國立交通大學

資訊科學與工程研究所

碩士論文

以知識庫的方式設計與實作一個可重新配置的論文審稿系統產生器

The Design and Implementation of a Reconfigurable Paper-Review System Generator using Knowledge Based Approaches

研 究 生：吳嘉祥

指導教授：曾憲雄　博士

中華民國 九十九年七月

以知識庫的方式設計與實作一個可重新配置的論文審稿系統產生器

The Design and Implementation of a Reconfigurable Paper-Review System Generator using Knowledge Based Approaches

研究生：吳嘉祥　　Student：Chia-Hsiang Wu

指導教授：曾憲雄　　Advisor：Shian-Shyong Tseng

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

July 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

以知識庫的方式設計與實作一個
可重新配置的論文審稿系統產生器

研究生：吳嘉祥　　　　　　　　　　　　　　　　指導教授：曾憲雄 博士

國立交通大學資訊學院
資訊科學與工程研究所

摘要

由於近幾年研討會/期刊論文的出版已有日漸成長的趨勢，對於出版研討會/期刊論文的研究單位而言，審稿機制已漸漸的受到重視。然而他們的稿件審稿流程，往往隨著研究單位的不同而不同，即便是對於同一個研究單位而言，其審稿流程需求也常常隨而時間而改變。基於重複使用性、維護性、以及可靠性的考量，本篇論文使用知識工程的方式。提出了一個可重新配置的論文審稿系統產生器。我們提出了一個**流程描述模型**，它是由**框架式知識表式方法**與**規則式流程控制邏輯**所組成，此模型可以讓使用者產生一個符合他們審稿流程的論文審稿系統。此篇論文亦探討了系統中使用者存取控制以及審稿過程匿名性控制的問題。我們也提出了一個互動式的流程編輯工具，讓使用者去配置他們的審稿流程。在此篇論文的的最後，我們設計了一個實驗，根據流程表達能力與動態修改能力對我們的系統進行評估，實驗結果顯示，與其他的幾個系統當中，我們的系統有較強的流程支援性與動態修改能力。

關鍵字：論文審稿系統、框架式知識表式方法、規則式流程控制、互動式的編輯工具

The Design and Implementation of a Reconfigurable Paper-Review System Generator using Knowledge Based Approaches

Student：Chia-Hsiang Wu **Advisors：Dr. Shian-Shyong Tseng**

Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

Since there is an increasing trend about publishing conference/journal, the paper-review process becomes an important process for the different conference/journal publisher. However, different research groups may have their own paper-review processes. Even for a specific research group, the requirement for the process may be variable. For the sake of reusability, maintainability, and reliability, this thesis proposed a reconfigurable paper review system generator, which was designed by knowledge based approach. We presented the **process description model** which consisting of **frame-based knowledge representation** and **rule-based process control logic** to facilitate users generating the paper-review systems with their own paper-review process. This thesis discussed the issue of access control and the anonymity control in the paper review system. An interactive authoring tool was proposed to facilitate user configure their paper-review process. In the end of this thesis, an experiment was presented to evaluate the expressive power and modifiability of the proposed system. Comparison among the proposed system, traditional, and template-based paper-review systems was provided and the result showed that the proposed system could support the most functions and was the most modifiable among the compared systems.

Keywords: paper-review system; frame-based knowledge representation; rule-based process control; interactive authoring tool

誌謝

這篇論文的完成，首先必需感謝我的指導教授，曾憲雄博士，在撰寫論文的過程當中，我不時發現自身能力的不足，在老師耐心的指導、提示之下，才讓我能夠在碩士的學術研究逐漸有了成果，回憶過去一年來，一次又一次的和老師的討論過程中，除了完成了這篇學術論文外，我也得到了老師對於知識工程這門學問的觀點、研究方法、思考方式與表達方式各方面的訓練，這些我認為又是比學術論文更珍貴的東西。也感謝袁賢銘老師、楊鎮華老師與洪宗貝老師，在口試的過程當中，也給了我不少寶貴的意見，讓整篇論文更加的完整，也為我的研究提示出更多未來可以發展的方向。

再來我要感謝的，是俊銘學長和喚宇學長，在我研究的過程當中，能夠不厭其煩的和我進行討論；在我研究過程中，能夠適時給予建議與協助，讓我能更順利的完成此篇論文，也謝謝實驗室其他的學長姐：瑞鋒學長、元昕學長、宗儒學長、哲青學長、怡利學姐、靖雅學姐，謝謝你們在平常給我的指導與建議。接著，當然就是實驗室的好同學們：杰峰、國彰、紹宜、佳榕與金龍，兩年的相處過程中，我們一起經歷了很多，除了曾一起打混，也曾一起熬夜看交大的日出；佳玫學妹、楷元學弟：雖然相處的時間只有一年…但有了你們，讓我們學長姐在水深火熱之中能舒緩一些壓力，謝謝你們！感覺我們碩班是實驗室中長不大的一群小孩，以後還是要常常聯絡喔！

接著要感謝的是實驗室畢業已久的順傑學長，感謝您啟發我對未來職場前進的方向，也多虧了您，讓我能夠遇到更多和我有相同目標的一群好朋友！也謝謝亨龍工業的紀先生和李先生，您們在我求學的過程中給予我最實質的幫助，這份恩情，我可能一輩子都還不清，但若日後能有所成就，我一定會多幫助其他也需要幫助的人！

最後的最後，我要感謝我的家人，感謝我的姑姑總是為我處理大小事情；感謝老哥如此的照顧我；感謝我的母親，很抱歉常常讓您操心；也感謝在另一個世界關心我的父親，我不是一個善待自己身體的人，能撐到現在，一定是您在另一個世界有保祐！也感謝我的女朋友文惠，感謝您無時無刻給我的支持與關心！

求學之路前前後後算是跌跌撞撞的走了二十六個年頭，雖然不算是很有效率的走完這段路…但我相信，在我生命中的每一段經歷，對於未來的我都有相當的幫助！我會謹記這段求學的過程，去面對未來更多的挑戰！

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

Academic journals and conferences are important media for researchers to share research ideas and findings, and provide studying resources for researchers to enter a new research domain. Many research groups and publishers published their own journals and conferences for specific research topics. For example, IEEE society [11] published many journals and conferences for technology domain, journals of ACM [12] focus on computer science research. According to the amount of call-for-paper, as shown in Figure 1.1, academic journals and conferences showed a trend of increasing over years. All these journals and conferences have a paper-review process, controlling a procedure of submitting, reviewing, and publishing articles. The processes may be complicated and varied for policies of various research groups, so manually managing a paper-review process is time consuming and costly. Thus, many research groups construct paper-review systems to automatically manage their paper-review processes.

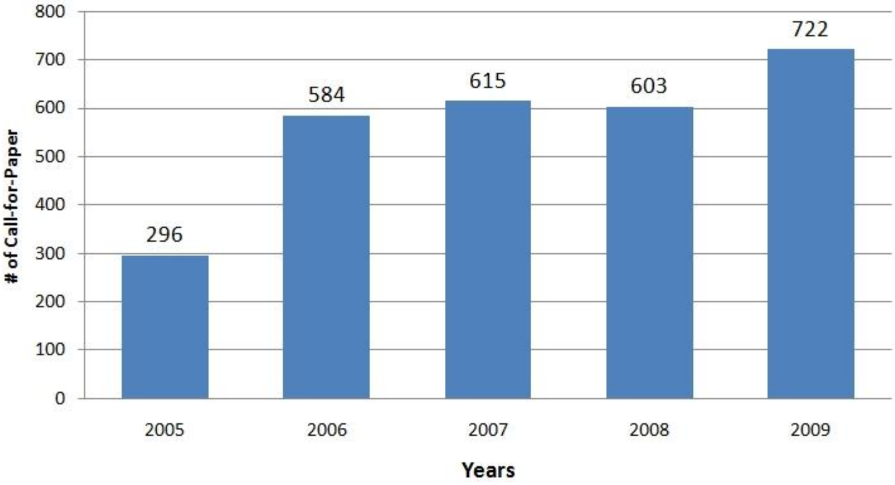


Figure 1.1: The trend of number of call-for-paper from 2005 to 2009, which was collected from ACM [12]

Maintainability of an online paper-review system is important because the paper-review process of a conference or a journal might be changed for various requirements or varied scale of the conference or journal. Many studies [1][2][3][13] proposed paper-review systems for specific paper review processes. However, reliability of these paper-review systems might be low because these systems were frequently reconstructed for new requirements. Besides, modifying processes in these hard-coded systems for new requirements was time-consuming and costly. Some studies [4][14][15][16] proposed template-based paper-review systems, where research groups could build a paper review system for specific requirements by just setting templates. The reusable templates could reduce the constructing cost and improve reliability after frequently reusing. However, the paper-review processes were limited to these predefined templates.

Both traditional and template-based paper-review systems are independent each other. However, a lot of the functionalities of them are similar. If different research groups just hold their own process configuration and share the same functionalities in the paper-review system, the reusability can be improved.

On the other hand, since many functionalities in the paper-review system are shared each other, bugs can be detected and fixed rapidly compared with traditional independent paper review systems. Thus, the reliability can be improved.

Thus, this thesis proposed a reconfigurable paper review system, where different research groups could define their own paper-review processes and then automatically generate paper-review systems for their specific requirements. In order to satisfy various requirements of paper-review processes, constructing a reconfigurable paper review system was difficult to extract the implicit knowledge of paper-review process from the program logic of systems.

To specify our discussion, this thesis defined the term **"user"** as the user who defined her/his paper-review process in the system, and the term **"end-user"** as the user who used the output paper-review system.

This thesis used a knowledge-based approach to construct the reconfigurable paper-review system because this approach was suitable for modeling highly dynamic processes. Firstly, paper-review process configuration was acquired from a user, and the acquired knowledge was represented using a proposed model. The acquired paper-review process configuration could be modified in the future. The proposed model, named **process description model**, consisting of **frame-based knowledge representation** and **rule-based process control logic**, was the approach to modeling a paper-review process. Since actions, users' roles, and delivered files were all stereotyped in various paper-review processes, the frame-based representation was used to represent the actions and resources of these processes. Besides, these processes contain constraints and principles of managing the actions and resources, so rule-based representation was used to express the control logic.

The actions of end-users and the anonymity setting during paper reviewing vary across different paper-review processes. Thus, this thesis introduced the role-based access control model (RBAC) [7][8] to control the actions in the process and the anonymity between users.

Based on the **process description model**, a **paper-review process authoring tool** was provided to assist users in constructing desired paper-review processes. In order to evaluate the expressive power and modifiability of the proposed system, functions used in current existing paper-review processes and criteria of modifiability were collected. The comparison among the proposed system, traditional, and template-based paper-review systems was provided and the result showed that the proposed system could support the most functions and was the most modifiable among the compared systems.

In Chapter 2, some related works about the paper-review system construction and the authoring approaches are introduced. Chapter 3 describes the proposed paper-review process description model. Chapter 4 introduces the role-based access control model to solve the access control problem in the paper-review system. Chapter 5 introduces the paper-review process authoring tool and the system generating methodology. The implementation for the reconfigurable paper-review system generator and experiment are discussed in Chapter 6. Finally, Chapter 7 gives the conclusions and future works.

Chapter 2 Related Works

2.1 Paper-Review System Development

2.1.1 Traditional Paper-Review System

Arzu Baloglu [1] explicitly modeled an online submission process using a finite state diagram, as shown in Figure 2.1. Although this model could represent various paper-review processes, the study did not propose an approach to implement various online submission processes in a system because the modifiability of this study was not emphasized.

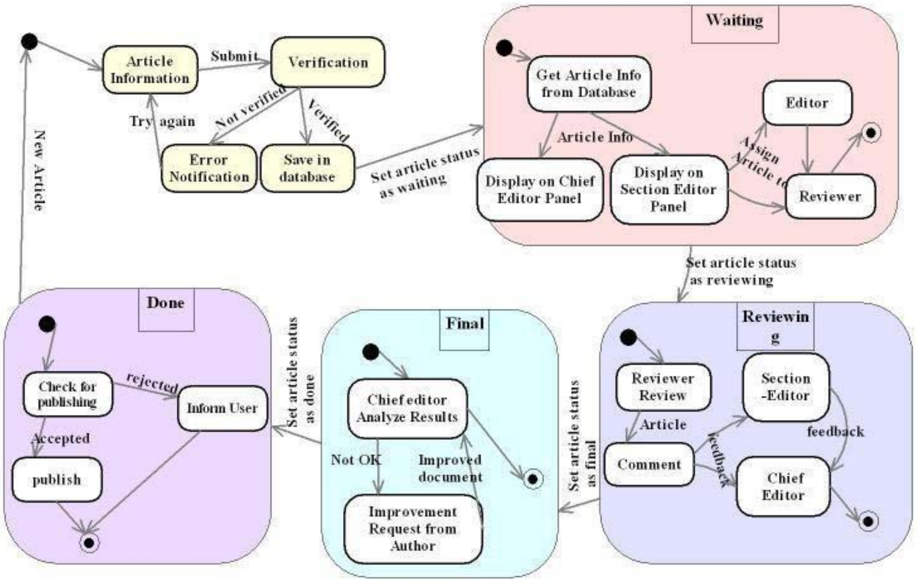


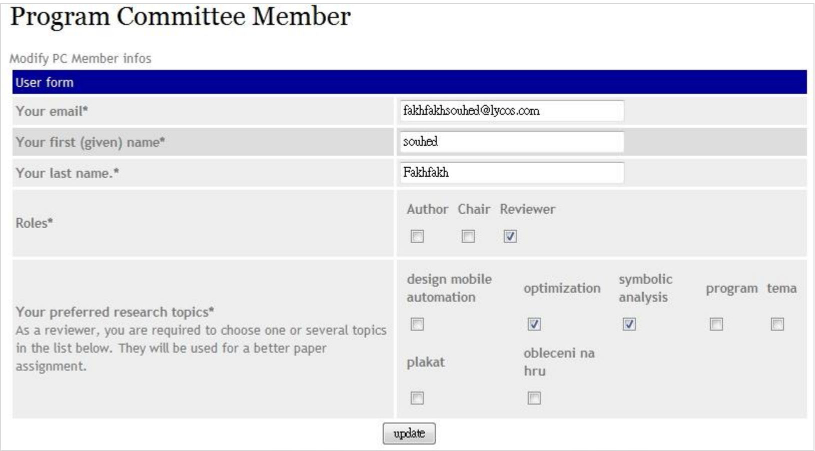
Figure 2.1: An example of article submission state diagram in COS

In recent years, many projects and studies about online conference/journal management system were proposed: IJDLT [13] was an online journal system, assisting author submitting papers. An editor of a journal could dispatch papers to reviewers, and after these reviewers reviewed the paper, the editor could decide that the papers should be accepted, rejected, or major/minor revised. The accepted paper was sent to the proof-reading process, and waited for publishing. Pradeep Gurunathan et al. [2] proposed an online conference system, which had a similar paper-review process except proof reading process. Additionally, the online conference system could manage conference sessions. Chun-I Fan et al. [3] proposed a truly anonymous paper submission and reviewing scheme has been proposed. Fan et al. [3] convinced that it should be totally anonymous under the paper submission and reviewing process among author, editor, and reviewer. The paper-review process in Anonymous was also similar to the above mentioned systems. However, these systems were developed for their specific paper-review process. The only way for modifying the process was modifying the code and that cost a lot of time.

2.1.2 Template-Based Paper-Review System

Some open-source projects developed template-based paper-review systems, providing templates for users to reconfigure to generate a paper-review system.

MyReview System [4][14] provided a configuration interface, as shown in Figure 2.2, for users to set configuration options of a conference, such as research topics, program committee, and reviewing criteria. Open Conference System (OCS) [15] and Open Journal System (OJS) [16] were similar to MyReview, but OJS was designed for managing journals. Moreover OJS and OCS supported more roles and more feature in conference/conference than MyReview.



Program Committee Member

Modify PC Member infos

User form	
Your email*	fakhfakhsouhed@lycos.com
Your first (given) name*	souhed
Your last name.*	Fakhfakh
Roles*	Author □ Chair □ Reviewer ☑
Your preferred research topics* As a reviewer, you are required to choose one or several topics in the list below. They will be used for a better paper assignment.	design mobile automation □ optimization ☑ symbolic analysis ☑ program □ tema □ plakat □ obleceni na hru □

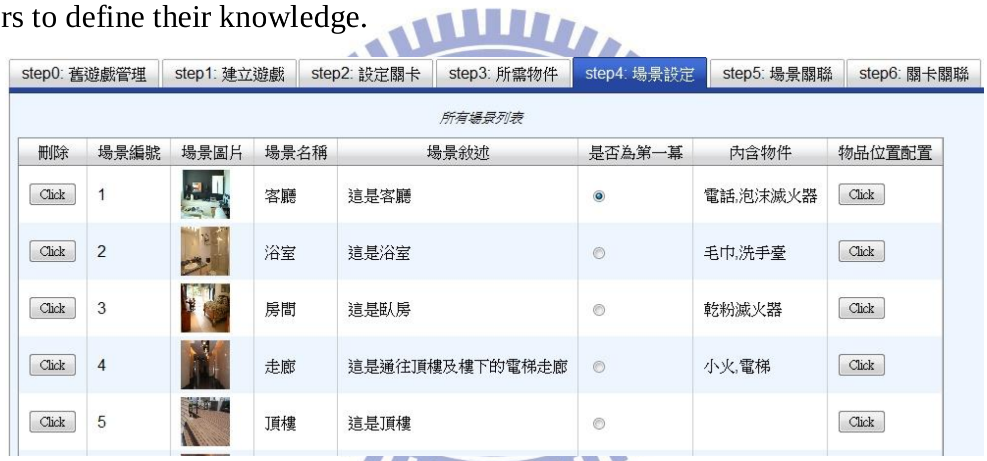
Figure 2.2: The program committee member setting interface of MyReview system

These projects could facilitate users to develop a paper-review system by only setting the detailed configuration of templates. Since the paper-review process was embedded in the predefined templates, a new paper-review process could not be implemented if no templates support this requirement.

