Redesign of a journal editorial workflow management system

Yijing Zeng

A Thesis in the

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science

(Quality Systems Engineering)

Concordia University

Montreal, Quebec, Canada

July 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Yijing Zeng

Entitled: Redesign of a journal editorial workflow management system

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Chair
Dr. Chun Wang

______________________ Examiner
Dr. Lingyu Wang

______________________ Examiner
Dr. Hua Ge

______________________ Supervisor
Dr. Yong Zeng

Approved by ______________________________
Chair of Department or Graduate Program Director

____
Dean of Faculty

Date ______________________________

ABSTRACT

Redesign of a journal editorial workflow management system

Yijing Zeng

This thesis presents the redesign and the implementation of an online journal editorial workflow management system. The previous version of the journal management system has the following problems: 1. It only supported the submission and review processes which are not enough to cover a complete editorial workflow for a journal; 2. It is neither adequate for user requirements nor it provides enough functions. Hence we are proving a new journal management system with the following features: 1. Complete the entire editorial workflow with the paper proofreading and publication processes; 2. Functions to help journal editors to manage the processes of academic journal. Our new design is based on Environment Based Design (EBD), an innovative design methodology. We used EBD to analyze the problems of the system and clarify the editorial workflow and user requirements, and to generate appropriate solutions. Our results show significant improvements based on user studies. The new system has already been used as an online submission system of an international scientific journal. We can continue to evolve the system with new feature like automated management easily because of the flexibility of the design.

Acknowledgements

I would like to express my gratitude to my supervisors Dr. Yong Zeng, for his invaluable help, guidance, and encouragement during my master study and research. I thank all the CIISE professors and staffs who have given me as well as other students a friendly academic environment.

I would also like to thank my colleagues in the Design Lab. I thank all the people worked with me in the Design Lab: Thanh An Nguyen, Suo Tan, Xu Xu, Xuan Sun, Xiaoguang Deng, Xiaoying Wang, Ronaldo Gutierrez, Wei Liu, Yi Dou, and Mengli Shu.

I send all my love to my family of father, mother, auntie, and my cousin. My cousin Haowen Ning, I sincerely thank him for his selfless help whenever I need him.

Table of contents

List of Figures

List of Tables

1 Introduction

1.1 Problem statement

The Journal of Integrated Design and Process Science[1] (JIDPS) is an official journal of the Society for Design and Process Science (SDPS). The journal serves to promote research of transdisciplinary design and process science from diverse disciplines such as mathematics, computer science, economics, engineering, management science, natural sciences, and social sciences. As an important part of the journal[2], there is an online submission system to manage the editorial workflow of manuscripts. It was originally developed by the Design Lab, Concordia University. The development of this system was based on concrete journal management requirements provided by the editors, authors and reviewers of the journal. Dr. Yong Zeng, one of the Editors in Chief of JIDPS to manage the routine work and also the supervisor of this thesis project. In 2013, the first version of the journal system was implemented and used in several journals.

However, the old system does not support its users in managing some crucial processes like the proofreading and the publishing. Also the old system was not capable of meeting functional and design requirements for system users. In addition, the old style interface doesn't have the mechanism for workflow management, and not easy for implementing the new functions and requirements. Its old software architecture prevents the journal management system from being further developed. Hence it is really important and necessary to redesign the system, helping it to evolve with the needs and to address the requirements of different users.

Due to the challenge of low functions in the old system, the changes of its functionality and the interface are imperative[3, 4]. The journal editorial workflow management system provides a network-based and three-dimensional collaborative paper processing platform for authors, journal editors and expert reviewers. The emerging demand for the redesign of the system not only reduces the workload of the journal editors, but also shortens the published cycle of our journal with higher efficiency. At the same time, it can also increase the system user's satisfaction, so that the journal can get the high-quality resource of papers easily[5].

1.2 Objectives

Based on the problem statement above, a redesign of the journal system is necessary. Our objective is to provide a new journal system with the following features: 1. Design the workflow and data model of the new system; 2. Functions to help journal editors to manage the processes of academic journal. To achieve our objective, we have divided our work into two major tasks in our redesign process:

1. Analysis of the redesign problem

As a real project, it aims to adapt an old journal management system to user needs within a short period with a certain deadline. To solve this redesign problem, questions like how to gather the necessary user requirements and how to design the workflow and data model of the new system should be analyzed and addressed. We will use an effective methodology called Environment Based Design (EBD), first proposed by Dr. Yong Zeng, to find a good solution of this redesign problem. Through the analysis process, more and more information will be gathered together to clarify requirements from the actual environment. We will not only use the EBD to collect user requirements, but also to design the entire new workflow of the current journal and the new data model of the new system.

2. Redesign and implement the new journal editorial workflow management system

We should design the new journal management system and implement it. As the result of the analysis of the redesign problem, our journals may have different operations, processes, rules and roles. The new system should strive for wider acceptance and be able to provide new functions for the new set of users to facilitate journal editorial workflow management and organization. Same as the old system, the new journal management system is an online tool used for the administration and organization of the Journal of Integrated Design and Process Science. The old system was originally developed by our design lab from scratch while the new system will be developed based on the previous work. We will incorporate the old and the forthcoming environment and technological elements to finish the architectural and visual overhaul of the system, including integration with the previous system.

1.3 Contributions

The result of our project is developed a new journal editorial workflow management system which derived from the redesign of the old system. We put our effect on analyzing the redesign problem and mainly worked on the design of the workflow and data model of the new system. The main contributions of this present thesis can be summarized as follows:

1. Proposed a new way to design the system workflow and data model

In this paper, we describe the application of the EBD methodology to design the workflow and data model of the new journal editorial workflow management system. The Environment Based Design methodology has been proven in many fields, for example the mechanical and concept design. Our thesis extends the range of this design theory to a new area which is the system workflow and data model design. Although many traditional methods such as interviews and brainstorming can also solve the design problem, they are unstructured methods and strongly rely on the user's experience. Compared with those methods, the methodology we adopted is a structured method which does not require the user to have a lot of experience. The Environment Based design theory ensures that the EBD approach to workflow and data model design can always generate the best solution in terms of the current environment.

2. Evolved the editorial workflow models of current journal

Although many methods and technologies can be applied to design a journal management system, with respect to our methodology, we focus on the editorial workflow model of the journal. We found that there were exercises and research to apply the workflow management concept in developing journal management systems. We summarized the concepts and tasks of journal management system based on concrete needs from routine journal management work using the asking the right question method from the EBD methodology. These results are further used to evolve the editorial workflow of current journal and to analyze the adaption of general editorial workflow concepts to our journal. We improved the editorial workflow models of the journal and these business models work better in providing them to develop the updated journal system.

3. Developed the new journal editorial workflow management system

Our contributions to the development of the journal management system, the advanced redesigned system, can be described from two aspects. First of all, we clarified the user functions and redesigned workflow and architecture of the system. We also improved the system database based on concrete requirements from the analysis result. Second, many changes and improvements we implemented as both client and server side application. Those enhancements include new system functions, new user interface development, new database implementation and code implementation.

1.4 Thesis organization

This thesis is organized as follows:

Chapter 2: Related works and previous system. This chapter reviews the literature related to the present thesis research. Since the topic of my thesis is solving the journal editorial workflow management system design problem, the review will contains two parts: The first part is related to journal editorial workflow management system, and the second part is related to the redesigned object - the old journal management system. The objective of this chapter is to clarify the following questions:

- What is journal editorial workflow management system?

- How about the previous version journal system?

Chapter 3: This chapter will analyze the redesign problem using EBD methodology. Introduction and explanations of the EBD are given first. Then the entire analysis procedure is presented in this chapter.

Chapter 4: This chapter describes the general design process of the new journal management system. In this chapter, we design the new journal management system based the analysis result from the previous chapter.

Chapter 5: This chapter presents all the new capabilities the redesigned system implements.

Chapter 6: This chapter presents the results of the redesign.

Chapter 7: The conclusion and future work. This chapter is the summary of the thesis. It contains the following sections:

- The conclusion of this thesis

- Future work

2 Related works and the previous system

This chapter discusses the some related works about journal editorial workflow management system. Then we will introduce the previous version system of our thesis project.

2.1 Related works

Because of the maturation of both software and hardware aspect of internet technologies, more and more application needs to share information and provide user communications turn to online version and become a web-based software. Along with the development of web techniques, increasing number of new web application implementing technical innovation, obtained a large portion of online application developments[6]. Different server side programming languages are used, such as Python, Java, PHP, Perl, etc. And many web application framework has been developed, for example Django, ASP.NET, JavaEE (Servlets), etc. These frameworks can support the development of dynamic websites, web applications, web services and web resources. And there are improving in internet hardware installations which including high-speed, reliable steady networks and support for a large number of concurrent connections. Due to immense advances in internet technologies in recent times, web based applications have been adopted by software designers and users.

The majority of scientific journals are run via the Internet[7]. They need strong user communications and plump information sharing. Hence, most journal systems are web-based. A journal editorial workflow management system is a web application that provides online support for the management of a new journal. It facilitates the process of editorial workflow of a journal, including the submission of a new paper, the process of peer review, the paper proofreading and various editorials associated functions. It helps the editors, the authors and the reviewers in their respective tasks.

As far as academic activity is concerned, journal management imposes some specific requirements that make it different from other activities. Journal management must adapt to professional editorship, that is, to process papers in time according to a well-defined and efficient algorithm and following the standards of peer reviewing, copyediting, layout editing, proofreading, and publication. These standards are an editorial workflow of each journal. Since different journal has their own workflow, the journal management system for each journal can be unique. To deal with the editorial problems, commercial management system, in addition to being expensive, hesitated to give services to journals with suitable requirement. While some open source journal systems may be considered a big chance, they may require dramatically modified to fit for the journal. Development of journal management systems from scratch also needs to consider the editorial problem. Software engineers should familiar with the principles of editorial management software, or the editors are able to give efficient technical information on the subject to them.

We will discuss three existing journal systems: Open journal system[8, 9], Digital Publishing System[10] and ePublishing Toolkit[11]. From analysis of these systems, gaining background knowledge our thesis project can based on.

2.1.1 Open journal system

The Open Journal System (OJS) is a free and open source software for the management of peer reviewed academic journals. The OJS enjoys a relatively long history of development. It has been designed and developed by the Public Knowledge Project[12] in Canada. And well supported by a partnership among the University of British Columbia, the Simon Fraser University, the Ontario Council of University Libraries, and in the USA the University of Pittsburgh and California Digital Library, and the School of Education at Stanford University. OJS was released under the GNU General Public License that the most current released version is 2.4.6. And its development was ongoing with the support of a large deployment and an active developer and user community. As the national platform of scientific periodicals, it was embedded in the Ukraine scientific publishing infrastructure [13].

The OJS developed by an open source component which is PHP scripting language. One OJS supposed to be used as a single platform to manage a group of electronic journals. It was also compatible with support for multiple discrete journals within a single application. The OJS was a platform independent system that can run on diverse platforms like Windows and Linux while it on not a dependent web server that should run on a server either Apache or IIS. It had a cloud software configuration and can be set up online. The installation of the software was standard for content management systems. One benefit of the system was the clear, many comprehensive and best, multilevel documentation.

This open system adopted the MVC[14] structure (Model -View - Controller), which divided the data storage level, user interface level and control level into different levels of interaction. It was a modular architecture and is well documented. While this architecture apparent complexity, it provided fault flexibility, tolerance and performance of the open source system. Built on this architecture, users can enjoy not only the use of existing functions, but appears to be highly extensible via the creation of user's own classes and modules.

OJS supported a wide range of academic and scientific journal production and publication process options and business models, from initial submission to final archiving, and commercial subscriptions. All editorial processes were shaped by the editors of each journal. OJS allowed users to read and review manuscripts in PDF and HTML formats using special tools. It was also available for handling with metadata and bibliographic data. The OJS had multi language support that needs to install the standard package that provides various functions which include a number of libraries and extensions. OJS models the entire academic journal editorial workflow process, from user account application and manuscript submissions, through peer-review, editing, to publication, and archiving. To support those in all the life cycle process of a paper, the OJS provided a role based access management and multistage publishing process control which includes well thought out administrative and participating roles and default workflow.

2.1.2 Digital publishing system

The Digital Publishing System (DPubS) is another open source electronic publishing platform software system that was designed and developed for scholarly communication and online publishing of scientific journals. The DPubS was developed by Cornell University and Pennsylvania State University during 2004 to 2008. It originated from the Euclid Project, "an online publishing platform for math and statistical journals"[15], was developed and applied by the Cornell University Library on the basis of this system. Since 2008, without further updating of the project, almost ten existing projects at that moment are associated with both the developers and organizations to form the basis of the DPubS.

The objective of DPubS was to offer services and assist all processes of scholarly publishing. It supported several types of academic publishing include proceedings, monographs and journal. The DPubS had a modular architecture with a group of interconnected functions. The modular architecture made DPubS support different types of publications due to its highly extensibility and customization. Based on this architecture, the DPubS consists of a collection of services such as an editorial service, a subscription service, and a user interface service, etc. With appropriate configuration, there were two business models allowed by the system which are open access and payment version.

The DPubS had one major issue of the system's functionality because it was designed by the Cornell University. It was not conceived by the scientific and educational communities. This had an effect on the peculiarities of DPubS which makes it developmental to consider more about the problems of the safety of information resources. It was not critical to all electronic libraries because there is support for that part of publishing software of information objects.

2.1.3 E-publishing toolkit

The ePublishing Toolkit (ePubTK) is a package of tools as a publishing software that was developed and maintained by the Max Planck Society[16] . It is utilized to operate a family of online scientific publications and open access journals called Living Reviews. The ePubTK was written in the Python scripting language which is the same as our system so there is no final installation package for it. And its license allowed for free use for non-commercial purpose and its latest sources codes packages were available in an online repository for developers. Another benefit was it is not really a trivial task for the installation of such system.

Because ePubTK consists of a family of journals as an information space, it caused a main difference from the previous DPubS. It is that the ePubTK was designed for a special type of publication which mentioned in previous so called living reviews. There were some unique features as "invited" and "liveliness" that distinguish the system from other online publishing systems. Almost all of the functionality of the system were connected with journals because each journal is a container which made this software unique.

The ePubTK was actively developed by the team of Living reviews. It had a role based access control model and mainly suitable for special publications with multi stage publishing process management. One major issue for this publishing system was that it is not good for the more standard and more general workflows because its workflow designed solely for inviting reviews of the Living reviews publisher.

2.2 The old journal editorial workflow management system

In this section, the previous version of the journal editorial workflow management system will be introduced in detail. This system worked for the support of the online paper submission of the JIDPS owed to the lack of an online system. It was developed under the supervision of Dr. Yong Zeng of Concordia University. The system provided functions for journal participants to support managing the whole editorial workflow of the manuscript. It was built on the web technique which developed mainly under the Django framework.

Journal management means to organize the activities and the efforts of users in order to achieve goals and objectives using available resources efficiently and effectively[17]. For this kind of system, the goal is adopting a collaborative editorial model in which the journal's roles work together in order to review submitted manuscripts and reach a final publication state within specified dates from the time of submission. Users were considered as roles and managed by the system. The activities of users referred to actions that roles use the functions in the editorial system.

We had a general idea of the journal management system. For the rest part of this section, we will examine the system, and analyze its structure, users and functions.

2.2.1 Structure of the system

The main structure of the system was a Web-Based Three Tier Model[18]. The three tiers are: the presentation tier, the business logic tier, and the data tier. The primary benefit of this 3-tier architecture is to separate the business logic from both the presentation and data access layers. The system had been developed as a Django web application. The three-tier module was similar to the Model -View - Controller architecture will be considered and therefore presented a little further. MVC (Model-view-controller) architectural approach separated the data model from the business rules and the user interface.

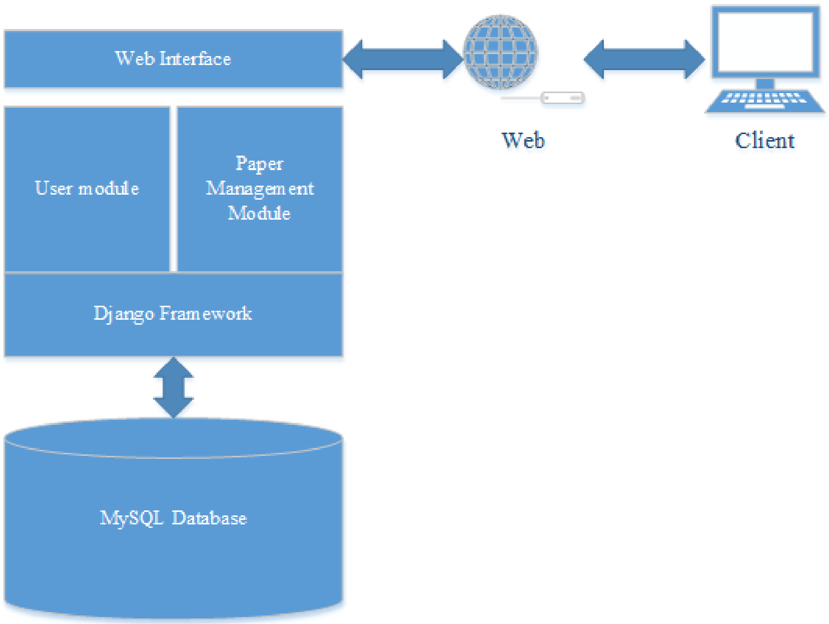


Figure 1 Structure of the old system

The data tier was used to store user and system data for the execution of the system. The data tier was often provided by a database management system (DBMS). In Django framework, this tire related to the model. As we have decided to use Django as our framework, we had a great flexibility in selecting the database. According to the database comparison, the most suitable database to choose will be MySQL. MySQL is an open source DBMS with the support for triggers and transactions. Both were heavily used to maintain database integrity and avoid possible race conditions.

The business logic tier can be regarded as the controller in MVC architecture. The middle business logic tier was the center of the system because it handles the "business logic", which was the core algorithms in the system. Controller represented the interface between the presentation layer and the database layer. It accepted and processed requests from the presentation tier, manipulate data stored in the data tier, and returned the result to the presentation tier.

In our system, there were two levels in the logic part. One was the Django framework and another was the application level. The Django framework was a software framework which function was to support the development of the application level. Usually database access, template frameworks and session management were some of the sections in which the web framework allows simultaneous work. The Django framework was written in Python. The system used Tomcat as the application server, with the assistance of Apache to provide static contents. It was implemented using Python and execute on a Tomcat server.

In the application level, there were two main modules: User module and paper management module. The paper management module was the main module and is responsible for editorial workflow management. The user module was responsible for user authentication and security. It included a custom-made authentication backend that allows users to authenticate. This function allowed the system to assume the user's identity when running processes on the server. Paper module contained all the functionality required to create and manage editorial workflows. This module was accessed through a log-in with the roles of the system being assigned certain functionalities. All users were required to use their unique username and password to login to the system.

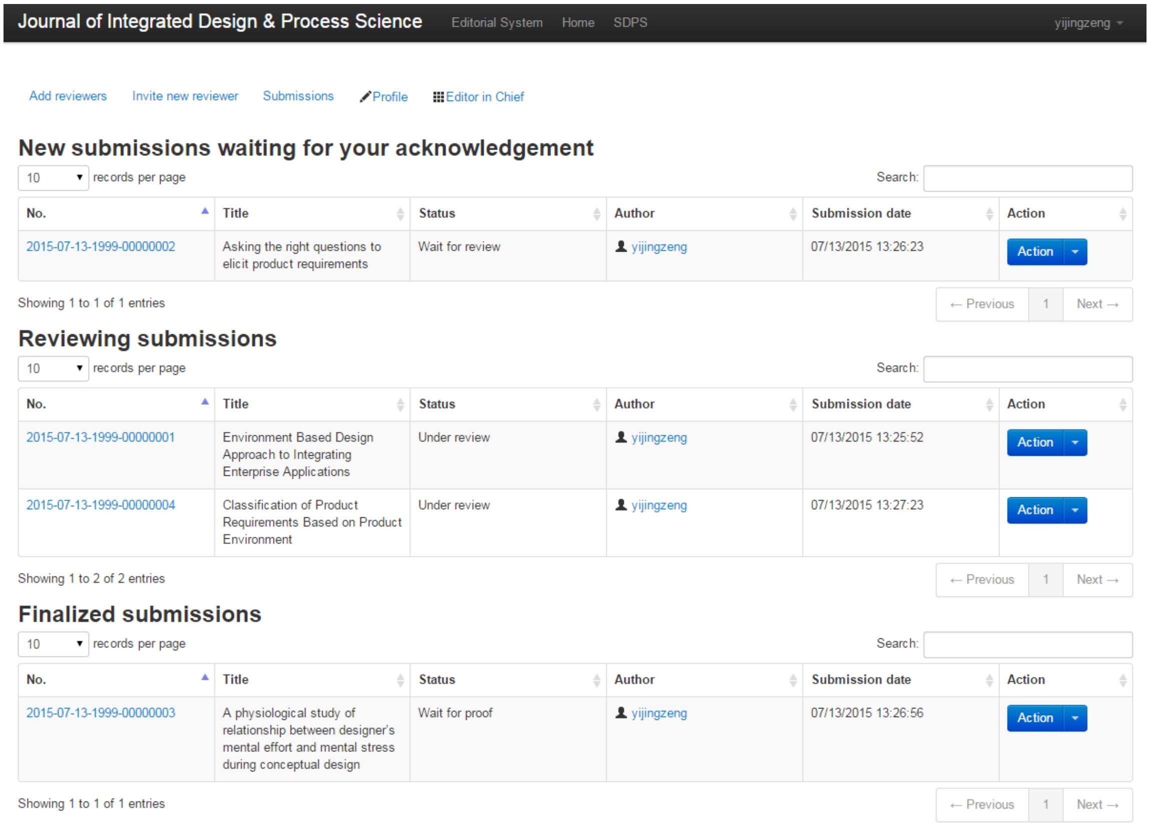
The presentation tier was always located at the client side and shown as web interfaces. In Django framework, the template was an instrument allows the system to define the visual aspect of the web content using HTML, CSS and JavaScript. In our project, the presentation tier was based on Django's template system, which generated the HTML that was displayed in the web browser in the user's computer. It was the web browser located at user's computer, which had an internet connection to access the system web site.

