Qualitative Description and Quantitative Optimization of Tactical Reconnaissance Agents System Organization

Xiong Li *
Department of Command and Administration, Academy of Armored Force Engineering
Beijing, China
E-mail: lixiong2609@126.com

Yonglong Chen
Department of Equipment Command, Academy of Equipment Command and Technology
Beijing, China
E-mail: jiansijia@sohu.com

Zhiming Dong
Faculty of Science Research, Academy of Armored Force Engineering
Beijing, China
E-mail: keny1003@163.com

Received 10 July 2011
Accepted 5 June 2012

Abstract

In this paper, the problem of qualitative description and quantitative optimization for tactical reconnaissance agents system organization is considered with objective of higher teamwork efficiency and more reasonable task balancing strategies. By analyzing tactical reconnaissance system and its environment, task-(role)-entity agent mapping mechanism and agents in system organization, the system framework is qualitatively described. By transforming the system into an interaction task request-service mechanism queuing system, a Markov chain of system state transition is obtained, since its state transition process in interaction is Markov process and accords with real tactical reconnaissance behaviors. By solving the state transition equations, the inherent relationship of tactical reconnaissance agents is found and the optimized system configuration is obtained. The established simulation demonstration system proves that the proposed approach and model are feasible and effective.

Keywords: agent; agent-based modeling and simulation; tactical reconnaissance; system organization; optimization; Markov process

1. Introduction

Intelligent agent [1-27] is a program that maps percepts to actions. It acquires information from its environment ("perceives" the environment) and decides about its actions and performs them. While there is no real consensus about the definition of intelligent agents, the above one adapted from [3] is intuitively clear and essentially describes the general concept of generic agent. All more specific intelligent agents can be derived from that concept. Thus a multi-agent system is a collection of agents co-operating with each other in order to fulfill common and individual goals (in some environments they may also compete). In a multi-agent system different agents often have different roles and

*Corresponding Author: Xiong Li
E-mail: lixiong2609@126.com

Published by Atlantis Press

individual goals [1, 3]. As for current fruits, multi-agent system technology has been used in many fields. Agent-oriented software engineering has been viewed as a novel paradigm for complex systems. The analysis and comprehensive understanding of complex systems is extremely difficult and in most cases impossible for humans to grasp without the assistance of advanced tools such as agent-oriented software engineering.

In information wars, a commander must comprehend the whole situations through various kinds of channels during engagement, and command promptly as soon as some situations occur according to collected, processed and composed battlefield intelligence. Therefore, the need of building fast and exact tactical reconnaissance system (TRS) has come to the fore. In highly distributed warfare system, it may be impossible for any of the individual battlefield nodes to perform a useful intelligence reconnaissance function without TRS.

On information battlefield, TRS is composed of a geographically dispersed organization of heterogeneous elements. All elements are tied together by a communications network with command and control applied at tactical centers. These elements include multiple intelligence reconnaissance platform members, such as photo-reconnaissance vehicles, electronic reconnaissance vehicles, armored reconnaissance vehicles, unmanned aerial vehicles (UAVs) and information processing vehicles. They have believes, desires, intentions, and may adopt a role and have relationships with each other in administrative levels. Therefore, it may be seen as a distributed multi-agent system revealing a kind of a synergy that would not be expected from the simple sum of its component agents. This synergy is an emergent property of the system as a whole. In this paper, we design an elementary agent-based model of TRS to lay a foundation for the advanced concept technology demonstration of intelligence reconnaissance actions.

In a multi-agent system, an interaction protocol contains the basic rules for the negotiation process and the communication. Thus it is explicit to point out existing and challenging problems that researchers are facing in designing and implementing this agent-based complex warfare system, such as

(1) how can we enforce the necessary teamwork, leading to coherent and effective results according to the overall system’s goals and making all tactical reconnaissance entity agents perform military actions and fulfill intelligence reconnaissance tasks with higher efficiency?

and (2) how can we design and implement this system in a way that avoids computational overload by means of intelligence reconnaissance tasks balancing strategies so that we can attain higher interaction efficiency?

However, current research on agent or agent-based modeling and simulation technology usually concentrates on domain specification, agent design and implementation problems [4-26]. Anita Raja and Victor Lesser develop a meta-level control approach based on the decision-theoretic use of an abstract representation of the agent state to resolve the meta-level control problem for agents operating in resource-bounded multi-agent environments [4]. David Meignan, Olivier Simonin and Abderrafiaa Koukam present a bus network simulation tools and adopt a multi-agent approach to describe the global system operation as behaviors of numerous autonomous entities such as buses and travelers [7]. Generally, these approaches are proposed to give agent and multi-agent system concept developers a view to the future by enabling them to simulate and manipulate in near real-time, the assets and operational conditions.

