

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

**CONFSYS2: A Redesigned Web-based Multi-Conference
Management System**

Min Huang

A Thesis
in
The Department
of
Computer Science and Software Engineering

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

August 2009

©Min Huang, 2009



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-63115-7
Our file *Notre référence*
ISBN: 978-0-494-63115-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

ABSTRACT

CONFSYS2: A Redesigned Web-based Multi-Conference Management System

Min Huang

This thesis presents the design and implementation of ConfSys2, an advanced redesign of ConfSys system, with additional features to help conference/journal organizers to manage the processes of academic conferences/journal, and to provide related services for author and conference participants. The ConfSys2 uses the same Tomcat – Java Servlet/JSP – MySQL platform as ConfSys, but redesigned database structure, software framework, and user interfaces. The features of ConfSys have been retained and improved both in user friendliness and efficiency in ConfSys2, and the lessons learnt from developing and using ConfSys have been incorporated. ConfSys2 not only implements better user interface for ConfSys's useful functions, such as automatically/manually allocating paper to reviewers, debating and rating paper, but also introduces new concepts, such as conference/journal series management, user-group-function management, and smart daemon in conference management to improve data sharing, reduce repetitive work and make management work more flexible. Furthermore, the redesign of databases and software framework provides clear structure and flexibility, thus making the maintenance and expanding of the software system easier.

Acknowledgements

It is a pleasure to thank those who made this thesis possible. First of all, I would like to express my deepest gratitude to my thesis supervisor Dr. Bipin C. Desai for his consistent guidance, great patience, and enthusiastic support. Secondly, I would like to thank the previous ConfSys developers, include Yuwei Feng, Jin Xin, Zhengwei Gu, Guan Hui. Without their experiences and help, the ConfSys2 could not be developed. Finally, I would like to show my gratitude to my wife and my parents for their love, support, and encouragement.

Table of Contents

List of Figures.....	vii
List of Tables.....	ix
1. Introduction	1
1.1. Problem statement and proposed solution	1
1.2. Organization of the thesis	5
2. Literature Review and Survey of Conference Management Systems.....	6
2.1. Existing work on conference management systems	6
2.2. Survey on conference management systems.....	10
2.3. Our Contribution.....	20
3. Features of ConfSys.....	23
3.1. History of Confsys	23
3.2. Structure and features of Confsys	24
3.3. Limitations of Confsys.....	33
3.4. New requirements and obstacles.....	36
4. ConfSys2: an advanced redesign of ConfSys	42
4.1. Database structure redesign	42
4.2. Application architecture redesign	56
4.3. User interface redesign	63

5. Enhancements and new features in ConfSys2	67
5.1. System initializing program.....	67
5.2. Uniform user interface and help system and their implementation	71
5.3. jQuery powered Ajax enabled friendly user interface	74
5.4. Conference/Journal series management.....	78
5.5. User management and community mode.....	81
5.6. Security management and User-Group-Function Mapping	83
5.7. Uniform parameter/template management	86
5.8. Flexible review system	88
5.9. Smart daemons.....	90
5.10. Dynamic user relationship to prevent conflict of interest.....	93
6. Experience	96
7. Conclusion and future work	99
7.1. Conclusion	99
7.2. Future work.....	101
Reference	103
Appendices.....	107

List of Figures

Figure 2-2-1 CMT's Role-Action-Phase relationship.....	14
Figure 3-2-1 Web-Based Three Tier Mode	25
Figure 3-2-2 Admin Main Menu.....	27
Figure 3-2-3 General Chair Main Menu	28
Figure 3-2-4 Author Main Menu.....	29
Figure 4-1-1 Existing ConfSys Database Structure - 1	43
Figure 4-1-2 Existing ConfSys Database Structure - 2.....	44
Figure 4-1-3 ConfSys2 Database Structure	52
Figure 4-2-1 ConfSys Application Architecture	56
Figure 4-2-2 ConfSys2 Application Architecture	59
Figure 4-3-1 ConfSys2 User Interface for User Login	63
Figure 4-3-2 ConfSys2 User Interface After Logged In	64
Figure 5-1-1 System Initializing Program Start.....	68
Figure 5-1-2 System Initializing Program Step Result	69
Figure 5-1-3 System Initializing Program System Parameter Setting.....	70
Figure 5-2-1 Uniform User Interface.....	71
Figure 5-2-2 ConfSys2 Help Center	72

Figure 5-2-3 ConfSys2 Online Help System	73
Figure 5-3-1 jQuery Powered Intuitive User Interface	75
Figure 5-3-2 Ajax Enabled Friendly User Interface -1	76
Figure 5-3-3 Ajax Enabled Friendly User Interface -2	77
Figure 5-4-1 ConfSys2's Conference/Journal Series Management	78
Figure 5-4-2 Conference/Journal Series Management – Add New Conference.....	79
Figure 5-4-3 Conference/Journal Series Management – Add New Term	80
Figure 5-5-1 ConfSys2 User Management	81
Figure 5-5-2 Community Mode	82
Figure 5-6-1 User – Group – Function Mapping	83
Figure 5-6-2 ConfSys2 Group Management.....	84
Figure 5-6-3 Group User Management.....	84
Figure 5-6-4 Group Function Management	85
Figure 5-7-1 Uniform Parameter Management.....	87
Figure 5-8-1 User Interface for Defining a Flexible Review Format	89
Figure 5-8-2 Flexible Review Presentation	90
Figure 5-9-1 ConfSys2 Smart Daemons	91
Figure 5-10-1 Dynamic User Relationship in ConfSys2 Database	93

List of Tables

Table 2-2-1 Comparison on conference management systems	19
---	----

1. Introduction

1.1. Problem statement and proposed solution

With the wide use of Internet, most academic conferences and journals are now using electronic format, with the Internet as the transfer method. While some academic conferences still use separate software to manage the workflow of conferences/journals, several integrated conference management software tools have been developed to help managing conferences. On Google, there are more than 46 millions search results about “conference management software” [1]. Conference management software, such as EasyChair[2], Microsoft’s Academic Conference Management Service[3], ConfTool[4], etc. provide functionalities to meet the conference organizers’ needs.

On Wikipedia[5], basic functions of a conference management system have been defined as the follows:

- Receiving paper submissions: allows authors to submit their electronic format files through the Internet.
- Hiding the author names from reviewers to enforce review impartiality. An enhancement of this function is to prevent distributing a paper to the acquaintance of the authors to avoid conflict of interest.
- Collecting reviewers’ domain of expertise.

- Assigning reviewers to papers according to reviewers' interests. The assigning process can be done automatically or manually.
- Distribute papers to reviewers to review. The system should allow the reviewers to download the papers assigned to him/her, and limit access to unassigned papers.
- Collect review result from reviewers, and allow privileged user to monitor the review procedure.
- Share review among reviewers of the same paper and enable them to debate on controversial decisions. This sharing should be done once all all reviewers have submitted their own scores and comments to ensure independence of reviews.
- Ranking review and set acceptance threshold.
- Inform the authors of the review result and program committee decision on the paper. The reviewer names should be hidden from the authors.
- Collecting final accepted versions.
- Registering attendees
- Publishing proceedings

ConfSys[6], one of the first integrated web based conference management software developed under the supervision of Dr. Bipin C. Desai of Concordia University since 1999, can provide the above functions for chairs, reviewers, and authors to support the

whole process of a conference. In addition, it manages the actual meeting by collecting the presentation slides and author introductions. Its use over the last few years has shown the maturity of its implementations of the concepts of managing the tasks over the entire span of a conference. Since the introduction of ConfSys, other systems have appeared on the scene and the participants have developed higher expectations and hungered for new features. These features include:

- To manage multiple conferences in a single installation of the system. This requires functions for admin to easily setup a new conference and its parameters, setting up the privileges for conference chairs, organizers, and other participants.
- To share the information between cycles of a conference series. For example, allow conference chair to copy the committee group to the next year's conference.
- To be able to manage the process of a journal, and automatically advance to the next volume of a journal.
- To enhance the communication between authors, program committee, and chairs
- To allow chairs to share part of his work with one or more person to help managing the conference process
- To enhance the security of system so that users can only access functions that he's assigned or supposed to access
- To make a number of routine management tasks to be done automatically at given

times and in certain situations, thus reduce the workload of the management groups.

- To provide better user interface and more help information to make the system easy to use.

However, previous version of ConfSys's database structure and software architecture lacks flexibility and thus make it difficult for further development. In addition, the old style hardcoded menu system doesn't have the mechanism for security management, and not suitable for implementing the new features and requirements. Hence there is a need to redesign ConfSys.

ConfSys2 is the advanced redesign of ConfSys system. It uses the same Tomcat – Java Servlet/JSP – MySQL platform as ConfSys, but redesigned database structure, software framework, and user interfaces. During the redesign, we not only consider the requirements of existing features in ConfSys and new features proposed, but also prepared for possible future changes and requirements. The redesign of databases and software framework provides clear structure and flexibility, thus make the maintenance and expanding of the software system easier. The features of ConfSys have been retained and enhanced in ConfSys2, and the lessons learnt from developing and using ConfSys have been incorporated. ConfSys2 not only implements better user interface for ConfSys's useful functions, such as automatically/manually allocating paper to reviewers,

debating and rating paper, but also introduces new concepts, such as multi conference/journal series management, user-group-function management, and smart daemon in conference management to improve data sharing, reduce repetitive work and make management work more flexible.

1.2. Organization of the thesis

The rest of the thesis is organized as following:

- In chapter 2, we review existing work on conference management systems and present a survey of some widely used conference management systems and analyze their features.
- In chapter 3, we specifically analyze the ConfSys system, its features and limitations, and new features that have been proposed.
- In chapter 4, we introduce ConfSys2, a redesign of ConfSys. This chapter includes the techniques used in ConfSys2, its new database structure, software architecture, and user interface.
- In chapter 5, we detail enhancements and new features in ConfSys2 and the implementation of those new features.
- In Chapter 6, we give experience in testing and expanding ConfSys2 system.
- In chapter 7, we give conclusion on ConfSys2 and future work

2. Literature Review and Survey of Conference Management Systems

With the rapid development of Internet, World-Wide-Web (WWW) has become the most commonly used facility to access and share information. Increasing numbers of academic conferences are now managed in the electronic format and using the internet as the transfer method, wherein conference management systems play important roles. Some well known conference management systems include: EasyChair [2], MACMS [3], ConfTool [4], OpenConf [7], Confious [8], ConfMaster [9], etc. As more and more conference management systems are developed, concepts and tasks in conference management systems have been identified. In this chapter, we will review existing work on conference management systems, and survey and compare several widely used conference management systems.

2.1. Existing work on conference management systems

The development of conference management systems (CMS) can be traced back to late 1990s. In Rick Snodgrass's survey on CMS in 1999[10], there were already 20 systems of this kind were developed as prototype or in production quality. Different programming languages were used, such as C++, Java, PHP, Perl, etc. Several systems adopted database management system (DBMS) to store the data in conferences. These DBMS include MYSQL-Lite, MS SQL, Sybase, etc. When transferring files or providing user

interface, some of those systems used WEB technique, which became the trend for developing online conference management system. From the survey, some major functionalities of a basic CMS surfaced. These functionalities include submission of paper, assignment to reviewers, submission of review, conference registration, etc. These functions help organizers to use computer and network to manage conference. With the development of the Internet and WEB technique, more and more conference management systems were developed, and the main tasks in managing a single conference has been standardized in such systems. Wikipedia's conference management system page [5] summarizes the typical functions and workflows supported by conference management systems. Besides the functionalities specified in Rick Snodgrass's survey, the functionalities include collecting reviewer's topics of interests, prevent conflict of interest, monitoring review coverage, etc. These functionalities further save the workload of conference management.

During the process of clarification on tasks in conference management system, researches on designing CMS showed up. In 2001, Stefano Ceri's paper "Designing multi-role, collaborative Web sites with WebML: a Conference Management System case study" [11] describes the modeling and design of the Conference Management System using WebML, a visual language. They used activity diagrams to illustrate the phases in a conference process, and used use cases to describe the roles in a CMS and functionalities

of roles. In structure modeling, they gave the basic objects and relationships between them, and further detailed the structure schema of these entities. They also used WebML tool suite to translate their design and made a prototype web-based conference management system which provide basic functions for conference management. In 2003, the paper “Developing multiagent systems: The Gaia methodology” [12] by Franco Zambonelli, Nicholas R. Jennings and Michael Wooldridge modeled conference management system as multiple Multi Agent System(MAS) organizations. Each organization corresponds to a phase in conference management procedure. And each such MAS can be viewed as being made up of agents that represent different roles (author, reviewer, chairs) involved in the process. This clarified the concepts of phase, roles and functions in a conference management system. Some later researches involved experiments to design and implement CMS using existing concept and modules. For example, the paper “Building Conference Proceedings Requires Adaptable Workflow and Content Management” [13] by Jutta A. Mulle, Klemens Bohm, Nicolas Roper and Tobias Sunder described their experience on building ProceedingsBuilder, a conference proceeding system. The ProceedingsBuilder has features of both content management system and workflow management system, and focused on submission verification workflow. Several problems were encountered during the use of this system for real conferences. The concept of combining content management and workflow management

proved not to be superior in designing and implementing conference management system. A major point from their experience is that the user requirements of conference management system were not well addressed. It shows how important the user requirements of a system should be analyzed before design and implementation. There were also researches on building conference management system and improving it. ConfSys is in such system developed since 1999. In Zhengwei Gu, Xin Jin and Bipin C. Desai's paper "CONFSYS: The CINDI Conference Support System" [14] of 2003, the sequence of events in conference was analyzed, and the design and implementation of ConfSys was presented. At the time, ConfSys has been implemented to provide adequate functions to manage a single conference. Since then, improvements were added to ConfSys. In 2004, the paper "CONFSYS : enhancements & integration" [15] by Yuwei Feng illustrates improvements on both functionalities and algorithms in ConfSys to manage the whole process of a conference. The paper [16] further clarified the concepts and functionalities in ConfSys. In [17], authors analyzed new requirements in multi-conference management, and the need for new features. These researches and experiments have recorded the experiences and lessons learnt in developing real CMS, and provide guidelines for other CMS developers.

Apart from these researches on designing conference management system, there were also several researches on algorithms used in conference management system. For

example, in [18], the authors illustrate their algorithm to extract semantic meaning from papers and allocating semantic similar papers to relevant sessions. This algorithm is useful for papers that have been submitted without topics or keyword. However, nowadays conferences require authors to submit their papers with topics. This makes the paper allocation easier and more precise, and other algorithms can be used in allocating of this kind of papers. In [15], Yuwei Feng describes an algorithm to automatically allocate papers to reviewers based on the topics of the papers and the reviewers' topics of interest and their bid on papers. This algorithm can give a good match rate on papers and reviewers on topics, and could save the time in manual tuning if one is needed. As reported in [15], the typical match rate is 100% if the reviewers have indicated their topics of interests and participated in the auction process. Without this input, the allocation can essentially be random.

Beside the researches on conference management systems, some of such systems have been implemented and used in real conferences.

2.2. Survey on conference management systems

The basic user roles in conference management system (CMS) include: chairs, program committee / reviewers, authors, and participants. Most of these systems can provide the basic conference management functions, such as receiving paper submissions, collecting reviewers' topics of interest and expertise, assign reviewers to papers,

collecting review results from reviewers, determining the acceptance threshold, informing authors of the final decision on their papers. Here we illustrate these concepts by survey of three conference management systems: EasyChair, Microsoft's Academic Conference Management Service, and ConfTool. Here we omit the features of ConfSys which has been reported in [15].

2.2.1. EasyChair

EasyChair is a web-based multi conference management system developed by Andrei Voronkov since 2002, when he was program chair of CADE and LPAR. It is then used in 2003 for LPAR and got recognized by many participants. EasyChair was improved and rewritten several times, and according to the author it is by far the most popular conference system. EasyChair was used by the Web conference, Bioinformatics Conferences, Computational Logic Conference, Algorithms conferences, and many other conferences - all together nearly 3,000 conferences and workshops.

EasyChair classifies its users into three main groups: chairs, reviewers, and authors. It provides the functions for chairs to manage and monitor the program committee, and to allow PC members and referees to setup their topics of interest. It allows authors to submit papers through the web interface. The system assigns these papers to PC members according to their preferences. Once the papers are reviewed, reviewers can submit their score through the system, and enable discussion of these scores. The review decisions are

sent to PC members, referees and authors through email, which can be monitored. In the author response phase, authors can respond to the reviews. EasyChair supports two models: the standard model is for conferences having single program committee, the multi-track version is for conferences that have multiple tracks and each track can have its own program committee.

EasyChair allows users to login once and act as different roles in conferences by selecting the preferred user roles in a certain conference. Its user interfaces are simple, but provide enough function to manage the process of multiple conferences. It focuses on functions for chairs and management group, and provides only basic functionality for author.

2.2.2. Microsoft's Academic Conference Management Service

The Conference Management Toolkit (CMT) is a conference management service sponsored by Microsoft Research. Its architect is Surajit Chaudhuri, a Research Area Manager at Microsoft Research. It is built using Microsoft technologies including Windows Server, ASP.Net [19], Internet Information Server (IIS) [20] and SQL Server [21]. CMT was first developed for ACM SIGKDD 1999. Since then, CMT has undergone several major improvements and extensions and has been used in over 700 conferences to date.

CMT defines three key roles in a typical conference: Chairs (e.g., General Chairs,

Program Committee chairs) who customize and control the workflow of the conference, Authors who submit papers, and Reviewers assigned by the Chair to review the submissions. It provides functions to three key roles as following: Chairs are able to setup initial parameters of a conference, setup and assign members to program committee, set up reviewing phases in a conference, perform auto or manual allocating papers to reviewers, track the process of reviewing, notify author of the final result, and other functions to manage the conference. Authors can setup his/her personal information, submit paper and enter related information such as title, abstract, the subjects, coauthors, etc, get notification of the review result, and upload camera ready version and slides if accepted. Reviewers are able to receive notification when he is added into program committee, login and setup personal information, setup topics of expertise and bid for papers, download and review the assigned papers and submit review result, reassign the paper to an external reviewer, start a discussion thread on paper.

Besides three key roles, CMT also support other roles to help manage the process of a conference. These roles include Associate Chair (Chair of a track in a multi-track conference), External Reviewer, Meta-Reviewers (coordinate reviewing of a set of papers), Proceedings Editor, Presentation Chair (coordinates presentations of accepted papers), Session Chair, and Presenters.

CMT manages a conference through the concept of clearly defined phases, and provide

a guideline of what actions should be done in certain phases. The Role-Action-Phase relationship is illustrated in its FAQ [22] as in Figure 2-2-1:

Role	Action	Phases					
		I	II	III	IV	V	VI
Conference Chair	Initial Setup**	■					
Authors	Paper Abstract Submission	■					
Authors	Paper Full Version Submission	■	■				
All Chairs*	Add Reviewers/Authors	■	■				
All Chairs*	Submit Paper for Author	■	■	■			
Reviewers	Set Bid/Preference		■				
All Chairs*	Assign Reviewers to Papers			■			
All Chairs*	Accept/Reject Decision				■	■	■
Reviewers	Submit Reviews				■		
Reviewers	Discuss/Finalize Reviews					■	
Conference Chair	Send Accept/Reject Notices to Authors						■

* Conference Chairs, and Associate Chairs.
 ** Edit Author Questions, Review/Meta-Review Criteria, Conference Details/Dates, Subject Areas

Figure 2.2-1 CMT's Role-Action-Phase relationship

CMT is a sophisticated conference management system that provides good amount of functionalities to manage single-track or multi-track conferences. Conferences managed by CMT are separated from each other and it's not easy to navigate from one conference to another. Its user interface uses too many icons instead of descriptive text, thus makes its learning curve slow.

