Computing and Informatics, Vol. 29, 2010, 389–405

ORGANIZATIONAL STRUCTURE-SATISFACTORY SOCIAL LAW DETERMINATION IN MULTIAGENT WORKFLOW SYSTEMS

Yichuan Jiang

School of Computer Science and Engineering
Southeast University
Nanjing 211189, China &
The State Key Laboratory for Manufacturing Systems Engineering
Xi'an Jiaotong University
Xi'an 710054, China
e-mail: yjiang@seu.edu.cn, jiangyichuan@yahoo.com.cn

Manuscript received 23 June 2008; revised 9 February 2009
Communicated by Patrick Brézillon

Abstract. The multiagent workflow systems can be formalized from an organizational structure viewpoint, which includes three parts: the interaction structure among agents, the temporal flow of activities, and the critical resource sharing relations among activities. While agents execute activities, they should decide their strategies to satisfy the constraints brought by the organizational structure of multiagent workflow system. To avoid collisions in the multiagent workflow system, this paper presents a method to determine social laws in the system to restrict the strategies of agents and activities; the determined social laws can satisfy the characteristics of organization structures so as to minimize the conflicts among agents and activities. Moreover, we also deal with the social law adjustment mechanism for the alternations of interaction relations, temporal flows, and critical resource sharing relations. It is proved that our model can produce useful social laws for organizational structure of multiagent workflow systems, i.e., the conflicts brought by the constraints of organization structure can be minimized.

Keywords: Multiagents, workflows, coordination, social laws, social strategies, organizational structures

Mathematics Subject Classification 2000: 68-XX: Computer Science

1 INTRODUCTION

Workflow and business process management have been achieving much attention in related research fields [1]. A workflow is a series of actions performed by actors, which is widely used to manage business processes for different users [2]. Multiagent techniques can achieve the merits of autonomy and automation, which are always used to achieve the dynamic enactment of workflows [2, 3, 4]. In multiagent workflow systems, each agent acts on behalf of an actual actor in the business process, and represents an actual individual in a business organization [5]; thus the agents may be located within certain organizational structures of the systems.

In the multiagent workflow systems, the agents may take actions autonomously on behalf of many participants that have different goals; to achieve the desired goal of whole system, we should make coordination among multiagents and activities. Coordination is the key for developing realistic multiagent systems [6, 7, 8, 9]. There are always two methods on agent coordination in previous works; one is centralized model, and the other is decentralized model. In the centralized model, a single controller is adopted to control the behaviors of all agents [10]. Obviously, such model is simple, but it also has many drawbacks, such as single point failure, controller may become performance neck, etc. [11]. Therefore, the centralized model can not satisfy the requirements for the mobility and dynamics of current multiagent workflow systems. By contrast, in the decentralized model, there is not a controller in the system. When the conflicts take place, the agents will execute appropriate negotiation mechanism to resolve those conflicts [12]. However, the fully decentralized model is inefficient and also leads to conflict [11]. Moreover, the system sometimes may not achieve the desired goal.

Social law is a new method to control the mulitagent systems [13, 14, 15, 16], whose basic idea is to add some social laws into the system to realize coordination among agents. The social law provides a spectrum between a totally centralized approach and a totally decentralized approach to coordinate multiagents; the social law allows the agents enough freedom on the one hand, but at the same time constrains them so that they will not interfere with each other [13, 14, 15, 16, 17, 18, 19]. Therefore, in the multiagent workflow systems, we can use social laws to constrain the actions of agents in the business process according to the environments, thus the desired goal and harmony of the system can be achieved.

Typically, a multiagent workflow system is an organization of coordinated agents that interact in order to implement the business actions to achieve desired goals [1, 2, 3, 4, 5]. In a multiagent workflow system, agents are constrained by their organizational relations, such as interaction structure, activity sequences and resource sharing. Organizational structure is an abstraction used to describe the overall architecture of multiagent workflow system and to define the constraining relations among agents and business actions. While an agent takes actions in the system, it should select the strategy which can match its organizational position in the system. If an agent's behavior strategy can not match its organizational position, it may collide with other agents' actions in the business processes. Therefore, as the set of restrictions on agent actions, the social law should be determined to satisfy the organizational structures in multiagent workflow systems.

In this paper, we mainly consider three kinds of constraints of organizational structures in multiagent workflow systems: 1) constraints brought by the interaction structures among agents; 2) constraints brought by the temporal relations among actions; 3) constraints brought by the sharing of critical resources among activities. Therefore, we will investigate how to endow social laws into the systems to satisfy those constraints of organization structures, thus minimizing the conflicts among agents and actions.

The rest of this paper is organized as follows. In Section 2, we formalize the organizational structures of multiagent workflows; in Section 3, we present the model of social saw determination for organization structures; in Section 4, we address the adjustment of social laws for structure alternations; finally, we conclude our paper and discuss our future work in Section 5.

2 ORGANIZATIONAL STRUCTURES OF MULTIAGENT WORKFLOWS

In a multiagent workflow, there are always some agents which can take actions for business process; and the executions of actions by agents need some resources. In the workflow system, some resources are critical since they can be accessed by only one agent at one time. Figure 1 is an example for the organization structure of a multiagent workflow system, which includes three parts: the interaction structure among agents a), the temporal flow of activities b), and the critical resource sharing relations among activities c).