2.2.2 Users and functions of the system

Since this system had been designed as a role based access standards, unregistered users, who have not created an account with the web application, cannot access to the system. The system classified users into seven roles: Editor in Chief, Reviewer, Managing Editor, Proof Editor, Proof Reader, Author, and Administrator.

Editor in Chief: This class may be customized to include multiple persons. This kind of user has a strong say in all matters relating to editorial and general policy. The editor in chief response for controlling over the entire review process with the capability to view the papers acknowledged by them. Each new paper has an editor in chief who is responding for overseeing the review process and recommending the decision on the submitted paper. If the submitted paper clearly does not meet the requirements or the rules of the journal, it may be rejected by the editor in chief without the review.

The editor in chief can assign reviewers to paper and decide whether the paper will be rejected, accepted or sent for author revision. In order to determine reviewers for the paper, the editor can search through the list of reviewers in their field of interests in the profile, or their names. The editor in chief can also add new users as reviewers into the system or assign a system user the reviewer role.



New submissions waiting for your acknowledgement

10 records per page   Search:

No.	Title	Status	Author	Submission date	Action
2015-07-13-1999-00000002	Asking the right questions to elicit product requirements	Wait for review	yijingzeng	07/13/2015 13:26:23	Action

Showing 1 to 1 of 1 entries   ← Previous 1 Next →

Reviewing submissions

10 records per page   Search:

No.	Title	Status	Author	Submission date	Action
2015-07-13-1999-00000001	Environment Based Design Approach to Integrating Enterprise Applications	Under review	yijingzeng	07/13/2015 13:25:52	Action
2015-07-13-1999-00000004	Classification of Product Requirements Based on Product Environment	Under review	yijingzeng	07/13/2015 13:27:23	Action

Showing 1 to 2 of 2 entries   ← Previous 1 Next →

Finalized submissions

10 records per page   Search:

No.	Title	Status	Author	Submission date	Action
2015-07-13-1999-00000003	A physiological study of relationship between designer's mental effort and mental stress during conceptual design	Wait for proof	yijingzeng	07/13/2015 13:26:56	Action

Showing 1 to 1 of 1 entries   ← Previous 1 Next →

Figure 2 Submissions page of Editor in chief

Reviewer: Reviewer is in charge of reviewing and giving comments to an assigned paper. A reviewer will see a list of papers that have been assigned to him. After receiving a review task, the reviewer needs to decide to accept or decline the task. The reviewer has an option to reject the review. The scientific review consists of: decision (accept/reject/revise), originality of the work, the significance of the work, sufficient referencing, relevance to this journal, logical presentation and organization of the manuscript (outstanding, good, acceptable, average, and poor), comments for the editor, and comments to the author.

Managing Editor: Managing editor has a responsibility to manage the special issue and all submissions. A special issue is a collection of papers on a specific topic that system provides functions for creating, activating and closing it. The managing editor also has the capability to view all the papers in the system. From this function, the managing editor can revoke any paper submitted to the system which means the paper will delete from the system. The system provides the function to the managing editor to export all papers stored in the system as an excel file. Tables contain all papers submitted to our system as well as their status, submitted author and date submitted.

Proof Editor: Proof Editor is responsible for the entire proof process. In the current proof process, the proof editor only has the function to acknowledge the proof task and submit proofed version of paper.

Proof Reader: Proof Reader is a role managed by the system while the system didn’t provide any function for the role now.

Author: Author is the largest group for the system also a basic role for each user. All normal users are considered as authors. This class consists of people who are in the process of making a new submission to our journal issue or have already submitted a paper. Authors can use the system to submit papers and track the transition progress of the paper.

Administrator: Administrator is responsible for installing a new instance of the system and initializing the database, to set up the user account and system parameters. This class of role comprises of the person in charge of configuring the system settings and maintaining the security and integrity of the system. In our project, this role is from Django framework. The functions are derived from the Django core.

2.2.3 Editorial workflow of the system

Having defined the user classes, we proceed to the generalization of the submission itself. We defined that the system should concentrate on the submitted paper and function rules in the system define how the users may interact with the submission. Therefore, we presented a workflow diagram[19] showing the various states of the paper and its possible transitions. The editorial system is a workflow based submission processing system and the workflow of processing a new submission is as the Figure 3.

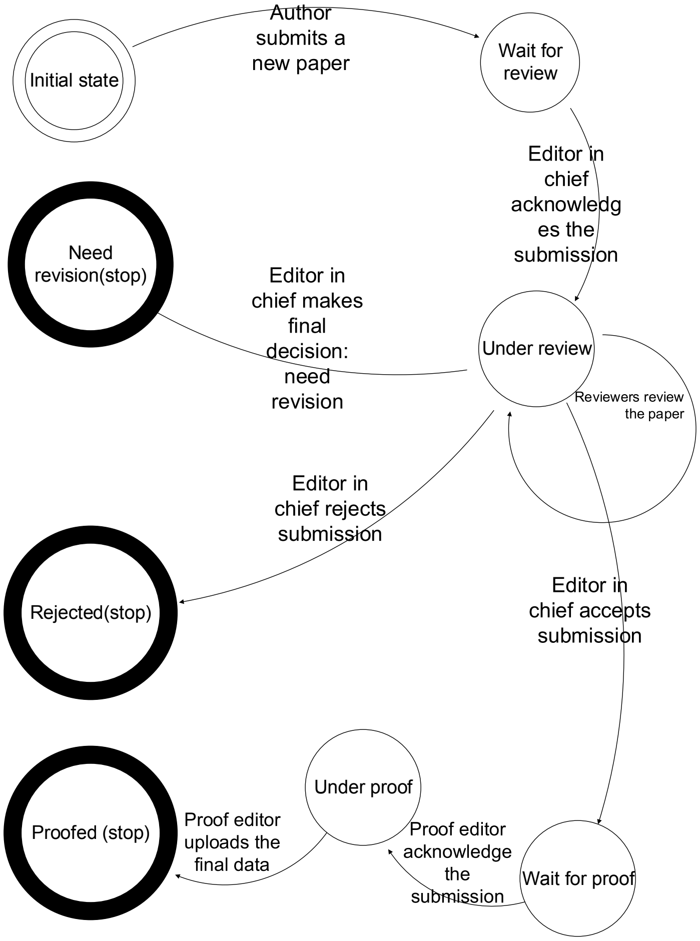

Initial state
Author submits a new paper
Wait for review
Editor in chief acknowledges the submission
Need revision(stop)
Editor in chief makes final decision: need revision
Under review
Reviewers review the paper
Editor in chief rejects submission
Rejected(stop)
Editor in chief accepts submission
Under proof
Proofed (stop)
Proof editor uploads the final data
Proof editor acknowledge the submission
Wait for proof

Figure 3 Workflow diagram of the old system

If an editor in chief decides that a submission needs revision. Then the author can login and submit a new version of submission which will be processed in the same workflow. The reviewers can be invited to review an article. The reviewers can accept or decline the invitation. If he accepts the invitation, he can then submit the review comments for the submission.

2.2.4 Other features of the system

The system supported the journal editorial management to manage users and papers in the entire workflow. Here introduce functions provided by the system in different respects.

User registration and login. This module allows a user 1. To log in to his existing account 2. To request a password-reset if he has forgotten his password 3. To register for a new account. User registration is a process for a new user to register with the system. There are several steps. First user needs to fulfill and submit a sign up form. The information on the sign up form is username, password and email address. After submitted, the user will receive an active email from the system in their registered email account. The user should click a link in that email to active their user accounts in the system. Finally, the user can use their username and password to login to the system. When they logged in, the default role to them is the author. If users forget their password, the system will ask the user to input an email address to send them an email to reset it.

Action-triggered email sending management. This mechanism is the functions of the system that send reminders and notices emails to users automatically. Email sending events is activated by the user's actions that the system response immediately and send an email message to the user. For example, when the author submitted a new paper, the system will send an email to notice editor in chief that a new submission is coming. The email can only send to user's registered email address. We're using the system to store contents of these auto-sending email message templates. While this mail system only responsible for sending emails, so it does not store these messages inside our system. They have to use their own Internet mailbox to check the mails from our system.

User management. User management are functions of the system provided for all users. These functions include: Submit and modify a personal profile, change roles, user account settings. The personal profile contains information as first name, last name, title, degree, phone number, fax number, street address, city, province or state, country, zip or postal code, position, affiliation, department, and research interests. The first name and last name are required fields. Because there are may be more than one role to a user, the user has the function to change roles. Roles list in the role selection list is roles the user assigned by the system. User account settings are functions for changing their password, the system language, the system time zone, and user's email address.

Log system. The system has a log system to record the important operations done by users, such as submitting papers, acknowledge review tasks, 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 an unusual sequence of operations.

Workflow management. We can notice from the workflow that the management based on the diversification of the paper’s state. The system provides functions for handling the state transformation of papers. It is implemented by a technician named Finite State Machine (FMS). It can add declarative state management for paper model.

3 Analysis of the redesign

As discussed in the previous chapter, the previous version of the journal management system is already a mature one with required functionalities. Nonetheless, it still has plenty of room for improvements with new requirements for its wider usage from other disciplines. This redesign, however, requires us to gain an understanding of journal management systems and their capabilities before we would feel comfortable to do this project. And that means we should clarify the goals of the redesign, old system main limitations, users, and user requirements that are all environment components of the product we are going to design. Thus we will use the EBD design methodology to assist in dealing with those tasks. This chapter presents the analysis process of the redesign problem.

3.1 Environment based design methodology

Environment Based Design[20, 21] (EBD) is a design methodology derived from the axiomatic theory of modeling of design[22] and found on the recursive logic of design[23]. The application of EBD can help designers meet their goal effectively. Different from traditional design methodologies, which are mainly based on the understanding that a generic design process comprises analysis, synthesis, and evaluation, EBD is based on the understanding of the environment of a design problem. EBD implies that design comes from the environment, serves the environment, and changes the environment, where design problems, design solutions, and design knowledge all originates with the environment.

What is the environment?

This environment refers to the production environment. The environment of a product is everything except the product itself theoretically[24]. A product originates with its environment, serves its environment, and changes its environment[20]. At a high level, for any product, there are three kinds of environment: natural, built, and human. The human environment refers to any human being who would interact with the virtual and real forms of a product. The natural environment is the natural universe except human beings. The built environment includes all of the products that have been built or created by human beings[24].

What is the design?

A design problem is a state of a design statement, which can be as short as one sentence and can be as long as hundreds of pages. A designer must first understand the design problem statement, which is often initially described using natural language[21].

What is the environment based design?

As illustrated in Figure 4, the EBD includes three main activities: environment analysis, conflict identification, and solution generation. These three activities work together progressively and simultaneously to generate and refine the design specifications and design solutions.

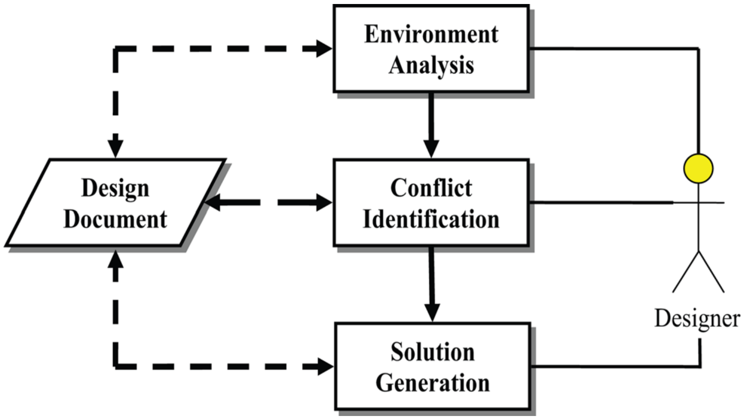


Figure 4 Environment-based design: process flow[21]

The redesign process can be totally guided by the EBD design methodology. From the above diagram, we can see that the process flow in this methodology is similar to redesign process contains analyzing existing systems, identifying problems and generating the solution of the problem.

3.2 Environment analysis

3.2.1 Overview

On Wikipedia[25], the definition of the noun “redesign” is “A plan for making changes to the structure and functions of an artifact, building or system so as to better serve the purpose of the original design, or to serve purposes different from those set forth in the original design”. In EBD, the activity of environment analysis aims to identify the product to be designed as well as explicit and implicit elements of product environment. In our thesis, this activity aims to gather the user requirements.

By following EBD, the whole design process starts from one sentence. We use the recursive object model (ROM) [26] to present the sentence. The ROM is a linguistic analysis tool and one of the key methods for environment analysis. Follow the Table 4 in the appendix, the initial ROM is shown as following:

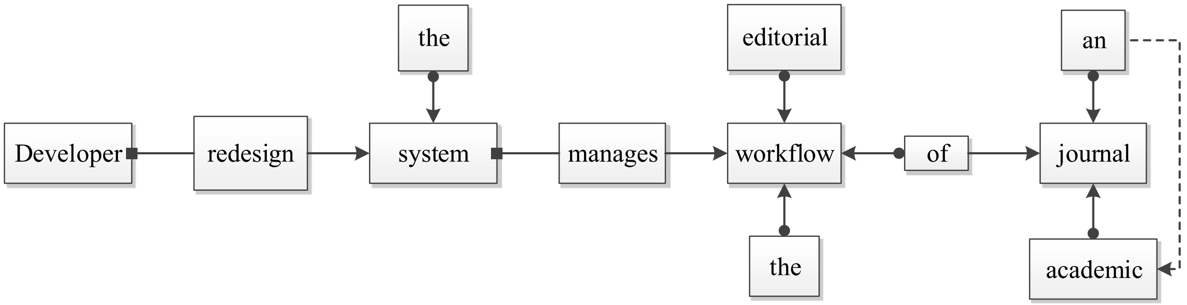


Figure 5 Initial ROM diagram

From the above diagram, we can clearly see that the entire redesign question can be separated into two parts. One is the redesign of the system and another is the journal management system. In EBD, we call these two parts interactions. The interaction defined as a relation in the ROM diagram from one object to another object and link with a verb. The two parts can transfer to two interactions: 1. Developer redesign the software. 2. The system manages the editorial workflow of an academic journal. Once they become interactions, there is a causal relation between them. We can analyze them and conclude that interaction 2 is the cause of interaction 1. It means that interaction 1 is dependent on interaction 2. In the environment analysis, we have the rule that interaction 2 should be analyzed first.

Now we start the environment analysis of interaction 2. We will use another essential method from EBD, which is question asking and answering. The question asking helps to determine the solution direction of the design problem. While a ROM diagram is generated, some questions should be asked to clarify every object in the ROM diagram. Based on the rules for question asking and question template list in the appendix, we generate a question list for interaction 2. Before generating the question, we exact the interaction from the initial ROM and code it, so that it is convenient for the question generation.

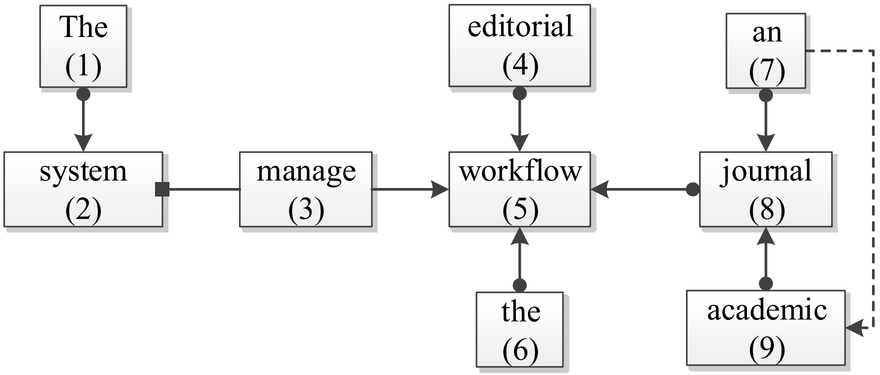


Figure 6 Coding ROM objects in the problem statement: "The system manages the editorial workflow of an academic journal."

Table 1 Questions of interaction 2

Object	Conditions	Questions
7&8&9	For an noun object A constraining an noun object N	What is an academic journal?
7&8&9&4&5&6	For an noun object A constraining an noun object N	What is the editorial workflow of an academic journal?
1&2	For a concrete, proper, or abstract noun object N without any constraint	What is the system?
3	For a verb V with its subject N1 and object N2	What do you mean by manage in the statement "system manage the editorial workflow"? How do/does system manage the editorial workflow?

In the following step, we will attempt to answer these design questions. The answers of these design questions are essential for identifying the product to be designed and the environment component of that product.

The question “What is an academic journal?” we can answer it according to the background of our thesis project. An academic journal consists of peer reviewed articles, published for an academic domain. The academic journal for our thesis project is one named Journal of Integrated Design & Process Science (JIDPS). The articles related to our thesis project are manuscripts. We only consider the editorial workflow of journal’s manuscript, which is also called submission or paper. For the third and last question, the system refers to the old journal editorial management system. The management be implemented by system users and functions. All the detailed information has been presented in chapter 2. Finally, there is one question about the editorial workflow left. The next section will analyze this problem totally. Because it can help us get the system requirements if we generalize the process of academic publishing abstracting specific and customized low-level processes that changes by the organization.

3.2.2 Journal editorial workflow analysis

Editorial workflow refers to activities that every paper submitted to the journal will undergo during the whole editorial process. It manages journal manuscripts in order to review submitted manuscripts and reach a final publication.

To better answer this question, we can consult guidelines for answering the designing related questions from EBD which is stated in the appendix. To answer the questions, the events and life cycle of a noun should be considered. Also, there is a question answering template in Table 7 for reference purpose. These methods from EBD will help us collect useful information sufficiently and necessarily available. First, we review the general journal editorial workflow to collect some general and basic concepts. After that, we will back to analyze the workflow of our journal.

3.2.2.1 General journal editorial workflow

In recent years, publishing of academic journal articles is undergoing significant changes as the significant change of it is transferred from paper form to electronic format. For a long time, the bulk of journal papers has mainly been through paper-based publishing as it is easy for authors to produce. There is no supporting technology that can help them review the submitted papers which is not convenient for editors and reviewers. The increasing cost and time in handling paper submissions and the widely use of the Internet during these years push the paper-based publishing goes to electronic. However, in order to analyze general journal editorial workflow, we need to consider the paper-based editorial workflow because the main processes involved in traditional and latter-day electronic academic publishing are almost the same. As mentioned by Campbell[27], the main processes can be described as editing, production, marketing, distribution, sale and promotion. Our thesis project mainly focuses on the two earlier stages: editing and production, since we only need online processing of submissions.

Journal editing process, which is mainly described as manuscript submission and peer-review process in the academic journal editorial workflow, contains the following common stages as indicated by Ciesielski[28]:

- Authors submit submissions to editors
- Peer-review among editors, reviewers, and authors
- Editors make decisions to submissions (reject, accept, and revise)
- Authors correct submissions and resubmit

The process of editing is usually managed and organized by journal editors, collaborated with authors, editors and reviewers. The signal of completeness is the article is accepted by the journal and waiting for the next step. A general peer-review process can be described as:

In the initial state of the workflow, the author submits a manuscript to academic journal editors. Editors will consider the paper and determine whether the manuscript is good enough to without peer review or needs to select external reviewers are not satisfy journal rules. The manuscript may be sent back to its author with a rejection letter or sent to reviewers. Reviewers handed back the manuscript to the editor with their recommendations or decisions and comments which varies from different peer-review models. Editors send the result back to the author which is either a request of revision or a rejection notification. The author corrects manuscript and resubmits to editors. Sometimes the same editor handles and sends revised manuscript back to external reviewers. Based on journal requirements and reviewers' opinion, the editor determines to accept or reject the manuscript.

The production process starts to happen since the submissions are finally accepted by the editors after the editing process. The production process will handle by production editors or publishers which covers two main stages: proofreading and publication.