2.2.3. ConfTool

ConfTool is another Web-based event management system that was developed to

support the organization of conferences, workshops, congresses and seminars. It was built using Apache [23] – PHP [24] – MySQL [25] platform, but also able to run under Microsoft IIS server. It has been developed since 2001, and was first used in 2004. Since then, ConfTool has been used at over 200 events and for different purposes.

ConfTool classifies user roles in conferences as Author, Co-Author, Presenter, Participant, Reviewer, Meta Reviewer, Conference Administrator, Conference Assistant, Conference Chair, Program committee chair, and Front Desk user. Here the Meta Reviewer is an expert who can access all other reviews. The Conference Assistant is to assist the Conference Chair managing the conference, and they can access part of the functions that belong to Conference Chair.

Like other conference management systems, ConfTool provides basic functions to suit the management requirement during different stages of a conference, such as accepting submission, bidding for review, auto assigning papers to reviewers, receiving review result, informing author the results, accepting final version, registering for the conference, etc. Besides, it implements some useful tools to facilitate the management of work, such as user/reviewer importing program, multi-file download tool, front desk registration function that often used for people who work at the registration desk like student volunteers. It's implemented as two versions: the standard version was designed for small events with up to 150 participants and offer a free license on request, the professional

version has been substantially enhanced to be more flexible and can support multi-conference with many participants.

ConfTool's user interface is equipped with multiple language option which allows users to select from 10 languages. It provides a complete set of documents for administrators and general users on system installation, term definition, and specification on functionalities for different roles. In addition, ConfTool provides a demo site for new users to experience the system. These facilities increase the degree of user satisfactory and make the system easy to use.

2.2.4. Comparison between CMSs

From the above survey, we have found that conference management systems have been developed on different platforms and techniques, from Linux to Windows, from ASP to PHP, from MySQL to SQL Server. However, they all used web-based three tier (Web Browser – Web Server - DBMS) mode in their implementation. This is because the World-Wide-Web has become the most commonly used facility to access and share information through the Internet. And its standardized protocol and widely supported software (browsers, web servers, and background support such as DBMS) provide flexibility to the end users and developers.

On user management, these systems classified the basic users of a conference into three main roles: namely Chair, Reviewer, Authors. Some systems detailed these roles

into Conference Chair, Program Committee Chair, Program Committee member, External Reviewer, Author, Co-Author, etc. Besides these main roles, other roles were extended, such as Administrator for system management, Participant for attending the meeting, Front Desk operator to help registration at the meeting.

Conference management systems provided functionalities to define and manage the different phases of the event, namely Call for Paper, Paper Collection, Paper Allocation, Paper Review, Paper Decision, Camera Ready Copy Collection, Program Arrangement, and Participants Registration. These phases are representation of different period in a conference. According to the roles and current phase, limited set of functions are assigned to users. Some of these functions are phase in-sensitive, such as user management, mail functions, etc. Other functions are phase-sensitive, and they should be only performed in certain period of time. For example, the uploading initial paper function for author must be performed before the review phase has started, and discussion function for reviewers must be after the review phase has ended.

Along the whole process of a conference, some key functions must be provided to manage the process of a conference. These functions include online submission of papers, setting up topics of interest and bidding for papers, assigning papers to reviewers, downloading of the assigned papers by the reviewers, submission of reviews, making decision on acceptance of papers and sending the decision to authors, online submission

of final version or camera ready version of papers, and participant registration. Other than these, functions such as group management, sending bulk mail, front desk registration could also help and facilitate the management of conference.

The table 2-2-1 compared the features of the above conference management systems, as well as the Confsys and redesigned ConfSys2. As we can see, that most of CMSs implemented sufficient functions for multi-conference management. ConfSys has its speciality on preventing conflict of interest. ConfSys2 not only implement the existing functionalities of ConfSys, but also introduced new features in conference management.

Features	Systems	EasyChair	Microsoft CMT	ConfTool	ConfSys	ConfSys2 ¹
Publishing call for paper		External	External	External	External	Integrated
Paper submissions		Y	Y	Y	Y	Y
Collecting reviewers' topic preferences		Y	Y	Y	Y	Y
Prevent conflicts of interest		Manual	Auto + Manual	Manual	Auto + Manual + DBLP	Auto + Manual + DBLP
Assigning reviewers to papers		Y	Y	Y	Y	Y
Automatic Paper allocation		Y	Y	Y	Y	Y
Monitoring auto allocation match rate					Y	Y
Collecting reviews		Y	Y	Y	Y	Y
Monitoring review coverage				Y	Y	Y
Monitoring review progress		Y	Y	Y	Y	Y
Sharing reviews		Y	Y	Y	Y	Y
Separate Paper review discussion		Y	Y	Y	Y	Y
Ensuring independence of reviews				Y		Y
Ranking reviews and setting acceptance threshold		Y	Y	Y	Y	Y
Anonymizing submissions		Y	Y	Y	Y	Y
Anonymizing reviews		Y	Y	Y	Y	Y
Reporting reviewers comments and program committee decision to authors		Y	Y	Y	Y	Y
Collecting final accepted versions		Y	Y	Y	Y	Y
Session arrangement		Y	Y	Y	Y	Y
Program generation		Y	Y	Y	Y	Y
Conference Registration			Y	Y	Y	Y
Multiple Conference Management		Y	Y	Y		Y
Multiple track management		Y	Y	Y		Y
Global user management		Y	Y	Y		Y
Uniform user interface		Y				Y
Supporting paper versions						Y
Flexible role/group management						Y
Role function management						Y
Conference homepage management						Y
Integrated message publication						Y
Flexible review definition and collection						Y

Table 2.2-1 Comparison on conference management systems

¹ ConfSys2 would be re-named to ConfSys to show its provenance.

2.3. Our Contribution

From the literature review, we found that there were researches and exercises to apply different concepts, such as content management and workflow management, in developing conference management systems (CMS). Those researches helped clarifying conference management concepts and tasks, such as roles, phases and functionalities of CMS. However, these experiments have not implemented a completed workable conference management system. Several issues were found in these experiments, but there were not further researches to address and solve the issues and improve the prototype systems. On the other hand, several other CMSs, such as the ones we surveyed, have been implemented and are suitable for the needs of conference management. However, due to the lack of “academic appeal” there is a lack of literature reporting on how these CMSs were implemented.

ConfSys is one of the earliest integrated web-based conference management systems. Its development is not to experiment with concepts, but based on concrete conference management requirements. The ConfSys development supervisor is Dr. Bipin C. Desai, who is also the general chair of IDEAS and C3S2E. His routine work on conference management provided the concrete requirements for advanced conference management system. The development of ConfSys was based on these requirements. From 1999 to

2005, the major functionalities for managing a single conference had been implemented in ConfSys, and were used in conferences without problem. However, the old database structure and software architecture prevent ConfSys from being further developed, and redesign of ConfSys is required.

Our work on ConfSys2, the redesign of ConfSys, will contribute to the development of conference management system from several aspects. First of all, we generalized the concepts and tasks of CMS based on concrete requirements from routine conference management work. These concepts and tasks will be illustrated in chapter 3 and 4. They can be further used to analyze the adaption from other concepts to CMS. Second, we formalized the CMS as a “Time limited role controlled function based system”. Through the milestone setting, the phase is defined. And through user-role-function mapping, the access privilege to certain functions has been controlled. Third, we introduce new concepts and new features of conference management system, such as conference series management and community mode. These concepts and features could have been used in other applications, such as virtual society systems, but have not formally introduced in conference management system development. These new features are listed in table 2-2-1, and will be further illustrated in chapter 5. Last but not least, we have fully implemented ConfSys2, which meets the requirements of conference management and is ready to be used in conferences. The mechanism of how ConfSys2 is implemented can be further

analyzed and used in development of similar systems, such as content evaluation systems.

The user roles, the conference phase, and the corresponding functionalities are the basic elements of a conference management system. In the following chapter, we will describe ConfSys, a web based single conference management system. We will analyze how those basic elements are incorporated into the system, its features, its limitations, and new functional requirements proposed.

3. Features of ConfSys

ConfSys is a conference management system developed under the supervision of Dr. Bipin C. Desai of Concordia University since 1996 when a need was felt for a Web based system to support the inaugurated IDEAS series of meetings. It is built using the web technique and Tomcat – Java Servlet/JSP – MySQL platform. It enables the system admin setting up the parameters and provides the functions for chairs, reviewers, and authors to support managing the whole process of a conference. In this chapter, we will examine ConfSys and analyze its features and structure.

3.1. History of Confsys

The development of ConfSys follows the developments of tools and technique in the web world. As new tools and technology became available, ConfSys used them to improve the functionalities required to manage the administrative task of organizing an academic conference. The first version of ConfSys was implemented by Dr. Bipin C. Desai in 1999. It was built on ODE (Object Database and Environment) [26], and used to manage the conference IDEAS 1999 and IDEAS 2000. In the following development of ConfSys, its DBMS was changed to Oracle and then to MySQL. The web request was handled with Java Servlet, and later introduced JSP to facilitate the user interface development. In the version of 2003, the Tomcat – Java Servlet/JSP – MySQL platform is

used for the development of ConfSys. And since then, the successor versions of ConfSys have been built upon this platform.

Since 2000, ConfSys has been used as the official conference management system of International Database Engineering and Applications Symposium (IDEAS). On 2008, it was adopted by C* Conference on Computer Science & Software Engineering (C³S²E) and its poster session. ConfSys, though a work in progress and not made publicly available, have also been used by several other conferences, such as Digital Hospital 04, Real-Time Computing Systems and Applications (RTCSEA), DCA 07, etc.

Many people have contributed to the development of ConfSys. They include the project supervisor Dr. Bipin C. Desai, developer W. Yang, Hui Guan, Zhengwei Gu, Xin Jin, Yuwei Feng. They analyzed the requirements and concepts in conference management, and incorporate the experiences of using the ConfSys. Many functions and features have been added to ConfSys to suit the needs of conference management work. Those functions and features will be presented in the following section.

3.2. Structure and features of Confsys

ConfSys was built using a Web-Based Three Tier Mode [27], which is illustrated in Figure 3-2-1.

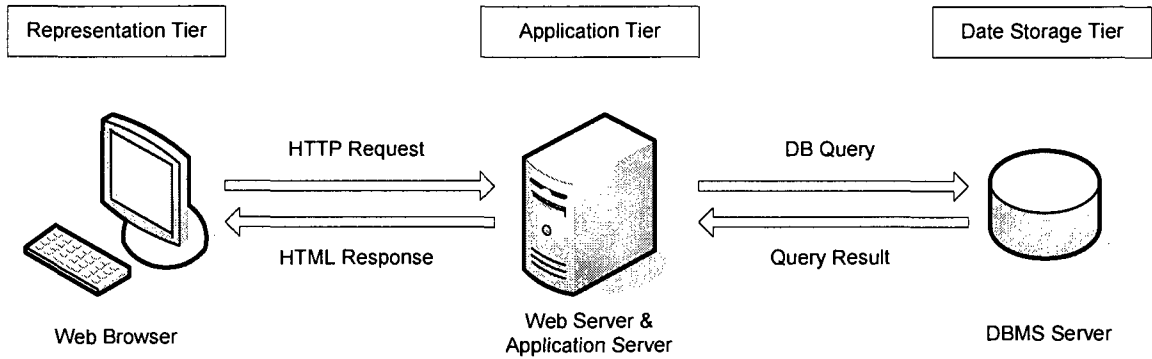


Figure 3.2-1 Web-Based Three Tier Mode

The Three Tire Mode includes representation tier, application tier, and data storage tier. The representation tier provides the user interfaces to accept input and show result to end users. In Web-Based application, the representation tier is located at client side, and normally a web browser is used to act as this tier. The application tier is the centre of the system. It accepts and processes requests from the representation tier, manipulate data stored in data storage tier, and return the result to representation tier. This tier normally contains the programs hosted on a web server and application server. The data storage tier is used to store user data and system data required for the execution of the system. A data base management system (DBMS) is normally used for the data storage in this tier.

In ConfSys system, the representation tier is the web browser located at user's computer, which has an internet connection to access the ConfSys web site. The application tier use Tomcat as the application server, with the assistance of Apache to provide static contents. The ConfSys is implemented using Java Servlet and Java Server Page (JSP) and execute on Tomcat server. At data storage tier, MySQL database system is

used to store required data.

To protect the application security, ConfSys implements two level of protection: communication level, and application level. At communication level, ConfSys uses HTTPS [28] protocol to communicate between the representation tier and application tier. HTTPS transfers HTTP protocol over Secure Socket Layer (SSL) [29], so the user requests and server responses are encrypted when they are transferred on the Internet. At application level, all users are required to use their unique username and password to login to the system. And then HttpSession on server side is used to track the user status, and ensure the user access to only assigned functions. Those two levels of protection prevent the information leak on network and illegal access by unprivileged users, thus make the ConfSys a safe system.

3.2.1. Users of ConfSys

ConfSys classifies users into six roles: Administrator, General Chair, Program Chair, Reviewer/Program committee member, and Author.

Administrator is responsible for initializing the database and installing a new instance of ConfSys for a new conference, to setup system parameters, to setup the user account

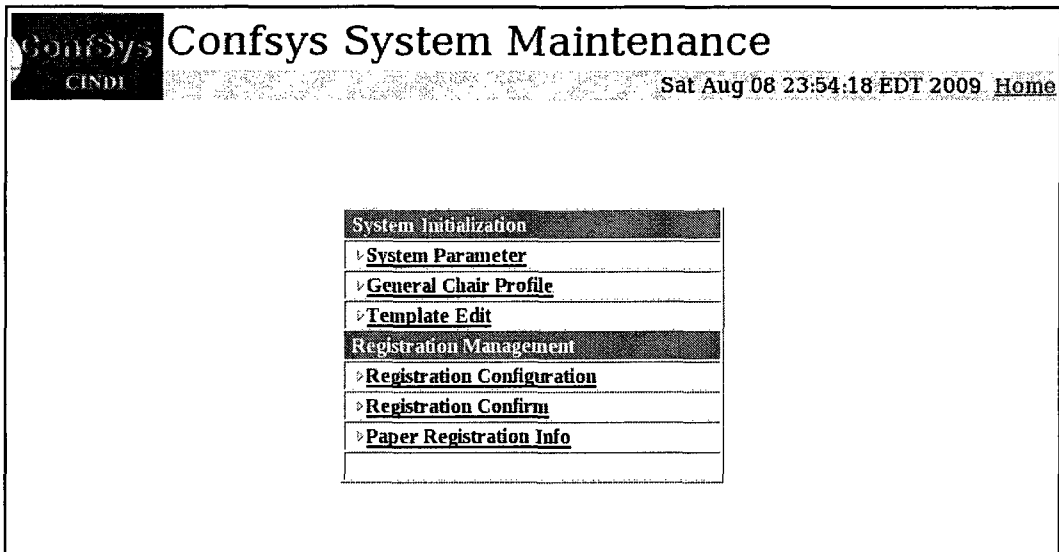


Figure 3.2-2 Admin Main Menu

for General Chair. Figure 3-2-2 illustrates the main menu of administrator.

General Chair (GC) is in charge of general management tasks during the entire process of a conference. GC is able to setup conference related parameters, add users to program chair and reviewer group, monitor the review process, setup and participate in the process of making decisions on papers, manage conference registration process, and arrange conference program. Figure 3-2-3 illustrates the main menu of GC.

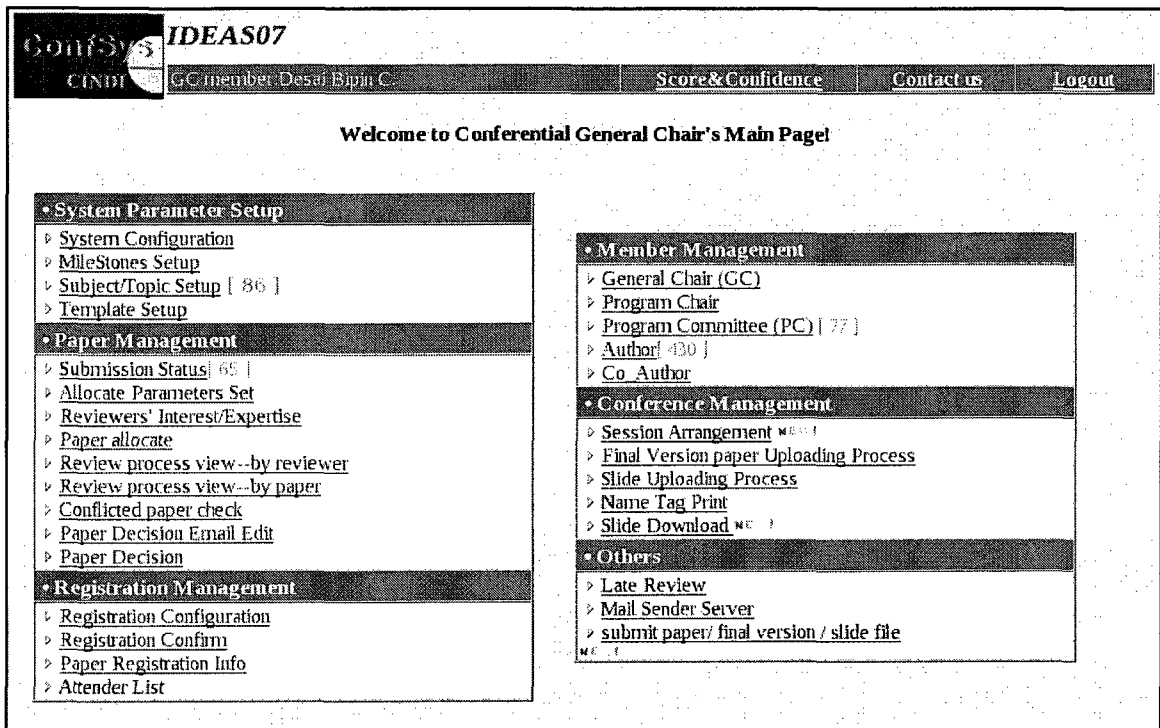


Figure 3.2-3 General Chair Main Menu

Program Chair (PC) is responsible to manage the program committee members and review process. PC has the ability to setup paper allocation parameters, to perform auto/manual allocation papers to reviewers, to monitor the review process, to setup paper judging criteria, and to make the final decision of each paper.

Reviewer / Program Committee Members are nominated by general chair and program chair. They are able to login and setup their profile and preferred topics, download the papers that are allocated to them, submit the review result, and join debate when there's conflicts on review.

Author is the largest group in a conference. Author can sign up for a conference and login to submit papers. They can set up paper title and abstract and co-authors, and

upload paper file. After the decision on the paper is made, they are able to view the review result and comments on their paper. When the paper is accepted, they can register for the conference; upload the camera ready copy and slides for the presentation.

