

Implementation of a Visual Modeling Tool for Defining Instance Aspect in Workflow

Jianxun Liu, Zefeng Zhu, Yiping Wen
 Key Lab of Knowledge Processing and Networked
 Manufacture, Hunan University of Science and
 Technology, Xiangtan, Hunan, 411201, China
 Ljx529@gmail.com, zzfking@yahoo.com.cn

Jinjun Chen
 Faculty of Information and Communication
 Technologies, Swinburne University of Technology,
 Melbourne, Australia 3122.
 jinjun.chen@gmail.com

Abstract—The instance-aspect oriented workflow management system is to vertically combine multiple workflow activity instances and submit them for execution as a whole according to some batch or combination logics. It is inspired by the idea of aspect-oriented programming methodology and aims at improving the execution efficiency of business processes. Traditional workflow systems do not support workflow model with instance aspects. In our previous work, we have studied workflow instance modeling technology. This paper makes a research on the principles, methods and implementation of a workflow visual GUI tool for modeling instance aspects in workflow. It is based on an open source GUI tool, Together Workflow Editor, and makes some expansion in instance aspect functionality.

Keywords - Workflow; Visual Modeling Tool; Instance Aspect

I INTRODUCTION

Workflow is the automation of business processes in whole or partly[8,9,17]. A workflow management system (WfMS) is a computer system that manages and defines a series of tasks within an organization to produce a final outcome or outcomes. WfMSs allow you to define different workflows for different types of jobs or processes[17]. Workflow model is the abstract representation of workflow processes. In WfMSs, there are some core components, such as a computerized workflow model representation (XPDL), a visual modeling tool for defining workflow model, a workflow engine for enactment of a workflow process instance. There are many projects, products and literatures about workflow model or WfMSs [1-16] right now. However, most of current workflow models are only concerns about separation of process logics from functional logics in information systems. The process logic view focuses on the static and horizontal connections or relations between functional components in information systems or departments in an organization. They did not take into consideration the separation of batch or combination logics in multiple workflow instances, which we call instance aspect modeling, within a WfMS from process logics furthermore to improve execution efficiency. This instance aspect modeling in workflow concerns about the dynamic and vertical modeling of relations between workflow instances. In our previous work [8,9], we investigated a special instance aspect model, batch processing in workflow and the modeling of instance aspects in workflow using a partially space model. There is a need of a computerized model language and visual modeling tool for representation

and definition of instance aspects in workflow. Based on XPDL and an open source WfMS, Enhydra Shark, this paper presents a framework and implementation solution in detailed for the problem. We first extend XPDL to represent instance aspects and then design and implement the visual modeling tool, which is an expansion of open source product, Together Workflow Editor. A case study is analyzed at last.

II BASIC PRINCIPLES

A. Visual Modeling Technology

With visual modeling tools such as UML, Object Model can be expressed accurately, directly and intuitively, so as to make communications between developers or customers and developers more conveniently. The foundation of a visual modeling tool is the graphical representation of abstract model. Therefore, it must support the following functions:

- Draw Graphics and connect lines: in the process of workflow modeling, we use activities and the relations of activities to present a process.
- Provide withdrawal and resuming functions: GUI editors have to provide functions of withdrawal and resuming so that people can edit those activities more conveniently.

B. Together Workflow Editor

Together Workflow Editor is the first graphical Java Workflow Editor fully implementing WfMC (Workflow Management Coalition) XPDL-Specifications (XML Process Definition Language). Every WfMC compliant XPDL-File can be viewed, edited and saved either from a local/remote mapped file system / drive or via Wf-XML directly from WfMC compliant workflow engines likes Together Workflow Server / Enhydra Shark, Fujitsu or TIBCO Staffware. Because of easy operation of Together Workflow Editor, it let people define and check workflow process quickly.

C. XPDL

XPDL Specifications is a process definition language based on XML. The top level in XPDL is a package which is used as a data container. Figure 1 shows the class diagram of XPDL model. There are entities, connections, references and etc. in the package. The package entity contains workflow process definition, workflow activities, transition information, workflow participants, workflow application definition, workflow related data, and the system and environment data.