Therefore, the constraints brought by the organizational structure of multiagent workflow system include three types: 1) constraints brought by the interaction structures among agents; 2) constraints brought by the temporal relations among actions; 3) constraints brought by the sharing of critical resources among activities.

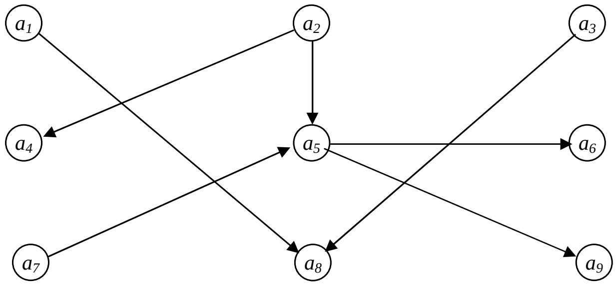
Figure 1 a) shows the interaction structure among agents, which defines the relational constraints among agents. For example, there is a relational constraint from a_1 to a_8, thus the strategy of a_8 will be constrained by a_1.

Figure 1 b) shows the temporal flow of activities, which defines the temporal constraints among activities. For example, action t_8 should be executed after t_5 and t_7, and action t_7 should be executed after t_5; thus we can endow a social law to ensure the execution flow of the three actions as $t_5 \to t_7 \to t_8$.

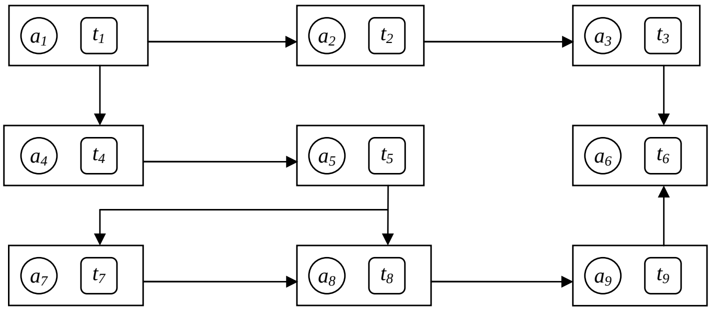
Figure 1 c) is the critical resource sharing relations among activities. For example, t_5 and t_3 will both access the critical resource r_3. Therefore, we should endow a social law to ensure the executions of these two actions are not simultaneous.

2.1 The Interaction Structure Among Agents

The interaction structure among agents can be shaped as the form of network in which the vertices denote the agents and the edges denote their interaction relations.



a)



b)

Agent	Activity	Critical resources	Agent	Activity	Critical resources
a_1	t_1	r_1, r_2	a_5	t_5	r_3
a_2	t_2	r_1, r_4	a_6	t_6	r_1, r_4
a_3	t_3	r_3	a_7	t_7	r_1, r_4, r_5
a_4	t_4	r_3, r_4	a_8	t_8	r_2

c)

Fig. 1. An example for multiagent workflow system; a) The interaction structure among agents, b) The temporal flow of activities, c) The critical resources accessed by agents in the execution of activities

Let the agent interaction network be $N = \langle A, R \rangle$, where A denotes the set of agents and R denotes the set of agent interaction relations. If agent a is the source of a directed relation $r \in R$, then we denote it as $a \odot r$, If agent a is the destination of $r \in R$,then we denote it as $a \otimes r$.

Definition 1. The *dependency structure* of an agent in multiagent workflow is defined as the set of "*in*" interaction relations of various types linking this agent with other agents. The 1$^{\text{st}}$-order dependency structure and dependency agents of agent a_i

is the union of its immediate "*in*" links:

$$Dep_{a_i} = \{\langle a_j, a_i\rangle |\ a_j \in A\ \wedge \langle a_j, a_i\rangle \in R\} \quad (1)$$
$$\mho_{a_i} = \{a_j \mid a_j \in A\ \wedge \langle a_j, a_i\rangle \in Dep_{a_i}\} = \{a_j \mid a_j \odot r \wedge r \in Dep_{a_i}\}. \quad (2)$$

Moreover, we can define the n^{th}-order dependency structure of agent a_i as:

$$\begin{aligned}\prod_n Dep_{a_i} &= \overbrace{Dep(Dep(\ldots(Dep_{a_i})\ldots))}^{n} \\ &= \{\langle a_n, a_{n-1}\rangle | a_1, a_2, \ldots, a_n \in A \wedge \langle a_n, a_{n-1}\rangle \in R \wedge \ldots \\ &\quad \wedge \langle a_2, a_1\rangle \in R \wedge \langle a_1, a_i\rangle \in R\}.\end{aligned} \quad (3)$$

The set of all agents within the all-orders dependency structures of agent a_i is:

$$\sum \mho_{a_i} = \bigcup_k \{a_j | a_j \odot r \wedge r \in \prod_k Dep_{a_i}\}. \quad (4)$$

On the other hand, an agent may also influence other agents' strategies in the interactions structure, so we have the following definition:

Definition 2. The *domination structure* of an agent in the interaction network is defined as the set of "*out*" interaction relations of various types linking this agent with other agents. The 1^{st}*-order domination structure* and *domination agents* of agent a_i are the union of its immediate "*out*" links:

$$Dom_{a_i} = \{\langle a_i, a_j\rangle |\ a_j \in A\ \wedge \langle a_i, a_j\rangle \in R\} \quad (5)$$
$$\Omega_{a_i} = \{a_j \mid a_j \in A\ \wedge \langle a_i, a_j\rangle \in Dom_{a_i}\} = \{a_j \mid a_j \otimes r \wedge r \in Dom_{a_i}\}. \quad (6)$$

Therefore, we can define the n^{th}-order domination structure of a_i as:

$$\begin{aligned}\prod_n Dom_{a_i} &= \overbrace{Dom(Dom(\ldots(Dom_{a_i})\ldots))}^{n} \\ &= \{\langle a_{n-1}, a_n\rangle | a_1 \in A \wedge a_2 \in A \wedge \ldots \wedge a_n \in A \wedge \langle a_i, a_1\rangle \in R \\ &\quad \wedge \langle a_1, a_2\rangle \in R \wedge \ldots \wedge \langle a_{n-1}, a_n\rangle \in R\}.\end{aligned} \quad (7)$$

Therefore, the set of all agents within the *all-orders* domination structures of agent a_i is:

$$\sum \Omega_{a_i} = \bigcup_k \{a_j | a_j \otimes r \wedge r \in \prod_k Dom_{a_i}\}. \quad (8)$$

Example 1. In Figure 1 a): 1) The 1^{st}-dependency structure of agent a_6 is $Dep_{a_6} = \{\langle a_5, a_6\rangle\}, \mho_{a_6} = \{a_5\}$; the 2^{nd}-dependency structure of agent a_6 is $\prod_2 Dep_{a_6} = \{\langle a_2, a_5\rangle, \langle a_7, a_5\rangle\}$; the set of all dependency agents of a_6 is $\sum \mho_{a_6} = \{a_5, a_2, a_7\}$. 2) The 1^{st}-domination structure of agent a_2 is $Dom_{a_2} = \{\langle a_2, a_5\rangle\}$, $\Omega_{a_2} = \{a_5\}$; the

2^{nd}-domination structure of agent a_2 is $\prod_2 Dom_{a_2} = \{\langle a_5, a_6\rangle, \langle a_5, a_9\rangle\}$; the set of all domination agents of a_2 is $\sum \Omega_{a_2} = \{a_5, a_6, a_9\rangle\}$.

2.2 The Temporal Flow of Activities

The temporal flow of activities can also be shaped as the form of network in which the vertices denote the activities and the edges denote their execution sequences. Let the temporal flow network be $N = \langle T, R\rangle$, where T denotes the set of activities and R denotes the set of temporal relations. If activity t is the source of a directed relation $r \in R$, then we denote it as $t \odot r$; if activity t is the destination of $r \in R$, then we denote it as $t \otimes r$. If there is a relation from t_i to t_j, t_j should be executed after the finish of t_i.

Being similar to the definitions in Section 2.1, now we give the definitions for the former and latter structures in the temporal flow of activities.

Definition 3. *The former structure in the temporal flow of activities.* The 1^{st}-order former structure and former activities of activity t_i is the union of its immediate "*in*" links:

$$For_{t_i} = \{\langle t_j, t_i\rangle | t_j \in T \wedge \langle t_j, t_i\rangle \in R\} \tag{9}$$

$$\Delta_{t_i} = \{t_j | t_j \in T \wedge \langle t_j, t_i\rangle \in For_{t_i}\} = \{t_j | t_j \odot r \wedge r \in For_{t_i}\}. \tag{10}$$

Therefore, we can define the n^{th}-order former structure and all former activities of t_i as:

$$\begin{aligned}\prod_n For_{t_i} &= \overbrace{For(For(\ldots(For_{t_i})\ldots))}^{n} \\ &= \{\langle t_n, t_{n-1}\rangle | t_1, t_2, \ldots, t_n \in T \wedge \langle t_n, t_{n-1}\rangle \in T \wedge \ldots \\ &\quad \wedge \langle t_2, t_1\rangle \in R \wedge \langle t_1, t_i\rangle \in R\}\end{aligned} \tag{11}$$

$$\sum \Delta_{t_i} = \bigcup_k \{t_j \mid t_j \odot r \wedge r \in \prod_k For_{t_i}\} \tag{12}$$

Definition 4. *The latter structure in the temporal flow of activities.* The 1^{st}-order latter structure and latter activities of activity t_i is the union of its immediate *out* links:

$$Lat_{t_i} = \{\langle t_i, t_j\rangle | t_j \in T \wedge \langle t_i, t_j\rangle \in R\} \tag{13}$$

$$\bigtriangledown_{t_i} = \{t_j | t_j \in T \wedge \langle t_i, t_j\rangle \in Lat_{t_i}\} = \{t_j | t_j \otimes r \wedge r \in Lat_{t_i}\}. \tag{14}$$