2.2 Script Authoring Approach

Scripts, representing specific knowledge, were widely used to communicate between users and systems. Sharable Content Object Reference Model (SCORM) [17] provided the standards and the specifications for e-learning. It could facilitate users to define the content package of teaching materials or course sequencing behavior as XML script files. Wu [9] proposed a mechanism for teachers to design a role playing learning game by writing an XML script. Huang [10] proposed an adventure game-based formative assessment framework, which could facilitate teachers to construct an adventure game for assessment. The proposed system provided an AVG games authoring tool, as shown in Figure 2.3, to assist teachers in writing the XML-based adventure game scripts.

These researches indicated that the script was suitable for representing knowledge, and an authoring tool had highly interaction with users could facilitate users to define their knowledge.



所有場景列表

刪除	場景編號	場景圖片	場景名稱	場景敘述	是否為第一幕	內含物件	物品位置配置
Click	1		客廳	這是客廳	●	電話,泡沫滅火器	Click
Click	2		浴室	這是浴室	○	毛巾,洗手臺	Click
Click	3		房間	這是臥房	○	乾粉滅火器	Click
Click	4		走廊	這是通往頂樓及樓下的電梯走廊	○	小火,電梯	Click
Click	5		頂樓	這是頂樓	○		Click

Figure 2.3: the AVG games authoring tool based on adventure game script approach

Chapter 3 Paper-Review Process Description Model

The process of a paper-review system requires to be continuously modified because the requirements of conferences are changed over time. For example, Editor may change the category structure of paper, or add/delete the new attribute in papers, or define new paper states in the paper-review process. Thus, a knowledge-based approach was applied to develop a flexible and reliable system, where a papers-review process could be designed and revised without changing the programs of the system.

By the observation, a paper-review process was a series of actions performed by various roles to access or generate files, and these actions and generated files were subject to some constraints defined in the papers-review system. Because knowledge of actions, roles, and file formats were stereotyped, this thesis modeled the knowledge using frame-based representation. Besides, if-then rules were suitable to express the system's processes and actions' constraints. Thus, a paper-review system could be controlled using inference with these frames and rules. The knowledge represented as frames can be categorized in detail as the actions of end-users and the objects in the system such as Paper, User, and Journal etc. In addition, those if-then rules can be classified by their different purpose: controlling the process or determining whether the specific frames can be accessed by user, or just verifying the content of specific frames. In this chapter, we will introduce the various frames and rules in this thesis.

The system architecture was illustrated in Figure 3.1. This thesis split the system into two subsystems: **Configuration System** and **Output Paper-Review System**. Both the subsystems used the web-based environment as interface. This thesis proposed the **Paper-Review System Description Model**, the frame instances and rule instances representing the knowledge of the paper-review system. After this thesis defined the knowledge format, we proposed an **Interview-Based Authoring Tool**, where users could configure the paper-review system and store the process definition authoring script into script database. In the Output Paper-Review System, the **Script Transform Engine** transformed the authoring script into frame/rule instances and loaded into knowledge base. When end-users used the output paper-review system, the system accessed data in resource database and inferred the knowledge through inference engine. In this chapter, this thesis proposed the frame-based and rule-based approach to describe the paper-review system. The specific definition of frames/rules was declared in the next two sections. After these definitions, this thesis introduced an example of a scenario with the user submitted paper.

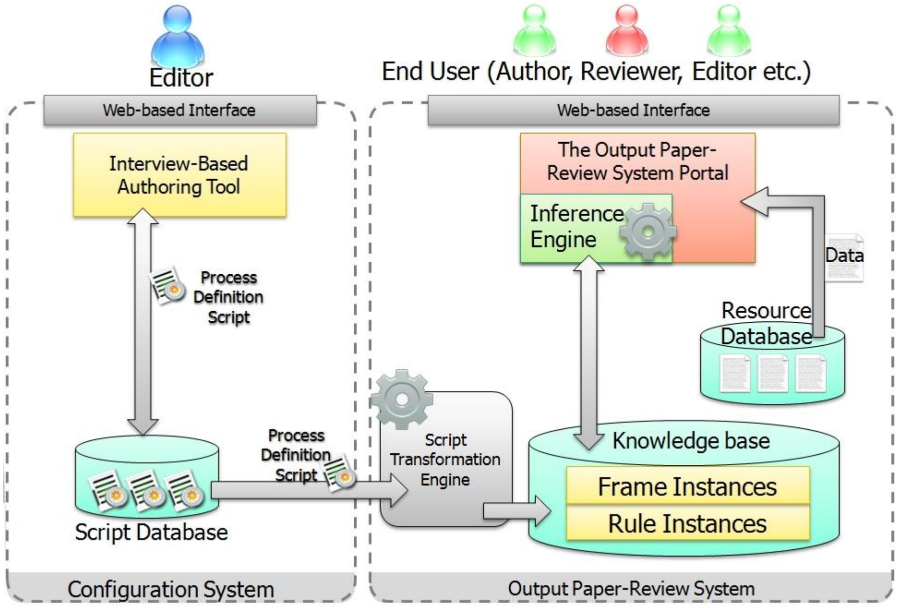

Editor
End User (Author, Reviewer, Editor etc.)
Web-based Interface
Web-based Interface
Interview-Based Authoring Tool
The Output Paper-Review System Portal
Inference Engine
Data
Process Definition Script
Resource Database
Process Definition Script
Script Transformation Engine
Knowledge base
Frame Instances
Rule Instances
Script Database
Configuration System
Output Paper-Review System

Figure 3.1: The architecture of a reconfigurable paper-review system generator

3.1 Using Frame Hierarchy to Represent the Paper-Review Process

Frame-based representation [5] is an appropriate hierarchical structure based upon object-oriented concept to describe stereotyped objects. A frame can contain slots, attached procedures, and rules. Each slot can have a default value and a slot value to represent the object's attribute. The attached procedures and rules, expressing the embedded logics among slots in single or multiple frames, can be triggered in the following common conditions:

if-need: the procedure is triggered while retrieving a slot value but a default value is not available.

if-added: the procedure is triggered while storing a value in a slot.

if-changed: the procedure is triggered while the slot value is changed

if-removed: the procedure is triggered while the slot value is removed

A frame can be inherited by other frames, which can have all slots contained by its parent frame and instanced to generate a frame instance. The relations of inheriting and instancing are named ***a-kind-of*** and ***is-a***, respectively.

This thesis used two kinds of frames, named **Resource Frame** and **Action Frame**, to represent the knowledge of used resources and the users' possible actions. A resource-frame hierarchy and action-frame hierarchy are shown in Figure 3.2(a) and 3.2(b) (c) (d) respectively. For Figure 3.2(a), all users were described by the *Role* frame hierarchy in this thesis. There were two *Roles* in the system: *User* and *Administrator*. The former is the end-user in the output paper-review system and the latter is the one who has the privilege to configure their paper-review process. The *User* in the output paper-review system consisted of *Editor, Reviewer*, and *Author*. For the *File* frame hierarchy, it can be divided into different types such as *Paper* frames, the *Paper Attachment File* frame, etc. according to their different purposes.

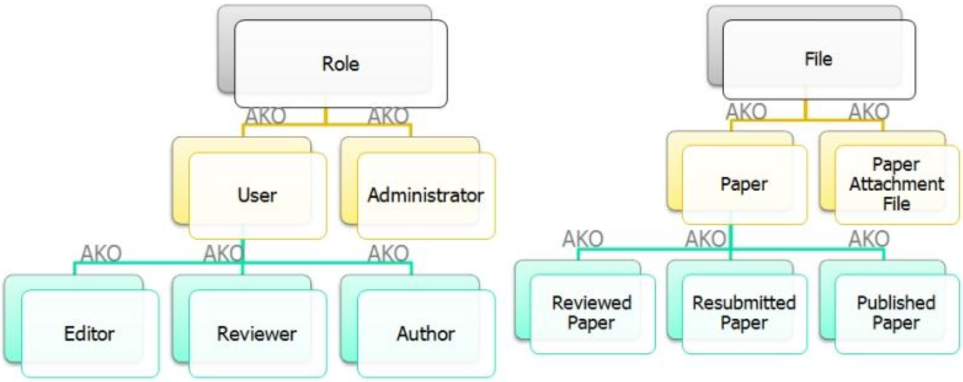


Figure 3.2(a): Role and File Resource Frame Hierarchy

Different with resource structure, the action frame hierarchy defined the actions might be performed by end-users in the output paper-review system. Each leaf node of this hierarchy might be a kind of its parent action frame. The action frame contained the configuration information of the action. It might have some default value or some procedure attachment. When user configure his/her paper-review system, action frames' slot values would change correspondingly. For example, when user configured the New Submit action, she/he would define the information which end-user should provide. Moreover, when user configured the Review action, she/he should define the criteria for evaluating papers.

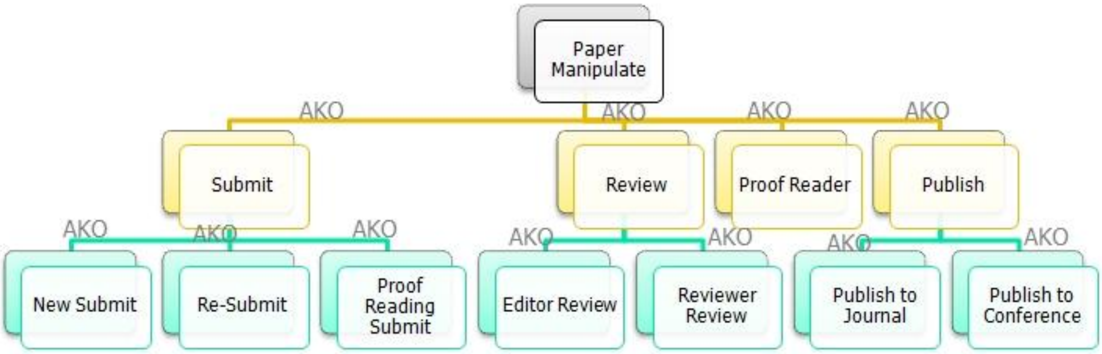


Figure 3.2(b): Paper Manipulate Action Frame Hierarchy

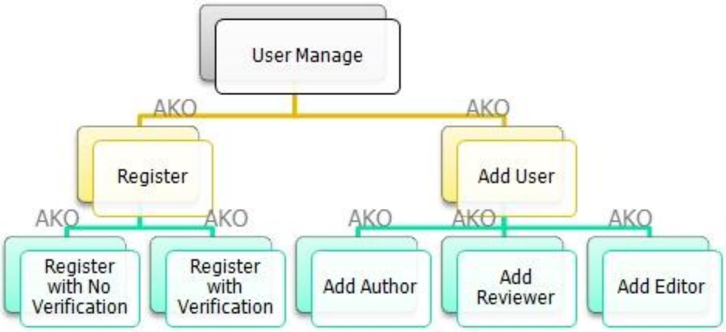


Figure 3.2(c): User Management Action Frame Hierarchy

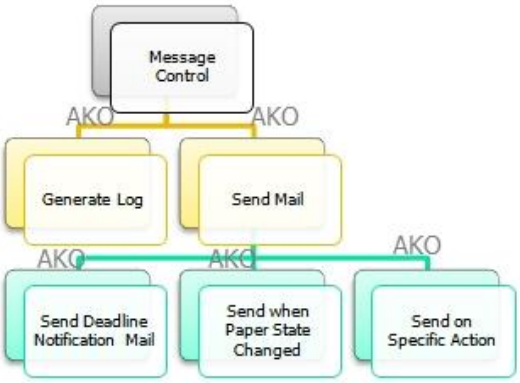


Figure 3.2(d): Message Controlling Action Frame Hierarchy

The definitions of resource frames and action frames, which would be used when we illustrated the action control management approach in Chapter 4 and defined our knowledge acquisition model in Chapter 5, were stated as follows

Definition 1: The **Resource Frame** is a 5-tuple: RF = (FN, FR, S, VR, DL) where

(a) FN: The name of this resource frame

(b) FR= (T, FN): The relation between this resource frame with other resource frame, where T is the type of relation. It may be **a-kind-of** relation which denoted the inheritance relation and **is-a** relation which denoted instantiation relation.

(c) S={(SN_i, SV_i, ST_i, DV_i, PA_i, FA_i) |for all i, 1<= i < n}: The finite set of slot information in this resource frame. It is a 6-tuple, where

- SN_i: The name of the i-th slot. It stands for the attribute name of this resource
- SV_i: The value of the i-th slot. It stands for the attribute value of this resource
- ST_i: The data type of the i-th slot. It can be string, integer, or float

number.

- DV_i: The default value of the i-th slot. If it is non-empty, the SV_i is assigned when new resource frame instance is generated.
- $PA_i = \{(T_j, P_j, PR_j) \mid \text{for all } j, 0 <= j < 4\}$: The finite set of procedure attachments of the i-th slot, where T_j is the type of this procedure attachment. It may be **if-need**, **if-changed**, **if-added**, or **if-removed**. PR_j is the set of parameters of this attached procedure.
- FA_i: The finite set of frame attachment of the i-th slot. When it is not null, it represent that this slot's value will be a set of other frames

(d) VR: The finite set of resource verification rules. When system tried to add a resource, it would trigger resource verification rules to check this new resource is valid or not. The detail definition of resource verification rules would state in Section 3.2

(e) DL: The link to the specific database table, where stored the data generated with this resource frame definition. For the leaf frame in the resource frame hierarchy, DL will not be null.

Definition 2: The **Action Frame** is a 4-tuple: AF = (FN, FR, S, FC) where

(a) FN: The name of this action frame

(b) FR=(T, FN): The relation between this action frame with other action frame, where T is the type of relation. It may be **a-kind-of** relation which denoted the inheritance relation and **is-a** relation which denoted instantiation relation.

(c) $S=\{(SN_i, SV_i, ST_i, DV_i, PA_i) \mid \text{for all } i, 1 <= i < n\}$: The finite set of slot information in this action frame. It is a 6-tuple, where

- SN_i : The name of the i-th slot. It stands for the configuration item of this action frame.
- SV_i : The value of the i-th slot. It stands for the configuration content of this action frame.
- ST_i : The data type of the i-th slot. It can be string, integer, or float number.
- DV_i : The default value of the i-th slot. If it is non-empty, the SV_i is assigned when new action frame instance is generated. For all DV_i in S, it stands for a stereotype about this action.
- $PA_i = \{(T_j, P_j, PR_j) \mid \text{for all } j, 0 <= j < 4\}$: The finite set of procedure attachments of the i-th slot, where T_j is the type of this procedure attachment. It may be **if-need**, **if-changed**, **if-added**, or **if-removed**. PR_j is the set of parameter of this attached procedure.

(d) FC : The finite set of facts that would be triggered after this action frame is called.

The difference of physical meaning between resource frames and action frames is that end-users may generate the data based on the structure of resource frame, and the inference engine may infer rules after end-users performed the action defined by action frame.

Example 3.1: The resource frame and action frame

Two examples of the usage of resource frames and action frames were shown in Figure 3.3(a) and Figure 3.3(b), respectively. In Figure 3.3(a), in addition to the slots in *File* frame, the *Paper* frame consisted the information about the papers. A *Reviewed Paper* had the information of reviewer and review result. In Figure 3.3(b), when a reviewer reviewed paper, the reviewer could suggest the paper state of judge result and sent the review result back to editor.

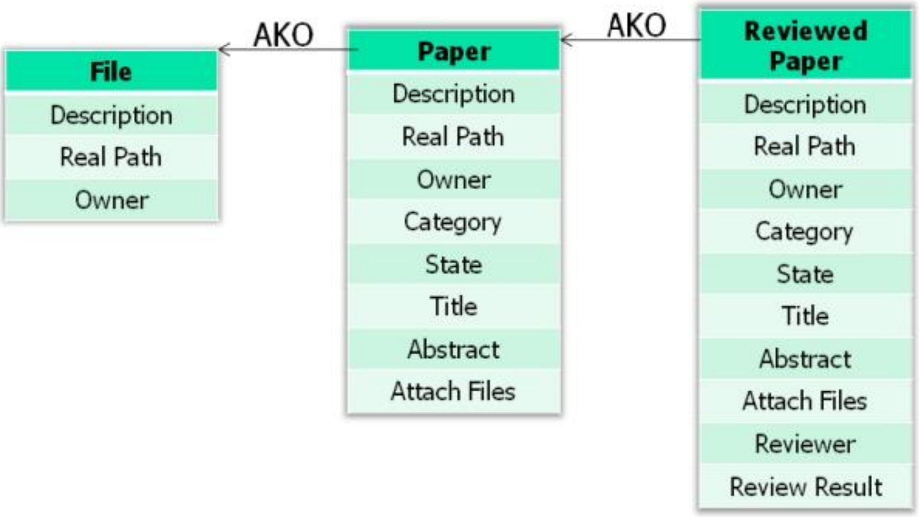


Figure 3.3(a): The “File-Paper-ReviewPaper” resource frame hierarchy

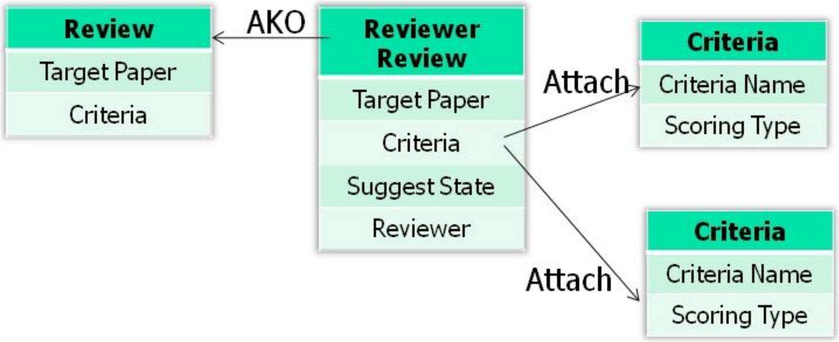


Figure 3.3(b): The “Review-ReviewerReview” action frame hierarchy

3.2 Using Rule to Control the Paper-Review Process

Rule-based representation [6] is a kind of knowledge representation, used to express cause-effect relations and reasoning logic. In the proposed paper-review system, this representation was used to represent process control logic, anonymity principle, and authentication rules to facilitate modification for frequently changeable requirements of research groups.