From the view that whether entities could participate process route directly, we divide the entities into two types:

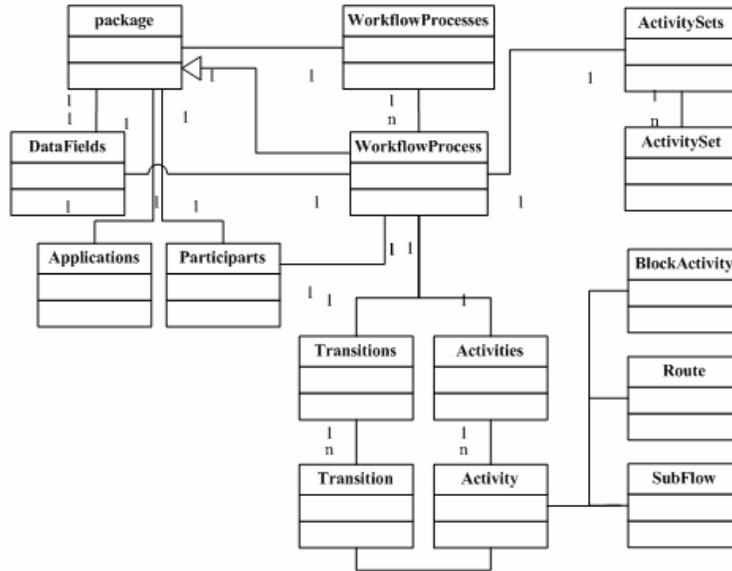


Figure 1. XPD model

static entity and dynamic entity. Static entities do not involved in the routing of process directly, and it defines the property information or static data. Dynamic entity can participant process route directly, and it defines nodes, routers and control condition.

Static entity includes definition head, process definition, workflow participant definition, workflow application definition, workflow related data, system data and data type.

Dynamic entity includes: activity, transition information. Each activity contains a logical, self-contained unit, it may be the smallest independent unit, and also may be assembled by a series of smaller independent modules of sub-process or block. We can also divide activity into two types: Atomic activity which contains normal activity, router activity and nesting activity which consists of block activity, sub-workflow activity. In workflow process, the relationship between activities is implemented by defining control transition information. Each transition contains predecessors, successors and the transition condition.

III WORKFLOW IA MODEL

A. Theory of IA in workflow

As a powerful tool in supporting business process, workflow has attracted enough attention from academic and industrial field. Unfortunately, existing workflow management systems do not provide mechanisms for separation of batch or group processing logic from process logics. Based on our previous work in batch processing in workflow [8,9], we use Aspect-Oriented Programming (Hereafter AOP) methodology to expand traditional workflow models so that they can model batch or group processing logics dynamically and vertically between workflow instances. We try to abstract execution requirements or constraints of activity instances to be processed in batch or group as an Instance Aspect (Hereafter IA). And then we embed this IA into workflow model.

Figure 2 shows an expansion in workflow meta-model to support this IA modeling. We add some AOP elements, such as Pointcut, Behavior PointcutExpression to traditional workflow model. IA is a group of instance activities which have the same instance characteristics, such as aspect of resource competition, batch processing or service priorities.

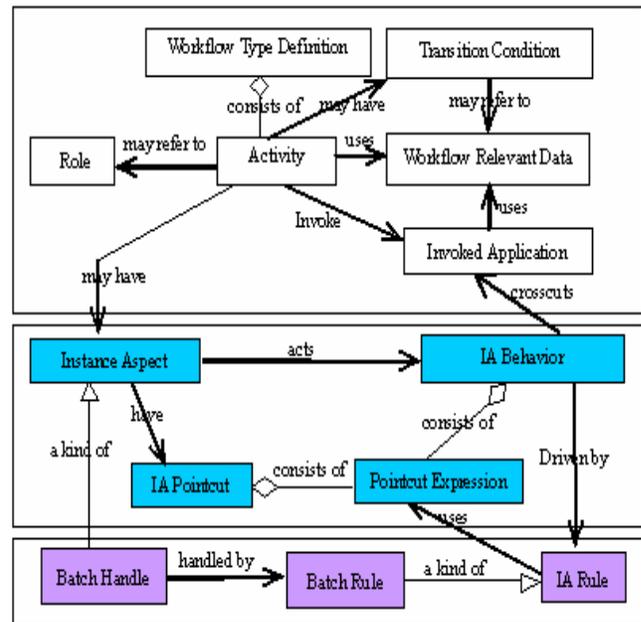


Figure 2. Workflow meta-model extension of aspect

Pointcut model follows the idea of AOP, which is a handling point of an IA. It usually has three styles: Incoming, Outcoming and Conjugated. Pointcut usually is a tuple with a PointcutExpression which is a logic expression, and Joint Point which is a connection of an IA and its Behavior.

