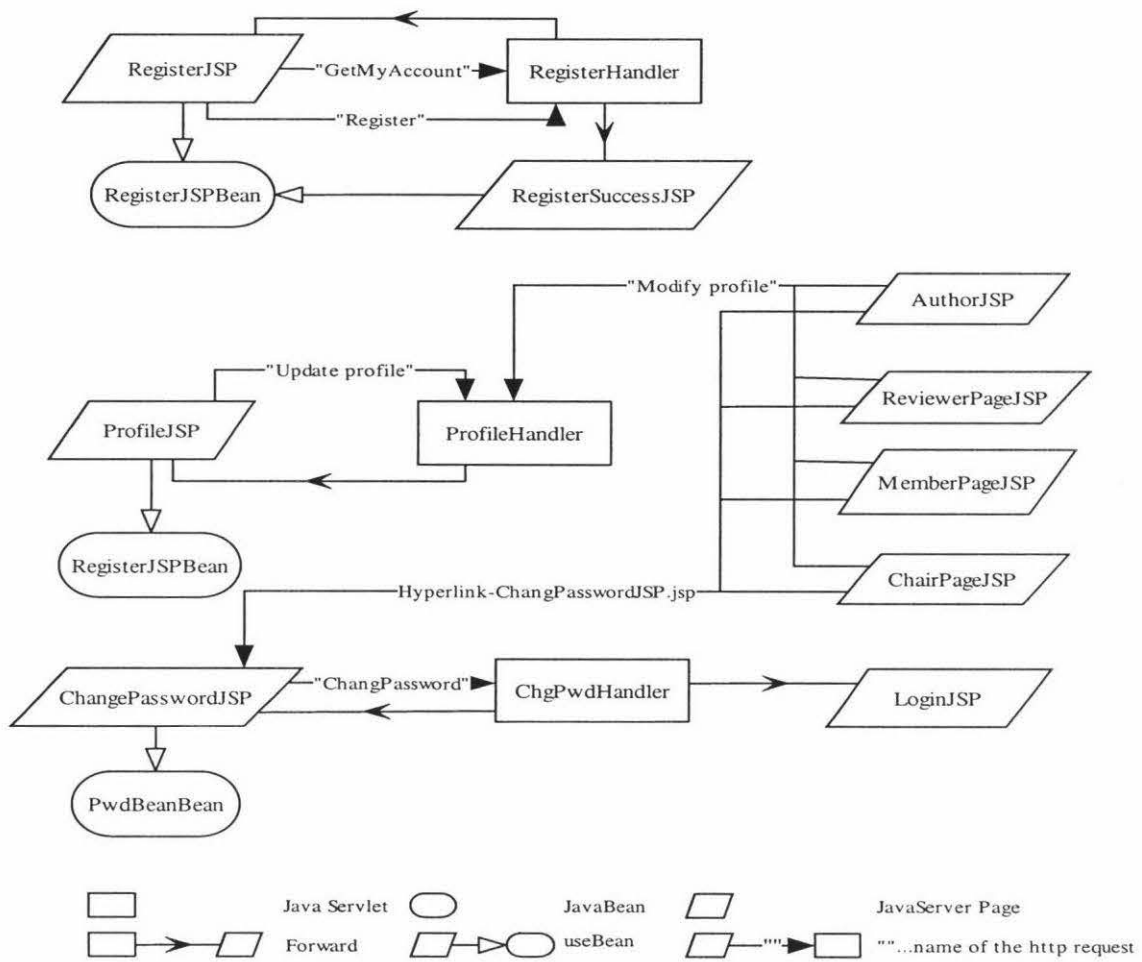


Figure 6-12 Components involved in registration and profile modification

Table 6-3

Request	Controller	Action
GetMyAccount	RegisterHandler	If password and user name is authenticated, user account information is retrieved and put in RegisterJSPBean, request is forwarded to registerJSP.
Register	RegisterHandler	If it's a new user, the submitted user name and email are checked for duplication in the database record, if not duplicate User record is created. If it's an existing user ("Register" request after "GetMyAccount" request) update the user record. For all the user types registered, new record is created in corresponding tables (Reviewer, PCMEBER, PCCHAIR) Request is forwarded to RegisterSuccessJSP.
Modify profile	ProfileHandler	Current user's account information is put in RegisterJSPBean, request is forwarded to ProfileJSP.
Update profile	ProfileHandler	User record is updated in database, request is forwarded to ProfileJSP.
ChangePassword	ChgPwdHandler	If user name and original password are authenticated, new password is recorded in database. Request is forwarded to LoginJSP.

Figure 6-13 shows the components and message flow between them in servicing the contact author's operations. Table 6-4 explains the servlets' processing in details.

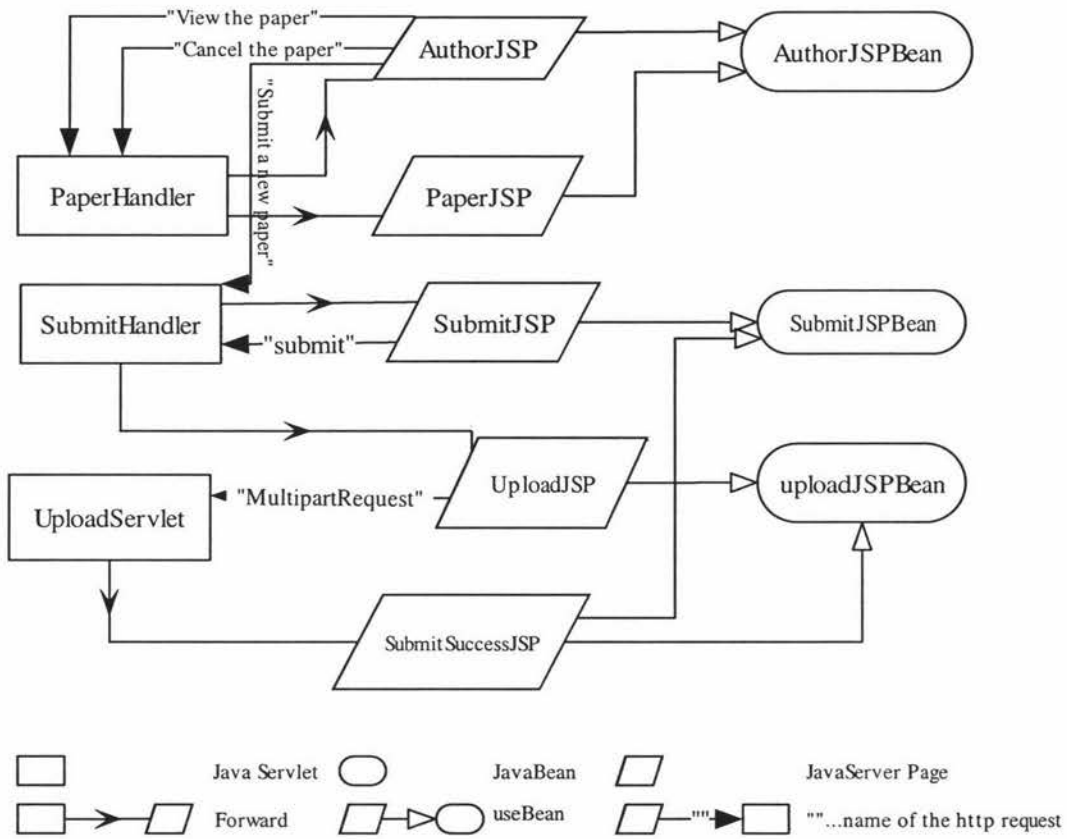


Figure 6-13 components involved in contact author's operation

Table 6-4

Request	Controller	Action
View the paper	PaperHandler	The selected paper's subscript in the paper object array in AuthorJSPBean is put in AuthorJSPBean. PaperJSP page is presented.
Cancel the paper	PaperHandler	The record of the selected paper in database is deleted, the records of its reviews are also deleted.
Submit a new paper	SubmitHandler	The current user's information is put in the SubmitJSPBean and SubmitJSP page is presented in browser.
Submit	SubmitHandler	A new paper record is created and the submitted paper's information is saved in database. The name and path of the target file to be uploaded is generated and put in uploadJSP Bean, the UploadJSP page is presented
MultipartRequest	UploadServlet	The file is uploaded as multipart requests to the designated path. The URL field is updated in the paper's record. Source file name, paper's URL and paper length in bytes is written into UploadJSPBean, SubmitSuccessJSP is presented.

Figure 6-14 illustrates the components serving reviewer's operation, table 6-5 describes the servlets' action in details.