By the observation of existing paper-review system, rules used in paper systems could be classified into *Process Control Rules*, *Authentication Rules*, and *Data Verification Rules* based on the purpose of rules.

(a) Process Control Rules

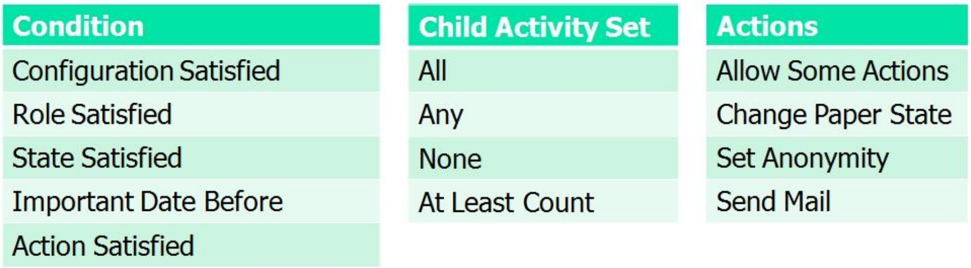
Process control rules determine the paper-review process, which can be modified by editors.

A process control rule may have preconditions, which were classified into four types: (i) Configuration Satisfied: the rule could be fired if a specific setting was in the action frame; (ii) Role Satisfied: the rule could be fired for specific roles; (iii) Paper State Satisfied: the rule could be fired if the targeted paper was in specific paper states; (iv) Important Date Before: the rule could be fired in specific dates. For example, the resubmission deadline may be a month later when the paper state was "*revise*". The important date could be a real date, such as the paper submission deadline was 2010/09/30; (v) Action Satisfied: the rule could be fired after some actions performed and generated the fact in the FC-tuple of action frame (see Definition 2).

If the preconditions were satisfied, four kinds of actions could be triggered: the action allowing the execution of specific actions; the action setting specific paper states; the action setting the anonymity under specific action; and the action sending messages by E-mail. The structure of process control rules was shown in Figure 3.4.

Example 3.2: Process Control Rule

If the end-user's role was *Reviewer,* and there was any *Reviewer Review* action frame which the *Reviewer* slot (See Figure 3.3(b)) pointed to this end-user then she/he could review paper.



Condition	Child Activity Set	Actions
Configuration Satisfied	All	Allow Some Actions
Role Satisfied	Any	Change Paper State
State Satisfied	None	Set Anonymity
Important Date Before	At Least Count	Send Mail
Action Satisfied		

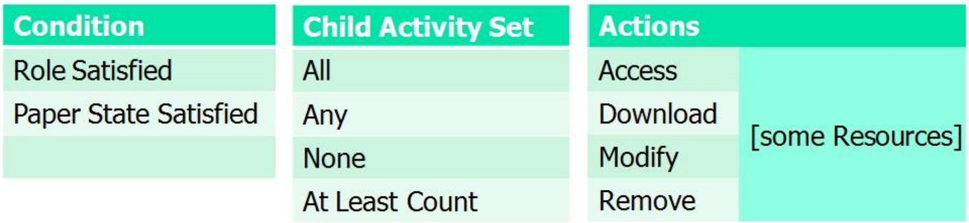
Figure 3.4: The Structure of Process Control Rules

(b) Authentication Rules

Files in a paper-review process had various accessing permissions for roles. For example, a paper under reviewing could only be accessed by a reviewer and an editor. In the system, authentication rules, defined by editors, could control these accessing principles. As shown in Figure 3.5, preconditions of an authentication rule had three types: (i) Role Satisfied: the access was permitted for specific roles; (ii) Paper State Satisfied: the access of a paper was permitted if the paper was in specific states (the *State* slot in Figure 3.3(a) Paper Frame). The permitted actions were accessing, downloading, modifying, or removing a specific resource.

Example 3.3: Authentication Rule

If the end-user's role was *Editor,* or the *State* slot of *Paper* frame was *Public* then this end-user could access this *Paper* frame.



Condition	Child Activity Set	Actions	
Role Satisfied	All	Access	[some Resources]
Paper State Satisfied	Any	Download	
	None	Modify	
	At Least Count	Remove	

Figure 3.5: The Structure of Authentication Rules

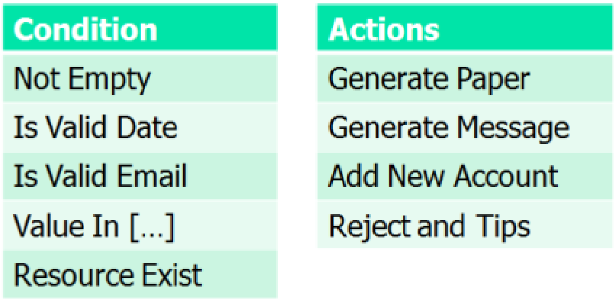
(c) Data Verification Rules

The user-generated data, such as information of a new paper or a new account were required to be verified to prevent wrong inputs. Thus, data verification rules, as shown in Figure 3.6, were defined to verify user-generated data in resource frames' slots. The preconditions of these rules had four types: (i) Not Empty: the input was allowed if a specific value was not empty; (ii) Is Valid Date: the input date was allowed if it had a correct format or in the valid duration; (iii) Is Valid Email: the input

E-mail was allowed if the E-mail's format is correct; (iv) Value in […]: the input data was allowed if it was in a range which defined by user; (v) Resource Exist: the input data was allowed if the referred resources were exist. If the input data was valid, these rules could allow user to generate paper, generate a message, or add a new account. Otherwise, data verification rules could reject inputs and give tips to users.

Example 3.4: Data Verification Rule

If the end-user's role was *Editor,* or the *State* slot of *Paper* frame was *Public* then this end-user could access this *Paper* frame.



Condition	Actions
Not Empty	Generate Paper
Is Valid Date	Generate Message
Is Valid Email	Add New Account
Value In […]	Reject and Tips
Resource Exist	

Figure 3.6: The Structure of Data Verification Rules

In the end of this chapter, this thesis introduced an example of a scenario with the author submitted paper. The scenario was shown in Figure 3.7.

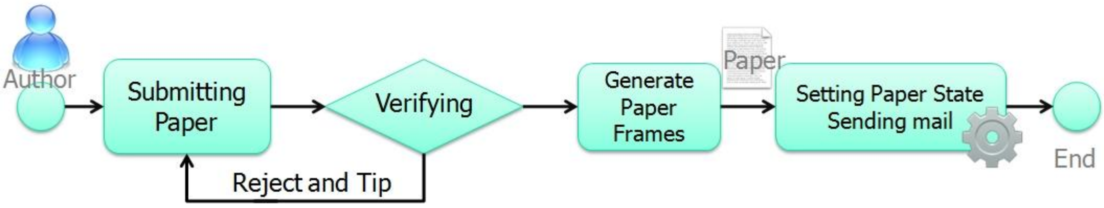


Figure 3.7: The scenario with author submitting paper in a paper-review system

When the author started to submit paper, inference engine got the configuration in *New Submit Paper* frame (action frame) and the structure of *Paper* frame (resource frame). Author provided the information of paper such as *Title*, *Abstract*, *Category*, etc. (see Figure 3.3(a) *Paper* frame).

When author submitted, inference engine inference the data verification rules defined in *Paper* frame to check if the content of new *Paper* frame was valid. If not, rejected this submission and tipped to author, otherwise, generate new instance of *Paper* frame (data verification rule).

After frame was generated, inference engine inferred the new state of the paper, and mails needed to be sent through the process control rules. In this example, the paper state might be set as "wait-for-review", a mail sent to author for keeping track of his paper after submitting, another mail sent to editor as a notification after the paper state set as "wait-for-review". After submitted, the author could access this paper, but another end-user was not editor or an author of the paper could not (authentication rules).

Through the user logging in and submitting scenario, this thesis presented a typical example of rules inference (verify paper→new paper state→notify editor). Of course, different research group might have different frames/rules setting. They could generate another process of submitting paper by modifying the frames/rules.

Chapter 4 Roles Setting with Role-Based Access Control Model

Humans' operation in this process management system is an important issue especially in such a process reconfigurable system. There are different users in the paper review system. Based on different responsibility in the paper review process, users may have different actions allow doing. For example, a reviewer is responsible for reviewing paper. Thus, an author is not allowed to review a paper. Similarly, a reviewer may have no right to publish the accepted papers, which is the responsibility of editors. Previous example is just a simple example, in some conference or journal workflow, the division of labor is more complicated.

For the situation stated before, if we solve it by the traditional programming approach. We can build up a series of "user group", and assign the corresponding action to each group, then allocate users into user group. When some users are allowed to do another specific action, we can use another special case decision mechanism in our programming to support. If the process have less variability, the traditional programming mechanism is useful enough. However, the paper-submit/review process is changed with time. Different research group may have different review process. For example: for a conference A, the chief editor is originally responsible for the dispatching, proof reading, and publishing of the paper. However, the more extension of the conference, the more complicated division of labor is. Chief editor may need to delegate the responsibility of paper proof reading. Thus, it often extracts the access control management by the knowledge-based approach for the modern paper review system. It can enable the user to manage the actions performed by end-users without altering the source code. That is a knowledge-based solution.

Nevertheless, there is another issue should be confronted with. Under the following situation: In a paper review process of someone conference, User A and User B are responsible for review paper. However, user A is additionally allowed to publish paper. For the knowledge-based approach, we create a "Reviewer User Group" which have the right of reviewing paper, and then set User A and User B as a member of "Reviewer User Group". Now, User A and User B can review paper. Next, we make another configuration, let User A have the right of publishing paper. However, if there are many end-users who need the additional assignment of access control, it may cost a lot of time to reset the access control of each end-user when we want to withdraw the right of publishing paper in the future.

For this reason, we need to manage the end-users and the actions they can perform more systematic. In this chapter, we introduced the Role-Based Access Control (RBAC) approach into our paper submit/review process. RBAC model can make the management of user action in a workflow process simpler.

Before introducing the RBAC model in our paper review process, we first defined the roles characteristic, relation between roles, and the actions based on those relation. In Section 4.2 we discussed how to use RBAC model in a paper review system. In the end of this chapter, we extended RBAC model to manage the anonymity setting in paper submit review process.

4.1 Role Characteristic and Relation

4.1.1 Roles in a Paper Review System

According to the actions user doing in the paper review system, we can divide several user types and the characteristic as follows:

a. Editor: Editor is mainly the manager of conference/journal. She/he is responsible for the establishment of conference/journal, call-for-papers, and publishing papers. Thus, she/he is an influential user of conference/journal. In this thesis, editor can make configuration of the paper review process. (Of course, we suggest that the "process editor" of our reconfigurable paper review system is limited to 1 or 2 specific users, or the paper review process may be chaotic under the condition that different user alter the paper review process) In some conference/journal, a "co-editor" may be added to support editor to manage the conference/journal. Co-editor's characteristic is almost like editor's but co-editor is added by editor and she/he cannot add another co-editor.

b. Reviewer: Reviewer is often invited by editor. Reviewer is responsible for the reviewing task of manuscripts. Editor often assign the reviewing job to appropriate reviewers based on the related research area of the manuscript. According to the reviewing result of reviewers, editor judges the manuscript is accepted to publish or not. Reviewer's characteristic in the paper review system is relatively simple because that she/he need only review the manuscripts editor assigned. Some conference/journal system may combine the reviewer's responsibility into editor's. That is, editor is also responsible for reviewing paper. Some conference/journal make reviewer add other "co-reviewers" to supporting the review task.

c. Author: Author is often the source of the manuscripts. She/he is mainly responsible for submitting papers and performing corresponding actions depend on the judge result of manuscript. For example: a reviewer resubmits paper when the judge result of the manuscript is “revise”, or enter the proof reading process when the paper is accepted to publish. Authors often have no relation with editor/reviewer. In general, author can be added by registration in the conference/journal system. Of course, there are some paper review systems may have a verification process after author’s registration.

In general, the role type definition is often fit in the previous types. Their characteristic can be classified into: (a) stand for the conference/journal (editor); (b)keep the quality of the manuscript of the conference/journal (reviewer); (c) the end-user who has no relation with the conference/journal (author). The grading order of interaction with the paper review system is editor > author > reviewer. Editor is responsible for communicating with reviewer and author in addition. Thus, the role definition may divide more complicate (such as: Assistant editor). Nevertheless, the characteristic of roles are still like these three role.

4.1.2 Relations between Roles

As stated before, there are different interaction relations between different roles. We can define the following role relation:

a. Editor-Reviewer Relation: this relation is mainly based on a paper under review state. Based on this relation, editor can dispatch a review job to reviewer; Reviewer can response the review job to editor.

b. Editor-Author Relation: We can define this relation as an end-user interacting with conference/journal author. The paper submission/ judgment are based on this relation.

c. Editor-Editor Relation: Editor may have another interaction with another editor or co-editor. That can make the operation of conference/journal more effective.

d. Reviewer-Reviewer Relation: Different reviewer may have some interaction each other when reviewing paper.

There are corresponding actions on the relation between roles. We summarize them in Figure 4.1. In addition, the actions list in Figure 4.1 is the actions based on different roles each other. There are other actions that doesn’t base on different roles such as publishing papers. Thus, in this thesis, we also defined these actions and store them in knowledge base as action frames which are defined in Chapter 3.1.

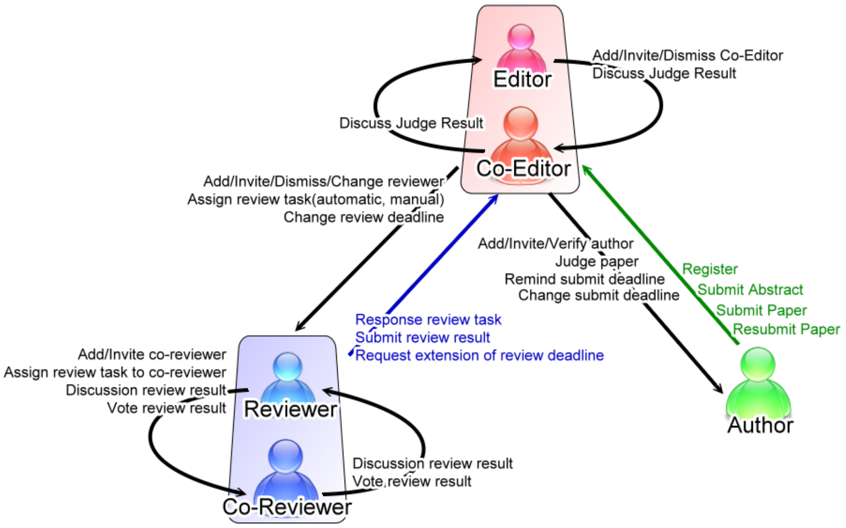


Figure 4.1: Role relations and corresponding actions

4.2 Role-Based Access Control Model in Paper Review System

4.2.1 RBAC Model Introduction

The general-purpose role based access control model was proposed in 1992 by David Ferraiolo and Rick Kuhn [7]. In 1996 Sandhu, Coyne, Feinstein, and Youman introduced a framework for RBAC models [8]. This model is commonly used as a mechanism of access control in the information security domain. In traditional information systems, every user may be assigned different access control right. Under RBAC model, the access control right is not assigned to the user. Instead, the access control right is assigned to the roles defined by the system, then binding these roles to different users. At this time, the user accesses some actions by the roles bound on her/him rather than her/his own. For an information system, the user base will grow up over time, but the roles in the system are relatively stable. Thus, using RBAC model to manage the access control right is flexible and easy to manage than the traditional Mandatory Access Control (MAC) and Discretionary Access Control (DAC)

In this thesis, there are many types of users involved in the paper submit/ review process. We've classified these users as different type of roles and characteristic. We've also summarized their relations and the actions based on those relations. In consequence, we must encounter the issue of access control in our paper review system. So we need a mechanism to manage the access control. After the survey of RBAC model, we decided to introduce this model which is commonly used in information security domain into this thesis.

4.2.2 Action Management under RBAC Model

In this thesis, we used the frames and rules to make knowledge representation of the whole paper review system. The advantage of frames incorporating with rules is stated in Chapter 3 (flexible, convenient when we need replace the knowledge). Thus, we implemented the RBAC model into our reconfigurable paper review system by frames and rules.

In the Configuration System, system manager defined the roles which would be used in the output paper review system of their conference/ journal. That is, by using our authoring tool to generate the role frames defined in Chapter 3. Every role frame was assigned the action that allowed doing. After the role frame definition, the role frames were attached into the user frame under the generation of user frame. Figure 4.2 was an example of RBAC model in the paper review system.