Unfortunately, there exists deficiency on agents system configuration for higher teamwork efficiency and more reasonable task balancing strategies. This is substantially a technology problem about tasks being effectively performed through system organization optimization.

The principal aim of this paper is to suggest possible model and approach to the above problem by using Markov chain method based on system organization description, interaction chain design and transformation from interaction chain to interaction task request-service mechanism queuing system.

The paper is organized as follows. In Section 2, system framework of TRS is described, followed in Section 3 with the queuing system model and quantitative calculation for TRS. In Section 4, performance demonstration for the model by computer simulation is presented. The main results are summarized in Section 5.

2. System Framework

As the term organization is used in multiple ways, it is necessary to specify that in this paper we consider mainly process-related type of organizations (organization as task or action). Actually, the research objects of system organization optimization also include functional organizations (organizations as permanent structures) and institutional organizations (as an actual purposeful structure within a social context). Since system framework description reflects subsystem-level representation, it is the precondition and basis of quantitative optimization for system organization.

2.1. *TRS and Its Environment*

Land warfare is a complex adaptive system. That is to say, land warfare is essentially a nonlinear dynamical system composed of many interacting autonomous and hierarchically organized agents continuously adapting to a changing environment [29]. Thus, complex warfare system can be defined as a coordinated group of individuals who collaborate for a common military task, on the basis of some tactical rules and resources. It can be represented in terms of the interactions and behaviors of warfare entities. These include standard interactions such as operations orders, time-driven information such as status reports and triggered exchanges such as spot reports and fragmentary orders. There is a close match between the real-world warfare process and the entity paradigm. Based on intelligence reconnaissance, the effects of a commander decision process are communicated to other members through message exchanges across the battlefield, and eventually result in action by battle group members.

Complex warfare system consists of a number of real warfare members. These members can be TRS members or other system members. During engagement a member can become a commander while stay as a battle member and a commander can become a battle member. On battlefield, a commander must have access to information about the world through his own direct sensations of the world and reported information from other members.

In complex warfare system, TRS plays a pivotal role, which is a set of elements that are interconnected so that changes in some elements, or their interrelations, produce changes in other parts of the system. Tactical reconnaissance operations are usually arranged in a hierarchical fashion with decentralized control, similar to a military chain-of-command. For example, the general reconnaissance vehicles cooperate with intelligence reconnaissance (intercommunicating, reporting to an intelligence processing vehicle, awaiting orders from commander); intelligence processing vehicles perform data fusion and processing (sorting intelligence, filtrating intelligence, analyzing situation, making decision, producing schemes, saving and updating information), and control the general reconnaissance vehicles (notifying reconnaissance requirements, notifying targets information, allocating tasks and adjusting tasks).

Thus, we can set up a mapping from TRS's internal members to entities, such as photo-reconnaissance vehicle → photo-reconnaissance vehicle entity, UAV → UAV entity, and information processing vehicle → information processing vehicle entity. To express and illustrate organization relation about the entities and their tasks and roles [22, 28] is the key for system organization optimization.

2.2. *Task, Role and Entity*

Traditional analysis on task / sub-task and entity, always centers on a tighten coupling of task-entity, as shown in Figure 1.

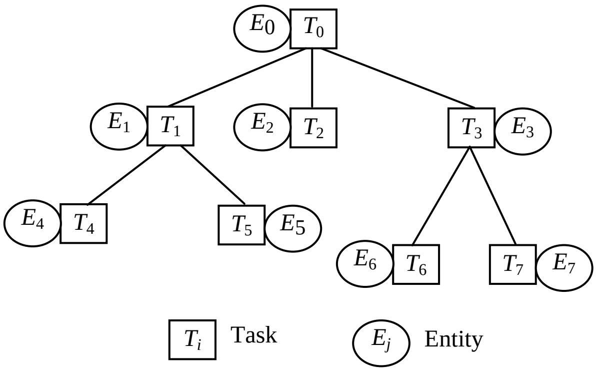


Fig. 1. Tighten coupling of task-entity.

Whenever entities have to work in a group setting, interactions take place to find a suitable organization (who does what) as well as to enable communication of results (when and to whom). All these interactive activities imply the need for a clear policy for co-ordination [3]. Reasoning about communication has to take into account the amount of interaction, demanded performance and resources, future activities and loads, imposed deadlines, and also knowledge representing an entity's desires, intentions, and beliefs. The larger the number of different possibilities and the set of the identified constraints for entities' joint work are, the richer the co-ordination policies will be.

In the case where entities are self-interested, interaction aims at maximizing an entity's utility. In the case where entities share an overall goal, the objective is to maintain global coherence without violating autonomy, thus avoiding explicit global control [4]. Performing tasks in the context of TRS could be considered in the scope of co-ordination of the entity's activities in a dynamic environment where resources may be scarce.

