Designing an Online Conference Management System

Kevin Daimi and Luming Li
Department of Mathematics, Computer Science and Software Engineering
University of Detroit Mercy,
4001 McNichols Road, Detroit, MI 48221
{daimikj, lilu}@udmercy.edu

ABSTRACT

Academic conferences play a key role in exchanging research ideas between participants and keeping researchers current. Many academic conferences, in various fields, are held annually. This leads to a dramatic increase in the number of submitted papers, and substantial effort to manage these many submissions. Such an intricate workflow of conference management results in frustration among many conference organizers. In this paper, we propose an online system to support the organization, management, and control of academic conferences.

Keywords

Conference Management Systems, Software Requirements Engineering, Software Engineering, Database, User Interface Design

I. INTRODUCTION

For faculty and researchers, attending at least one academic conference annually in their fields of interest is inevitable. In such conferences, many stakeholders are involved in various conference tasks. These include, but are not limited to, program committee chair, program committee members (reviewers), general chair, publicity chair, and authors [3], [4], and [5]. For a conference organization to be successful, a process should be in place. The process of conference organization consists of many phases, such as call for papers, paper submission, paper review, review discussion, paper re-submission, and author notification [5]. Stakeholders with varying viewpoints, in addition to the complex conference organization process, make organizers, especially those without any prior professional organization skills, feel unenthusiastic about managing an academic conference, and possibly quit the task.

With the presence of advanced technology affecting all perspectives of our life, academic conferences are increasing in great number. This is accompanied by an enormous increase in the number of submitted papers. To cope with such large a number of papers and to keep reviewing loads manageable, the number of program committee members has to significantly increase. Consequently, scheduling a face-to-face program committee meeting to review and confer paper submissions is deemed impractical [2].

Based on what is mentioned above, it is vital to develop an online conference management system that facilitates the task of conference organization using the techniques of software engineering. Software Engineering first emerged in the 1968 NATO Software Engineering Conference [10], and [11]. It is defined as “The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software” [11]. Currently, various methods are available to assist software engineers in the analysis, specification, design, implementation, and verification of software products. Furthermore, software developers have improved their comprehension of the activities involved in the software development process. Generally speaking, the software engineering process is divided into the following sub processes: requirement engineering, software design, testing, and maintenance [6].

The first step in the software development process is requirements engineering (RE). RE defines the purpose of a software product, the constraints on it, and the needs of the stakeholders [6], and [12]. RE is generally an intricate interaction and negotiation process involving different stakeholders, such as customers, designers, managers, government and legal bodies, testers, and maintainers. The goal of requirement engineering is to extract functional requirements and non-functional requirements for the software product. In other words, RE deals with making decisions on what needs to be done and when should it be done, the constraints that need to be satisfied, what information the proposed system should provide, and the tools used to achieve the goal [6], [13] and [16]. It should be obvious that requirement engineering is the most important sub-process. The outcomes of requirement engineering are tightly interconnected, and they direct all the subsequent subprocesses of the software development process. A number of software projects have failed due to unclear, ambiguous, inconsistent, and incomplete requirements [6], and [19]. Requirements are listed in natural languages to facilitate understanding them, and various models are used to assist in their analysis. Software engineers and researchers argue that such a combination of the list of requirements and the models used is a superior approach to improve the quality of the requirement engineering process. Modeling techniques are usually communicative, accurate, and promote the development team's specifications and grasping of the requirements. Lists of requirements expressed in natural language represent an agreement between customers and developers, and they abridge requirements management [1], [6], and [14].

The subprocess following requirement engineering is software design. Software design is the fundamental focus of software engineering. During the design phase, the software developer concludes how the software system will be constructed [6], [18], and [20]. The first step in the design is determining the software architecture, which is the set of principal design decisions [7], and [18]. Using the requirements specification, software designers employ the functional requirements mainly to construct the system's architecture [15]. However, this is not an efficient practice for designing high quality software. To achieve high quality, nonfunctional requirements, such as performance, safety, portability, security, and maintainability should be incorporated into the architecture [16]. Software architecture comprises efforst in modeling and representation, design techniques, analysis, visualization, converting the designs into code, furnishing reuse, and deployment [7], and [18].

