J. Cent. South Univ. (2015) 22: 2629−2637
DOI: 10.1007/s11771-015-2793-2

Springer

Collaboration strategy for software dynamic evolution of multi-agent system

LI Qing-shan(李青山), CHU Hua(褚华), ZHANG Man(张曼), LI Min(李敏), DIAO Liang(刁亮)

Software Engineering Institute, Xidian University, Xi'an 710071, China

Abstract: As the ability of a single agent is limited while information and resources in multi-agent systems are distributed, cooperation is necessary for agents to accomplish a complex task. In the open and changeable environment on the Internet, it is of great significance to research a system flexible and capable in dynamic evolution that can find a collaboration method for agents which can be used in dynamic evolution process. With such a method, agents accomplish tasks for an overall target and at the same time, the collaborative relationship of agents can be adjusted with the change of environment. A method of task decomposition and collaboration of agents by improved contract net protocol is introduced. Finally, analysis on the result of the experiments is performed to verify the improved contract net protocol can greatly increase the efficiency of communication and collaboration in multi-agent system.

Key words: multi-agent system; dynamic evolution; task decomposition; collaboration

1 Introduction

With the rapid development of the computer hardware and software and the increasing enormous scale of system, software functions are getting more and more complex. But software functions are not the same in different environments and situations. In order to solve this problem, software integration gradually becomes one of the main methods adopted in developing the massive and complex systems. Agent has some characteristics such as initiative, autonomy, sociality and intelligence [1]. So, it is feasible to apply the agent technology to achieve a flexible integration process of the heterogeneous systems. Thus, many problems in the flexible dynamic integration process for the heterogeneous systems used in some fields can be solved.

As the Internet is developing rapidly and becoming popular, the static and closed operating environment of software systems gradually becomes open, dynamic and changeable. In order to adapt themselves to such a trend, software systems present the characteristics of autonomy, reactive, evolutional, collaborative and polymorphic and their software architectures are dynamic [2]. In such a system, the collaborative relationships of agents are changing corresponding to the changes of the actual demands. These changes will occur in different running stages of the system. Thus, the system performs well in adaptation and evolution.

The main contribution of this work is introducing the agent technology to the evolution process of integrated system. Based on the independent problem solving process by a single agent, we focus on the collaboration strategy of agents. With such a strategy, the software obtains evolution ability in dynamic and changeable environment. This paper is organized as follow. The second part describes the structure of multi-agent system which supports the dynamic evolution. This part will also introduce the hierarchical relationship between two types of agents in the system. The third part describes not only the methods of task decomposition and allocation but also the dynamic collaboration strategy of multiple agents [3]. The fourth part will verify the correctness and efficiency of the evolution process by an application example. The fifth part introduces the current related work. Finally, the sixth part is a conclusion of the work and discussion of the further research directions.

2 Organization of multi-agent system based on federal structure

The organization of multi-agent system (MAS) determines the behavior characteristics of the system, the way of interaction among agents and the structure of

Foundation item: Projects(61173026, 61373045, 61202039) supported by the National Natural Science Foundation of China; Projects(K5051223008, BDY221411) supported by the Fundamental Research Funds for the Central Universities of China; Project(2012AA02A603) supported by the High-Tech Research and Development Program of China
Received date: 2014−06−03; **Accepted date:** 2014−11−01
Corresponding author: LI Qing-shan, PhD, Professor; Tel: +86−13991916115; E-mail: qshli@mail.xidian.edu.cn

problem-solving process [4]. It has great impact on problem solving efficiency and operating performance of the system. The main structures include hierarchical structure, federal structure and completely autonomous structure. A multi-agent system is presented in this work considering the characteristics of system evolution. Its organization is shown in Fig. 1.

The hierarchical structure has three levels: the upper level agent is the control agent, the middle level agent is the service agent, and the lower level agent is the function agent. The control agent of evolution process records the capacity table of all service agents in the MAS. It can analyze the received user tasks and distribute the subtasks to the appropriate service agents. Once the evolution requirement changes, new tasks will be loaded to the control agent. The control agent will decompose the new tasks and then distribute the new subtasks to the appropriate service agents. The service agents who receive subtasks will adjust their collaborative relationships with other agents dynamically. Their collaboration is the main part of the autonomous structure. Thus, the process of dynamic integration and evolution is accomplished.

The federal structure is adopted in organizing the agents in the agents layer. These agents consist of function agents and service agents. Service agent combines with several function agents to construct a federal structure. The service agent has the capacity table of the function agents within the federal structure [5]. It provides a complex and compound capacity by analyzing the received subtasks and forming the collaboration logic of several function agents. The function agents provide atomic capability which cannot be decomposed. Different function agents can collaborate with each other directly in a service agent.

3 Task decomposition and collaboration strategy

3.1 Decomposition and allocation of collaborative tasks

3.1.1 Process of task decomposition

The typical task decomposition process is hierarchical. That means that the task decomposition process will not finish until the subtasks can be performed appropriately by a single agent. For this reason, several problems are needed to be solved, such as determining the tasks' decomposition granularity and finding a decomposition method to optimize the overall performance of multi-agent system. It will affect the optimization of task allocation and the execution efficiency whether the task is decomposed properly or not.