Proofreading is performed by relevant editors and to check all about the presentation of a paper: deal with layouts, fonts, headings and others to ensure that the format of the article is inconsistent with the style of the journal. Related editors should also respond to check the grammar and spelling errors of the article and make sure the referencing is correct. The author should participate proofreading to review and correct proofs at the early stage of the production process. As Dale[29] indicated, automated assistance would be more likely to happen in this process because proofreading is an error-prone and time-consuming process. However, through the web observation, it is difficult to find information regarding the details of this process which seems inadequate. Before developing any automated assistance, we would better understand the production process well enough.

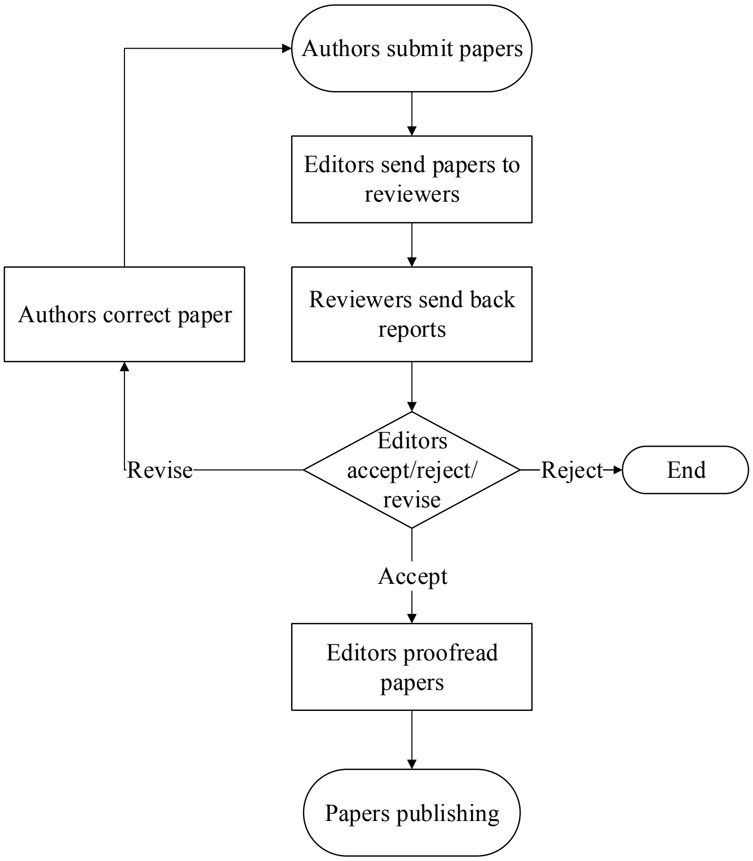


Figure 7 General journal editorial workflow

3.2.2.2 Specific journal editorial workflow

In order to verify the completeness of the extracted environment components and their relations, a roadmap was proposed as guidance for requirements modeling[30]. In this roadmap[31] (see Figure 35), requirements (structural or performance) are categorized by two criteria in terms of different partitions of product environment: One criterion classifies the product requirements by partitioning product environment in terms of product lifecycle (refer to Figure 34) and the other one classifies them by partitioning the product environment into eight levels (refer to Figure 34). The eight levels can be grouped into the natural, built, and human environments for better extraction of the environment components.

From the general editorial workflow, we can define events in a manuscript's life cycle. The entire editorial workflow for a manuscript often contains four events: submission, review, proofreading and publication. Not all the paper will undergo these four events. But these activities can be the typical events in the editorial workflow. We then consider the environment component of each event in the life cycle. Here, the answer will collect the abstract and general information which can be the guideline for answering detailed questions.

Generalized from Figure 35, product environment can be partitioned into 3 categories, which are natural, built and human environments. Combine these three categories to above four events, and consider the workflow definition from Ryan[32]: "A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information." We can get the following information.

The natural element for all events is the time. The built elements includes manuscripts and all related documents and information. All those things can be whether electronic or paper text or documents. In our thesis project, we are using software to manage manuscripts. So all the materials are electronic text or documents. Because of this decision, a computer and the Internet also become the elements of the built environment. And the most important built element is the system function. Functions derived from actions is done by a person to manipulate the manuscript which are different in each event. According to Figure 35, functions can be generally classified as basic functions and extended functions. For the human environment, there are different people with different roles to take different actions in this part.

Finally, we can organize these conclusions into a table as Table 2. It is a guideline we have extended from the EBD methodology by adding an in-depth analysis of current questions or work practices in order to gather JIDPS real workflow and evaluate whether the previous system supports those activities.

Table 2 Guideline questions to collect system requirements

Classification	Environment elements
Natural	Time?
Build	What basic functions in each event? What extended functions in each event? Which role take functions? When does the role take the function? What is the input material for each function? What is the output material of each function?
Human	What roles evolved in each event?

To answer above questions, the interviews which were held in a particular group of people or individuals who used the system in a distinct manner were to gather user requirements. After answering questions, we can collect enough information from the committee of the JIDPS. The detailed editorial workflow listed below to illustrate this characteristic we grouped them in clusters.

Submission process: Author submits the paper to Journal online. In JIDPS, there are two main types of paper: regular paper and special issue paper. The regular papers should go to the editor in chiefs and the special issue paper should go to handling editors separately. The basic function of the author is to submit his/her submission into the system database. While authors want extended function to review their manuscripts before final submission. And functions allow them to choose to submit it immediately or return to it later to add more information and submit later.

Review process: One editor in chief will choose to process the regular paper. This process is similar to that in the general editorial workflow. While for the special issue paper, it has different transitions. One handling editor will choose to process the special issue paper or pass it to other editors. The handling editor will read the paper and make first decision. The first decision includes:

pass the paper to regular type, pass the paper to this special issue's guest editor and reject. If the first decision is passing the paper to regular type, this paper will goes to all editor in chiefs. If the first decision is passed to guest editors, this paper will goes to all related guest editors. One guest editor will accept to process the paper. The guest editor can look for reviewers to review this paper or not. Finally, the guest editor should make a decision and comment to the paper. Then this decision and comment will be sent back to the handling editor. Finally the handling editor can make a decision and comment to the paper based on guest editor's comment. The editor in chief and handling editor's final decision for paper includes: Accept, Minor revision, Major revision and Reject. Decision and comment from them will send back to the author.

The main element of the review process is the time, since review reports and review processes demand a lot of time for final proof and publish the manuscript. For this purpose all participated editors want extended functions to reduce time-consuming at this process. It seems like paper assignment and tracking functions. Event logging retrieves list of appropriate/available editors and or reviewers and tracks those who choose or are assigned to particular articles. Making it easy to check on the status of reviews (completed or pending). Keeps log of actions performed (such as revised/resubmit requests). Automates the assignment of reviewers based on article categories.

Proofreading process: Once a paper is accepted by journal editors, the paper goes to proof editors for proofreading. One proof editor will choose to process the paper. The proof editor can invite proof readers to format/edit/proofread the paper or not. The proof readers will send their result back to the proof editor. Upon the receipt of the corrected paper, the proof editor should make the first version of proofed paper. The proof editor will send proof to the author for review. Author must send feedback comments with the copyright form to the proof editor.

Once proofs are received from the author, the proof editor will do the final editing and check. Then the proof editor will make the final version of proofed paper with its DOI number and send it to managing editor. The paper goes to the managing editor. If the managing editor accepts the final proofed version of paper, then it will go to the editor in chief. If the managing editor decides the proofed version of the paper needs modification, the managing editor should make a comment. The comment with the paper will goes back to the proof editor to revise. Here the same process for the editor in chief to handle the proofed paper. Once the editor in chief accepts the final proofed version of the paper, the paper will goes to publishers.

Publication process: The publication process, controlled by proof editors and publishers, then takes a paper through publishing phase. The publisher will set the paper to be pre-press posted. Once the paper is pre-press posted, it will go to the proof editor to add volume, issue, page number and year. The proof editor will add and send it to the publisher. The publisher will set it to publish.

3.3 Problem identification

3.3.1 Overview

Directly, we can see that the previous management system has not fulfilled functions to support the above mentioned analysis result of journal transactions. Besides these kinds of conflict, we should evaluate the old system finding defects. There are many methods to evaluate the defections of the software. We can simply conduct the inspection method to observe users and analyze their feedbacks. Fortunately, we are mainly responsible for the transitions of JIDPS and we host a number of businesses every year while the result is not quite positive. Taking into account the queries and comments received from old system users, a most common problem faced by some users is the inconvenience and difficulty to use some functions provided by the system to perform some tasks[33]. While we are putting our effort on developing more advanced journal editorial workflow management system, we should define those questions which come from normal participants of journal transactions.

It is said that if developing and designing systems required customer awareness and clear information, it is a critical piece to design ways to acquire audience input in the best development workflow solutions[34]. As the EBD design methodology guide, the design process is a recursive model. In our redesign problem, the recursive model appears when we implemented a new function or modified the old system, the entire relationship of functions as well as interfaces is changed simultaneously. It guides us to consider the occasion to identify problems that must be not too early or too late.

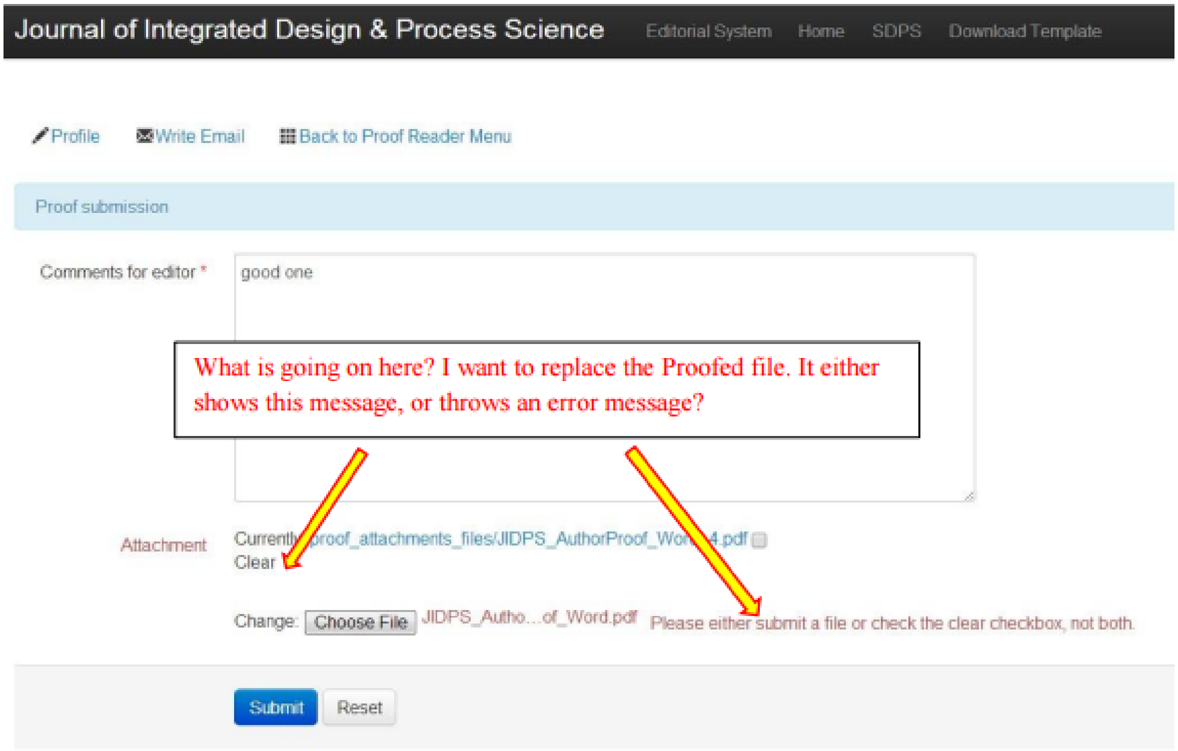
In order to identify potential problems the user may face, we conducted user tests and interviews[35]. All interviews are informal that can be applied often to gather various aspects of information for the user requirements. The interview questions are asked to provoke thoughts and discussion about the system as well as ideas for improvements. Those questions are generated by using the method of asking the right question and which process is similar to that presented in the previous section. Interview answers from customers help provide requests for improvement and thus aids in managing change and raising the importance of requirements. The interview is an effective method for generating ideas for innovations and enhancements by seeking direct feedback from participants and learning how they use the system. The results of the interviews can often use as indicators of usability problems.

User test is another method we conducted for identifying the problem and it is a good method to evaluate the accuracy and completeness of the testing system and find bugs of system functionality. According to the analysis of EBD methodology, we organized two formal user test activities when the system has major changes. Here we introduce one internal test we did before the system release, which we called a pre-release version test. We have six participants from our Design lab and some of them are actual staff of the journal. Because the limitation of testers, we should assign more than one role to each person to hold for a particular group of people who test the system in a distinct manner as shown in Table 3.

Table 3 User test participants

No	Participants role	Number of participants
1	Editor in chief	3
2	Handling editor	2
3	Guest Editor	3
4	Proof Editor	2
5	Proof reader	4
6	Publisher	1
7	Managing Editor	1
8	Author	6
9	Reviewer	6

Prior to the test we established an online testing system with an accessible website address. In order to ensure all stakeholders understand the tasks, context and goals of the system test, we prepared a test guideline document contains the role profile and sent to user help to improve the efficiency of testing. During the user test, simulated users are invited to perform a series of tasks working together to complete a paper's editorial events from submission to publication while being observed. For example, tasks of author role include: register for the system, login to the system, submitting a new paper to the system, observing the status change of the paper, check notification email sent from the system and so on. Observations and tests take place in our lab at the university with a standard university network. Because there is no software to monitor the test process, we ask all participants to write down their opinions about the existing functions and new functions they want as well as any questions for improvements and we will collect those feedbacks later. After the test, we ask testers to evaluate the current system, e.g. ask the user to explain what they want to do with the system, how the system operation does not work very well for them and how they would like that to be changed. Through the user test, we gathered many useful problems of the system and it is shown to a user feedback in Figure 8.



Comments for editor * good one

What is going on here? I want to replace the Proofed file. It either shows this message, or throws an error message?

Attachment Currently: proof_attachments_files/JIDPS_AuthorProof_Wor 4.pdf □ Clear

Change: JIDPS_Autho...of_Word.pdf Please either submit a file or check the clear checkbox, not both.

Figure 8 User test feedback from tester

3.3.2 Summary of problems

From chapter 2, we could conclude that system already has numerous functions as a journal editorial workflow management system while it still has much room for improvements with new requirements from the JIDPS. We can categorize the negative feedbacks for the old system into two major categories: drawbacks and limitations. The first category indicates that the previous system lacks functions to provide services for users that they can't get assistance from the system to finish their task. It also indicates that this kind of problem has the high priority of user requirements. The second category relates to the efficiency issues that mainly the problem existed in the old system. It shows the opportunity to improve, automate and combine a number of the old functions and provide the interface can be easily understood for some users to access the required information. In general, the solution for this kind of problem is to provide similar functions in better ways.

3.3.2.1 Drawbacks

Drawbacks of the system always exist in the entire redesign cycle. For example, compared the analysis result of the workflow and the old system, it shows obvious that the old system lacks several important phases which are the main drawbacks of it. Here we list several main drawbacks as examples.

Lack of proofreading and publication process management. As the analysis part said, there is only a minimal proofreading function of the proof editor in the system and the role proof reader, there is not any function provided for them. The system uses the Internet in a lesser degree as in that it merely implements submission and review of papers online but processes the proofreading and publication offline. For proofreading management, the system lacks both the proofreading management and functions. As the final stage of the editorial workflow for a paper, the publication phase is also important, but the system lacks the mechanism for publishing. The lack of them largely affects the completeness of editorial workflow management of the system, which is an essential drawback.

Lack of review process management of special issue paper. According to the editorial workflow from the existing journal, the review process of special issue paper should be handled in a different way while the old system does not support this difference. Editor in chief manages all paper review in the same way that raises the low efficiency problem of the system. Lacking of certain review management mechanism will reduce the quality of the journal paper and may reduce the efficiency of the editors of the journal.

Lack of auxiliary system management. Because of the use of the Django framework, it limits the number of the administrator and the authority of admin is too large to manage only small parts of the system. As a result of the interview, we need an auxiliary role with similar functions to the admin but handle the system user roles. The duty of this role is similar to the administrator while the Django core does not provide these functions.

Lack of functions in some editorial events. In each editorial process, the functions are not adequate for all roles to manage paper's editorial workflow. For example, in the review phase, there is no reminder function for the editor in chief to remind reviewers about the review task. The entire editorial process is operated by hand without any automation mechanism. Lacking of some necessary functions may affect the utility of the system users and further may affect the quality of work.

3.3.2.2 Limitations

The same as the drawbacks, limitation problem will not finish being discovered. The old system runs stable for more than one year and many limitation problem with it. According to observations, several limitations of the old system have shown up as given below:

Bad interfaces for roles. The system provides every role a fixed set of functions to get the task done and these functions are actions to handle the different status of papers. One problem is the system put papers in various status together into one page which is not a good organization that may mix the classification of papers and easy to take inappropriate actions for papers. However, when a new function will be added to the system or some functions are needed to be assigned to a specified user's role, it is not convenient for the developer taking this task. This problem may affect both the functionality and maintenance of the system.

Inconvenient role switching function. Our system provides a security mechanism to protect the system and different roles require 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. While, system users require the switching role function to change their current role to another role to carry out their works. Related function in the old system is a drop-down list on an individual webpage while the whole operation takes at least three steps to complete the role change. It is inconvenient for users who have multiple roles, such as author and reviewer.

3.4 Solution generation

Now we can go back to the interaction 1: Developer redesigns the system. From this interaction, it shows that a redesigned new journal editorial workflow management system is the product of design problem. Guide by EBD methodology, we still ask questions for this interaction. The questions are: What do you mean by redesign? How does the redesign?

The first question to be asked is the purpose and goal of the redesign[36] activity. Because the product is a new journal editorial workflow management system and we have an existing one. The development of a software system mainly is design and implementation. While the redesign activity can base on the old system, we can reuse its architecture and code. Hence the redesign can be an enhancement and improvement activity of the old system. The advantages of the redesign included: improved usefulness; improved efficiency and productivity; reduced learning time; improved usability; and increased acceptance among users. The goal is to improve both the efficiency and effectiveness of the publication system. There is an opportunity to improve the productivity of all the major stakeholders in the publishing system by taking greater advantage of Internet technologies. There is also an opportunity to advance effectiveness by making greater use of market mechanisms and structuring the knowledge embedded in journal articles.

This project seeks to redesign the system in order to finish the following main tasks: 1) A new proofreading and publication online management module; 2) Implement extended functions presented our current journal editorial workflow, and 3) A number of enhancements have to make to improve user experiences when using the system.

The second question asks how to redesign the system. In addition to these limitations found during the use of the old system, new user requirements are proposed to implement in the new system. When trying to implement those additional features in the previous version of the system, many obstacles are encountered, including the inadequacy of database structure, software architecture, and user interface. Because of this reason, we need to design the system first and then implement it. We will introduce those processes in the following chapters.

According to the above analysis result, first we need to generate the abstract description of our requirement. We can use the workflow diagram to present the improvements. From the workflow identification, we consider the materials, services and information we need. Based on the statement of the editorial process in the previous section, we will design all new workflow diagrams. New workflows include: special issue paper review process, proof process and publish process. The review process of regular paper and paper submit process workflow will be updated based on the previous workflow. All results will be present in next chapters.

The original system was not able to meet its needs as did not capture work practices in a way that was recognizable to the users. Compared with the JIDPS workflow to system limitations, the redesign process includes evolving the business model, designing workflow diagrams of new phases, redesigning the old system to solve limitations, and implementing all redesign tasks. The evolution of modeling is the process of redesigning the paper management module in the business logic tier. For the implementation, we can reuse the architecture of the old system because it runs stably over these technical, thus the new system can also run on it.

4 New journal editorial workflow management system design

In this chapter, we will redesign the old system mainly in the paper management module which is also the core of the new system. Some major new features will be presented. These new features and the underlying functions are needed based on the new requirements of a wider range of users and the practical experience of hosting the journal in the past year. It appears from an extensive analysis of the old system and editorial workflow of journal, although by no means exhaustive that each one offers detailed customizations specific to their requirements. Therefore, in an attempt to abstract away the details and simplify the process, we first study the classes of users who will access the system, consider here as a black box then moves on to the actual processes and show how these emerge from the different transition states of a particular submission within such a publishing system.

As there are already many functions existed in the old system, we mainly work on users' use case diagram and workflow analysis. Later, combined with the old system's architecture, we have a detailed system architecture diagram and system function structure. Finally, we discuss the database modification. Based on the above requirements analysis, the design of the new system is illustrated from the following three perspectives: function and workflow design, architecture design, and database design.

4.1 Profile of the system function requirements

In this section we provide a description of the development functions. One main redesign part is the role of system, as we analyzed, in the new system, the main user roles include Editor in chief, Handling Editor, Guest Editor, Proof Editor, Proof Reader, Publisher, Managing Editor, Journal Manager, Administrator, Author and Reviewer. Compared with users in the old system, the number of roles is increased and the responsibility of some role is changed. The general use case diagram of the system is shown in Figure 9. The detailed use cases can then be identified according to the elicited requirements presented in the previous chapter. Based on the analysis result, we can generate all of the use case diagrams and here we show the new some use case diagram relates to the redesign. We will explain new system functions grouped by phases and roles.