In this paper, we present an Online Conference Management System to assist the conference chair in organizing the academic conference that follows the process of software engineering. It is a browser/server (B/S) based system, and runs under the .NET platform. The programming language C# is used to implement the system. The rest of the paper is organized as follows: in Section II, system requirements, both functional and nonfunctional, will be presented. Section III will introduce the characteristics, architecture, and user interface of the proposed system. Future work and conclusions are presented in the last section.

II. System requirements

Requirement engineering is the first and most important step of software development. It states the customer's needs that the system must satisfy. The outcomes of requirement engineering ─functional and non-functional requirements─ are critical to the success of any software project [6].

A. *Functional Requirements*

Functional requirements state what functions the system should perform. A sample list representing the functionality of the Online Conference Management System is given below.

1. The system should allow its users (chair, reviewer and author) to sign in using their user name and password.
2. The system should allow its users (chair, reviewer and author) to upload their personal information.
3. The system should allow the chair to submit basic conference information, which includes conference name, URL, subtitles, and main organizer's contact information.
4. The system should allow the chair to set and modify the deadline for paper submission, paper review, acceptance notification, and the uploading of accepted paper (camera ready-copies).
5. The system should allow the chair to set up the topics of interest.
6. The system should allow authors to submit an abstract of a paper in addition to the paper title, authors, their emails, addresses, keywords, and the topic of interest that applies to their paper.
7. The system should allow authors to upload the full paper in a specific format (e.g. PDF file or Word file).
8. The system should allow reviewers to specify the topics that falls into his/her area of expertise.
9. The system should allow reviewers to bid for papers that they are interested in reviewing.
10. The system should allow reviewers to indicate any conflict of interest.
11. The system should be able to assign papers to reviewers automatically.
12. The system should allow a reviewer to submit an evaluation of a paper that was assigned to him/her.

13. The system should allow the reviewer to submit a special comment that can be read by other reviewers.
14. The system should allow the chair to make the final decision on accepting or rejecting a specific paper.
15. The system should allow the chair to assign an accepted paper to a specific conference session.
16. The system should allow the author to upload the camera-ready copy of an accepted paper.

B. Nonfunctional Requirements

Non-functional Requirements describe quality measures by which a software product must abide. In this section, several non-functional requirements are presented. These requirements cover the performance, security, reliability, availability and maintainability of The Online Conference Management System.

1. The system should allow every user to sign in within 5 seconds.
2. The system should be able to display the detailed conference information, within 5 seconds, when requested by chair.
3. The system should be able to list a summary of all the system users' records (authors and reviewers), when requested by the chair, within 5 seconds.
4. The system should be able to display the detailed information of a specific user, when requested by that user, within 5 seconds.
5. The system should be able to list a summary of all the paper submission records, when requested by the chair, within 20 seconds.
6. The system should be able to list detailed paper submission information, when requested by the chair, the author, or the reviewer, within 5 seconds.
7. The system must grant each authorized user a unique ID, user name, and password.
8. The system shall authenticate each user with a unique identification.
9. The system should ensure thatthe user's personal information is confidential.
10. The system should ensure that the user's identification and password cannot be modified by any other person.
11. The system shall guarantee that the paper upload and paper download activities are processed in a strictly secure manner.
12. The system shall send emails to the users in a strictly secure manner.
13. The system should be available 24 hours per day until the conclusion of the conference.
14. The system should be able to detect and correct faults.
15. The system's mean-time-between-failures should be at least 6 mouths.
16. The system should be able to restart after a failure.
17. The system should be able to back up on daily basis.
18. If the system crashes, it should be able to recover using backup copies.
19. The backup copies should be stored in an external storage device and placed in a secure location.
20. The system should be able to correct errors automatically.
21. The system should allow future improvements.
22. The system should be able to generate fault reports automatically.
23. The system should be able to generate various logs to record user's behavior.
24. The system should be able to export fault reports into files.
25. The system should be able to export logs into files.