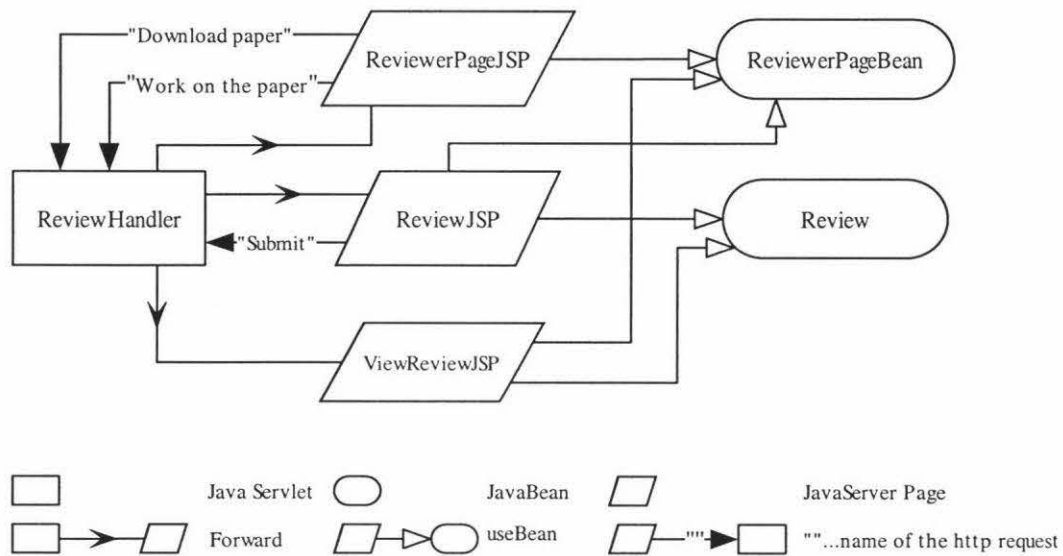
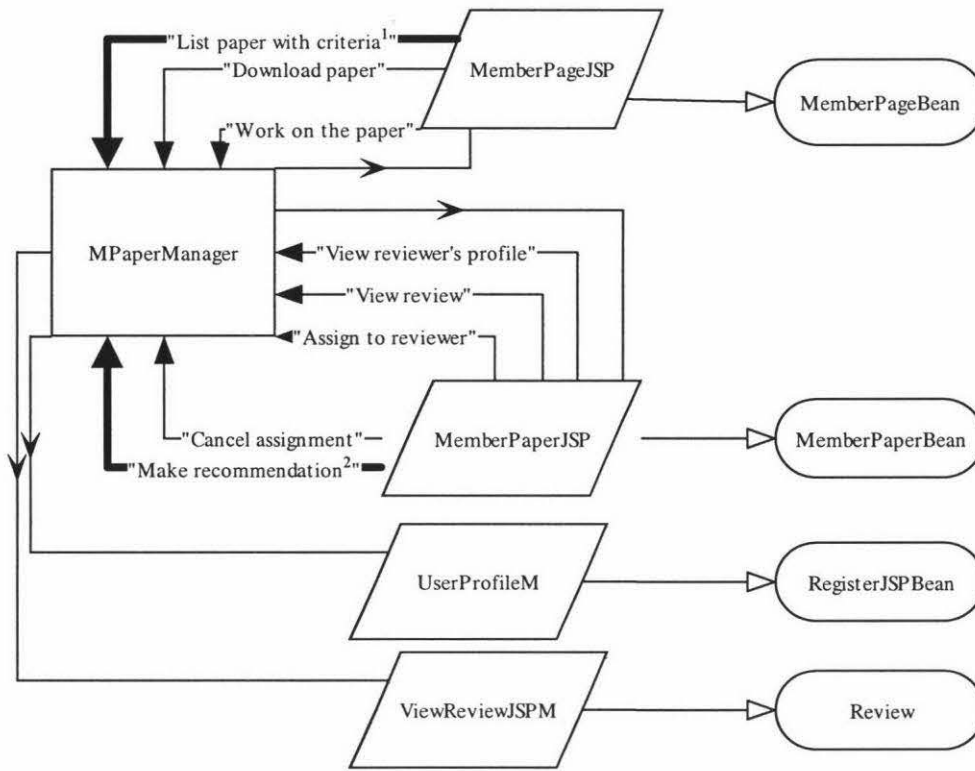


Figure 6-14 Components involved in reviewer's operation

Table 6-5

Request	Controller	Action
Download paper	ReviewHandler	Get the selected paper's URL, copy the file from server to the output stream of the servlet. Transfer the file name to the browser and invoke "save file" dialogue box.
Work on the paper	ReviewHandler	Retrieve the review info from database and put it in the "Review" bean. Forward the request to ReviewJSP
Submit	ReviewHandler	Update the review record in database, forward the request to ViewReviewJSP

Figure 6-15 illustrates the cooperation between components for processing committee members' operations, table 6-6 gives the details of the controlling servlets' processing.



Note:1. represents five kinds of requests: "List papers by paper info", "List papers by reviewer ID", "List papers by paper-reviewer assignment status", "List papers by review status", List paper by recommendation".

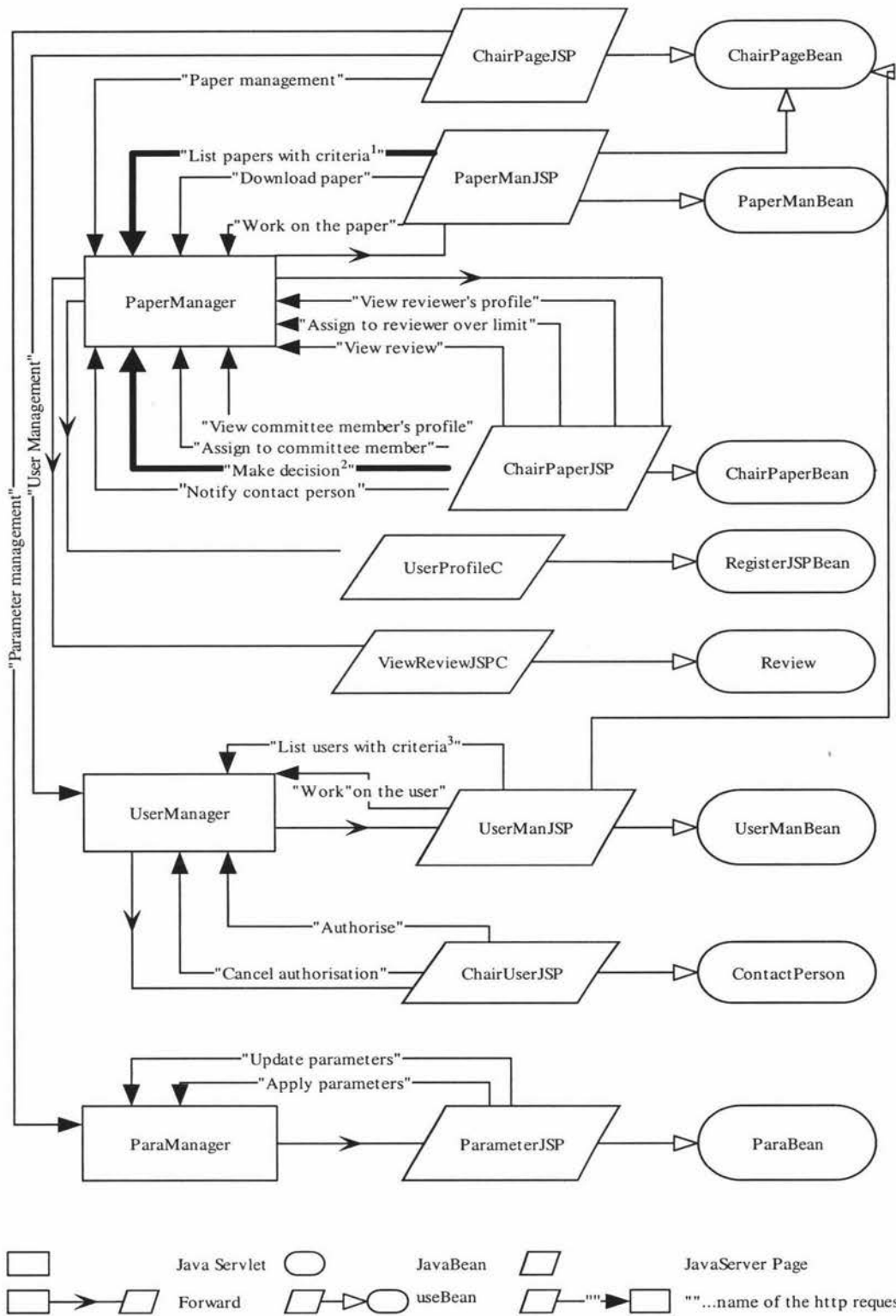
2. represents four kinds of requests: "Accept paper", "Reject paper", "Change category", "Cancel recommendation".

Figure 6-15 Components involving in committee member's operation

Table 6-6

Request	Controller	Action
List paper with criteria	MpaperManager	Put the submitted query type and criteria in MemberPageBean, generate the paper search query, execute the query, put the retrieved papers in the arraylist of paper objects in MemberPageBean. Forward the request to MemberPageJSP
Download paper	MpaperManager	See table 6-5
Work on the paper	MpaperManager	For the selected paper, get all its reviews and put them in the array of Review object in MemberPaperBean, get all the keyword-matching reviewers and put them in the arraylist of MatchReviewer in MemberPaperBean sorted in the matching extent, forward the request to MemberPaperJSP.
View reviewer's profile	MpaperManager	Get the selected reviewer's ID from the arraylist of MatchReviewer in MemberPaperBean, retrieve the reviewer's account information with the ID and put it in RegisterJSPBean, forward the request to UserProfileM
View review	MpaperManager	Get the selected review from thre Review array in MemberPaperBean, and put it in review(here it works as a bean), forward the request to ViewReviewJSPM
AssignToReviewer	MpaperManager	Get the selected reviewer's account and judge if the reviewer is OK for reviewing the paper(not the author, not from the same affiliation as any author, not has been assigned to this paper, not over papers-per-reviewer limit) , if it is, create a new record of review, add a new item in Review array in MemberPaperBean, forward the request to MemberPaperJSP.
Cancel assignment	MpaperManager	Delete the selected review both in database and in the review array in MemberPaperBean, forward the request to MemberPaperJSP.
Make recommendation	MpaperManager	For "Accept paper", "Reject paper", "Cancel recommendation" simply change the "recommd" field in paper table and "recommendation" attribute in MemberPaperBean to "ACCEPT","REJECT","NOTDONE". For "Chang category", change the "recommd" field in paper table and "recommendation" attribute in MemberPaperBean to "ACCEPTASNEWCATEGORY", and change the "category" field of paper table to the value of "ChgToCategory" parameter submitted along with the request. Forward the request to MemberPaperJSP.

Figure 6-16 illustrates the components and message flow in serving the committee chairs' functions. Table 6-7 gives the controlling servlets' processing details.



Notes: 1. represents eight kind of requests: "List papers by paper info", "List papers by reviewer ID", "List papers by committee member ID", "List papers by paper-reviewer assignment status", "List papers by paper-member assignment status", "List papers by review status", "List papers by recommendation", "List papers by decision".
 2. decisions include: "Accept paper", "Reject paper", "Cancel decision".
 3. criteria include: "User type" and "Authorised".