Therefore, we can define the n^{th}-order latter structure and all latter activities of an activity as:

$$\begin{aligned}\prod_n Lat_{t_i} &= \overbrace{Lat(Lat(\ldots(Lat_{t_i})\ldots))}^{n} \\ &= \{\langle t_{n-1}, t_n\rangle | t_1 \in T \wedge t_2 \in T \wedge \ldots \wedge t_n \in T \wedge \langle t_i, t_1\rangle \in R \\ &\quad \wedge \langle t_1, t_2\rangle \in R \wedge \ldots \wedge \in \langle t_{n-1}, t_n\rangle \in R\} \qquad (15)\end{aligned}$$

$$\sum \nabla_{t_i} = \bigcup_k \{t_j | t_j \otimes r \wedge r \in \prod_k Lat_{t_i}\}. \qquad (16)$$

Example 2. In Figure 1 b): 1) The 1^{st}-former structure of activity t_5 is $For_{t_5} = \{\langle t_4, t_5\rangle\}, \Delta_{t5} = \{t_4\}$; the 2^{nd}-former structure of activity t_5 is $\prod_2 For_{t_5} = \{\langle t_1, t_4\rangle\}$; the set of all former activities of t_5 is $\sum \Delta_{a_6} = \{t_4, t_1\}$; therefore, t_5 should be executed after t_4 and t_1. 2) The 1^{st}-latter structure of activity t_5 is $Lat_{t_5} = \{\langle t_5, t_7\rangle, \langle t_5, t_8\rangle\}$, $\nabla_{t_5} = \{t_7, t_8\}$ the 2^{nd}-latter structure of activity t_5 is $\prod_2 Lat_{t_5} = \{\langle t_7, t_8\rangle\}$; the set of all latter activities of t_5 is $\sum \nabla_{t_5} = \{t_7, t_8\}$; therefore, t_5 should be executed before t_7 and t_8.

2.3 The Critical Resource Sharing Relations Among Activities

In the workflow system, some agents may only access their self-owned resources, which can not bring out conflicts. However, in the system there are some resources which are critical; a critical resource can only be accessed by one agent at the same time. For accessing the critical resources without conflicts, the activities should be implemented to satisfy the constraints brought by the critical resource sharing relations.

If an activity, t_i, is in the former (or latter) activities of t_j, t_j should be executed after (or before) t_i; thus t_i and t_j can not be executed simultaneously, so there are no resource conflicts between them. Therefore, we can give the definition of resource-constrained activities for one activity as follows.

Definition 5. *Resource-constrained activities for one activity.* Let there be an activity t_i, the set of activities that may bring out resource constraints to t_i are those that are not in the all-orders former and latter structures of t_i but share the same critical resource r_m, which can be denoted as $\Re_i^m$.

Example 3. In Figure 1 c): 1) For t_1 and r_1, though t_2, t_6 and t_7 will access the critical resource r_1, they are all in the latter structures of t_1, so they can not bring out resource constraints to t_1; thus $\Re_1^1 = \{\}$. 2) For t_3 and r_3, now t_4 and t_5 will access the critical resource r_3; since t_4 and t_5 are both not in the former or latter structures of t_3, thus $\Re_3^3 = \{t_4, t_5\}$; therefore, we should endow social laws to restrict the strategies of those three activities to realize mutual exclusive accessing of r_3.

3 SOCIAL LAW DETERMINATION FOR ORGANIZATIONAL STRUCTURES

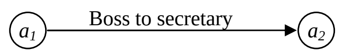
3.1 Strategy and Social Laws

Definition 6. *Strategy in the multiagent workflow systems.* The strategy that an agent takes to execute action includes three parts: 1) the strategy that an agent should take to satisfy the constraints brought by the interaction structures on it; 2) the strategy that an agent should take to execute an activity to satisfy the temporal constraints with other agents; 3) the strategy that an agent should take to execute an activity to satisfy the constraints brought by sharing of critical resources among activities.

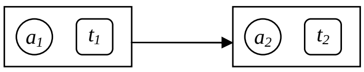
Example 4. Figure 2 a) is an example which denotes two agents for booking hotels. Agent a_1 books hotel on behalf a boss, and a_2 books hotel on behalf of the secretary. Now, to take care of the boss, the secretary should book a room that is near to the boss's. Therefore, the strategy of a_2 for booking hotel is determined by a_1's; and a_2's strategy (now it is the room) should be near to a_1's.

Figure 2 b) is an example which denotes the temporal sequence between two activities. Therefore, the temporal strategy of a_2 to execute t_2 has to obey "t_2 should be executed after t_1 is finished".

Figure 2 c) is an example which denotes the critical resource sharing among activities. Therefore, the strategy of a_1 to execute t_1 has to obey "t_1 should not be executed at the same time as t_2".



a)



b)

Agent	Activity	Critical resources
a_1	t_1	r_1, r_2
a_2	t_2	r_1

c)

Fig. 2. An example of the strategies for the three kinds of constraints

As said above, we should set some restrictions to the agents' strategies to execute activities, thus the collisions can be avoided. Therefore, we can endow social laws into the system.

