A WORKFLOW MANAGEMENT SYSTEM FOR DYNAMIC ALLIANCE

Zhiqiang Gu[1], Gangyan Li[2], Kunpeng Wang[2]

1Science and Technology Office, Wuhan University of Technology, Wuhan, P.R. China; Email: gxkwhut@mail.whut.edu.cn. 2School of Electronic and Mechanical Engineering, Wuhan University of Technology; Email: gangyanli@mail.whut.edu.cn, robbie82@sohu.com.

Abstract: In order to support operation recombination of enterprise, based on analyzing some important problems, including flow start-up, flow navigation, resource subsequent control, activity performance and task management, a workflow system support dynamic alliance was designed. A resource conflict resolution strategy based on queue and request in the process of workflow was introduced. Strategies of occupancy and release resource in inaction were introduced. The main module and structure of database of the workflow management system were designed.

Key words: Dynamic Alliance, Workflow Management System, Control, Database.

1. INTRODUCTION

Workflow management is a kind of rising technology span subjects. It refers to many subject fields such as information, management, psychology, and so on. At the same time, it can be applied in many fields such as e-business, product design and remote-distance education and so on. Researching the technology of workflow management thoroughly is significant to improve informational degree, efficiency and competitive abilities of enterprise and government department. Based on analyzing some important problems, including flow start-up, flow navigation, resource subsequent control, activity performance and task management, a workflow

This research is funded and supported by the National Natural Science Foundation of China (NO.E0550335020), the National Key Technologies R&D Program of China (NO.2003BA103C) and the Key Technologies R&D Program of Wuhan City (NO. 20043007077-02).

Please use the following format when citing this chapter:
Gu, Zhiqiang, Li, Gangyan, Wang, Kunpeng, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 950-959.

system support dynamic alliance was designed. A resource conflict resolution strategy based on queue and request in process of workflow was introduced. Strategies of occupancy and release resource in inaction were introduced. The main module and structure of database of the workflow management system support were designed.

2. OBJECT DESIGN OF SYSTEM

The Object-oriented method (OO) is applied to encapsulate all the data in the form of object. All services offer their exterior interfaces in the form of objects.

2.1 Object of system

The objects in the system are as follows: workflow definition, node definition, resource instance, user instance, role instance, activity instance, application data and mission.

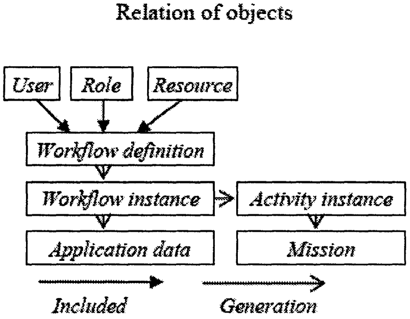

Figure 1. Mapping relation

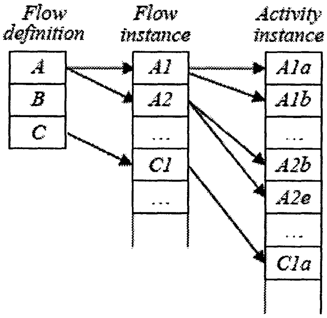

Figure 2. Mapping relation

Relations between all data objects in the system are illustrated in Figure 1. Making the work flow run normally according to the defined operation model is the main aim of workflow management system. A great deal of activity instances can be derived from a flow instance. (There are several nodes in the workflow model). At the same time, a great deal of flow instances can be derived from a workflow instance so that an enterprise can deal with several same businesses at the same time.

Figure 2 shows the mapping relation between flow definition, flow instance and activity instance.

3. DATABASE DESIGN

The relational database management system is applied to save the data running in workflow. Relational database is based on a solid relation theory foundation. There is a series of design methods in relational database design. Furthermore, there are various commercial database softwares suitable for developing various software systems. Figure 3 illustrates a logic model of system database based on above discussion.

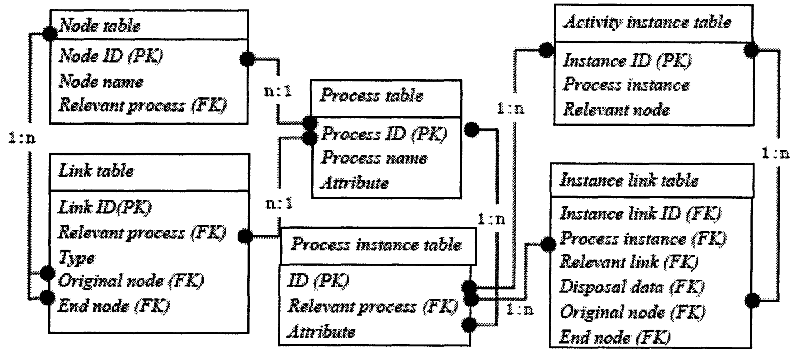

Figure 3. Logic model of database.

