The Design of Cooperative Workflow Management Model
Based on Agent

Zhu Yunlong  Li Hongxin  Xue Jinsong  Wang Hongtao
(Shenyang Institute of Automation,
Chinese Academy of Sciences  110015, P.R.C.)
E-mail: yZhu@mail.sy.cims.edu.cn

Abstract

This paper analyzes the state of art in workflow management system and then presents cooperative workflow management model based on agent (COWBA), which comprises three levels of architecture. The first level is process definition level that defines properties related to activities; The second level is workflow enactment service level that includes instantiated agent, cooperative agent and activity agent; The third level is user interface and invoked application level. Utilizing agent and object-oriented technology, cooperative agent, instantiated agent and activity agent can be built on the basis of object-oriented workflow meta-process model. Therefore, the ability of self-organizing, self-learning and coordinating has been enhanced.

Keywords: Workflow  Agent  OO  Cooperative model

1. Introduction

Workflow management is one of the most rapidly developed technologies in computer application field in recent years, the main feature of which is realizing automation in the combination process of computer interactive events with humans. It is extensively applied in office automation (OA), insurance and banking businesses firstly and considerable achievements have been made. With the widely accepted managing concept of BPR[2][3] (Business Process Reengineering), Workflow management system focusing on process managing is accepted and applied gradually, the relevant software such as Flowmark[4], Flowpath[5] and Inaction have already been presented. These products not only inherited task process control features of early OA systems, but also made predominant progress in user interface and decentralized coordination using present technology advances. Workflow Management Coalition[6] issued the reference model of workflow management system, providing the standard interface for the interoperation of various workflow management softwares.

However, some problems still exists in present workflow management systems such as confined applying fields, low flexibility, and inability to cope with exception and to estimate the performance of process operations, as well as unable to realize process reengineering. Especially in production, present workflow management systems do not provide the function of production scheduling and optimum control and are lack of self-organizing, self-learning and cooperative abilities.

Consequently, WfMS will definitely trend towards an intelligent system with self-organizing, self-learning and coordinating abilities. In this paper, a Cooperative Workflow Management model based on agent (COWBA) is proposed. It comprises three levels of architecture, the first
is process definition that defines properties related to activities, the second is workflow enactment service that includes instantiated agent, cooperative agent and activity agent which are built utilizing agent and object-oriented technology. The third is user interface and invoked application. The COWBA will enhance the self-organizing, self-learning and coordinating abilities of the system.

2. Basic concepts

2.1 Meta-process model

Essentially speaking, workflow is an automatically running process among multi-participants to transfer documents, information or tasks according to some previous defined rules to realize an expected goal or to promote this goal to be realized. Therefore, the process definition (i.e. workflow model) is the basis of workflow management system.

Nevertheless, from abstract point of view, the processes composed with activities, however complicated, have same attributes and characteristics. Here we integrate the objects with attributes related to activities in workflow model to form a whole object. It is referred as meta-process model, whose kernel is activity. The involved objects in this model are workflow type (wf_type), role, control data (ctrl_data), relevant data, transition conditions, invoked application and application data (see Figure 1). The basic object group in meta-process model is as follows:

\[
<\text{meta\_process}> :: = <\text{wf\_type}, \text{activity}, \text{role}, \text{wf\_relevant\_data}, \text{wf\_ctrl\_data}, \text{wf\_ctrl\_data}, \text{transition\_conditions}, \text{app\_data}, \text{invoked\_app}>;
\]

Each object is composed of a group of parameters, of which:

- \(\text{WF\_type} :: = <\text{process\_name, version\_no, start\_condition, end\_condition, ctrl\_data, security, ...}>;\)
- \(\text{Activity} :: = <\text{name, type, pre\_condition, post\_condition}>;\)
- \(\text{Role} :: = <\text{name, org\_entity}>;\)
- \(\text{WF\_relevant\_data} :: = <\text{data\_name, data\_path, data\_type}>;\)
- \(\text{WF\_ctrl\_data} :: = <\text{data\_name, type, exec\_parameter, location}>;\)
- \(\text{Invoked\_app} :: = <\text{name, type, exec\_parameter, location}>;\)

The meta-process model can essentially and effectively describe the information relevant to an activity, such as pre/post condition for the operation of an activity, data structure, control parameters, invoked application and operation mode etc. This information will be inherited by related agent model.

![Fig. 1 Meta-process model](image)
2.2 Instantiated Agent

Instantiated agent is designed mainly to solve two problems: one is the link of process model definition with workflow management system; the other is the creation of activity agent.

In workflow management system, the managing of activity is actually the managing of human behavior. Meta-process model only abstractly describe the characteristic (attributes) of process objects. It is a logical description rather than an actual activity. In other word, it is not instantiated.

Instantiated agent creates the activity set of instance application program according to meta-process model and practical business process. It inherits the attributes of meta-process and dispatches actual business activities to each organization through the interpreter of itself. Thus, instantiation is accomplished and activity agents are sorted and created by function. For example, from the view of function, product design process can be divided into product design, process planning and computer aided manufacturing (CAM). The system will accordingly build product design agent, process planning agent and CAM agent by the instantiated agent. Each activity agent takes charge of related concrete activities.