C. Use Case Modeling

Use case modeling is a graphical representation form of functional requirements [1], [6], and [7]. In this modeling, the use case diagram consists of several possible scenarios related to the usage of the software system. In the object-oriented (OO) analysis and design process, use cases are the key input to the design phase. From a use case diagram, a developer could acquire an unambiguous idea about the system's boundary, actors involved in the system, and the actions the actors can perform. Figures 1 and 2 illustrate the use cases of the Online Conference Management System.

III. SYSTEM DESIGN

Software design is a creative task. During the design phase, a software developer concludes how to implement a software system to meet customers' needs [6], and [18]. A good design should exhibit high cohesion and low coupling [6]. Many design methods have been put forward to achieve this goal. Several case tools have matured to the point that they can help developers efficiently walk through the design process. For our design approach, a UML-based case tool, Sparx Systems' Enterprise Architect, as well as Microsoft Visio are used.

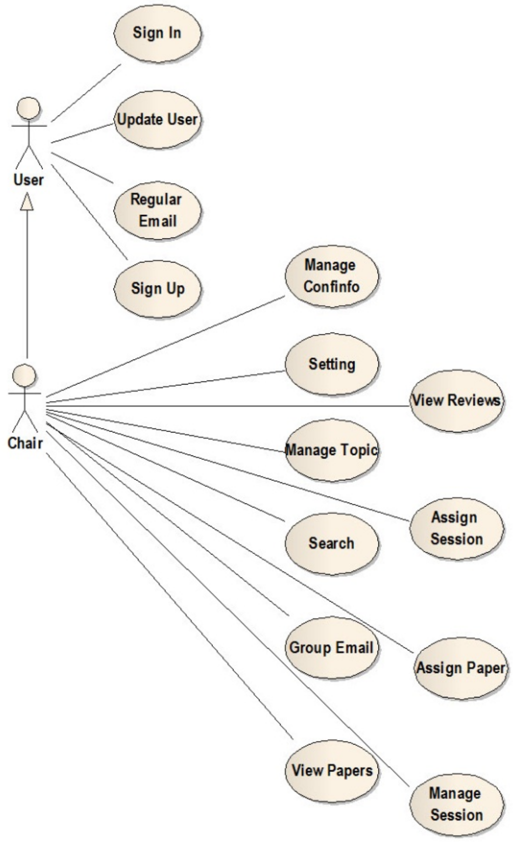


Figure 1: Use Case 1

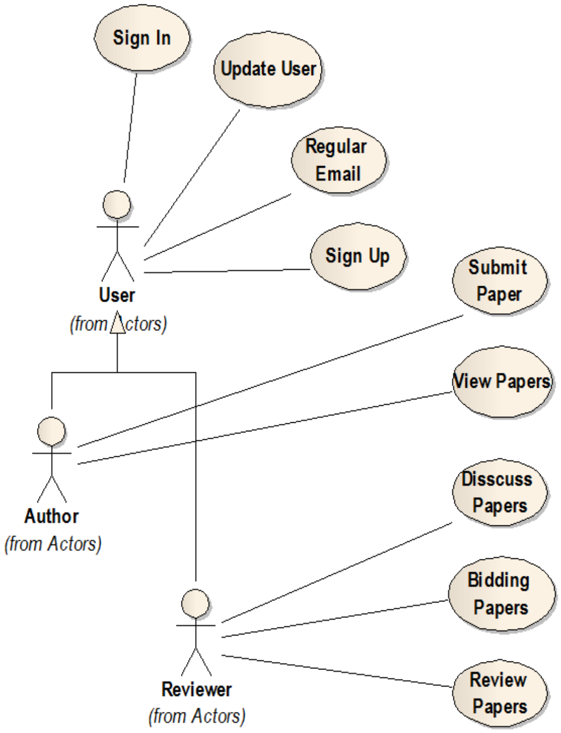


Figure 2: Use Case 2

A. *System Characteristics*

The system we propose is a web-based information management system supporting conference management. Four user groups are defined in the Online Conference Management System:

- System administrator: The system administrator is the person who installs, updates, and restores the system. He/she is also in charge of setting up an account for chairs if their conference request is accepted
- Program chair: The person who makes the overall decisions including assigning papers to reviewers.
- Reviewer: Reviewer is a member of the program committee assigned by the chair to review authors' submissions.
- Author: The person who submits abstracts and full papers.

