

CRS: Registration Extension of the OpenConf™ Conference Management System

Ihsan Mohd Yassin^{#1}, Yuslindawati Md Yusof[#], Aiman Johari[#], Azlee Zabidi[#], Hasliza Abu Hassan^{*}

[#]*Faculty of Electrical Engineering, Universiti Teknologi Mara
40450, Shah Alam, MALAYSIA*

¹ihsan.yassin@gmail.com

^{*}*Faculty of Electrical Engineering, Universiti Industri Selangor,
Bestari Jaya, Malaysia*

Abstract— Academic conferences provide a social platform for participants to present, learn and discuss about new and interesting findings of their research. Management of these conferences typically utilize Conference Management Systems (CMSs) to simplify its management. However, most CMSs lack the registration function, which is a key component apart from the submission and review process. We describe modifications of the popular OpenConf™ CMS to include participant registrations and payment acceptance. The process involves adding several new tables to the existing database structure, as well as to develop the new interface for the added functionality. The modifications are detailed in this paper.

Keywords— *Conference Management Systems (CMSs); academic conference; conference management.*

I. INTRODUCTION

Academic conferences provide a social platform for participants to present, learn about, and discuss current research between them. These events are appealing because they create an environment that supports mutual revelation (participants can present about their works and learn about others). Typically, academic conferences encourage different types of interactions between participants, namely formal (presentation of keynotes, research papers, and posters) and informal (casual exchanges between participants during breaks and receptions) [1].

Conference management is a concept of planning and delivery of a conference and post-conference review [2]. Organization of an academic conference is divided into several processes, namely submission, review, decision, and camera-ready paper collection [3]. These processes involve much correspondence between the organizer, participant and reviewers. Because of this, many Conference Management Systems (CMSs) offer online software packages to simplify the management of these conferences. Reference [4] provides an excellent analysis of online CMS available. OpenConf™ [5] is an online CMS offered by the Zakon Group LLC. It is available in two versions – the free Community edition (which performs basic CMS features) and the paid Professional edition (with additional features). The OpenConf™ system has been used to manage various conference, research proposal application and journal submission systems [6].

Apart from the four processes listed, there are additional processes not supported by most CMSs, including

OpenConf™. They are presenter registration, non-presenting participant registration and payment acceptance. These functionalities are offered by established and paid CMS solutions (such as EDAS [7]).

We propose an extension of the OpenConf™ CMS to perform these additional processes in order for it to become a comparable free CMS solution. In this paper, we describe the implementation, as well as describe key design considerations for the extension.

The rest of this paper is organized as follows: Sections II and III describes the OpenConf™ CMS. This is followed by proposed modifications to the original OpenConf™ system (Section IV). The implementation of the proposed extension is described in Section V. Finally, we conclude our findings in Section VI.

II. ORGANIZING AN ACADEMIC CONFERENCE

As mentioned before, there are several processes involved in organizing an academic conference, namely submission, review, decision and collection of camera-ready submissions. During the submission process, the author/potential participant is required to submit their draft research paper to the organizer, after which the organizer returns some type of confirmation that the paper was received. During the review phase, the program committee assigns submitted papers to reviewers for an independent evaluation of the quality. Reviewers give aggregated decisions [8] and feedback to the organizer by giving comments on the technical merit of the paper as well as suggestions on how the paper can be improved. Based on the reviews, the organizer will then decide whether the paper should be accepted for the conference. After decisions are made, the authors are informed and are required to amend their paper based on suggestions by reviewers. Finalized papers will then be collected and published in conference proceedings.

Apart from these processes, additional processes such as presenter registration, non-presenting participant registration and payment acceptance are typically not supported by most CMSs. The presenter registration process links presenters to papers that they will present at the conference. This is necessary because: 1) most papers have multiple authors, and

‘9999’ was chosen to indicate that the participant is non-presenting.

Table *proof* was used to store the path of the uploaded scanned copies of proof of payments, student identification cards (to be eligible for student discounts), and professional membership cards (to be eligible for professional membership discounts). Since many files can be uploaded by the same presenter, the relationship between *presenter* and *proof* is one-to-many.