In order to solve the above problems, a task decomposition method based on multi-agent collaboration and improved AND/OR tree is presented in this work. With such a method, this work analyzes and studies the problem solving process. The task decomposition process adopts a hierarchical structure—tree structure. The root of the tree is the general task (final goal), and the nodes in the same layer have three kinds of relationships: "∩", "+" and "∪". The task can be expressed as $T=\{t_1, \cdots, t_i\}$, where t_i is the atomic task. There may be several possible cases for task decomposition.

1) $t_i \cap t_j$ means that t_i and t_j must be both accomplished for completing their father task, but they can be performed concurrently.

2) $t_i + t_j$ means that t_i and t_j must be both accomplished for completing the father task, but they are performed serially where t_i is executed first, then t_j is executed.

3) $t_i \cup t_j$ means that accomplish of either t_i or t_j is sufficient for completing the father task.

The process of task decomposition and the node pruning is shown in Fig. 2. The task decomposition is performed layer by layer and it will not finish until all of the leaf nodes cannot be decomposed. The general task will be discomposed into subtasks which have three kinds of relations of "∩", "+" and "∪" with each other. The pruning process is performed in the following steps: Calculating the cost of branches with a "∪" relationship is performed firstly; Then, the branch with the least cost will remain while the other branches and their father node will be removed. As shown in Fig. 2, suppose the cost of the branch whose root node is D is greater than

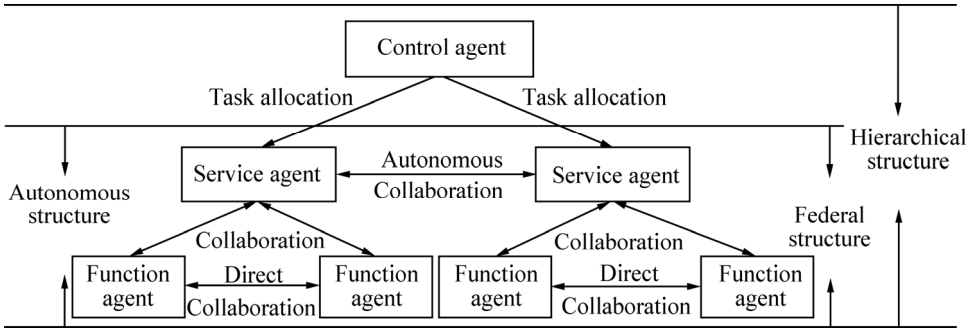

Fig. 1 Organization of multi-agent based on federal layer

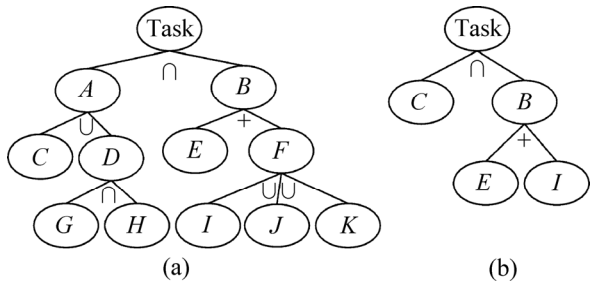

Fig. 2 Process of task decomposition (a) and node pruning (b)

the cost of branch with C as the root node, the former branch and root node A will be removed. Similarly, the node J, node K and root node F will be removed, too. Finally, the set of atomic tasks is $\{C, E, I\}$ and the execution sequence is $C\cap(E+I)$ after pruning.

3.1.2 Process of task allocation

In MAS, the efficient and suitable task allocation mechanism is helpful in not only helping each agent to try their best in executing task, but also reducing resources occupancy. What's more, if the capacity of single agent is not enough to complete the current task, several agents can cooperate to accomplish the task. To do that, they should communicate and negotiate with each other based on the existing mechanism. As shown in Fig. 3, there are mainly two types of task allocation methods: one is centralized method, and the other is distributed method.

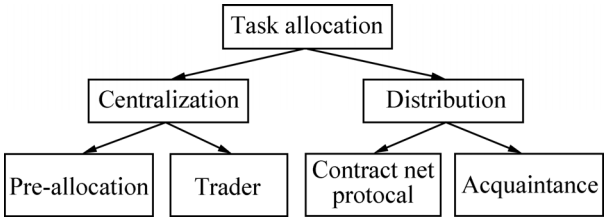

Fig. 3 Method of task allocation

There are mainly two kinds of centralized methods. One is manual pre-allocation method. In this method, the designer will distribute the tasks to agents in the system. The other is that an agent called Trader takes charge of distributing tasks. It uses a table to record all the abilities of current agents in the MAS. When a task is published, Trader will search for agents with the relevant abilities. And then, according to the current state table, it will check whether the agents agree to perform this task or not. If a consent message is received, Trader will inform the publisher of this task. Otherwise, the publisher will be told that there is no agent who can complete this task currently. In distributed method, the task decomposition and allocation process is realized by the agents to determine their action according to their interaction and perception of the environment combining with their knowledge. There are also two kinds of distributed methods: method using contract net protocol and method using acquaintances.