The Online Conference Management System facilitates the following:

- Covering all the stages of conference management life cycle.
- Allowing the author to submit papers in multiple file formats.
- Automatically assigning papers to a reviewer based on his/her area of expertise and interest.
- Enabling reviewers to make comments and discuss a paper remotely.
- Supporting a group email function.
- Providing phase management support to allow the chair to open and close a conference phase depending on his/her needs.
- Permitting the chair to produce templates for group emails.
- Providing multiple statistics to help the chair in monitoring the review process.
- Granting conference session management to the chair to help him/her prepare the conference agenda.
- Offering a search function to let the chair search for varied information based on varied search criteria.
- Hosting installation on the server. Conference organizers are relieved from problems associated with downloading, installation, and configuration.

B. *System Architecture*

A number of design styles and patterns, such as the Model View Controller (MVC), Presentation–

Abstraction-Control, Philips and REST, are available. [10], [17] and [18]. The proposed system follows the 3-tier architecture. The 3-tier approach encompasses the presentation, the business logic, and the data store layers. Each tier is relatively independent. The application in one tier can request service from a tier below it. To connect different tiers, different connectors are used. Figure 3 illustrates the system's architecture.

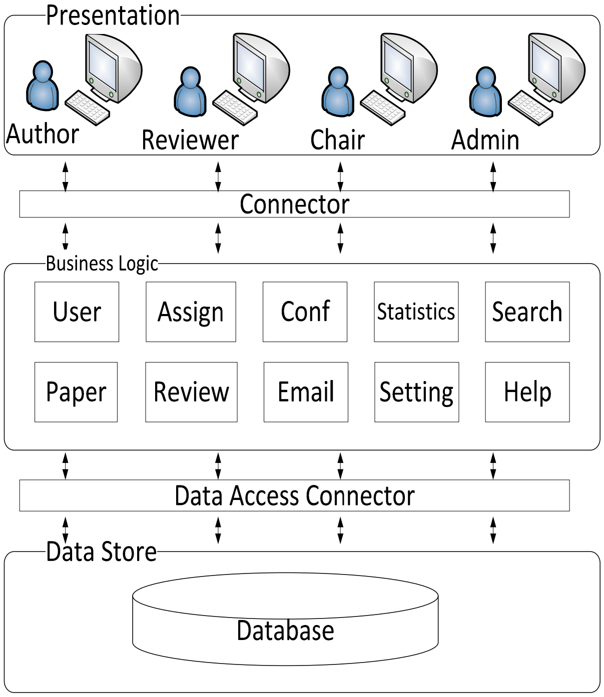


Figure 3: System Architecture

The system's business logic layer consists of 10 components. These components can be further decomposed into several modules. Each module focuses on a specific kind of task. Our goal is to make these components task-independent and easy to reuse. This will enhance the flexibility and maintainability of the entire system. The components are briefly explained below.

User Component: The User Component encompasses two modules; the *Sign in* module to handle the user's sign-in process, and the *Profile* to allow system users to manage their personal information.

Paper Component: This component consists of two modules. The *Edit* module allows authors to edit the abstracts of their papers, and the *Upload* permits authors to upload papers (files).