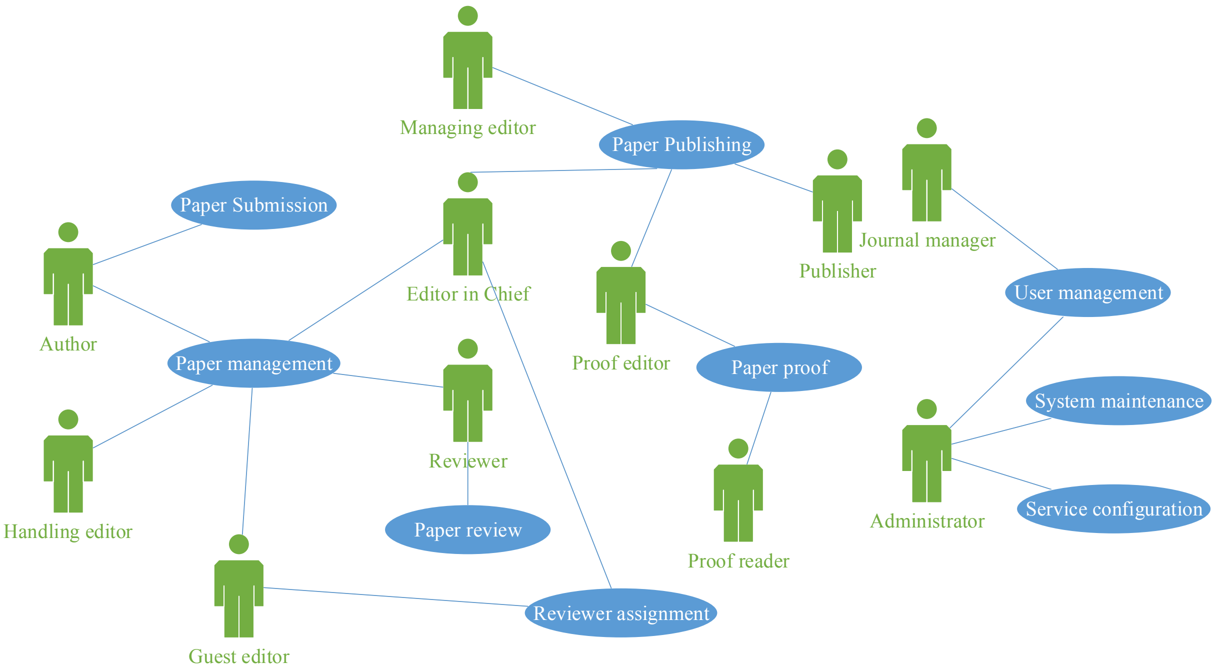


Figure 9 General use case diagram of the new system

Handling editor and Guest Editor manage the special issue paper. They are new roles which response for review phase of the special issue. Handling editor has the tasks: (1) Decide whether the new submitted paper can be a special issue paper or send back to regular paper or reject; (2) Send the acknowledged paper to guest editor; (3) Receive and review comments from the guest editor and reviewers and make a final decision on the paper. The guest editor has the tasks: (1) Accept the review task of the paper from handling editor; (2) Invite reviewers to review the paper; (3) Collect reviewer's comments and make their comments on paper and send all information back to the handling editor.

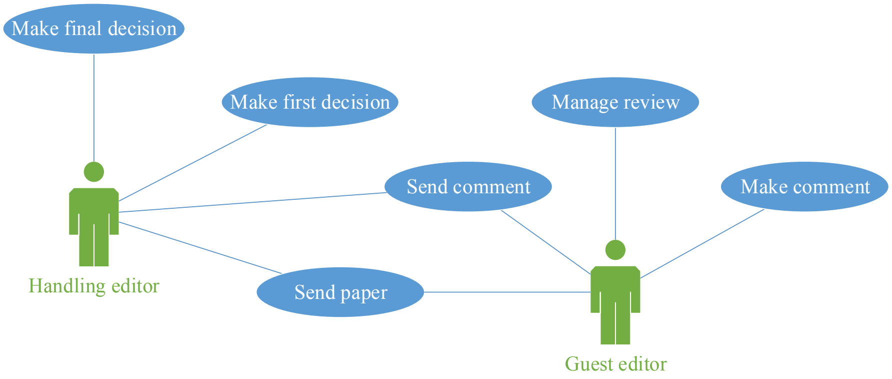


Figure 10 Use case diagram of handling editor and guest editor

Publisher's response for publication process. Publisher is a new role, adding response for publishing of the paper who cooperates with proof editor to manage the paper publishing. The publisher needs to view papers and selects paper to be published. The proof editor needs to submit the publication information of the paper.

Journal manager assists to manage the new system. In order to solve the auto-sending email management problem, a new role named journal manager has been added. Journal manager is a new role comprises of the person/s in charge of maintaining the integrity and security of the system. New functions for this role are managing system user roles and editing system email templates. He needs a friendly dashboard to monitor and modify the roles of system users. The predefined e-mail template content can be altered, if desired.

Extended paper submission functions. The new paper submission procedure requires the author to submit more information. Besides the basic information about paper, authors also need to enter the paper's corresponding authors' information. The system should support more the function that submits more kinds of attached files of the paper. Authors want to extend the function to review and modify their paper submission before it finally submit to the system.

New functions of the proofreading process. Proof editor has the function to invite proof reader doing proof read tasks. Then proof readers should have the function to submit their proof reading result. The paper's proofreading result will submit by the proof editor. And it must undergo the managing editor and editor in chief approval to ensure the correctness of the proof reading result.

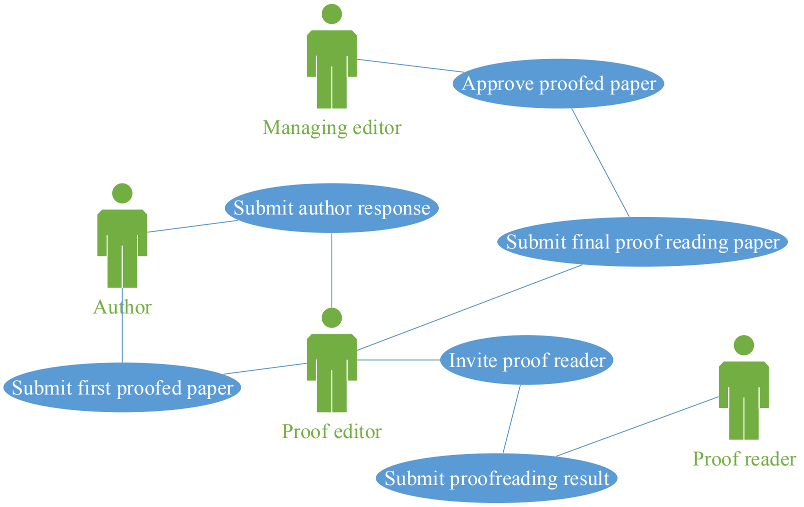


Figure 11 Use case diagram of proofreading process

New functions provide for editor in chief. In order to invite reviewers for the paper in a better way, the editor can modify the invitation letter sending to reviewers. And the editor in chief can also invite new users. For the review process, in case that the paper's editor is unavailable, the editor in chief can monitor review progress or send them reminders. The editor in chief creates a new special issue by selecting the handling editor and guest editors from a list of accepted and lectured system users. The editor in chief can also see a table showing the all of the papers in the system.

In order to ensure the system's stability and maintainability, and to improve the system functionality and overall performance, according to the software engineering development process and above statements, the system is strictly planned. From the analysis, the system contains online submission, online review, online proofreading, and online publication management sub-systems. With the system, editors, authors and reviewers can communicate between each other, working together, common to complete the papers' processing.

4.2 Updated and new workflow diagram

By requirement analysis, we know that the core business[37] of the system is paper that all the operations of the journal system are carried out around the paper. It must be noted that workflow and business are merely abstractions and any actual implementation would require considerable customization of both the states and the transitions. A transition may can be totally automatic or may require additional human intervention and it depends on the discretion of the developer or customer or the requirements of a specific implementation.

Above use case diagrams provide a static mode of design, but for editorial workflow, time is a natural element required to consider. Depend on the use case diagram, we need to generate their dynamic model which is a workflow diagram which is the abstract level of the redesign result. The more detailed redesign result is functioning for roles and interfaces that will be introduced in the next chapter. From those workflow diagrams we can see that the obvious change is the increasing number of states of paper and roles. In order to manage more transactions on paper, the new system should have more roles to assign to users and as the new event like proof and publish, there are should be new functions provided by the new journal management system. The new and updated workflow diagram that each state along with the relevant transitions and processes are discussed individually below.

1. Submission process

The initial state of an editorial workflow is always a submitting or equivalent state. Essentially, we need one state to represent the situation where the author has partially entered information about his/her manuscript and related inputs into the system database. The author may then choose to submit the submission immediately whereupon the state of it transfers to submit or return to it later to add more information or make modification and submit later.

As a redesign result shown in Figure 12, we design three new states for paper: Submitting unfinished, submitting and submitted. Papers with the state "submitting unfinished" represents the paper has submitted basic information and obtained a file number from the system. Papers with the submitting state refers to the paper that has uploaded all information waiting to do final submission. The last one is the stop state of the submission process represents the paper has been submitted to the system and sent to the editor in chief.

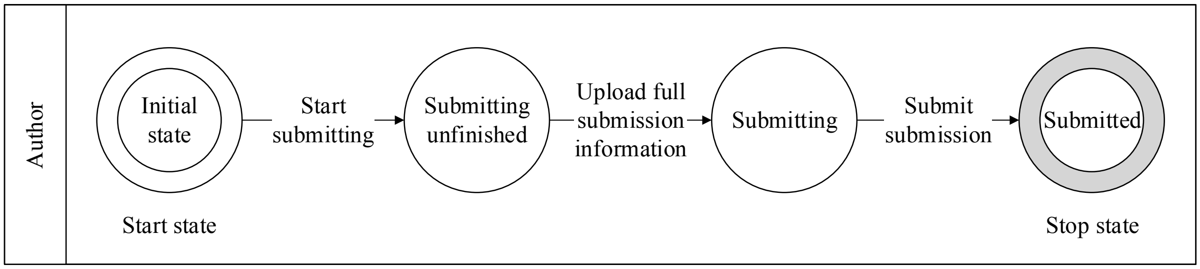


Figure 12 Updated submission process workflow

2. Review process of regular paper

When all final changes to a document has been made, the author submits the document and the submission now passes into the submitted state, as shown in the state transition diagram in above figure. At the review stage this regular submission becomes accessible to the editor in chief for further management. Recently, there has been a strong focus on auto management in order to remove editor in chief's workload. We add a new state named "required review completed" to the workflow of the old system to control the automated transfer of papers. Another new state is the state "Waiting for final decision" which designed for the function of saving decision and comments temporarily.

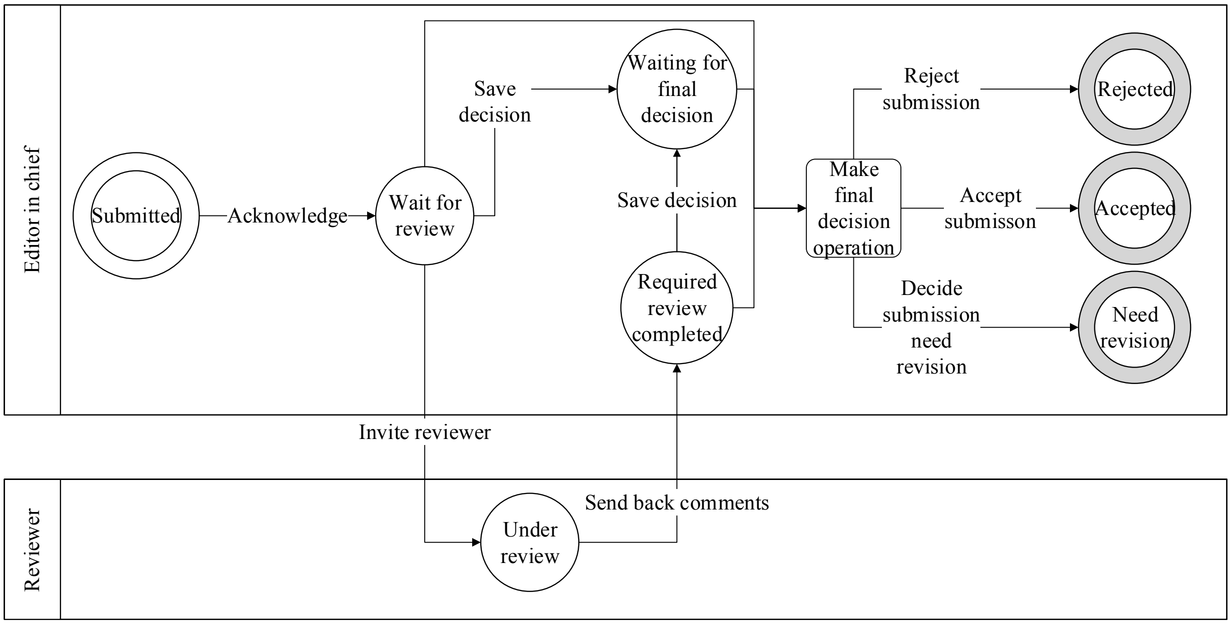


Figure 13 Updated review process workflow of regular paper

3. Review process of special issue paper

From the interview result, we know the review process of special issue paper should be different from above paper workflow diagram. As shown in the use case diagram that tasks of handling editor and guest editor are similar to the editor in chief, but this process needs more states for to manage transitions. We list the new states with explanation of details as follows.

1) Wait for first decision

The handling editor should view the manuscript first to check whether this paper fit for this special issue or not. So this state is designed for papers has been acknowledged by the handling editor, but waiting to make the initial decision. The decisions include: Pass to Guest Editor, Pass to the regular issue and Reject. The first decision means the handling editor accepts this paper to be a special issue one and allows the paper transfer to the next review state. The second result is different compared with the first one that the handling editor decides the paper do not meet the requirement of this special issue. Then this paper should send back to the submitted state and send to the editor in chief to handle.

2) Wait for guest editor acknowledge

As the fact, there are many guest editors' responses for one special issue at the same time. All of them will receive the notification email of new paper after handling editor made the decision "Pass to Guest Editor". This state is similar to other acknowledgement functions that paper under this state is waiting for one guest editor to accept the review task. The guest editor who acknowledges this paper first will response it and the system will record this result.

3) Wait for guest editor decision

This state refers to papers wait for the guest editor to make a decision. Although the guest editors can make the decision for a paper, their decision and comments will not stop the review process of special issue paper. Only the handling editor has the right to make the final decision on the paper. Comments from guest editor are provided to assist the handling editor finish the review.

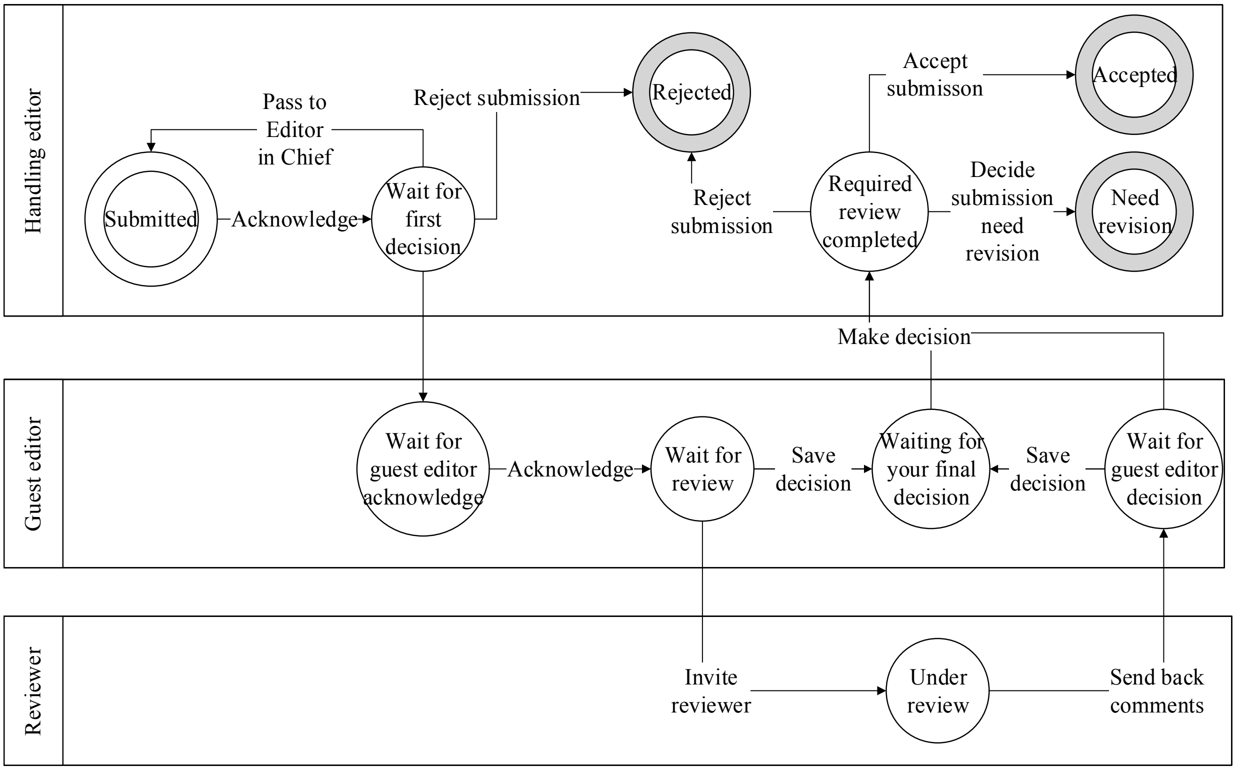


Figure 14 New review process workflow of special issue paper

4. Proofreading process

The submissions with state "Accepted" now undergo the proofreading process. Based on the proofreading workflow descripted in chapter 3, after several rounds of question asking and answer activities, we summary the transition diagram as shown in Figure 15. The whole process can be divided into three steps. The first main step is the proof editor invites proof readers and the next one is the proof editor asks the author taking part into the proofreading. Finally, the proofreading result should be approved by the managing editor and editor in chief respectively.

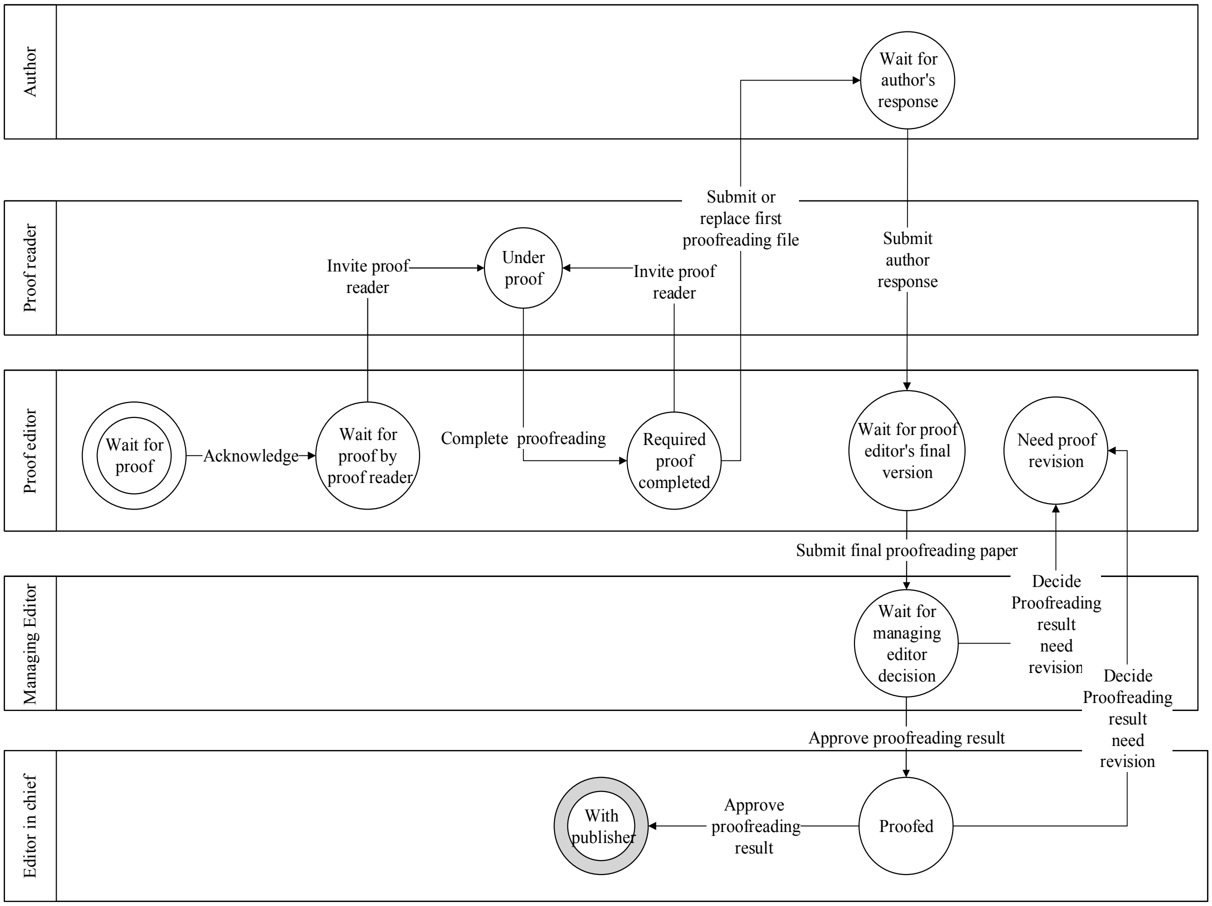

Author
Wait for author's response
Proof reader
Submit or replace first proofreading file
Submit author response
Invite proof reader
Under proof
Invite proof reader
Proof editor
Wait for proof
Acknowledge
Wait for proof by proof reader
Complete proofreading
Required proof completed
Wait for proof editor's final version
Need proof revision
Submit final proofreading paper
Managing Editor
Wait for managing editor decision
Decide Proofreading result need revision
Decide Proofreading result need revision
Approve proofreading result
Editor in chief
With publisher
Approve proofreading result
Proofed

Figure 15 New proofreading process workflow

5. Publication process

This stage is the last process of the entire journal editorial workflow which starts with the state "With publisher". Papers under this state are waiting for the publisher to select to be prepressed. Once the publisher selected a paper, he can change the paper's state to pre-press posted. This status is the last state before the publication and papers are waiting for the proof editor to submit publication information. Papers with state "Publishable" are ready to be publishing completely. Finally, the publisher can publish any paper by changing the state of the paper. The published state is the last one of paper which means the paper goes through the whole editorial workflow of our journal and publish successfully.