Definition 7. Given an environment $\langle A, S, P \rangle$, where $A = \{a_1, a_2, \ldots, a_n\}$ is a set of agents, $S = \{S_i^j\}$, S_i^j is the set of strategies available to agent a_i to execute activity t_j, and P is the organizational structure of the workflow system. A social law is a set of restrictions $SL = \langle \tilde{S_i^j} \rangle$ such that $\tilde{S_i^j} \subseteq S_i^j$ restricts the set of strategies for each agent to execute activity to avoid collisions.

Hence, a social law in this case should make each agent adopt right strategies to keep its position in the organizational structure, thus the collisions can be minimized.

Definition 8. Given an environment $\langle A, S, P \rangle$, a *useful law* is a social law that guarantees that the strategies of agents to execute activities can satisfy the requirements of organizational structure of the multiagent workflow system, i.e., the collision among agents and activities can be minimized.

3.2 Strategy Coordination among Resource-Constrained Activities

Let the set of critical resources R, r_i denote a critical resource. First, we can set an array for the critical resources, $N_R[i]$ denotes the number of free r_i which is now available to the activities. For example, if $N_R[2] = 2$, it denotes that there are two r_2 which are now free and can be accessed. We can use $N_R[i]$ to realize the mutual exclusion on the accessing of critical resources.

Therefore, while agents execute activities, they should make strategy coordination on the basis of their mutual exclusive accessing critical resources. Let the set of activities be T, the number of categories of critical resources be $|R|$, and the set of critical resources that can be accessed by activity t_i be R_{t_i}; now we can design an algorithm to make strategy coordination among resource-constrained activities, shown as Algorithm 1.

Algorithm 1. Strategy coordination among resource-constrained activities

- *For(int* $i = 1; i <= |R|; i++)$
  Set the initial $N_R[i]$*;*
- $for\ \forall t_j \in T$
  {
  1) $int\ tag = 1;$
  2) $for\ \forall r_i \in R_{t_j}:$
  $if\ N_R[i] == 0, then\ tag = 0;$
  3) $if\ tag == 0, then$
  { 3.1) $for\ \forall r_i \in R_{t_j}:\ N_R[i]--;$
  3.2) $Execute\ t_j;$
  3.3) $for\ \forall r_i \in R_{t_j}:\ N_R[i]++;$}
  }

3.3 Conditional Strategy and Social Law Determination

Now, we will propose the concept of conditional social strategy to determine the social law for the organizational structure.

Definition 9. *Conditional social strategy.* Given an environment $\langle A, S, P\rangle$, let agent a_i execute activity t_j. Now the 1st-order dependency structure of agent $a_i \in A$ is Dep_{a_i}, the set of social strategies available to the agents of Dep_{a_i} is $S_{\mho_{ai}}$; the set of former activities of t_j is Δ_{t_j}, the set of social strategies available to the activities Δ_{t_j} is $S_{\Delta_{t_j}}$; the set of resource-constrained activities of t_j is $\sum_m \Re_j^m$, the set of social strategies available to the activities $\sum_m \Re_j^m$ is $S_{\sum_m \Re_j^m}$. Therefore, the final set of social strategies available to agent a_i to execute activity t_j is called the conditional social strategy

$$S_i^j(a_i, t_j | \mho_{a_i}, \Delta_{t_j}, \sum_m \Re_j^m). \tag{17}$$

Therefore, the social law of the whole system can be the joint distribution of the conditional social strategies for all agents and activities which can satisfy the requirements of organizational structure of the workflow system. We have:

$$SL = \bigcap_{ij} S_i^j(a_i, t_j | \mho_{a_i}, \Delta_{t_j}, \sum_m \Re_j^m). \tag{18}$$

Therefore, if we want to set the social strategies of an agent to execute an activity, we will first set the strategies for all agents and activities of its 1st-order dependency structure, 1st-order former structure and resource-constrained activities. The idea in our algorithm is expressed as follows:

1. First, we consider the two kinds of constraints: constraints brought by the interaction structures among agents, and constraints brought by the temporal relations among actions. First we can select an agent with no “in” links or whose “in” links were all marked, and set the social strategy for it according to Equation 17; then we mark all “out” links of that agent. We will repeat such iteration until the agents without unmarked “in” links can not be found. By the same rule, we can also set strategies for the constraints brought by the temporal relations among actions.

2. Second, we consider the constraints brought by the sharing of critical resources among activities. Now we can use Algorithm 1 to do this.

Therefore, the final algorithm for the social law determination is shown as Algorithm 2.

Algorithm 2. Organizational structure – satisfactory social law determination

- *input the organizational structure of multiagent workflow system.*
- $for\ \forall a_i \in A$
  $for\ \forall t_j \in T$
  $\{$*Set the initial* S_i^j;
  Compute the Dep_{a_i} *and* Dom_{a_i}, For_{t_j} *and* $Lat_{t_j}\}$;
- *creatstack (stack)*; //create a stack.
- $for\ \forall a_i \in A$
  $for\ \forall t_j \in T$
  $\{$*if* $(Dep_{a_i} = \{\})\&(For_{t_j} = \{\})$;
  push $(\langle a_i, t_j\rangle, stack)\}$;
- *whlie(!empty(stack)) do* :
  1)$\langle a_i, t_j\rangle = pop(stack)$;
  2)*Set the* S_i^j;
  3)*for agent* $a_m \in \Omega_{a_i}$ *do* :
  $Dep_{a_m} = Dep_{a_m} - \{\langle a_i, a_m\rangle\}$;
  4)*for activity* $t_n \in \nabla_{t_j}$ *do* :
  $For_{t_n} = For_{t_n} - \{\langle t_j, t_n\rangle\}$;
  5)*if* $(Dep_{a_m} = \{\})\&(For_{t_n} = \{\})$:
  push$(\langle a_m, t_n\rangle, stack)$;
- *if(!empty(stack)),return (error)*;
  else set the S_i^jagain for critical resources-sharing constraints by calling Algorithm 1.