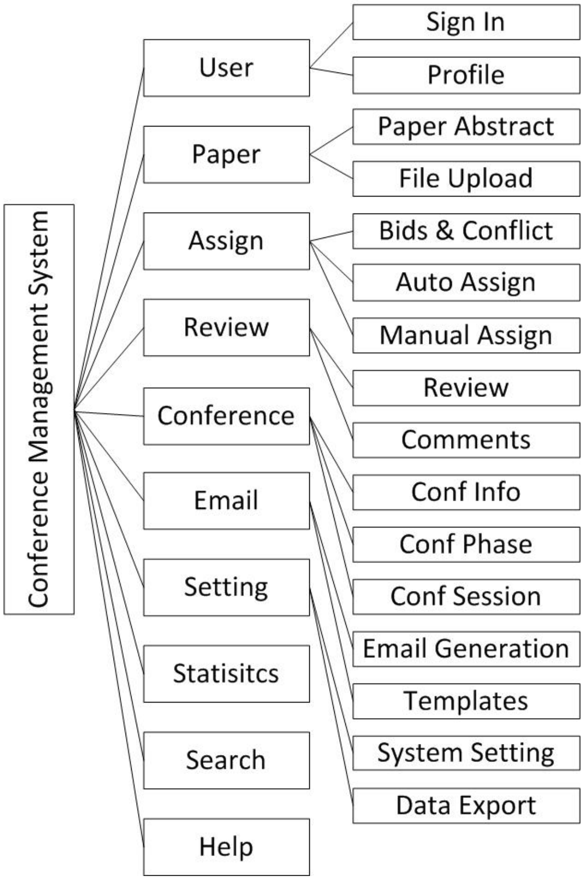


Figure 4: Decomposition of Business Tiers

Assign Component: The Assign component incorporates three modules. The *Bids & Conflicts* module allows reviewers to determine which papers they want to review, and indicate any conflict of interest. This module is optional and can be left out if papers are assigned to reviewers based on other criteria. The *Auto Assignment* module focuses on automatically assigning papers to reviewers based on their expertise and interests. The *Manual Assignment* module allows the chair to override the *Auto Assignment* results and make the final paper assignment decision.

Review Component: This component embraces two modules. The review management is taken care of by the *Rev-Manage* module, and the *Com-Manage* module focuses on comments management, making reviewers' comments available for discussion.

Conference Component: The Conference Component is comprised of three modules. Conference information is handled by the *Information* module. The *Phase* module helps the chair to open/close a conference phase. The *Session* module facilitates conference session management.

Email Component: This component is composed of two modules; the *Template* module is used to help the chair to manage email templates, and the *Composition* module is responsible for composing and sending emails to different users.

Setting Component: The Setting Component includes two modules. The *Change* module helps the chair to change the system's setting. Exporting data is the responsibility of the *Export* module.

Statistics Component: This component provides different statistics to the chair, such as the count of authors and reviewers, and bidding information statistics.

Search Component: The Search Component returns search results to the chair based on search criteria.

Help Component: This component provides detailed help information to assist users in using the system.

C. Database Design

As a core of information exchange and processing, database plays an important role in information systems. It forms the foundation of web services and web-based systems [8]. A very large number of computer applications are database related, and almost every web-based application uses databases to store information. A good database facilitates almost every aspect of an information management system [9].

In our design, the MS SQL Server 2008 is used since it can fully support the .NET platform. Our Online Conference Management System's database is constructed of several database tables, which store various data. Furthermore, a number of stored procedures, which encapsulate the operations on database tables, are used. Table 1 shows the main database tables used in the online conference management system.

D. User Interfaces

User interface plays an important role in any software product. It is through this component that user interaction with the system takes place. In this section, the interface design of the Online Conference Management System is explained. The goal is to make the user interface clear, simple, and friendly. The tool Enterprise Architect, which was constructed by Sparx Systems, is employed to support the user interface design.

Figure 5 presents the starting page of the Online Conference Management System. Different entries for different users are made available. This will facilitate the log-in process. The web page is diveded into five blocks: the header, main menu, navigation bar, main body, and footnote.