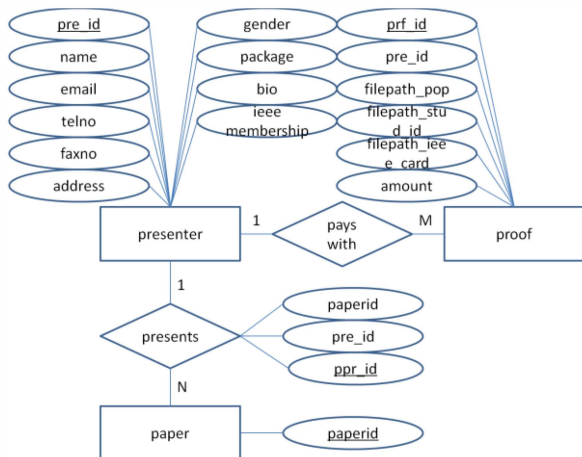


Figure 4. ERD of proposed modifications to OpenConf™

B. Functionality Requirements

In this section, several expected functions of the new extension are defined. The system must cater to two types of users, namely the presenter and the administrator. Several tasks for the presenter are defined:

1. Enter contact information and biography: The system would provide interface to obtain contact information and biography from the presenter. Contact information is required for the organizer to correspond with the presenter, while presenter biography is required to introduce the presenter prior to his/her presentation.
2. Select paper to present: After entering the contact information, the system should obtain information from the presenter on which paper he/she will present. A paper may be presented by one presenter only. Therefore the system must ensure that after a paper has been selected by a presenter, it cannot be selected by other authors.
3. Get information on fees due: After papers have been selected, the system should display the amount of fees due. Most conferences have different fee structures for students, international participants, and/or members of specific organizations. Furthermore, a conference may permit presentation of multiple papers under one registration. All these requirements should also be included in the system.
4. Upload Proof of Payment (PoP), student ID, membership cards, etc: The system should be able to accept a maximum of three uploaded files from the user. The PoP file indicates a successful deposit into the conference

account. Optional student ID and membership cards are required for fee discounts.

5. Submission of camera-ready paper: The system should provide a link to OpenConf™ so that authors can upload their camera-ready papers.

Administrator tasks are focused towards monitoring and updating presenter information. The administrator tasks are:

1. View participant information: The system should display the records of individual participants in order to check their registration status.
2. Link & unlink papers: The system should allow the administrator to assign papers to presenters, as well as to un-assign papers from them.
3. Monitor number of participants & registered papers: The system should display the number of presenting/non-presenting participants for track assignment or logistics purposes.

C. Implementation in localhost

The system was developed on a Personal Computer (PC) installed with Apache to convert it to a web server. The web server would be used to host the web pages and the database, which can then be accessed by any web browser locally or from any computer network.

The typical 3-tier client server architecture was implemented, as shown in Fig. 5 and Fig. 6. The 3-tier client-server architecture is organized as follows [9]:

1. First (client) tier - This tier consists of client applications such as the web browser. This tier is responsible to control the presentation logic to the client.
2. Second (business) tier - This tier typically consists of an application server, which hosts distributed applications for the client to access.
3. Third (enterprise information systems) tier - This tier is responsible in storing the Resource Planning Systems (ERPS), mainframe-transaction-processing systems, databases, and legacy applications of the enterprise.

As can be seen from Fig. 5 and Fig. 6, the client-side communicates with the server via a thin client. The thin client is the web browser, which is responsible to process the presentation logic. The server controls the business logic, and will have access to the database. Any request made by the client will be processed by the server.

D. Database & Website Development

MySQL was chosen as the Database Management System (DBMS). The structures for the new tables are shown in Table III.

The website provides a user interface for the user to use the system. All scripts were written in PHP. The website was developed using an open-source Content Management System (CMS) called Joomla!.