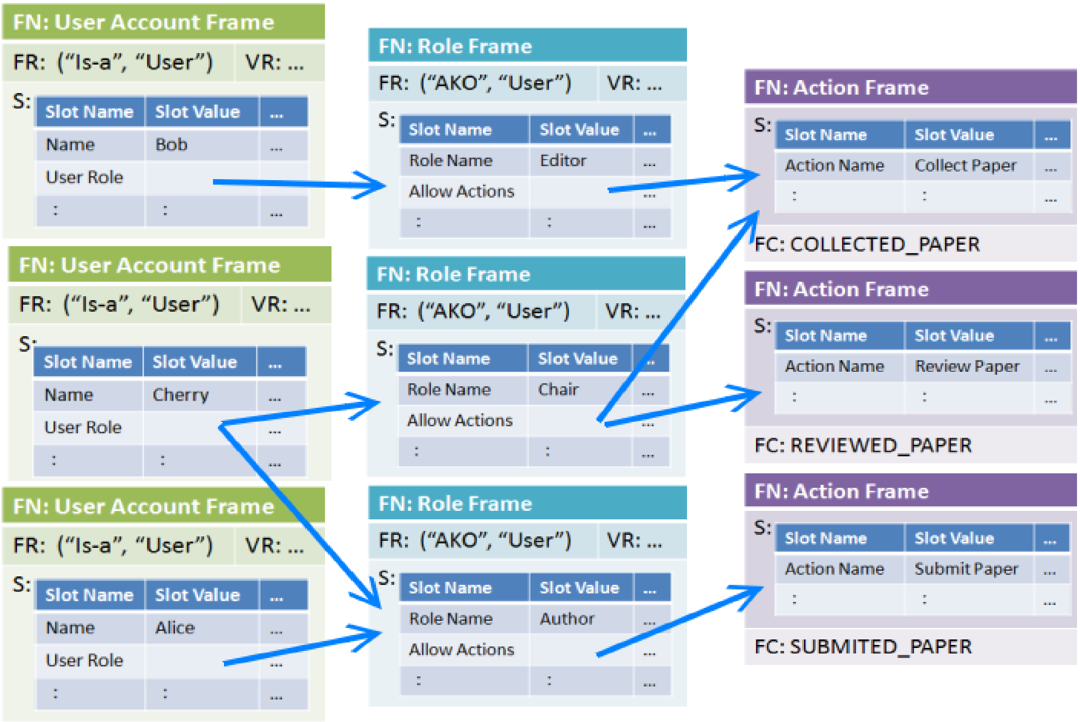

FN: User Account Frame
FR: ("Is-a", "User") VR: ...
S:
Slot Name | Slot Value | ...
Name | Bob | ...
User Role
FN: User Account Frame
FR: ("Is-a", "User") VR: ...
S:
Slot Name | Slot Value | ...
Name | Cherry | ...
User Role
FN: User Account Frame
FR: ("Is-a", "User") VR: ...
S:
Slot Name | Slot Value | ...
Name | Alice | ...
User Role
FN: Role Frame
FR: ("AKO", "User") VR: ...
S:
Slot Name | Slot Value | ...
Role Name | Editor | ...
Allow Actions
FN: Role Frame
FR: ("AKO", "User") VR: ...
S:
Slot Name | Slot Value | ...
Role Name | Chair | ...
Allow Actions
FN: Role Frame
FR: ("AKO", "User") VR: ...
S:
Slot Name | Slot Value | ...
Role Name | Author | ...
Allow Actions
FN: Action Frame
S:
Slot Name | Slot Value | ...
Action Name | Collect Paper | ...
FC: COLLECTED_PAPER
FN: Action Frame
S:
Slot Name | Slot Value | ...
Action Name | Review Paper | ...
FC: REVIEWED_PAPER
FN: Action Frame
S:
Slot Name | Slot Value | ...
Action Name | Submit Paper | ...
FC: SUBMITED_PAPER

Figure 4.2: The RBAC model used in paper review system

Under the RBAC model, every user in the paper review system may be assigned one or more role. When a user attempts to access someone action frame, system will trigger the process control rules which defined in Chapter 3.2 to check if the user has the right to do that action. If yes, then system allows the user doing that action. Otherwise, system notifies the user that she/he is not allowed doing that action.

Using RBAC model, access control management becomes more convenient. However, in a reconfigurable paper review system, the paper review process was defined by user and stored in knowledge base as different process control rules. At this time, there was another issue we confronted. Because that a user may be more than one roles in a paper review system. And the actions allow to each role which user may have an intersection. (For instance, in Figure 4.2, both Role "Chair" and Role "editor" have the right to "publish paper") However, different role doing the same action may have the different inference result. For example: User A is an author in a conference, she/he can submit paper. However User A has an excellent contribution under a specific domain. In consequence, the session chief of the sub-conference under the conference assigned him a role "invited speaker". When this role submits paper, it will be accepted directly. Based on this example, we can conclude that when User A submitting a paper, different roles may confront to different paper review process.

Consequently, when a user does some actions in our reconfigurable paper review system, it is necessary to keep track of what is the role of the user. It can avoid the ambiguity inference result when rule inference. Nevertheless, we still don't expect that bugging end-user to check the role she/he is when she/he doing an action. Thus, we used a "Control Panel" to make user change her/his role freely. When the user open the control panel of Role A, the action she/he do will be seemed as Role A. Similarly, when she/he opens the control panel of Role B, she/he will change his role to Role B. In consequence, the system can have a correct rule inference.

4.3 Anonymity Control Based On RBAC Model

4.3.1 Anonymity in a Paper Review Process

In the process of paper submission, to take fairness into account, anonymous reviewing mechanism is emphasized in some conference/journal. No matter the chief editor or reviewer an end-user is, there is no influence about the judge result by the author's identity. The previously research [3] also shows that anonymous Reviewing helps fairness of paper review, and the openly stated criticism might have some influences upon the reviewers' careers. So the anonymity issue should be taken into account.

Although the anonymity reviewing is important, different research group may have different levels of anonymity requirement. For example: Conference A claims that reviewer is anonymous with author. Event the reviewer's comment about the manuscript is not opened to author; however, conference B claims that reviewer is anonymous with author, but the reviewer's comment about the manuscript is accessible by author. For the different anonymity control, we have to propose a management approach.

Before propose the anonymity control management approach, we can summarize the section need to keep anonymous in the paper review process based on Figure 4.1. They are follows (the role pair (A➔B) in each section stands for that "Should Role A's identity be known by Role B?"):

a. Paper submission (Author➔Editor): When submitting paper, it may have an influence on editor about the decision making when dispatching paper. That is, dispatching the paper to the reviewer who review paper loosely, to rise up the accept rate of this paper; or when judging the paper, the judge result may be influenced by the editor's preference for the author.

b. Paper dispatching (Reviewer➔Editor): Editor should dispatch the submitted paper based on the correlation between the content of paper and reviewer's research domain. It seems like that there is no need to keep anonymous between reviewer and editor in this section. However a research [3] showed that *because the editor knows the relationship between the reviewers and their comments on a paper. She/he is able to convince the author that someone has reviewed the paper. And that may cause the privacy leakage problem.* Therefore, we assume that there is a conference/journal would like to keep the anonymity in the paper dispatching section.

c. Responding the paper reviewing job (Author➔Reviewer): When reviewer receiving the commission of reviewing paper, he/she should decide review this paper or not based on the content of the paper and the workload on herself/ himself. However, if review knows the author's identity, reviewer may take the relation between her/him and the author in consideration of review this paper or not. For example: if the author used to be an advisor of the reviewer, the reviewer may refuse this reviewing job because she/he doesn't like to offend on the author.

d. Reviewing paper(Author➔Reviewer): Same as c., When reviewer accepted the review job assignment, the reviewer's comment may be influenced by the relation between her/him and the author.

e. Judged Paper(Reviewer➔Author): After the author receiving the paper judge result, editor will have appropriate feedback on the paper. Moreover, reviewer may have some feedback, too. Since an editor always stands for the conference/journal, there is no need to maintain the anonymity between editor and author. However, if there is no anonymity between reviewer and author, it may have negative influence between them if the reviewer has a negative rating about the paper.

We have summarized the sections which need to care the anonymity in the paper review process. For different research groups, they may have different anonymity setting on his conference/journal paper review process. They may think that not all sections stated before should be kept anonymous. In consequence, we proposed a management approach, which extended from RBAC model introduced in Chapter 4.2. Because that we found that based on RBAC model, the anonymity setting can be managed conveniently.

4.3.2 Using RBAC Model to Control the Anonymity

This thesis used the RBAC model to manage the anonymity of the whole paper submit/review process. At first, we let system manager define the anonymity between roles when she/he is defining the role frames. After defining the anonymity between roles, system can generate a 2-dimension anonymity setting matrix like Table 4.1. The Matrix [i][j] stands for the user$_i$ is anonymous with user$_j$ or not. For example: Matrix [1][2] is Anonymous. It stands for the user information of Author should be hidden with Editor.

Table 4.1: Anonymity Setting Matrix

	Author	**Editor**	**Reviewer**	**Administrator**
Author	Public	Anonymous	Anonymous	Public
Editor	Public	Public	Public	Public
Reviewer	Public	Public	Anonymous	Public
Administrator	Public	Public	Public	Public

We can notice that the anonymity setting matrix in Table 4.1 is not a symmetric matrix because that the anonymity relation is not an equivalence relation, author is anonymous with editor doesn’t stand for that editor should keep anonymous with author.

After generating the anonymity setting matrix, we should let system manager define the “Public Part” and “Private Part” in the section that need the anonymity control (summarize in Chapter 4.3.1). For example, some information should be hidden when paper submission. System manager should declare what’s the private information and what’s public information. After the declaration, the preparation of anonymity control based on RBAC is done. When end-user submitting paper, in the process of the interaction with other user, the system may decide to show the private information or not based on the both end users’ role relation in the anonymity setting matrix. In Figure 4.3(a), Role 1 is anonymous with another Role 1, If User A becomes Role 1, and has some interactive with User C, who is another Role 1. User A should be anonymous with User C, so the private information should be hidden from User C. However, in Figure 4.3(b), Role 2 is public with Role 1. When User A becomes Role 2, and have some interactive with User C, who is Role 1, User A should not be anonymous with User C, so the private information should be accessed by User C.

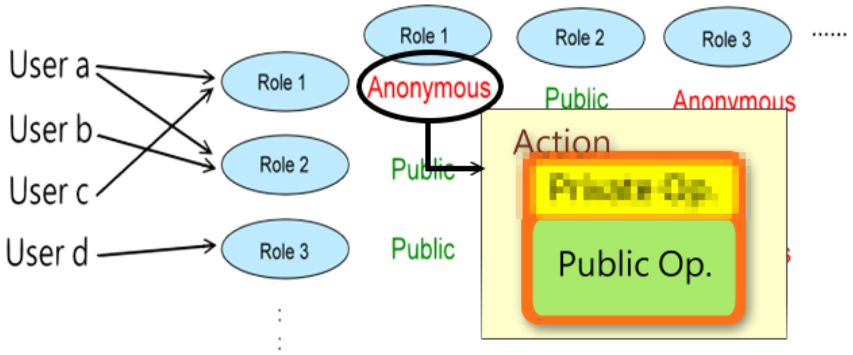


Figure 4.3(a): The anonymous setting between Role 1 and another Role 1

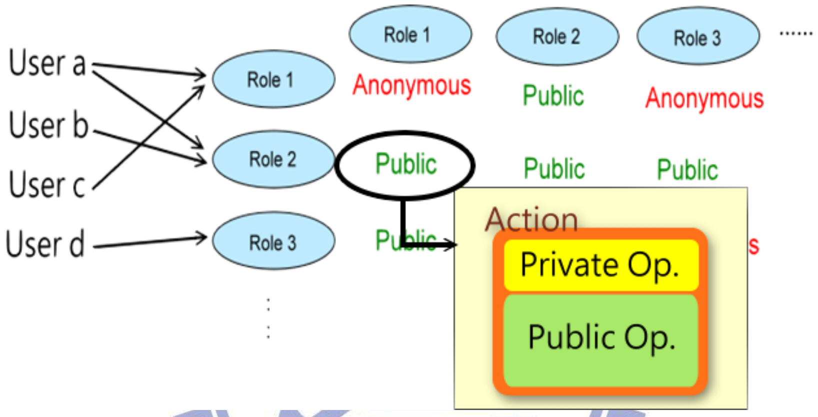


Figure 4.3(b): The anonymous setting between Role 2 and Role 1

From Figure 4.3(a) and Figure 4.3(b), we can notice that even if the interaction between the same user pair, under the RBAC model, if the role of user is different, it may cause different anonymity control result, and that is just one of the characteristic of RBAC.

Chapter 5 Requirement Acquisition and System Generating Method

This thesis defined several types of frame and rule and then used them to represent the paper-review process in Chapter 3. Chapter 4 introduced the actions access control and the anonymity management in the interaction between different roles. In this chapter, this thesis presented an authoring approach that facilitated user constructing her/his paper-review process. With the interaction mechanism between user and authoring tool, the construction time cost could be reduced.

Thus, this thesis proposed a friendly configuration interface for user. User could edit the paper-review process thought this interface and watch the corresponding change immediately. She/he could load the configuration and edit it again and again. The construction process was a kind of rapid prototyping. The interaction between the authoring tool and user was emphasized.

As shown in Figure 5.1, the requirement acquisition method was proposed. First, this thesis illustrated the architecture of the authoring tool, the interaction between user and authoring tool, and the operation of frames and rules during user's configuration. In Chapter 5.2, in addition to summarizing how to convert the user configuration to process authoring script, this thesis also explained that how to generate the paper-review system by the authoring script. Finally, this thesis discussed some problems might occur in the output paper-review system, and brought up the solutions of them.

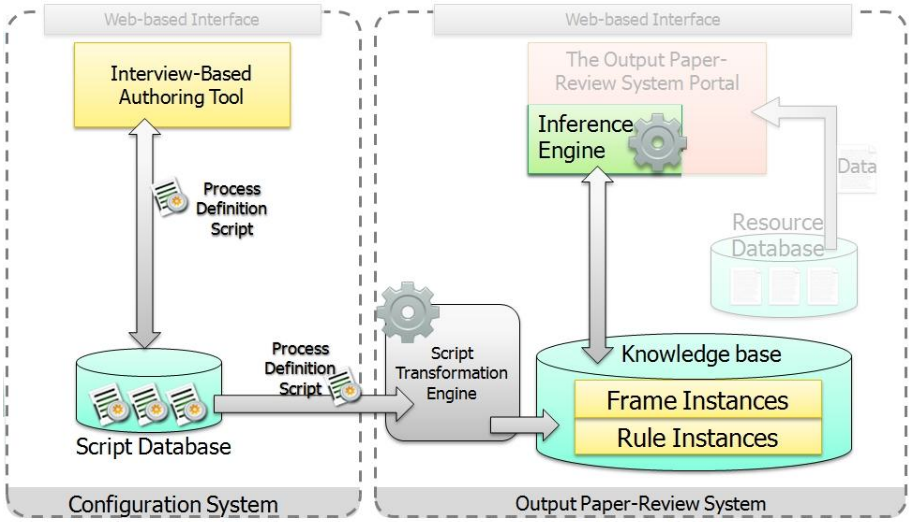

Web-based Interface
Interview-Based Authoring Tool
Process Definition Script
Script Database
Process Definition Script
Configuration System
Web-based Interface
The Output Paper-Review System Portal
Inference Engine
Data
Resource Database
Script Transformation Engine
Knowledge base
Frame Instances
Rule Instances
Output Paper-Review System

Figure 5.1: The detailed architecture of requirement acquisition and system generating model

5.1 An Interview-Based Authoring Tool

5.1.1 Overview about the Interview-Based Authoring Tool

As stated in the architecture in Chapter 3, the authoring tool was a web-based application. User could just launch the browser and connect to the paper-review process authoring tool. By the observation, it could split the process of construction into 7 phases (These phases were illustrated in Chapter 5.2 one by one). By the "step by step" operation environment, this thesis facilitated user to complete the process construction work.

In order to collect the configuration in each phase set by user, several interview-based questions were proposed. Through the process of answering questions, the authoring tool made the corresponding setting. In addition to the ordinary multiple choice questions, this thesis also prepared the form-filling style questions for user. It was suitable for user answering specific types of questions. For example: different conference deadline, the criteria of reviewing paper, or the role frame setting described in Chapter 4.

In the process of answering questions, different questions could be mapped to different frames and rules setting. This thesis divided these questions into several types as follows:

a. Frame Structure Setting: after answering the questions, it could generate the corresponding frames. Different users might generate different frame structures after answering questions. For example: when user decide the information slots that end-user should provide, our authoring tool would alter the S-tuple of ResourceFrame("User", FR, S, VR) (see Definition 1 in Section 3.1) correspondingly.

b. Frame Instance Generating: There were some existent frame structures in the reconfigurable paper-review system generator. User could define new frame instances according to her/his demand. For example: When user defined the roles in his paper-review system, she/he just generated new role frames and made different setting on the slots in each role frame.

c. Rule Generating: During the user answering questions, it might add or change different types of rules defined in Chapter 3. These rules are generated by user after answering questions. For example: System generated the process control rules when user defined the mail sending rules; system generated the authentication rule when user set whether the paper is public or not; Moreover, when user defined the paper frame structure, it may generate some verification rules. (such as “Non-Empty”, “Restricted in Some Options”) Data verification rules were attached in the VR-tuple (Verification Rules) of the 5-tuples in resource frame.

d. Action frame authentication setting: This section was related to role-based access control described in Chapter 4. When user made the configuration in role frames, she/he might set the allowed actions in each role.

User could switch between phases in Configuration System and reconfigure in each phase. Of course, the configuration in previous phases sometimes affected the configuration in following phases. That was the interaction between authoring tool and user, which were elaborated in Section 5.1.3. In next section, this thesis introduced the purpose of each phase in the process of paper-review process construction.