According to the hierarchical structure of agents described in previous section, task allocation process is performed in two major steps. In the first step, Trader of the centralized method is used. In this step, the whole tasks are decomposed into subtasks. The service agents are able to perform these subtasks by cooperation with each other. Then, the subtasks will be distributed to the appropriate service agents. The control agent of evolution process at the top of the hierarchical structure uses a table to record the capacity of all the service agents in the current MAS so that it can perform the task allocation. After receiving the requests from user, the control agent will look for service agents which are able to complete the subtasks according to the capacity table and distribute the subtasks to them. In the second step, the service agent will analyze the received subtask for some information about the subtask and the agents who will cooperate with it. At the same time, according to the information in the knowledge library, service agent decomposes the subtask into simple tasks, and then it will choose some acquaintances or find some agents using the bidding mechanism to complete the subtasks. There are only two possible types of tasks. One is atomic task named t, which can be completed by function agents. The other is collaborative task named T, which needs to be done by several function agents. Atomic task corresponds to a capability of function agent. Collaborative task corresponds to a service provided by service agent. $T=\{t_i, t_j, t_k\cdots\}$. That means that each collaborative task can be decomposed into a serial of atomic tasks by task decomposition process. Atomic tasks are accomplished by function agents. If all the subtasks are accomplished by function agents in the federal structure, the service provided by a service agent is realized

3.2 Multi-agent collaboration strategy based on improved CNP

3.2.1 Improved scheme

After the task decomposition and allocation process, it is a key problem to find a method by which agents cooperate with each other to complete the assigned task. The traditional Contract Net Protocol (CNP) always plays an important role in Multi-agent collaboration. In the traditional CNP method, the collaboration agents will be divided into manager and executive. The two types of agents cooperate with each other adopting the market mechanism. But the traditional CNP has some shortages listed as follows.

1) In process of publishing biding document, using broadcast to publish biding document results in both frequent communication and the waste of resources.

Meanwhile, the collaboration is inefficient because there is no limitation for the bidder.

2) In the case that state change of agent occurs, only the static problem solving capabilities of agent is taken into account. Some dynamic factors which will change according to the state of function agents are not considered, such as their workload, quality of solution, and the cost.

3) Some problems exist in receiving the biding document. After task decomposition which is changeless for certain kind of task, service agent will distribute subtasks to the function agents for execution. Function agents can only accept tasks passively so that no function agent can choose the subtasks according to its plan and capacity. This changeless task allocation will lead to inevitable inefficiency.

In order to solve the above problems as well as meet the demands of the dynamic evolution of the system, this work presents a collaboration strategy named "preferential acquaintances based on improved CNP (PAICNP)". This strategy is more suitable for the communication and collaboration of agents. Traditional CNP is improved in the following three aspects based on analysis and comparison.

1) The range of biding invitation and the number of biding document are restricted. The range of bidding invitation can be narrowed down by adding acquaintance library for each service agent. Thus, the system has the advantages, such as a good real time performance, and less amount of communication.

2) The capacities of bidders are constantly changing. The mental status parameters are used in the bidding process. According to the evaluation results of an accomplished task, the parameters are updated timely. So, the publisher can know the latest capability information of each agent.

3) Some bidders will be filtered out and the function agents will perceive the biding document actively. Unlimited publishing of biding document will inevitably lead to accumulative tasks and overweight burden for some agents. Thus, the intermediary service organization (ISO) is added to the system. It will monitor the state of bidder agents and perform initial filter on their bids. At the same time, the functions agents are improved so that they can perceive the published biding document according to its own state. By this method, the accomplish rate of tasks and efficiency are improved.

3.2.2 Preferential acquaintances based on improved CNP

The PAICNP improves the traditional CNP in three aspects: join of the acquaintance collaboration, mental status parameters of agents and the intermediary service organization. Details about them are described below.

1) Acquaintance collaboration

When the system is initialized, some function agents will be added into acquaintances library of a service agent. After decomposing the collaborative task into atomic tasks, service agent will search the acquaintance library firstly for function agents who are able to complete these atomic tasks. If such agents are found, the service agent will send messages directly to ask them to perform the atomic tasks.

An attribute $D_{\text{familarity}}$ is defined to indicate the familiarity between service agent and function agent. Its value is based on the different collaboration frequency between agents. $D_{\text{familarity}}(a,b)$ is the familiarity between agent a and agent b. The values of $D_{\text{familarity}}(a,b)$ are different because of the different cooperation times of each pair of agents. At the same time, if cooperation is completed, the value of familiarity and the acquaintance library may be modified. So, the acquaintance library is usually changing. The modifications are described below.

Join of acquaintance: The value of $D_{\text{familarity}}(a,b)$ is checked firstly. If it is not less than the threshold R_{m}, agent b and its ability will be added into the acquaintance library and the acquaintance capability library of agent a, respectively.