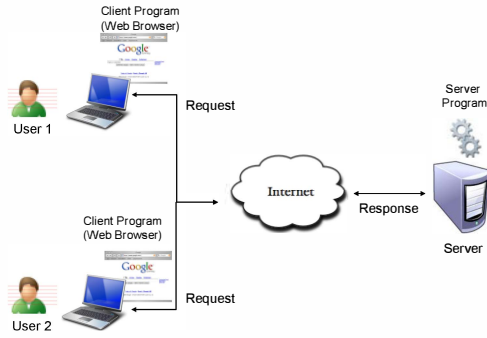


Figure 5. Client-server implementation on localhost

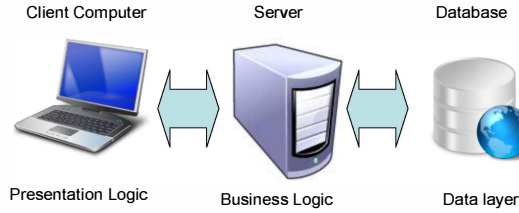


Figure 6. Three-tier structure adopted in the project

```
CREATE TABLE IF NOT EXISTS PRESENTER (
  PRE_ID INT(255) NOT NULL AUTO_INCREMENT,
  PRE_NAME VARCHAR(1000) NOT NULL,
  PRE_PACKAGE VARCHAR(1000) NOT NULL,
  PRE_GENDER VARCHAR(1000) NOT NULL,
  PRE_IEEE VARCHAR(1000) NOT NULL,
  PRE_IEEENO VARCHAR(1000) NOT NULL,
  PRE_TELNO VARCHAR(1000) NOT NULL,
  PRE_FAXNO VARCHAR(1000) NOT NULL,
  PRE_EMAIL VARCHAR(1000) NOT NULL,
  PRE_CORRADD VARCHAR(1000) NOT NULL,
  PRE_COUNTRY VARCHAR(1000) NOT NULL,
  PRE_BIO VARCHAR(5000) NOT NULL,
  PRIMARY KEY (PRE_ID)
) ENGINE=MYISAM DEFAULT CHARSET=LATIN1
AUTO_INCREMENT=1;

CREATE TABLE IF NOT EXISTS PRESENTS (
  PPR_ID INT(255) NOT NULL AUTO_INCREMENT,
  PAPERID INT(255) NOT NULL,
  PRE_ID INT(255) NOT NULL,
  PRIMARY KEY (PPR_ID)
) ENGINE=MYISAM DEFAULT CHARSET=LATIN1
AUTO_INCREMENT=1;

CREATE TABLE IF NOT EXISTS PROOF (
  PRF_ID INT(255) NOT NULL AUTO_INCREMENT,
  PRE_ID INT(255) NOT NULL,
  PRF_DUE VARCHAR(1000) NOT NULL,
  PRF_REFNO VARCHAR(1000) NOT NULL,
  PRF_FILEPATH VARCHAR(1000) NOT NULL,
  PRF_COMMENTS VARCHAR(1000) NOT NULL,
  PRF_FILEPATH2 VARCHAR(1000) NOT NULL,
  PRF_COMMENTS2 VARCHAR(1000) NOT NULL,
  PRF_FILEPATH3 VARCHAR(1000) NOT NULL,
  PRF_COMMENTS3 VARCHAR(1000) NOT NULL,
  PRIMARY KEY (PRF_ID)
) ENGINE=MYISAM DEFAULT CHARSET=LATIN1
AUTO_INCREMENT=1;
```

Figure 7. Structure for new tables

V. RESULTS & DISCUSSIONS

A. User Interface

Fig. 8 shows the main interface of the proposed system. The main interface lists the tasks that need to be performed by the user to complete the registration process. Menu items guide the registration process. The administration tab allows access to administrative tasks.

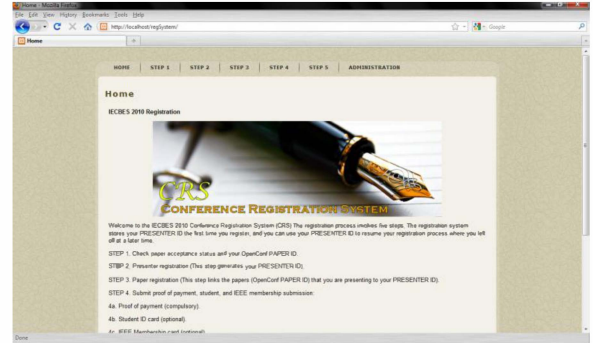


Figure 8. Main interface for proposed Conference Registration System (CRS)

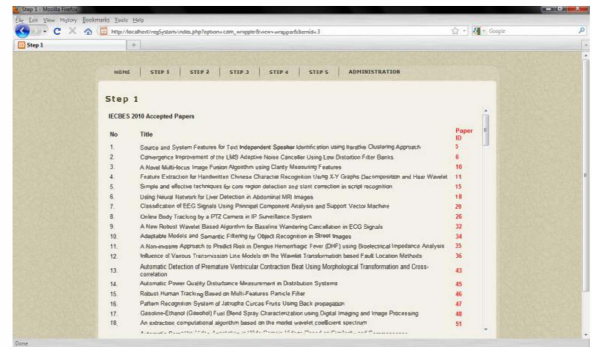


Figure 9. Step 1 – view list of accepted papers

In Step 2 (Fig. 10), the participant is required to enter his/her contact information and biography. After the contact information is submitted, the system generates the presenter id (PRE_ID) for the registration. The PRE_ID is needed to proceed to the next registration step.