Algorithm 2 is $O(n \cdot m)$, where n denotes the number of agents, m denotes the number of activities.

Theorem 1. The social law determined by Algorithm 2 is a useful social law for the organizational structure of multiagent workflow system.

Proof. The constraints brought by organizational structure include three parts: 1) constraints brought by the interaction structures among agents; 2) constraints brought by the temporal relations among actions; 3) constraints brought by the sharing of critical resources among activities. Therefore, now we will prove that our algorithm can avoid the conflicts brought by those three kinds of constraints.

1. Let there be two agents a_i and a_j, if a_i is in the 1^{st}-order dependency structure of a_j, obviously our algorithm can avoid the conflicts brought by the interaction structures. Now we will consider the situation where a_i is in the n^{th}-order $(n > 1)$ dependency structure of a_j. According to Algorithm 2, a_j can set its strategy only after all its 1^{st}-order dependency agents,$\mho_{a_j}$, are set; for $\forall a_x \in \mho_{a_j}$, a_x can set its strategy only after all its 1^{st}-order dependency agents (which are the 2^{nd}-order dependency agents of a_j) are set. Such process will repeat until all agents in a_j's all-orders dependency structures are set. Since a_i is in the n^{th}-order $(n > 1)$ dependency structure of a_j, thus a_j will set its strategy according to its

dependency structure from a_i. Therefore, the conflicts brought by the interaction structures between a_i and a_j will be avoided.

2. Let there be two activities t_i and t_j; if t_i is in the 1^{st}-order former structure of t_j, obviously our algorithm can avoid the conflicts brought by the temporal relations between them. Now we will consider the situation where t_i is in the n^{th}-order $(n > 1)$ former structure of t_j. According to Algorithm 2, t_j can set its strategy only after all its 1^{st}-order former activities, Δ_{t_j}, are set; for $\forall t_x \in \Delta_{t_j}$, t_x can set its strategy only after all its 1^{st}-order former activities (which are the 2^{nd}-order former activities of t_j) are set. Such process will repeat until all activities in t_j's all-orders former structures are set. Since t_i is in the n^{th}-order $(n > 1)$ former structure of t_j, t_j will set its strategy according to its former structure from t_i. Therefore, the conflicts brought by the temporal relation between t_i and t_j will be avoided.

3. Let there be two activities t_i and t_j, and they will access the same critical resources. Now, Algorithm 2 will call Algorithm 1 to make strategy coordination between them. In Algorithm 1, while an activity accesses a critical resource, it will first set a tag to exclude other activities' accesses; only after it finishes its execution, it can release the critical resources. Therefore, for any critical resource, it can only be accessed by one activity simultaneously, thus the conflicts brought by the critical resource sharing between t_i and t_j will be avoided.

Therefore, Algorithm 2 can guarantee that the strategies of agents to execute activities satisfy the requirements of organizational structure of multiagent workflow system, i.e., the collisions among agents and activities can be avoided. Thus we can get Theorem 1. □

4 ADJUSTMENTS OF SOCIAL LAWS FOR STRUCTURE ALTERNATIONS

4.1 Adjustment for the Alternation of Agent Interaction Structures

In the operations of workflow system, some new interaction relations may be added into the interaction structure as well as some old interaction relations may be deleted from the existing interaction structure. Now we should make some adjustment for such alternation of agent interaction relations.

Let an existing relation $\langle a_i, a_j \rangle$ be deleted from the interaction structure, or a new relation $\langle a_i, a_j \rangle$ be added into the interaction structure. Now the social law in multiagent workflow system can be adjusted on the strategies of a_j and a_j's all orders domination agents. Let the set of activities executed by agent a_x be T_x; the adjustment for the alternation of agent interaction relations is shown as Algorithm 3.

Algorithm 3. Adjustment for the alternation of agent interaction relations

- $for\ \forall t_m \in T_j$: Re-set s_j^m;
- Create $Queue$(Q);
- Insert (Q, a_j)
- *while* (!empty(Q)) *do*
  1) $a_{temp1} = outQueue(\mathrm{Q})$;
  2) $for\ \forall a_{temp2} \in \Omega_{a_{temp1}} do:$
     {For$\forall t_m \in T_{temp2}$: Re-set s^m_{temp2};
     Insert(Q,a_{temp2})};

Theorem 2. The social law after adjustment of Algorithm 3 is a useful social law for the new organizational structure of multiagent workflow system.