Exit of acquaintance: The value of $D_{\text{familarity}}(a,b)$ is checked. If it is less than the threshold R_l or agent b cannot accept a task in a fixed time T_l, agent b and the ability of agent b will be removed from the acquaintance library and acquaintance capability library of agent a separately.

Initialization of familiarity of acquaintances: During the initialization process of the system, some function agents will be added into acquaintance library of a service agent according to the rules given by Trader. The initial value of familiarity $D_{\text{familarity}}$ of these function agents and the service agent is R_{m} and the exit threshold is R_l. At the beginning, R_{m} is greater than R_l so that the function agents will not be removed from the acquaintance library if they didn't participate in the first collaboration process. The value of familiarity $D_{\text{familarity}}$ will be zero between service agent and the function agents which is not in its acquaintances library.

Change of $D_{\text{familarity}}$: After a successful collaboration, the service agent will check its acquaintance library. The values of $D_{\text{familarity}}$ of all the acquaintances will be modified. If function agent b participated in the collaboration process with service agent a, $D_{\text{familarity}}(a,b)$ will be modified according to Eq. (1). If function agent c didn't participate in the collaboration process with service agent a, $D_{\text{familarity}}(a,c)$ will be modified according to Eq. (2).

$$D_{\text{familarity}}(a,b)=\min[D_{\text{familarity}}(a,b)+\Delta r, R_{\text{m}}] \tag{1}$$

$$D_{\text{familarity}}(a,c)=\max[0, D_{\text{familarity}}(a,c)-\Delta r] \tag{2}$$

where Δr is the variation of familiarity.

2) Introduction of mental status parameters

Because the operating environment of function agents is dynamic and changeable, some mental status parameters are used in the bidding process of CNP, such as confidence degree, friendliness degree and active degree. The system will evaluate each agent and update their capacities regularly; and then it will modify the collaborative relationships according to the mental status parameters to improve the collaboration quality.

Definition 1: $D_{\text{perception}}(a,b,t)$ is the degree of perception which represents the capacity of agent b to complete the task t published by agent a. Usually, it depends on the computing resources owned by agent b, the CPU utilization and available memory of the host computer and workload of function agent b. If all resources of the current function agent are available and no task is on-going in the system, the degree of perception is 1.

$$D_{\text{perception}}(a,b,t) = C_1 \times R_{\text{b}} + C_2 \times \frac{T_{\text{used}}}{T_{\text{total}}} \tag{3}$$

In Eq. (3), R_{b} is the available resources that agent b owns; T_{used} is the elapsed time; T_{total} is the total time needed to complete the task; $T_{\text{used}}/T_{\text{total}}$ indicates workload of function agent b and is used to show whether the agent is busy or not; C_1 and C_2 are the weights of resources factor and agent workload, respectively. The weight can be dynamically updated by the urgency of required resources for different tasks.

Definition 2: $D_{\text{confidence}}(a,b,t)$ represents the ability of agent b to accomplish task t. It is evaluated by agent a in MAS. Its value is primarily based on the result of completing task t by agent b. If agent b doesn't have the ability to complete task t, the $D_{\text{confidence}}(a,b,t)$ will be initialized to 0; otherwise, it will be 0.5. Degree of confidence seems similar to familiarity degree of acquaintance, but there are certain differences between them. Degree of confidence focuses on the effect brings by direct cooperation of two agents to complete a specific task. But familiarity degree of acquaintance focuses on whether the two agents used to cooperate to complete any task or not while the content of the collaboration task is not concerned.

For a successful collaboration, the degree of confidence between service agent a and function agent b which are related to this collaboration is modified according to Eq. (4). If the collaboration fails, it will be modified according to Eq. (5).

$$D_{\text{confidence}}(a,b,t) = \min[D_{\text{confidence}}(a,b,t) + \Delta p, 1] \tag{4}$$

$$D_{\text{confidence}}(a,b,t) = \max[0, D_{\text{confidence}}(a,b,t) - \Delta a] \tag{5}$$

where Δp is the increment and Δa is the decrement.

Definition 3: $D_{\text{friendliness}}(a,b,t)$ is the ratio of the successful times of task t completed by agent b to the total times of the task t assigned by agent a to agent b in the MAS.

$$D_{\text{friendliness}}(a,b,t) = \frac{N_{ab}^t}{N_a^t} \tag{6}$$

In Eq. (6), N_{ab}^t is the successful times of task t given by agent a and completed by agent b; N_a^t is the total times of task t given by agent a to agent b. The degree of friendliness is one of the important indicators in CNP. Agent b will receive many bids at a moment. But it prefers to bids published by an agent who has a greater $D_{\text{friendliness}}$ with it.

Definition 4: $D_{\text{active}}(a,b,t)$ is the active degree. It is the ratio of the bidding times of agent b for task t to the bidding times of all agents. Task t is published by agent a.

$$D_{\text{active}}(a,b,t) = \frac{N_b^t}{\sum_{k=1}^{n} N_k^t} \tag{7}$$

In Eq. (7), N_b^t is the bidding time of agent b for task t; $\sum_{k=1}^{n} N_k^t$ is the bidding time of all agents for task t.