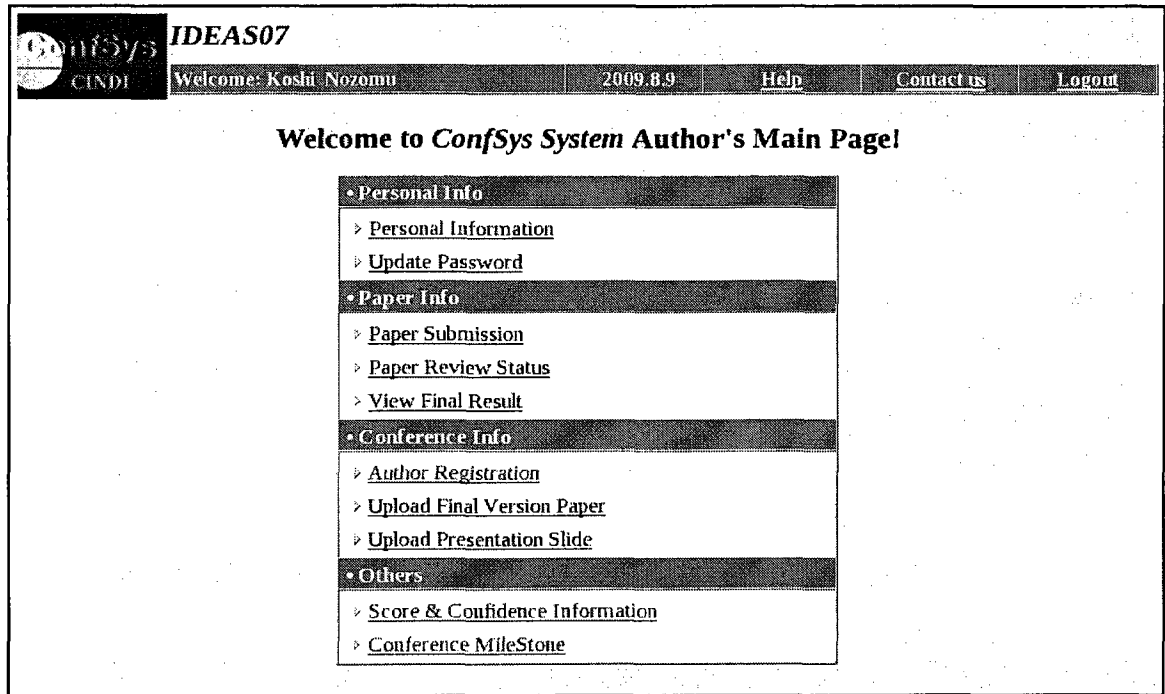


Figure 3.2-4 Author Main Menu

Participant is the group of user who want to participate in the conference meeting. They can register for the conference through the ConfSys and provide participants information and make on-line payment.

3.2.2. Phases and Functions Supported by ConfSys

ConfSys support phase management to manage the different period during a conference. Those phases are controlled through setting up of milestone by general chair.

The phases include:

Setup Phase is the phase before a conference starts. In this phase, administrator can setup a new instance of ConfSys, setup system parameters and GC account. General Chair is able to set up conference parameters.

Call For Paper Phase is the first phase of a conference. General Chair can add users to program committee group, and setup program chair. Authors can sign up for the conference and submit papers. Reviewers can set their preferred topics for paper review. All users are able to login and setup their profile.

Auction Phase is the phase start after the CFP is over. General Chair and Program Chair can setup paper auction parameters. Program committee members/reviewers can perform the auction for the paper they want to review.

Allocation Phase is the phase when General Chair or Program Chair use auto/manual allocation function to assign reviewers to papers.

Review Phase is the phase when the paper review is in progress. In this phase, program committee members and reviewers are able to download the papers assigned to them and submit their initial review result. Chairs are able to check for the controversial paper reviews and setup a paper debate. Author can check the reviewing progress of his paper.

Debate Phase is required since there are always controversial reviews for some papers.

In this phase, reviewers are able to debate on their position with respect to such papers in order to arrive at a consensus. All final review result should be submitted before the end of this phase.

Final Decision Phase is the phase when General Chair along with the Program Chair collect all final review result and make decision on acceptance of papers. After the final decision is made, reviewers and authors are able to see the final decision. The program chair could use ConfSys to arrange a draft program which could be fine tuned to the final program.

Registration Phase is after the Final Decision Phase. In this phase, participants are able to register for the conference. This phase can be detailed in to early registration and normal registration. The early registration is mainly for author with accepted papers to reserve their seats in conference, and they can upload camera ready copy of the paper and slide for presentation. The normal registration is for other participants to register to join the meeting. In registration phase, chairs are able to confirm the registration applications.

3.2.3. Other Features of ConfSys

ConfSys has some special features to control the system and can provide more convenience in managing conference. They are given below:

Parameters Management: ConfSys has more than 60 parameters to allow administrator and chairs to control how the system works. Some key parameters include

PAPER_TYPE to control the file types that author can upload, PAPER_SIZE to limit the size of a paper that can be uploaded, DIR_PAPER to direct the system where to store the upload papers, etc.

Email Templates: ConfSys enable chairs to edit a number of email templates to simplify the management work. Those email templates can contain parameters to suite the needs of different situations. For example, \$CONF_NAME in email templates will be replaced with the actual conference name when the email is sent out.

Automatic Paper Allocation and Avoiding Conflict of Interest: This function is based on several factors: allocation parameters, paper auction by reviewers, papers' subjects and reviewers' topics of interest and expertise. The automatic paper allocation program analyzes these factors and assigns papers to reviewers to meet the requirement, such as: limit the number of papers for a reviewer, limit the number of reviewers for a paper, avoid conflict of interests, etc. DBLP [30] is a database that provides bibliographic information on major computer science journals and proceedings. ConfSys uses DBLP to check the relation between authors and reviewers to avoid assigning a paper to reviewers who may have been coauthors in the past.

Automatic Session Arrangement: This function help chairs to arrange the programs of the conference. It can arrange the sessions based on its length (full, short or poster paper) and its topics, so the papers in a session have similar size and focused on similar

topics.

Integrate with CINDI System: The Concordia Indexing and Discovery System (CINDI) [31] is a digital library build with user-friendly interface with which resource providers or contributors can “publish” their resources. As a sub-system of CINDI, ConfSys is able to merge the submitted paper and its related information into CINDI after the call for paper phase is ended.

ConfSys, in its various versions, is functional and has been used in production since 1999. It illustrates the concept of Role-Phase-Function in conference management. It provides good number of features and tools to help organizers be able manage the entire process of a conference from call for papers to the actual support of the meeting by providing a central site for all presentations. However, ConfSys also has some limitations.

3.3. Limitations of Confsys

During several years of use and improvement on ConfSys, this system has become stable. However, several limitations of ConfSys have shown up as given below:

Single conference management: ConfSys was designed to manage the whole process of a conference, and it was implemented as a single conference management system. It cannot manage multiple conferences or a multi-track conference in a single installation of the system. Every time a new conference starts, a new instance of ConfSys is required to be initialized, and a new database is required to be manually created and initialized.

Normally, this is done by dumping the database from last conference term and then importing it to a new database and making proper modification to suit the needs of the new conference. These works, though automated by scripts, nonetheless require the involvement of database administrator, and manual operations are time consuming and error prone.

Multiple role login requirement: ConfSys provides a security mechanism to protect the system and different roles requires the use of different username and password to login to the system and access the corresponding functions. Unfortunately, the implementation of this mechanism is not flexible. First of all, different roles require the login from different login interface. Second, each such login grants access to a limited function associated with the role. To switch to another role, a user must quit the current role and re-login for the new role. This is inconvenient for users who have multiple roles, such as author and program committee members etc.

Lack of internal email support: ConfSys has a mail system which uses the Internet Mail as the means to communicate between users, but it does not store these messages inside ConfSys system. When users have some questions about the past message or need to reference mails, they can not access those messages directly from inside the ConfSys. They have to use their own Internet mail box to check the mails from ConfSys. This makes the communication between ConfSys users inconvenient.

Fixed Menu System and Hardcoded Security Management: ConfSys provides every role a fixed set of functions to get the job done. This in general is good enough for routine tasks. However, when a new function is added to the system or some functions are needed to be assigned to a certain user role, it can only be done by modifying the menu page for that role. In addition, the role privilege check is hardcoded into the function page, when a function is assigned to a new role, that function page has to be re-programmed.

Lack of Log System: ConfSys does not have a log system to record the important operations done by users, such as submitting papers, download papers, registration, etc. The log information is useful to analyze the performance of the system, and could be used to find the root of problems caused by unusual sequence of operations.

Lack of Help System: ConfSys provides some help information on several pages that perform complicated functions. However, help information does not generally exist on functional pages. When users do not know the exact function of a page, they make guesses and could make mistakes, or they write to organizers to get help. This increases the workload of organizers. Because those help information are hardcoded into the web pages of the function, any modification to the help information need to done in source code of the program and this may accidently introduce errors in these programs.

3.4. New requirements and obstacles

In addition to these limitations found during the use of ConfSys, new user requirements are proposed to implement in fourth generation conference management system. These main requirements include: Multiple Conference and Multi-Track Management, Uniform user interface, Internal Mail Support, Flexible Menu System and Security Management, System Log, and Help System. When trying to implement those additional features in the current version of ConfSys, many obstacles were encountered including the inadequate of database structure, software architecture, and user interface.

3.4.1. Multiple Conference and Multi-Track Management

ConfSys was implemented as a single conference management system. To manage multiple conferences or a multi-track conference, multiple instances of ConfSys are required to be installed. This is inconvenient for system administrators and conference organizers. Multiple conference and multi-track management was proposed to incorporate into ConfSys. This means that with a single installation, the system would enable administrator to create multiple conferences and create multiple tracks in a conference. Each conference or track can have its own set of parameters, and its own set of users and management groups.

To achieve this object, a set of tables is required to be added to the ConfSys database to store the basic information of the conferences and tracks. Besides, almost all tables in

original database are required to add a new column to indicate which conference the information belongs to. Furthermore, because ConfSys programs use arrays and basic Java objects as the media to communicate between different program modules and make the connections between program and the base database structure tight, all source code are required to be modified to suit the modification of the database structures. The workload for this improvement would be huge.

3.4.2. Uniform user interface

ConfSys requires different user roles to login from different web pages to get access to the relevant functions. Switching between roles can only be completed by logging-off the current role and re-login with a different set of username and password. This is inconvenient for users who have multiple roles in a conference. A uniform user interface was proposed to be implemented in ConfSys. In the uniform user interface, all users are supposed to login to the system through a single login tunnel. And once logged in, with this single user account, a user is able to switch roles freely without logging off. Furthermore, in a multiple conference management system, user should be able to access different conferences and switch user roles during a single login.

To implement such as system, a centralized user database structure is required to store the user information and their roles in different conferences. And application level should have a corresponding module to handle the role switch. However, ConfSys uses multiple

DB tables to store the user information for each role. To allow user login through a uniform tunnel, the program must check all those tables. Besides, one single modification on user information, such as add a new column for user information, will result in modifications on all those tables, and this in turn require modification to the application level. This entailed a lot of programming, and would result in error prone program.

Apart from the database structure, ConfSys does not have a centralized user management module to handle the role switch. A new module must be implemented to achieve the uniform user interface. If this module is based on the existing database structure, it could magnify the drawback brought by the multi user role tables.

3.4.3. Internal Mail Support

ConfSys uses the external Internet email system as the media to enable communication between users. No mechanism has been implemented to store those messages within ConfSys. Users have to use external email software to access information from conference, which is inconvenient. An internal mail system is proposed to facilitate browsing and organizing ConfSys messages.

Creating a new database table to store these messages and implementing programs to handle them will involve the connection to user tables. However, as mentioned before, ConfSys database uses multiple user role tables to store user information, it is not practical to link messages to all those tables. The existing database structure made the

implementation of this feature difficult.

3.4.4. Flexible Menu System and Security Management

In ConfSys, fixed menu system is used for different user roles. After a user successfully logs in to a certain role, a hardcoded menu will be presented to the user to allow the user to access a set of functions. It is natural that sometimes chairs want to share part of his work with assistants. This can only be done by modifying the source code of the menu of that role or make a new page to lead to a limited set of functions. Also, the security check for those functions is also required to be modified to suit this modification, which again is error prone. A new flexible menu system was proposed for ConfSys to allow chairs to easily share his work with associated chairs.

Implementing this feature involves several basic requirements from the system. First of all, a User-Role-Function mapping is required to be stored and modifiable by privileged users. Second, a flexible security management module should be implemented to ensure privileged access by analyzing this mapping. Third, an automatic menu generation module is required to extract the function menu from this mapping, and generate the menu for current user. ConfSys does not have any of these modules. Its hardcoded security check mechanism means that almost all functional programs are required to be replaced ones that would be suitable for this feature. And because of the multiple user role tables, implementing the User-Role-Function mapping is impractical.

3.4.5. System Log

System log records important operations by users. It's important for administrator to monitor the status of the system and could be a good tool to solve problems. Several of ConfSys programs does write debug information to Tomcat's log, but this information is limited and not easy to use because it is merged with other Tomcat logs. A log system that records user operations and system status is proposed to be implemented for ConfSys.

To implement this feature, a table is required to be added to the existing database, and a log module is needed to facilitate logging operations, such as writing log and search for a log according to given conditions. The log table should record the detail information about the operation being logged, and along with the ID of the user who performed the operation. However, the ConfSys use multiple user account for a natural person and implement multiple user role tables in database; it is not easy to link a log item to all these tables, and the links could be unnatural. Implementing this feature based on existing structure of ConfSys could introduce a large workload.

3.4.6. User Help System

ConfSys provides help information on some complicated function pages, but help information are not generally available. It should be an advantage that every function page has its own help information. Since ConfSys does not have a uniform user interface, help information or the link to help pages can only be manually added to each function

page. And ConfSys uses both Java Servlet and JSP to generate the user interface, the work of adding help to page is heavy and error prone.

Due to ConfSys's dated database structure and software architecture, the workload of adding new features to ConfSys is large, and the work is error prone. Some features would be very hard to implement using the existing framework, and the solution could be unnatural. As a result of these obstacles, we felt a need to redesign ConfSys including its database structure, software architecture, and user interface. As such, ConfSys2 is the advanced redesign of ConfSys.

4. ConfSys2: an advanced redesign of ConfSys

Before redesigning the ConfSys, many factors were taken into consideration. First of all, the new design should support the existing functions and features of ConfSys and proposed features to eliminate its limitations, and a better user interface is required. Second, the new software system should be easy for developers to maintain. Third, the new design should be flexible enough to meet the requirement of further development, which means there should not to be needed to make fundamental changes when adding new features.

To meet these expectations, we analyzed the functionalities and features of ConfSys, and redesigned its database structure, software architecture, and its user interface.

4.1. Database structure redesign

We firstly analyzed ConfSys's database structure. ConfSys uses a single database implemented with MySQL on each instance of its execution to store the system and user data. By reverse engineering the database of ConfSys, we get a map of tables existing in ConfSys. Figure 4-1-1 and Figure 4-1-2 illustrate these table maps.

4.1.1. Analysis of ConfSys database structure

The table map shows that there are five main types of tables in ConfSys. They are user tables, paper tables, conference arrangement and management tables, parameter and

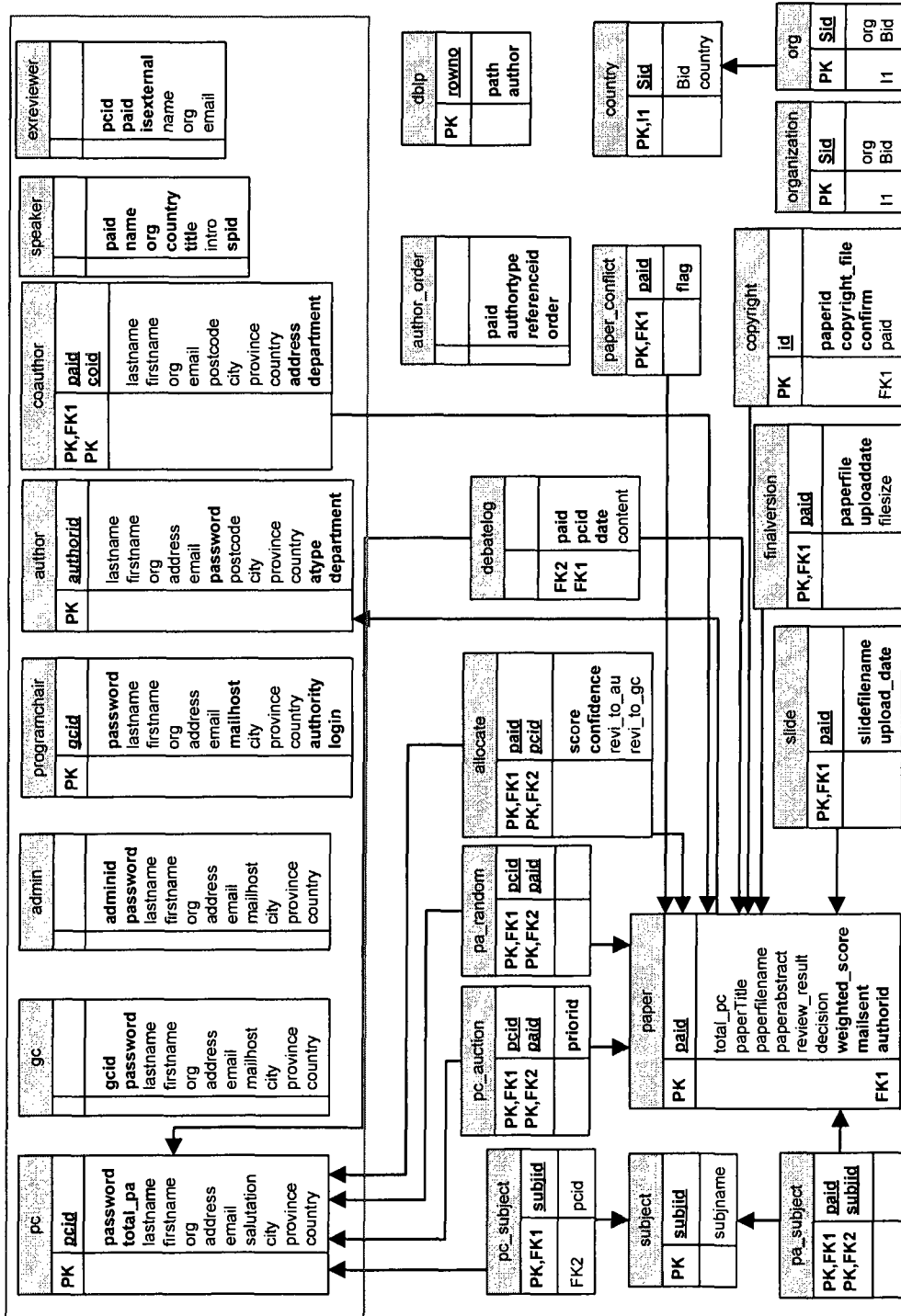


Figure 4.1-1 Existing ConfSys Database Structure - 1

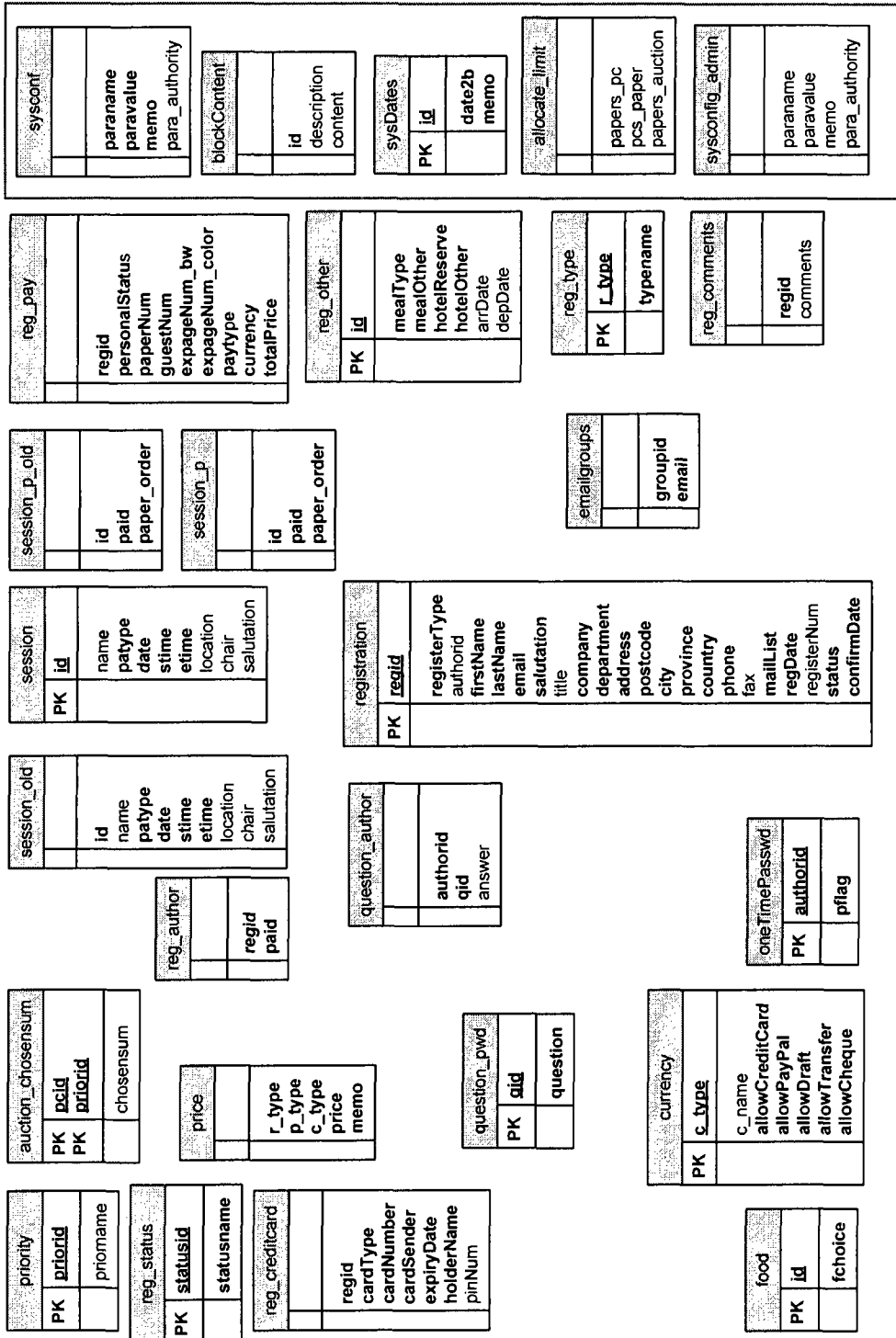


Figure 4.1-2 Existing ConfSys Database Structure - 2

template tables, and accessory tables.