In order to realize that data are separated from program, database is applied to save data. A mass of running information are saved in database according to the designed relation. The design information of main table of the database is as follows:

Process table: process table includes workflow ID, workflow name, workflow content and other attributes.

Process instance. Process instance table logs the flow running in system. Instance information of flow is saved in database. A workflow definition can include several instances.

Node table.

Activity instance table. It includes instance ID, instance name, Input data.

Variable table. It logs all variable of all nodes. Figure 4 shows data model of system.

Variable instance table. Variable instance is the object encapsulated from control data produced while flows are running. In this system, data ID, relevant flow, relevant activity, data content are encapsulated to a variable instance.

Variable type table. Control data are used to navigate and assign. The types of control data are defined in advance. We can choose the types from the table when we design the variables of nodes.

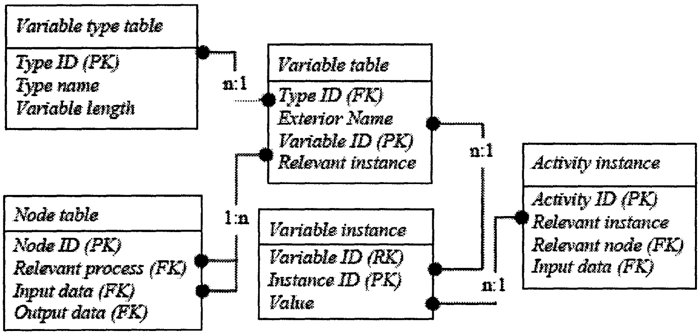

Figure 4. Logic model of data.

Role. Role which includes someone's rights and function is a set of a group right. Role is an object which is used to cooperate among department. Member should have different roles in different projects. A task is assigned to a role while a workflow is defined.

Resource. Resource is composed of equipments, workshop and other hardware equipments. It should include human resource especially.

People. People include all employees of enterprise. Attributes of personnel are as follows: name, age, length of service, department, strong suit and so on. Each employee has an ID as only identifier.

Task. Task is a command to relevant employee sent by activity instance running. Employee can get task information from task manager after login. Task is generated from activity instance.

Alliance table. The table is used to memorize information of alliance in that enterprise joined. Workflow engine is associated with enterprise's name according to the data in the table. Local enterprise chooses enterprise to communicate with by name of enterprise. The attributes of the table include name of enterprise, address of enterprise, and type of enterprise, about enterprise and so on.

4. DESCRIPTION OF WORKFLOW ARITHMETIC

4.1 Design of system class

It is necessary to design the detail part of class. In order to decrease length of code and reuse existent code, the design function of Object Oriented is introduced to encapsulate function models of system to classes.

Running process of system is as follow:

The chief control workflow system marks out flow and decomposes task. Then, it assigns task unit to relevant enterprise workflow system.

Initializing run environment of alliance, synchronizing all subsystem workflow system, preparing for communication among system.

Administrator starts up a work flow.

System flow begins using activity navigation service. Activity data object is initialized. Process control parameters are modified. Automatic flow nodes are executed. Activity is transferred to another workflow system according to assignation. Then, next activity is navigated.

Manpower atom activity execution: all data of activity are initialized. All need resources are configured. Task notice is created and added in task list.

Sub-process execution: Flow is dynamic decomposed and initialized. Then, flow begins.

Workflow system watches flow run, activity execution, and task submission and so on.

4.2 The start up of flow

There are three modes to start flow up: administrator start, user start and automatic start. Flow is started up by flow watching model. Watching model initializes flow instance according to the flow definition (see figure 5.).

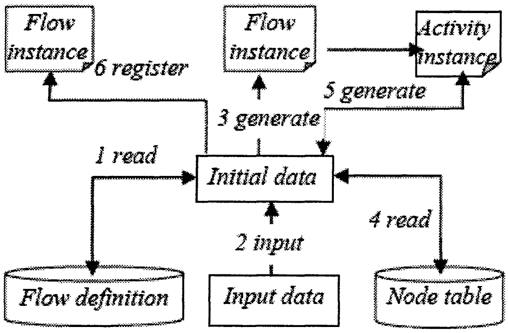

Figure 5. Initializing flow.

The arithmetic of flow Initialization is described as follows (see figure 6.):

Choose a workflow definition from workflow table. If finished, next.

Create a flow instance according to workflow definition and provide input interface. Users must input dynamic attributes of the instance.

Initialize activity instance according to node definition and prepare data boxes and variable instance.