Figure 6-16 Components involving in committee chair's operation

Table 6-7

Request	Controller	Action
Paper management	PaperManager	Forward the request to PaperManJSP, the PaperManJSP creates a PaperManBean by itself.
List Paper with criteria	PaperManager	Put the submitted query type and criteria in PaperManBean, generate the paper search query, execute the query, put the retrieved papers in the arraylist of paper objects in PaperManBean. Forward the request to PaperManJSP.
Download paper	PaperManager	See table 6-5
Work on the paper	PaperManager	For the selected paper, get all its reviews and put them in the array of Review object in ChairPaperBean, get all the keyword-matching reviewers and put them in the arraylist of MatchReviewer in ChairPaperBean sorted in the matching extent, get all the keyword-matching committee members and put them in the arraylist of MatchPCMember in ChairPaperBean Bean sorted in the matching extent, then forward the request to MemberPaperJSP.
View reviewer's profile	PaperManager	Get the selected reviewer's ID from the arraylist of MatchReviewer in ChairPaperBean, retrieve the reviewer's account information with the ID and put it in RegisterJSPBean, forward the request to UserProfileC
View review	PaperManager	Get the selected review from the Review array in ChairPaperBean, and put it in review(here it works as a bean), forward the request to ViewReviewJSPC
AssignToReviewer over limit	PaperManager	Get the selected reviewer's account and judge if the reviewer is OK for reviewing the paper(not the author, not from the same affiliation as any author, not has been assigned to this paper) , if it is, create a new record of review, add a new item in Review array in ChairPaperBean, forward the request to ChairPaperJSP.
Cancel assignment	PaperManager	Delete the selected review both in database and in the review array in ChairPaperBean, forward the request to ChairPaperJSP.
Make decision	PaperManager	For "Accept paper", "Reject paper", "Cancel decision" change the "decision" field in paper table and "decision" attribute in MemberPaperBean to "ACCEPT","REJECT","NOTDONE". Forward the request to ChairPaperJSP.
Notify contact person	PaperManager	Send a message for rejecting or accepting the paper to its contact person's e-mail address.
User management	UserManager	Forward the request to UserManJSP, it creates a UserManBean by itself.
List user with criteria	UserManager	Generate the SQL query for searching users meeting the criteria , put the users account information in the arraylist of ContactPerson in UserManBean, forward request to UserManJSP.
Work on the user	UserManager	Put the selected user's account information in ContactPerson()works as a bean for ChairUserJSP, forward the request to ChairUserJSP.
Authorise	UserManager	Get the register type to be authorised, change the "Authorised" field of the particular user record to be "TRUE" in the particular register type table. Set the Authorised attribute of the RegisterType object in the RegisterType array in ContactPerson to be "true", forward the request to ChairUserJSP.
Cancel authoriation	UserManager	Get the register type to be authorised, change the "Authorised" field of the particular user record to be "FALSE" in the particular register type table. Set the Authorised attribute of the RegisterType object in the RegisterType array in ContactPerson to be "false", forward the request to ChairUserJSP.
Parameter management	ParaManager	Get the parameters from database and put in the ParaBean, forward the request to ParameterJSP
Update parameters	ParaManager	Update the parameters in database with that in the ParaBean. Forward request to ParameterJSP
Apply parameters	ParaManager	Load the parameters from database and assign them to the Public parameter variables in SuperJSPBean and DBHandler so to apply them in the system.

6.6 Summary

This chapter has described the system design and implementation. The interface was designed in groups of web pages navigated as one user interacting with the system. The techniques to incorporate dynamic data in to web page with JSPs using beans are explained for individual HTML form controls. The logic view of the system database model is presented in Entity Relationship Diagram. The system was implemented with Java components. JavaServer Pages are used to implement the interface presentation. Java Beans model the business objects. Java servlets act as process controller reacting to the user requests and providing the workflow control. The message flow between these components is illustrated with a series of diagrams. Event-response tables gave details of the control logic.

Next chapter will conclude the work of the thesis, describe the usage of system, review the system functionality and address some future work.

Chapter 7 Conclusion and future work

This thesis aimed to design and implement a web-based application to facilitate an efficient and effective management of paper submission, review and selection process in academic conferences. Currently existing paper submission and review systems are analysed in terms of functionality and limitation. Use case analysis is applied to gather the system requirement. Java component-based solution is selected based on the investigation of different web techniques. A database with ten tables is setup for storage of permanent information of system. 26 JSPs, 12 Java Servlets and 23 Java Beans are designed and implemented to represent the user interfaces and complete the business logic. In the following sections, the usage of the system will be presented, the functionality of the system will be reviewed, some future work will be addressed.

7.1 The usage of the conference submission system

This section will describe the usage of the implemented system with screenshots of major user interfaces.

On the home page of a conference, the "Register" and "Login" buttons are provided. Before users can login to the system, they should first choose "Register" to register their personal information to the system. The registration page is presented for user to enter the contact information and expertise information as shown in figure 7-1.

After registration, each time a user chooses to "Login", a login screen is presented where user can select appropriate role, as shown in figure 7-2. User is allowed to select from only those roles that user is authorised for.

If user selects "Contact person" role, the system gives the "Author page", where user can view or cancel submitted papers or submit a new paper, as shown in figure 7-3. If user selects to submit a new paper for the second step, the system gives the "Paper submission page", where the user contact information is presented and paper information can be entered. With paper information submitted correctly, the system gives the "Upload file"

screen, where user can select a local file and upload it to the system. The "Paper submission page" and "Upload file" screen are shown in figure 7-4.

If the user selects "Reviewer" role, the system gives the "Reviewer page" where user can select an assigned paper and choose to download it or to work on it, as shown in figure 7-5. If user chooses to work on the paper, the system presents the review form on the "Review page" as shown in figure 7-5. The review form is pretty long so it is separated into three sections in figure 7-6.

Conference:AAAA REGISTRATION PAGE

[Go to Login Page](#)
[Forget my user name or password](#)

If you register as "contact person" you only need to fill the basic user information part of the form. For other types you need to fill the whole form. To login as committee chair or committee member you need to be authorised.
If you already have an account please input your username and password and click the "get my account" button to get you existing account.

User name Password

Basic user information

User name	<input type="text"/>	Password	<input type="password"/>
Re-enter password	<input type="password"/>		
Emai address:	<input type="text"/>		
First name	<input type="text"/>	Last name	<input type="text"/>
		Title	<input type="text" value="Mr."/>
Register type(To be able to login as committee chair or committee member you need to be authorised.)			
<input checked="" type="checkbox"/> Contact Person <input type="checkbox"/> Reviewer <input type="checkbox"/> Committe member <input type="checkbox"/> Committee chair			
Affiliation(university,school or company name)	<input type="text"/>		
Address1	<input type="text"/>		
Address2	<input type="text"/>		
City	<input type="text"/>		
State/Province	<input type="text"/>		
Country	<input type="text"/>		
Zip/Postal Code	<input type="text"/>		
Work phone	<input type="text"/>		
Fax	<input type="text"/>		

Special user information(only for reviewers, committee members, committee chairs.)

Keyword1	<input type="text" value="Education"/>
Keyword2	<input type="text" value="Not selected"/>
Keyword3	<input type="text" value="Not selected"/>
Keyword4	<input type="text" value="Education"/>
Keyword5	<input type="text" value="Machine"/>
Position(organisation)	<input type="text"/>
Brief CV(less than 200 words)	<input type="text"/>
Input your brief CV here	
<input type="button" value="Submit"/>	

[Go to Login Page](#)
[Forget my user name or password](#)

Figure 7-1 Interface for user registration

If the user selects "Committee member" role, the system gives the "Committee member's page", where the user can set various criteria and list the papers which satisfy the criteria, as shown in figure 7-7. The user can then choose to work on one of the papers in the list, the system gives the "Committee member's paper page", where user can download the paper, assign the paper to a reviewer, or make recommendation on the paper, as shown in figure 7-8.

If the user selects the "Committee chair" role, the system gives the "Committee chair page" where user has three sub functions to select: *paper management*, *user management*, and *set up conference parameter*, as shown in figure 7-9.

For the *paper management* sub function the system gives the "Paper management page" where user can set various searching criteria and list papers meeting the criteria, as shown in figure 7-10. The user can choose to work on one of the paper in the list, the system then gives the "Committee chair's paper page" where user can assign the paper to a PC member, assign the paper to a reviewer over limit, make decision on the paper and notify the contact person of the final decision, as shown in figure 7-11.

For the *user management* sub function, system gives the "User management" page where the PC chair can list different types of users e.g. "Not authorised reviewers", as shown in figure 7-12. Once the PC chair selects one user from the list and chooses to work on it, the system gives the "Committee chair's user page", where PC chair can authorise the user to be able to login as particular roles or cancel the authorisation, as shown in figure 7-13.

For the *Setup conference parameter* sub function, system gives the "Parameter settings" page where user can edit the system parameters and apply them, as shown in figure 7-14.

The hyperlinks for *change password*, *modify profile* and *logout* are provided on major user pages, i.e. "Author page" in figure 7-3, "Reviewer's page" in figure 7-5, "Committee member's page" in figure 7-7, "Committee chair's page" in figure 7-9. *Change password* and *modify profile* link to the corresponding pages respectively. If the user chooses to

logout, the current user session is terminated and the "Login page" is presented, as shown in figure 7-2.

Forget my user name or password
Please enter your user name and Password.