User Tables in ConfSys are a set of tables to store user information for different roles. They are used to authenticate users during login, and for contacting users. As illustrated in Figure 4-1-1 and 4-1-2, they include pc, gc, admin, programchair, author, coauthor, speaker, exreviewer. The registration table also contains information of participants. We can find that those tables contain similar set of information, such as user's name, address, country, etc. Some tables have foreign keys constraints: for example paper table is link to the author table to show that the relationship between an author and her paper. This design has disadvantages. When the system want to add a new type of information to users, all user tables are required to add the same column. And these modifications will expand to data access modules, thus greatly increasing the development workload. We actually encountered this problem when trying to add organization information to users.

Paper Tables are a set of tables that store paper information, and operations that have been done on papers. This type of tables includes paper, pa_subject, slide, final version, copyright, etc. These tables store the basic paper information, paper subjects, and digital files associated with the paper. Other tables such as pc_auction, pa_random, allocate store the user operations on papers. For example, pc_auction stores the reviewers' auction information on paper, and allocate table stores the information about the allocation of papers to reviewers.

Conference Arrangement and Management Tables store the information related to conference arrangements and management. They are used to arrange the sessions of the actual meeting, and to manage the registration of participants. Table session, session_p are used to store the information of session arrangement and papers in the sessions. The table reg_type, price, currency store the available register options for potential participants. Once registered, the table registration records the participant information. As we can find, that the registration table include some redundant information of users. Because most participants are conference system users, their information should be in user tables.

Parameter and Template Tables store the system and conference parameters, templates, allocation settings. These tables are shown in the box at the right side of figure 4-1-2. They include sysconf, blockContent, sysDates, allocate_limit, sysconfig_admin. They are used to control the execution of the ConfSys and the management of different phases of conference.

Accessory Tables are those tables which do not have direct connection with users, papers or system control. However, they are required for certain function of ConfSys. This type of tables includes DBLP, country, and organization. Table DBLP is used during paper allocation to check an author and review relationship, and avoid assigning paper to reviewers who were co-authors in the past. The country and organization tables are used

to standardize the user input when they fill up their personal information.

Those five types of table form the database structure of ConfSys. The two major disadvantages of this structure include:

Redundant structure and information. Many tables have similar structure and store redundant information. For example, the multiple user tables to store different roles, and template tables which have similar structure.

Lack of flexibility. The fixed role-table relation makes it hard to expand. For example, if a new user role is required to be added to the system, it cannot be fitted to any tables. A new table must be created to suit this requirement.

Beside these disadvantages, the existing database lack tables to support the proposed new features, such as multiple conference management, log, internal mails, etc. A new database structure is designed to eliminate the disadvantages and to support the new features.

4.1.2. Redesigning database structure

To meet the requirement of database structure to support existing functionalities and proposed new features, as well as to prepare for the further changes, we analyzed the user requirements and summarized basic concepts in a multi conference management system and the database structure to support those concepts. The concepts include: System Administration, Multi Conferences Management, User Management, Function

Management, User-Role-Function Mapping, Parameter and Template Management, Paper Management, Session Arrangement, Registration and Payment Management, and Accessory Support.

System Administration: System administrator should have the facility to install the system and control how the system works. This is normally done by enabling administrator to set up system wide parameters, and enable the program to read those parameters and abide the execution of the program by the settings.

Multi Conferences Management: System should be able to store the information (such as the conference name, date, location, etc.) of multiple conferences, and allow privileged users to access and modify the information. Besides, conference related information such as paper, roles, and conference wide parameters should have an indication of which conference the information belong to.

User Management: The concept of “user” represents a real person who uses the system. The system should be able to store users’ personal information, and record the user status, such as which conference the user joined, what roles the user act, etc.

Function Management: The functionalities provided by the system should be managed centralized. This mean a set of data should be used to store the information about the functions and where to find and execute the functions.

Security Management and User-Role-Function Mapping: Security management

involves privilege checking, in which user can only access functions that have been assigned to the user. Access to unassigned function should be strictly prohibited. A simple user-function mapping may achieve this requirement, but this type of mapping introduces too much workload on assigning functions to users and hence impractical. The concept of role is introduced in this situation to simplify the work. A role is a virtual concept to define which set of function can be accessed by whom. Assigning a role to a user is often done by adding the user to a certain group. For example, when a user is assigned a role reviewer or is added to the reviewer group, the user should be able to access the functions assigned to “reviewer”. Security management is usually implemented by checking the user-role-mapping to decide if a certain user has access to certain functions.

Parameter and Template Management: Every conference could have its own set of parameters or templates to control the process of the conference. Such parameters may include the conference name, phase dates, subjects, etc. The system should be able store those parameters and templates, and provide the facilities for privileged user to modify them. To enable referencing these parameters and templates, a centralized parameter and template management mode would be required.

Paper Management: Paper management is the most important part in a conference management system because almost all actions are related to papers. The system should be able to store paper information and status of operations on papers. The paper

information could include the basic paper information (such as title, abstract), and associated information about papers (such as related file information for the papers). The operational status information for papers would include reviewer auction information, paper allocation, paper review process, debate details and review result, etc. Paper management represents the status of papers at different phase of the conference.

Session Arrangement: Session arrangement enables conference organizers to assign similar papers into sessions, and plan on session time and session chairs.

Registration and Payment Management: Registration normally has multiple options for participant to choose, such as the status, number of papers, extra pages, guests. Based on the registration date, the options may change. This part of management should be able to store such options, method to pay for the registration, user's choice, and the status of participant's registration.

Accessory Support: Accessory information is not directly related to a conference, but is required to support the proper execution of certain functions. For example, country and organization information are required in user personal information management to standardize the user input. DBLP information is used to analyze the relationship between users.

Based on these concepts and required features, a new database structure is designed to support the data storage in ConfSys2.

4.1.3. ConfSys2's new database structure

ConfSys2's database structure consists of one main database and a number of conference databases, as illustrated in Figure 4-1-3. The main database stores system wide data, such as system parameters, user, functional contents, system wide accessories, etc. The conference databases store conference specific data. Each conference database contains data for a conference series, which include terms/tracks in the conference, papers, conference status, session, etc.

There are three main factors that were considered for using multiple database structure: database performance, error separation, and further expansion.

Database Performance: a conference user's operation normally concentrates in a single conference at a given time. Separated conference database can eliminate the performance/speed loss when the operation involves non-relevant conference data, which happens when using a united multi-conference database.

Error separation: The separated conference databases limit the operations of a conference to a single conference database. This avoids mis-operations on one conference affecting other conferences.

Future Expansion: Though the current design is for managing limited number of conferences, it is possible to expand the system to manage large amount of conferences. The multi conference database design retains the possibility of using distributed remote

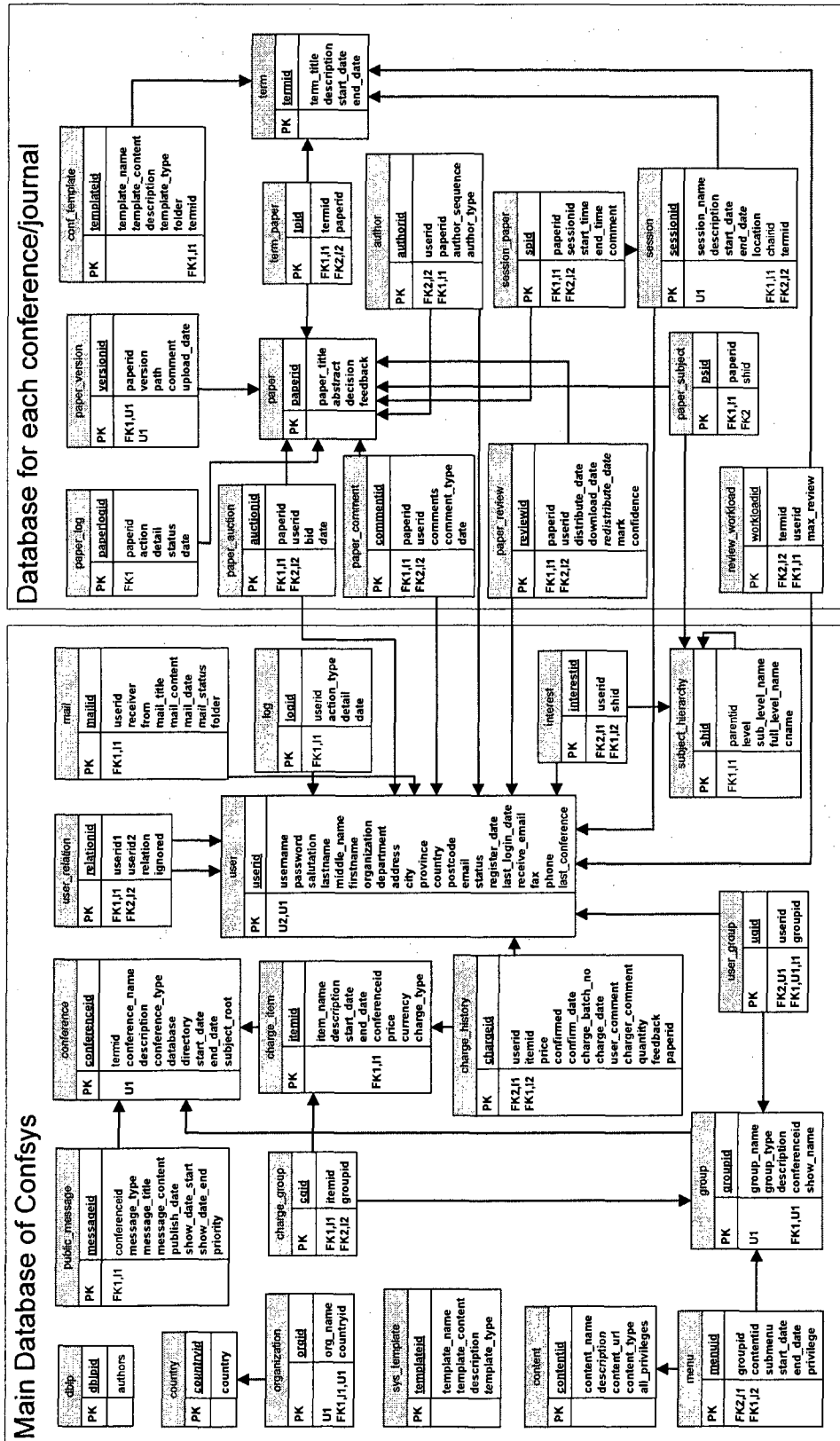


Figure 4.1-3 ConfSys2 Database Structure

database system to manage the conference databases without need for large modifications of the database structure in the future.

In the main database, there are mainly five groups of tables: System Parameter, Conference, User-Group-Function Mapping, Registration & Payment, and System Wide Accessory.

System Parameter: the table *sys_template* is used to store system wide parameters.

Conference: the table *conference* is used to store the information of conferences managed by the system. Conference information include conference title, description, conference type (conference or journal), the conference database, etc.

User-Group-Function Mapping: the table *user* stores user's basic information. Table *group* stores groups/roles in a conference. Table *content* contains functional contents managed by the system. These three tables, along with the table *user_group* and *menu*, define the user-group-function relationship, which can be used to support the security management and flexibility menu system.

Registration & Payment: the table *charge_item* and *charge_group* define the registration options in a conference and the options' applicable groups. Table *charge_history* contains the registration information of a user, and the status of the registration.

System Wide Accessory: this type of tables store function specific information. Table

country and *organization* are used to standardize the data during entry of the user personal information. Table *dblp* and *user_relation* are used to identify user relationship, which is used for conflict check in paper allocation. Table *public_message* is used to store public accessible messages, such as news, due dates, events, etc. Table *mail* is to support internal mail system. Table *log* stores important log information of user operations and system status. Table *subject_hierachy* and *interest* store the users' topics of expertise information.

The conference database is responsible for storing conference specific information, which can be classified into four types: Term/Track, Paper, Review Status, and Session.

Term/Track: A conference may contain multiple terms/tracks. Table *term* defines terms/tracks in a conference, and *conf_template* stores parameters and templates in these terms/tracks.

Paper: Papers are the core objects in a conference. Table *paper*, *term_paper*, *author*, and *paper_subject* store the basic information of papers. Table *paper_version* is used to maintain multiple versions of files for a given paper. Table *paper_log* logs the operations done on a paper.

Review Status: The main review operations include paper auction, paper allocation, debating, and submitting review result. Table *auction* contains the paper auction information and the conflict of interest. Table *paper_review* stores the information of

paper allocation and the review result. Table *paper_comment* is used to store the comments of paper. To give the flexibility for individual reviewer's workload, the table *review_workload* defines the max number of papers that can be assigned to certain reviewers.

Session: table *session* stores the session arrangements in a conference term/track, and *session_paper* contains papers in sessions.

4.1.4. Comparison between old and new database structure

Compared to the original ConfSys database structure, the new design contains smaller number of tables while providing data storage support to the existing and proposed features, as well as more flexibility. The number of tables is reduced from 51 to 33. This is achieved by generalizing the concepts in conference management. For example, we use user-group mapping to replace the original fixed user tables, and we use a unified parameter and template table to replace the original multiple uni-functional template tables. The reduction in table numbers would also reduce the difficulties in programming. Furthermore, the new database structure provides more flexibility because of the generalizing of concepts. For example, user and roles information are now stored in user-group mapping. To add a new role/group in the system, we can simply add a new group in table *group*, and add a user-group mapping. This is much easier than adding a new table in original ConfSys database structure and writing programs to define the

functions and the interface web pages. Apart from redesigning the database structure, we also redesigned the application architecture of ConfSys.

4.2. Application architecture redesign

During the process of debugging and improving ConfSys, some obstacles on programming were discovered. We found that large parts of these obstacles were the result of the application architecture of ConfSys.

4.2.1. Analysis of ConfSys application architecture

As shown in Figure 4-2-1, ConfSys adopted multi-layer architecture [32]. There are three layers in ConfSys: Data Storage Layer, Data Access Layer, and Presentation & Logic Layer.

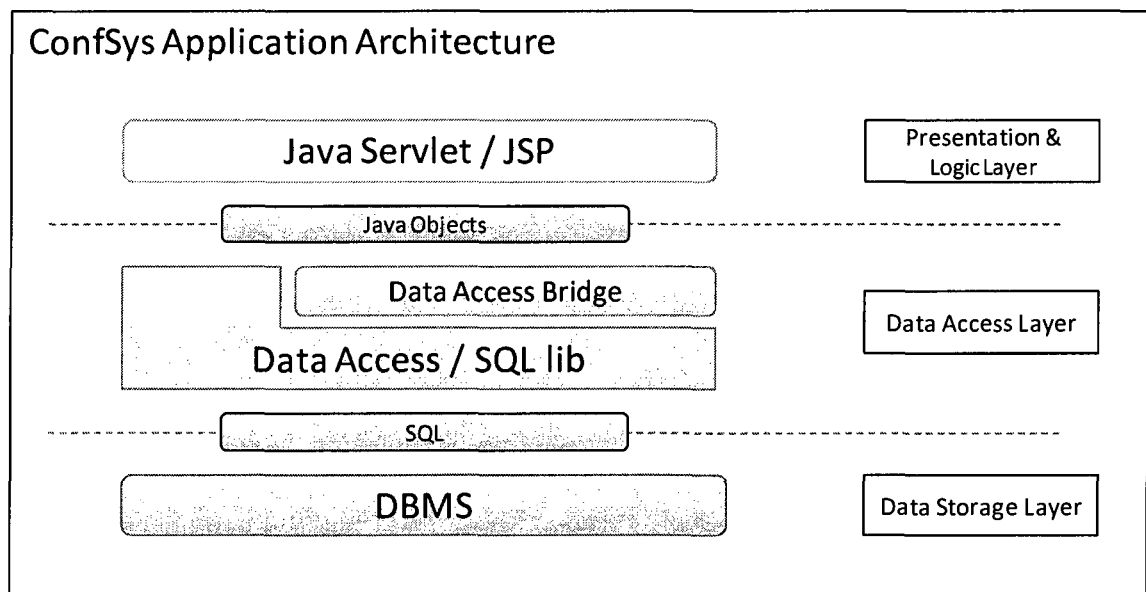


Figure 4.2-1 ConfSys Application Architecture

Data Storage Layer: This layer is responsible to provide data storage of ConfSys system. MySQL, a database management system (DBMS), is used in this layer. Through JDBC [33] api, data access layer is able to use SQL [34] language to manipulate the data in data storage layer.

Data Access Layer: This layer acts as a translator between the upper layer and data storage layer. ConfSys developers implemented a large data access / SQL library in this layer to prevent direct operations on DBMS. It accepts requests from presentation and logic layer then use SQL language to perform data queries or manipulations on DBMS, and return the results to presentation and logic layer. During the development of ConfSys, the SQL lib was found huge and complicated, so a Data Access Bridge was developed to reduce the complexity. The data access bridge is based on SQL lib, but generalized similar operations in SQL lib and provide a smaller function interface. This makes the operations on data easier and more standardized. The media between data access layer and presentation and logic layer are Java objects, such as Hashtable, Vector, etc.

Presentation & Logic Layer: The presentation and Logic layer is implemented with Java servlet and JSP technique. This layer presents user interface for operation, accepts user input, then manipulates data according to the business logic through data access layer, and return the operation result to the user. In this layer, both Java servlets and JSP pages can accept user input and generate result pages.

Some major disadvantages were found in this application architecture and the implementation of this architecture. They are list below:

Lack of unified logic definition: This is because the presentation and business logic are merged into one layer. Single business logic could be, and normally is, used in several places of presentation, thus need to be defined multiple times in these places. If the logic needs to be modified, it must be modified in all these places. This greatly increases the workload of programming, and is error prone.