Behavior corresponds to the Advice in AOP, which represents the action logics of this IA.

We can see from Figure 2 that the instance-based workflow model consists of two parts, i.e., traditional workflow model and IA description. Batch processing in workflow could be regarded as a special kind of IA in workflow model. In IA, the assembly and split of data flow in WfMSs are the key problem.

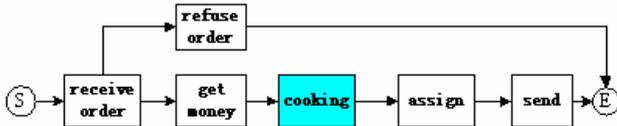


Figure.3. A process of car rental

Figure 3 is an example of Instance-based workflow, i.e., batch processing. It consists of six activities: “Receive Order”, “Refuse order”, “Get money”, “Cooking”, “assign”, “Send”. In normal situations, the instances of cooking are independent of each other, i.e., a cooking activity only serves for a customer order. However, if there are many customers who order the same dish at the same time, we can group these orders and cook together, which can not only save resources, but also improve the efficiency. In other words, several activity cases from multiple instances of the same workflow can be grouped vertically by certain rules, and it is the group being submitted for execution instead of each activity instance. This is the concept of batch processing in workflow as we proposed in [8] and it is a typical example of IA in workflow.

B. Extension of XPDL

XPDL is the representation language of workflow, but traditional XPDL does not support IA. So we have to make extensions to XPDL to denote IA in workflow. A new attribute IA is added. It is shown in Table I. In Together Workflow Editor, we use class XMLComplexChoice to manage XPDL and then we need to expand XMLComplexChoice component.

Attribute	Description
Id	IA id
Name	Name of IA
Type	Types of operation of IA
Operation	Operations of IA

IV DESIGN OF VISUAL MODELING TOOL FOR IA IN WORKFLOW

A. General Framework

The general framework of a visual modeling tool for IA in workflow is as Figure 4. It consists of 4 parts: User Interface (Hereafter UI) layer, control layer, model layer and XPDL storage layer. In UI, we take use of Together Workflow Editor’s UI layer and add some new interfaces, e.g., the IA node definition interface.

Control layer is used to monitor events in UI and creating node objects generated by users. Event monitor inherits all the events in Java. And it has extended some

events so that the tool can monitor the event more perfectly.

After event monitor gets the event happened, it transfer those event parameters to event processing module, and the event processing module will answer to those events. In event processing module, there are two major classes: GraphController and Handler. GraphController will handle some simple events such as how to answer the exceptions

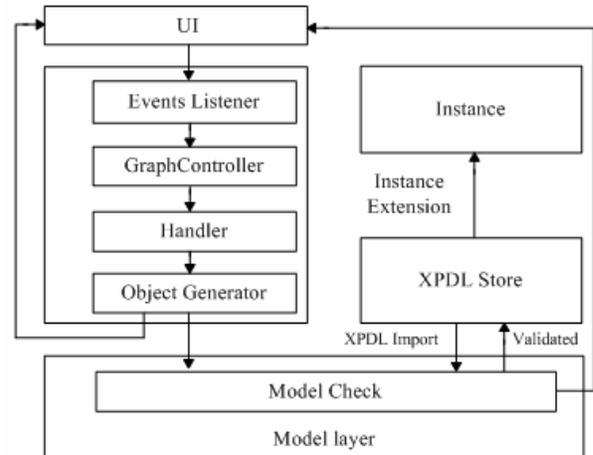


Figure 4. Framework for workflow modeling tools

while Handler will handle some instance event such as add activity, delete activity and set parameters.

To those instance nodes which we create in visual modeling workflow tool, if there is no need to store them in disk, it will be show in UI right away. But if we want to save the whole process in disk, the model layer will check the process. Only those processes which have been validated will be saved, or there would throw an exception.