TABLE 1
DATABASE TABLES DESCRIPTION

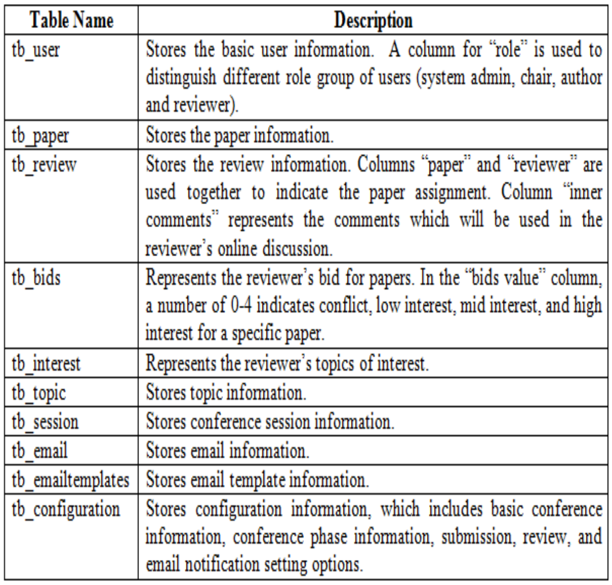
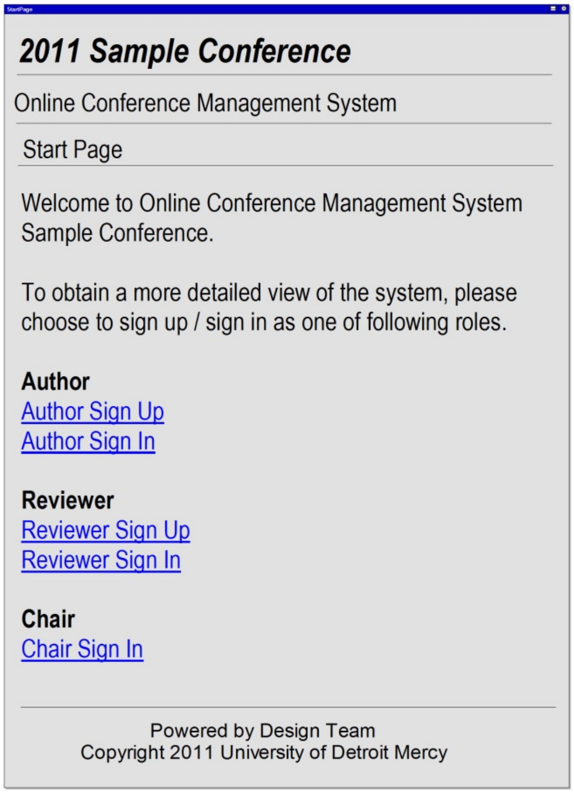


Table Name	Description
tb_user	Stores the basic user information. A column for "role" is used to distinguish different role group of users (system admin, chair, author and reviewer).
tb_paper	Stores the paper information.
tb_review	Stores the review information. Columns "paper" and "reviewer" are used together to indicate the paper assignment. Column "inner comments" represents the comments which will be used in the reviewer's online discussion.
tb_bids	Represents the reviewer's bid for papers. In the "bids value" column, a number of 0-4 indicates conflict, low interest, mid interest, and high interest for a specific paper.
tb_interest	Represents the reviewer's topics of interest.
tb_topic	Stores topic information.
tb_session	Stores conference session information.
tb_email	Stores email information.
tb_emailtemplates	Stores email template information.
tb_configuration	Stores configuration information, which includes basic conference information, conference phase information, submission, review, and email notification setting options.



2011 Sample Conference

Online Conference Management System

Start Page

Welcome to Online Conference Management System Sample Conference.

To obtain a more detailed view of the system, please choose to sign up / sign in as one of following roles.