The active degree is one of the factors taken into account when sponsors invite for bidding in the CNP. It is used to avoid that some agents bid actively but fail to be selected because of various reasons. The bidding inviter agent prefers to the bidder with higher degree of active, thus avoiding a result that the number of friend agents is limited so that the system will be increasing better in the collaboration of agents.

3) Introduction of intermediary service organization

The mainly two functions of the ISO are described as follows. Firstly, it restricts the range of bidding invitation. The ISO records all relevant information of agents in the system. Before the information of bidding is published, it can roughly estimate the number of the candidate agents which have ability to complete the tasks. Through filtering out some bidder agents, it is reduced the traffic loads in the network and the workload after receiving bids from bidder agents. Secondly, it adopts active perception for agents to solve the problem that the function agents can only accept tasks passively. The bidding inviter agent publishes the bidding document into the ISO. If the function agent is busy in its current task or its available resources are limited, it will not send message to ISO but concentrate on the current task. Otherwise, it will send a message to the ISO to get the current valid bidding document.

3.2.3 Process of collaboration

Suppose the current service agent is agent a. It is responsible to manage tasks and invite for bidding for the task t. The bidding process of the PAICNP is shown in Fig. 4. The procedures of task allocation and consultation

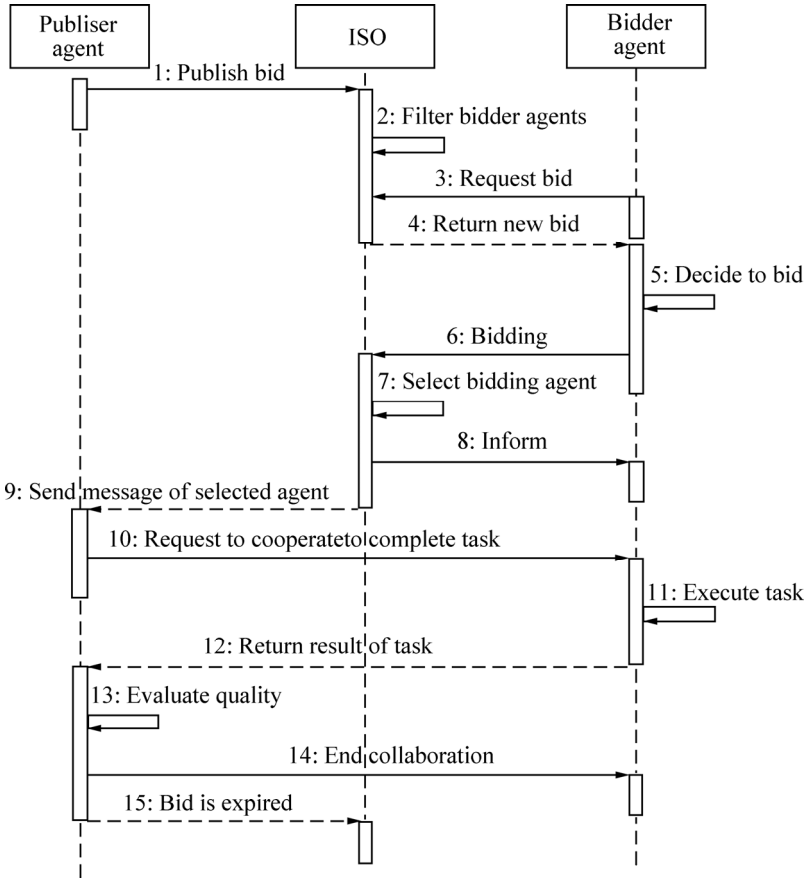

Fig. 4 Process of collaboration

are as follows.

1) Select bidders. The service agent checks whether their acquaintances in the acquaintance library can complete the task or not. If the acquaintances have ability for accomplishing the task, the service agent will send the collaboration message directly to them. Otherwise, agent a sends invitation for bidding to all function agents except its acquaintances.

2) Invite for bidding. Agent a publishes the bidding document to the ISO. The function agents perceive the bidding document regularly.

3) Bidding process. Function agent begins bidding according to its own workload and resources. The bidding strategy is given in Eq. (8).

$$D_{\text{bidding}}(a,b,t)=\alpha\times D_{\text{perception}}(a,b,t)+\beta\times D_{\text{confidence}}(a,b,t)+\gamma\times D_{\text{friendliness}}(a,b,t)+\delta\times D_{\text{active}}(a,b,t) \quad (8)$$

where $D_{\text{bidding}}(a,b,t)$ is the degree of expectation which indicates the desire of agent b to bid for task t; $D_{\text{perception}}(a,b,t)$ is the degree of perception that agent b perceives the bid; $D_{\text{confidence}}(a,b,t)$ is the degree of confidence which indicates the confidence of agent a to agent b to accomplish task t; $D_{\text{friendliness}}(a,b,t)$ is the degree of friendliness between agent a and agent b; $D_{\text{active}}(a,b,t)$ is the degree of active between agent a and agent b; α, β, γ and δ are the weights, $\alpha+\beta+\gamma+\delta=1$.