User name

Login as CONTACTPERSON

CONTACTPERSON
REVIEWER
PCMEMBER
PCCHAIR

Password

Login

Figure 7-2 Login page

Conference:AAAA (System parameter)
Author Page
Deadline for paper submission:2002-05-05
Contact person name: He Meng (System parameter)
[Change my password](#)
[Modify my profile](#)
[Logout](#)

You have submitted 5 papers:

Select	Number	Title
<input type="radio"/>	1	p1
<input type="radio"/>	2	p2
<input type="radio"/>	3	p3
<input checked="" type="radio"/>	16	Paper number seventeen
<input type="radio"/>	29	hhhhhddd

Figure 7-3 Interface for contact author

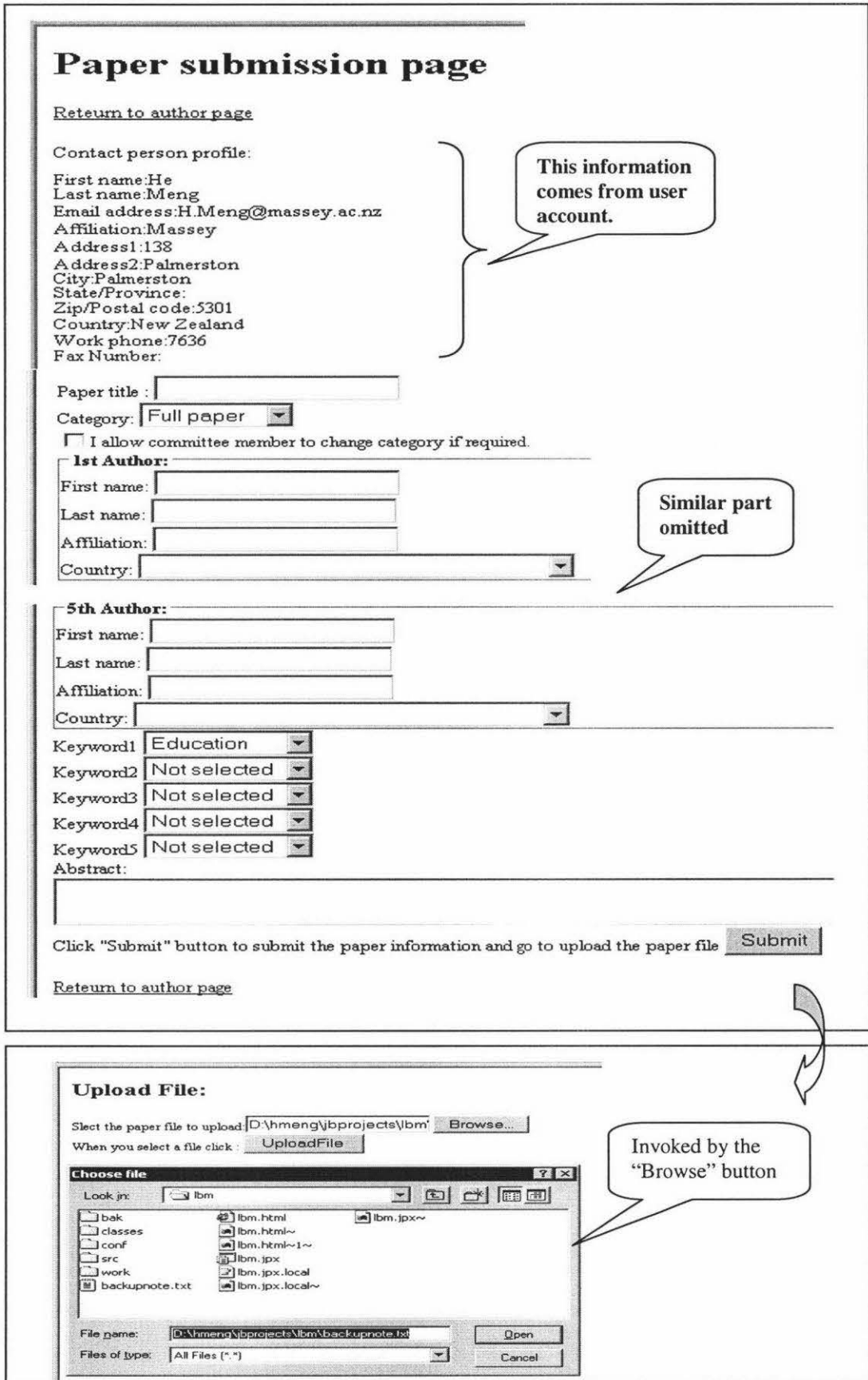


Figure 7-4 Paper submission interface

Conference:AAAA

Reviewer page
Deadline for review submission:2002-06-06

Reviewer name: He Meng

[Change my password](#)
[Modify my profile](#)
[Logout](#)

You have been assigned with 4 papers:

Select	Paper No.	Paper title	Review Made
<input type="radio"/>	3	p3	false
<input type="radio"/>	11	A paper submit by hmeng2 at 5:00PM	false
<input type="radio"/>	2	p2	false
<input checked="" type="radio"/>	33	The HTML complete reference	false

Figure 7-5 Reviewer's page

The screenshot shows a Microsoft Internet Explorer window displaying the reviewer's page from Figure 7-5. A "File Download" dialog box is open, showing the file path: "fm.examples.ReviewHandler from #007471.massey.ac.nz". The dialog asks "What would you like to do with this file?" and offers two options: "Open this file from its current location" (selected) and "Save this file to disk". A callout bubble points to the file path with the text "File path is hidden from the reviewer".

Figure 7-5 Reviewer's page

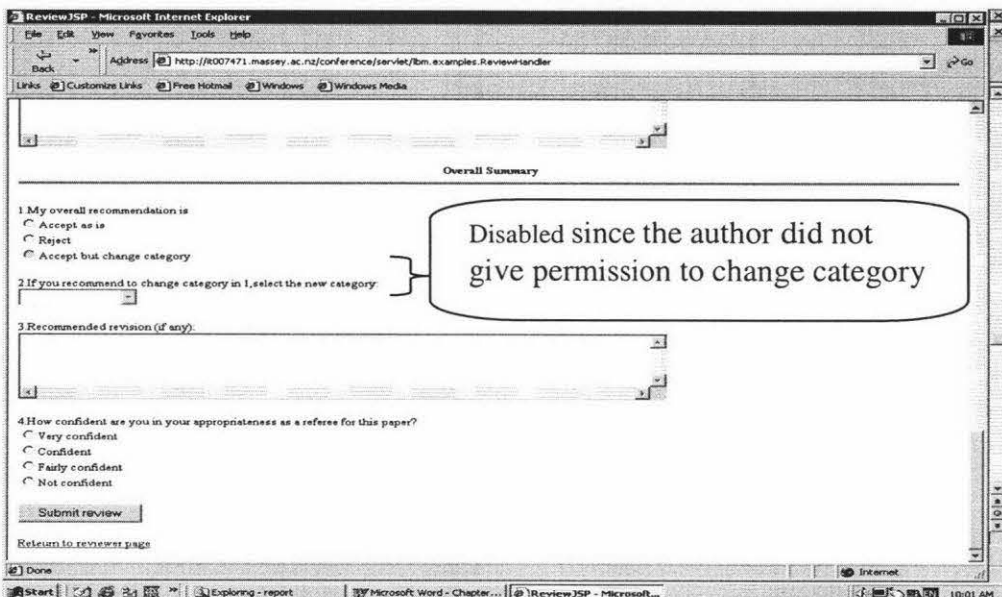
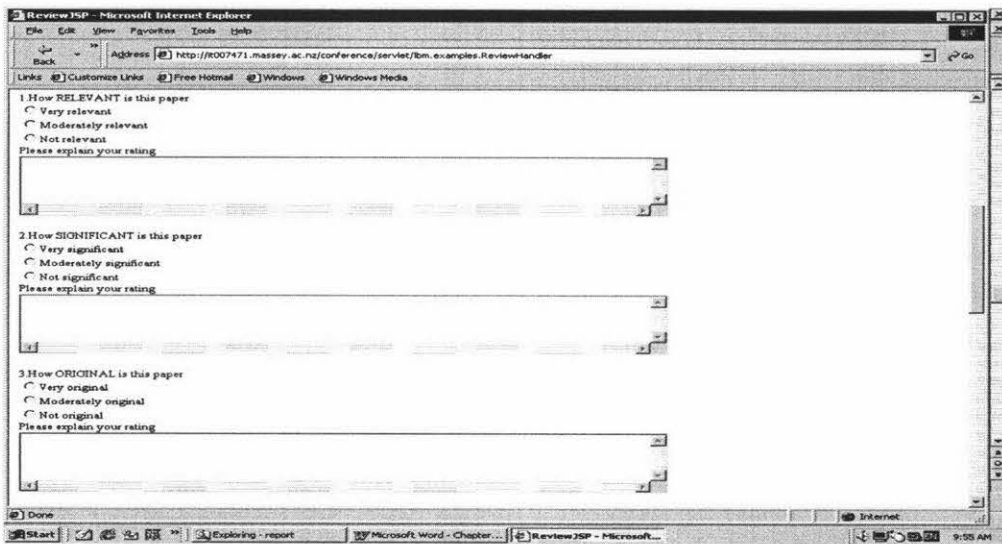
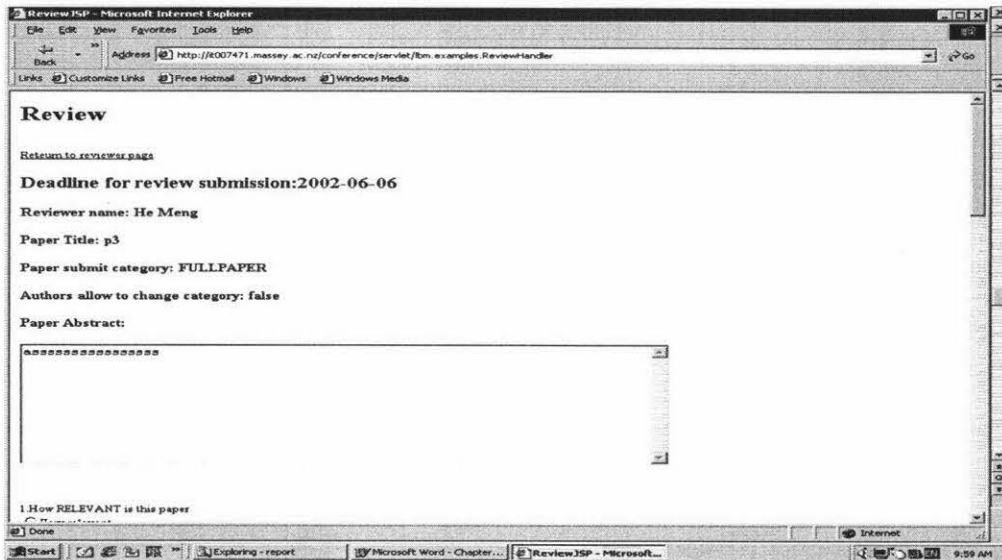