Based on the above analysis, we can obtain interaction tasks performing system framework description by establishing a task-(role)-entity mapping mechanism, shown in Figure 2, in which an entity can play a role, as agent, to fulfill tactical reconnaissance tasks with more flexibility in the TRS model.

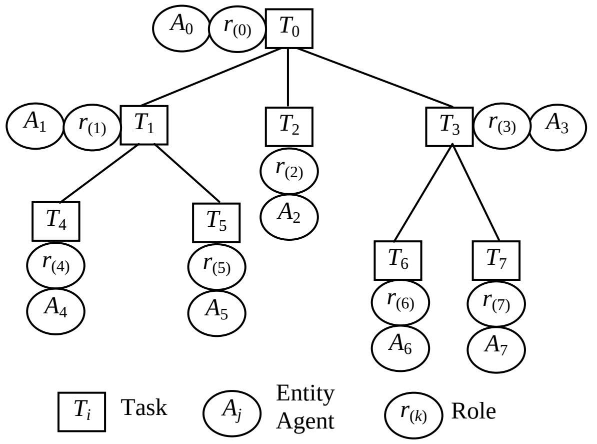


Fig. 2. Flexible coupling of task-(role)-entity agent.

By this method, we can conveniently implement subsystem-level representation for establishing the interaction tasks optimization model.

2.3. *Tactical Reconnaissance Agents in System Organization*

A tactical reconnaissance agent with human being properties such as autonomy, sociality, adaptability and information can act as a human to perform intelligence reconnaissance tasks in teamwork mode. Especially tactical reconnaissance agents system considers how a group of intelligent and autonomous reconnaissance entity agents coordinate their capacities and plan in order to achieve certain military goals. Tactical reconnaissance agents may be seen as a natural extension of the concept of software objects. Object-oriented programming added abstraction entities, i.e., objects, which have persistent local states to the structured programming paradigm. Similarly, agent-based programming adds abstraction entities, i.e., agents, which have an independent execution thread to the object-oriented paradigm. Thus, compared to an object, a tactical reconnaissance agent is able to act in a goal-directed fashion (e.g., by interacting with other agents, reading sensors, or sending commands to effectors) rather than only passively react to procedure calls, as shown in Figure 3.

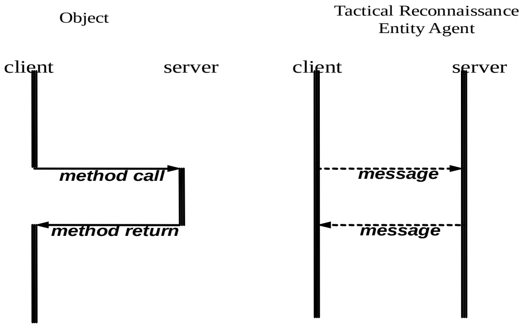


Fig. 3. Comparison of an object and a tactical reconnaissance entity agent.

In addition, a tactical reconnaissance entity agent typically has also one or more of the following abilities: to communicate with other tactical reconnaissance entity agents, to learn from experience and adapt to changes in the environment, to make plans, to reason using, e.g., logic or game theory, to negotiate with other tactical reconnaissance entity agents. Also, tactical reconnaissance entity agents are sometimes programmed, or at least modeled, in terms of "mental states", such as, believes, desires, and intentions. Tactical reconnaissance agents system is an organization of reconnaissance entity agents in teamwork mode to perform common military goals concentrating on battlefield intelligence reconnaissance.

According the above analysis, TRS can be mapped as a multi-agent system since the internal members of TRS has been mapped as agents, i.e., photo-reconnaissance vehicle → photo-reconnaissance vehicle agent, UAV → UAV agent, and information processing vehicle → information processing vehicle agent.

We can further define these tactical reconnaissance entity agents as two multi-agent sub-systems as follows:

$MAS_1=A=\{A_i\}$;

$MAS_2=B=\{B_i\}$;

where MAS_1 consists of all general reconnaissance platform agents and MAS_2 consists of all information processing platform agents. They exchange information by network system (NS). Thus, tactical reconnaissance agents system can be illustrated as Figure 4.

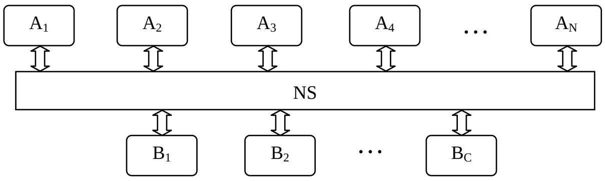


Fig. 4. Tactical reconnaissance agents system organization.