5.1.2 Different Configuration Phase in Authoring Tool

As stated before, this thesis defined several phases based on different purpose of the interview-based questions. For the mainly paper-review process construction, 5 construction phases were defined. In addition, this thesis defined 2 additional phase in the head and tail of these 5 construction phases for the initializing and saving configuration purpose. The structure of our authoring tool was shown in Figure 5.2. The purposes of all the seven phases were stated as follows:

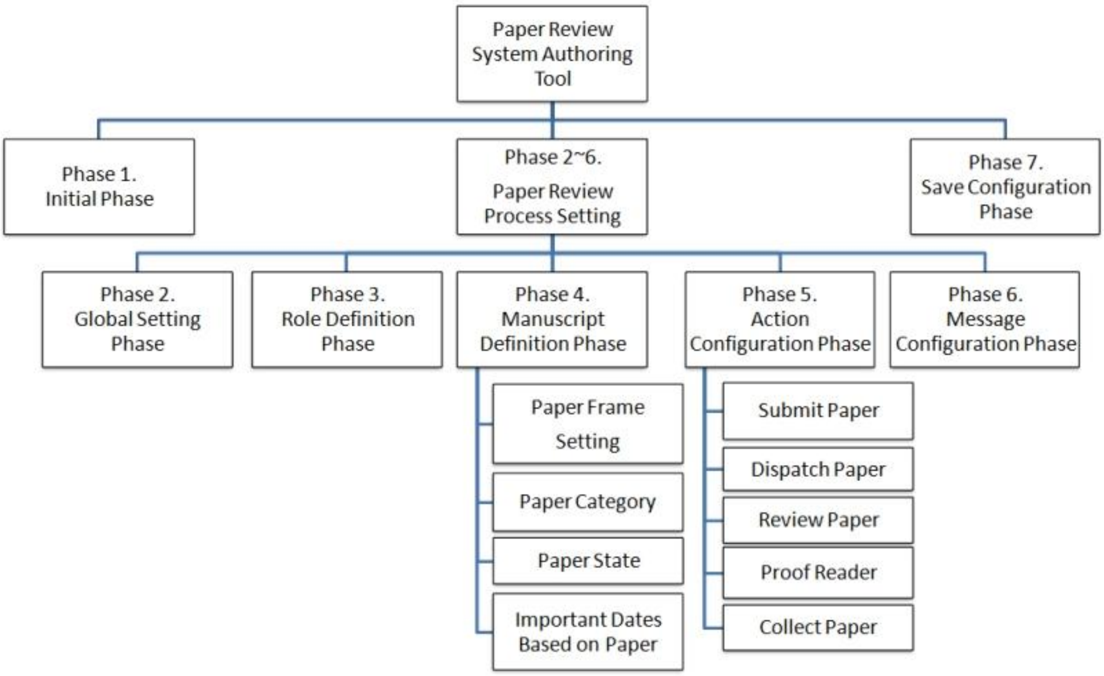

Paper Review System Authoring Tool
Phase 1. Initial Phase
Phase 2~6. Paper Review Process Setting
Phase 7. Save Configuration Phase
Phase 2. Global Setting Phase
Phase 3. Role Definition Phase
Phase 4. Manuscript Definition Phase
Phase 5. Action Configuration Phase
Phase 6. Message Configuration Phase
Paper Frame Setting
Paper Category
Paper State
Important Dates Based on Paper
Submit Paper
Dispatch Paper
Review Paper
Proof Reader
Collect Paper

Figure 5.2: Different configuration phase in authoring tool

Phase 1(Initial Phase): Since this thesis made different users generate their own paper-review process through the authoring tool, this thesis let user create a new configuration session. After creating the session, they could make the following configuration. User had to create a system manager identity. Different paper-review processes could be recognized through this identity. In addition, user could modify her/his configuration in the future through this identity.

Phase 2(Global Setting Phase): In this phase, user decided the purpose of the paper-review system. For a paper-review system, the paper-review process was similar no matter what the purpose of her/his system is. However, when papers were published, publishing to conference and publishing to journal were totally different. Thus, we made user have a choice. User could also set the profile about this conference/journal. In addition, user could add the other static information such as call-for-paper, location information, etc. Moreover, user could define some important dates as well as the subtopics in the conference/journal. They would be the inference facts of the rule which were defined in following phases

Phase 3(Role Definition Phase): In this phase, user generated *Role* frames and defined the *User* frame structure. User had to set the actions which could be performed by each role when setting *Role* frames. Different roles might contain a subsidiary relation (for example: in Figure 4.1, co-editor could be generated by editor) as well as the setting of anonymity. In addition to setting the slot/slot type of *User* frame, user has to setting the verification rules of *User* frame (if need).

Phase 4(Manuscript Definition Phase): In this phase, user made the manuscript-related configuration. For a reconfigurable paper-review system generator, there were several setting options about the manuscript. Thus, this thesis divided this phase into 4 sub-phases: **(a) Paper Frame Setting:** To define the *Paper* frame structure. **(b) Paper Category setting:** To define the category structure. If user had defined the subtopics information in Phase 2, it generated a sub category stood for each subtopic in advance. (c) Paper State Setting: To define the varied paper states of the manuscript could be. (Ex: accept, reject, revise…) These states defined here might be the inference facts of the process control rules. (d) Paper Important Dates Setting: There were some important dates belonging to the conference/journal as well as belonging to the manuscript itself. (for example: the paper re-upload deadline was always set as a few days later after the paper state was set as "revise") Thus, in this sub-phase, user could define the important dates based on the manuscripts.

The configuration in Phase 2~Phase 4 was mainly the configuration on frames.

There were some data verification rules, authentication rules setting among them. The following 2 phases focus on the process control rules setting.

Phase 5(Action Configuration Phase): In this phase, user made configuration about each action in the paper-review process. This thesis split this phase into 5 sub-phases based on the difference of actions as follows: **(a) Submit setting:** User could define the 2-stage submission process in this sub-phase. Some paper-review process of conference/ journal request authors submit a part of information of the manuscript then continued the submission after editor permitting. If user wanted to use the 2-stage submission, she/he had to set what information should be provided first. (This information is referred to the *Paper* frame structure in Phase 4) **(b) Dispatch Setting:** If the manuscript would be reviewed by reviewers, user had to make the related configuration. **(c) Review Setting:** In This sub-phase, user set the review criteria about the submitted manuscripts. **(d) Proof Reader Setting:** User set the information which should be provided when proof reading as well as the following operation after proofreading. **(e) Publishing Setting:** According to the different purpose settings in Phase 2, the configuration in this sub-phase may be different. If this paper-review system was for a journal system, user could decide the indexing format of the journal and set the *Journal* frame structure. If the purpose of paper-review system was for conference, user could determine the agenda of this conference.

There was another configuration in Phase 5, which was defined the paper state transition rules under different actions. It might have several state transition rules within an action. These rules might form the flow of paper-review process as the example of Figure 5.3.

Phase 6(Messages Configuring Phase): In this phase, user could configure about the message passing. To simplify the problem, this thesis assumed that the entire messages among the paper-review system were sent by e-mail. Thus, user could define the mailing rules in the paper-review system. The timing of sending email could be classified as follows: (a) Send mails on the change of paper state (mapping to the large dots on the arrows in Figure 5.3) (b) Send mails before the important dates of system or manuscript. (c) Send mail after specific actions that would not cause the paper state transition. User could define the *Mail* frame, where contained the title, content, and receiver of this mail. Then set when to send this mail. In addition, user could use the “dynamic slot” of other frames in the title/content of the mail. The dynamic slot could be replaced with the corresponding slot values of specific frames before the mail was sent.

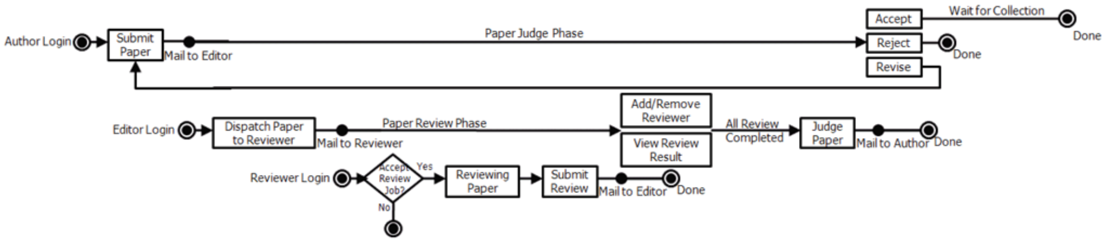


Figure 5.3: A scenario of paper-review process: the paper state transition rules were mapped to the arrows between rectangles of this figure. On the other hand, some of the mailing setting rules were mapped to the large dots on the arrows.

Phase 7(Saving Configuration Phase): After making the configuration in Phase 1~Phase 5, user could save the configuration at this phase. Authoring tool could check if there was an error in previous configuration phase and report the error to user. If there was no missing in configuration, this phase could generate the authoring script of the paper-review system. Then gave a URL linked to the output paper-review system. User could access the output paper-review system immediately. If there was still other configuration need to be made, or the configuration should be changed, user could reconfigure in the previous phases immediately.

5.1.3 Humans' Interaction with Authoring Tool

In order to facilitate user generate her/his own paper-review system, this thesis emphasized the interaction between authoring tool and user, where could help user avoid the unnecessary configure error. The interaction mechanisms were stated as follows:

a. Default Knowledge Loading: It is difficult for user to construct her/his paper-review process from the very beginning. Since the paper-review process were stereotyped (that is why we use frames to represent our paper-review process knowledge). Before user started up the paper-review process construction, the authoring tool loaded the default setting of the resource/action frames in advance. A part of setting was changeable or delectable but the other part of setting wasn't because that it was involved in the system operation. For example: The email slot in user frame was the identification for end-user, so it could not be deleted from *User* frame. In addition to loading the default knowledge, user could load the authoring script into the authoring tool (the authoring script may created by other users) and modify it to generate a new paper-review process.

b. Enabled/Disabled Some Interview-based Questions: User's answer of the question in previous phases might cause that the other questions have no need to be answered. An obvious example was: When user set the purpose of his paper-review system as for journal, there was no need to answer the questions about conference setting. To avoid user's confusion, our authoring tool enabled/disabled automatically the corresponding questions.

c. Data Consistency Maintenance: Splitting the authoring tool into several phases might cause the data consistency issue. Some data setting in Phase B came from the previous Phase A. After user setting the data in Phase B, if user altered the referenced data in Phase A, the data in Phase B might still remain old data. Thus, this thesis proposed a mechanism that could automatically update the data which came from the data in previous phases when user modified them.

In addition to the mechanism stated before, the reminding mechanism during the construction was presented. When there was an obvious error in user's configuration (such as creating the same role frame), the authoring tool would tip to users immediately. Of course, as stated in previous section, the authoring tool checked if there was an error in previous configuration phase and reported the error to user in Phase 7. That was also a part of interaction between user and authoring tool.

5.2 System Build Up Algorithm

In this section, the approaches for generating the authoring script from user's configuration (Part A of Figure 5.4) and generating the corresponding paper-review system by the authoring script (Part B of Figure 5.4) were presented.

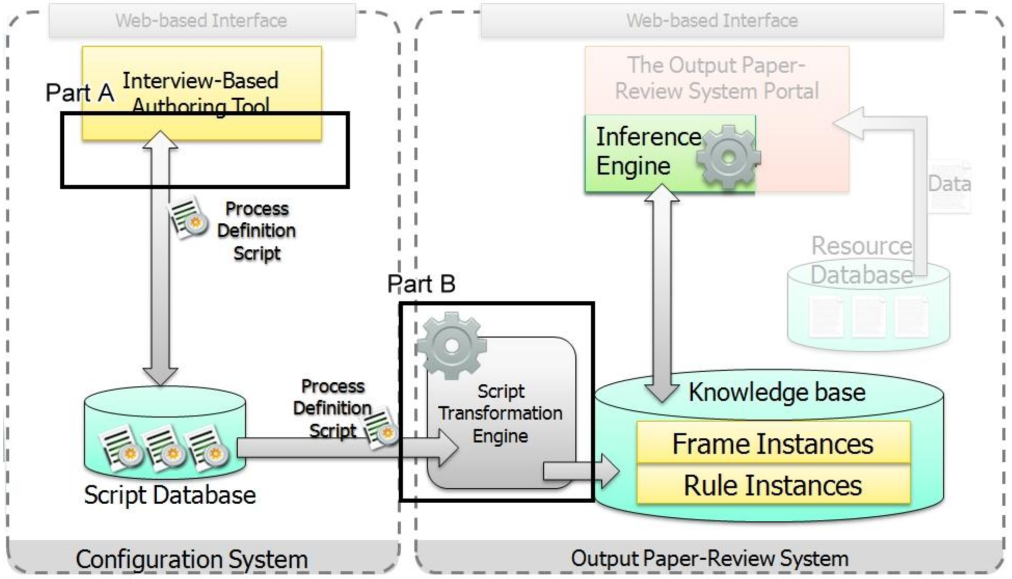

Web-based Interface
Part A
Interview-Based Authoring Tool
Process Definition Script
Script Database
Process Definition Script
Configuration System
Web-based Interface
The Output Paper-Review System Portal
Inference Engine
Data
Resource Database
Part B
Script Transformation Engine
Knowledge base
Frame Instances
Rule Instances
Output Paper-Review System

Figure 5.4: The detailed architecture of requirement acquisition and system generating model: The algorithm of authoring script generating will be used in Part A, and the system generating algorithm will be used in Part B.

5.2.1 Authoring Script Generating Algorithm

ALGORITHM 1 □ Authoring Script Generating Algorithm

Input: User's configuration about the output paper-review system collect by our authoring tool

Output: The XML-based authoring script (Figure 5.5)

Step 1: Define the <paperReviewSystem> node, set name/abbreviation of the conference/journal and the purpose of this authoring script. This information was defined in Phase 2.

Step 2: Set the system profile, which consisted of basis information, system important dates, static pages, and subtopics information. This information was defined in Phase 2, too.

Step 3: Save the *Role* frames information and read the *Role* frame definition in Phase 3. For each role definition, write slot values to each slot.

Step 4: Maintain the paper category structure. The structure information was defined in Phase 4.2. It was a tree-liked structure. It created a <category> node for each superior category. If there were some sub-categories in someone superior category, it created another <category> and insert into the superior category node recursively.

Step 5: Save the paper state information, which are defined in Phase 4.3

Step 6: Save the important dates information based on papers. There were two types of important dates: relatively and absolutely, which were defined in Phase 4.4. For each relatively paper's important date, set the trigger state of some paper and the day-shift information.
For each absolutely papers' important date, it just saved the date information about this important date.

Step 7: Collect the data form information which may be used in the output paper-review system. They are consists of **register form** (defined in Phase 3), **paper frame** (defined in Phase 4.1), **proof reader form** (defined in Phase 5.4), and **journal frame** (defined in Phase 5.5, if the purpose of the output system is conference, this frame will be skipped). The slot information of different resource frames were stored into <attribute> node, which consisted of the slot type, the verification rules about this slot, and the information would be shown below this attribute when user registering.

Step 8: Save the process configuration, it was divided into 5 different action setting: submit, dispatch, review, proof reader, and publish (defined in Phase 5.1~5.5).

Step 8.1: Set the slot names for those need submit first when users using a

2-stage submit mechanism and assigning the max # of attach files into <submitSetting> node.

Step 8.2: Assign the min/max # of reviewer could be dispatch within a paper into <dispatchSetting> node.

Step 8.3: Set the criteria information into <reviewSetting> node.

Step 8.4: Set whether the editor need re-upload a final version of each paper or not into <prSetting> node.

Step 8.5: When user allowed paper public apart from journal/conference, she/he saved the pre/post state of the paper into <prePublic> node. If the purpose of the output system was conference, it saved the time slot information about the conference. Otherwise, it saved the indexing information about the journal.

Step 9: Save the state transition rules defined in Phase 5.1 ~ 5.5, each <rule> node consisted of <precondition> and <postcondition> node. Precondition might have 1 or more pre-state limitation, role limitation, or sub-topic in conference/journal limitation. Post condition might assign a post-state of paper

Step 10: Save the mailing rules into <mailingDefinition> node. There were 3 types of mailing rules (defined in Phase 6).

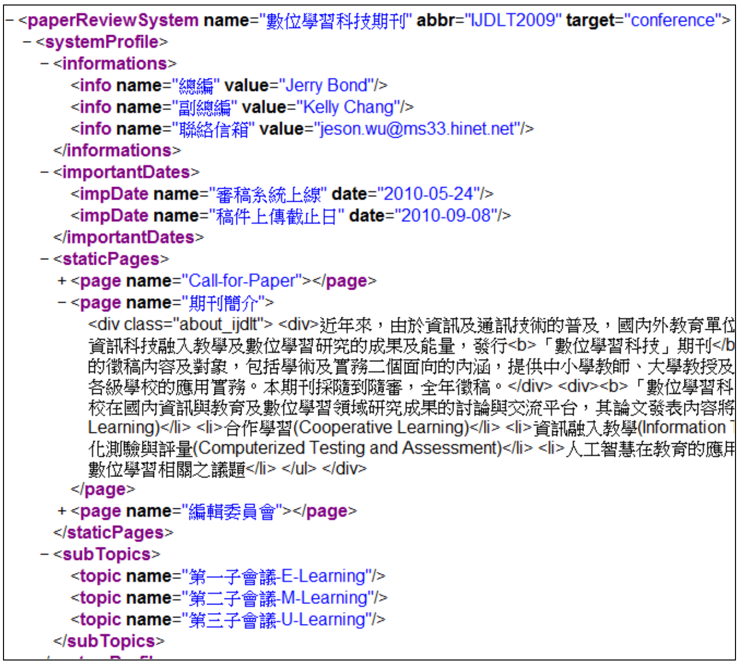
Step 10.1: For paper state transition mail, save the target state and day-shift information into the <precondition> node.

Step 10.2: For important date mail, save the important date type, day-shift information, and the state limit into the <precondition> node.

Step 10.3: For other action mail, save the action type, and role limit involved in the action, and the state limit of paper into the <precondition> node.

No matter the mailing type was, save the mail receiver information into <sendTo> node. Finally, it saved the mail title/content in this mailing rule.