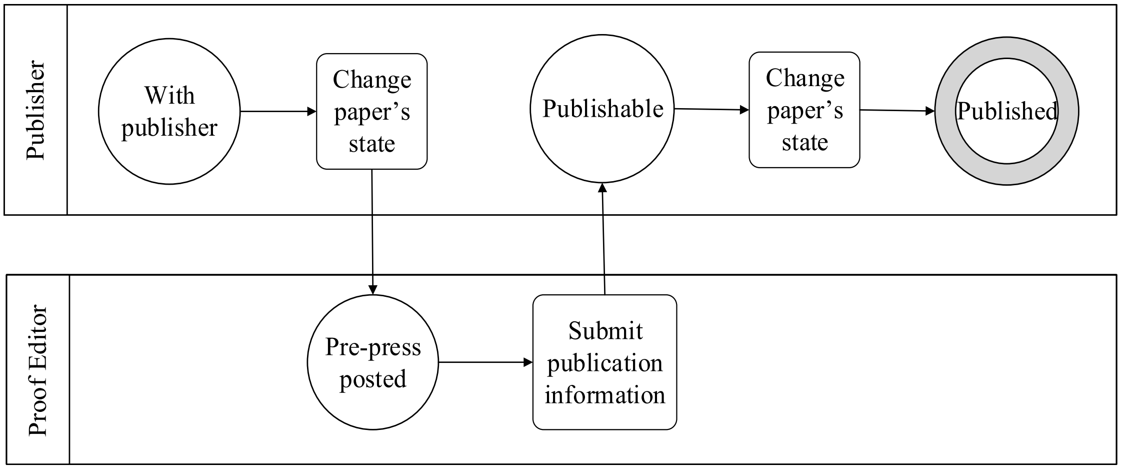


Figure 16 New publication process workflow

4.3 Updated system architecture

As we mentioned before, the architecture of the old system is a three-tier web service structure which is stable and we will keep most of it. While the new system becomes more complicated, we need to make some modification to the architecture to meet the requirements of generation, customization, operation and deployment. The architecture of the new system must be stable and useful in nature to satisfy the distinguished users' expectations. It is implemented by 4-layer architecture, which includes Presentation Layer, Logic Layer, Web Service Layer and Data Layer. The system architecture is shown in Figure 17 and functions of each layer are described as follows:

Presentation layer: This layer provides the interactive interface to the users. System interfaces offered by Django framework are template-based. It also can support diverse webpage definition languages, for example CSS, JavaScript and HTML, to develop user interfaces. There are multiple interface template generation methods in Django can be selected by system developers to make preferred interface templates which will be consistent with the style of the journal. The developer can also upload logos and images for to decorate the interfaces. The presentation layer is designed completely separate from the business logic for flexibility. Thus, it will have no impact on the underlying data and business logic layer either updating or even completely rebuild the presentation layer.

Logic layer: This layer is in charge of general business logic management and judgment. This layer is responsible for analyzing all the requests from the presentation layer, and then transfer them to executable models and assign them to different application server based on the system rules and load balance policy. It consists of user account status, user's authority, submission status, and service status, etc.

Web Service Layer: This layer is composed of two parts of services. The first part consists of application services including different functionalities which are necessary for editors, authors, and reviewers, such as online manuscript submission, manuscript revision, notification service and online submission. The other part consists of the system services which include basic operations such as user management, email template configuration, administration, monitoring to ensure the system operated smoothly and normally.

Data layer: This layer consists of various constant database servers physically. In our system, it has two types of data. The first type is the structured data that support the entire business, for example author information, journal information, paper information, reviewing and proofreading process information and service information in the system. Those kind of information is stored in the database. The second type is the unstructured data like submissions in the form of Word, PDF, ZIP and various other required documents are stored in the file system.

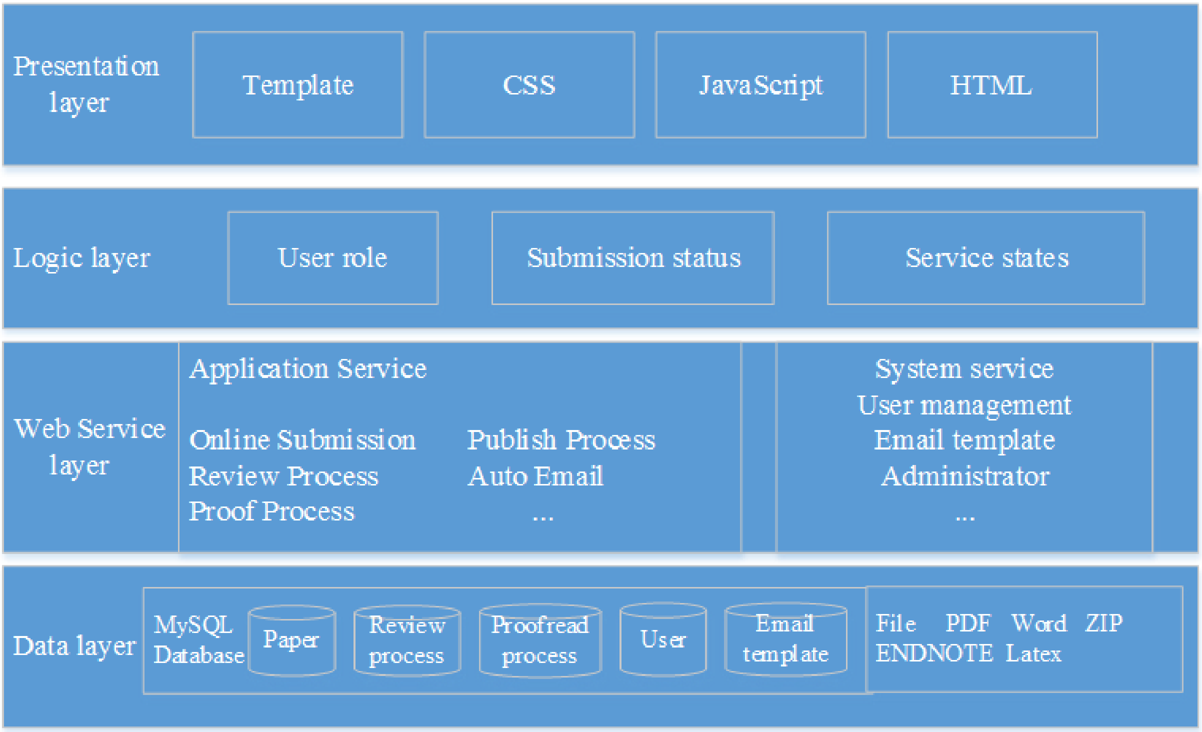


Figure 17 New system architecture

4.4 Updated system database

In this section, we will illustrate the updated database structure of the paper management module. All redesign mainly for this module, database for other parts can keep the same as in the old system. Our system uses a single database implemented with MySQL on each instance of its execution to store the system and user data. To meet the requirement of the database structure to support existing functionalities and proposed new functions, based on the user requirements and the design results, we summarized the added data fields in database tables as shown in Figure 18.

There are mainly four updates of tables:

Submission table: The submission table is responsible for storing system paper information. The papers are the core object in a journal. In this table, we delete some fields which do not require and add many new fields according to design results to support system functionalities. For example, the variable at the end of the submission table is utilized to store the publication information includes: volume number, issue number, page number and year of a journal on paper.

Author table: From the user requirement, it shows that the previous database of the submission is lack of information. As the analysis result, information missed of the manuscript includes: all authors' basic information (first name, middle name, last name, academic degrees, affiliation, and email address) and more kinds of attaching files. For relationship involving two entity sets, the author object and the paper object, their relationship is many - to - many. An author may correlate to more than one paper and paper can contain numerous authors. The database design for the author can be a class instead of fields which showed in Figure 18.

Special issue's table: This table is used to store the required information of special issue managed by the system which is not a new table in the database. While the mechanism of how to manage it has been changed, in this new table, we add two roles fields: one for handling editor and another is for guest editor. The system will store a handling editor and a list of guest editors who will respond to this special issue.

Proofreading process table: It is a newly designed table used to store the information about the process for proofreading. This table, similar to the review process table, defines proofreading required fields, which can be used to support the entire workflow. Main proofreading operations include invites proof reader, and submitting proofreading result.

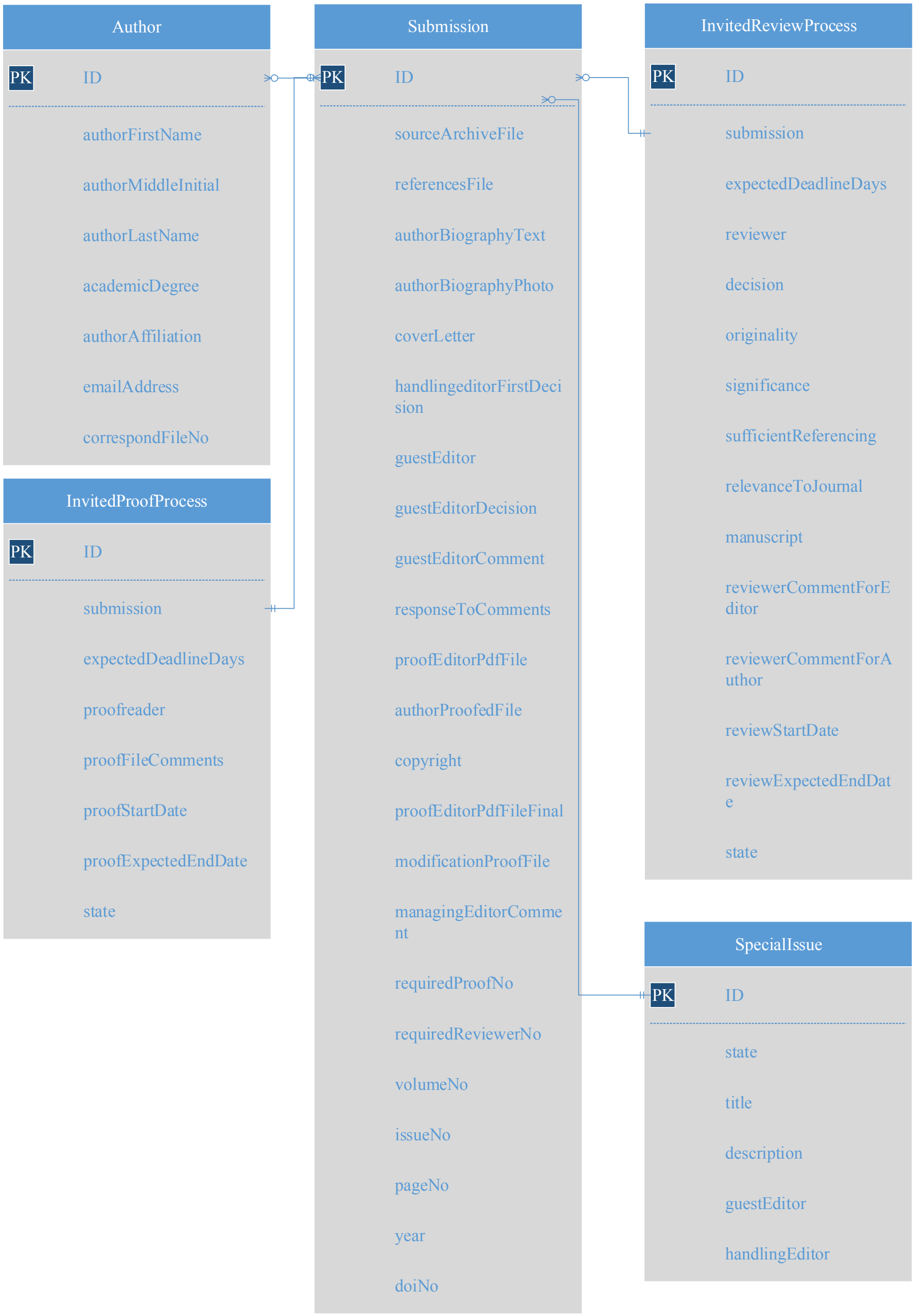


Figure 18 Updated database structure of paper management module

5 Improvements in new journal editorial workflow management system

Previous chapters present the analysis and design of the new system that several different functional module designs within a journal system architecture are defined. With the updated database structure, application architecture, and the framework of new workflow, we have implemented the new journal editorial workflow management system. We not only retain the functionalities of the old system to manage the entire process of the journal, but also add new functions and features. In this chapter, we provide a description of the developed functionalities and we will present the most improvement results including functions and interfaces.

5.1 New proofreading and publication process management

In chapter 4, we designed the new workflow representing the proofreading phase. Compared with the old editorial system, the new proofreading business flow contains more functions to process more information. The old system provides the necessary roles and functionalities to manage a simple proofreading process. Our improvements will build on the foundation of the old system and following sections will introduce all these enhancements.

5.1.1 First proofreading stage

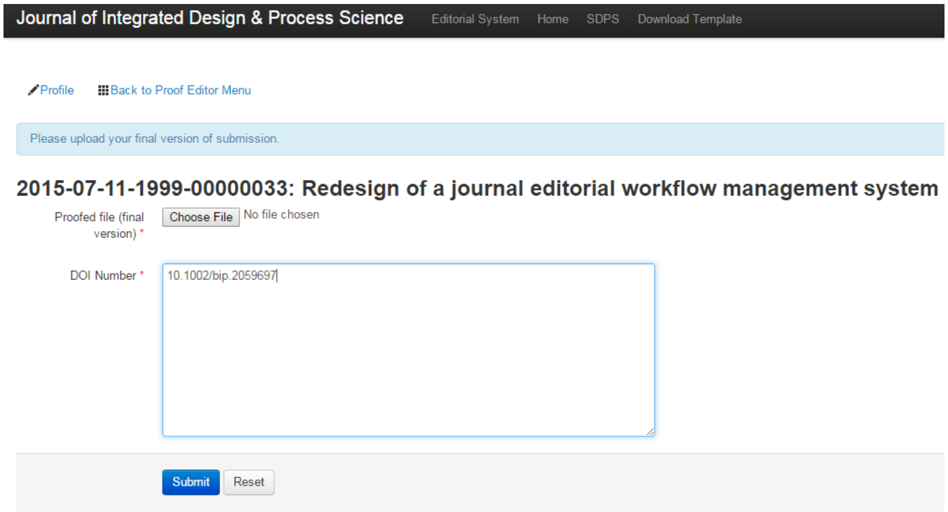
From the workflow diagram, we note that at the first proof reading stage the proof editor can either submit the first version of proofreading papers directly or invite a proof reader to do the proof reading task. It is a new function that invites the proof reader to do the proofreading work for a paper because the users with proof editor role have no functions in the old system. The new system is required to provide those functions for them to accomplish proofreading tasks. The page for the proof editor inviting the proof reader similar to Figure 31. This function is similar to the function that the editor in chief invites reviewers so it can be implemented by copying and modifying the code from the review process. Through this function, the proof editor can select any proof reader and set the proofreading task deadline. The default number of finishing the proofreading task is seven, which be different from that of the review. There is a link “Add proof reader” in the page which is a similar function as “add a new reviewer” that can invite a new user to be a proof reader in our system.

The proofreading cycle is manually transcribed by a proof reader onto a clean version of the proof. The new system only provides functions for the proof reader to review papers that they need to proofread and to submit the proofreading paper. These are similar functions for the reviewers, but proof readers can't refuse the proofread task. The proofreading task for proof readers can be finished once they submit a proofreading paper file to the system. This proofreading paper will sent to the proof editors to assist them completing their first proofreading task. After the proof editor submitted the first version of proofreading papers, the first proofreading stage of the paper is finished.

5.1.2 Final proofreading stage

Through the first proofreading stage process, the first version of proofreading paper has been generated by the proof editor. Both proof editor and proof reader may ask some questions about proof in the paper. So, in the second proofreading stage, the first version of proofreading paper will be forwarded to the author with those proofreading questions. The author should proofread his paper and answer questions, then returns all information back to the proof editor. This is the final opportunity for the author to check pages for any errors within the manuscript. System will send an email to the author that he received a paper required be proofread. This paper will appear in a new folder named "Submissions waiting for your response of proof" under the author's main menu. It contains all the paper that is waiting for the author to submit response files. There are two kinds of files required to submit. One is the author's response includes proofreading result from the author and question answers for the first version of proofreading paper. Another is the copyright form which is an agreement form for publishing. Meanwhile the proof editor waiting for the author's response until the system sent a notification email to him after author submits required files.

There are two functions for the proof editor at this stage. The first function is to correct the first version of proofreading paper and submit a replaced file to the system. It is a supplementary mechanism of the first proofreading stage and produce a full cover correction cycle to the proof editor to modify previous work. Another function is a similar function as in the first stage that submit final version of proofreading paper shows as Figure 19. Before this submission, the proof editor must carefully read the author's response to complete the final proofreading admission as well as finish the typesetting for the paper. The proof editor should submit the final version of proofreading paper with a digital object identifier (DOI) number, a character string used to uniquely identify a digital object, with the system. After this step, the second proofreading stage is finished.


Journal of Integrated Design & Process Science
Editorial System
Home
SDPS
Download Template
Profile
Back to Proof Editor Menu
Choose File
Submit
Reset

Please upload your final version of submission.

2015-07-11-1999-00000033: Redesign of a journal editorial workflow management system

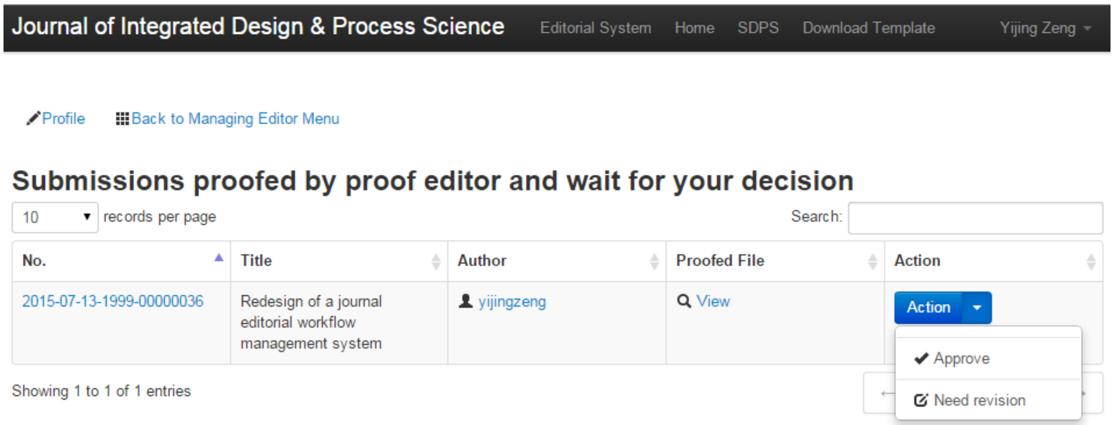
Proofed file (final version) * No file chosen

DOI Number * 10.1002/bip.2059697

Figure 19 Proof editor submits final version of proofreading paper

5.1.3 Approval of final proofreading paper stage

At this stage, the main event is sending the final version of proofreading a paper to both the managing editor and the editor in chief to approve. The approval order is the managing editor check first and then editor in chief. Functions developed for them for proofreading paper approval include viewing the final proofreading paper and make a decision that either it can be accepted or need revision. The workflow process of this stage is shown in Figure 16 and the main functions are shown in Figure 20. Since the main functions for both managing editor and editor in chief are the same, we list the page of managing editor here. The button "Approve" under the action title presents the approval function that the final proofreading paper being approved by managing editor or editor in chief once they click it. Another button "Need revision" which presents need revision decision is a link to a new webpage to write the comments to the proof editor for proofreading revision. Papers need to be revised will return to the previous stage with the comments assist the proof editor to do revision. Papers approved by the managing editor and the editor in chief mean the entire proofreading phase is finished. They will go to the publishing phase.