Therefore, tactical reconnaissance agents system organization is transformed as a computational multi-agent system in which two or more reconnaissance entity agents interact and work together to perform a set of reconnaissance tasks and to satisfy a set of military goals. The system consists of different intelligent nodes that are represented by reconnaissance entity agents respectively, as is discussed above. A key characteristic of tactical reconnaissance agents system is a lot of different tactical reconnaissance entity agents with information interaction (See Figure 5).

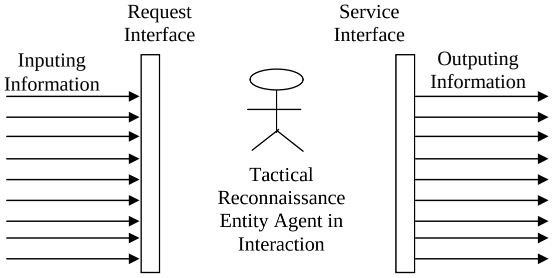


Fig. 5. A tactical reconnaissance entity agent and information interaction.

In our system the distributed request and service information interaction protocol has the following properties:

(1) The interaction leads to a finite number of states.

(2) The interaction process does not enter cyclic or infinite sequences but always reaches a terminal state.

3. Modeling and Solving

Based on qualitative description for tactical reconnaissance agents system organization, we can establish an interaction chain mechanism model, and transform it to a Markov chain model, thus propose the quantitative optimization approach. Our quantitative optimization model and approach are designed to demonstrate tactical reconnaissance behaviors on dynamic and distributed battlefield and explore some issues, such as the role of speed in information processing, how tasks balancing strategies might change when more information processing platforms are added to a force or force structure is changed, and how distribution of information processing capability affects overall force effectiveness.

3.1. *Queuing System Model*

The multi-agent teamwork with optimization tasks performing mechanism has the following features [26]:

(1) There is a seeding tactical reconnaissance entity agent who initiates the interaction.

(2) Each tactical reconnaissance entity agent can be active or inactive.

(3) Initially all tactical reconnaissance entity agents are inactive except for a specified seeding agent, which initiates the computation.

(4) An active tactical reconnaissance entity agent can do local computation, send and receive messages and can spontaneously become inactive.

(5) An inactive tactical reconnaissance entity agent becomes active, if and only if, it receives a message.

(6) Each tactical reconnaissance entity agent may retain its current belief, revise or update its belief as a result of receiving a new message by performing a local computation. If it modifies its belief, it communicates its new belief to other concerned agents; else it does not modify its belief and remains silent. Figure 6 shows the interaction behaviors process.

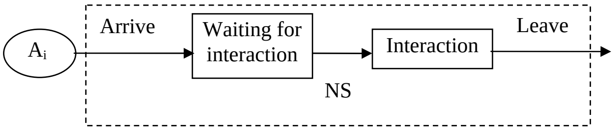


Fig. 6. Interaction behaviors process.

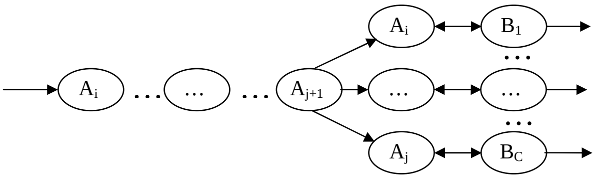


Fig. 7. Law of tasks performing priority.

The law of tasks performing priority [27] can be illustrated as Figure 7.

In our tactical reconnaissance agents system organization model, there are two classes of agents making up each chain-of-reconnaissance, i.e., general intelligence reconnaissance vehicle agents and information processing vehicle agents, which exchange information by interaction chain mechanism, as shown in Figure 8. Tactical reconnaissance agents within the same class have unique characteristics such as experience and unit cohesion. In addition, tactical reconnaissance agents following the same decision path, may take different amounts of time to complete the same tasks. This is accomplished by running each agent as a separate thread of execution or mini-program.

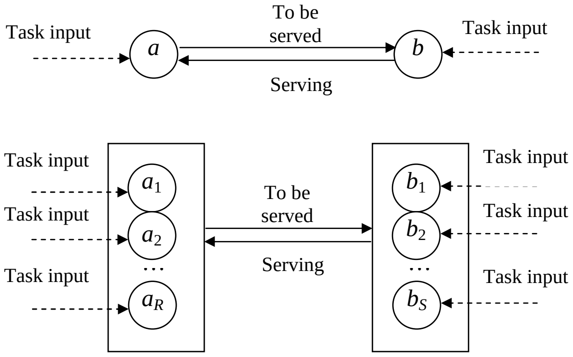


Fig. 8. Interaction chain of a served agent and a serving agent (above) and interaction chain of served agents and serving agents (under).