Figure 7-6 Interface for making review

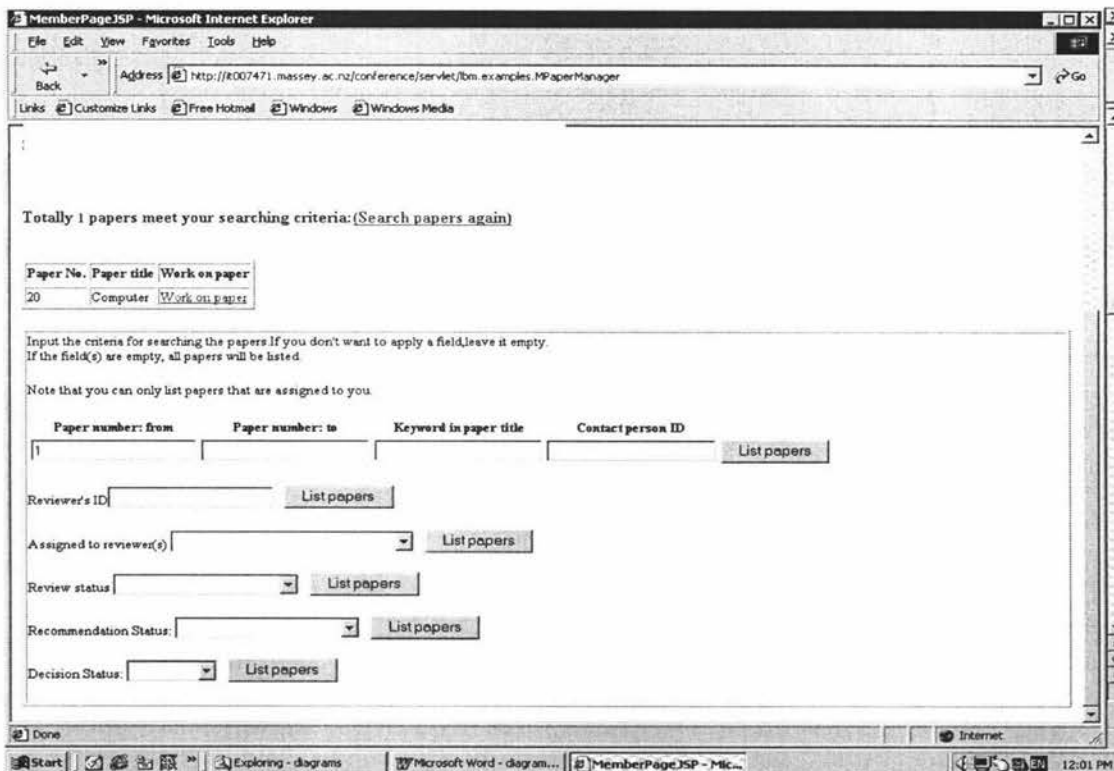
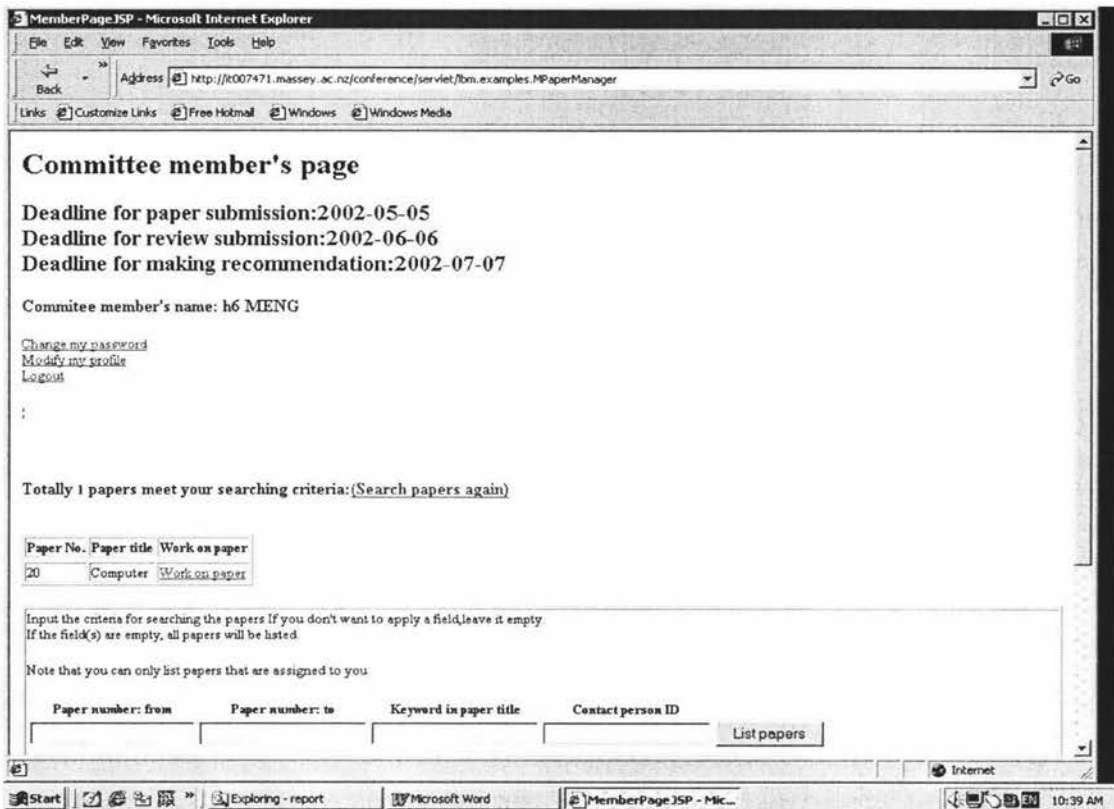


Figure 7-7 committee member's page

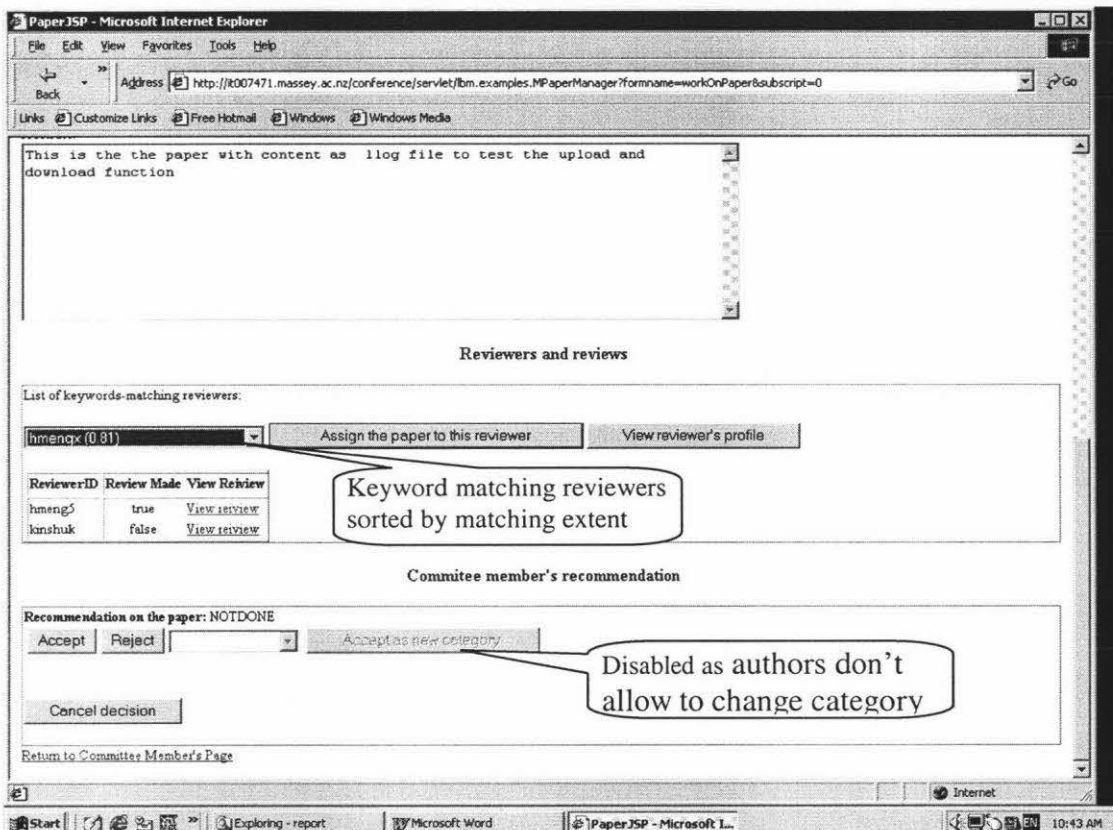
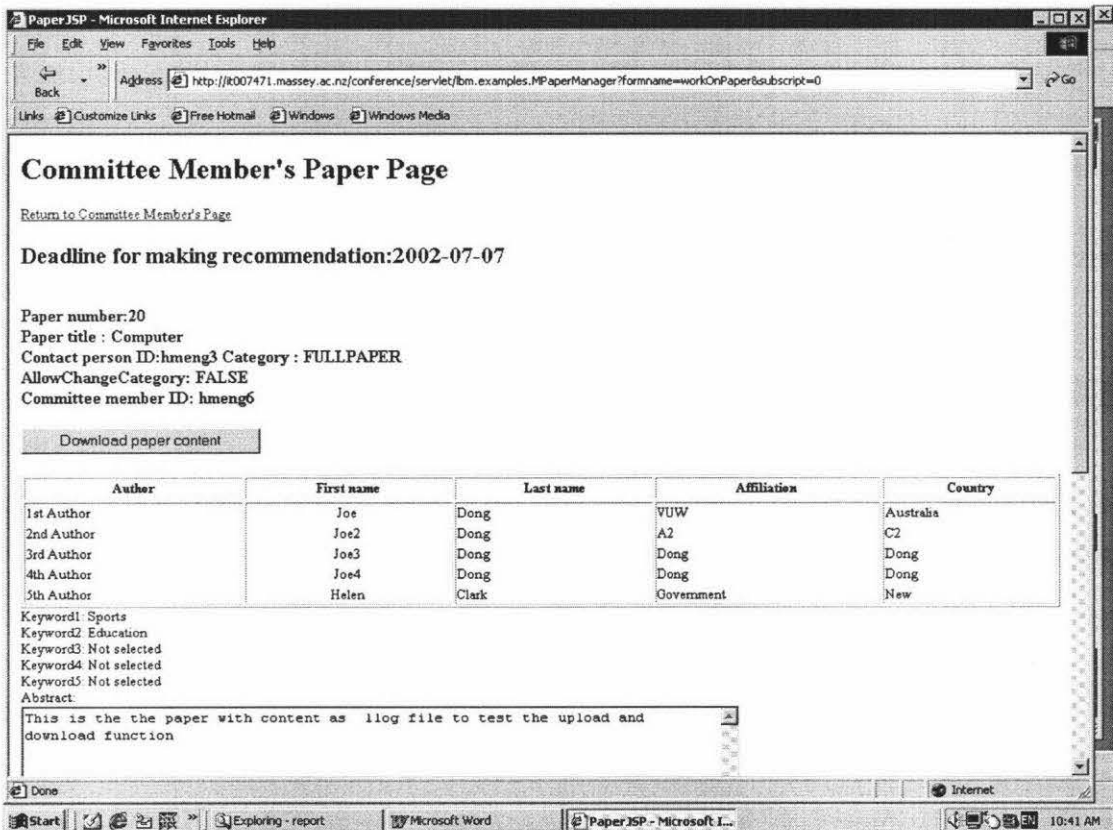