Strong coupling [35]: This exists from the presentation and logic layer to data storage layer. It's because the transfer media between the logic layer and data access layer are basic Java objects, such as Hashtable, Vectors, etc. By using the vector objects to transfer data, the presentation layer needs to know the definition of the data and the sequence or places of those data. If the database structure changes, such as adding a column to a table, the sequence and place of the data will be modified accordingly. In such situation, the presentation layer is also required to be modified to extract proper data in the vectors from data access layer. Since one data could be used in many places in presentation and logic layer, such simple modification in data storage layer would introduce large amount of modifications workload in presentation layer. And such modifications are error prone.

Data access layer is difficult to use: In the implementation of the ConfSys, there are 2 major classes in data access layer. The class *conferenceSql* contains all necessary SQL

calls to DBMS, and the bridge class *dataConnect* contains bridge functions that based on *conferenceSql*. All data access requests are through these two classes, and each of them has hundreds of functions to support ConfSys. To develop a new function at presentation layer, developer may need to go through all those functions to find a proper one to use.

These disadvantages made ConfSys difficult to maintain and expand, so we redesigned and implemented the new application architecture for ConfSys2.

4.2.2. ConfSys2's redesigned Application Architecture

The new application architecture is illustrated in Figure 4-2-2. Several factors were taken into consideration when designing the new application architecture: unified business logic, reduced coupling, easy to use programming interface.

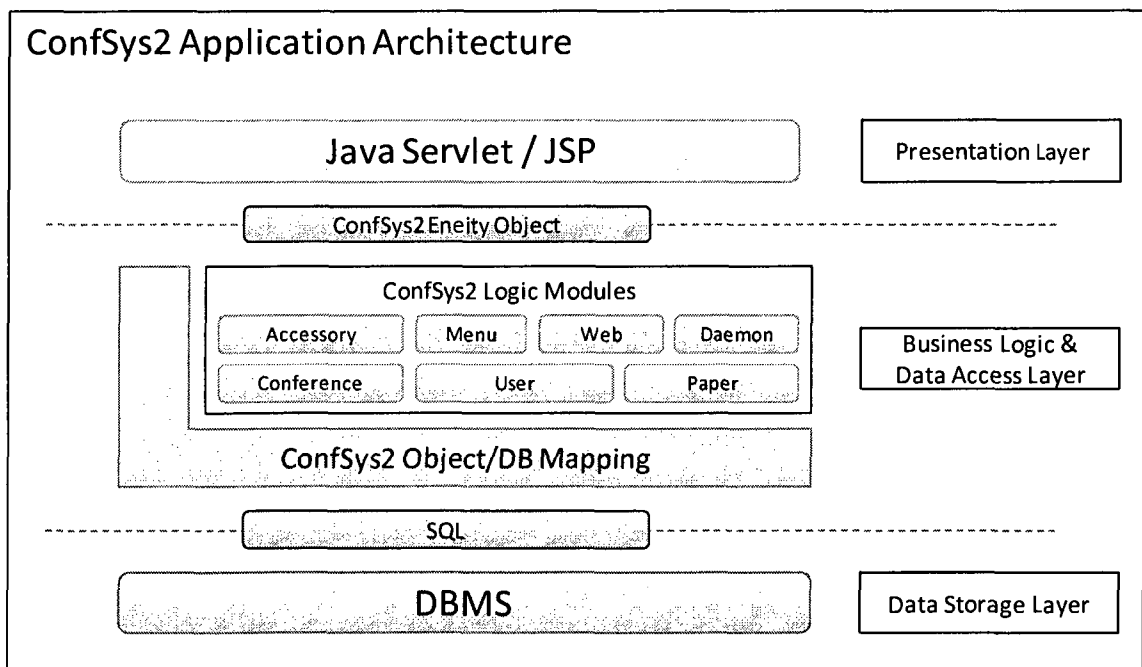


Figure 4.2-2 ConfSys2 Application Architecture

Unified Business Logic: this means that the business logic is defined only once in the entire system. Any function that requires dealing with certain logic should access the logic from only this location. To achieve this object, the presentation and logic were separated into two layers. The business logics are defined only in Business Logic Layer and presentation layer only deal with user request and provide the operation result. When the presentation layer needs to perform business logic related operations, it must request the business logic layer to perform the operations, so the business logic is kept unified.

Reduced coupling: this is achieved by using ConfSys2 entity objects as the media to transfer between layers. ConfSys2 entity object encrypts the entity's private data structure inside the object, and provides the necessary functions to access those data in a controlled manner. By doing this, the upper layer applications do not need to have the knowledge of how the data of the entity is constructed, they only need to request the data through functions provided by the entity object. By doing this, the coupling between presentation modules and data access modules and business logic modules are reduced.

Easy to use programming interface: this uses the concepts of easy to find, easy to learn, with sufficient functionality. To implement this, we analyzed the concepts (described in section 4.1.2) that was generalized at database design stage and implemented them as modules. Each module is responsible for the functionalities belonging to a concept, so it's easy to find by programmers. Furthermore, we designed

interfaces to standardize the operations on those modules, and provided adequate documents for these interfaces and functions to facilitate debugging and maintenance process. We designed modules to cover all the concepts we generalized, and implemented generalized requested operations to provide the functionality.

The new designed application architecture consists of three layers: data storage layer, business logic and data access layer, and presentation layer.

Data Storage Layer: this layer is similar to the old ConfSys data storage layer, except the database structure is totally redesigned. It provides the data storage function to support the system.

Business Logic and Data Access Layer: in this layer, we firstly implemented an Object – Relational Mapping (ORM) [36] to facilitate the access to database, so the higher level program does not need to use SQL directly to operate on the DBMS. Then, based on the concepts we generalized in section 4.1.2, we create business logic modules. These modules include: Conference, User, Paper, Accessory, Menu, Web, and Daemon. The Conference modules are responsible for the data structure and operations on conferences. The User modules are responsible for user, group, and the user-group mapping. The Paper modules handle paper related information and operations on papers. The Accessory modules provide information to support certain ConfSys2 functions. The menu modules deal with system functionalities and the user-group-function mapping to

support the flexible menu system and security management. Web modules provide facilities to make the web page generation easier. Finally, a daemon module defines the smart daemons that run in the background to automatically execute designated jobs at predetermined events and time.

Presentation Layer: This layer provides user interface to operate on ConfSys2. In this layer, the model 2 design model [37] is used to standardize the programming. In this model, Java servlet is used to accept the user request and perform operations on business logic layer. After the operations, it chooses the JSP page to pass the request, and let the JSP show the result to the user. This pattern separates the business operation and the respond page generation, thus making the presentation layer easy to maintain.

4.2.3. Comparison between ConfSys and ConfSys2's Application Architecture

Compared to ConfSys application architecture, the ConfSys2's architecture has several advantages. First, it enables the implementation of a unified business logic model, which reduces the difficulty of maintenance and the workload for modifications. Second, it weakens the coupling between different modules by using the ConfSys2 entity objects as the communication media. This reduces the chances of making errors when the data structure is changed. Third, the generalized concepts based modular design makes the programming interface easier to use.

Based on the redesigned database structure and better application architecture, we

redesigned the user interface which provides flexible menu system and security management.

4.3. User interface redesign

ConfSys uses multiple user login and fixed menu system to ensure the system security (as discussed in section 3.3). This is inconvenient for users who have multiple roles in the system. We redesigned the user interface to provide a standard tunnel to login and to be uniform for different user roles, thus making it more user friendly.

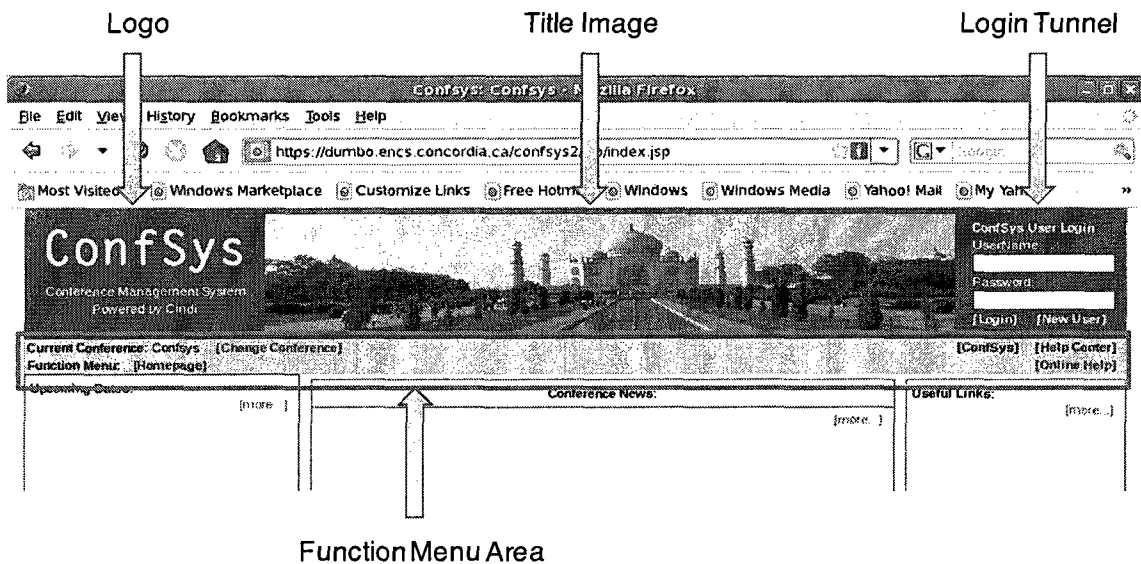


Figure 4.3-1 ConfSys2 User Interface for User Login

As shown in Figure 4-3-1, the ConfSys2 user interface's top part is the title area, which contains the ConfSys logo at its left, title image in the middle, and a user login panel at the right side. Below the title area is the function menu area, which has several parts. On the top left side of the menu area, there is an indication of which conference the current

user is operating, a user is able to change current conference by clicking the “Change Conference” button. Below the conference information, there is a function menu, where user can select functions to operate after login. On the right side of the function menu area, there are buttons linking to help system. The “help center” link to help page which provide frequently requested helps, such as retrieving forgotten password through email. The “Online help” button provide the help information about the current working interface. The title area and function menu area are always available to user no matter what operation the user is working on. Below these two areas is the working area, where the user interface of all function is shown.

Before a user logs in to the system, the login panel on the top right of the interface provides the username and password for the user to login. And an unregistered user can use the “New User” button to sign-up as a new user to ConfSys2 system.

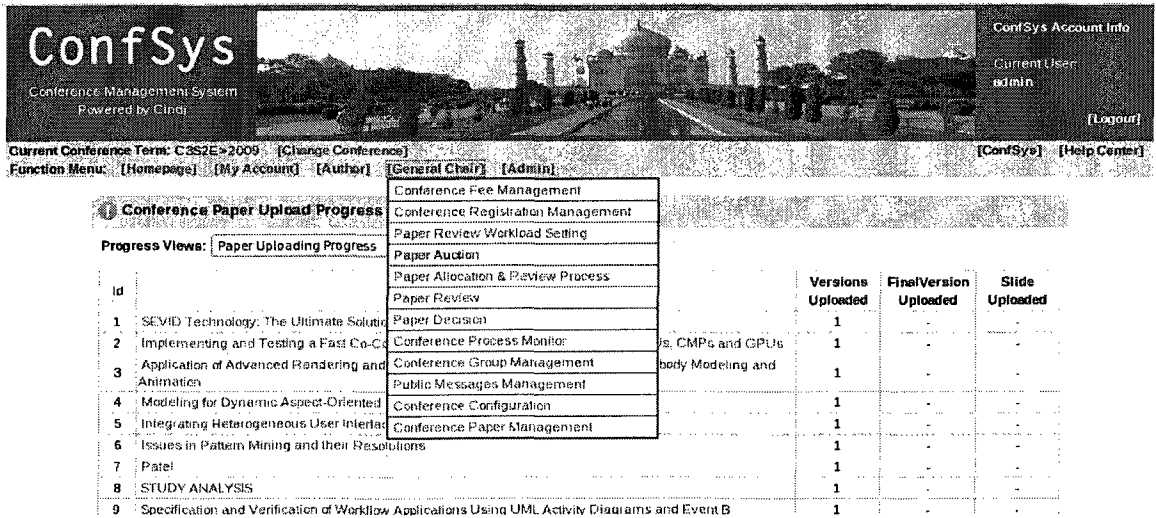


Figure 4.3-2 ConfSys2 User Interface After Logged In

After the user logs in to ConfSys2, the current user information and the menu system is shown to the user. Figure 4-3-2 illustrates the user interface after user login. The login panel on the top right of the interface will show the user account information, and provide a “Logout” button for the logged user to log out of the system. In function menu area, the flexible menu system is shown to the current user. Only the functions accessible to the current user will be shown. The menu items/functions are classified according to the role of the user. When user click the role name, such as “Author” or “General Chair”, a pull-down submenu will be displayed, where all functions accessible for that role are listed. By clicking the function link in this submenu, the user can access that function. All the operation panels of the functions use the same framework, where function working area is below the title area and menu area. A subtitle which indicates the current running function will be shown in the working area, and user interface for the function is shown under the subtitle.

Compared to the ConfSys, ConfSys2’s user interface has several advantages. First of all, with single login, user can access all functions he can access, not only for multiple roles, but also in multiple conferences. Second, once logged in, user can access the menu system throughout the process, no matter the function being used. This is convenient for user to easily find and access desired function during the middle of a task. Third, because the menu system is similar to the menu system used by most operation systems, it is easy

for users to learn how to use it. Besides, there are some other techniques, such as jQuery [38] and Ajax [39], have been used to increase the usability of the interface, which will be described in chapter 5.

5. Enhancements and new features in ConfSys2

With the new designed database structure, application architecture, and the framework of new user interface design, we have implemented ConfSys2, a web based multi-conference management system. In ConfSys2, we not only retain the functionalities of ConfSys to manage the entire process of a conference, but also add new functions such as multi conference or journal management, unified user management, flexible menu system and security management. This chapter discusses these enhancements and features of ConfSys2.

5.1. System initializing program

In ConfSys2, we have introduced a dynamic web based system setup and initialization program, which helps the system administrator in the initial installation and initialization the ConfSys2 system. The steps are described bellow:

The application server Tomcat and DBMS system MySQL should have been properly installed before installing ConfSys2. Good introduction to this is in the Apache-Tomcat[40] and MySQL [41] website and not covered here.

After copying the ConfSys2 program files to Tomcat's application directory, administrator should be able to access the initialization start page through Tomcat. The user interface for the initialization start page is illustrated in Figure 5-1-1. Here

administrator can input parameters for the main database to be used by ConfSys2. If the database has been created by DBMS administrator, and have been granted access privileges to the ConfSys2 database user, only the parameters in “ConfSys database connection setting” are required to be provided to execute a series of SQL script to create tables in main database. If the database is not pre-created, ConfSys2 administrator also has the option to fill in the Mysql Administrator’s account information to allow ConfSys2 to run the SQL script as the MySQL administrator to create the database and grant access to ConfSys2. Once the necessary database information is filled, administrator can click the “Next>>” to submit the information and continue to the next step of initialization.

ConfSys Initialization Step1: provide operating account

(Required) Confsys Database Connection Setting:

Please make sure the database and username meet the Mysql syntax. It's suggested to use character and number only.

Database Server (host:port):

Confsys Main Database:

DB Connection User Name:

DB Connection Password:

(Optional) Mysql Admin Account: (The account has sufficient privilege to create main database for confsys2)

Mysql admin username:

Mysql admin password:

* If you don't provide this information, you must first ask Mysql admin user to create Confsys main database (as your input above) and grant all privilege of this database to the Confsys DB Connection User (as your input above) before this step, so the system will be able to create necessary tables.

* If you provide this admin account, the Confsys will automatically create the necessary database and tables using admin account (for this time only).

Figure 5.1-1 System Initializing Program Start

In step2, the program will use the database information provided by administrator, and start the initialization process. As shown in Figure 5-1-2, the program will create the required database and tables to be used as the main database of ConfSys2, and perform initializing work, such as adding the “everyone” group, importing accessory data, etc. The program will collect the status of these operations, and show them to the administrator for analysis. If all the operations are satisfactory, the administrator can click “Next>>” to allow the program to create the system admin account in step3.

```
Confsys Initialization Step2: create system tables

Start initializing main database

CREATE DATABASE `confsys_main` succeeded.
Grant privilege to user `testsysdba`@`localhost` succeeded.
CREATE TABLE `confsys_main`.`user` succeeded.
CREATE TABLE `confsys_main`.`subject_hierarchy` succeeded.
CREATE TABLE `confsys_main`.`interest` succeeded.
CREATE TABLE `confsys_main`.`log` succeeded.
CREATE TABLE `confsys_main`.`user_relation` succeeded.
CREATE TABLE `confsys_main`.`conference` succeeded.
CREATE TABLE `confsys_main`.`group` succeeded.
CREATE TABLE `confsys_main`.`charge_item` succeeded.
CREATE TABLE `confsys_main`.`charge_group` succeeded.
CREATE TABLE `confsys_main`.`charge_history` succeeded.
CREATE TABLE `confsys_main`.`user_group` succeeded.
CREATE TABLE `confsys_main`.`content` succeeded.
CREATE TABLE `confsys_main`.`menu` succeeded.
CREATE TABLE `confsys_main`.`country` succeeded.
CREATE TABLE `confsys_main`.`organization` succeeded.
CREATE TABLE `confsys_main`.`sys_template` succeeded.
CREATE TABLE `confsys_main`.`mail` succeeded.
CREATE TABLE `confsys_main`.`public_message` succeeded.
Creating `everyone` group succeeded.

Initializing main database succeeded.

Debug Stops after step 13
Creating system tables succeeded.
Country, organization, subject_hierarchy data have been filled into the database.
182 countries were added, 0 inserting failed.
12016 organizations, 0 inserting failed.
505 subject hierarchies were added, 0 inserting failed.

Click "Next >>" to go to step3 to create Confsys admin account and fill data

Next >>
```

Figure 5.1-2 System Initializing Program Step Result

After the ConfSys2 admin is created in step 3, administrator can view the default system parameters and modify them if necessary. As shown in Figure 5-1-3, some system parameters are preset, and some environment sensitive parameters have been automatically detected by the program. Administrator can modify the parameters as desired, and then submit them to the initialization program. After this step, ConfSys2 is initialized and ready to use.

Confsys Initialization Step4: setup Confsys configuration

Initial Confsys System Parameters

CONFSYS_NAME	Confsys The name of the confsys system
CONFSYS_URL	http://localhost:8080/Confsys2 Confsys website entry
CONFSYS_FILE_DIRECTORY	/disk2/storage/eclipse_workspace/metadata/plugins/org.eclipse.wst.server.core/tmp0/wtpweb Directory to store files, such as papers, slides
CONFSYS_DEBUG	Off Debug setting, on off
CONFSYS_MAIL_SERVER	localhost The mail server to send out email
CONFSYS_MAIL_ADDRESS	bcdesai@confsys.concordia.ca Confsys system/admin email address
CONFSYS_MAIL_ADDON_SUBJECT	Confsys mail: The string that will be automatically added to the title of every user mail send out
CONFSYS_MAIL_ADDON_HEAD	This mail is send by Confsys system. Please don't reply. The string that will be automatically added to the front of the content of every user mail send out

Figure 5.1-3 System Initializing Program System Parameter Setting

Unlike the ConfSys's initializing process, which require the administrator to manually create database and tables and modify parameters, ConfSys2 use the web-based initialization program to standardize the initializing process. This eliminates the possibilities of making errors during the manual database operations. Besides, auto detection of environment sensitive parameters reduces the initializing workload.

5.2. Uniform user interface and help system and their implementation

ConfSys2 adopted a uniform user interface framework, as described in section 4.3.

This framework, as illustrated in Figure 5-2-1, allows authenticated users to access the flexible menu system throughout the working process, no matter which function the user is using. This is implemented through the use of JSP inclusion [42].