The above analysis is not enough when we want to probe into countermeasure of enhancing system efficiency to cater for the requirements on battlefield intelligence reconnaissance. The speed at which a tactical reconnaissance entity agent moves through its decision cycle is driven by internal factors such as unit cohesion and external factors such as resources available. This is in substance a problem on multi-agent organization optimization. We assume that served agents work independently and bring forward randomly requests on tactical reconnaissance tasks, and the probability of having *k* agents bringing forward requests in time *t* is a Poisson distribution, i.e., $F_k(\lambda t)=\frac{(\lambda t)^k e^{-\lambda t}}{k!}$, where λ is the rate of bringing forward requests and λt is the Poisson distribution parameter.

By military experts' evaluation, these hypotheses are consistent with real tactical intelligence reconnaissance actions, i.e., they reflect battlefield intelligence collecting and processing rules in some senses. In addition, according to the military experiences on tactical reconnaissance process on distributed battlefield, we can think that there is only one intelligence reconnaissance task produced by an arbitrary served agent at any time, i.e., it is impossible for two or more intelligence reconnaissance tasks to be produced by a served agent simultaneity. Of course, it is reasonable to think that an intelligence reconnaissance task can be performed by a serving agent. Moreover, as far as the military experiences, the serving time of a serving agent for a served agent can be assumed to be an exponentially distribution, i.e., $F(t)=1-e^{-\mu t}$, where μ is the serving rate. Thus we can establish a model of interaction task request-service mechanism queuing system, which is shown as Figure 9. In this queuing system, we assume that there are *n* served agents and *s* serving agents.

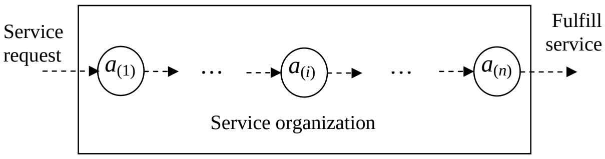


Fig. 9. Interaction task request-service mechanism queuing system.

Since the system changes randomly, it is generally impossible to predict the exact state of the system in the future. However, the statistical properties of the system's future can be predicted. In many applications it is these statistical properties that are important. The changes of state of the system are called transitions, and the probabilities associated with various state-changes are called transition probabilities.

In this system, obviously, when $n\le s$, for *n* served agents, the total mean serving rate of the system with *s* serving agents is $n\mu$; when $n\ge(s+1)$, for *n* served agents, the total mean serving rate is $s\mu$ since the system has only *s* serving agents. The total number of entities in this system increases 1, when a certain served agent arrives; it decreases 1 while a certain served agent leaves. Thus, we can analyze exactly the state transitions.

The rate diagram (see Figure 10) expresses the state transitions and its rate matrix is given in Eq. (1).

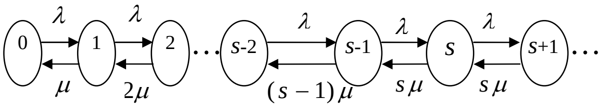


Fig. 10. Rate Diagram.

$$Q=\begin{bmatrix} q_{00} & q_{01} & q_{02} & \cdots & \cdots \\ q_{10} & q_{11} & q_{12} & \cdots & \cdots \\ q_{20} & q_{21} & q_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}=\begin{bmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots \\ 0 & 2\mu & -(\lambda+2\mu) & -\lambda & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (1)$$

In Figure 10, edges in diagram represent rates. Thus, there is a certain rate of transition to each other connected state (no need for self-edges). In Eq. (1), q_{ij} represents a certain rate of transition from state *i* to state *j*. It is obvious that the rate diagram and the rate matrix are consistent and reflect the state transitions analyzed as above.

3.2. *Solving by Using Markov Chain Method*

Formally, a Markov chain [30-36] is a discrete (discrete-time) random process with the Markov property. Often, the term “Markov chain” is used to mean a Markov process which has a discrete (finite or countable) state-space. Usually a Markov chain would be defined for a discrete set of times (i.e. a discrete-time Markov chain) although some authors use the same terminology where “time” can take continuous values.

A Markov chain is a sequence of random values whose probabilities at a time interval depend upon the value of the number at the previous time. The controlling factor in a Markov chain is the transition probability, which is a conditional probability for the system to go to a particular new state, given the current state of the system.

A “discrete-time” random process involves a system which is in a certain state at each “step”, with the state changing randomly between steps. The steps are often thought of as time, but they can equally well refer to physical distance or any other discrete measurement; formally, the steps are just the integers or natural numbers, and the random process is a mapping of these to states [32-39].

The Markov property states that the conditional probability distribution for the system at the next step (and in fact at all future steps) given its current state depends only on the current state of the system, and not additionally on the state of the system at previous steps.

In fact, in our problem the probability is only relevant to the previous time, i.e., future evolution of random process depends only on current state. Therefore the process is Markov Process and can be represented by a discrete-time Markov Chain with state transitions.