Figure 7-8 Committee member's paper page

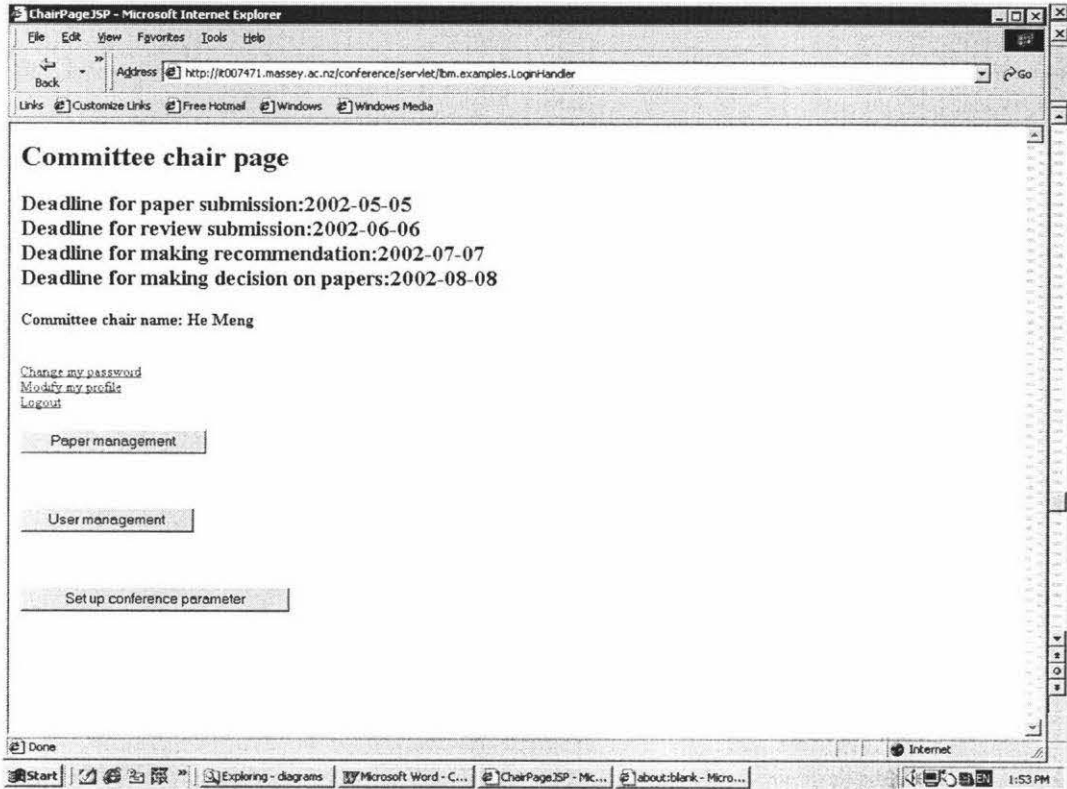


Figure 7-9 PC chair's major page

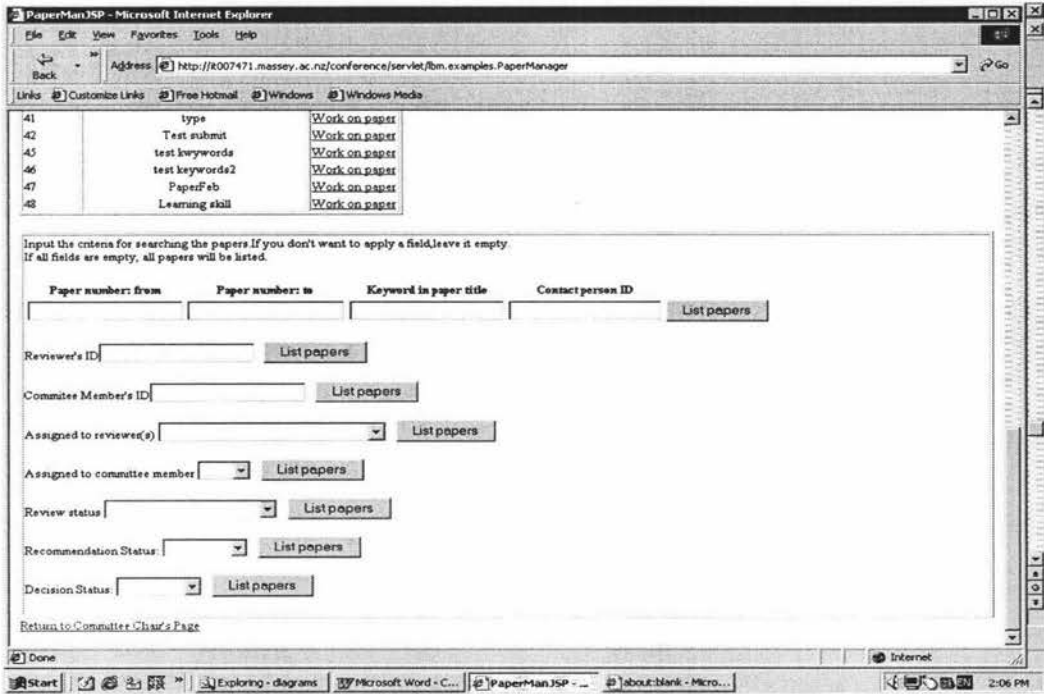
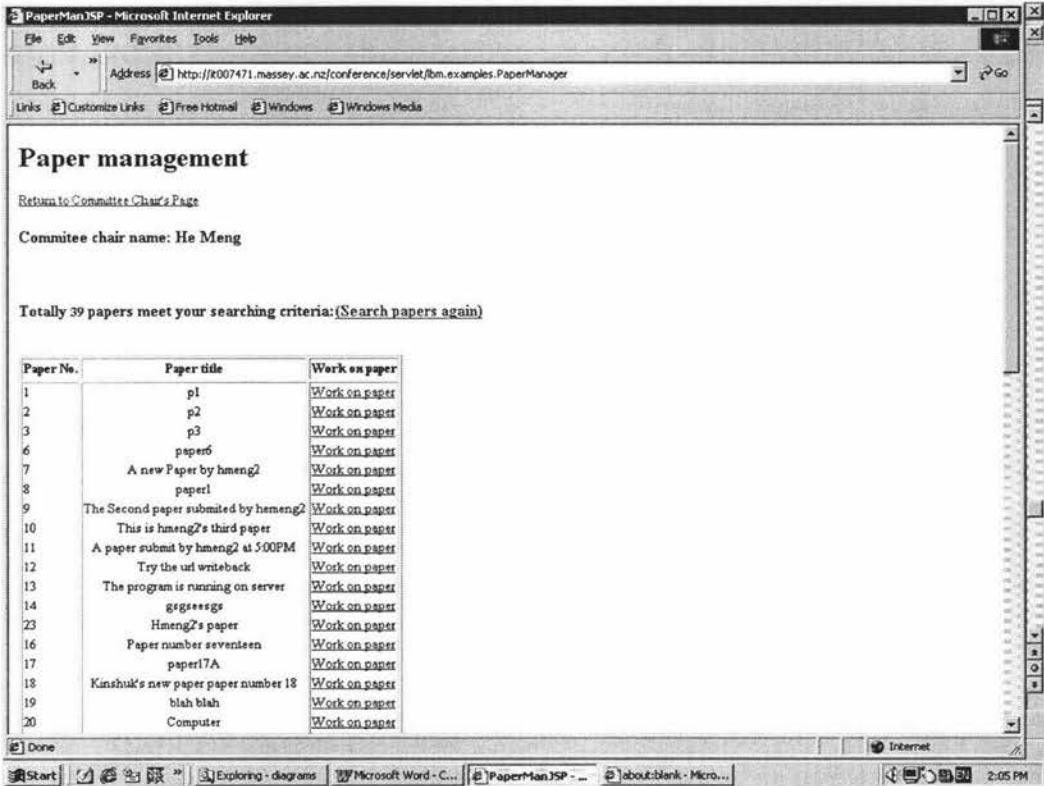