The screenshot shows the ConfSys user interface. The header includes the ConfSys logo, navigation links for 'Current Conference Term: C3S2E-2009', and user account information. Below the header is a table titled 'Group List' with columns for Group Name, Description, Members, Group Type, and Operation.

Group Name	Description	Members	Group Type	Operation
1 General Chair	General Chair	2	Predefined	
2 Program Chair	Program Chair	2	Predefined	
3 Treasure Chair	Treasure Chair	0	Predefined	
4 Program Committee	Program Committee	56	Predefined	
5 Reviewer	Reviewer	0	Predefined	
6 Author	Author	101	Predefined	
7 Author With Accepted Paper	Author With Accepted Paper	58	Predefined	
8 Author With Rejected Paper	Author With Rejected Paper	34	Predefined	
New	Group Name: <input type="text"/> Show Name: <input type="text"/>		Created	Add

Figure 5.2-1 Uniform User Interface

The title area and menu area are implemented in a single JSP page, *header.jsp*, which is responsible only for the title, login tunnel, and menu generation. And the bottom area is implemented in another JSP page, *bottom.jsp*, which act as the end bracket of the framework. While we use model 2 [34] in presentation layer, all user interfaces are generated by JSP pages. In each functional JSP page, we include the *header.jsp* at the

start of the page, and include the *bottom.jsp* at the end of the page, then the functional JSP page is embedded into the framework.

This implementation has several advantages. First of all, it is simple. With 2 lines of “<jsp:include” code, the functional JSP page is able to adopt the framework. Second, functional JSP page need only focus on its own functional user interface design, without considering the html heading and styles because those tasks will be done by the header and bottom page. Third, the appearance of the system is easy to be unified. When the style of the appearance is required to be changed, we only need to modify the header page to modify the style, and all functional pages that include the header page will be modified accordingly.



Figure 5.2-2 ConfSys2 Help Center

Beside the uniform user interface framework, ConfSys2 also provide a uniform help system. The help system consists of two major parts: Help Center and Online Help Page.

Help Center: Help center contains some frequently requested help information and

solutions. For example, when a user forgets the username or password, s/he can go to help center by clicking the red “Help Center” at right side of the menu area. There s/he will be directed to the password retrieving function, and can retrieve the username and password through the registered internet email.

Online Help Page: online help page provide instant help information for the current working page. As shown in Figure 5-2-3, when a user has some questions about the page s/he’s working on, s/he can click the green “Online Help” at the right side of the menu area. The help information about the current page will be pop up to the user for reference.

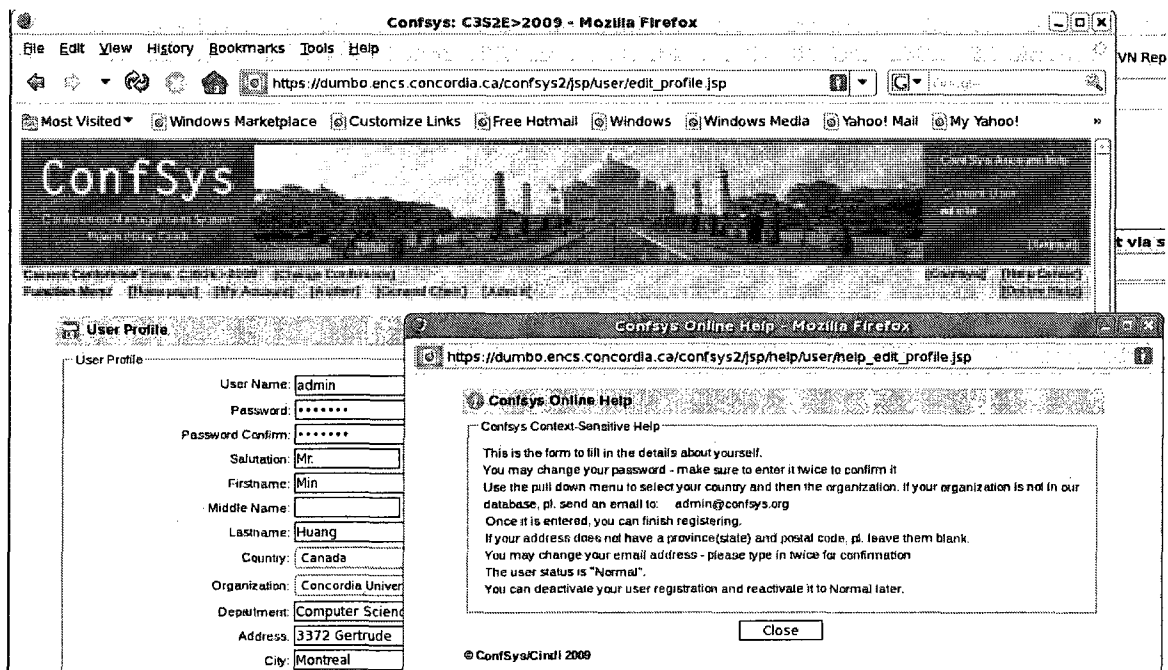


Figure 5.2-3 ConfSys2 Online Help System

The online help system is implemented through a mapping between JSP pages and help pages. Because all user interface are generated by JSP pages, and they use a uniform user

interface framework, the help mapping module in the header file can detect which function/JSP file is currently embedded into the framework. In this implementation, we use a convention that the help page name is generated by adding the “help_” before the working JSP file name. By employing this convention, the help mapping module can easily find the link to the current function/JSP file. The implementation of online help system separate the help information from the presentation JSP pages, thus eliminating the possibility of introducing errors when modifying help information in functional JSP pages.

5.3. jQuery powered Ajax enabled friendly user interface

To increase the friendliness of ConfSys2’s user interface, we adopted jQuery [35] library and Ajax [36] technique.

The jQuery is a widely used JavaScript [43] library with many features to simplify the browser side JavaScript programming. First of all, it is light-weight. The basic jQuery library is only 19K bytes in size, which adds little weight on web application transmission. Second, it is functional supporting document traversing, event handling, animating, and Ajax interactions to allow fast web development. Third, it supports multiple frequently used web browsers, such as Internet Explorer, Firefox, Safari, Opera, etc. Last but not least, there are many JavaScript modules that have been developed based on jQuery to help rapid web development.

With jQuery and its supplement modules, we are able to develop friendly user interface with ease. The figure 5-3-1 illustrates a user interface in ConfSys2 that uses jQuery. In this public message publishing function, we used jQuery's calendar module to standardize the user input on date information. When user moves input cursor to the date field, a calendar will pop up for user to choose the date by clicking on desired date. In this situation, user also has the option to manually input the date, and the calendar will perform the check on input to make sure that user inputs a correct date. This not only makes the user interface friendly, but also reduce the programming workload on checking date format.

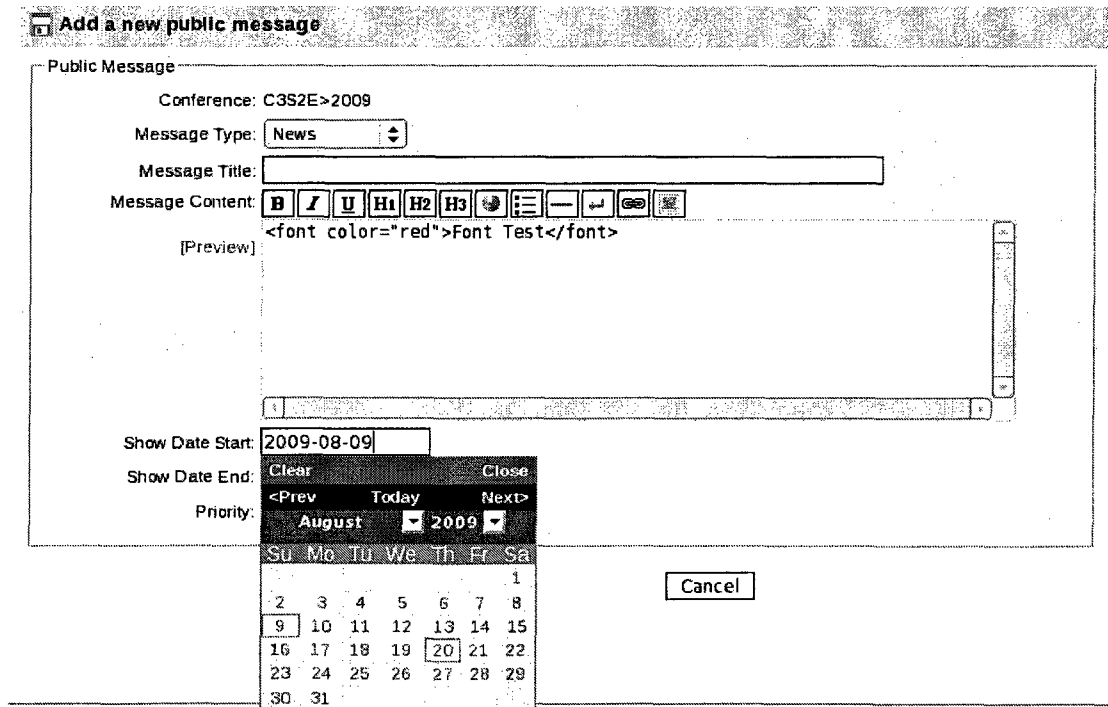
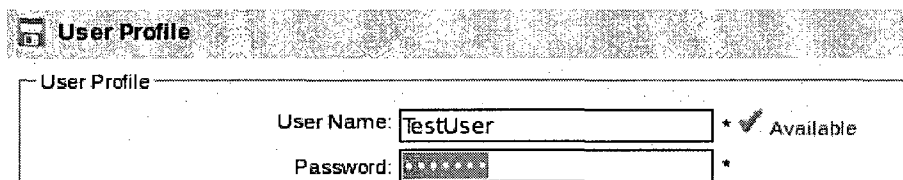


Figure 5.3-1 jQuery Powered Intuitive User Interface

Ajax is another technique used in ConfSys2 to make friendly user interface. Ajax is the

technique that uses JavaScript to communicate with server in the background without interfering with the display and behavior of current working page. This communication could transfer only partial user input, thus reducing the amount of data transmission. Since JavaScript runs at the background, the current working page is not required to be re-generated, thus reducing the workload for both the server and the client side.

Figure 5-3-2 illustrates the Ajax based user profile editor. When a user modifies the username, it is required to check if the username is in conflict with already registered usernames. In this function, after a user input the new username, the JavaScript program sends the new username to the server to check the availability. If the new username is unique, the program will show an “Available” mark beside the username. If the new username has been used by another user, an “Unavailable” mark would be shown to alert the user. In the whole process, the page is not reloaded, thus it would not interfere with the user’s current operation.



The image shows a web form titled "User Profile". It contains two input fields: "User Name" and "Password". The "User Name" field contains the text "TestUser" and is followed by a checkmark icon and the word "Available". The "Password" field contains several asterisks and is followed by an asterisk icon. The form is enclosed in a border with a title bar that says "User Profile".

Figure 5.3-2 Ajax Enabled Friendly User Interface -1

The Figure 5-3-3 illustrates another function which uses Ajax technique. In the manual paper allocation function, a chair will need to check the reviewers’ topics of interest and the papers he is current reviewing. He can do this by selecting a reviewer from the

reviewer list. When a reviewer is selected, the JavaScript will transfer the selected reviewer id to the server to query for the reviewer's information, and show the server response below the reviewer selection interface. The whole process does not need the chair's involvement, and the working page is not re-generated, thus making the operation more fluent.

Allocated Reviewers				
Id	Reviewer Info	Download Date	Mark	Operation
1	Ziad Kobi (University of Windsor, Canada)	-	5.0	Delete
2	Michael McAllister (Dalhousie University, Canada)	-	5.0	Delete
3	Eric Pardeda (La Trobe University, Australia)	-	6.0	Delete
4	Oliver Gerbe ()	-	-	Delete
New	6. (Bid High) Annie Ying (IBM, United States) Allocated Papers: 5			Allocate
	Reviewing Papers: 1. An Automated Procedure for Identifying Poorly Documented Object Oriented Software Components () 2. Integrating Usability Engineering and Software Engineering in Mixed Reality System Development () 3. Towards Bridging the Semantic Gap in Software Industry () 4. Decision Support System Environment for Software Architecture Style Selection (DESAS v1.0) () 5. Markovian Workload Modeling for Enterprise Application Servers ()			
	Reviewer's Interests: 1. C3S2E::Software engineering			

Figure 5.3-3 Ajax Enabled Friendly User Interface -2

By adopting jQuery and Ajax technique, ConfSys2 gained many advantages over the original ConfSys. First of all, the user interface is more intuitive and easier to use. Second, the communication between the browser and server are reduced, thus reducing the workload on both server and client side. Third, with the use of JavaScript library and pre-created modules, the workload of programming on user interface has been greatly reduced.

5.4. Conference/Journal series management

In ConfSys2, we introduce the concept of conference/journal series. This is based on the fact that most conferences will hold meetings periodically and journals work in similar fashion with regular issues. The Figure 5-4-1 illustrates the conference/journal organization structure in ConfSys2. A conference/journal series is a series of cycles that

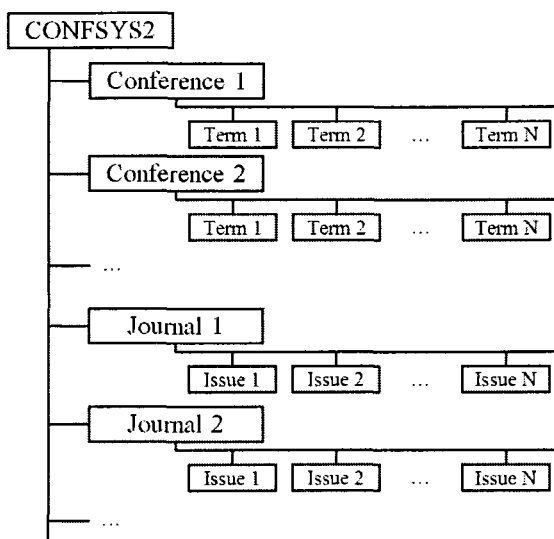


Figure 5.4-1 ConfSys2's Conference/Journal Series Management

run under a same conference/journal title, such as C3S2E, C3S2EJ and IDEAS. In ConfSys2, those cycles are called terms of a conference or issues of a journal or simply events. Each event can have its own set of settings, such as title, dates, program committee members, etc. And each event can have its own set of groups/roles to manage the event or participate in the event. This makes the event management flexible. ConfSys2 capitalizes from the observations that such events generally use similar settings or patterns over time. Editors seldom changes between issues; program committee has

on-going membership and the schedule is consistent. To reduce manual work load in setting up new event, Confsys2 allow conference organizers to propagate settings and scheduling patterns between two related events and allow finer automated or manual tuning.

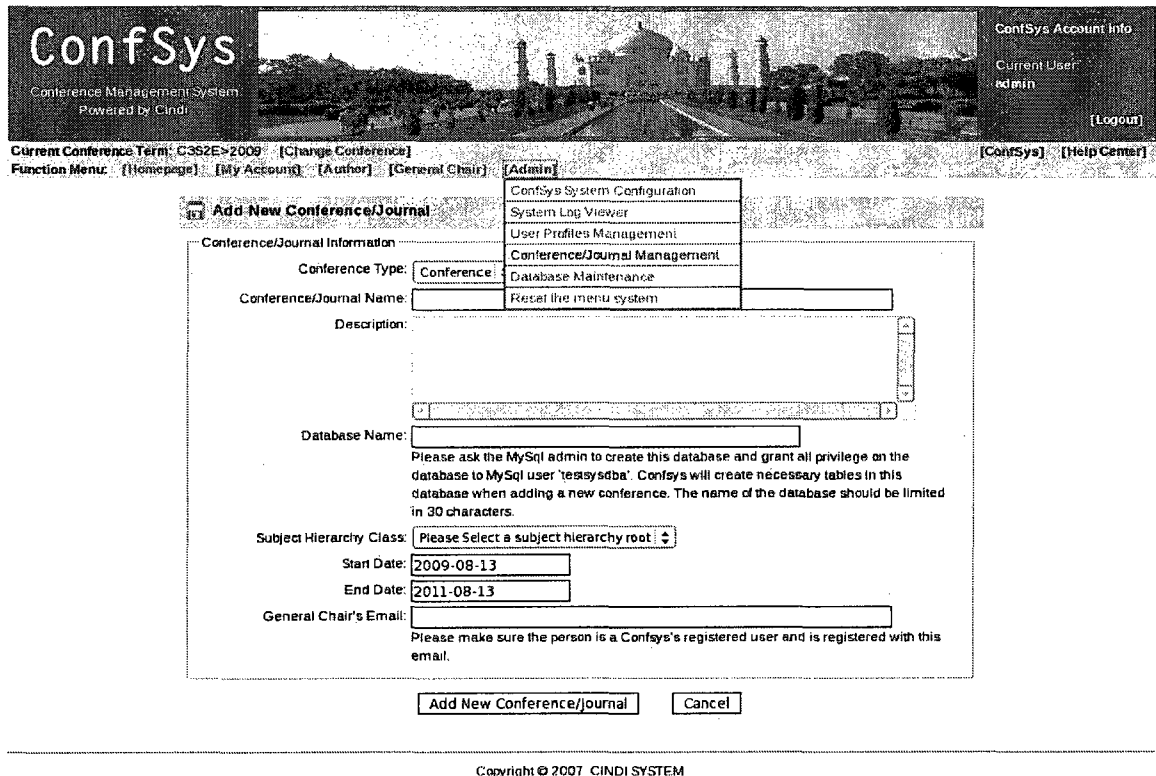


Figure 5.4-2 Conference/Journal Series Management – Add New Conference

As illustrated in Figure 5-4-2, ConfSys2 administrator can add a new conference through the Conference/Journal Management function. In the user interface, administrator can set the basic conference information, such as type (whether it is a conference or a journal), conference name, description, the database used to store conference data, date, and the conference chair's email. After adding the conference, the

general chair can login and manage the conference, such as add a new term, tune and/or set up term parameters, etc.

Add a new term	
Add a new term for conference: IDEAS	
Term Title	IDEAS 10
Term Description	IDEAS 10
Subject Hierarchy Class	IDEAS
Start Date	2009-08-13
End Date	2010-08-13
Scheduling Template	IDEAS09
Copy Groups From	IDEAS09
<input type="button" value="Add this new term"/> <input type="button" value="Cancel"/>	

Figure 5.4-3 Conference/Journal Series Management – Add New Term

A general chair can manage a conference series through the “Conference Configuration” function. As illustrated in Figure 5-4-3, general chair can add a new term for a conference or a new issue for a journal. ConfSys2 will create corresponding pre-defined groups for conference term or journal issue. In a conference term, pre-defined groups include General Chair, Program Chair, Program Committee, Treasure Chair, Reviewer, and Author. In a journal term, pre-defined groups include General Editor, Editor, Treasure Chair, Review Board, Reviewer, and Author. If terms or issues have existed in the conference series, general chair has the option to copy members of group from existing term, and can choose to use a scheduling template from a previous term.

Compared to ConfSys’s single conference management, ConfSys2 provide a great flexibility in event management. It not only allows the system administrator to manage multiple conferences and journals, but also enable the general chair of a conference series to manage multiple tracks at the same time and multiple ongoing terms or issues. Furthermore, the feature of copying groups and scheduling template from other terms reduces the workload of chairs on setting up multiple tracks and terms/issues.