Proof. While interaction relation $\langle a_i, a_j\rangle$ is changed, obviously the strategies of a_j will be influenced; if a_j's strategies are changed, the strategies of its 1st-order domination agents will also be influenced. By the same token, the strategies of a_j's all-orders domination agents will be influenced. Obviously, we should re-set the strategies of a_j's all-orders domination agents step by step. Therefore, while $\langle a_i, a_j\rangle$ is changed, only the strategies of a_j and a_j's all-orders domination agents need to be adjusted. Algorithm 3 re-set the strategies of a_j and a_j's all-orders domination agents conditionally if their dependent agents are set, which accords with Equation (17). Therefore, the conflicts brought by the alternation of $\langle a_i, a_j\rangle$ can be avoided with Algorithm 3, thus the social law after adjustment of Algorithm 3 is a useful social law for the new organizational structure of multiagent workflow system. □

4.2 Adjustment for the Alternation of Temporal Flow of Activities

In the workflow system operations, some new temporal relations may be added into the temporal flow as well as some old temporal relations may be deleted from the existing temporal flow. Now we should make some adjustment for such alternation of activity temporal flow.

Let an existing temporal relation $\langle t_i, t_j\rangle$ is deleted from the temporal flow, or a new relation $\langle t_i, t_j\rangle$ is added into the temporal flow. Now the social law in the multiagent workflow system can be adjusted on the strategies of t_j and t_j's all orders latter activities.

Algorithm 4. Adjustment for the alternation of activity temporal flow

- Re-set the temporal strategy of t_j;
- Create $Queue$(Q);
- Insert (Q, t_j)
- *while* (!empty(Q)) *do*
  1) $t_{temp1} = outQueue(\mathrm{Q})$;
  2) $for\ \forall t_{temp2} \in \nabla_{a_{temp1}} do:$
     {Re-set the temporal strategy of t_{temp2};
     Insert(Q,t_{temp2})};

Theorem 3. The social law after adjustment of Algorithm 4 is a useful social law for the new organizational structure of multiagent workflow system.

Proof. While temporal relation $\langle t_i, t_j \rangle$ is changed, obviously the temporal strategies of t_j will be influenced; if t_j's temporal strategies are changed, the temporal strategies of its 1^{st}-order latter activities will also be influenced. By the same token, the temporal strategies of t_j's all-orders latter activities will be influenced. Obviously, we should re-set the temporal strategies of t_j's all-orders latter activities step by step. Therefore, while $\langle t_i, t_j \rangle$ is changed, only the temporal strategies of t_j and t_j's all-orders latter activities need to be adjusted. Algorithm 4 re-set the temporal strategies of t_j and t_j's all-orders latter activities conditionally if their former activities are set, which accords with Equation (17). Thus the conflicts brought by the alternation of $\langle t_i, t_j \rangle$ can be avoided with Algorithm 4, so the social law after adjustment of Algorithm 4 is a useful social law for the new organizational structure of multiagent workflow system. □

4.3 Adjustment for the Alternation of Resource Constraints

In the workflow system operations, some activities may change their accessing strategies on critical resources. For example, an activity may cancel some of its accessing critical resources, or switch to access some other critical resources. Therefore, for an activity t_i, if it alters its accessing strategies on critical resources (RM), its resource-constrained activities $\sum_{r_m \in RM} \Re_i^m$ will be changed. In fact, now we can also use Algorithm 1 to adjust the strategies of all activities.

Theorem 4. While the resource constraints are changed in the operations of workflow systems, Algorithm 1 can make the accessing strategies on critical resources of all activities not to collide with each other.

Proof. We can see from Algorithm 1 that any critical resource can only be accessed by one agent at the same time, which is realized by the exclusive accessing tags. Therefore, no matter how the resource constraints are changed, any critical resource can not be accessed by more than one agent simultaneously. Therefore, the accessing on critical resources of all activities can not collide with each other. □

5 CONCLUSION

In a multiagent workflow system, agents are constrained by their organizational relations, such as interaction structure, activity sequences and resource sharing. Those relations can organize the multiagents together and each agent locates on a given position in the organizational structure. Therefore, while agents execute activities, their strategies should be designed to satisfy the organizational structure. To restrict the strategies to avoid collision in the system, we propose a model for social law determination.

In this paper, we mainly consider three kinds of constraints on organizational structures in multiagent workflow systems: 1) constraints brought by the interaction structures among agents; 2) constraints brought by the temporal relations among actions; 3) constraints brought by the sharing of critical resources among activities. We proposed the model and algorithms on how to endow social laws into the systems to satisfy the constraints of organization structure, thus to minimize the conflicts among agents and actions. The purpose of our model is to present a general guideline for determining social laws according to the organizational structures. Therefore, though our model is mainly explained by the case of multiagent workflow system, it can be also used for other general organizational systems. While we want to determine the social law for other general organizational systems, we only need to design an appropriate conditional strategy.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 60803060), the National High Technology Research and Development Program of China (863 Program, No. 2009AA01Z118), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 200802861077, No. 20090092110048), the Program for New Century Excellent Talents in University, State Education Ministry of China (NCET-09-0289) , and the Excellent Young Teachers Program of Southeast University (No. 4050181013).