If all nodes and links are initialized successfully, run state of flow will be configured and the instance will be registered in global information.

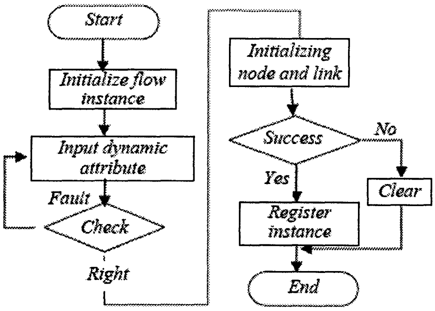

Figure 6. Process of initialization.

4.3 Flow navigation

System needs to confirm the next execution activity according to data, instance and link running in flow when an activity finished. Activity assigning engine answers for navigation of activity (see figure 7.).

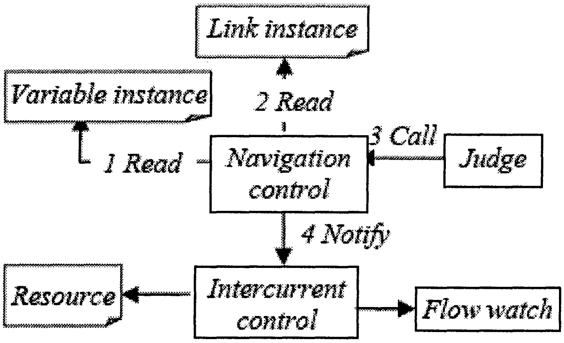

Figure 7. Flow navigation.

The arithmetic of navigation flow is described as follows:

Read next link, variable and judgment expression according to finished activity.

Get next activity calling judgment model.

Endow the activity with dynamic attributes and examine intercurrent control.

Intercurrent control model examines activities which are able to begin according to assignation rules. Flow watch model is notified to begin to execute the activity.

4.4 Intercurrent control

Some instances of a model or different instances of different models maybe exist at the same time while system is running. Furthermore, there will be competition for resources among instances. Intercurrent control model is used to deal with this problem.

4.4.1 Intercurrent resolution

We introduced the preemptive resolution. Each resource has an activity competition queue. When the priority of activity in queue changed, the queue must be adjusted. Activities which have the same priority hold the resource according to the order. Figure 8 shows an activity wait queue of resource.

In order to avoid deadlock (a state that a suspended activity remains hold the resource) while activity wait and execute, we introduced a method to adjust activity. The method is as follows:

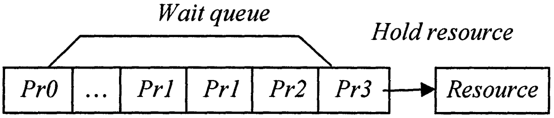

Figure 8. Activity resource wait queue.

Priority modulus (Pr) is set 0 when the activity is suspended. Pr=0.

Priority modulus must be counted and reset when an activity starts or resumes.   Pr=wait modulus+ hold modulus +import modulus

System count waiting modulus of activity timely and adjust the queue.

According to above discussion, intercurrent control model should count wait modulus of activity in resource wait queue when the activity requests resource. Executing activity will be deprived of resource he held for lessening of wait modulus caused by holding resource long time excessively. Figure 9 shows the expressions of wait modulus.

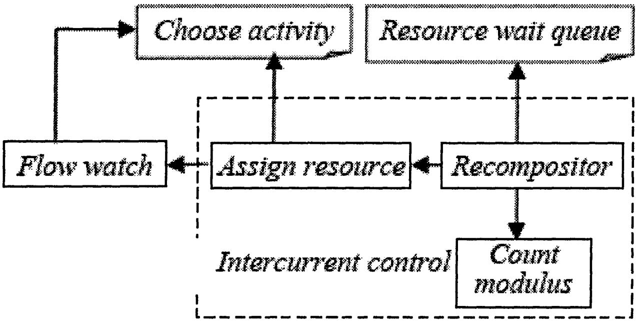

Figure 9. Process of Intercurrent control.

4.4.2 The arithmetic of intercurrent control

The arithmetic of intercurrent control is described as follows:

Count waits modulus of activities in resource wait queue.

Count waits modulus of activity hold the resource.

Choose an activity that has the most wait modulus.

Assign resource to the resource and notify watch model that the activity begins.

4.5 Execution control of activity

An instance of activity has six states: initialization, execution, ready, wait, suspended and achieved. Initialization state means period between creating instance and executing it. Ready state means that the instance has gotten all need parameters and other conditions. Wait state means that an activity is waiting for a time or resources to execute. Suspended state means that instance is suspended when there is suddenness. Figure 10 shows Activity instance state transition.