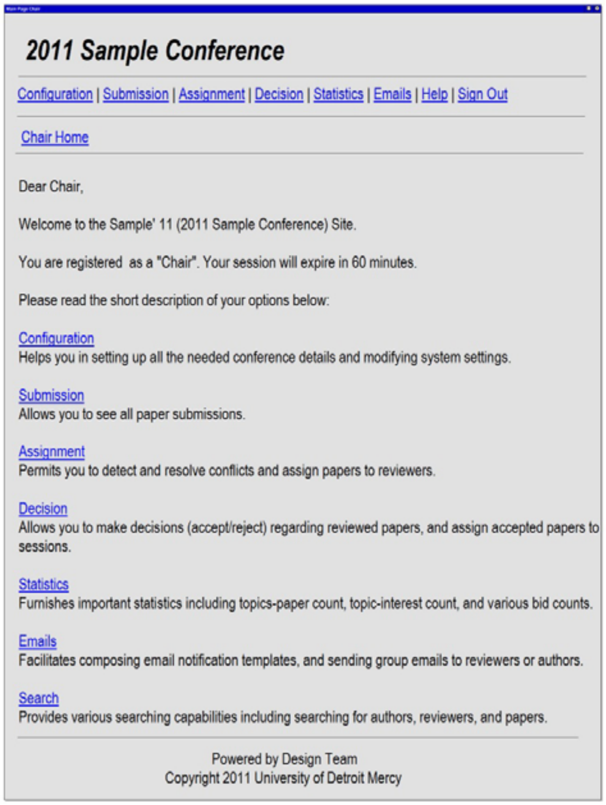
Author
Author Sign Up
Author Sign In

Reviewer
Reviewer Sign Up
Reviewer Sign In

Chair
Chair Sign In

Powered by Design Team
Copyright 2011 University of Detroit Mercy

Figure 5: The Starting Page



2011 Sample Conference

Configuration | Submission | Assignment | Decision | Statistics | Emails | Help | Sign Out

Chair Home

Dear Chair,

Welcome to the Sample' 11 (2011 Sample Conference) Site.

You are registered as a "Chair". Your session will expire in 60 minutes.

Please read the short description of your options below:

Configuration
Helps you in setting up all the needed conference details and modifying system settings.

Submission
Allows you to see all paper submissions.

Assignment
Permits you to detect and resolve conflicts and assign papers to reviewers.

Decision
Allows you to make decisions (accept/reject) regarding reviewed papers, and assign accepted papers to sessions.

Statistics
Furnishes important statistics including topics-paper count, topic-interest count, and various bid counts.

Emails
Facilitates composing email notification templates, and sending group emails to reviewers or authors.

Search
Provides various searching capabilities including searching for authors, reviewers, and papers.

Powered by Design Team
Copyright 2011 University of Detroit Mercy

Figure 6: Chair's Main Page

Fiigure 6 illustrates the chair's main page. All the system's pages have the same header and footer, but the contents of the menubar and main body varies. This will help us maintain a unified style of the system's interface. The menubar of the chair's main page contains the following options:

- **Configuration**: Using the confiruation page, the chair can set up conference information and configurae system settings.
- **Submission**: From the submission page, the chair can view all the papers submittd by different authors.
- **Assignment**: Through the assignment page, the chair can check for and resolve conflicts, and assign papers to reviewers.
- **Decision**: This page allows the chair to see all the reviews made by reviewers. The chair can make the final decision on whether to accept or reject a paper. It also allows the chair to assign accepted papers to conference sessions.
- **Statistics**: This page permits the chair to observe a variety of statistical information, such as bids for papers, bids for contribution, conflicts, and paper submission according to the topic of interest.
- **Emails**: Through the email page, the chair can modify email templates, and send emails to a spcific user or a group of users.
- **Search**: At the search page, the chair can search for papers, reviews, and comments using different searching criteria.

IV. Conclusions

With the increase in the number of academic conferences, in the number of papers submitted to such conferences, as well as the complexity of managing such conferences, it is critical to promote an online conference management system that facilitates the task of conference organization using the software engineering process. This paper presents the requirement analysis and design of An Online Conference Management System. Both functional and non-functional requirements are examined. The architecture and data modeling of the system are introduced. Furthermore, a user interface design is proposed and is being implemented. A C# implementation of the system is in progress. For our final version of the system, we plan to improve the system's design to support multi-track conferences and multi-conferences.