Step 11: output the XML.



```
- <paperReviewSystem name="數位學習科技期刊" abbr="IJDLT2009" target="conference">
  - <systemProfile>
    - <informations>
        <info name="總編" value="Jerry Bond"/>
        <info name="副總編" value="Kelly Chang"/>
        <info name="聯絡信箱" value="jeson.wu@ms33.hinet.net"/>
      </informations>
    - <importantDates>
        <impDate name="審稿系統上線" date="2010-05-24"/>
        <impDate name="稿件上傳截止日" date="2010-09-08"/>
      </importantDates>
    - <staticPages>
      + <page name="Call-for-Paper"></page>
      - <page name="期刊簡介">
          <div class="about_ijdlt"> <div>近年來，由於資訊及通訊技術的普及，國內外教育單位
          資訊科技融入教學及數位學習研究的成果及能量，發行<b>「數位學習科技」期刊</b
          的徵稿內容及對象，包括學術及實務二個面向的內涵，提供中小學教師、大學教授及
          各級學校的應用實務。本期刊採隨到隨審，全年徵稿。</div> <div><b>「數位學習科
          校在國內資訊與教育及數位學習領域研究成果的討論與交流平台，其論文發表內容將
          Learning)</li> <li>合作學習(Cooperative Learning)</li> <li>資訊融入教學(Information
          化測驗與評量(Computerized Testing and Assessment)</li> <li>人工智慧在教育的應用
          數位學習相關之議題</li> </ul> </div>
        </page>
      + <page name="編輯委員會"></page>
      </staticPages>
    - <subTopics>
        <topic name="第一子會議-E-Learning"/>
        <topic name="第二子會議-M-Learning"/>
        <topic name="第三子會議-U-Learning"/>
      </subTopics>
```

Figure 5.5: The XML-based authoring script

5.2.2 Another Algorithms in the Output Paper-review system

After generating the authoring script, this thesis generated the corresponding paper-review system from the authoring script. Because the operation of the output paper-review system involved in the system implementation which was illustrated in chapter 6 in details. We just described the meta-algorithm of generating the output paper-review system at this chapter and focus on the rule inference on the paper state transition and mailing.

META-ALGORITHM □ Paper-review system Generating Algorithm

Input: The authoring script of the paper-review process generated in algorithm 1

Output: The web-based paper-review system followed the user-defined paper-review process.

Step 1: Load all the static pages definition in authoring script and rendering the hyperlink.

Step 2: Generate the dynamic pages such as index page, conference/journal papers in public.

Step 3: Maintain the register/login/logout process based on the setting of role frames and user frame.

Step 4: Maintain the access control into the actions in the paper-review process based on the setting of *Role* frames and *User* frame.

Step 5: For each actions in paper-review process, load the corresponding configuration. It would be briefly stated as below:

Step5.1: For the paper submission: load the structure of paper frame, the definition of subtopic and structure of paper category. And the other setting in the <submissionSetting> node in authoring script

Step5.2: For the paper dispatch: load the setting in the <dispatchSetting> node in authoring script

Step5.3: For the paper reviewing: load the criteria definition and the other setting in the <reviewSetting> node in authoring script.

Step5.4: For the paper proof reading: load the proof reading data form structure and the other setting in the <prSetting> node in authoring script.

Step5.5: For the paper publishing: for a journal system, it generated the journal publishing interface based on the journal frame structure; for a conference system, it generated the paper publish interface if the agenda of the conference needed published paper. Another setting about this action can be achieved in <publishSetting> node in authoring script.

Step 6: After performing different actions in the paper-review process, infer the paper state transition rules and mailing rules. This inference algorithm will be illustrated in Algorithm 2 and Algorithm 3.A in detail.

Step 7: Execute the Mailing Sending Algorithm which is illustrated in Algorithm 3.B periodically.

Because the mechanisms of rules inference make our paper-review system reconfigurable, we will especially illustrate the algorithm of paper state inference as follows.

ALGORITHM 2 □ Paper State Inference Algorithm

Input: a) Paper state transition rules defined in authoring scripts
b) Paper frame and other facts (ex: current action).

Output: The next state of current paper frame

Step 1: Set Candidate_State as an empty array

Step 2: Load all paper state transition rules into inference engine.

Step 3: For each paper state transition rule, check if the precondition of the rule matches with current paper.

Step 4: If there was any rule that its precondition satisfied, push the <poststate> information into Canditdate_State

Step 5: If the size of Candidate_State is 0, output "After (action name) (###)" Where (action name) was current action name, (###) was a unique serial number.

Step 6: If the size of Candidate_State was equal or greater than 1, output Candidate_State[0] + "(###)". Where (###) was a unique serial number.

In algorithm 2, the size of Candidate_State should be 1 normally if the setting of paper state transition rules was correct. However, the paper state transition rules were defined by users and they might make mistake in defining these rules. The size of Candidate_State might not equal to zero. This thesis discussed this condition in Chapter 5.3. On the other hand, this thesis proposed the mechanism of mailing management.

ALGORITHM 3.A □ Mailing Generating Algorithm

Input: Mailing rules defined in authoring scripts

Output: The sending mail jobs saved into database

Step 1: If the output system is loaded in first time, add mailing jobs if there was any mailing rule which the precondition was for the system.

Step 2: When some paper states have been changed, check if the pre-state in precondition of *state transition mailing rules* was current state of the paper. If yes, add a mailing job, set the sending time as current date plus the shifted days, set the state limit into database.

Step 3: When accessed some action frames and perform the action, checking if the action name in *other action mailing rule* was equal to current action name and current user/the role of the target of current action was match the role limit or not. If yes, it added a mailing job, set the sending time as current time, set the state limit into database.

Obviously, after the execution of algorithm 3.A, the mails were not sent out immediately. Instead, they were stored into database. That was because some mails would be check whether they still had to be sent or not at the time to be sent. Thus, this thesis executed an algorithm for mail sending periodically.

ALGORITHM 3.B □ Mailing Sending Algorithm

Input: The generated mail stored in database

Output: The mail sent into the receiver

Step 1: Select the generated mails stored in database where the sending time was between (current time – t) and current time, where t was the time interval of invoking this algorithm.

Step 2: For each mail data, if the data consisted a paper state limitation, system checked the current state of corresponding paper, if not under the limit anymore, just skip this data.

Step 3: Based on the sendTo information of the data, assign the receivers' e-mail of each mail.

Step 4: Before the mail being sent into the receiver, system found all the dynamic fields in mail's title or content. If there was any dynamic field, extract the dynamic field type and replace it with corresponding frame slot value.

By Algorithm 3.B, the mails in the paper-review system would be sent at a given time. In step 4, why did not the system replace all the dynamic fields in mails until the mails were about to be sent? The reason was that those dynamic fields could not be accessed necessarily. For example: The mails based on the system important dates would be imported to the database when the system loading this system first time. However, the corresponding dynamic fields such as reviewers' name cannot be accessed because there was no user as a reviewer role yet in the system.

5.3 Another Issue of the Output Paper-review system

There were other issues need be taken consideration when used frames and rules as a representation of paper-review process. First, we controlled the paper-review process by the mechanism of rule inference. What should we do if there was no inference result or more than 1 inference results? In addition, we made user define the frame structure as user's wish. How did we update those frames when the frame structures were changed?

For the condition of getting empty or more than one inference result, we had stated briefly in Algorithm 2. In order to keep the execution of paper-review process as usual, the system still gave an inference result. When the size of Candidate_Paper was zero, the system assigned a "Pseudo" paper state as an inference result. (As stated in Step 5 of Algorithm 2) At the same time, the system stored all the facts of current inference process into database as an "inference exception case". These inference exceptions would be reported to user (system manager). User could modify current paper-review process by adding some paper state transition rules, and execute the inference process again. Similarly, if there were more than one inference results, the system would follow the same process of inference exception reporting. It could help user define more specific rules and execute the inference process again.

On the other hand, when a frame structure in the paper-review system was changed, how to fix those frame instances which were created based on the old frame structure definition? Obviously, we could not update the slot value one by one because we might not the owner of those frame instances. Thus, we contain the old frame structure information of them. However, when the owner of those frames logged in, system would notify her/him to make some change on some slot value in those frames. For example: When end-user A registered a user account, system generated a user frame. If system manager modify the *User* frame structure, end-user A would receive a notification to update her/his profile because of the modification of *User* frame structure. Thus, we had solved the problem on the modification of frames structure.

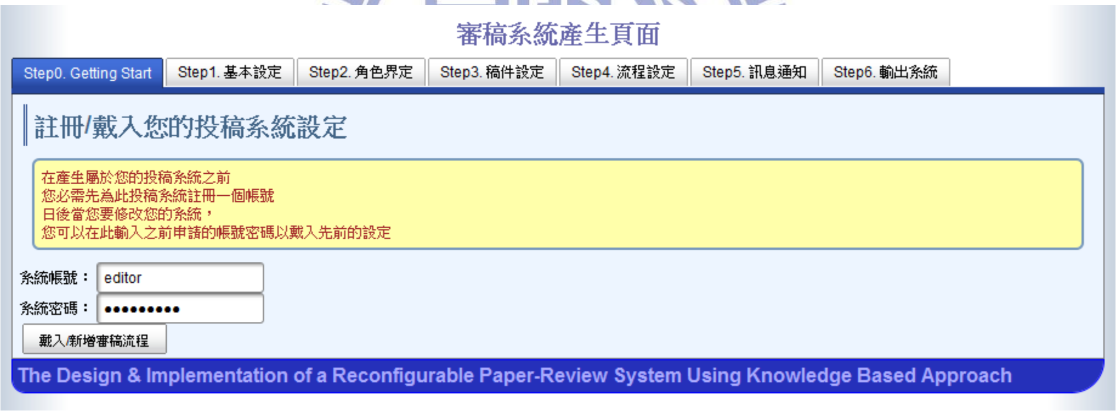
Chapter 6 Implementation and Experiment

This thesis had implemented a reconfigurable paper-review system generator. As shown in Figure 3.1, the reconfigurable paper-review system generator consisted of authoring tool and the output paper-review system. When user generated an authoring script by the authoring tool, she/he could load the script into the transform engine and generate the corresponding paper-review system. In this chapter, we first introduced the interface which implemented in this thesis. To evaluate the reconfigurable paper-review system generator, this thesis collected the mainly existing paper-review processes and simulated them by the authoring tool in this thesis. In the end of this chapter, this thesis discussed about the evaluation result.

6.1 System Implementation

6.1.1 Implementation of a Paper-Review Process Authoring Tool

Figure 6.1~6.8 were the interface of the paper-review process authoring tool. The paper-review process had been illustrated in Ch5.1. This thesis split the process of construction into several phases. User could create a new process here.



審稿系統產生頁面

註冊/戴入您的投稿系統設定

在產生屬於您的投稿系統之前
您必需先為此投稿系統註冊一個帳號
日後當您要修改您的系統，
您可以在此輸入之前申請的帳號密碼以戴入先前的設定

系統帳號：editor

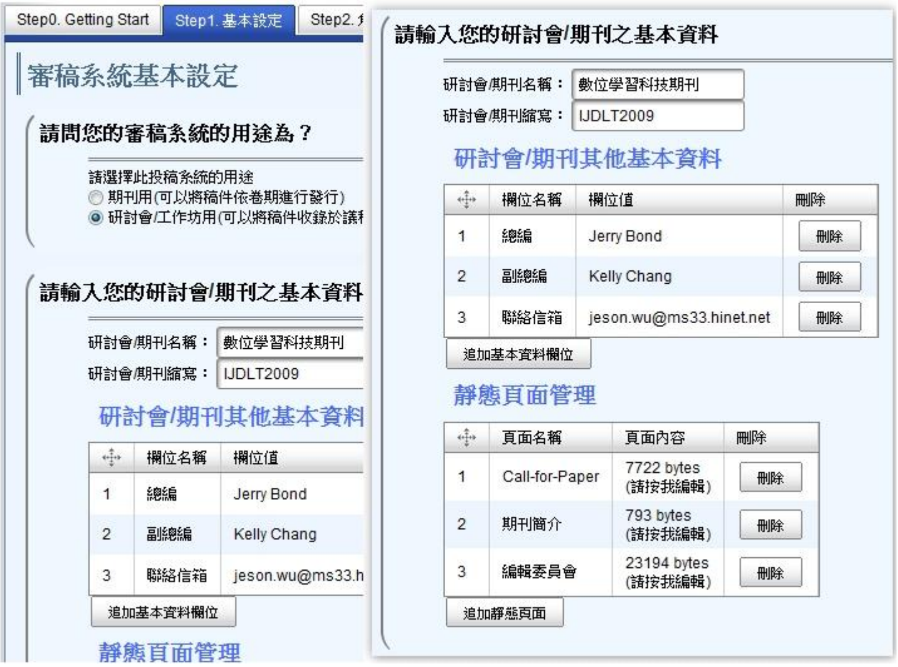
系統密碼：●●●●●●●●●●

The Design & Implementation of a Reconfigurable Paper-Review System Using Knowledge Based Approach

Figure 6.1: The initializing phase of the authoring tool

If it was the first time user entered this interface, she/he would create an account as an identity of her/his paper-review process. After this action, user could configure their process.

The authoring tool consisted of a sequence of interview-based questions. For the frame editing interface, user could create a new frame/rule by clicking the add button. She/he could directly click the cells in tables to make the corresponding modification.



Step0. Getting Start | Step1. 基本設定 | Step2.

審稿系統基本設定

請問您的審稿系統的用途為？

請選擇此投稿系統的用途
○ 期刊用(可以將稿件依卷期進行發行)
● 研討會/工作坊用(可以將稿件收錄於議

請輸入您的研討會/期刊之基本資料

研討會/期刊名稱： 數位學習科技期刊
研討會/期刊縮寫： IJDLT2009

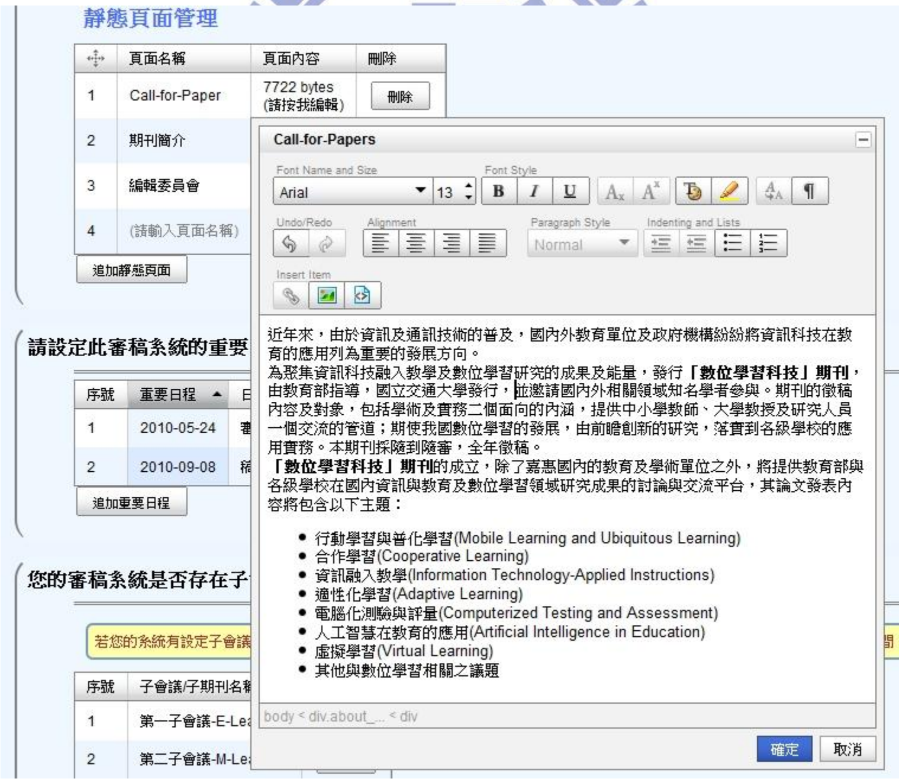
研討會/期刊其他基本資料

	欄位名稱	欄位值	刪除
1	總編	Jerry Bond	刪除
2	副總編	Kelly Chang	刪除
3	聯絡信箱	jeson.wu@ms33.hinet.net	刪除

靜態頁面管理

	頁面名稱	頁面內容	刪除
1	Call-for-Paper	7722 bytes (請按我編輯)	刪除
2	期刊簡介	793 bytes (請按我編輯)	刪除
3	編輯委員會	23194 bytes (請按我編輯)	刪除

Figure 6.2: The interview-based questions in Phase 1



靜態頁面管理

	頁面名稱	頁面內容	刪除
1	Call-for-Paper	7722 bytes (請按我編輯)	刪除
2	期刊簡介		
3	編輯委員會		
4	(請輸入頁面名稱)		

請設定此審稿系統的重要

序號	重要日程 ▲	日
1	2010-05-24	審
2	2010-09-08	稿

您的審稿系統是否存在子

若您的系統有設定子會議

序號	子會議/子期刊名稱
1	第一子會議-E-Lea
2	第二子會議-M-Lea

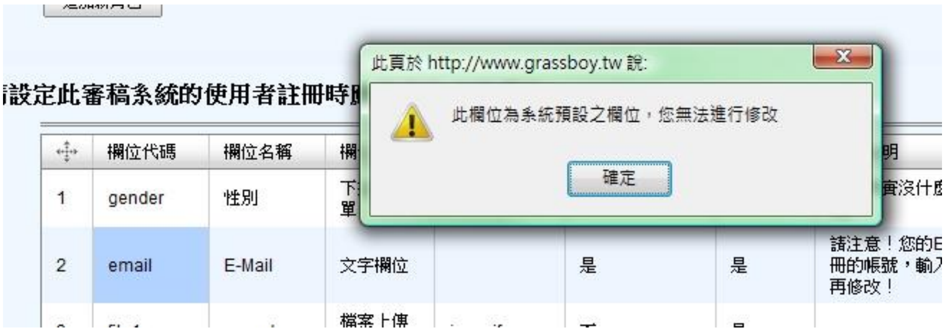
Call-for-Papers

近年來，由於資訊及通訊技術的普及，國內外教育單位及政府機構紛紛將資訊科技在教育的應用列為重要的發展方向。
為聚集資訊科技融入教學及數位學習研究的成果及能量，發行**「數位學習科技」期刊**，由教育部指導，國立交通大學發行，並邀請國內外相關領域知名學者參與。期刊的徵稿內容及對象，包括學術及實務二個面向的內涵，提供中小學教師、大學教授及研究人員一個交流的管道；期使我國數位學習的發展，由前瞻創新的研究，落實到各級學校的應用實務。本期刊採隨到隨審，全年徵稿。
「數位學習科技」期刊的成立，除了嘉惠國內的教育及學術單位之外，將提供教育部與各級學校在國內資訊與教育及數位學習領域研究成果的討論與交流平台，其論文發表內容將包含以下主題：