Figure 7-10 PC chair's paper management page

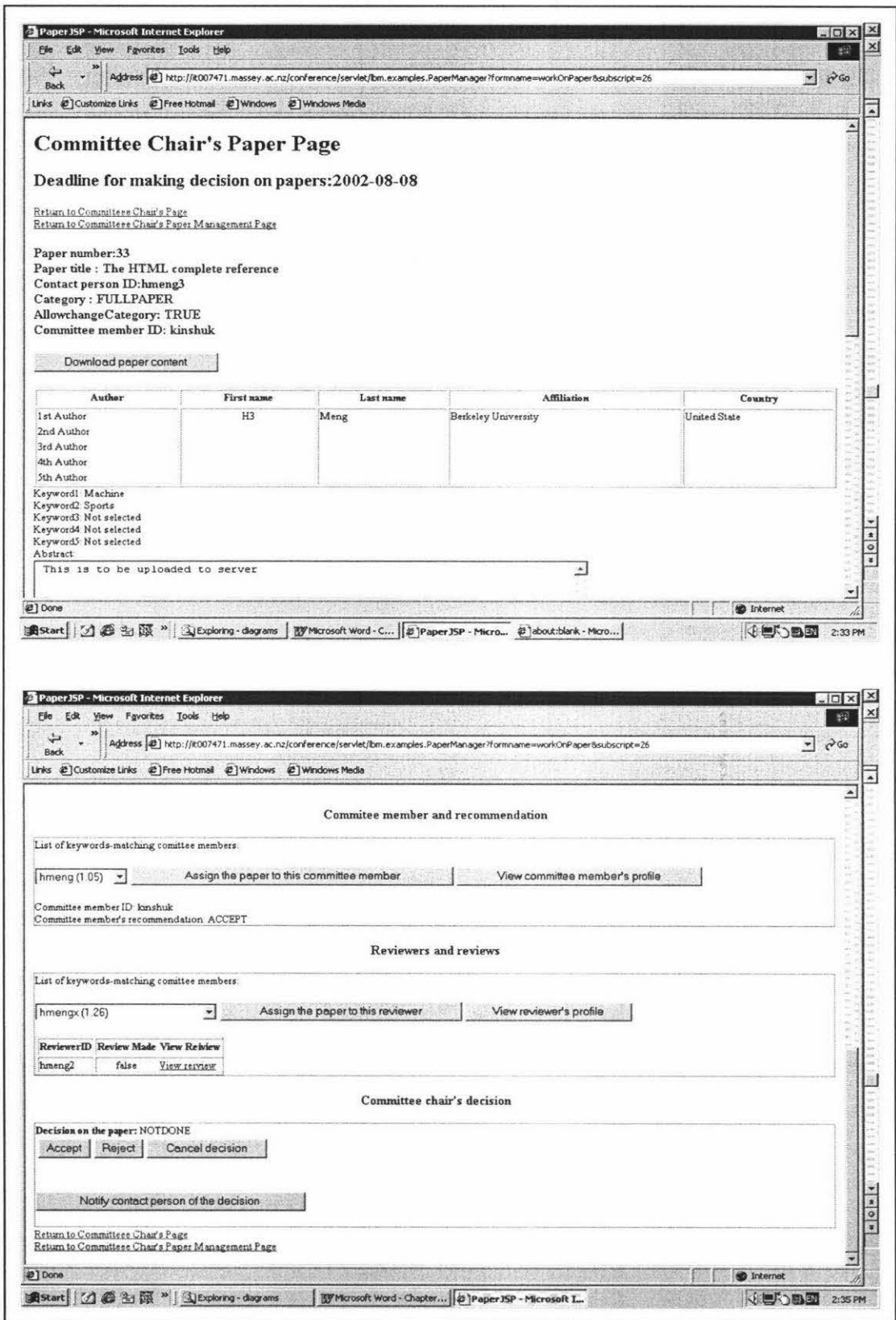


Figure 7-11 PC chair's paper page

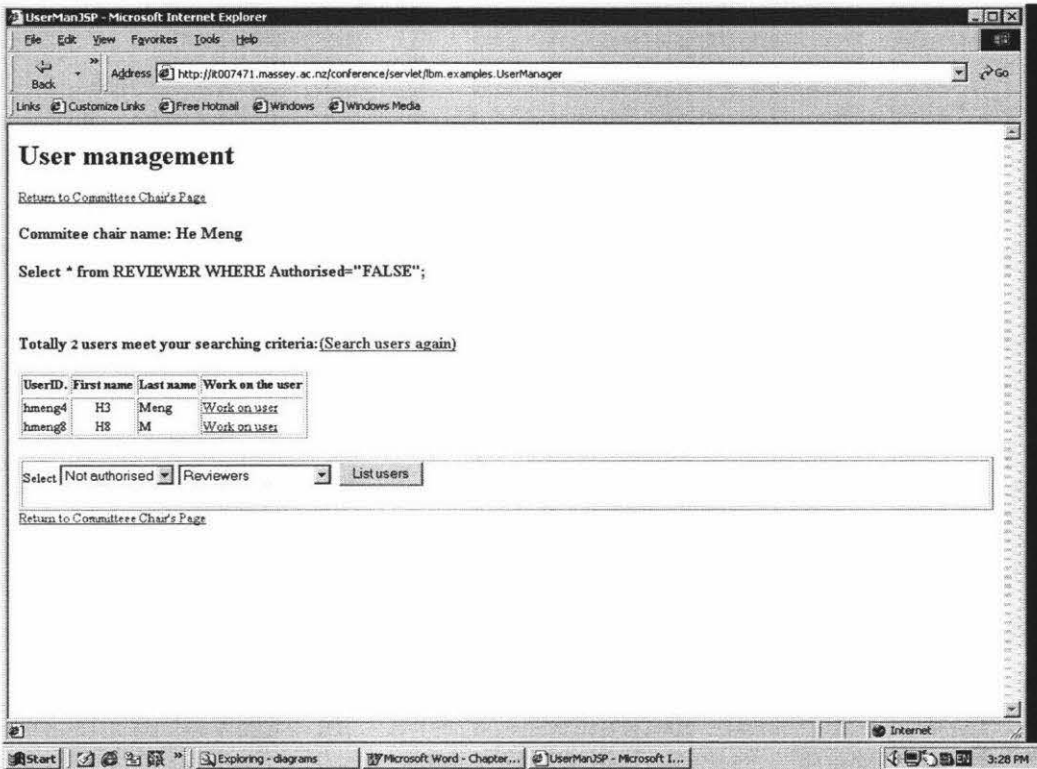


Figure 7-12 User management page

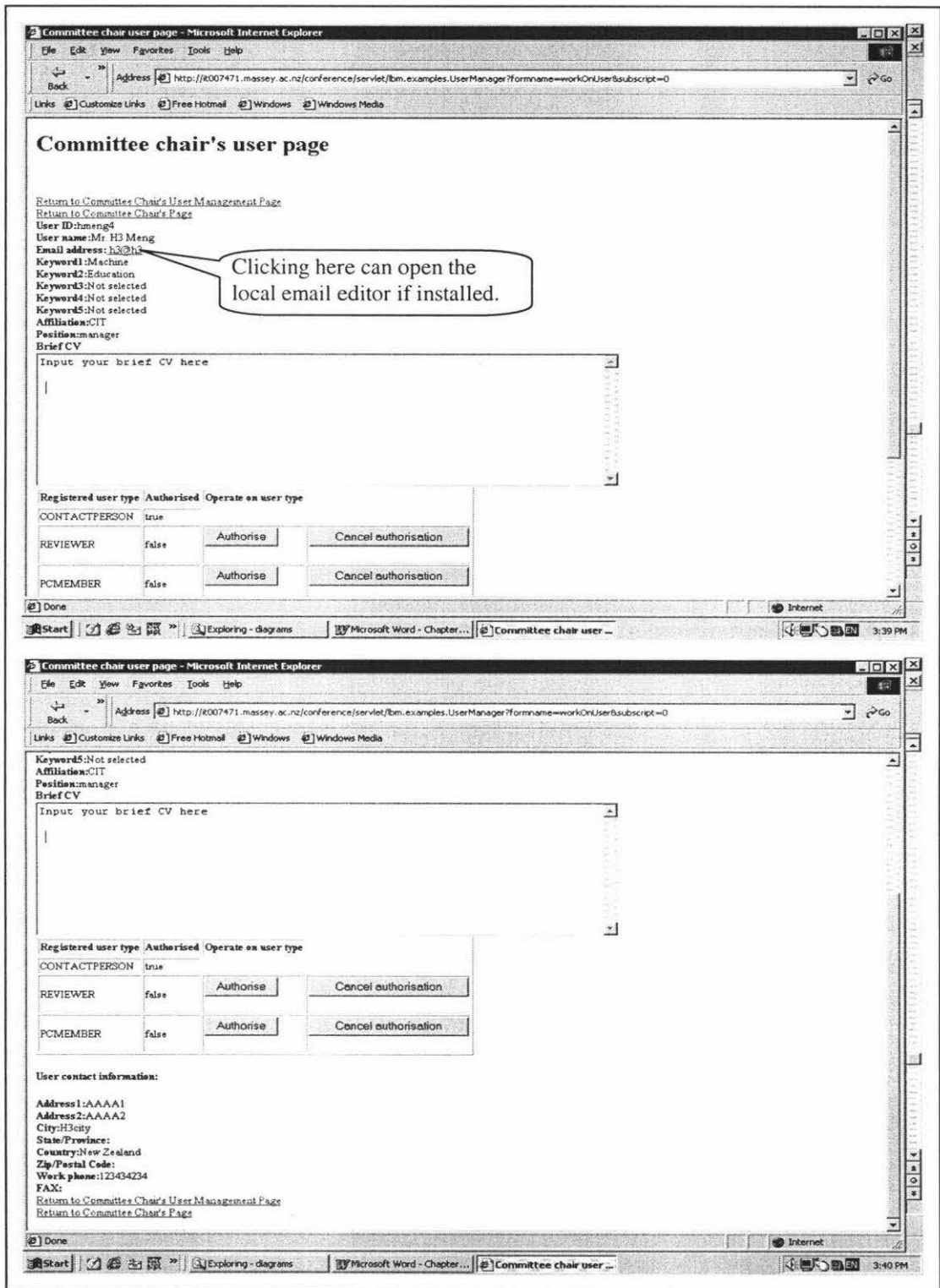


Figure 7-13 PC chair's user page

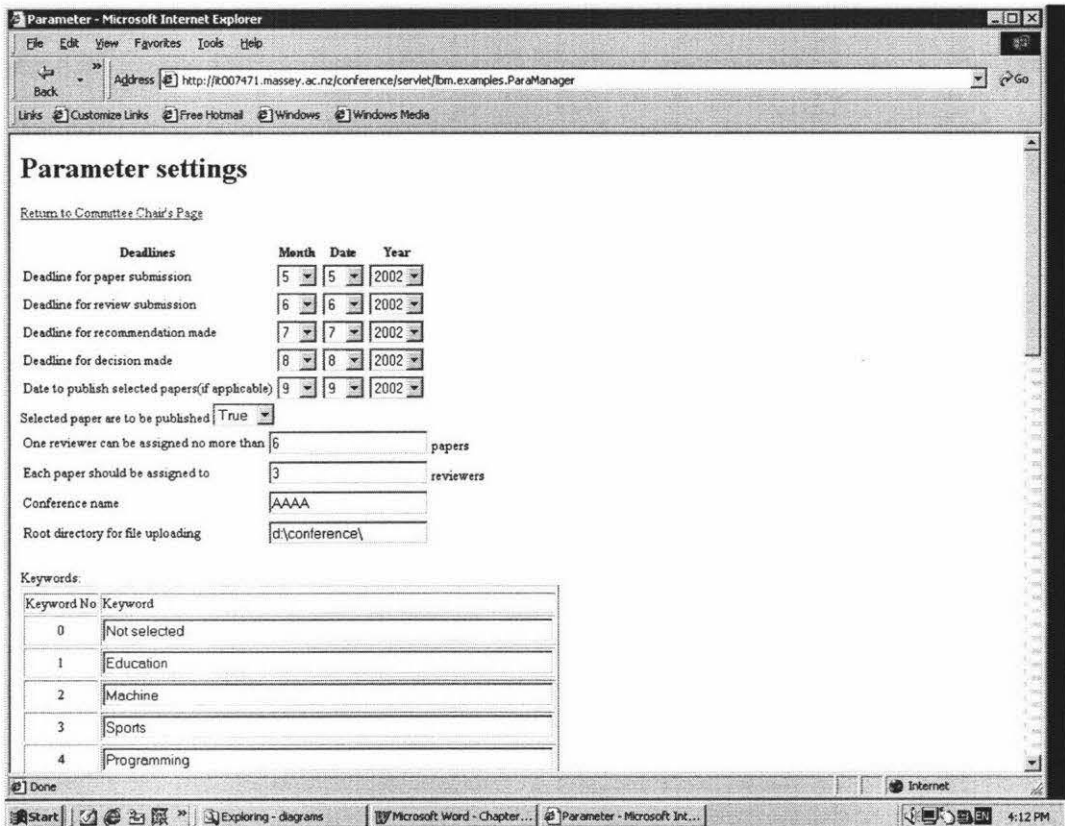


FIGURE 7-14 PARAMETER SETTING PAGE

7.2 Review of the system functionality

The system was implemented as a real web-based application rather than a group of ad hoc scripts that most conferences provide for on line paper and review submission. Authors, reviewers, committee members and committee chairs are considered as system users with different privileges. They all have personalised interfaces with which to perform their various kinds of tasks online.

The 3-tier client/server architecture design of the system separates the presentation logic and business logic clearly and eases the maintenance and further improvement. The backend database keeps records of user account, paper, review and system information. This enables the different system users to share the data and helps to provide them with a common workspace.

Applying a pure Java solution makes the application independent of platform requiring only the Java virtual machine, so the server side program can work on any operating system.

The feature of multi-threading with light-weight thread context switch of Java programs greatly enhances the system performance in contrast with most of the existing systems relying on CGI programs that require heavier weight process startup and initialization code on each request (Kirby and Lee, 2000).

Database connection pools technique reduces the affect of the time consuming database connection, therefore accelerates the response speed.

The uniform user registration allows the users who play more than one role in the system to enter personal information only once. Self-registering facility saves the PC chair's effort to generate unique ID and set up account for each reviewer or committee member.

The technique of JSP using JavaBean helps to realise real dynamic web pages, e.g. after logging in, different reviewers see the same JSP file but with different paper list. Whenever dynamic information changes, it is reflected on web page the next time the web page is presented.

The handy and easy-to-use session tracking technique in Java servlet and JSP provides a convenient way for user authentication.

Form based file uploading and system generation of unique paper ID saves the work for contact authors to submit the paper in two separate steps and saves the work for PC chair to generate the unique paper ID.

The servlet-supported file downloading requires only one copy of each paper on the server and allows the papers to be saved securely outside of publicly accessed web server area. This is in contrast to some existing systems, where papers need to be copied to each of their reviewers' directory, which must be within the web server area. With uploaded files saved out of the web server area and paper URL kept in secure database, the

reviewers can only download papers that are assigned to them. The servlet-supported file downloading also makes tracing of paper downloading process possible.

Keyword matching algorithm facilitates the paper-to-committee member assignment and paper-to-reviewer assignment.

Searching papers with various criteria enables the committee members and committee chairs to monitor the review process and eases the recommendation making and decision making.

7.3 Future works

Limited by time and scope of the project, some nice-to-have features are identified for future implementation.

The layout of the interface can be elaborated. Fonts and colours, can be adjusted to better present the content, HTML frames can be used to improve the look and feel of the interfaces and enhance the usability. The display format of the date could also be flexible to make the system user-friendly for people from different parts of the world.

Since the paper downloading is done via servlets, the tracing of paper downloading by the reviewers is possible. The process can therefore be enhanced by adding "download time" function to the review record. This will enable the committee members to check if reviewers have downloaded the assigned papers or not and remind them if necessary. It could also be helpful for copyright issues.