References

[1] J. Arlow, I. Neustadt, *UML 2 and the Unified Process*, Upper Saddle River, NJ: Pearson Education, 2005, ch.1 and 4.

[2] M. Franklin, "Rethinking the Conference Reviewing Process," *in Proc. 2004 ACM SIGMOD International Conference on Management of Data*, New York, 2004, pp. 957-957.

[3] M. Huang, Y. Feng, and B. Desai, "CONFSYS2: An Improved Web-Based Multi-Conference Management System," *in Proc. 2nd Canadian Conference on Computer Science and Software Engineering,* Montreal, Canada, 2009, pp. 155-159.

[4] M. Huang, Y. Feng, and B. Desai, "CONFSYS: A Web-based Adademic Conference Management System," *in Proc. Canadian Conference on Computer Science and Software Engineering,* Montreal, Canada, 2008, pp. 141-143.

[5] P. Noimanee, and Y. Limpiyakom, "Towards a RESTful Process of Conference Management System," *in Proc. International Multi Conference of Engineers and Computer Scientists*, Hong Kong, 2009, pp. 991-995.

[6] S. L. Pfleeger and J.M. Atlee, *Software Engineering Theory and Practice,* Upper Saddle River, NJ: Pearson Higher Education, 2010, ch. 4- 6.

[7] R. N. Taylor, N. Medvidvic and E. M. Dashofy, *Software Architecure Fondations, Theory and Practice,* Hoboken NJ: John Wiley & Sons, 2010, ch. 3-4.

[8] G. Post and A. Kagan, "Database Management Systems: Design Considerations and Attribute Facilities," *Journal of Systems and Software*, Vol. 56, No. 2, pp. 183-193, Mar. 2001.

[9] R. Stephons, *Beginning Database Design Soultions*, Indianapolis, IN: John Wiley, 2008, ch. 1-3.

[10] P. Naur, and B. Randell, "Software Engineering: Report of a Conference Sponsored by the NATO Science Committee," *in Proc. NATO Software Engineering Conference,* Garmisch, Germany, 1968, pp.9-65.

[11] J. A. Wang, "Towards Component-Based Software Engineering," *Journal of Computer Science in College,* Vol. 16, No. 1, pp. 177-189, Oct. 2000.

[12] B. Nuseibeh, and S. Easterbrook, "Requirement Engineering: A Roadmap," *in Proc. 22nd International Conference on Software Engineering*, Limerick, 2000, pp. 35-46.

[13] A. Aurum, and C. Wohlin, "The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process," *Information and Software Technology*, Vol. 45, No. 14, pp. 945-954, Nov. 2003.

[14] J. Nicolás, and A. Toval, "On the Generation of Requirements Specification from Software Engineering," *Information and Software Technology,* Vol. 51, No. 9, pp. 1291-1307, Sep. 2009.

[15] J. Bosch, and L. Lundber, "Software Architecture – Engineering Quality Attributes," *The Journal of Systems and Software*, Vol. 66, No. 3, pp. 183-186, Jun. 2003.

[16] E. Folmer, and J. Bosch, "Architecting for Usability: a Survey," *Journal of Systems and Software*, Vol. 70, No. 1-2, pp. 61-78, Feb. 2004.

[17] L. Bass, and B. E. John, "Linking Usability to Software Architecture Patterns Through General Scenarios," *Journal of Systems and Software*, Vol. 66, No. 3, pp. 187-197, Jun. 2003.

[18] R. N. Taylor, and A. Hoek, "Software Design and Architecture: The Once and Future Focus of Software Engineering," *in Proc. 2007 Future of Software Engineering*, Washington, DC, 2009, pp. 226-243.

[19] R. N. Ferrari, and N. H. Madhavji, "Architecting-Problems Rooted in Requirements," *Information and Software Technology*, Vol. 50, No.1-2, pp. 53-66, Jan. 2008.

[20] A. Tang, A. Aleti, J. Burge, and H. Vliet, "What Makes Software Design Effective?" *Design Studies*, Vol. 31, No. 6, pp. 614-640, Nov. 2010.