

Research and Implementation of Key Technologies in Multi-Agent System to Support Distributed Workflow

Tianheng Pan*

Wuhan Institute of Bioengineering, Wuhan 430000, China

*Corresponding author e-mail: 116493626@qq.com

Abstract. In recent years, the combination of workflow management system and Multi-agent technology is a hot research field. The problem of lack of flexibility in workflow management system can be improved by introducing multi-agent collaborative management. The workflow management system adopts distributed structure. It solves the problem that the traditional centralized workflow structure is fragile. In this paper, the agent of Distributed workflow management system is divided according to its function. The execution process of each type of agent is analyzed. The key technologies such as process execution and resource management are analyzed.

1. Introduction

Distributed workflows are developed from centralized workflow. Centralized workflow is too dependent on the central server. The whole workflow system will fail once the server has an error. Distributed workflows solve this problem. The distributed system structure can provide the necessary communication capability for the components of the workflow, and help the workflow management system to realize the dynamic scheduling of tasks. However, the distributed workflow still has the problem of poor flexibility and lack of interactivity. In order to solve the problem of distributed workflow, the concept of agent is gradually adopted by distributed workflow management system. We build a multi-agent platform that supports distributed workflow management. Agents are divided into various roles by function, and they assist each other. A workflow business process can naturally be seen as a number of agents capable of accomplishing their own tasks and scheduling tasks in concert.

2. Design of Agent in Distributed Workflow Management System

The most important part of the system is to introduce the agent to replace the important module in the traditional system. The functions of each part of the system are accomplished by designing various agents with different functions. Each agent negotiates through message passing. According to the implementation function, the workflow system based on mobile agent designs the following agents:

(1) Manage agent: The management agent is the most important management organization in this system. It directly or indirectly deploys and manages other agents in the system. This management activity does not take the form of a simple command, but rather a message negotiation. This ensures a certain degree of autonomy to other agents.

(2) Case agent: It is responsible for the execution of a process instance. It generates a task agent through the interpretation process definition, and controls the execution and scheduling of the task agent. When a workflow process instance is started, the system creates a case agent that corresponds to it. The



case agent completes specific tasks by generating task agents, and processes the process instances through the sequence of activities defined in the process definition and the actual information. In the implementation of the process instance, the current node state of the workflow should be evaluated, and the task agent should be generated to execute the next node.

(3) Task agent: The case agent creates a unique task agent that is responsible for executing each activity in a process instance. The case agent manages the task agent. The task agent interacts primarily through messages with case agents. The task agents are responsible for handling activity instances, and most of the work is done by interaction with resources.

(4) Resource agents: Most activities in a workflow management system are accomplished by interacting with resources. The resources involved in the workflow are diverse. Resources can be people, software objects (such as component objects, applications, etc.), hardware devices, and so on. The system creates a corresponding resource agent for each resource. The system is connected with resources through resource agents.

(5) Resource management agent: This agent is mainly used to manage resource information and resource agent, and provide relevant information to the agent who wants to obtain the resource information. Other agents must first query the resource management agent if they are to interact with a resource. Each resource agent is registered with the resource management agent when it starts running. The resource agent is deleted from the resource management agent when the resource agent is not needed.

3. Process Execution Analysis

Process execution is performed and managed through the workflow engine. The task agent is used in the system to process each activity in the process execution. The workflow management system is executed by transforming the process definition into the corresponding instance object.

The main work of the workflow engine is to establish a workflow table and a workflow tracking table for storing process instance information in the database. In the process of the system work, the workflow engine will find the current processing node and the next node in the workflow table. Determine whether the precondition of the current node is satisfied. If the precondition of the current node is satisfied, the task agent is allowed to process the current node. After the current node is processed, the next node becomes the processing node.

The process execution of multi-agent system supporting distributed workflow is shown below

(1) First, start the management agent. The startup of the management agent can be triggered by the system administrator and the general user. It mainly completes the initialization of the system, such as initializing the relevant data of the system, loading the necessary process definition model, creating and managing the required case agent, and so on.