- 行動學習與普化學習(Mobile Learning and Ubiquitous Learning)
- 合作學習(Cooperative Learning)
- 資訊融入教學(Information Technology-Applied Instructions)
- 適性化學習(Adaptive Learning)
- 電腦化測驗與評量(Computerized Testing and Assessment)
- 人工智慧在教育的應用(Artificial Intelligence in Education)
- 虛擬學習(Virtual Learning)
- 其他與數位學習相關之議題

Figure 6.3: User could direct edit the resource frames by clicking the cell in tables

Because the paper-review process was stereotyped, some default knowledge could not be modified. When user tried to modify the default knowledge, the authoring tool would notify her/him as shown in Figure 6.4



請設定此審稿系統的使用者註冊時應

✣	欄位代碼	欄位名稱	欄				明
1	gender	性別	下 單				實沒什麼
2	email	E-Mail	文字欄位		是	是	請注意！您的E 冊的帳號，輸入 再修改！

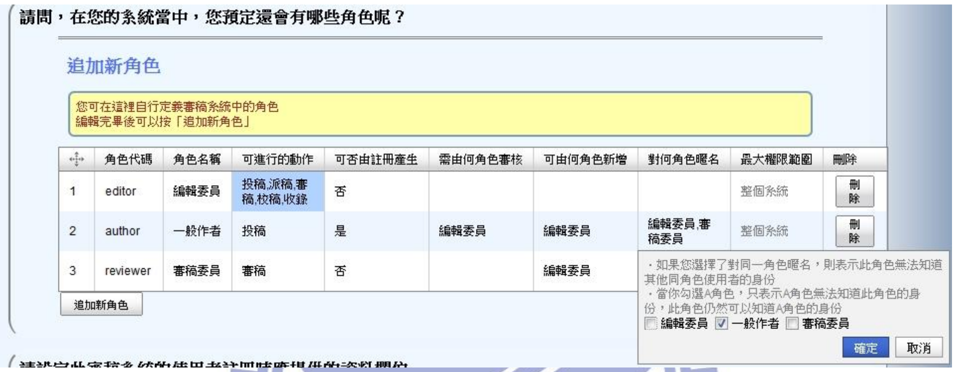
此頁於 http://www.grassboy.tw 說:

此欄位為系統預設之欄位，您無法進行修改

確定

Figure 6.4: The notification when user tried to modify the default knowledge

This authoring tool also implemented the RBAC model, and the anonymity management interface as shown in Figure 6.5



請問，在您的系統當中，您預定還會有哪些角色呢？

追加新角色

您可在這裡自行定義審稿系統中的角色
編輯完畢後可以按「追加新角色」

✣	角色代碼	角色名稱	可進行的動作	可否由註冊產生	需由何角色審核	可由何角色新增	對何角色匿名	最大權限範圍	刪除
1	editor	編輯委員	投稿,派稿,審稿,校稿,收錄	否				整個系統	刪除
2	author	一般作者	投稿	是	編輯委員	編輯委員	編輯委員,審稿委員	整個系統	刪除
3	reviewer	審稿委員	審稿	否		編輯委員			

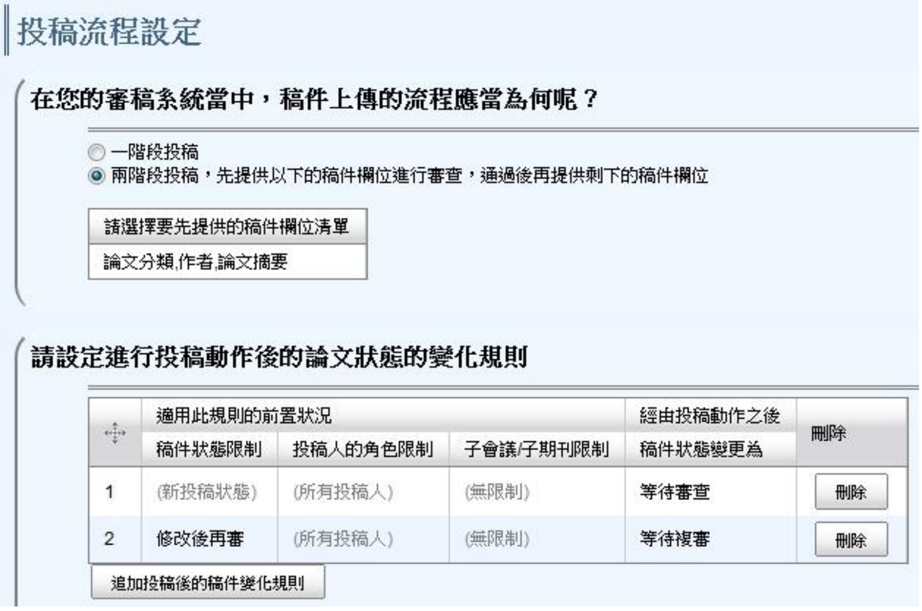
追加新角色

・如果您選擇了對同一角色匿名，則表示此角色無法知道其他同角色使用者的身份
・當你勾選A角色，只表示A角色無法知道此角色的身份，此角色仍然可以知道A角色的身份
□ 編輯委員 ☑ 一般作者 □ 審稿委員

確定 取消

Figure 6.5: Role frame definition

In addition to frame editing, user had to set the process control rule about paper state transition and mailing rule as shown in Figure 6.6, Figure 6.7



投稿流程設定

在您的審稿系統當中，稿件上傳的流程應當為何呢？

○ 一階段投稿
● 兩階段投稿，先提供以下的稿件欄位進行審查，通過後再提供剩下的稿件欄位

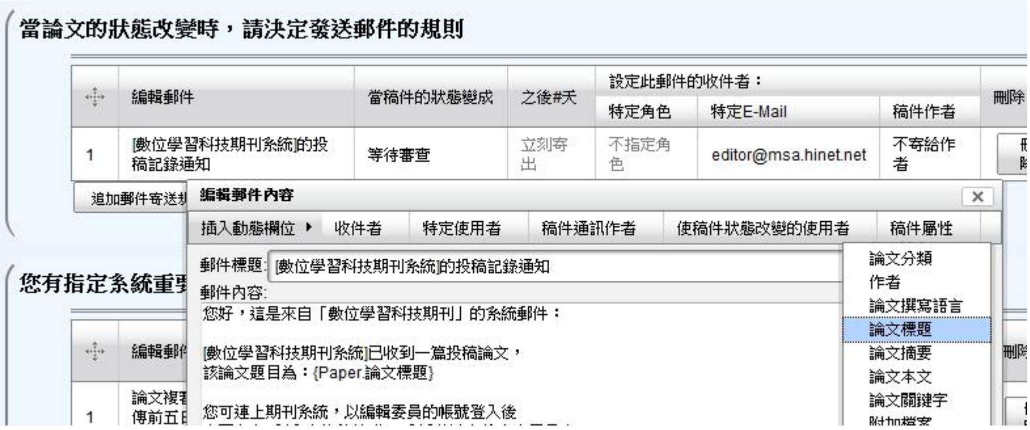
請選擇要先提供的稿件欄位清單
論文分類,作者,論文摘要

請設定進行投稿動作後的論文狀態的變化規則

✣	適用此規則的前置狀況			經由投稿動作之後	刪除
	稿件狀態限制	投稿人的角色限制	子會議/子期刊限制	稿件狀態變更為	
1	(新投稿狀態)	(所有投稿人)	(無限制)	等待審查	刪除
2	修改後再審	(所有投稿人)	(無限制)	等待複審	刪除

追加投稿後的稿件變化規則

Figure 6.6: Process control rules setting interface.



當論文的狀態改變時，請決定發送郵件的規則

	編輯郵件	當稿件的狀態變成	之後#天	設定此郵件的收件者：			刪除
				特定角色	特定E-Mail	稿件作者	
1	[數位學習科技期刊系統]的投稿記錄通知	等待審查	立刻寄出	不指定角色	editor@msa.hinet.net	不寄給作者	

編輯郵件內容

郵件標題: [數位學習科技期刊系統]的投稿記錄通知

郵件內容:

您好，這是來自「數位學習科技期刊」的系統郵件：

[數位學習科技期刊系統]已收到一篇投稿論文，
該論文題目為：{Paper.論文標題}

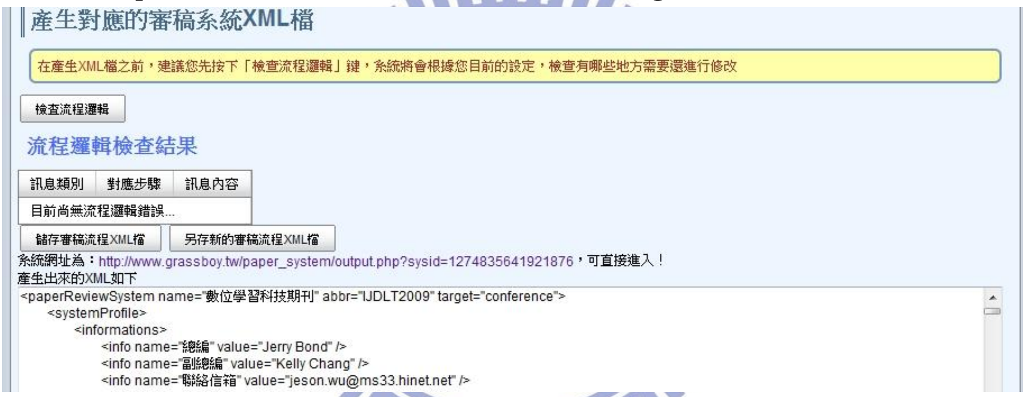
您可連上期刊系統，以編輯委員的帳號登入後

您有指定系統重

	編輯郵	刪除
1	論文複審 傳前五日	

Figure 6.6: Mailing rules setting interface

After user made the entire configuration about the paper-review process, she/he could generate the authoring script at the final phase as shown in Figure 6.7. As stated in Ch5.1.2, the authoring tool checked if there was an error in previous configuration phase and reported the error to user. It is shown in Figure 6.8.



產生對應的審稿系統XML檔

在產生XML檔之前，建議您先按下「檢查流程邏輯」鍵，系統將會根據您目前的設定，檢查有哪些地方需要還進行修改

流程邏輯檢查結果

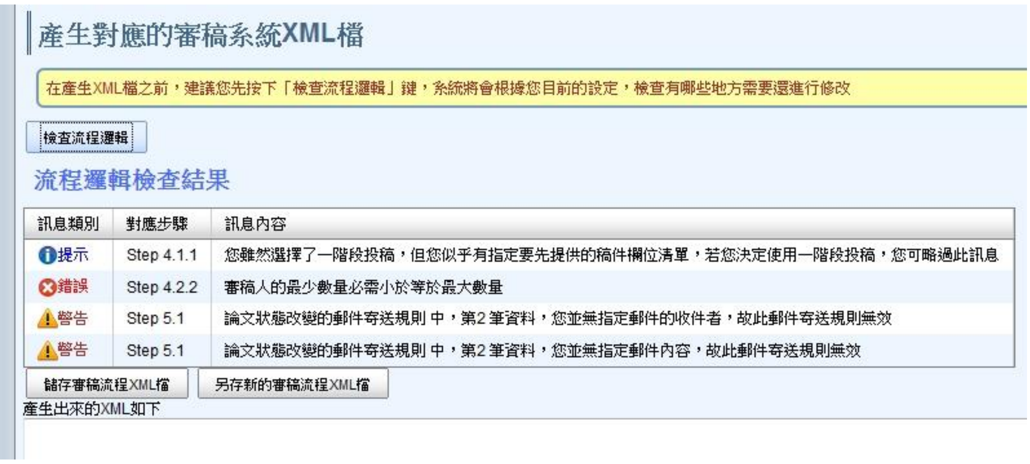
訊息類別	對應步驟	訊息內容
目前尚無流程邏輯錯誤...		

系統網址為：http://www.grassboy.tw/paper_system/output.php?sysid=1274835641921876，可直接進入！

產生出來的XML如下

```
<paperReviewSystem name="數位學習科技期刊" abbr="IJDLT2009" target="conference">
    <systemProfile>
        <informations>
            <info name="總編" value="Jerry Bond" />
            <info name="副總編" value="Kelly Chang" />
            <info name="聯絡信箱" value="jeson.wu@ms33.hinet.net" />
```

Figure 6.7: Generating the authoring script



產生對應的審稿系統XML檔

在產生XML檔之前，建議您先按下「檢查流程邏輯」鍵，系統將會根據您目前的設定，檢查有哪些地方需要還進行修改

流程邏輯檢查結果

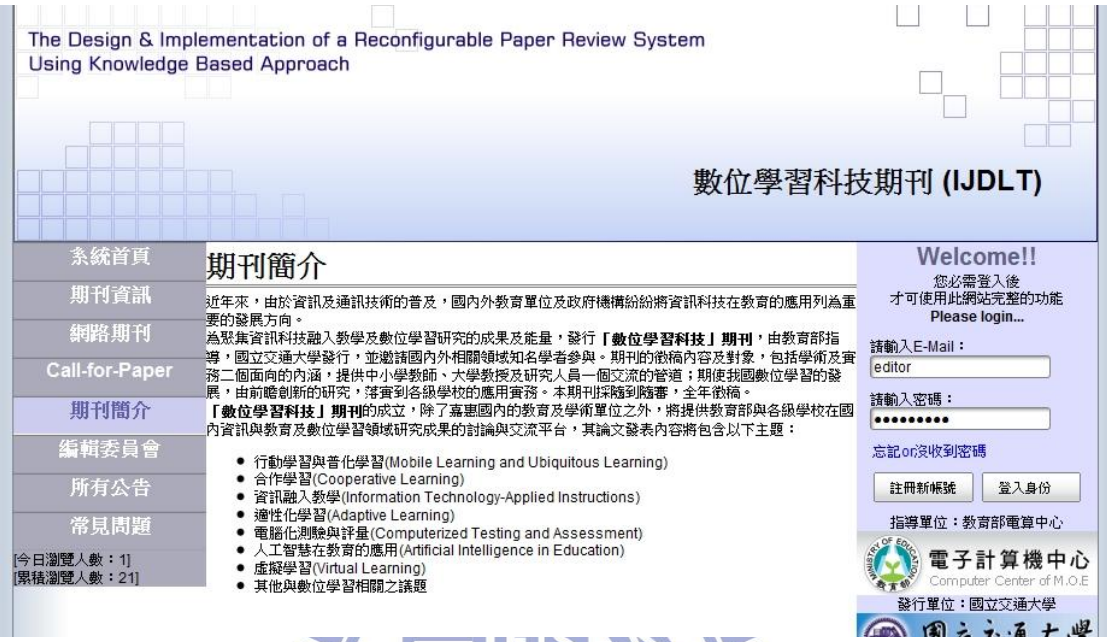
訊息類別	對應步驟	訊息內容
ℹ提示	Step 4.1.1	您雖然選擇了一階段投稿，但您似乎有指定要先提供的稿件欄位清單，若您決定使用一階段投稿，您可略過此訊息
⊗錯誤	Step 4.2.2	審稿人的最少數量必需小於等於最大數量
⚠警告	Step 5.1	論文狀態改變的郵件寄送規則 中，第2 筆資料，您並無指定郵件的收件者，故此郵件寄送規則無效
⚠警告	Step 5.1	論文狀態改變的郵件寄送規則 中，第2 筆資料，您並無指定郵件內容，故此郵件寄送規則無效

產生出來的XML如下

Figure 6.8: The interaction between authoring tool and user

6.1.2 Implementation of the Output Paper-Review System

After generated the authoring script, user could load the script into the transform engine. Then it generated a paper-review system such as shown in Figure 6.9.



期刊簡介

近年來，由於資訊及通訊技術的普及，國內外教育單位及政府機構紛紛將資訊科技在教育的應用列為重要的發展方向。
為聚集資訊科技融入教學及數位學習研究的成果及能量，發行**「數位學習科技」期刊**，由教育部指導，國立交通大學發行，並邀請國內外相關領域知名學者參與。期刊的徵稿內容及對象，包括學術及實務二個面向的內涵，提供中小學教師、大學教授及研究人員一個交流的管道；期使我國數位學習的發展，由前瞻創新的研究，落實到各級學校的應用實務。本期刊採隨到隨審，全年徵稿。
「數位學習科技」期刊的成立，除了嘉惠國內的教育及學術單位之外，將提供教育部與各級學校在國內資訊與教育及數位學習領域研究成果的討論與交流平台，其論文發表內容將包含以下主題：

- 行動學習與普化學習(Mobile Learning and Ubiquitous Learning)
- 合作學習(Cooperative Learning)
- 資訊融入教學(Information Technology-Applied Instructions)
- 適性化學習(Adaptive Learning)
- 電腦化測驗與評量(Computerized Testing and Assessment)
- 人工智慧在教育的應用(Artificial Intelligence in Education)
- 虛擬學習(Virtual Learning)
- 其他與數位學習相關之議題

[今日瀏覽人數：1]
[累積瀏覽人數：21]

Welcome!!
您必需登入後
才可使用此網站完整的功能
Please login...