REFERENCES

[1] Lin, D.: Compatibility Analysis of Local Process Views in Interorganizational Workflow. Proceedings of IEEE International Conference on the 9th E-Commerce Technology, and the 4th Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007), 23–26 July 2007, Tokyo, Japan, pp. 192–199.

[2] Savarimuthu, B. T. R.—Purvis, M.—Fleurke, M.: Monitoring and Controlling of a Multi-Agent Based Work-Flow System. Proceedings of the Australasian Workshop on Data Mining and Web Intelligence (DMWI 2004), Dunedin, New Zealand, 21 January 2004, pp. 127–132.

[3] Vidal, J. M.—Buhler, P.—Stahl, Ch.: Multiagent Systems with Workflows. IEEE Internet Computing, Vol. 8, 2004, No. 1, pp. 76–82.

[4] Huhns, M. N.—Singh, M. P.: Workflow Agents. IEEE Internet Computing, Vol. 2, 1998, No. 4, pp. 94–96.

[5] Chang, J. W.—Scott, C. T.: Agent-Based Workflow: TRP Support Environment (TSE). Computer Networks and ISDN Systems, Vol. 28, 1996, No. 7–11, pp. 1501–1511.

[6] Davis, R.—Smith, R.: Negotiation As a Metaphor for Distributed Problem Solving. Artificial Intelligence, Vol. 20, pp. 63–109, 1983.

[7] Kraus, S.: Negotiation and Cooperation in Multi-Agent Environment. Artificial Intelligence, Vol. 94, pp. 79–97, 1997.

[8] Jiang, Y.—Jiang, J. C.: A Multi-Agent Coordination Model for the Variation of Underlying Network Topology. Expert Systems with Applications, Vol. 29, 2005, No. 2, pp. 372–382.

[9] Jiang, Y.—Jiang, J.: Contextual Resource Negotiation-Based Task Allocation and Load Balancing in Complex Software Systems. IEEE Transactions on Parallel and Distributed Systems, Vol. 20, 2009, No. 5, pp. 641–653.

[10] Buckley, S. J.: Fast Motion Planning for Multiple Moving Robots. Proceeding of 1989 IEEE International Conference on Robotics and Automation, Scottsdale, AZ 1989, pp. 322–326.

[11] Tomlin, C.—Pappas, G. J.—Sastry, S.: Conflict Resolution for Air Traffic Management: A Case Study in Multi-Agent Hybrid Systems. IEEE Transactions on Automatic Control, Vol. 43, 1998, No. 4, pp. 509–521.

[12] Davis, R.—Smith, R. G.: Negotiation as a Metaphor for Distributed Problem Solving. Artificial Intelligence, Vol. 20, 1983, No. 1, pp. 63–109.

[13] Shoham, Y.—Tennenholtz, M.: On Social Laws for Artificial Agent Societies: Off-Line Design. Artificial Intelligence, Vol. 73, pp. 231–252, 1995.

[14] Tennenholtz, M.: On Stable Social Laws and Qualitative Equilibria. Artificial Intelligence, Vol. 102, pp. 1–20, 1998.

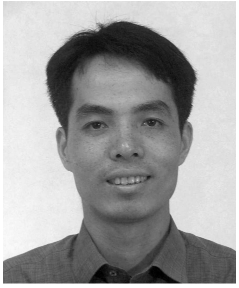
[15] Jiang, Y.—Ishida, T.: Local Interaction and Non-Local Coordination in Agent Social Law Diffusion. Expert Systems with Applications, Vol. 34, 2008, No. 1, pp. 87–95.

[16] Jiang, Y.: Extracting Social Laws from Unilateral Binary Constraint Relation Topologies in Multiagent Systems. Expert Systems with Applications, Vol. 35, 2008, No. 4, pp. 2004–2012.

[17] Boella, G.—van der Torre, L.: The Evolution of Artificial Social Systems. Proceeding of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 30 July–5 August 2005, pp. 1655–1656.

[18] Jiang, Y.: On Self-Adjustment of Social Conventions to Small Perturbations. Chinese Physics Letters, Vol. 25, 2008, No. 12, pp. 4215–4218.

[19] Jiang, Y.—Jiang, J.—Ishida, T.: Compatibility Between the Local and Social Performances of Multi-Agent Societies. Expert Systems with Applications, Vol. 36, 2009, No. 3-P1, pp. 4443–4450.



Yichuan Jiang received the Ph. D. degree in computer science from Fudan University, China, 2005. He is currently a Professor in the Lab of Complex Systems and Social Computing, School of Computer Science and Engineering, Southeast University, Nanjing, China. His main research interests include multiagent systems, complex distributed systems and social computing. He was awarded the New Century Excellent Talents in University of State Education Ministry of China, the Best Paper Award from PRIMA2006, and the Nomination Award for the National Excellent Doctoral Dissertation of China. He is also the recipient of the HwaYing Young Scholar Award in 2008 from The HwaYing Education and Culture Foundation. He has published more than 50 scientific articles in refereed journals and conference proceedings, such as IEEE TPDS, IEEE TSMC-C, IJCAI and AAMAS. He is a member of IEEE and IEEE Computer Society, a member of ACM, and a member of the editorial board of Chinese Journal of Computers.