According to Figure 11, we can obtain the state transition equations as follows:

$$\frac{dP_i(t)}{dt}=\sum_{k\in I}P_k(t)q_{ki}$$
$$=-P_i(t)(\lambda+i\mu)+P_{i-1}(t)\lambda+P_{i+1}(t)(i+1)\mu \quad (2)$$

In Eq. (2), $P_i(t)$ represents the probability of the system in state *i* in time *t*.

As $P_i=\lim_{t\to+\infty}P_i(t)$, which represents the steady-state probability of the system in balance state *i*, $\frac{dP_i(t)}{dt}=0$.

Accordingly, Eq. (2) can be transformed as follows:

$$-P_i(\lambda+\mu_i)+P_{i-1}\lambda+P_{i+1}\mu_{i+1}=0 \quad (3)$$

where

$$\mu_i=\begin{cases} i\mu \ (i=1,2,...,s) \\ s\mu \ (i=s,s+1,...) \end{cases}$$

If we define $P_{-1}(t)=0$, then by Eq. (3) we can get

$$-\lambda P_0+\mu P_1=0 \quad (4)$$
$$-(\lambda+\mu)P_1+\lambda P_0+2\mu P_2=0 \quad (5)$$
$$-(\lambda+2\mu)P_2+\lambda P_1+3\mu P_3=0 \quad (6)$$
$$\cdots$$
$$-[\lambda+(s-1)\mu]P_{s-1}+\lambda P_{s-2}+s\mu P_s=0 \quad (7)$$
$$-(\lambda+s\mu)P_s+\lambda P_{s-1}+s\mu P_{s+1}=0 \quad (8)$$
$$\cdots$$

According to Eq. (4) we can calculate the steady-state probability of the system in state 1, i.e., P_1 (expressed by P_0). According to Eq. (5) we can calculate P_2 (expressed by P_0) by P_1 in the above result. In the light of this recursive approach, we can calculate P_k. In balance state *k*,

When $0\le k\le s$

$$P_k=\frac{\lambda\lambda...\lambda}{(k\mu)[(k-1)\mu]...(2\mu)\mu}P_0=\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k P_0; \quad (9)$$

When $k\ge s+1$

$$P_k=\frac{\lambda\lambda...\lambda}{(s\mu)[(s-1)\mu]...(2\mu)\mu}\frac{\lambda\lambda...\lambda}{(s\mu)(s\mu)...(s\mu)}P_0$$

$$=\frac{1}{s!}\frac{1}{s^{k-s}}\left(\frac{\lambda}{\mu}\right)^k P_0. \quad (10)$$

Because $\sum_{k=0}^{+\infty}P_k=1$, i.e., $P_0+\sum_{k=1}^{s-1}P_k+\sum_{k=s}^{+\infty}P_k=1$,

$$P_0 = \frac{1}{1+\sum_{k=1}^{s-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \sum_{k=s}^{+\infty}\frac{1}{s!}\left(\frac{1}{s}\right)^{k-s}\left(\frac{\lambda}{\mu}\right)^k}$$

$$= \frac{1}{\sum_{k=0}^{s-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \frac{1}{(s-1)!}\left(\frac{\lambda}{\mu}\right)^{s-1}\sum_{k=1}^{+\infty}\left(\frac{\lambda}{s\mu}\right)^k} \qquad (11)$$

If we define $\rho = \frac{\lambda}{s\mu}$, when $\rho < 1$,

$$\sum_{k=1}^{+\infty}\left(\frac{\lambda}{s\mu}\right)^k = \sum_{k=1}^{+\infty}\rho^k = \frac{\rho}{1-\rho} = \frac{\lambda}{s\mu-\lambda}. \qquad (12)$$

Then by Eq. (12) we can calculate P_0 in Eq. (11) as follows:

$$P_0 = \frac{1}{\sum_{k=0}^{s-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \frac{1}{(s-1)!(s-\frac{\lambda}{\mu})}\left(\frac{\lambda}{\mu}\right)^s}. \qquad (13)$$

If we put P_0 in Eq. (13) to the expressions of P_k in Eq. (9) and Eq. (10), then we can obtain P_k.