Submissions proofed by proof editor and wait for your decision

No.	Title	Author	Proofed File	Action
2015-07-13-1999-00000036	Redesign of a journal editorial workflow management system	yijingzeng	View	Action

Showing 1 to 1 of 1 entries

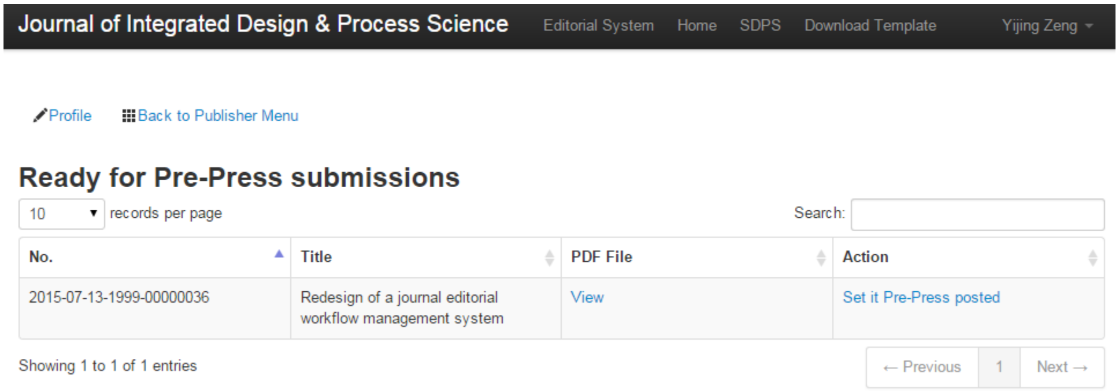
Figure 20 Managing editor approves final proofreading paper

5.1.4 New publication process

The old system does not contain any publication event that all functions in publication phase are newly developed. According to the workflow diagram of publication, main function the new system for the proof editor is to submit publication information and for the publisher is to update the state of the paper.

The duty of the proof editor in the publication phase is setting the publication information for the paper. We can note from the screen cut that the publication information includes: a definite file of final proofreading papers, volume number, issue number, page number and year of the paper. All this information is obtained from the actual data of the publishing journal. Also, this is the last chance for the proof editor to modify the paper. The proof editor completes all proofreading tasks of the paper after this step.

The new publisher role should be added to the system role list first to further function implementation. The main functions provided to the publisher are viewing the papers in the publication phase and updating their states at a certain point. As the Figure 21 show up, the "view" button is the function to view the paper listed under current process and the button "Set it Pre-Press posted" represents the function to update the state of the paper to pre-press posted sate from the previous one by clicking it. The event triggers the publisher to do this action is that the publisher selects the paper as it is ready to be published so the pre-press posted state means the paper is online for publishing without publication information. Another function of the publisher is similar to the first one, but with a different button named "Set it published". This action is done by the publisher updates the paper state as to be published which means the paper will be printed soon. When the paper goes to the published state, the paper goes through the entire editorial process.



Ready for Pre-Press submissions

10 records per page   Search:

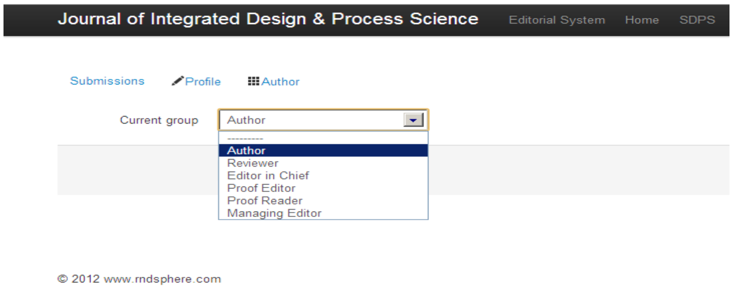
No.	Title	PDF File	Action
2015-07-13-1999-00000036	Redesign of a journal editorial workflow management system	View	Set it Pre-Press posted

Showing 1 to 1 of 1 entries

Figure 21 Publisher set paper state

5.2 Uniform user main page and main menu

The user interface has been prepared from scratch by the developers by hand. As we mentioned there are limitations of the old system that it has no main page for users and the menu system is not convenient for users. Also, some functions consume too much time or need too many steps for customer operation. The typical one is the function for user to switch roles. The function interface in old system shown as Figure 22.



Current group: Author

Figure 22 Switching role function in the old system

On one hand, the user should log in to the main webpage first, then jump to another webpage to switch the role which makes inconvenient actions so the improvement solution is merging the webpage of role switching into the main webpage. As a result, we can reduce one step operation when the users change their roles. As the result of less steps of operation, the process time is decreased simultaneously. On the other hand, the selection method for switching role in the old function is a drop-down list form. It requires two steps to finish the switching process by completing both the selection and submission operation. This is the reason that we will think of other items which can remove the number of steps when users click the button to change roles. We find that the link button in a webpage may meet our requirements. So link buttons with the name of system roles can be used to replace the drop-down list form because new functions can complete the task to achieve the same effect. And the link button only needs one operation of click instead of two steps of the old function.

As the result of improvement, the new user main page, as illustrated in Figure 23, allows authenticated users to access the flexible role switching function throughout the working process, no matter what role they are in. In addition to the role lists, we create main menus for users titled as “My tasks as Editor in Chief” and “My actions as Editor in Chief” in the body part of the page. The first menu used to organize all submissions by classifying them by different states and put into key folders. And we also account for the submission number and added it at the end of each folder. For example, the key folder named “Submission sent for proof” contains papers have been accepted by the editor in chief and sent to do proofreading. The number one shows in the bracket means there is only one paper under this status. Similar to the tasks menu, the second menu lists all additional system functions provided to this user.

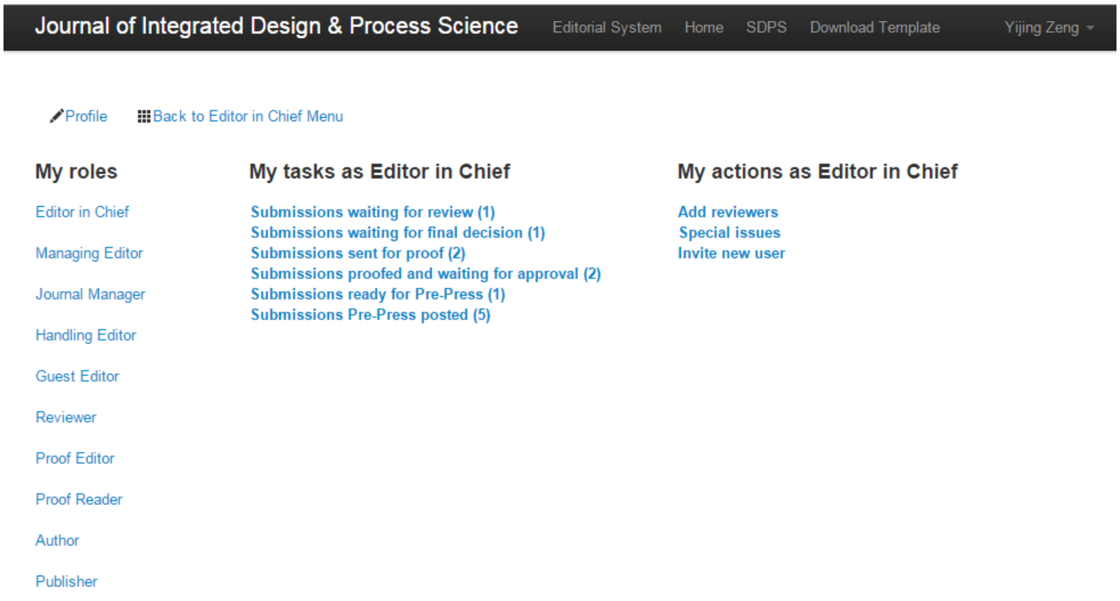
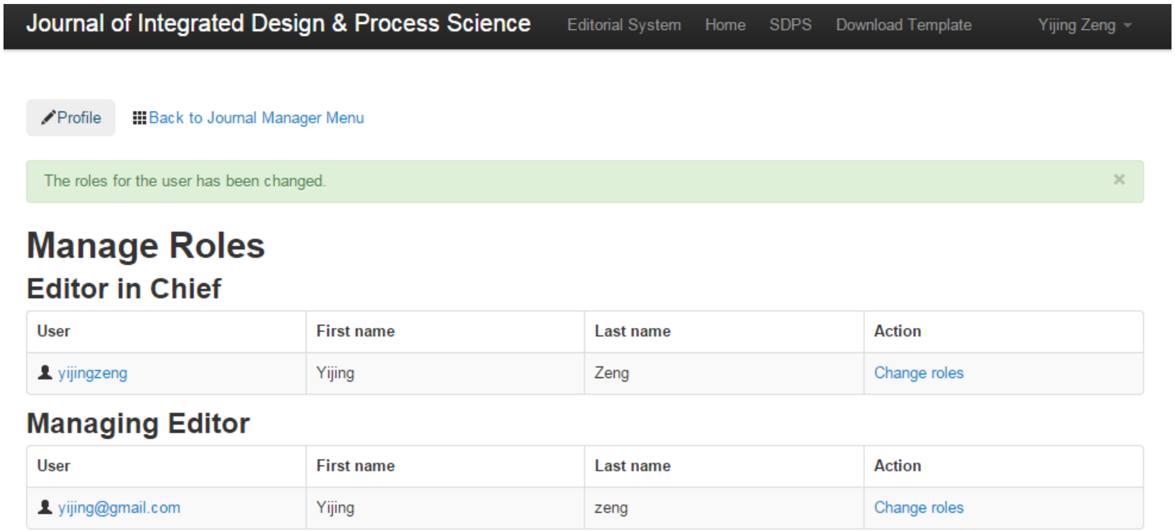

Journal of Integrated Design & Process Science
Editorial System
Home
SDPS
Download Template
Yijing Zeng
Profile
Back to Editor in Chief Menu
My roles
Editor in Chief
Managing Editor
Journal Manager
Handling Editor
Guest Editor
Reviewer
Proof Editor
Proof Reader
Author
Publisher
My tasks as Editor in Chief
Submissions waiting for review (1)
Submissions waiting for final decision (1)
Submissions sent for proof (2)
Submissions proofed and waiting for approval (2)
Submissions ready for Pre-Press (1)
Submissions Pre-Press posted (5)
My actions as Editor in Chief
Add reviewers
Special issues
Invite new user

Figure 23 Improved user main page

5.3 New functions of journal manager

Functions implemented for the journal manager include editing the system email template and managing users' role. The former function implemented by providing an interface to control and modify all system email templates. We list the link of each template which named the subject of that email. To implement the modification function, the link is associated with a new page which contains the email subject and content. The journal manager can edit the email template and save changes by clicking the submit button. Another function is the user role management and the interface is shown in Figure 24. This screen cut is not the whole page because the it is too large to put it all here. The interface of this function is similar to the former one that we listed all users classified by roles in the interface. At the end of this page, we develop a table contains all users of the system for the journal manager to select. The "Change roles" button is the function to view and change roles the user have. This implementation is based on system role mechanism, considered as extended function of the system management.


Journal of Integrated Design & Process Science
Editorial System
Home
SDPS
Download Template
Yijing Zeng
Profile
Back to Journal Manager Menu

The roles for the user has been changed.

Manage Roles

Editor in Chief

User	First name	Last name	Action
yijingzeng	Yijing	Zeng	Change roles

Managing Editor

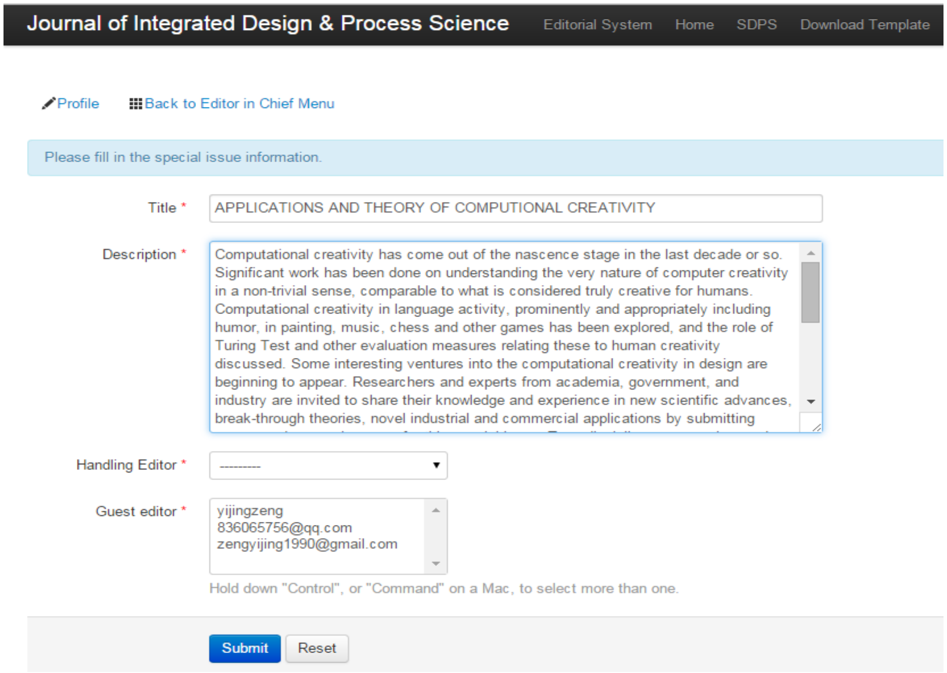
User	First name	Last name	Action
yijing@gmail.com	Yijing	zeng	Change roles

Figure 24 Journal manager manages users' role

5.4 New functions of special issue management

As we mentioned before, the editorial process of special issue paper differs from that of regular paper. Although the difference in the process, the functions and user interfaces of the handling editor and guest editor are very similar. So the development and implementation of this part can reuse the code of regular paper review. The main functions and user interfaces can be inherited from that of the editor in chief. Only several new functions required to newly develop. While we must modify the function of the new special issue creation due to the updated business workflow.

The special issue will be managed by the handling editor and the guest editor, we need to add functions to assign the special issue to certain users. We added two functions in the creation page of the special issue. The first one is to select handling editor implemented by setting a drop list. Because one special issue only needs one handling editor. While there is not only one guest editor for one special issue, so the selection of guest editor should be a multi-select function.


Journal of Integrated Design & Process Science
Editorial System
Home
SDPS
Download Template
Profile
Back to Editor in Chief Menu
Submit
Reset

Please fill in the special issue information.

Title * APPLICATIONS AND THEORY OF COMPUTIONAL CREATIVITY

Description * Computational creativity has come out of the nascence stage in the last decade or so. Significant work has been done on understanding the very nature of computer creativity in a non-trivial sense, comparable to what is considered truly creative for humans. Computational creativity in language activity, prominently and appropriately including humor, in painting, music, chess and other games has been explored, and the role of Turing Test and other evaluation measures relating these to human creativity discussed. Some interesting ventures into the computational creativity in design are beginning to appear. Researchers and experts from academia, government, and industry are invited to share their knowledge and experience in new scientific advances, break-through theories, novel industrial and commercial applications by submitting

Handling Editor * ---------

Guest editor * yijingzeng
836065756@qq.com
zengyijing1990@gmail.com

Hold down "Control", or "Command" on a Mac, to select more than one.

Figure 25 New special issue creation

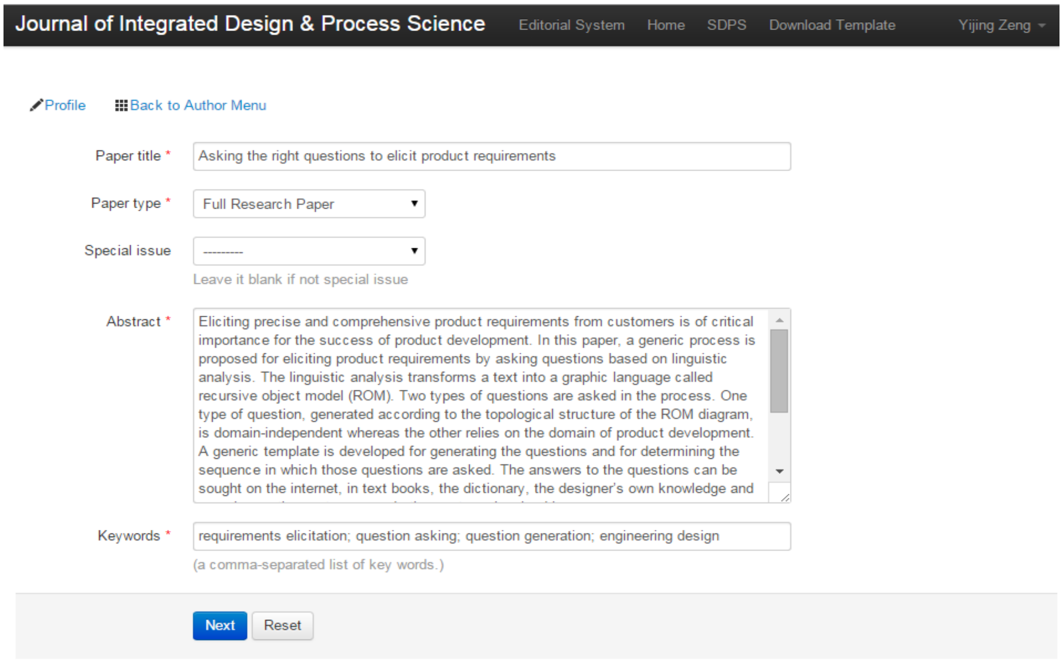
5.5 Improved paper submission process

In the old journal editorial system, the paper submitting process is simply which handled in a single webpage. Based on the previous page, the main intent of the submission phase redesign includes gathering more related information about the paper and giving authors more services to fulfill their satisfaction. Firstly, we modified the submission database to store more paper information submission by adding new fields in the database table. The next phase in the redesign is changing the aforementioned one-step paper uploading activity to a new three-step. Finally, according to the new submission workflow diagram, we will develop associated functions to support the new submission process.

5.5.1 New paper uploading steps

With the increasing amount of paper information, the one-step paper uploading with one webpage can’t load much submitting forms. So, we need to more webpages that should separate this activity into several steps. For implementation, we need to consider how to arrange new uploading action to support both existing functionalities and propose additional required direct inputs. We divide the entire uploading activity to three steps by analyzing the importance and relevance of proper information and reviewing a few online journal editorial system. To improve the development efficiency, we better reuse the old submission webpage and create two more new pages.

We adjust the original webpage of paper uploading to be the first submitting page in the new system by deleting three unnecessary options. The remained fields for display are paper title, paper type, special issue, abstract and keywords which are common combinations of input from the user accepted by existing journal systems. The previous button named “Submit” has been modified to the “Next” and the link of this button also altered to the webpage of the next uploading step. The first step of our editorial system manuscript uploading functions process is shown in Figure 26.



Paper title * Asking the right questions to elicit product requirements

Paper type * Full Research Paper

Special issue ---------
Leave it blank if not special issue

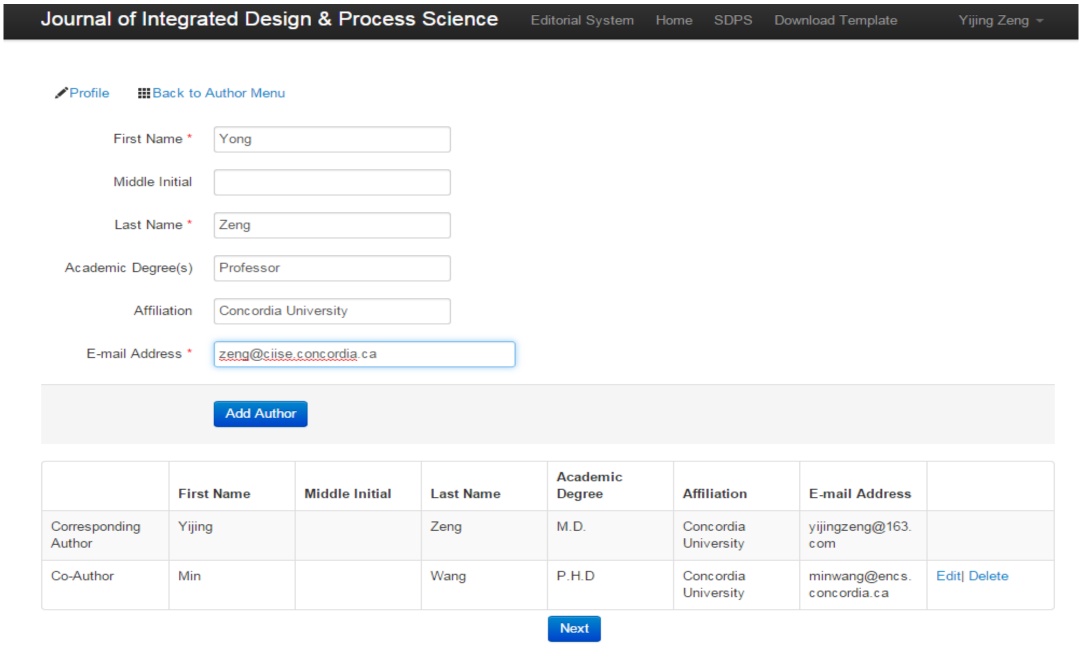
Abstract * Eliciting precise and comprehensive product requirements from customers is of critical importance for the success of product development. In this paper, a generic process is proposed for eliciting product requirements by asking questions based on linguistic analysis. The linguistic analysis transforms a text into a graphic language called recursive object model (ROM). Two types of questions are asked in the process. One type of question, generated according to the topological structure of the ROM diagram, is domain-independent whereas the other relies on the domain of product development. A generic template is developed for generating the questions and for determining the sequence in which those questions are asked. The answers to the questions can be sought on the internet, in text books, the dictionary, the designer's own knowledge and

Keywords * requirements elicitation; question asking; question generation; engineering design
(a comma-separated list of key words.)