(2) When the management agent receives a request to create a new workflow, it creates a case agent corresponding to the process, and then to pass the parameters of the case agent need to it, such as the corresponding process definition, workflow relevant data, and so on. When the case agent is initialized, the process is started.

(3) After starting a process, the case agent should explain the process definition of the process, generate the process path and corresponding strategy, and create each task agent in sequence according to the process path. The specific definitions of activities in the process model are passed to the task agent as parameters. Task agent is responsible for each specific task.

(4) The task agent starts the corresponding activity after it gets permission to start executing. During activity execution, the task agent can interact with a user or invoke a resource. The task agent is also responsible for communicating with each resource agent, asking whether the resources required by the activity are currently idle and available, thereby obtaining and working together with the resource.

(5) After executing its own task, the task agent returns the results to the case agent. The case agent determines the next step of the execution of the activity according to the process's execution rules, and creates the corresponding task agent to continue. If no activity is found that can be performed, determine whether the process is over and report the results to the case agent. If the process is over, the case agent logs out of itself after all the actions have been performed.

The process execution sequence diagram is shown in Fig. 1.

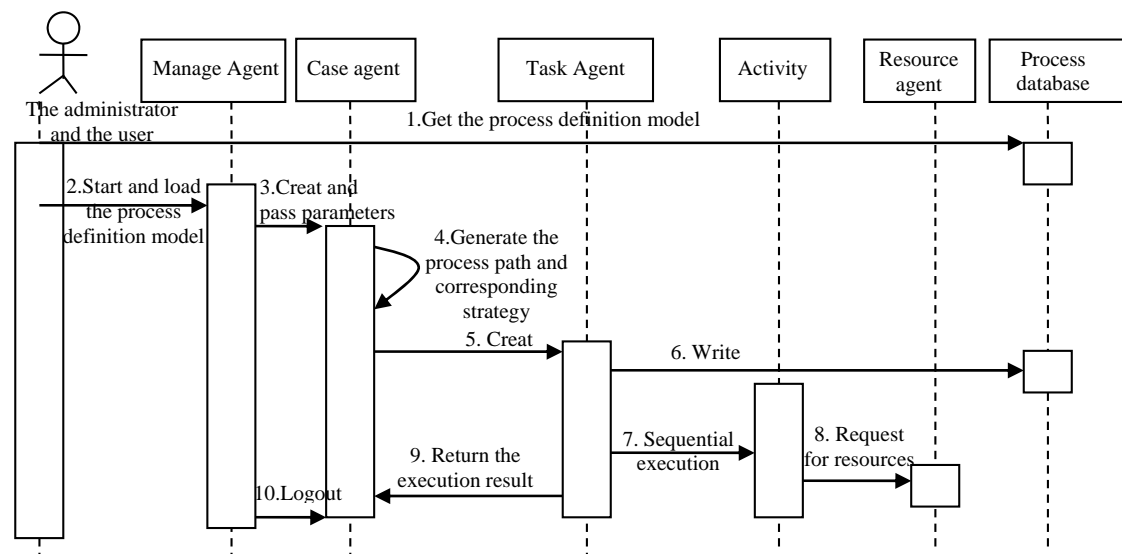


Figure 1. The process execution sequence diagram.

4. Resource Management Analysis

4.1. Resolving Resource Allocation Problem

The core problem of resource management is resource allocation problem. The task agents usually have to do their tasks by interacting with a resource. There is usually more than one resource for a task in the system. Therefore, how to select the appropriate resource agent is a problem to be solved in this system. This is mainly resolved through consultations between agents. The negotiation of agent is that in a multi-agent environment, the agents can jointly complete a task or obtain a vote result by means of mutual expression of intent or discussion.

The negotiation mode between agents can be divided into dynamic, static, centralized and distributed, and the dynamic centralized structure is adopted in this system. The resource management agent is responsible for the unified management of each resource agent, accepts the resource request from the task agent, sends the negotiation request to the resource agents that satisfies the request, and determines the most suitable resource agent according to the allocation rule. The information about the resource has been specified in the resource allocation request message sent to the resource management agent by the task agent. There are two types of resource requests:

(1) After receiving this request, the resource management agent will directly find the corresponding resource agent and feed its related information back to the task agent. The task agent negotiates the tasks with the resource agent.