The main parameters are listed as follows:

$$L_q = \sum_{k=s}^{+\infty}(k-s)P_k = \sum_{i=0}^{+\infty} iP_{s+i} = \sum_{i=0}^{+\infty} i\frac{\rho^i}{s!}\left(\frac{\lambda}{\mu}\right)^s P_0$$

$$= P_0\frac{1}{s!}\left(\frac{\lambda}{\mu}\right)^s \rho\sum_{i=0}^{+\infty}\frac{d(\rho^i)}{d\rho} = P_0\frac{1}{s!}\left(\frac{\lambda}{\mu}\right)^s \rho\frac{d}{d\rho}[\sum_{i=0}^{+\infty}\rho^i]$$

$$= P_0\frac{1}{s!}\left(\frac{\lambda}{\mu}\right)^s \rho\frac{d}{d\rho}[\frac{1}{1-\rho}] = \rho\frac{1}{s!(1-\rho)^2}\left(\frac{\lambda}{\mu}\right)^s P_0$$

$$= \frac{1}{(s-1)!(s-\frac{\lambda}{\mu})^2}\left(\frac{\lambda}{\mu}\right)^{s+1} P_0 \cdot \qquad (14)$$

$$T_q = \frac{L_q}{\lambda} \qquad (15)$$

$$T = T_q + \frac{1}{\mu} = \frac{L_q}{\lambda} + \frac{1}{\mu} \qquad (16)$$

$$L = \lambda T = L_q + \frac{\lambda}{\mu} \qquad (17)$$

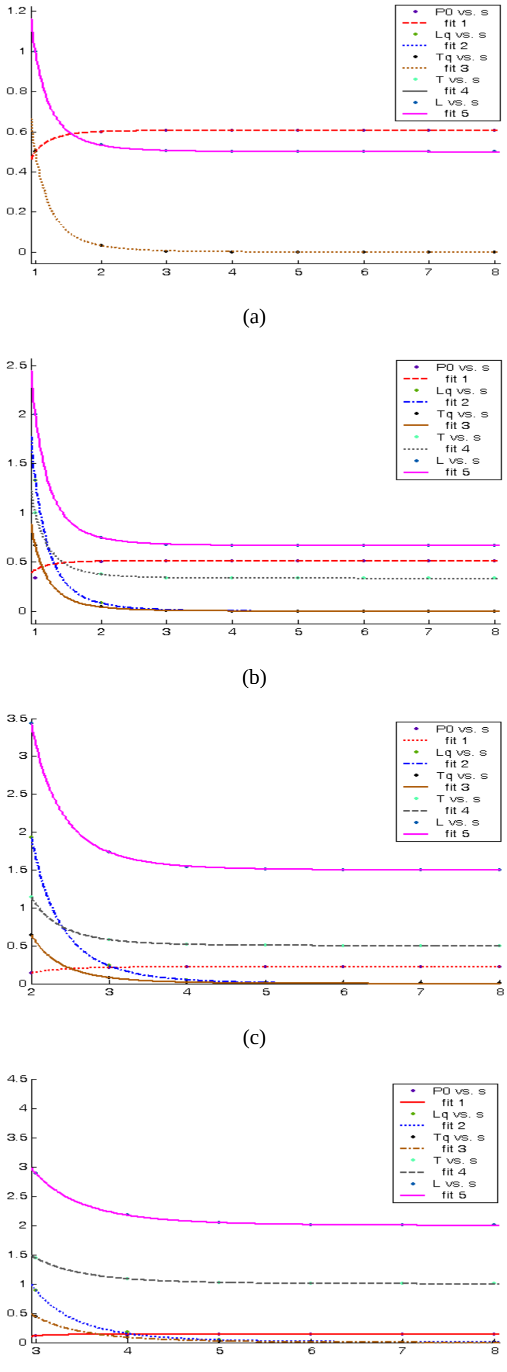
L_q represents the mean number of served agents waiting in line in the system;

T_q represents the mean time from a served agent arriving at the system to it starting to be served;

T represents the mean time from a served agent arriving at the system to it finishing being served and leaving;

L represents the total mean number of entity agents in the system.

These parameters substantially reflect interaction chain attributes, therefore which can be used as the factors of tactical reconnaissance agents system organization optimization.



(a)

(b)

(c)

(d)

Fig.11. Parameters variety from case 1 to case 4.

Case 1: $\lambda = 1$, $\mu = 2$, when the sum of serving agents increases from 1 to 8, parameters P_0, L_q, L, T_q, T alter as Figure 11(a).

Case 2: $\lambda = 2$, $\mu = 3$, when the sum of serving agents increases from 1 to 8, parameters P_0, L_q, L, T_q, T alter as Figure 11(b).

Case 3: $\lambda = 3$, $\mu = 2$, when the sum of serving agents increases from 2 to 8, parameters P_0, L_q, L, T_q, T alter as Figure 11(c).

Case 4: $\lambda = 2$, $\mu = 1$, when the sum of serving agents increases from 3 to 8, parameters P_0, L_q, L, T_q, T alter as Figure 11(d).

Thus we can obtain the optimized sum of serving agents in different conditions, as listed in Table 1.

Table 1. Optimized sum of serving agents in different conditions.