When the degree of expectation of function agent is greater than the given threshold W, the function agent will bid for the task. If function agent perceives more than one bidding documents simultaneously, it only bids for the first two whose publishers have higher degree of expectation with this function agent. If only one bidding document is perceived and its degree of expectation is greater than the given threshold, the function agent will bid for it.

1) A function agent wins the bidding. According to the large amount of bidding information provided by the bidders, the publisher agent selects the most appropriate bidder agent as successful bidder.

2) Sign the contract. After receiving the request to confirm the contract, the selected bidder agent signs the contract with publisher agent and begins to perform the task t. If the bidder agent accomplishes the task, it will return the result to the publisher agent and turns to procedure 6. If it doesn't accomplish the task, the publisher agent will decide whether it should pack the rest of the task and generate a new bidding document or not. If the rest of task is packed, it will turn to procedure 1.

3) Task t is completed. The publisher agent evaluates the result and updates the degree of familiarity,

friendliness, active and confidence of the bidder agent. Then, it informs the ISO that the bidding document is invalid.

3.3 Process of evolution

In order to adapt themselves to the open and changeful environment, software systems gradually present the characteristic of dynamic evolution. In the case that some changes of collaboration tasks occur in the system, new collaboration tasks should be loaded into the system. The operating platform often needs to be shut down and then load the tasks. After loading the new tasks, the platform will restart. However, the cost of halting devices is usually very high; especially, some special devices cannot be shut down. So, it is a major problem to be solved switching to the new tasks without stopping running. According to the hierarchy structure of agents as well as the task granularity in the evolution presented in this work, the system's evolution process is classified into two categories. One is the system-level evolution which is the evolution among service agents. It means that the system will load the new collaboration tasks on receiving them and then perform the new tasks at the next operation cycle. The other one is the evolution of the federal organization which is the evolution between service agents and function agents. In the case that the function agent who provides ability for service agents is missing or abnormal, the system will look for another function agent with the same ability through a certain collaboration mechanism to participate in the collaboration process.

The process of system-level evolution is shown in Fig. 5, the current tasks executed in the system is $A+(B\cap C)+D$. When the system receives the new tasks $(A\cap B)+C+D$, the control agent explains the new tasks immediately, divides them into some subtasks and distributes them. At this moment, the running tasks are the old tasks. The system will not switch to the new tasks until all the new tasks are distributed successfully.

The evolution between service agents and function agents means that when the function agent which provides ability for service agents is missing or abnormal, the system will look for another function agent with the same ability to participate in the collaboration process by means of inviting for bidding using PAICNP strategy. At the same time, evolution of collaboration relationship of each function module in a single service agent is performed in order that the service agent can provide the same service.

4 Case study

To verify the superiority of the PAICNP in both communication traffic and efficiency to execute the tasks, this section will give simulation experiments of the traditional CNP and the PAICNP and then analyze the results by a traffic simulation experiment with large number of subtasks.

4.1 Simulation of experimental data

Assuming that the system contains six function agents and each of them has different number of function agents in their acquaintance library. The value of each parameter is shown in Table 1. For convenience in comparing, the values of these parameters are the same for all function agents participating in the collaborations process.

Some descriptions of the experiment are as follows.

1) Range of the familiarity and confidence: $D_{\text{confidence}}(a,b)\in(0,1)$, $D_{\text{confidence}}(a,b)\in(0,1)$.

2) The interval for publishing tasks is the same in the case that the parameters are the same in the initial state.

3) The end of a task is identified by the last communication received by the publisher about this task.

4) Numbers 0 and 1 are used to present whether the task is executed successfully or not. The number 0

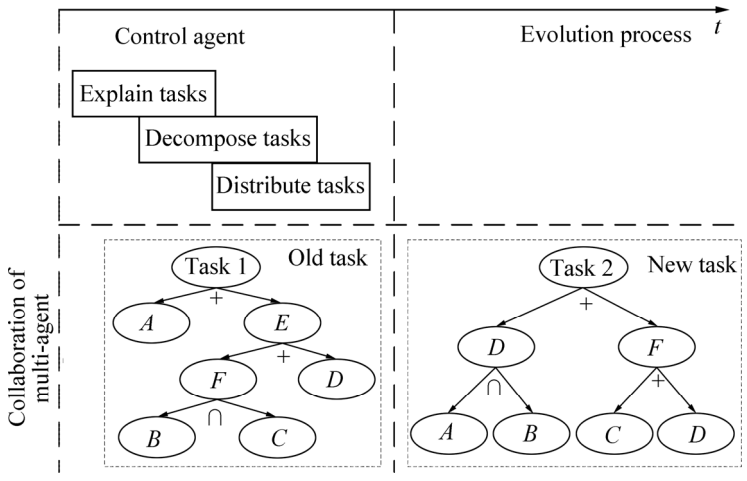

Fig. 5 Process of system-level evolution

Table 1 Simulated parameters

Parameter	Statement	Value
R_m	Threshold used in adding acquaintance	0.5
R_l	Threshold used in removing acquaintance	0.1
Δr	Variation of familiarity	0.01
$D_{confidence}$	Degree of confidence	0.5
Δp	Increment of confidence	0.001
Δa	Decrement of confidence	0.01
$D_{friendliness}$	Degree of friendliness	0.0
D_{active}	Degree of active	0.0