5.5. User management and community mode

The screenshot displays the 'User List' management interface. At the top, there is a navigation menu with the following items: ConfSys System Configuration, System Log Viewer, User Profiles Management, Conference/Journal Management, Database Maintenance, and Reset the menu system. Below the menu, there is a search and pagination area with '15 users / page' and 'Page 1 of 58'. The main content is a table with the following data:

No.	User Name	Name		Organization	Status
1	admin	Min Huang	ikhwo2002@yahoo.com	Concordia University, Canada	normal
2	user10000	Adel Khelili	a.khelili@alhosnu.ae		registered
3	user10001	Abdelwahab Hamou-Lhadi	abdelw@ece.concordia.ca	Concordia University, Canada	registered
4	user10002	Hassan Abolhassani	abolhassani@sharif.edu	Sharif University of Technology, Iran	registered
5	user10003	kamani supriya	absobloodylutelycool@gmail.com		registered

Figure 5.5-1 ConfSys2 User Management

ConfSys2 manages all its users in a centralized manner. All people who participate in conferences in ConfSys2 must sign up and become a user of the system. As illustrated in Figure 5-5-1, the system administrator is able to manage all user information through “User Profile Management”. In this function, administrator is able to browse users by paged list or search for a user through username, email, country, and other basic user information. Administrator can also view and modify the user information and set up the

status of user to enable or disable a user account as necessary.

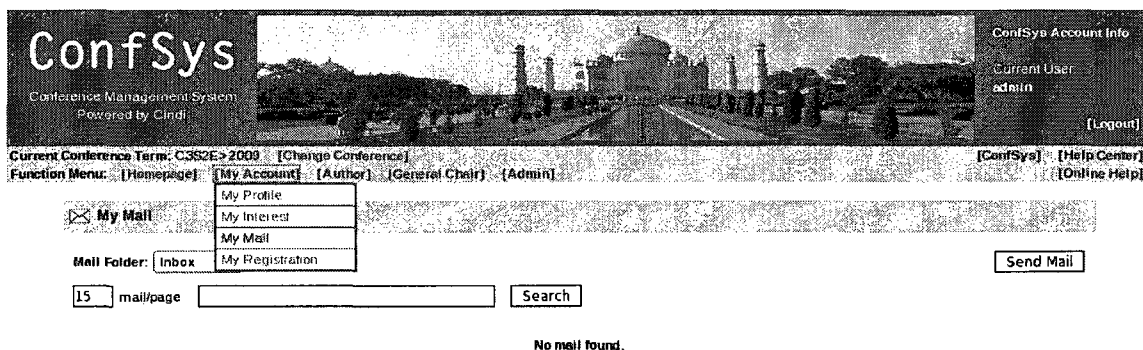


Figure 5.5-2 Community Mode

Unlike ConfSys and most other conference management systems, ConfSys2 does not enforce a role for a user. Every non-admin user has the same status when they sign up to the system. They only take on roles once they participate in an event. For example, after they submit a paper to a conference, they become an author in that conference. And they become program committee members of a conference only when they are nominated by a general chair of that conference. As illustrated in Figure 5-5-2, ConfSys2 provide basic functions to every new user, such as updating profile and password, modifying topics of interest, internal mail, etc. And ConfSys2 allow every user to roam between events, get the information about an event, and participate in the event if desired. This creates a community that consists of conferences, participants, and papers. This mode can enhance the communication between conferences of similar domain, and between people with similar topics of interest.

5.6. Security management and User-Group-Function Mapping

ConfSys2 implemented two level of protection. On transmission level, ConfSys2 use HTTPS to protect the information when transferring through Internet. On application level, ConfSys2 uses User-Group-Function mapping to ensure certain functions accessible only by associated users.



Figure 5.6-1 User – Group – Function Mapping

As illustrated in Figure 5-6-1, there are multiple many-to-many relations. A user can belong to multiple groups, and a group can contain multiple users. A function can be assigned to multiple groups, and a group can be assigned multiple functions. By analyzing the transitive relation, we can get a mapping of which functions can be performed by a certain user. ConfSys2 uses this mapping to control the user access to functions. Only functions could be accessed by a given user would be shown to the user, hence the user is able to execute these functions for the given event. Other functions that don't have such relation to the user are not accessible to this user.

ConfSys2 provides a set of facilities for a privileged user to manage this mapping. First of all, there are some predefined groups in conferences, such as Program Chair and

Reviewer, and general chairs are able to add new groups and manage non-predefined groups. The group management user interface is illustrated in Figure 5-6-2.

Group Name	Description	Members	Group Type	Operation
1 General Chair	Description: General Chair	2	Predefined	
2 Program Chair	Description: Program Chair	2	Predefined	
3 Treasure Chair	Description: Treasure Chair	0	Predefined	
4 Program Committee	Description: Program Committee	56	Predefined	
5 Reviewer	Description: Reviewer	0	Predefined	
6 Author	Description: Author	101	Predefined	
7 Author With Accepted Paper	Description: Author With Accepted Paper	58	Predefined	
8 Author With Rejected Paper	Description: Author With Rejected Paper	34	Predefined	
9 Chair Assistant	Description: Chair Assistant	1	Created	Delete

Figure 5.6-2 ConfSys2 Group Management

Second, general chair is able to manage the members in a group. As shown in Figure 5-6-3, group members can be added through their email because users can be uniquely

UserName	Name	Email	Operation
1 khtwo3000	Min Huang	khtwo3000@yahoo.com	Delete
New Member Email:			Add

Figure 5.6-3 Group User Management

identified by their email address. The user in the group can be deleted easily by clicking

the delete button included in the user in group management interface.

	Function	Menu	Start Date	End Date	Privilege	Operation
1	Conference Fee Management	0Chair Assistant:0Conference Fee Management	2009-08-08	2009-08-13	conf_edit_10	Delete

Function: Paper Review Workload Setting

Privilege: conf_edit_10

SubMenu: 0Chair Assistant:1Paper Revi

Start Date: 2009-08-07

End Date: 2009-08-13

Format: [MainSequence 0-9A-Z][MainMenuName]:[BarSequence 0-9A-Z][MenuBarName]

Buttons: Send Group Mail, Group Member Management

Figure 5.6-4 Group Function Management

ConfSys2 also provide a “Group Function Management” interface to manage functions that are accessible to a group. As illustrated in Figure 5-6-4, this function allows privileged users to add existing functions to the current group and setup the privilege string. The privilege string is in the format of scope_operation_level. The scope field can be “sys” or “conf”, which defines if the group can use this function in system wide or current conference wide. The operation field can be “view” or “edit”, which defines if the group have the privilege of read or write. The level field can be from “11” to “13” which defines the group’s access level. A function can analyze the privilege string and provide the group only limited operations defined by the string. The privilege setting has been simplified that only supported privileges will be listed and a privilege description will be provided. The user can also set up the menu and submenu for the group users. The menu generation module will analyze the submenu string and automatically generate the menu for the user at run time. The submenu string is in the format of [MainMenu

Sequence][Mainmenu Name]::[Submenu Sequence] [Submenu Name]. For example, a menu string “1Assist Chair::2Registration Management” associate to a function will make the group user be able to access the function through the first main menu named “Assist Chair” and 2nd submenu bar named “Registration Management”. Furthermore, the available data can also be set up to limit the access to a certain period of time.

The User – Group – Function mapping introduced greater flexibility in ConfSys2. First of all, a flexible menu can be generated through analyzing this mapping. Second, the application security is enforced by allowing accessibility to only those functions accessible to those users who have the associated privileges. Third, with the management user interface of user, group, and function, general chair can easily share part of his work with associate chairs by adding them to a special group and assign functions to that group.

5.7. Uniform parameter/template management

ConfSys2 standardizes the conference parameter and template management. They are now managed through a single function “Conference Configuration”. The user interface for this is illustrated in Figure 5-7-1. In this interface, privileged users can select the parameter/template type and access to these parameters.

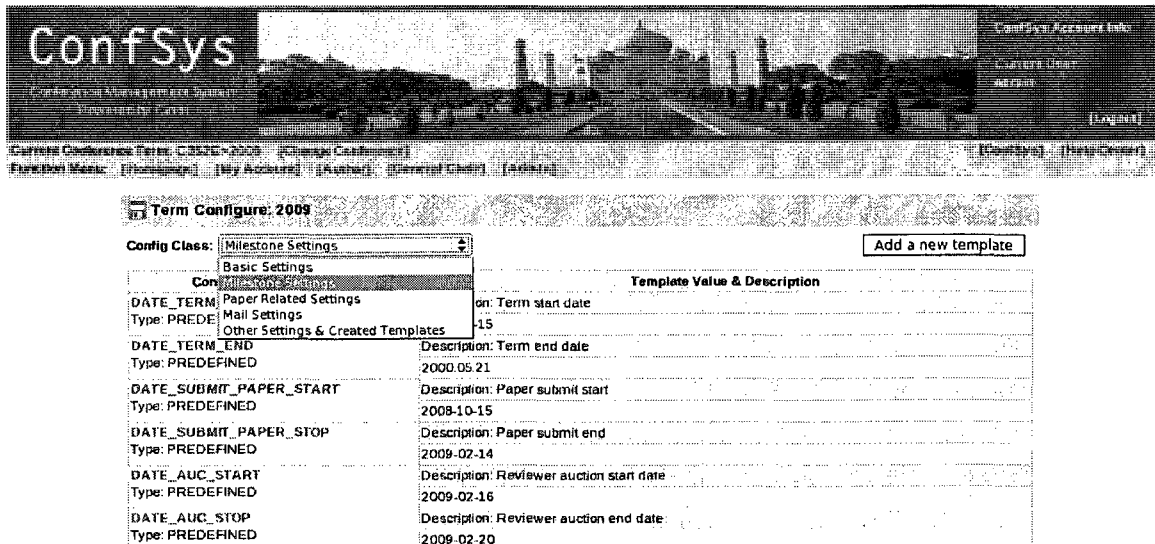


Figure 5.7-1 Uniform Parameter Management

Compared to ConfSys, ConfSys2 provides more powerful and flexible features for parameter/template management. First of all, privileged users are able to add their own parameters or templates, and manage them under the same user interface. For example, general chair can create new mail templates for special use. When the template is needed, s/he can just include the template name in the mail, and the mail will be automatically translated to the template content. Second, all parameters and templates, include those user created parameters, are accessible throughout the conference functions which is applicable. For example, in public message publishing, the general chair can publish a public message which contains parameters such as the due dates or paper related parameters. Third, all parameters and templates can include other parameters and templates. Recursive inclusion is also supported up to three levels depth because of the performance consideration. Last but not least, in milestone parameter management, the

group dates shifting function is implemented so that a certain milestone and subsequent milestones can be shifted in a single operation. These features make the parameter and template management more flexible and easier to use.

5.8. Flexible review system

It is natural that different conferences might want to evaluate papers using different criteria. To meet such needs, ConfSys2 implemented a flexible review system, which uses XML [44] to store the review template and review result.

The review template is one of the conference parameter which is named “REVIEW_TEMPLATE”. It contains a XML document to define the review fields and related settings, which include Field-Name, Field-Description, Field-Value, Field-To-Author, Field-Type, and Field-Option.

Field-Name defines the name of the field.

Field-Description gives the description of the field, and it will be shown to a reviewer when s/he moves cursor to this field.

Field-Value defines the default value of this field.

Field-To-Author defines if this review field is accessible to the author of the paper.

Field-Type defines the input method of this field. It can be a single line text box, a large text area, or a selection.

Field-Option can occur multiple times to define the options for selecting when the

filed type is selection.

The review template can be modified by a privileged user through the parameter and template management. ConfSys2 provides a sample review template for each conference; the conference organizers can reference the sample and modify the template to suit their needs. The Figure 5-8-1 illustrates the interface to modify this template.

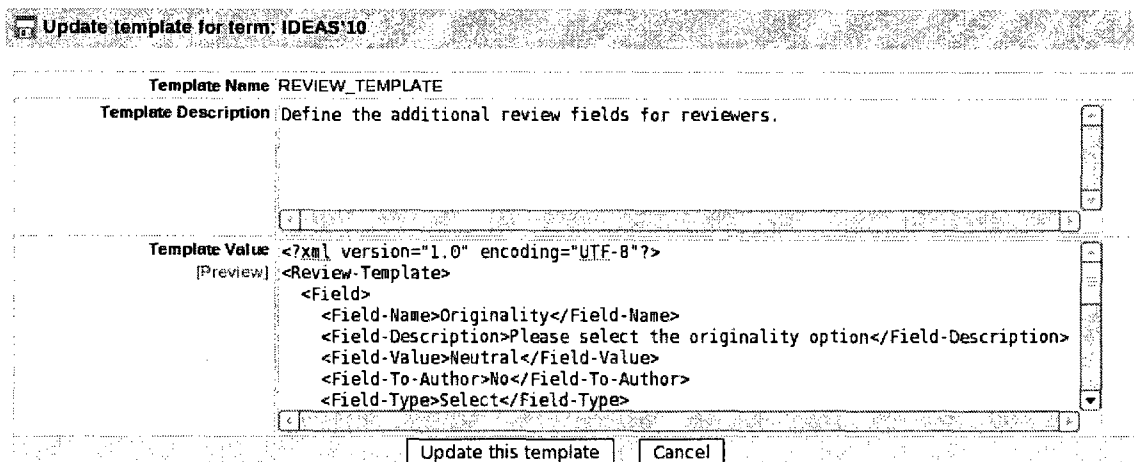


Figure 5.8-1 User Interface for Defining a Flexible Review Format

When the review template is set up and the conference enters the review phase, the template will be translated by the flexible review module, and the defined review fields, along with paper score and confidence, will be shown to reviewers for completion. As illustrated in Figure 5-8-2, the review fields Originality, Strong Point, and Other Comment are user defined review fields. After the reviewer completes these fields, the review system will store the completed values in a XML document and associate it with the reviewer. This review result XML document and review template XML document

will be combined when the chairs view the review results.

The image shows a web form titled "My Review Result". It contains several input fields and a text area:

- Paper Score:** A dropdown menu with the value "6.0" selected. Below it is a text box containing the text: "A weak vote for acceptance. A reasonable contribution to an interesting problem/application - or maybe the contribution is good but the authors don't seem to understand what it is and/or express it well - or maybe it's a good paper, but the subject area is marginal for the conference."
- My Confidence:** A dropdown menu with the value "3" selected. Below it is the text: "Consider me an 'expert' on this paper I understand it in detail. Interpretation of score ratings."
- Originality:** A dropdown menu with the value "Neutral" selected.
- Strong Point:** An empty text input field.
- Other Comment:** A text area with the placeholder text "Please input other comments about this paper".

At the bottom of the form is a "Submit" button.

Figure 5.8-2 Flexible Review Presentation

Compared to ConfSys, the implementation of flexible review system gives the conference chairs more flexibility to create, store and retrieve their preferred review template and the corresponding results. Furthermore, this feature eliminates the requirement of re-programming when more review fields are required or fields attributes need to be changed.

5.9. Smart daemons

Most conferences or journals have many repetitive tasks, such as sending reminder mails, setting up next journal issue, etc. Confsys2 has implemented smart daemons to automatically schedule and carry out these tasks. Figure 5-9-1 illustrates some of the Confsys2's functions during various phases of a conference term and the ones that can be done automatically by Confsys2 daemons.

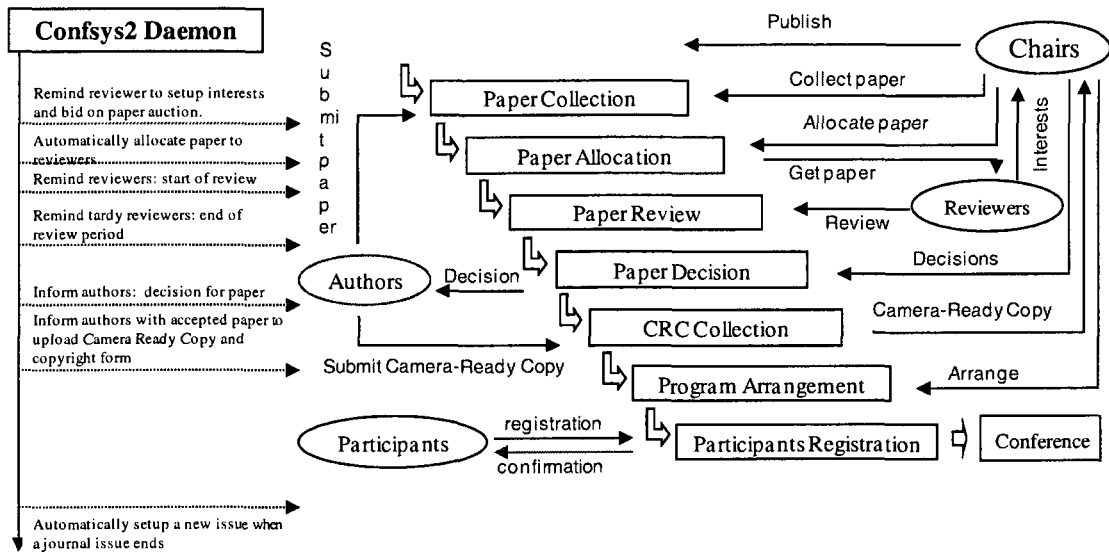


Figure 5.9-1 Confsys2 Smart Daemons

Smart daemons not only could complete repetitive works, but also measure the trends and history of stages of events and perform smart jobs. For example, when the end of review session is near and some reviewers are found have not accessed the papers assigned to them or have not submitted their reviews, the smart daemon will inform such tardy reviewers of the upcoming of the deadline. Other functions of smart daemons include:

- Automatically update DBLP table and update the user relation table, which is used to check for conflict between reviewers and authors

- Automatically set up a new issue based on the cycle for the journal in question

While designed for automatic jobs, Confsys2 daemons could be controlled. Privileged users can set and prevent the daemon from doing certain tasks to allow these to be done

manually or semi-manually. All ConfSys2 daemons are implemented as Java thread classes. Every time ConfSys2 starts up in Tomcat, a daemon loader will automatically start all these daemons. Every daemon will periodically check the control parameters set by administrator or general chairs and decide see if certain tasks should be performed in ConfSys2 scope or in conference scope at given time. When ConfSys2 is shut down by Tomcat, these daemons will be informed to quit. The daemons will perform required clean up work and stop the execution.

There are mainly two types of daemons: mail sender daemons and system maintenance daemons. Mail sender daemons' job is to send reminder mails at given time and certain situations. This type of daemons includes paper auction reminder, review start reminder, tardy reviewer reminder, etc. System maintenance daemons are responsible to update system status to keep the ConfSys2 work at good status. This type of daemons includes DBLP updater, automatic paper allocator, journal issue updater, etc.

Compared to ConfSys, these smart daemons greatly reduce the workload of conference organizer and make such work normalized.

5.10. Dynamic user relationship to prevent conflict of interest

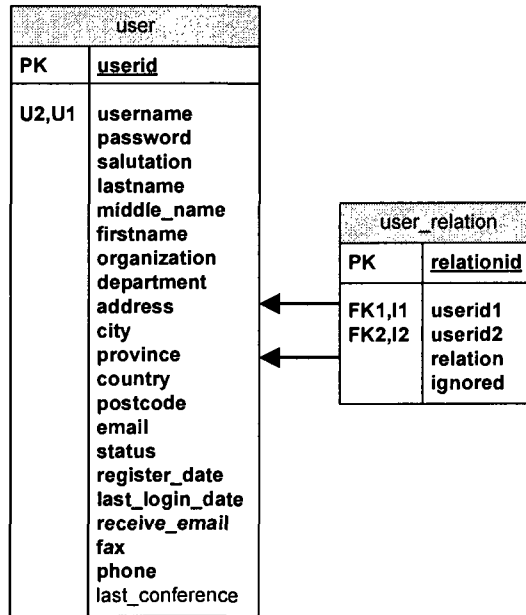


Figure 5.10-1 Dynamic User Relationship in ConfSys2 Database

In conference management, specifically in paper review related phases, a very important task is to prevent conflict of interest. It simply means not let a paper to be assigned to reviewers who may have been coauthors in the past or colleague of one of the authors of a paper. This is to avoid conflict of interests.