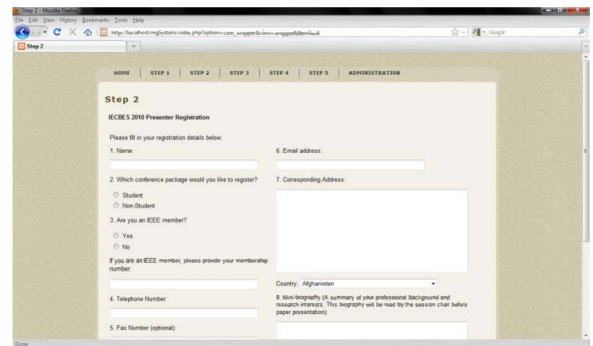
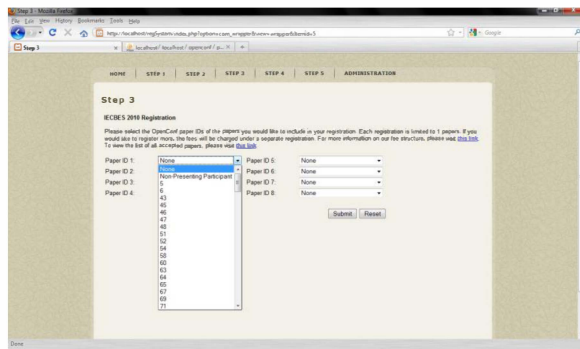


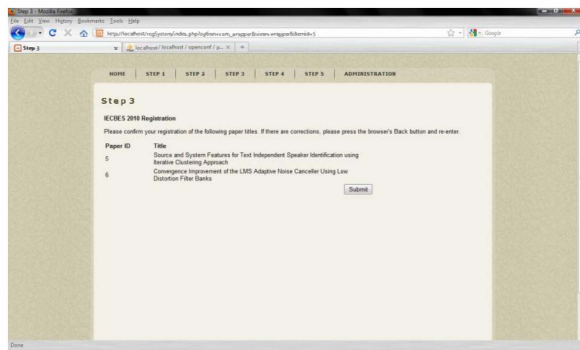
Figure 10. Step 2 – enter contact information and biography

In Step 3 (Fig. 11), the system will assign the presenter with the selected paper id. As a security measure, the presenter needs to enter his/her presenter id and email before being able

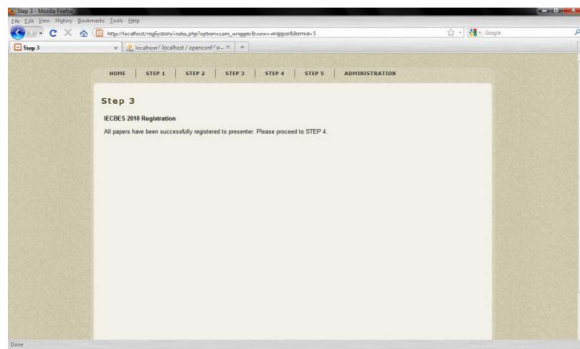
to do the assignment. Non-presenting participants can choose the “Non-presenting participant” option to register.



(a)



(b)



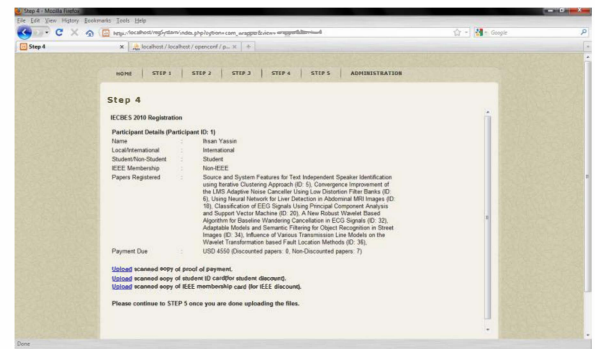
(c)

Figure 11. Step 3 – (a) select paper id to link with presenter. (b) confirmation page. (c) link successful.