indicates failure while 1 indicates success. If the returned result is 1, the degree of familiarity and confidence will be updated using Eq. (1) and Eq. (4) separately. If the returned result is 0, they will be updated using Eq. (2) and Eq. (5), respectively. While the degree of friendliness and active will be adjusted using Eqs. (6) and (7), respectively.

The following two strategies are used in the experiments.

1) In experiment using the traditional CNP, no mental parameter is contained and service agent broadcasts the published bidding documents.

2) In experiment using the PAICNP, the values of familiarity, confidence, friendliness and active are set. The function agent will decide whether it can bid or not using Eq. (8).

4.2 Process of experiment and analysis of result

Figure 6 shows the communication rate of the CNP and the PAICNP. They are the communication rates of systems using the two methods in the same period of time when completing the same tasks. It can be seen from the Fig. 6 that at the initial phase of the system, lack of acquaintances in acquaintance library makes it necessary to invite for bidding to ask some agents to accomplish the task. This leads to large amount of communications. However, with the increasing cooperation times among agents, the acquaintance library will record some information of the acquaintances. In this case, the system can choose some acquaintances to perform the task so that less communication is needed and thus the communication traffic will be gradually less and stable.

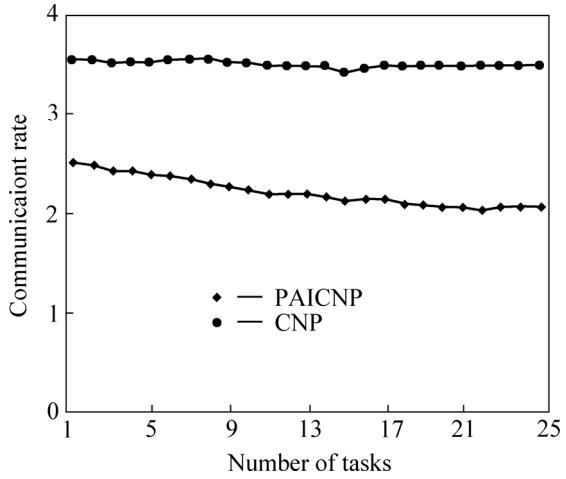

Fig. 6 Test of communication rate

Figure 7 shows the comparison between results of tests using the CNP and PAICNP. From Fig. 7, it can be summarized that when the MAS and the tasks are the same, using the traditional CNP leads to higher time consumption than that using PAICNP. At the beginning, every agent has equivalent ability. Due to the characteristic of active perceiving, the PAICNP is better than the CNP. The traditional cooperation seems to be always static and passive; while the network of acquaintance relationships is formed in the PAICNP. In other words, when two agents who used to cooperate with each other perform new collaborative tasks, all they need to do is to look up their own acquaintance library. It will avoid the process of bidding and save time. Besides, with the increased amount of the tasks, the former method requires more time than the latter one; so, the PAICNP performs better.

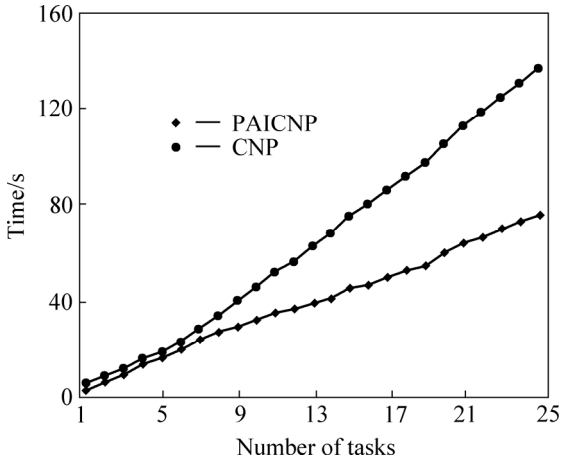

Fig. 7 Test of rate of completing tasks

The advantages of the PAICNP over the traditional CNP are shown as follows according to the data analysis.

1) Less communication. The ISO filters some agents in acquaintance library and some busy agents to reduce the number of communication and selects the agents with better mental status parameters than others to perform subtasks in order to decrease the bidding times and polling times.

2) Less collaboration time. The introduced acquaintance library avoids the bidding process in some certain cases. The service agents can directly communicate with acquaintances and the collaboration time is decreased.

3) Higher efficiency to complete a task. The agent

can decide whether it is able to perform a collaborative task or not according to its current state contained in the introduced mental parameters. Thus, they can accomplish a task better. After running for a period time, the collaborative relation will tend to be better by dynamic update of the mental status parameters.