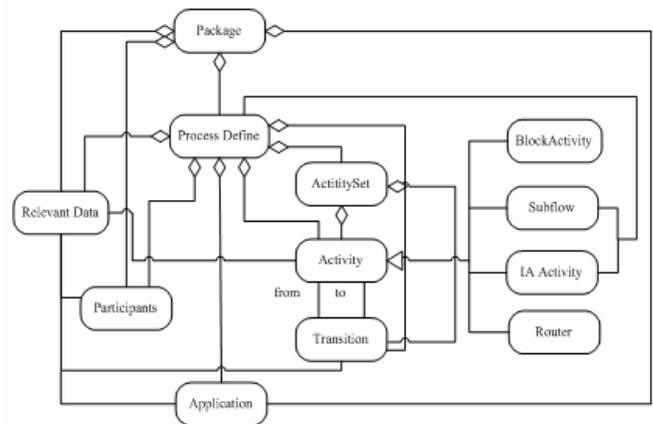


Figure 5. Entity relationship

Model check has two principles:

- Verification of general process: Any node in processes can not exist independently. A workflow process must have “Start” activity and “End” activity. Otherwise an exception will be thrown out
- Verification of IA in workflow: IA node object

could only be used by process object. It can not exist independently.

The main function of XPDL storage layer is importing and exporting the XPDL documents. Those processes which have been validated will be imported.

impact on system operation efficiency.

The function module which we have introduced before is the basis of the realization of model layer, and model layer are mainly used for model checking.

The class diagram of the entire model layer is shown in

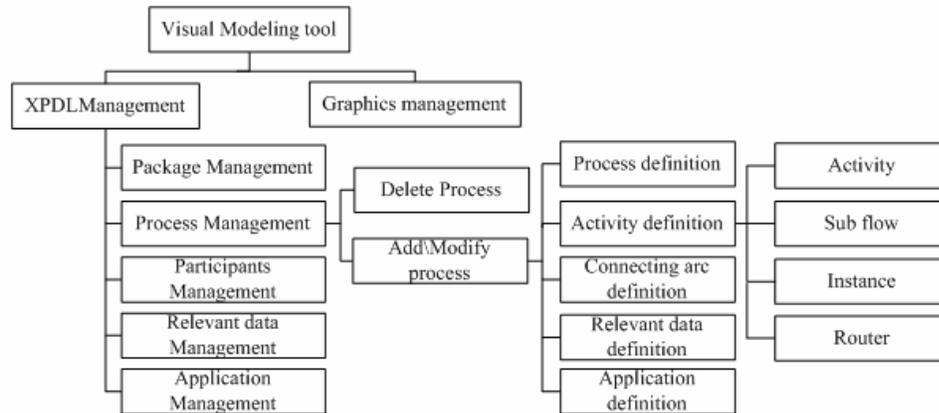


Figure 6. The module of system function

B. Extension of Entities

The entity relationship of Together Workflow Editor is shown as Figure 5. Each process package has one or more processes, participants, process relevant data and applications. Participants, process relevant data and applications can be defined in package or process. At the same time, the process also defines one or more activities, the activities set, as well as connecting arc. Activity set is a set of activities, and it is the composition of a number of activities. However, the structure of the original tool does not include the node of IA activities, so we extend the together Workflow Editor's activity. After the expansion, the node is divided into four types: block of activities, sub-processes, IA activities and routing activities.

C. Extension of Functions

Figure 6 is the function module diagram of IA workflow visual modeling tool. Since the top entity of workflow is XPDL package, in our design we make the XPDL management module and XPDL visual module to be the top function module.

The main job of XPDL visual management is to drag and drop activity in GUI interface. Process management module includes addition, modification and deletion functions. Compared to other ordinary visual modeling tool, IA visual modeling tool add an IA activity in activity definition module. IA activities responsible for the process with the same or similar function in the model.

V IMPLEMENTATION OF VISUAL MODELING TOOL FOR IA WORKFLOW

A. Class Diagram of Model Layer

The implementation of the whole system must starts from the model layer. Because model layer is essential in MVC Three-tier structure, and it will have an important

Figure 7, it includes three parts: basic process entities part, verification part and IA handle part. Basic process entities include all the entity type which is necessary in process. We add an IA entity in this part. Class Verification checks workflow process. XPDLBase is responsible for the import and export of XPDL documents. IAHandle extends from IAHandleAPI, and it is responsible for the invocation of operation primitives.