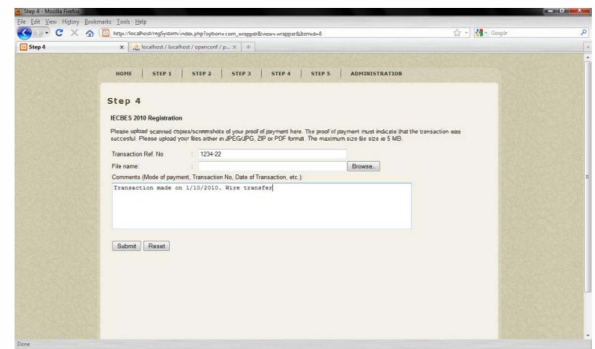
Step 4 (Fig. 12) confirms the papers selected by the presenter, and automatically calculates discounted fees (if applicable). This page also allows the participant to upload the PoP, student identification card, and professional membership card as proof to get the discount rates. Step 5 (camera-ready paper submission) is done through the OpenConf system.

B. Administrator Interface

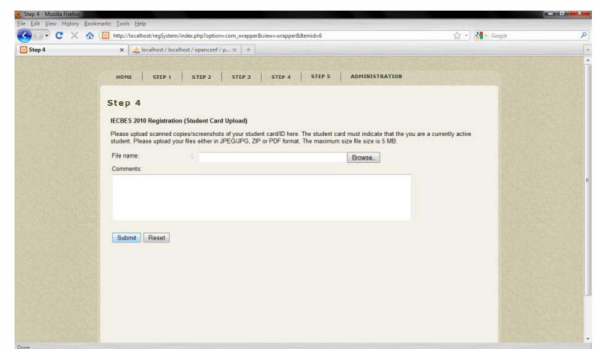
The administrator interface is shown in Fig. 13 and Fig. 14. Tasks that can be performed by the administrator are monitoring of papers and participants, as well as linking and unlinking papers. Before entering this page, the administrator is required to enter a username and password as a security measure. New fees are automatically calculated as papers that are linked or unlinked.



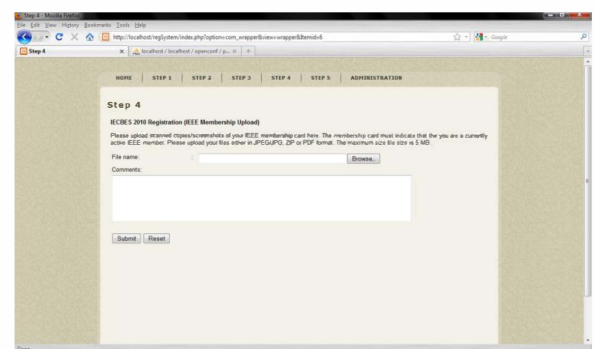
(a)



(b)

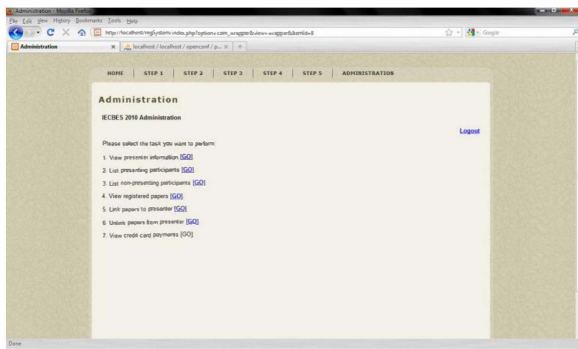


(c)

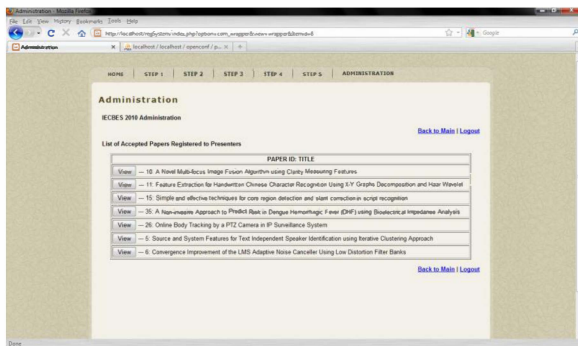


(d)

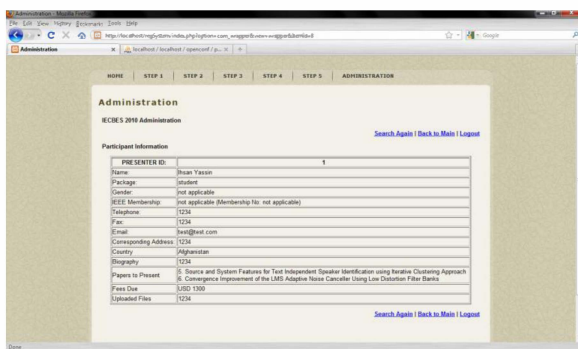
Figure 12. Step 4 – (a) confirmation page. (b) file upload (PoP). (c) file upload (student card). (d) file upload (professional membership).