5 Related works

There are many existing researches on agent. DAVIS and SMITH [6] and SMITH [7] proposed the contract net protocol (CNP) whose main idea was to perform negotiation by means of communication in problem solving process avoiding possible conflict. In the foundation of traditional CNP, a number of improvements and expansions have been made in order to increase the efficiency and expand the range of application, but traditional CNP limits the number of agents. Fischer introduced mechanism of interim trust/ reject and simulated trade to the CNP to optimize the task distribution [8]. We draw lessons from it to optimize distributing time; CONRY et al [9] used multi-level consultation protocol to solve the problems of task distribution and resource allocation. SHIFFER [10] introduced a cooperative planning system (CPS) based on bulletin board; LESSER and CORKILL introduced the functionally accurate, cooperative (FA/C) [11]; Concepts of hierarchical structure and marginal cost calculation proposed by SANDHOLM and LESSER [12] and bidding threshold proposed by CHEN [13] are introduced into the decision-making process of CNP, in order to reduce the calculation time of the consultation process as well as the cost of communication. LUO et al [14] firstly proposed the acquaintance model to obtain the basic information of collaborative agent; PECHOUCEK and SISLAK et al [15] proposed the tri-base acquaintance model based on the twin-base acquaintance model [16]. MICHAL and VLADIMIR proposed the CPlanT acquaintance model in 2002, which was to achieving the planning of agent's intelligent belief in multi-agent systems [17].

6 Conclusions

This work not only presents a hierarchical structure of multi-agent system similar to a federation but also defines the service agent and function agent. The improved CNP can greatly increase the efficiency of communication and collaboration between large number of subtasks by cutting down the task decomposition and allocation time. It is assumed that the agent is charitable and friendly. It remains for further study to determine what measures should be taken when the agent is malicious.

References

[1] LAUDON K C, LAUDON J P. Management information systems: Managing the digital firm [M]. New Jersey: Prentice Hall, 2003: 82−86.

[2] MAO Xin-jun. Agent-oriented software development [M]. Beijing: Tsinghua University Press, 2005: 39−46. (in Chinese)

[3] LI Qing-shan, CHEN Wei, ZHU Meng-xia. An agent-based system d ynamic integration method for multi-level evolution [J]. International Interdisciplinary Journal, 2012, 15: 315−322.

[4] FABIO B, GIOVANNI C, DOMINIE G. Developing multi-agent systems with JADE [M]. Hoboken: Wiley, 2007: 52−53.

[5] MAO Xin-jun, HU Cui-yun, SUN Yue-kun, WANG Huai-min. Research on agent-oriented programming [J]. Journal of Software, 2012, 23(11): 11−13.

[6] DAVIS R, SMITH R G. Negotiation as a metaphor for distributed problem solving [J]. Artificial Intelligence, 1983, 20(1): 63−109.

[7] SMITH R G. The contract net protocol: High-level communication and control in a distributed problem solve [J]. IEEE Transactions on Computers, 1980, 29(12): 1104−1113.

[8] FISCHER K, MULLER J P. A model for cooperative transportation s cheduling [C]// LESSER V, GASSER L. Proceeding of the First International Conference on Multiagent Systems. Cambridge, M A: MIT Press, 1995: 109−116.

[9] CONRY S E, MEYER R A, POPE R P. Impact of local decisions in distributed planning [J]. Distributed Artificial Intelligence, 2014, 2: 245.

[10] SHIFFER M J. Towards a collaborative planning system [J]. Environment and Planning B, 1993, 19: 709−709.

[11] LESSER V R, CORKILL D D Functionally accurate, cooperative distributed systems [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1981, 11(1): 81−96.

[12] SANDHOLM T W, LESSER V R. Issues in automated negotiation and electronic commerce: Extending the contract net framework [C]// LESSER V, GASER L. Proceeding of the First International Conference on Multi-agent Systems. Combridge, M A: MIT Press, 1995.

[13] CHEN Xue-guang. Further extensions of FIPA contract net protocol: Threshold plus DoA [C]// Proc ACM Symposium on Applied Computing. ACM Press, 2004: 45−51.

[14] LUO X, MIAO C, JENNINGS N R, HE M, SHEN I, ZHANG M. KEMNAD: A knowledge engineering methodology for negotiating agent development [J]. Computational Intelligence, 2012, 28(1): 51−105.

[15] SISLAK D, PECHOUCEK M, MARIK V. System and method for planning/replanning collision free flight plans in real or accelerated time: U S Patent 8538673 [P]. 2013−9−17.

[16] MAFIK V, PECHOUCEK M, STEPANKOVA O. ACQUAINTANCE MODEL IN [J]. Advances in Networked Enterprises: Virtual Organizations, Balanced Automation, and Systems Integration, 2013, 53: 175.

[17] BOISSERIER L, BORDINI R H, HUBNER J F, RICCID A, SANTI A. Multi-agent oriented programming with JaCaMo [J]. Science of Computer Programming, 2012, 4(2): 13−15.

(Edited by YANG Hua)