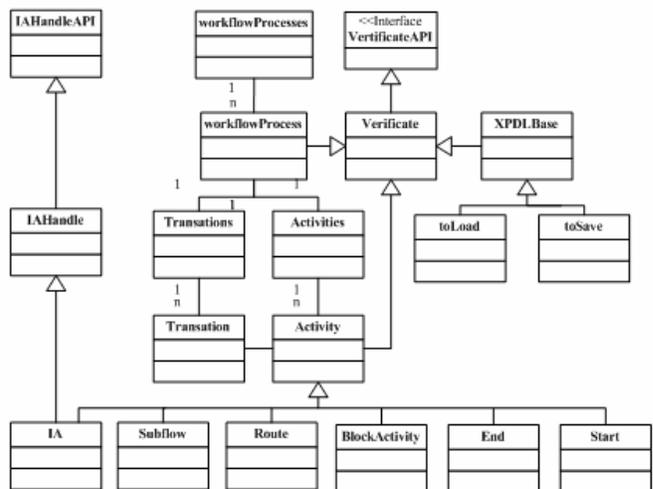


Fig.7 Class diagram of model

B. Implementation of Control layer

We have mentioned GraphController and Handler in the introduction of visual modeling tool's framework. GraphController in the control layer is responsible for the handling of simple events, such as exception. Handler is responsible for dealing with process events, such as nodes creation, deletion, and modification of parameters, which

TABLE II. THE FUNCTION OF HANDLER

Entity	Handler	Function	Entity	Handler	Function
Process	ProgressChangeId	Change process id	Activity	SetStartMode	Set start mode
	ProgressChangeName	Change process name		SetFinishMode	Set finish mode
	ProgressChangeLevel	Change process level		CopyActivity	Copy activity
	AddNewProgress	Add new process		DeleteActivity	Delete activity
	DeleteProgress	Delete process		PasteActivity	Paste activity
	CopyProgress	Copy process			
Activity	PasteProgress	Paste process	Transition	AddTransition	Add transition
	AddActivity	Add activity		ChangeTransitionId	Change transition id
	ChangeActivityId	Change activity id		ChangeTransitionName	Change transition name
	ChangeActivityName	Change activity name		DeleteTransition	Delete transition
	SetActivityPerformer	Change performer		SetTransitionFrom	Set start transition
				SetTransitionTo	Set finish transition

encapsulates all data operations and realizes some model function. Workflow modeling tools will achieve a lot of the Handler function, the most important function is to add, delete, and move the process entity in the canvas and the operation of data storage and modification. Table II shows some handler and their functions.

C. Implementation of IA functions

In the implementation of workflow visual modeling tool, the main work is to implement IA node. IA node must have a good definition, and make a relevant XPDL encapsulation,

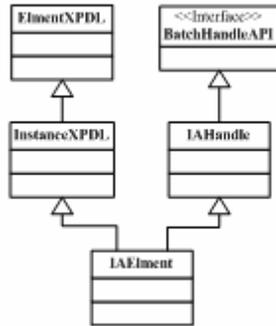


Figure 8. Architecture of instance

and let the system generate XPDL file automatically after the IA node have been called, the structure of IA function is shown in Figure 8. Class IAHandle is the core of the IA process. It is an extension from IAHandleAPI, which implements the control to IA process. Different instance activity calls different operation primitives, each type of IA activity extends from the basic class ElementXPDL, and all encapsulate corresponding XPDL labels, so when we have a visual operation, it could be able to generate the corresponding XPDL elements.

VI CASE STUDY

We use this visual modeling tool to define the workflow process with IA in Figure 3. The GUI definition is shown in Figure 9. The steps for defining the workflow process are as follows:

- a) Start modeling tools.
- b) Create a new process package, and create a new

process in process package and set a name for this process.

- c) Draw the process in canvas area.
- d) Set the IA parameters, in this case, cooking is the IA node.
- e) Save the XPDL documentation which is generated by system.

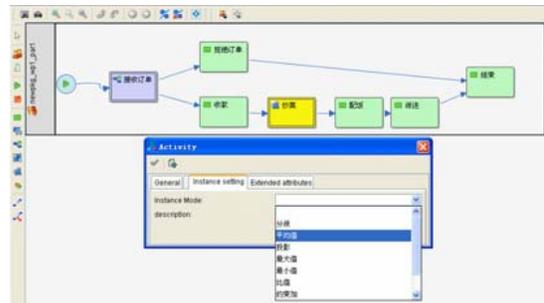


Figure 9. Configuration of properties for IA node

VII CONCLUSION

In our previous work, we studied the IA modeling, especially batch processing, in workflow. There is a need of a GUI tool to define workflow model with IA features. In this paper, we make a research on the principles, methods and implementation of a workflow visual modeling tool for defining IA in workflow. We first extend XPDL to represent IA and then design and implement the visual modeling tool, which is an expansion of open source product, Together Workflow Editor. The general framework of the tool, the extension to Together Workflow Editor and the Implementation are detail. Finally, a case study is analyzed.