With the help of session tracking technique, audit trail for the system access could be implemented without difficulty.

Java language provides excellent *error handling with exceptions* (Eckel, 2000). The error logging could therefore be implemented by redirecting the error message to error log file. This will enable the system administrator to be aware of the problems and correct them in time.

REFERENCES:

- AAAI, (2001). *Eighteenth national conference on artificial intelligence call for papers* <http://www.aaai.org/Conference/National/2002/Calls/aaai02-call.html>
- AERA, (2000) *AERA Electronic Proposal Submission System (Tiger) Tutorial*
<http://www.klick.org/2000aera/aeratrain/index.html>
- Coward, D., 2000 *Java Servlet Specification Version 2.3*
http://jcp.org/aboutJava/communityprocess/first/jsr053/servlet23_PFD.pdf
- Darby, C. (1998). Applet and servlet communication. *Java Developer's Journal*, Sep, 1998. <http://www.j-nine.com/pubs/applet2servlet/Applet2Servlet.html>
- Eckel, B. (2000). *Thinking in Java* (2nd ed.). Upper Saddle River, New Jersey: Prentice-Hall, Inc. p531
- Ford, S., Wells, D., and Wells, N. (1996). *Web programming language*
<http://oopweb.com/Dictionaries/Documents/WebPrograming/VolumeFrames.html>
- Gerber, R., Hollingsworth, J., Porter, A. (1999) START: submission tracking and reviewing toolset. <http://www.cs.umd.edu/~rich/START/>
- Hughes, C., Birznieks, G. (2002). *Serving up web server basics*
<http://webcompare.internet.com/webbasics/>
- Jacobson, I. (1992). *Object-oriented software engineering: a use case driven approach*
- Kirby, G. (1998) *Electronic conference submission system*
<http://www-ppg.dcs.st-and.ac.uk/Conferences/admin-system/Readme.html>
- Kirby, J., Lee, C. 2000 *Developing JDBC Servlet applications-A replacement for*

CGI: A white paper by Dynamic Information Systems Corporation.
<http://www.disc.com/jdbcserv.html>

Microsoft Research (2001a). *Microsoft's Conference Management Site*
<http://cmt.research.microsoft.com/cmt/>

Microsoft Research (2001b). *VLDB 2001 Americas-submission guideline*
<http://cmt.research.microsoft.com/VLDB2001U/CallForPapers.asp>

Microsoft Research (2001c). *KDD 2001 -submission guideline*
<http://cmt.research.microsoft.com/KDD2001/CallForPapers.asp>

Microsoft Research (2001d). *Microsoft's Conference Management Site*
<http://cmt.research.microsoft.com/cmt/FAQ.htm>

MYSQL AB, 2002 <http://www.mysql.com/downloads/gui-mysqgui.html>

Nicol , D.M. (2001) *WIMPE : Web Interface for Managing Programs Electronically*
<http://www.cs.dartmouth.edu/~nicol/wimpe/wimpe6.1.html>

Nierstrasz,O. (1999) *Identify the Champion: An Organisational Pattern Language for Program Committees.*
<http://www.iam.unibe.ch/~oscar/Champion/champion.pdf>

Powell, T.A. (1998) *HTML: The Complet Reference* (3rd ed.).

Preuss, T., Lund, K., Halvorsen, P. (1999). *ConfMan mannual*
<http://confman.unik.no/~confman/MANUAL/>

Rick Snodgrass (1999): *Summary of Conference Management Software*
<http://www.acm.org/sigs/sgb/summary.html>

Schulzrinne, H. (2001) *EDAS - Editor's Assistant*
<http://www.cs.columbia.edu/~hgs/edas/edas.html>

Sharland,R.(1991) *Package Evaluation: A Practical guide to Selectcting Applications and System Software*. Aldershot Hants, England

Snodgrass,R.(1999) *Summary of Conference Management Software*
<http://www.acm.org/sigs/sgb/summary.html>

Stadt, R. (2000). *CyberChair-an online submission and review system, or: a program chair's best friend* <http://www9.org/final-posters/poster66.html>

Sun MicroSystem Inc. (2001a). *Java™ 2 Platform, Enterprise Edition Overview*
<http://java.sun.com/j2ee/overview2.html>

Sun MicroSystem Inc. (2001b). *Enterprise JavaBeans™ technology*,
<http://java.sun.com/products/ejb/>

Webopedia
<http://www.webopedia.com/>

White, S., Hapner, M. (1998). *JDBC™ 2.0 API*, Sun Microsystems Inc.

WU, A.W., Wang, H., Wilkins, D. (2000). *Performance comparison of alternative solutions for web-to-database applications: Proceeding of southern conference on computing*, The university of southern Mississippi, Oct 26-28,2000
http://rain.vislabs.olemiss.edu/~ww1/Slide_Show/Images/SCC_Amanda/SCC_Amanda.pdf