(2) Specify a resource set, organizational unit, or role: Typically, a resource set, organizational unit, or role is specified when the process model definition is defined. Specific resources are dynamically bound at run time. In this case, the allocation of the resource agent is determined by bidding. Each resource agent sends a price tag to the resource management agent according to its own situation, and the resource management agent chooses one of the lowest prices.

4.2. Resource Management Module

The portion of the system related to the resource management module is shown in Fig. 2. From Fig. 2, we can see that resource management is realized through negotiation of the task agent, resource management agent and resource agent.

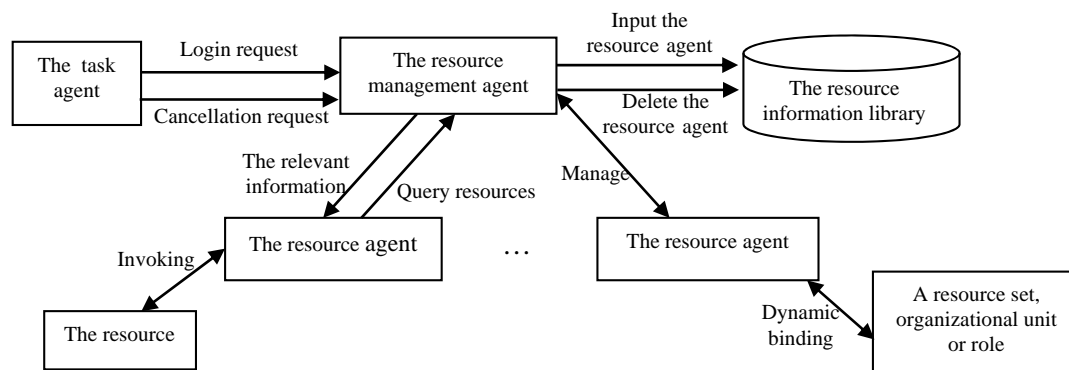


Figure 2. Resource management module.

The implementation strategy of specific resource management is as follows:

When a new resource agent requests registration, a registration request is made to the resource management agent. The resource management agent makes a judgment on the request, primarily to see whether definitions such as roles meet the requirements. The judgment is implemented by well-defined constraint-judgment functions. For example, when assigning a role, and the constraint function associated with the role is invoked to see if it is authenticated. If registration passes, the resource agent is added to the resource information library, and the information of the resource agent and the corresponding resource information are kept in the library. When a resource agent sends a cancellation request, the relevant information in the library is deleted. When the task agent queries the resource for the resource management agent, it finds the corresponding resources according to the resource allocation strategy, and then sends the address of the resource agent corresponding to the resource and the relevant information to the task agent.

5. Conclusion

The combination of multi-agent technology and distributed workflow management system can solve the problems existing in the traditional workflow system. This paper analyzes the agents with different functions in distributed workflow management system. And the important function process execution of workflow management system is studied deeply. In the design section, the system structure of the resource management module in the workflow management system is given and the implementation strategy of resource management is discussed. In the workflow management system there are still some key technologies to be studied, such as agent migration strategy, communication security, agent generation control and so on. These contents will be further improved in the following research work

Acknowledgments

This work was financially supported by Wuhan Education Bureau fund, Project Number: 2007k68

References

- [1] Li Yang, Wenbin Chen, Jian Zhang, Application research on distributed database access of Down-Hole Operation producing system based on Mobile-Agent technology, IEEE, 2010.
- [2] Nick Russell, Wil M. P. Van der Aalst, Arthur H. M. ter Hofstede Workflow Patterns: The Definitive Guide, MIT Press, 2016.
- [3] M Huang, Y Tang, J Huan, B Sun, Research on Distributed Information Search based on Mobile Agent, Information Engineering Letters, 2013.
- [4] Xin Fu, Research on improved distributed information access system based on agent module, IEEE, Changchun, 2011: 3263-3266.