ACKNOWLEDGEMENT

This work was supported by National Basic Research Program of China, 973 Plan, under grant no: 2003CB317007, Natural Science Foundation of China, under grant no: 60673119, 90818004.

REFERENCES

[1] K. Bill, D. Panagiotakis, F. George. Workflow Requirements

- Modeling Using XML , Requirements Engineering 2002:124~138
- [2] B.Z.Tao L. Hu, Kong.A Web Service Cooperative Scheduling Model for Workflow Management System. Professional Communication Conference, 2008 IEEE International
- [3] Borkowski, J.,Kopanski, D.,Tudruj, M.. Dynamic Workflow Implementation Based on Synchronizers. Software Engineering and Advanced Applications, 2006. SEAA '06. Aug. 29 2006-Sept. 1 2006 Page(s):448 - -455
- [4] B.Chaparro, G. A. Santiago, et al. Measuring quantitative dependability attributes in digital pub-lishing using petri-net workflow modeling. Proceeding of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, Indianapolis, Indiana, USA, September 29, 2006, 119–128.
- [5] H. A. Reijers, M. Song, B. Jeong. Analysis of a collaborative workflow process with distributed actors. Information Systems Frontiers, 2008
- [6] H. Shimosaka,T. Hiroyasu,Mitsunori Miki. Distributed Workflow Management System based on Publish-Subscribe Notification for Web Services. New Generation Computing,2007,25(4)
- [7] H.B. luo, Y.S. Fan, C. Wu. Overview of Workflow Technology, 2000, 11(7):899-907
- [8] J.X. Liu, J.M. Hu. Dynamic Batch Processing in workflow: Model and Implementation, Future Generation Computer Systems, Vol.23(3),2007:338-347
- [9] T. Li, J.X. Liu. PVS-WDBM: a Workflow Data Batch-Operation Model based on Partial Vector Space. Proc. of the 4th International Conference on Semantics, Knowledge, and Grid, IEEE Publisher, 2008, Beijing
- [10] X. Jin, J.L. Xu,. The Design and Implementation of XML-based Workflow Engine. IEEE, Volume 3, July 30 2007-Aug. 1 2007 Page(s):137 – 142
- [11] P. Sun, C.J. Jian, X.M. Li. Workflow Process Analysis Responding to Structural Changes, Journal of System Simulation,2008,20(7)
- [12] Q.L. Wu, J.Y. Wang. Design and Implementation of J2EE Framework-Based Workflow Management System, Computer Technology and Development,2008, 4(33) Reijers, H. A., & van der Aalst, W. M. P. (2005). The effectiveness of workflow management systems: Predictions and lessons learned. International Journal of Information Management, 56(5), 457–471.
- [13] R.Z. Sun, M.L. Shi. Schedule of Activity Instances in Workflow Management System, Chinese Journal of Computers, 2005, 16(3):400-406 (in Chinese)
- [14] S.H. Zhang, Y.W. Zong, et al. Workflow-oriented grid service composition and scheduling. Proceedings of the International Conference on Information Technology: Coding and Computing, Washington, USA, April 2005, vol.2, 214–219.
- [15] T. Li, J.X. Liu. Research on Aspect-Oriented Modeling and Application, Computer Technology and Development, 2009
- [16] WfMC, Workflow process definition interface-XML process definition language, WfMC-TC-1025, 2002
- [17] X.Y. Shao, P. Jiang. et al. Workflow modeling for virtual enterprise: A petri-net based process-view approach. Proceeding of the 10th International Conference on Computer Supported Cooperative Work in Design, Nanjing, China, May 3, 2006, 1213–1218.
- [18] Y. Wang. A relation-based workflow model for planning management system. Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. Volume 1, 12-15 Oct. 2008 Page(s):93 – 96
- [19] Z.J. Xiao, Q.C. Huang ,Qi Chen,Y.G. Liu. The syntactical analysis of workflow model with loop structure. Systems, Man and Cybernetics, 2004 IEEE International Conference, 2004 Page(s):4911 - 4916 vol.5
- [20] Z. Gan, M. Chen, Modeling & Scheduling of Data Flows in Workflow of Activity Instances, CAD&CG, 2008, 20(5)
- [21] Z.F. Zhu, J.X. Liu . Analysis of Together Workflow Editor, Microcomputer Applications, 2008.
- [22] Z.Y. Peng, Y. Luo, Z. Shan, Q. Li. Realization of Workflow Views Based on Object Deputy Model. Chinese Journal of Computers, 2005, 4(28) (in Chinese)