請輸入E-Mail：editor

請輸入密碼：••••••••••

忘記or沒收到密碼

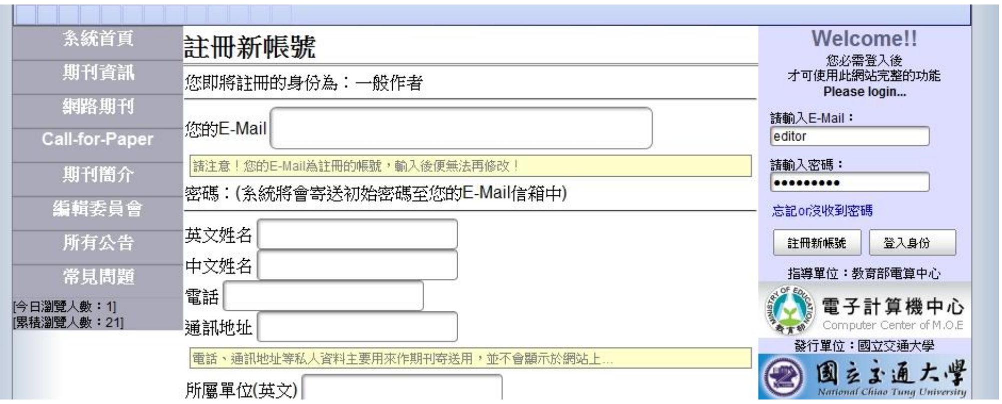
指導單位：教育部電算中心

發行單位：國立交通大學

Figure 6.9: The overview of the output paper-review system

In Figure 6.9, the two external links (“Announcement” and “frequently asked questions”) were the built-in module. Because they were not a part of paper-review process and not the focus in this thesis, we just implemented them by the tradition programming approach.

When end-users connect to the output paper-review system, they register as an author (or someone role could register defined in the system). The register interface shown in Figure 6.10 was generated by referencing the user frame and role frames.



[今日瀏覽人數：1]
[累積瀏覽人數：21]

註冊新帳號

您即將註冊的身份為：一般作者

您的E-Mail

請注意！您的E-Mail為註冊的帳號，輸入後便無法再修改！

密碼：(系統將會寄送初始密碼至您的E-Mail信箱中)

英文姓名

中文姓名

電話

通訊地址

電話、通訊地址等私人資料主要用來作期刊寄送用，並不會顯示於網站上...

所屬單位(英文)

Welcome!!
您必需登入後
才可使用此網站完整的功能
Please login...

請輸入E-Mail：editor

請輸入密碼：●●●●●●●●●

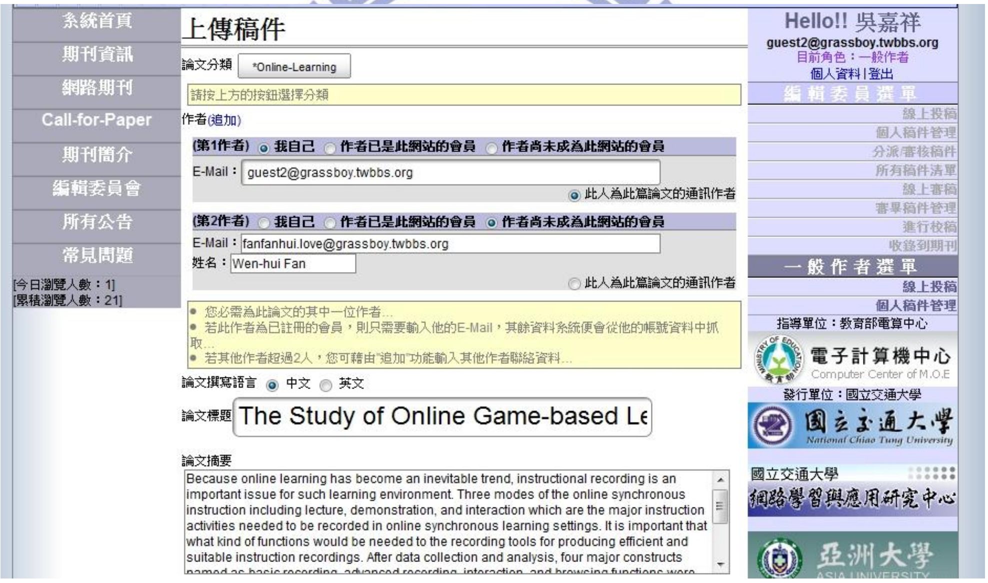
忘記or沒收到密碼

指導單位：教育部電算中心

發行單位：國立交通大學

Figure 6.10: The register interface of the output paper-review system

After the registration process, end-user could start to use this paper-review system. Figure 6.11 was the paper submission interface. Of course, this interface was generated by referencing the paper frame. End-user could submit paper and then start the paper-review process.



[今日瀏覽人數：1]
[累積瀏覽人數：21]

上傳稿件

論文分類 *Online-Learning

請按上方的按鈕選擇分類

作者(追加)

(第1作者) ● 我自己 ○ 作者已是此網站的會員 ○ 作者尚未成為此網站的會員

E-Mail：guest2@grassboy.twbbs.org

● 此人為此篇論文的通訊作者

(第2作者) ○ 我自己 ○ 作者已是此網站的會員 ● 作者尚未成為此網站的會員

E-Mail：fanfanhui.love@grassboy.twbbs.org

姓名：Wen-hui Fan

○ 此人為此篇論文的通訊作者

- 您必需為此論文的其中一位作者...
- 若此作者為已註冊的會員，則只需要輸入他的E-Mail，其餘資料系統便會從他的帳號資料中抓取...
- 若其他作者超過2人，您可藉由"追加"功能輸入其他作者聯絡資料...

論文撰寫語言 ● 中文 ○ 英文

論文標題 The Study of Online Game-based Le

論文摘要

Because online learning has become an inevitable trend, instructional recording is an important issue for such learning environment. Three modes of the online synchronous instruction including lecture, demonstration, and interaction which are the major instruction activities needed to be recorded in online synchronous learning settings. It is important that what kind of functions would be needed to the recording tools for producing efficient and suitable instruction recordings. After data collection and analysis, four major constructs named as basic recording, advanced recording, interaction, and browsing functions were

Hello!! 吳嘉祥
guest2@grassboy.twbbs.org
目前角色：一般作者
個人資料 | 登出

指導單位：教育部電算中心

發行單位：國立交通大學

Figure 6.11: The paper submission interface of the output paper-review system

6.2 Experiment and Evaluation

6.2.1 Experiment Design

This thesis used the authoring tool to generate the different paper-review processes for journal or conference, and compared this thesis with current existing template-based paper/conference systems such as OJS, OCS, MyReview, and the traditional paper system such as IJDLT in supportability. For a reconfigurable paper-review system generator, there were many criteria of the dynamic system supportability. Thus, we also compared the system in our thesis with other template-based paper-review system in these criteria.

At first, we briefly introduced each paper-review process would be generate. They might be collected by referencing the existing paper as well as collecting by an interview with the chief of a specific conference or journal.

Case 1: Paper-review process in [13]

The process was for a journal paper system. There were 3 roles in the system: editor, reviewer, and author. User could register as an author then submit paper. Editor received the papers and dispatched them to no more than 3 other reviewers or editor for reviewing. The paper might be judge as accept, reject, major revise, or minor revise. When paper was accepted, it entered the proofreading process. Finally, it waited for being collected into the journal. There were several types mail sent to author, reviewer or editor automatically such as the revise deadline notification. Editor could add/remove author or reviewer. In addition editor could define the category ontology for paper submission.

Case 2: Paper-review process in [2]

The process was for a conference management system. There were 3 roles in the system: administrator, reviewer, authors. The process was different with Case 1. Author could withdraw a submitted paper. Editor could assign the reviewing job to review based on reviewers' preferences. Administrator could only accept or reject the paper. There was only the judge result mail sent to author automatically. Finally, the administrator could manage the session of conference.

Case 3: Paper-review process in [3]

Fan et al. [3] presented an anonymous paper-review mechanism. In their paper-review process, there existed the anonymity between author, editor, and review each other. In case 1, all users were not anonymous with editor. (That is, editor could know who the author of a paper is or who the review of a paper was.) This thesis just presented an anonymous paper-review process model. It didn’t propose a real system.

Case 4: Paper-review process in Conference A

This process was collected by interviewing with a conference chief editor. In their conference, there existed a hierarchical role definition. That is, chief editor could generate several session chairs for different topic of conference. Different session chairs could generate reviewers themselves. When author submitted paper to different topic in the conference, the corresponding session chair could dispatch the paper to reviewer under this topic. After each session chair published the accept papers. They could send back to the chief editor for the publication.

Case 5: Paper-review process in Journal B

This process was collected by interviewing with a journal editor. In their journal system, there exists another role “assistant editor”. Assistant editor was not responsible for paper dispatching. This role was only responsible for the proof reading task. On the other hand, when author submitted paper, the journal system asked user provide the information of number of words it helped end-user.

By observation of Case 1~5, we summarized different functionality of a paper-review system and listed the supportability in different case with different functionality as Table 6.1.

In addition to the supportability for the existing paper-review process, we compared this thesis with current existing template-based paper-review systems in the criteria of dynamic system supportability, these criteria are stated as follows.

a. **Setup time**: The construction time when start to construct the paper-review system a user needs.
b. **Custom user profile**: The ability of determining the user information should be provided when author register.
c. **Custom paper attribute**: The ability of determining the manuscript information should be provided when author submit paper.
d. **Custom paper-review criteria**: The ability of determining the criteria for evaluating paper when reviewing.
e. **Dynamic roles definition**: The ability of defining the role types in the paper-review system.
f. **Knowledge sharing**: The ability of constructing a paper-review system based on the others’ paper-review process.
g. **Custom mailing rules**: The ability of defining the automatically mailing rules and mail template.

h. **Custom process control**: The ability of defining the different paper-review workflow.

Table 6.1: The functionality in different paper-review process cases

Cases Support? Functionality	Case 1	Case 2	Case 3	Case 4	Case 5
editor, reviewer, author	Yes	Yes	Yes	Yes	Yes
Assistant editor	No	No	No	No	Yes
Hierarchical reviewing process	No	No	No	Yes	No
Paper category management	Yes	Not mentioned	Not mentioned	Not mentioned	Yes
Submission	Yes	Yes	Yes	Yes	Yes
Number of words of paper	No	No	No	No	Yes
Dispatching	Yes	Yes	Yes	Yes	Yes
Reviewing	Yes	Yes	Yes	Yes	Yes
Proof Reading	Yes	No	No	Yes	Yes
Publish to journal	Yes	No	No	No	Yes
Publish to conference	No	Yes	No	Yes	No

6.2.2 Experiment Result and Discussion

Table 6.2 showed the functionality supportability of this thesis and another paper-review system with the current existing paper review process.(Table 6.1) In Table 6.2, because the other paper-review system restricted the role definition in their systems, they could not generate the role “assistant editor” or define the hierarchical reviewing process. The MyReview System, OJS, OCS were partially support the paper category management because that they could only define one-level of paper categories. They could not define a paper sub-category under a specific paper category. For other systems, they were developed for the specific purpose, so they could not support to generate the journal system or conference system as users’ wish.

Table 6.2: The supportability in different researches with different functionality

Systems Support? Functionality	RPS	MyReview	OJS	OCS	IJDLT
editor, reviewer, author	Yes	Yes	Yes	Yes	Yes
Assistant editor	Yes	No	Yes	Yes	No
Hierarchical reviewing process	Yes	No	No	No	No
Paper category management	Yes	Partially	Partially	Partially	Yes
Submission	Yes	Yes	Yes	Yes	Yes
Number of words of paper	Yes	Yes	Yes	Yes	No
Dispatching	Yes	Yes	Yes	Yes	Yes
Reviewing	Yes	Yes	Yes	Yes	Yes
Proof Reading	Yes	Yes	Yes	Yes	Yes
Publish to journal	Yes	No	Yes	No	Yes
Publish to conference	Yes	Yes	No	Yes	No

The comparison with the other template-based paper review systems in the criteria of dynamic system supportability was shown in Table 6.3.

Table 6.3: The comparison among different researches in the criteria of dynamic system supportability

	RPS	**MyReview**	**OJS**	**OCS**
Setup time cost	Lower	Higher	Higher	Higher
Custom user profile	Support	Not Support	Not Support	Not Support
Dynamic roles definition	Support	Not Support	Not Support	Not Support
Custom paper attribute	Support	Support	Support	Support
Custom paper-review criteria	Support	Support	Support	Support
Knowledge sharing	Support	Not Support	Not Support	Not Support
Custom process control	Support	Partially	Not Support	Not Support
Mailing template	Support	Support	Support	Support
Custom mailing rules	Support	Not Support	Not Support	Not Support

For the criteria that this thesis powerful/weak than other system, we would describe why we had the following evaluating result.

Setup time cost

When users constructed their paper-review system through the other systems, they had to setup the web server and database environment first and then install the system and configure the system. In the system of this thesis, users just need make their configuration online. They did not spend any time on the web and database environment setup.

Custom user profile

For different research group, the users' profile in their systems may be different because some information of user is not so concerned. By the help of frame system, we could make users define their own user profile definition.

Dynamic roles definition

Because other system thought that there were only 3 roles in a paper-review process, they thought that there is no need to generate other roles. However, by defining different roles, we could define more paper-review process complicate such as hierarchical reviewing process.

Knowledge sharing

For the other systems, different outputted paper-review systems were installed in different server. Moreover, the system configuration and the data were stored in the database together. Therefore, they could not support the knowledge sharing. That is, they could not use the configuration made by other user and modify it to generate a new paper-review system.

Custom process control

For MyReview system, it could only define the papers' judge-result in the output paper-review system. However, there were no dependency on the judge-result and the process control in MyReview system. Therefore, user still could not customize the paper-review process. In this thesis, by defining the paper state transition rules after each action in the paper-review process, users could generate their own paper review processes.

Custom mailing rules

For the other systems, they could define the mailing template and insert dynamic fields such as user's name or paper's title. However, except mailing to author at the time of paper was judged, they could only send mails manually. Based on the mailing rules in this thesis, users' paper-review system could send the specific mails automatically at the specific time.

Chapter 7 Conclusions and Future Works

In this thesis, we used frames and rules to model a paper-review/system. This thesis separated the knowledge of paper-review process from the real data in the paper-review system. In order to manage users' access control in the system easily, this thesis used the role-base access control model. Based on this model, the various anonymity control settings in the paper-review process could be managed too.

Compared with the traditional programming approach, the resources and the process control rules in the paper-review process were reconfigurable. For the organizers of the more and more conferences/journals, this thesis could help them define their own paper-review processes, rather than using the process defined by the system developers. On the other hand, there were still many open-source paper-review system projects could help user construct the paper-paper review system. However, to construct those systems, users had to prepare the environment of web server, database. It would be a time cost tasks. In this thesis, the only thing users have to worry about is configuring the paper-review process they want.

To help them configure the process easily, this thesis proposed a paper-review process authoring tool. This thesis provided the interaction mechanism between users and authoring tool as well as the default knowledge setting in different construct phases. It would reduce time cost on construction tasks. Most important, since this thesis separated the knowledge from real data and program algorithm in the output paper-review system, it could easily achieve the purpose of knowledge sharing when users loading the other users' configurations as the default knowledge.

In the future, this thesis would become a service on internet. The organizers of the conference/journal can apply a paper-review system and configure their own process. In order to be a widely used service, the functionality of this thesis will still be enriched. On the other hand, the ability of customizing should be concerned when this thesis become a service. Moreover, after more and more users generate the paper-review processes through this thesis, the ability of assisting users generate the process they want faster by the help of previous knowledge in this thesis may be a new research direction.

Reference

[1] Arzu Baloglu, "Cooperative Online Submission: Business Analysis and a Workflow Model," sitis, Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, pp.409-414, 2007

[2] Pradeep Gurunathan, Seethalakshmi Pandian, "A Novel Approach for Web-based Conference Management System," iccet, vol. 2, International Conference on Computer Engineering and Technology, pp.482-486, 2009

[3] Chun-I Fan, Ming-Te Chen, Lung-Hsien Chen, "Truly Anonymous Paper Submission and Review Scheme," ares, International Conference on Availability, Reliability and Security, pp.960-965, 2009

[4] Philippe Rigaux, "An iterative rating method: application to web-based conference management," SAC '04 Proceedings of the 2004 ACM symposium on Applied computing, 2004

[5] Richard Fikes, Tom Kehler, "The role of frame-based representation in reasoning," Communications of the ACM, vol. 28, no. 9, pp. 904-920, 1985

[6] Han Reichgelt, "Knowledge representation, an AI perspective, " Norwood, NJ: Ablex Publishing Corporation, 1991

[7] D.F. Ferraiolo, D.R. Kuhn "Role Based Access Control" 15th National Computer Security Conference, Oct 13-16, 1992, pp. 554-563, 1992

[8] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, Charles E. Youman, "Role-Based Access Control Models," Computer, vol. 29, no. 2, pp. 38-47, 1996

[9] Wu, D.C., Lin, H.Y., Tseng, S.S., Weng, J.F., and Su, J.M., "Constructing a Role Playing Interactive Learning Content Model," Ninth IEEE International Symposium on Multimedia 2007 – Workshops, 2007.

[10] Huang, Ching-Ya "A Study of Adventure Game Design for Disaster Prevention Education, " master thesis, National Chiao-Tung University, 2009

[11] IEEE Society, [WWW page] http://www.ieee.org/index.html

[12] ACM Digital Library, [WWW page] http://portal.acm.org/dl.cfm

[13] International Journal of Digital Learning Technology, IJDLT, [WWW page] http://ijdlt.org

[14] MyReview: open-source web-based paper submission and review system, [WWW page] http://myreview.sourceforge.net/

[15] Open Conference Systems, [WWW page] http://pkp.sfu.ca/?q=ocs

[16] Open Journal Systems, [WWW page] http://pkp.sfu.ca/?q=ojs

[17] ADL, “Sharable Content Object Reference Model (SCORM) 2004 2nd Edition,” http://www.adlnet.org/, 2004