Figure 26 First step of paper uploading

The task of the secondary step is to add/edit/remove paper’s corresponding authors’ information submitting to keep a record of the list of people who have authored this submission. After the user completes the first step and clicked the next button, he jumps to the webpage newly designed and implemented. The author who starts this submission will be listed as the corresponding author by default and the rest authors will be added in this step considered as the co-author. The Co-Authors do not have to be registered in our system.

As shown in Figure 27, there is a form for users to enter inputs that the first name, last name and e-mail address are required information. Once a user completes the form and clicks the add author button, the entered data will display in the table below the form. In order to provide convenience for users, we design functions to edit and remove added co-authors do not work for the corresponding author.



First Name *	Yong
Middle Initial	
Last Name *	Zeng
Academic Degree(s)	Professor
Affiliation	Concordia University
E-mail Address *	zeng@ciise.concordia.ca

	First Name	Middle Initial	Last Name	Academic Degree	Affiliation	E-mail Address	
Corresponding Author	Yijing		Zeng	M.D.	Concordia University	yijingzeng@163.com	
Co-Author	Min		Wang	P.H.D	Concordia University	minwang@encs.concordia.ca	Edit| Delete

Figure 27 Second step of paper uploading

The final step is to ask the authors to submit all the files associated with the paper to the editorial system. These files including PDF file, Source file, Reference file, Source archive, Author Biography/text, Author Biography/photo and Cover Letter that only the pdf file and source file is mandatory. There is a button named “Upload” in the final uploading page that related to both attach files’ submission and the whole paper uploading operation. Once this button has been clicked, the whole submission process is finished. The paper will store in the database and sent to the editor in chief.

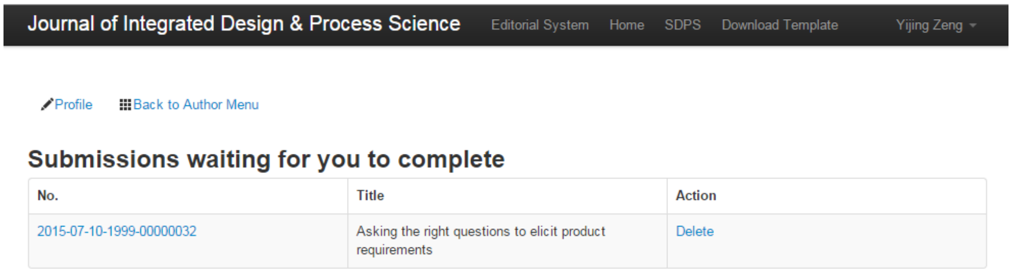
5.5.2 Extended paper submission functions

After we change the submission actions from one-step to three steps, the transaction of submission has been effected simultaneously. Above improvements make the diversification of submission workflow that the old system contains only one-step activity; users either finish their submitting or cancel the action. While the new submission process has three steps, which means users may break submitting before the final step. Considering the updated workflow diagram, we have to add additional functions to deal with the new paper status.

In order to manage possible situations, we must design the relative workflow control method. With the consideration of the authors' habits, we decide related rules. The first rule is that the paper will be recorded in the system after the author finished the first submission step and clicked the "Next" button. At this point, the system assigns a file number to the paper and saved its basic information. This file number is a unique number of distinct papers. The next rule is that all data entered during secondary and last submission step cannot be saved to the system until the author clicks the "Upload" button on the last step. This rule controls the break of submitting actions. Finally, it is better to allow authors approval their submissions before they are sent to our editorial system.

Because of the increasing variables in the paper table, these modifications lead to correlative workflow control functions as well as the interface change. The new system should provide corresponding functions to manage the new submission workflow. Compared with the old system, we need to create a main menu for authors and list new key folders about submission: submissions waiting for you to complete and submissions waiting for you to submit. These two items are collections for paper in those two new states and the number following each folder indicate how many submissions in each category are awaiting authors' attention.

The key folder "Submissions waiting for you to complete" contains papers in state submitting unfinished which are waiting for the author to complete the paper submission. Our system provides the function allows the author to continue his submitting actions. When users click the key folder "Submissions waiting for you to complete", the link of this item leads to jumping to a new webpage shows in Figure 28.



Submissions waiting for you to complete

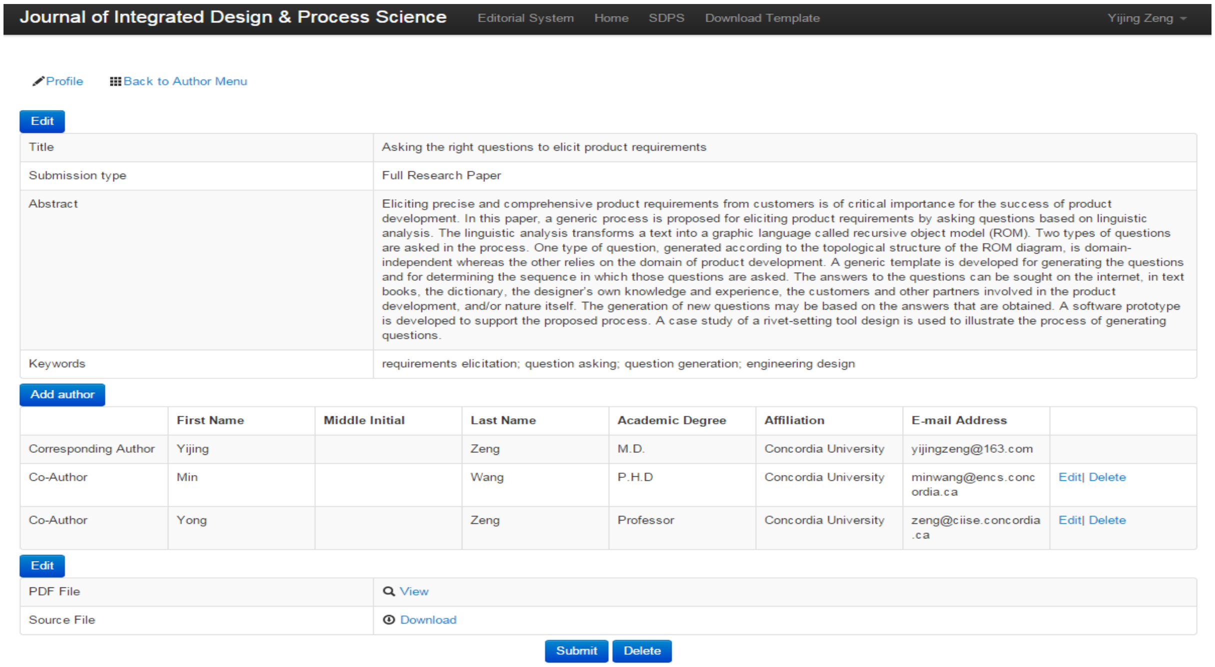
No.	Title	Action
2015-07-10-1999-00000032	Asking the right questions to elicit product requirements	Delete

Figure 28 Submissions waiting to complete folder

There is a table on this page lists all the submissions haven't finished submitting and waiting for the author to finish. Authors can also delete the paper by clicking the "Delete" button or click on the hyperlinked file number of any listed submission and review it. Click the link under the "No." column, the system will jump to the first paper uploading webpage with the entered data of the paper appear in the form for an author to complete this maintained submission continually.

We implement functions for authors to approve their submissions by connecting the key folder "Submissions waiting for you to submit" to a webpage similar to Figure 28. In this page, it contains all the submissions finished submitting but waiting for the author to approve and the link of submission file number will bring you to your submission's reviewing webpage shows in Figure 27. On this page, authors can either view all the information about the paper submission to verify the correction of listed content or make any required modifications. Here, the authors could revise the paper information by clicking the button in the top left of each table where buttons link to the corresponding paper uploading webpage. Users can modify their paper information in the same way as they submitted.

Finally, if the author makes sure that the paper has been approved, he can submit this submission to journal editors by clicking the "Submit" button under the table. After the author takes this action, the paper is successfully submitted to system editors. The state of paper will be changed and the paper will pass to the Editor in Chiefs of the journal and transits to the next reviewing phase.



Title	Asking the right questions to elicit product requirements
Submission type	Full Research Paper
Abstract	Eliciting precise and comprehensive product requirements from customers is of critical importance for the success of product development. In this paper, a generic process is proposed for eliciting product requirements by asking questions based on linguistic analysis. The linguistic analysis transforms a text into a graphic language called recursive object model (ROM). Two types of questions are asked in the process. One type of question, generated according to the topological structure of the ROM diagram, is domain-independent whereas the other relies on the domain of product development. A generic template is developed for generating the questions and for determining the sequence in which those questions are asked. The answers to the questions can be sought on the internet, in text books, the dictionary, the designer's own knowledge and experience, the customers and other partners involved in the product development, and/or nature itself. The generation of new questions may be based on the answers that are obtained. A software prototype is developed to support the proposed process. A case study of a rivet-setting tool design is used to illustrate the process of generating questions.
Keywords	requirements elicitation; question asking; question generation; engineering design

	First Name	Middle Initial	Last Name	Academic Degree	Affiliation	E-mail Address	
Corresponding Author	Yijing		Zeng	M.D.	Concordia University	yijingzeng@163.com	
Co-Author	Min		Wang	P.H.D	Concordia University	minwang@encs.concordia.ca	Edit| Delete
Co-Author	Yong		Zeng	Professor	Concordia University	zeng@ciise.concordia.ca	Edit| Delete

PDF File	View
Source File	Download

Figure 29 Submission confirms page

5.6 Major improvements in reviewing process

Although there are a lot of functions in the old system supporting review process, it is still plenty of room for enhancement. The improvements in this process in order to enhance the usability and utility of the system. Here we will introduce most of the new functions in the new version. Additionally, we will still describe several updated points to see how we refactored the old system to better serve users of our new system.

5.6.1 New status of paper

The workflow diagram in chapter 3 reflects the major redesign of it in the review process is the new state named "required review completed". This status indicates that the number of reviewers who complete their review tasks is equal to or more than the quantity number set by the editor in chief before. It is an automatic mechanism for peer review management when the peer reviews have been completed the state of paper changed automatically. In order to achieve this new feature, the new system required the implementation of a series of new functions.

1. Set required number of reviewers

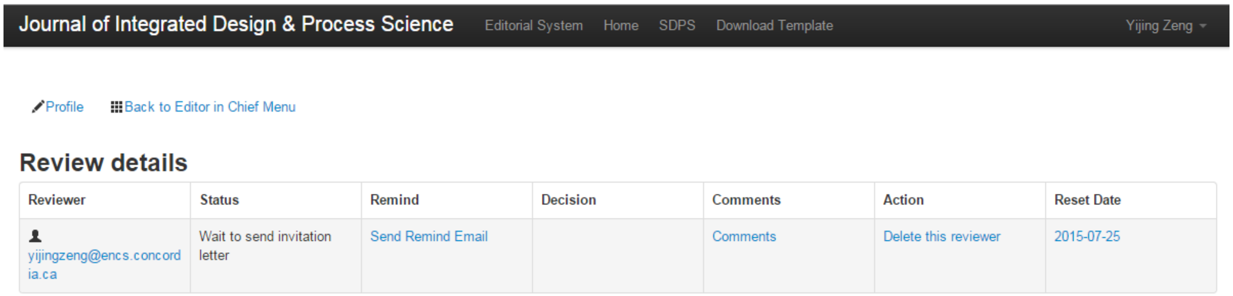
The first function the current system has to provide is the setting of required number of reviewers. After discussion, we locate the function under the wait for review state since this is the initial state of the formal review process. Every new submission can have its own number variable and the default value given by the system is set to 2. The purpose of this number is to manage the completeness of review, but it can also be used to take care of acceptable response on the review task. For this purpose, we implement this control rule that if the accumulated number of reviewers who accept the review tasks assigned by the editor in chief is not lower than the required number of reviewers, the paper will move from the state waiting for review to next state under review.

2. Notice reviewers to terminate the review task

The rule of the required number of reviewers is it has no condition of the make modification when the peer review is completed. There may be a situation that some reviewers still doing the review work while the state of the paper has been transferred to required review completed due to the automatic control. To deal with this problem and avoid the displeasure of reviewers, we should send an email message to those reviewers to notice them the situation they meet and remind them to terminate the review task. This function must be an action triggered auto send email message because the system response should be immediate.

5.6.2 New functions to manage reviewers

To solve the limitation problems stated in the problem analysis chapter, we develop new functions for the editor in chief to better manage the reviewing activity. As shown in Figure 30, there are several new functions listed in the table. First, under the remind column, the button named “Send Remind Email” is a reminder function that an email to remind reviewers about their review task will be sent automatically by clicking it. Another new function is to delete the reviewer provided to the editor in chief to decrease the workload of managing reviewers. The last new function is to reset the review deadline for the reviewer which is a convenient method for both reviewers and editor in chiefs. These new functions are designed based on the user test activity.


Journal of Integrated Design & Process Science
Editorial System
Home
SDPS
Download Template
Yijing Zeng
Profile
Back to Editor in Chief Menu

Review details

Reviewer	Status	Remind	Decision	Comments	Action	Reset Date
yijingzeng@encs.concordia.ca	Wait to send invitation letter	Send Remind Email		Comments	Delete this reviewer	2015-07-25

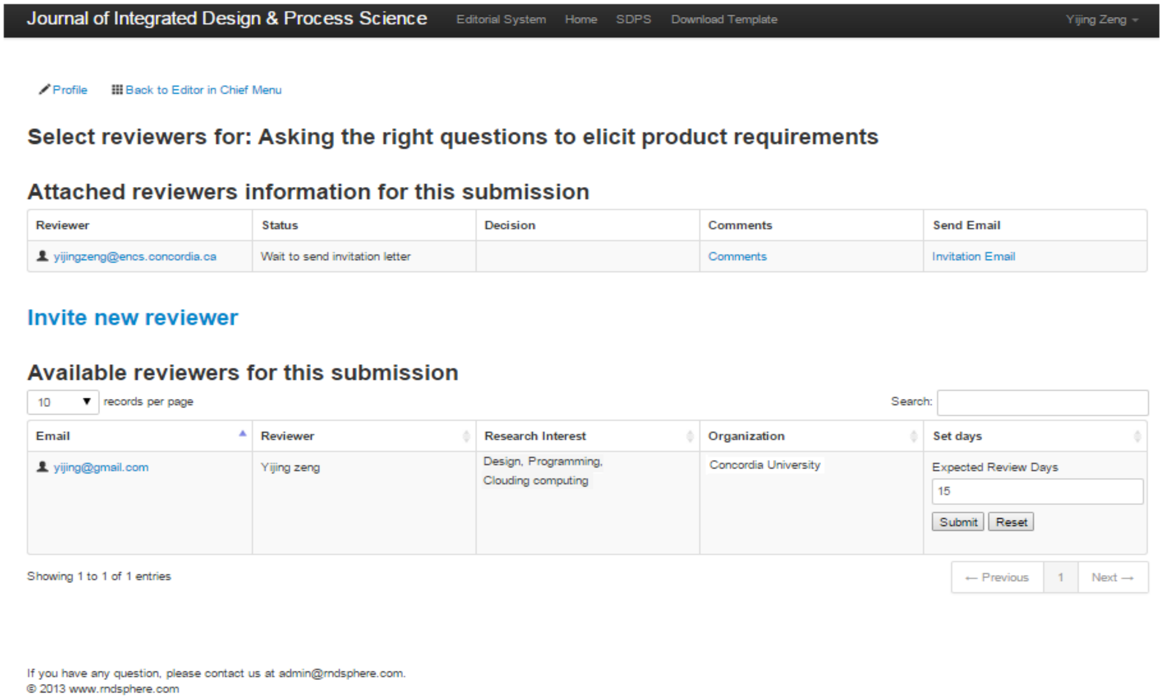
Figure 30 Editor in chief managing reviewers

5.6.3 Save comments function

This function is required based on the actual user experience of various classes of editors who often take the operation of submitting results. As they say, the task of writing comments always takes time and easily be broken off. As a solution result, the new system provides a function to save the decision and comments for several editors. Considering the comments should be able to save and access at any time, the implementation of this function get the help from the state management system. We develop the saving function by adding a proper state named waiting for final decision which correspond to the papers has saved a temporary comments and decision. Because we add a state for the submission, we need to add related key folder in the main page under the main tasks menu. The rule of this function is that users can save decision and comments all the time by using saving function before they submitting the final version of review result.

5.6.4 Improvement of reviewer invitation

In the old system, the system invitation function is a link which links to another webpage to enter the expected review days. The editor in chief is required to modify the invitation letter before it has been sent to authors by the system. The first problem can be solved by reducing the number of options that we can integrate the content in the expected review days input webpage to the review invitation page. For the latter one, we generate a new function which is a webpage for the editor in chief to review and modify the invitation letter. The function is the link named “Invitation Email” at the end column of attached reviewer’s table as shown in Figure 31. On this webpage, we also make a little change in the available reviewer’s table that the column of research interest and organizations are added. This enhancement is not big, but can help the editor to select reviewers by reviewing their relative profile information directly.


Journal of Integrated Design & Process Science
Editorial System
Home
SDPS
Download Template
Yijing Zeng
Profile
Back to Editor in Chief Menu

Select reviewers for: Asking the right questions to elicit product requirements

Attached reviewers information for this submission

Reviewer	Status	Decision	Comments	Send Email
yijingzeng@encs.concordia.ca	Wait to send invitation letter		Comments	Invitation Email

Invite new reviewer

Available reviewers for this submission

10 records per page   Search:

Email	Reviewer	Research Interest	Organization	Set days
yijing@gmail.com	Yijing zeng	Design, Programming, Clouding computing	Concordia University	Expected Review Days 15 Submit Reset

Showing 1 to 1 of 1 entries

If you have any question, please contact us at admin@rndsphere.com.
© 2013 www.rndsphere.com

Figure 31 Updated reviewer invitation webpage

5.7 Summary of improvements

In this chapter, we discussed the major enhancements to existing functions in our new journal editorial workflow management system. The new journal system has a large number of newly designed functions and lots of updates functions enhance the system utility and complete the practicality for journal management. They solve the drawbacks and limitations of the old system easily and efficiently by designing appropriate functions to the right user at the right moment and providing users a better interface. We put the effect to improve the usability of users while implementing these new functions. Besides the functions inherited from the old system, addition of extended functions would allow the administrators of journals to easily manage business tractions by using the system to support most journal events.

6 Result

In order to evaluate the new journal editorial workflow management system, we conducted some surveys and user test. Comparisons were made between the old and new system in order to make a comparative evaluation. The survey we execute acquired information from the log files of those systems. Participants of user test are actual users of the journal who represents the user base of the journal management system.

The result contains three aspects: the comparison of submission functions, usefulness of proofreading and publication management, comparative user satisfaction test of the review functionality.

1. Comparison of submission functions

There are major modifications of the submission process in the new system. As a result, users have a better experience with the new system and the error rate of submission is decreased. The error here indicates to the wrong submitted paper information. We did a survey on the quality of submissions submitted in one year. We found that the error rate of submission changed from 10% (6/58 papers) in the previous system down to nearly 1% (1/78papers) in the new system. It shows a great enhancement of the redesigned workflow and newly developed functions.

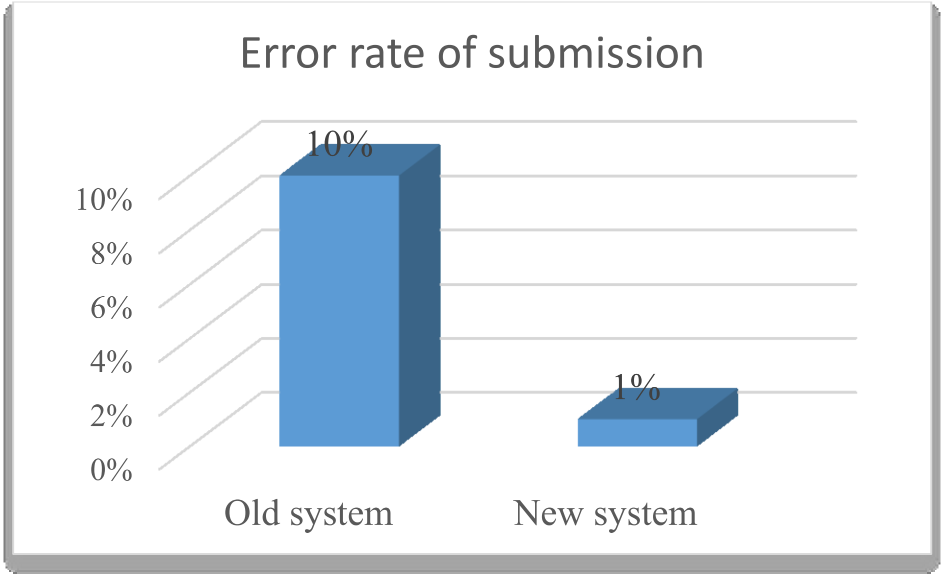


Figure 32 Comparison of submission error rate

2. Proofreading and publication process management

Functions to manage both the proofreading and publication processes are newly designed and implemented. It was not supported in the old system that these processes are highly manual, insecure and time consuming. We conducted an interview to see the difference of managing the journal proofreading and publishing processes by hand and with the assistance of the system.