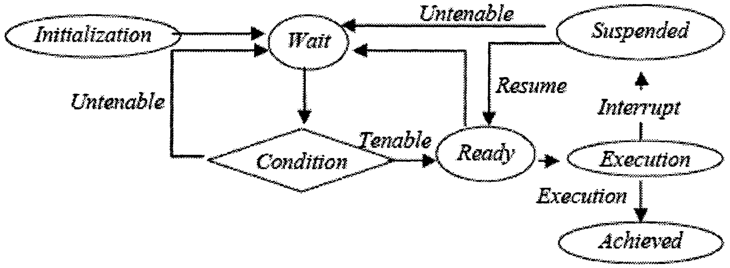

Figure 10. Activity instance state transition.

Activity may need various resources while it is executing. So it is state transition based on multi-resource and preemptive model.

The arithmetic of activity state control is described as follows:

Search resource that activity needs and put the activity in wait queue of the resource.

Execute the activity which is ready and set right state of activity according to its execution state.

Examine the suspended activity whether it hold resource. If it is deprived of resource, its state will be set wait state.

Examine the ready activity whether it hold resource. If it is deprived of resource, its state will be set wait state.

4.6 Task management

As a manpower activity, system needs to translate activity to customization form and gives notice to user as task message. Task management model provides of function which manages task list of user. The function gives notice to flow watch model and instance manager to run flow control.

Task is oriented user. Generalized user contains some resources needed in activity execution. A manpower activity means that the activity needs people (or resource) to execute it. So Task list is manpower resource wait queue.

Figure 11 shows four submodel of task management model and relation among them. Execution submodel executes manpower activity wholly. The process is as follows:

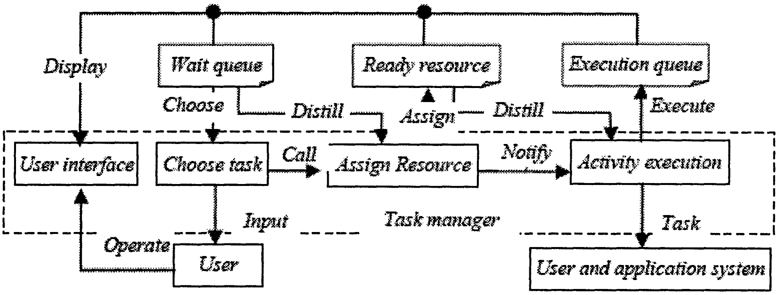

Figure 11. Activity instance state transition.

Distill an activity of manpower which do not begin or suspend.

Search other resources needed.

Choose an activity which hold all resources from flow ready queue to execute if it needs user's input data, user inputs data through user interface provided by task management.

If activity needs one more different type resources, resource wait of activity is very complex. The activity may get all resources at the same time for ever, which is called deadlock of resource.

In order to solve the deadlock of resource, we designed a resolution based on resource message. It is means that user can modify the position of activities in resource wait queue, force-deprive of held resource of activity. The mechanism of solution is defined as follows:

The deprivation is unconditional. But when a resource is held by other activity in un-deprivation pattern, user can not hold the resource until the activity achieves. Un-deprivation hold pattern is designed for activity which is not allowed to be interrupted while is executing.

In situation that activity needs several people to join in, user can send messages to other people of the activity. The activity begins after all people replied.

In situation that activity needs various hardware resource (all resources except people), user cannot operate for no user interface. System runs intercurrent control in background. The arithmetic is used to avoid deadlock. System force-assigns other resource to activity when the activity hold a resource.

5. CONCLUSION

A novel workflow system support dynamitic alliance was designed. The realization method of the system was introduced. The memory formats in database of workflow definition and workflow instance were designed. Intercurrent conflict resolution and harmony strategy for several resources request were introduce.

6. REFERENCES

1. Guangtai Min, Tao Feng, *Dynamic Flexible Workflow Management System Based on Middleware Technology*, Computer Integrated Manufacturing System, 2002, 8(8), pp.655-659 (in Chinese).
2. Tao Feng, Qian Wang, *An Approach to the Flexible Workflow Management Based on Doc*, Computer Application, 2002, 22(2), pp.20-23 (in Chinese).
3. Tong Lin, Zaizhong Zhang, Hongchen Li, *Process Models in Workflow Management System*, Computer Application, 2002, 22(6), pp.13-16 (in Chinese).
4. Yongquan Fan, Zhujun Wang, Yujun Luo, *A Workflow Management System Based on relation Database*, Computer Application, 2002, 22(7), pp.75-77 (in Chinese).
5. Bussler C., User mobility in workflow-management systems, *Proc Telecommunications In formation Networking Conference*, Melbourne, 1995.