ConfSys2 uses dynamic user relationship management to prevent conflict of interest. The Figure 5-10-1 illustrates the database structure to support this management. The table *user_relation* is used to store the relationship between users. It is dynamic because the relationships between users are built dynamically as described below:

First of all, when a person signs up for the system and while filling in his personal information, ConfSys2, will check in existing users who are in the same organization

with this user. And the relationship between these users and the new user are established.

Second, for each update of the DBLP [30] database, a background coauthor relationship check will be performed. Since DBLP is a large database that contain most CS related publications along with the author information, the coauthor relationship between existing users will be updated in the *user_relation* table.

Third, ConfSys2 provide the facility that allows reviewers to manually indicate the conflict of interest in paper auction phase. And when allocating papers to reviewers, this conflict of interest indication will be used to prevent such allocation.

With these automated checking and manual indication, the conflict of interest is avoided in ConfSys2.

Compared to ConfSys's static conflict of interest checking, ConfSys2 dynamic user relationship management has several advantages. First of all, because the DBLP is a huge database, currently with more than 1.2 million records, ConfSys can only perform the conflict of interest check at the time of an automatic paper allocation. It cannot give indication before a paper is assigned to a review. ConfSys2 uses of dynamic user relationship management, reduces the author group, so it can search for the user relationship with a relative small cost and could provide the information and enforce the conflict of interest at any phase. Second, ConfSys2 provide a standard means to store the relationship between users and the relationship can be classified for future check. Third,

the dynamic user relationship management uses ID of users to store the relationship, which makes it faster to be indexed and searched than using text based relationship description. And because the building processes are mostly done by background daemons, this mechanism can provide better response time than in ConfSys.

The use of dynamic user relationship especially brings big difference in automatic paper allocation function. In ConfSys2, we used the similar algorithm as in “CONFSYS: enhancements & integration” [15] to perform automatic paper allocation. The only difference is that ConfSys2 uses the dynamic user relation, which is a pre-processed result of user relationships. The test shows that ConfSys2 takes only 2 seconds to finish the auto paper allocation that ConfSys need 5 minutes to do. This greatly increase the response speed when user performing auto paper allocation, and thus provide better user experience.

6. Experience

ConfSys2 is a redesigned web-based multi-conference management system. Our experiences from maintaining and developing ConfSys have been incorporated into ConfSys2. These experiences have been previously discussed in section 3.3 and 3.4, which are mainly about the limitation we found in ConfSys and obstacles we meet when trying to add and further develop ConfSys. It validates the principle that a good design is very important in software development. A good design needs to consider the user's current requirements, as well as the possible changes that could happen in the future. This is simply because changes are always needed in the real world, and these creates new requirements.

Generalizing concepts in a system is a good way to approach a good design. This normally provides more flexibility in the design, and can accommodate more changes in the future. Adopting some good design patterns is another good way to approach a good design. Because these patterns have been widely discussed and adopted in projects, it is easy to find discussion on them and know their benefits and drawbacks, and evaluate their suitability for a given task.

Through ConfSys2 is yet to be used in "production", we did many unit and system testing during the implementation of the ConfSys2, which were satisfactory. Here are some examples.

After implementing most of the features, a new requirement is proposed to add an indication of whether a certain user will receive Internet email from the system. This is a simple request, but could need to modify the database table structure. In ConfSys, such modification would result in the modification of a number of classes related to the user, or else the system would generate a lot of errors. Since ConfSys2 uses a new designed application architecture which reduced the coupling between layers, and the object-relational mapping modules are implemented with good convention, the new requirement is implemented very easily. First of all, adding new field to the database table did not influence the proper execution of related programs. Second, we modified the related object-relational mapping module to reflect the modifications in data storage layer. This didn't affect the proper execution of related programs in presentation layer. Finally, we modified the programs in presentation layer to let users actually access the new feature. Because the access to such field is through a standard get and set function, the modification work is completed fast without introducing bugs. This shows that our re-designed application architecture does separate the problems at their own layer, and the implementation on the layers is acceptable.

Another example is the request to make the system flexible. In ConfSys, phases in a conference are controlled by hard locks, which mean once a date is passed certain function will not be available, not even for the chair. In such hard lock mechanism, the

only way to access that function is to let chair change the phase date to a time when the function is available, perform the function and reset the phase date. This is very inconvenient for chair's work. In ConfSys2, the problem is solved by not using hard lock. The function will still check if the date has passed. However, it will not prohibit the chair from accessing the function. Instead, it will show chair a warning message to tell the chair that the function is not supposed to be accessed at this time, and let chair override this warning. If the chair overrides this warning, the function would execute and the job will be done. It may appear that the phases are not enforced, however it solves the problem in real life. It happens in real life that a late submitted paper is required to be assigned to reviewers and be reviewed. A hard locked system will only make the operation unnatural, and the warning message mechanism just solves the problem with ease.

We conclude from our experiences that building a good application requires not only a good analysis on proposed requirement, but also need to be flexible for the future. The flexibility of ConfSys2 actually performed well. After implementing ConfSys2, a new requirement was proposed to import older conferences to ConfSys2. With adequate mapping from ConfSys to ConfSys2, this requirement has been implemented without problem.

7. Conclusion and future work

7.1. Conclusion

ConfSys2, a redesigned web-based multi-conference management system, was developed based on the experiences learnt from existing ConfSys system and concepts generalized from the user requirements. The design is based on widely used web application three tier mode, and adopted mature systems and techniques such as Tomcat, MySQL, HTTPS, jQuery, Ajax, etc. Using these design guidelines and technique, ConfSys2 is implemented as a flexible system that can manage multiple conferences with multiple tracks, and supply functionalities and tools to support the entire process of conferences.

Compared to other conference management systems, ConfSys2 has the following features and advantages:

- ConfSys2 is implemented based on mature systems such as Tomcat, MySQL, which can minimize the development cost and provide good technological support
- ConfSys2 is a implemented based on Java technique that can take full advantage of Java language and large amount of modules provided with Java SDK
- ConfSys2 is portable because it does not rely on system features. It can be deployed on both Linux and Windows system that has Tomcat and MySQL

installed.

- ConfSys2 is secure because of its two level protections. It adopts HTTPS to protect security at transmission level, and uses User-Group-Function mapping to ensure the security access at application level.
- ConfSys2 provides a easy to use initialization program to help the administrator deploying it in a new computing system and platform
- ConfSys2 provides a uniform user interface and help system to make it easy to use. And with the support of jQuery and Ajax, the user interface are made intuitive and user friendly.
- ConfSys2 introduced the concept of conference/journal series management, which makes the multi-conference management work in a continuous manner and easy to follow
- ConfSys2 provides a centralized user management function, and propose the community mode that consists of conferences, participants, and papers. This can enhance the communication between conferences.
- ConfSys2 implemented a flexible review system, which allows the conference chairs to customize the review aspects and standings.
- ConfSys2 implemented smart daemons to handle repetitive tasks in conferences and journals. This can greatly reduce the workload of conference organizers.

With these features, ConfSys2 has been implemented as a flexible and powerful web-based multi-conference management system that could enhance the communication between conferences, and greatly save the workload of conference organizers.

7.2. Future work

Though ConfSys2 has been implemented as a multi-conference management system, it has not been tested on managing large amount of conferences in a single installation. There are still further works need to be done. These works include:

- Completing the help system.

Though an online help system has been designed and ready to use, there still lack of detail help information for most functions. These help information is part of the user interface, and would increase the friendliness of the user interface.

- Implementing a backup system

The development work on ConfSys2 was mainly focused on implementing the functionalities of multi-conference management. However, a backup system is missing except for an administrative level backup by a system dumps and copy using *rsync*. This is adequate. However, to keep the data safe, a backup system is required to be implemented in the future. The backup system should not only backup the databases used by ConfSys2, but also back the files associated with the system.

- Testing and analyzing the performance of ConfSys2 when used in managing large

number of conferences. We have considered the possibility of ConfSys2 being used by a large group of users and managing large number of conferences, and our design and implementations is tuned for this situation. However, the real status could be different. If possible, we need to test the performance of ConfSys2 when it's used in managing large number of conferences. Some implementations and algorithms might need to be tune to suit the situation.

Reference

- [1] Conference management software – Google Search, <http://www.google.ca/search?hl=en&q=conference+management+software&meta=>, 2009.08
- [2] EasyChair, <http://www.easychair.org>, 2009.08
- [3] Microsoft's Academic Conference Management Service, <http://cmt.research.microsoft.com/cmt/>, 2009.08
- [4] ConfTool, <http://www.conftool.net/>, 2009.08
- [5] Conference Management System, http://en.wikipedia.org/wiki/Conference_management_system, 2009.08
- [6] ConfSys, <http://confsys.encs.concordia.ca/>, 2009.08
- [7] OpenConf, <http://www.openconf.com/>, 2009.08
- [8] Confious, <http://www.confious.com/>, 2009.08
- [9] ConfMaster, <http://www.confmaster.net/>, 2009.08
- [10] R. Snodgrass, Summary of Conference Management Software, <http://www.acm.org/sigs/sgb/summary.html>, 2009.08
- [11] Stefano Ceri, Piero Fraternali, Maristella Matera, Andrea Maurino, *Designing multi-role, collaborative Web sites with WebML: a Conference Management System case study*, 1st Workshop on Web-oriented Software Technology, 2001
- [12] Franco Zambonelli, Nicholas R. Jennings, Michael Wooldridge, *Developing*

multiagent systems: The Gaia methodology, Transactions on Software Engineering and Methodology (TOSEM) , Volume 12 Issue 3, July 2003

[13] Jutta A. Mulle, Klemens Bohm, Nicolas Roper, Tobias Sunder, ***Building Conference Proceedings Requires Adaptable Workflow and Content Management***, Proceedings of the 32nd international conference on Very large data bases, Pages: 1129 - 1139, 2006

[14] Zhengwei Gu, Xin Jin, Bipin C. Desai, ***CONFSYS: The CINDI Conference Support System***, Seventh International Database Engineering and Applications Symposium (IDEAS'03), 2003 Hong Kong, SAR July 16-July 18 pp 414-418

[15] Yuwei Feng, ***CONFSYS : enhancements & integration***, thesis, Concordia University, 2004

[16] Min Huang, Yuwei Feng, Bipin C Desai, ***CONFSYS: a web-based academic conference management system***, Proceedings of the 2008 C3S2E, Montreal, Quebec, Canada pp 141-143

[17] Min Huang, Yuwei Feng, Bipin C Desai, ***CONFSYS2: An Improved Web-based Multi-Conference Management System***, Proceedings of the 2009 C3S2E, Montreal, Quebec, Canada pp 155-159

[18] Kloptchenko Antonina, Back Barbro, Vanharanta Hannu, Toivonen Jarmo, Visa Ari, ***Prototype-Matching System for Allocating Conference Papers***, hicss, vol. 3, pp.70, 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 3, 2003

- [19] ASP.NET, <http://www.asp.net/>, 2009.08
- [20] Microsoft IIS Server, <http://www.iis.net/>, 2009.08
- [21] Microsoft SQL Server, <http://www.microsoft.com/sqlserver>, 2009.08
- [22] CMT FAQ, http://cmt.research.microsoft.com/cmt/faq/faq_author.asp, 2009.08
- [23] Apache Http Server, <http://httpd.apache.org/>, 2009.08
- [24] PHP, <http://www.php.net/>, 2009.08
- [25] MySQL, <http://www.mysql.com/>, 2009.08
- [26] R. Agrawal, N.H. Gehani, *ODE (Object Database and Environment): the language and the data model*, ACM SIGMOD, 1989
- [27] Web Application – Wikipedia, http://en.wikipedia.org/wiki/Web_application,
2009.08
- [28] HTTPS, <http://en.wikipedia.org/wiki/Https>, 2009.08
- [29] Secure Socket Layer, http://en.wikipedia.org/wiki/Transport_Layer_Security,
2009.08
- [30] DBLP reference site, <http://www.informatik.uni-trier.de/~ley/db/welcome.html>,
2009.08
- [31] Desai, Bipin C., A System for Seamless Search of Distributed Information Source,
<http://users.encs.concordia.ca/~bcdesai/web-publ/w3-paper.html>, 2009.08
- [32] Layered Application, <http://msdn.microsoft.com/en-us/library/ms978678.aspx>,

2009.08

[33] JDBC, <http://java.sun.com/javase/technologies/database/>, 2009.08

[34] SQL, <http://en.wikipedia.org/wiki/SQL>, 2009.08

[35] Coupling (computer science), [http://en.wikipedia.org/wiki/Coupling_\(computer science\)](http://en.wikipedia.org/wiki/Coupling_(computer_science)), 2009.08

[36] Object – Relational Mapping (ORM), http://en.wikipedia.org/wiki/Object-relational_mapping, 2009. 08

[37] Model 2, <http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>, 2009. 08

[38] jQuery, <http://www.jquery.com/>, 2009.08

[39] Ajax, [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)) , 2009.08

[40] Tomcat installation, <http://tomcat.apache.org/tomcat-6.0-doc/setup.html>, 2009.08

[41] MySQL installation, <http://dev.mysql.com/doc/refman/6.0/en/installing.html>, 2009.08

[42] JSP Include, <http://java.sun.com/products/jsp/syntax/1.2/syntaxref1214.html>, 2009.08

[43] JavaScript, <https://developer.mozilla.org/en/JavaScript>, 2009.08

[44] XML, <http://en.wikipedia.org/wiki/XML>, 2009.08

Appendices

Appendix 1. ConfSys2 File Directory Structure.....	108
Appendix 2. Importing a conference from ConfSys to ConfSys2.....	111

Appendix 1. ConfSys2 File Directory Structure

The ConfSys2 is installed by simply copy all files and directories in ConfSys2 installation package to a directory under Tomcat's web application directory, and then call its web-based initialization program. Once ConfSys2 has been initialized, the directory structure should be as given below:

(ConfSys2 Web Application Directory)

```
|--- index.html
|--- [confsys_files]
|   |--- [Conference Subdirectories]
|       |--- [Conference Term Subdirectories]
|--- [images]
|--- [jsp]
|   |--- [dbinit]
|   |   |--- confsys2_init_start.jsp
|   |--- [accessory]
|   |--- [conference]
|   |--- [help]
|   |--- [maintenance]
|   |--- [system]
|   |--- [user]
|--- [scripts]
|--- [styles]
|--- [WEB-INF]
|   |--- web.xml
|   |--- confsys.cfg
|   |--- [classes]
|   |--- [lib]
```

The index.html under the ConfSys2 web application directory is the entry page of ConfSys2 system. It directs the web browser to load the ConfSys2's uniform user

interface.

The [confsys_files] directory contains all user files such as papers, slides, etc. All such files are classified according to the conference and terms the files belong to. Under [confsys_files] directory, there are Conference Subdirectories, whose names are the same as the conferences' database names. And in each Conference Subdirectory, there are term subdirectories. The name of the term subdirectory is in the format of [TermID]_[Start Date]-[End Date].

The [images] directory contains pictures and icons used by ConfSys2 web interface.

The [jsp] directory contains all JSP pages in ConfSys2, and be further classified by their function and stored in subdirectories. The subdirectories under [jsp] include [dbinit], [accessory], [conference], [help], [maintenance], [system], [user]. The [dbinit] contain JSP pages for initializing the main database of ConfSys2. The file *confsys2_init_start.jsp* is the entry page of the initializing program. The [help] directory contains all help pages of ConfSys2. Other subdirectories under [jsp] contain JSP pages related to the functions described by the subdirectory names.

The [scripts] directory contains JavaScript libraries used in ConfSys2 web interface.

The [styles] directory contains the Cascading Style Sheets used to customize the ConfSys2 web page styles.

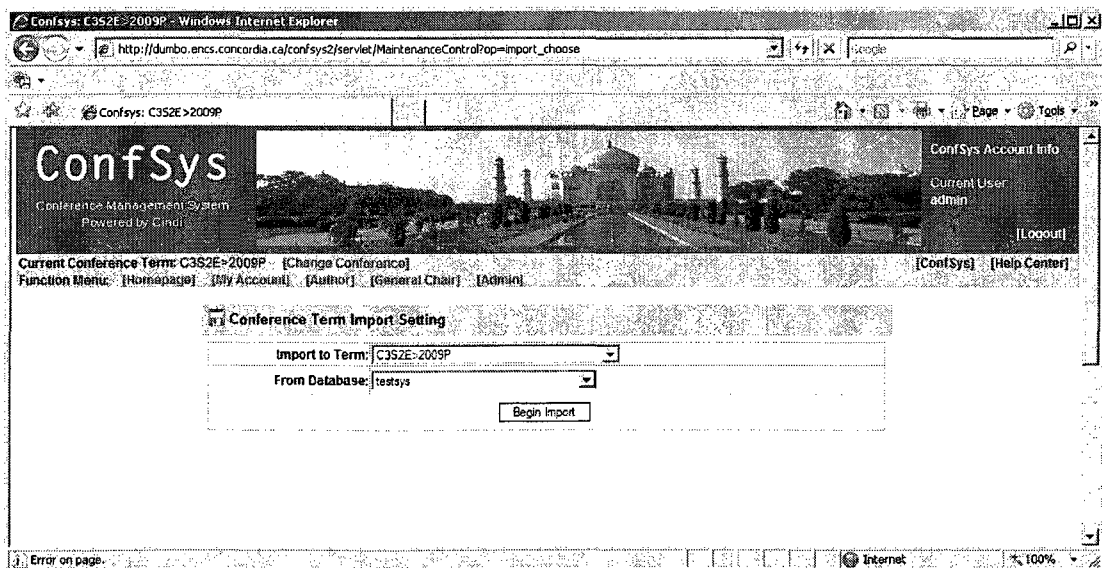
The [WEB-INF] directory contains the ConfSys2 configure, ConfSys2 classes and

Java libraries. The *web.xml* defines the ConfSys2 web application parameters used in Tomcat and Java Servlet mappings. The *confsys.cfg* is created during the initialization of ConfSys2 and contains ConfSys2's main database parameters. If *confsys.cfg* is found in this directory, it means that an initialization has been performed and cannot initialize again. To re-initialize ConfSys2, the *confsys.cfg* must be removed and tables in main database must be erased. The [classes] directory contains all ConfSys2 Java classes and Java servlets. The [lib] directory contains Java libraries that are used by ConfSys2 system.

Appendix 2. Importing a conference from ConfSys to ConfSys2

ConfSys2 provides an importing program which allows ConfSys2 administrator to import a conference from ConfSys to ConfSys2. The requirements and procedures are described below:

1. Before importing a conference from ConfSys to ConfSys2, the ConfSys2 DB account must be granted full access to the ConfSys database which contains the data of the conference. Because ConfSys uses several incompatible table column attributes, ConfSys2 will need to modify these attributes to suit the importing needs. It is suggested to dump the original ConfSys database and then import it to a temporary database for use by ConfSys2 importing program.
2. After the importing database is prepared, a conference and a term will be required to be created in ConfSys2 to accept the import data.



3. Then ConfSys2 administrator can access importing program through the menu [Admin] > [Database Maintenance], and then select “Import conference term from an ancient Confsys database”. In this interface, administrator can select the term to be imported from the “Import to term” list, and the database which to be used as the ConfSys source database. Then administrator can click “Begin Import” to start the importing process.
4. After the import succeeded, the importing program will give a message to specify the directory of the term imported. ConfSys2 administrator can copy the paper related files, such as first version, final version, slides, etc. from ConfSys to that directory, so ConfSys2 users are able to access their paper files through ConfSys2 interface.

After importing the conference from ConfSys to ConfSys2, all roles, such as author, reviewer, etc. in original ConfSys will be imported as general users in ConfSys2. Their roles are kept in the ConfSys2 term. The imported conference is fully functional as a ConfSys2 term, and user roles can perform general conference tasks in the term in question.