From related system users' feedback, we can summarize two advantages of the system. First, it is secure and convenient to manage papers using the system. The system can record most of the transitions and various versions of papers which reduce their workload to manage the electronic documents by hand. All information is kept in the system that the security risk like losses of files is eliminated. Second, the time spent on proofreading and publishing was seen reduced. One obvious change is the case that before they contacted and sent manuscripts through e-mail wrote manually. But now the system can send the notification email automatically and the papers can be directly received through the system. As a proof editor estimated, he can save nearly 15 days after use the system. Reduced the time in the proofreading process is benefit of system users.

3. Comparative user test of review functions

We enhanced the review process in details in our redesign process. To show the development result, we organized a satisfactory investigation among users with editor in chief role. We conducted a user test which asks participants complete some test tasks as performance indicators and capture their evaluation result. In order to make contrast, this test carried out on both the old and new systems.

Test Tasks:

1. Register a new account of the system;

2. Login to the system;

3. Acknowledge a paper titled "Asking the right question";

4. Find a reviewer whose research interest contains "Design logic";

5. Invite a reviewer whose email is yijing@encs.concordia.ca;

6. Set the deadline for above review invitation as 20 days;

7. Invite three reviewers for a paper titled “Journal management system analysis and design”;

8. Find a paper titled “Test article” with the under review state;

9. Check the review status of a paper titled “An Analysis of Platforms for Scholarly Publication”;

10. Make a decision and comments for a paper that its file number is 2015-04-11-1999-00000024;

11. Revoke a paper submitted by the author “test user 1”.

During the user tests, we give the criteria for users to evaluate their test tasks by marking the task as:

Failed to fulfill task: 0 point

Completed: 1 point

Completed within two minutes: 2 points

The results are shown in Figure 33 and we can see that the satisfaction score of the new system is higher than the old one. It is evident that a considerable improvement in convenience and efficiency with the main review tasks were completed on the new system.

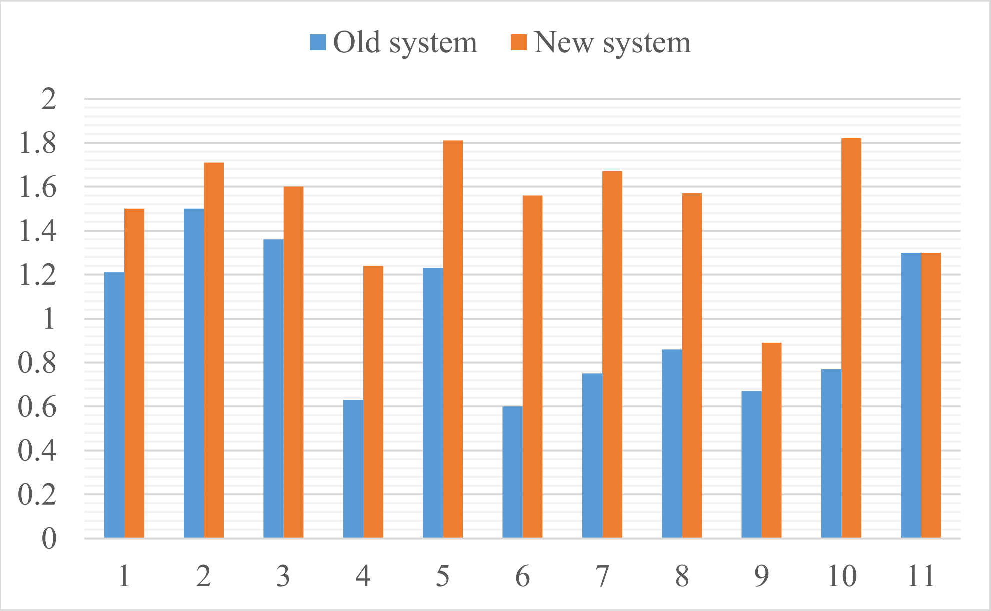


Figure 33 Satisfaction scores of test tasks for old and new system

7 Conclusion and future work

7.1 Conclusion

In conclusion, we have discussed the redesign problem of a journal editorial workflow management system. We have used EBD processes including environment analysis, problem identification and solution generation processes to analyze the problem fully and systematically.

In the environment analysis process, we have summarized the general workflow a journal management system should have. Based on those results, we have obtained the abstract workflow of our journal which is also the user requirement by using the method of asking the right question. The analysis result of this stage was used to support the other two activities.

In the problem identification process, we have focused on finding problems of the old journal management system by conducting user tests and interviews. We generally classified the problems of the system into drawbacks and limitations. The solution of the problem has been broken down to solve those drawbacks and limitations of the system.

In the solution generation process, the new system has been developed based on the analysis result of the redesign problem, concepts generalized from the user requirements and experiences learnt from the previous version of the system. In the new system's design phase, use case and updated editorial workflow diagrams have been drawn. And then, system data model, system architecture and physical database redesigns have been completed.

We presented many improvements and enhancements in functions and user interfaces of the new system. As the result stated in chapter 5 and chapter 6, the new system enhances the efficiency and accuracy of the system by providing easy and rapid management of the electronic journal compared to the old system. The new system implements the management of the automated transfer of papers. It also enables a journal manuscript to be published entirely on the web by adding the proofreading and publication editorial processes. Thereby reducing the time from manuscript submission to publishing. The new journal management system has been deployed in a cloud server and used for more than one year. The transaction logs show the system is highly reliable.

7.2 Future work

It is very easy to further improvement of the system with new features because of the flexibility of the design. Built with the current system, it can be predicted that the functions of the system will become more automated with more advances in information technology. Both the server and client side application of information technology in a journal editorial workflow management system can be expected to illuminate new vistas of innovation. To sum up, our journal management system can be enhanced to get better performance by the developing technology. Some addition works can be considered in the future to make the system better are listed.

1. An intelligent help system

The new system now provides some context instructions in some webpage to the users, but many users paid less attention to them when using the system. So we need an intelligent help system to provide better services and give users the real-time tips.

2. Multi-Language Support

With increasing globalization, journal systems are expected to support various kinds of languages to be more international. Multi-language support functionality could facilitate many important processes and be convenient for some non-English users when managing an international journal. So in the next step, the developers of the journal system may consider putting effort in multi-language support.

3. A backup system

Redesign and development work was mainly focused on implementing the functionalities of journal management while a backup system is missing. A simple administrative level backup with system copy and dumps is adequate. However, to keep the software and data safe, a backup system is required to be implemented in the future. The backup system should not only backup the databases used by the system, but also the files associated with the system.

4. Multi-journal support

The journal management system is currently developed for the JIDPS journal only. The future work will focus on realizing a multi-press journal management system to host more than one journal in one application and put in common a set of services. This approach could improve the versatility, quality and flexibility of electronic journal editorial workflow management system, as well as to extend access to research.

References

[1] *JIDPS*. Available: http://www.iospress.nl/journal/journal-of-integrated-design-process-science/

[2] R. Clarke and D. Kingsley, "e-Publishing's impacts on journals and journal articles," *Journal of Internet Commerce,* vol. 7, pp. 120-151, 2008.

[3] M. Huang, Y. Feng, and B. C. Desai, "ConfSys2: an improved web-based multi-conference management system," presented at the Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering, Montreal, Quebec, Canada, 2009.

[4] M. Huang, "CONFSYS2: a redesigned web-based multi-conference management system," Concordia University, 2009.

[5] L. Jiaguo, "Design and Implement of Journal Office Automation System Based on WWF," in *2010 International Conference on Electrical and Control Engineering*, pp. 2718-2721, 2010.

[6] S. K. Lippert and C. Govindarajulu, "Technological, organizational, and environmental antecedents to web services adoption," *Communications of the IIMA,* vol. 6, pp. 14, 2015.

[7] A. Elizarov, D. Zuev, and E. Lipachev, "Electronic scientific journal-management systems," *Scientific and Technical Information Processing,* vol. 41, pp. 66-72, 2014.

[8] S. Muthayan, "Open access research and the public domain in South African universities: the Public Knowledge Project's Open Journal systems," *Open Access and the Public Domain in Digital Data and Information for Science,* pp. 134-145, 2004.

[9] S. P. Muir, M. Leggott, and J. Willinsky, "Open journal systems: An example of open source software for journal management and publishing," *Library hi tech,* vol. 23, pp. 504-519, 2005.

[10] T. Ehling, "DPubs: the development of an open source publishing system," *Publishing research quarterly,* vol. 20, pp. 41-43, 2005.

[11] M. Pickton, "Northampton Open Journals: a publishing service for researchers," *ALISS Quarterly,* vol. 8, pp. 16-21, 2013.

[12] *Public Knowledge Project*. Available: https://pkp.sfu.ca/ojs/

[13] *State and Perspectives of Ukrainian scientific periodic development*. Available: http://nv.nmu.org.ua/index.php/ru/glavnaya/38-ruscat/novosti/86-sostoyanie-i-perspektivy-razvitiya-nauchnoj-periodiki-ukrainy

[14] A. Leff and J. T. Rayfield, "Web-application development using the model/view/controller design pattern," in *Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International*, pp. 118-127, 2001.

[15] *Euclid Project* Available: www.projecteuclid.org

[16] *Max Planck Society*. Available: https://dev.livingreviews.org/projects/epubtk/)

[17] *Management*. Available: https://en.wikipedia.org/wiki/Management

[18] H. Bogunović, E. Pek, S. Lončarić, and V. Mornar, "An electronic journal management system," in *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*, pp. 231-236, 2003.

[19] W. Van Der Aalst and K. M. Van Hee, *Workflow management: models, methods, and systems*: MIT press, 2004.

[20] Y. Zeng, "Environment-based design (EBD)," in *Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2011)*, 2011.

[21] Y. Zeng, *Environment-based design*. Berlin: Springer, 2015, to appear.

[22] Y. Zeng, "Axiomatic theory of design modeling," *Journal of Integrated Design and Process Science,* vol. 6, pp. 1-28, 2002.

[23] Y. Zeng and G. D. Cheng, "On the logic of design," *Design Studies,* vol. 12, pp. 137-141, 1991.

[24] Y. Zeng, "Environment-based formulation of design problem," *Transaction of SDPS: Journal of Integrated Design and Process Science,* vol. 8, pp. 45-63, 2004.

[25] *Redesign*. Available: https://en.wiktionary.org/wiki/redesign

[26] Y. Zeng, "Recursive object model (ROM) - Modelling of linguistic information in engineering design," *Computers in Industry,* vol. 59, pp. 612-625, 2008.

[27] G. Page, R. Campbell, and J. Meadows, *Journal publishing*: Cambridge University Press, 1997.

[28] J. Ciesielski, "How are electronic journals cooked?," in *11 th Conference on Professional Information Resources, Prague*, 2005.

[29] R. Dale 1, "A RULE‐BASED APPROACH TO COMPUTER‐ASSISTED COPY‐EDITING," *Computer Assisted Language Learning,* vol. 2, pp. 59-67, 1990.

[30] H. A. Reijers, *Design and control of workflow processes: business process management for the service industry*: Springer-Verlag, 2003.

[31] Z. Y. Chen and Y. Zeng, "Classification of product requirements based on product environment," *Concurrent Engineering: Research and Applications, An International Journal,* vol. 14, pp. 219-230, September 2006.

[32] R. K. Ko, S. S. Lee, and E. Wah Lee, "Business process management (BPM) standards: a survey," *Business Process Management Journal,* vol. 15, pp. 744-791, 2009.

[33] K. Zhao, "ConfSys3: An Online Academic Conference System," Concordia University, 2012.

[34] B. van Loggem, "User documentation: the Cinderella of information systems," in *Advances in Information Systems and Technologies*, ed: Springer, pp. 167-177, 2013.

[35] R. Pierce, "Using customer input to drive change in user assistance," in *Proceedings of the 26th annual ACM international conference on Design of communication*, pp. 23-30, 2008.

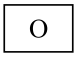
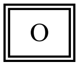
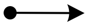
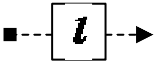
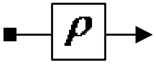
[36] R. T. Watson, "The academic publication system: A time for redesign," ed: University of Georgia, 2005.

[37] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distributed and parallel Databases,* vol. 3, pp. 119-153, 1995.

Appendices

The recursive object model[26] (ROM) was proposed by Zeng to conduct this work. It is used for analysis the linguistic design problem statement. The ROM includes two types of objects, which are objects and compound object, and three kinds of relations between any two objects: connection, constraint, and predicate (as shown in Table X).

Table 4 Elements of Recursive Object Model (ROM)

Type		Graphic Representation	Definition
Object	Object	 O	Everything in the universe is an object.
	Compound Object	 O	It is an object that includes at least two other objects in it.
Relations	Constraint		It is a descriptive, limiting, or particularizing relation of one object to another.
	Connection	 ι	It is to connect two objects that do not constrain each other.
	Predicate	 ρ	It describes an act of an object on another or that describes the states of an object.

Another essential activity is question asking and answering. The question asking help to determine the solution direction of the design problem. The template[21] and rule for asking questions is list following:

Table 5 Rules for question asking

Rule 1	Before an object can be further defined, the objects constraining them should be refined.
Rule 2	An object with the most undefined constraints should be considered first.
Rule 3	If an object has the most number of constrains and/or predicates on other objects, then it should be considered first.

Table 6 Question template for object analysis

#	Conditions	Question template

T1	For a concrete, proper, or abstract noun object N without any constraint	What/Who is N?
T2	For a concrete, proper, or abstract noun N with an adjective constraint A	What is A N?
T3	For an noun object A constraining an noun object N	What is/are A N?
T4	For a verb V with its subject N1 and object N2	What do you mean by V in the statement “N1 V N2”? How do/does N1 V N2? Why do/does N1 V N2? When do/does N1 V N2? Where do/does N1 V N2?
T5	For a verb object V constrained by an adverb A with its subject N1 and object N2	What do you mean by V A? Why do/does N1 V A N2? When do/does N1 V A N2? Where do/does N1 V A N2?
T6	For a verb V with an object N, but missing its subject	What/Who V N?

And the answering strategy is a roadmap which was proposed as guidance for requirements modeling. Chen and Zeng[31] categorized product requirements into eight levels: natural laws, social law and regulations, technical limitation, cost, time and human resource, basic functions, extended functions, exception control level, and human-machine interface, as is shown in Figure. They can be divided into two major groups: non-functional requirements, and functional requirements. The lower four: natural law and rules, social law, regulations, technical limitations, cost, time, and human resource level are usually non-functional requirements. The upper four, including basic functions, extended functions, exception control, and human-machine interface, are usually functional requirements.

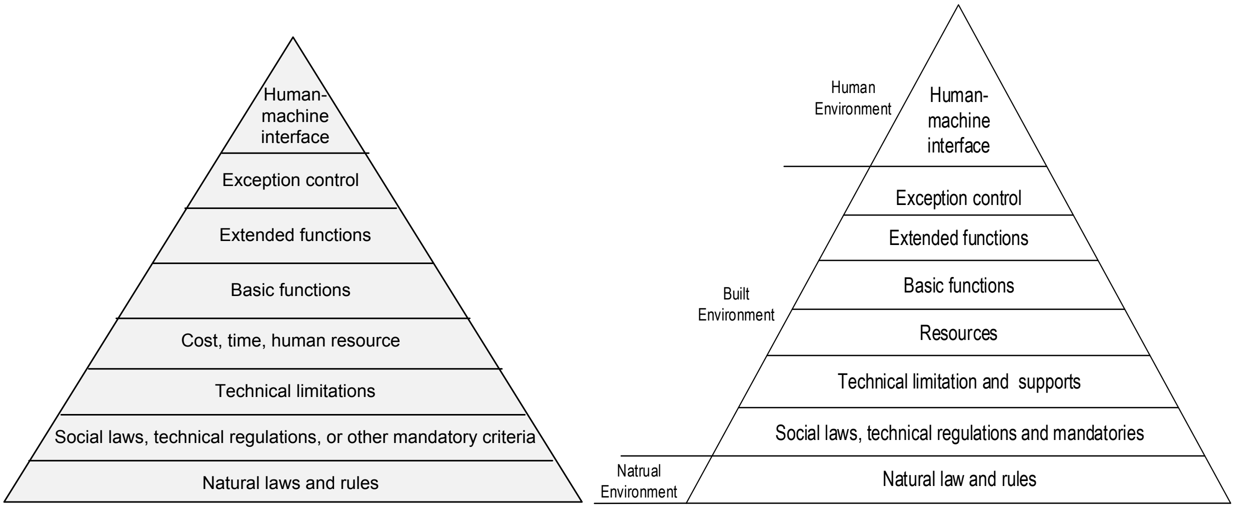


Figure 34: Levels of product requirements

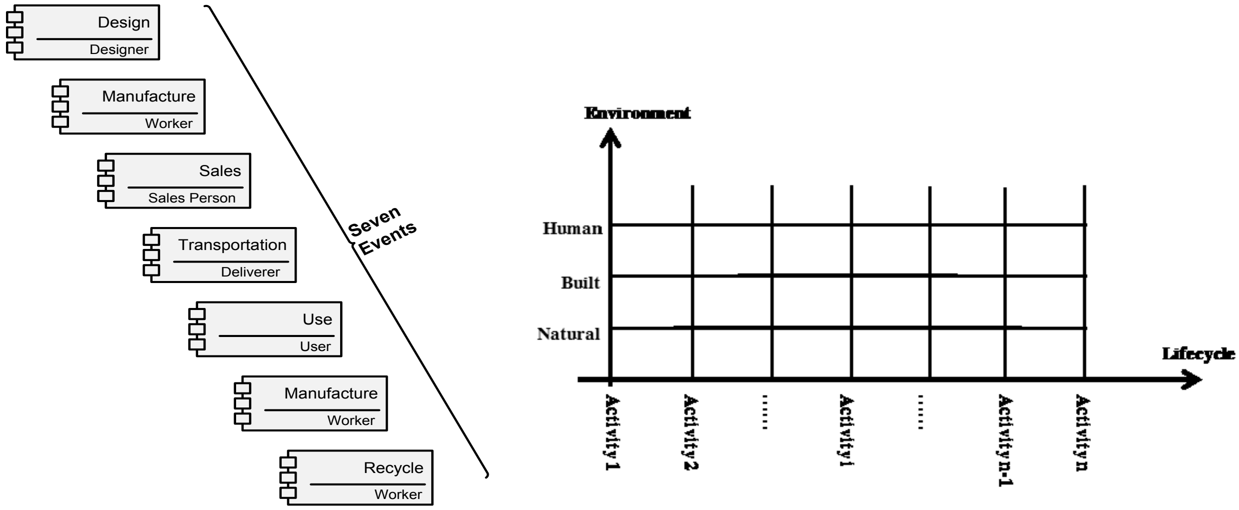


Figure 35: Classification of product environment

Based on above roadmap, EBD provides the answering template[21] as following to help designer.

Table 7 Guideline to answer question

#	Questions	Guideline
G1	What/Who is N? N: a concrete, proper, or abstract noun object	a) If (A)N is the product to be designed, then the answer should address 1) the purpose of (A)N; 2) the definition of (A)N according to Figure 34;
	What is A N?	b) Else, if N is an environment component of a product, then the

	A: an adjective constraint	answer should define (A)N according to Figure 34; c) Else, the components and attributes of N should be described.
G2	What/Who do/does V N? V: a verb	For N1 that V N, the answer should define the components and attributes of N1 in the context of V.
G3	When do/does N1 V N2?	The answer may assume one of the following two forms: a) In/On a time, N1 V(A) N2; b) When/During/While N3 Va N4, N1 V(A) N2.
	When do/does N1 V A N2?	
G4	Where do/does N1 V N2?	The answer may assume one of the following two forms: a) In/Along/Through a place, N1 V(A) N2; b) N3 Va N4, where N1 V(A) N2.
	Where do/does N1 V A N2?	
G5	Why do/does N1 V N2?	The answer should be organized as: To Va Na, N1 V (A) N2.
	Why do/does N1 V A N2?	
G6	What do you mean by V?	a) If the subject (N1) or object (N2) of V is not the product, then the answer should include all activities included in V-ing in the context of N1 and N2; b) Else, skip the question and leave for solution generation.
	What do you mean by V A?	
	How do/does N1 V N2?	