2.3 Activity agent

Activity agent is created by instantiated agent according to functional division. Each type of activity agent posses certain similar features such as similar function or roles. Because of the complexity and variety of concrete activities in workflow management system, the administration of the system is a heavy task. However, the difficulties in coordination and administration will be drastically simplified by dividing activities into different type in accordance with function.

Activity agent has some features of its own on the basis of inheriting the attributes of instantiated agent, mainly the methods of scheduling, activity monitoring and proxy requests.

2.4 Cooperative agent

Cooperative agent is designed mainly to coordinate activities that are inter-related to each other in activity agent, ensuring all of activities executed orderly.

3. System structure

Workflow management system based on agent is composed of three layers. The first layer is process definition layer. It is an abstract description of the process. The meta-process model is defined here as the core of definition layer. The second layer is workflow enactment service, it is composed of instantiated agent, cooperative agent and activity agent. The instantiated agent takes charge of transforming the process in real world into formal language and creates correspondent activity agent by function. The third layer is user interface layer.

Because cooperative agent and activity agent takes master/slave structure, activity agent realizes its function in system under the monitor of cooperative agent, and passively executes the tasks assigned by cooperative agent. It monitors the running states of itself and feeds back the monitoring information to cooperative agent. Its cooperative abilities mainly are inner activity cooperation and autonomy. It does not have the ability to cooperate with other activity agents.

4. Cooperative Mechanism
COWBA (cooperative workflow management based on agent) is a multi-agent which collaborates with others to accomplish a goal in a dynamic environment. First, instantiated agent creates various activity-agent according to some properties defined in meta-process such as workflow type, function, organization/role, and then decompose concrete business processes into practical activities related to activity-agent. Each activity-agent takes charge of managing, scheduling activities and communicates with cooperative-agent.

Because cooperation among activities involves the global goal of a system, post-coordination will not assure the optimized effects of general cooperation. Therefore, activity-agent must keep the progress of activities in mind. It should be able to predict and judge the activity development, and should be able to commit the plan for next step made by real-time running states of activities to cooperative-agent for arbitration. It can effectively avoid passive situation resulted from the cooperation conducted by cooperative agent when conflicting.

Cooperative-agent cooperates with activities of activity-agent. When some exception emerges such as resource conflicting, activity-agent commits it to cooperative-agent and requests it to coordinate activities. The results from cooperative-agent will be issued to each activity-agent. Because cooperative agent and activity agents take advantage of master/slave structure, the cooperative difficulty is reduced.

Fig. 2 Cooperative workflow management framework based on agent

Cooperative-agent is a core of COWBA, it consists of knowledge library part and control part.

4.1 Knowledge Library
Knowledge library concretely describes the knowledge needed in coordination, including strategy knowledge library, activity states library and resource library.

Strategy knowledge library, including method library and model library, describes the coordinating strategy of the activity agents. Coordinating strategy adopts multi-agents planning method, i.e., making correspond plan according to the object of respective activity or sub-object of global objects. The detailed operation lies in the activity plan exchanging of respective agents that need coordination or interaction. The occurrence of conflicts is avoided by making correspondent rule-modifying strategy in line with the plans of two parts. The mutual profitable relationship of each other are sufficiently exploited to realize global coordination.

Activity state library records the activity state of each concrete business activity in activity agent, which includes states as follows: initiate, resume, suspend, terminate, abort, active, complete, running and waiting time required for activity running.

Resource library includes static resource library and dynamic resource library. The former describes the distributing situation of the resource of actors and devices related to the activities in the system, as well as the time for the running of concrete business activities. The latter records the resource consumption of current activities.

4.2 Control part

Control part is composed of Cooperative Reasoning Machine (CORM), task execution controller, activity monitor, and message issuing mechanism respectively.

Cooperative reasoning machine decides interdependence of the participants in activity agent and cooperative degree among activities according to the local goal of each activity and consumed resource. Then, CORM generates a set of results by cooperative objects, the running states of current activities and the relevant models and methods in strategy knowledge library. After that it comprehensively judges the analyzed results and selects suitable one as commonly accepted objects to execute.

Task Execution Controller assigns the coordination results to each activity agent for compulsory execution. Activity Monitor monitors the execution of activities in each agent and stores the running states and consumed resource in activity states library and resource library respectively.

Message Issuing Mechanism takes charge of request, issuing and communication of activity agent.

The cooperative agent model can be referred to figure 3.

In order to enhance the ability of self-organizing, self-learning and adaptive, cooperative agent will derive some parameters from practical activities of each activity agent and modify the strategy knowledge library. In the meanwhile, model library also provides performance model to estimate whether a practical business process is rational. The results will feed back instantiated agent that adjusts some properties of meta-process model. Each agent inherits these properties. To a certain
extent, the system has the ability of self-organizing.

5. Conclusion and future work

We present a framework of cooperative workflow management using object-oriented technology and agent. It make the system have the ability of self-organizing and self-learning that will be useful to support business process reconfiguration and improvement. Next, our study will focus on how to reengineer process rationally and what estimating index is suitable.

Reference