(a)



(b)

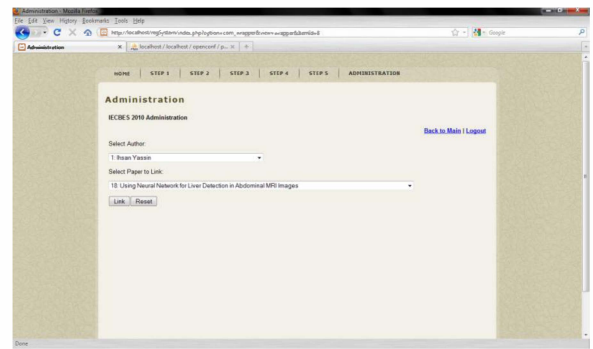


(c)

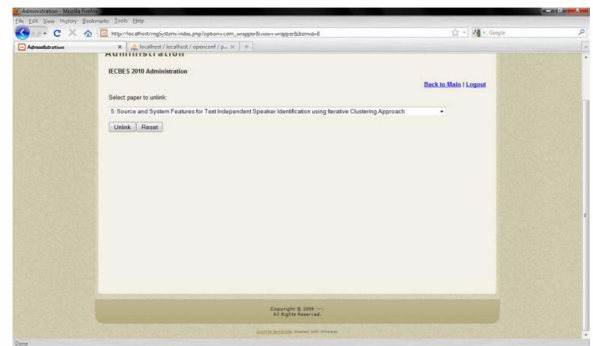
Figure 13. Administrator page – (a) main interface. (b) paper/presenter monitoring. (c) view records (d) file upload (professional membership).

VI. CONCLUSIONS & FUTURE WORK

An extension of the OpenConf™ CMS for registration purposes is presented in this paper. The modifications involve adding several new tables to the existing OpenConf™ database in order to store presenter and proof of payment information. We are currently exploring means to integrate credit card payments into the existing system to offer services similar to other CMS [7].



(a)



(b)

Figure 14. Administrator page (cont.) – (a) link paper. (b) unlink paper.

REFERENCES

- [1] J. F. McCarthy, D. W. McDonald, S. Soroczak, D. H. Nguyen, and A. M. Rashid, "Augmenting the Social Space of an Academic Conference," in *Proc. of the 2004 ACM Conf. on Computer Supported Cooperative Work (CSCW)*, 2004, pp. 39-48.
- [2] B. Subramaniam, "WEBCON: An Ontology Engine Based Conference Organizer," MSc. Thesis, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, 2008.
- [3] L. Padgham, J. Thangarajah, and M. Winikoff, "The Prometheus Design Tool - A Conference Management System Case Study," *Agent-Oriented Software Engineering VIII, 8th Int. Workshop, (AOSE 2007)*, pp. 197-211, 2007.
- [4] MadhurJain, T. K. Tewari, and S. K. Singh, "Survey of Conference Management Systems," *Int. J. Computer Applications*, vol. 2(2), pp. 14-20, 2010.
- [5] Z. G. LLC. (2010, 13th August). *OpenConf Conference Management Peer-Review Software & Hosting Service* [Online]. Available: <http://www.openconf.com>
- [6] Z. G. LLC. (2010, 13th August). *OpenConf Portfolio* [Online]. Available: <http://www.openconf.com/portfolio>
- [7] EDAS. (2010, 13th August). *EDAS Webpage* [Online]. Available: <http://edas.info>
- [8] J. R. Douceur, "Paper rating vs. paper ranking," presented at the Proc. Conf. on Organizing Workshops, Conferences, and Symposia for Computer Systems, San Francisco, CA, 2008.
- [9] J. McGovern, R. Adatia, Y. Fain, J. Gordon, E. Henry, W. Hurst, A. Jain, M. Little, V. Nagarajan, H. Oak, and L. A. Phillips, *Java™ 2 Enterprise Edition 1.4 Bible*. Indianapolis, IN: Wiley Publishing, Inc., 2003.