$\frac{\lambda}{\mu}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{2}$	2
Optimized sum of S	2	2	3	4

4. Performance Demonstration for the Model

Performance demonstration for our model is implemented by agent-based simulation. In the simulation demonstration system that we set up, multi-agent interactions relationship in military reconnaissance operations is given in Figure 12, in which T and t represent task inputs for Red Force agents and Blue Force agents respectively.

The designed entity agents in this simulation demonstration system are listed as follows:

r_1: UAV Agent 1 (Red Force);

r_2: UAV Agent 2 (Red Force);

r_3: Electronic Reconnaissance Vehicle Agent (Red Force);

r_4: Photo-reconnaissance Vehicle Agent (Red Force);

r_5: Armored Reconnaissance Vehicle Agent (Red Force);

r_6: Battlefield Sensor Agent 1 (Red Force);

r_7: Battlefield Sensor Agent 2 (Red Force);

r_8: Command Vehicle Agent (Red Force);

b_1: Electronic Reconnaissance Vehicle Agent (Blue Force);

b_2: Armored Reconnaissance Vehicle Agent (Blue Force);

b_3: Information Processing Vehicle Agent (Blue Force).

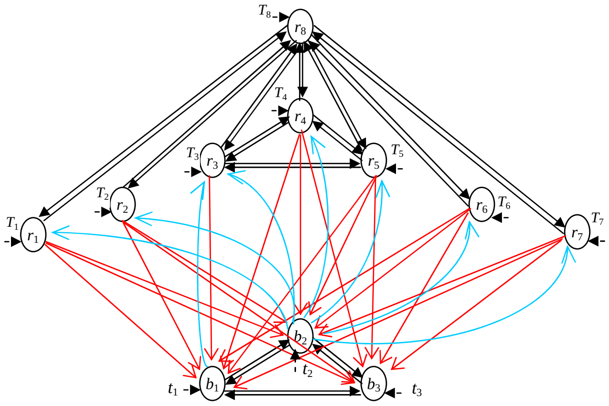
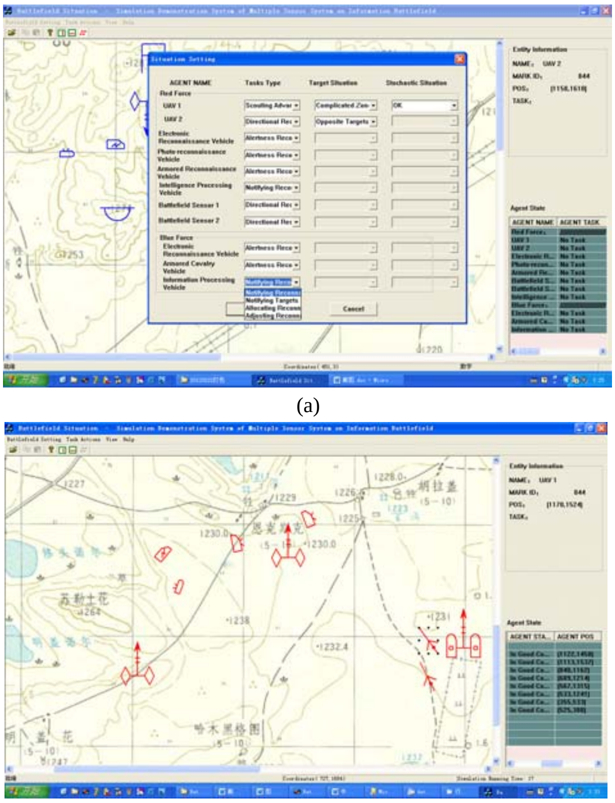


Fig.12. Multi-agent interactions relationship.

Simulation implementation can be illustrated by Figure 13. When we run the system, we can obtain some results, and find that these TRS entities performed successfully intelligence reconnaissance task on tactical virtual battlefield.


Battlefield Situation - Simulation Demonstration System of Multiple Sensor System on Information Battlefield
Situation Setting
Cancel
Entity Information
NAME: UAV 1
MARK ID: 644
TASK:
Agent State

AGENT NAME	Tasks Type	Target Situation	Stochastic Situation
Red Force			
UAV 1	Scouting Advar	Complicated Zon	OK
UAV 2	Directional Rec	Opposite Targets	
Electronic Reconnaissance Vehicle	Alertness Reco		
Photo-reconnaissance Vehicle	Alertness Reco		
Armored Reconnaissance Vehicle	Alertness Reco		
Intelligence Processing Vehicle	Notifying Reco		
Battlefield Sensor 1	Directional Rec		
Battlefield Sensor 2	Directional Rec		
Blue Force			
Electronic Reconnaissance Vehicle	Alertness Reco		
Armored Cavalry Vehicle	Alertness Reco		
Information Processing Vehicle	Notifying Reco Notifying Reconn Notifying Targets Allocating Reconn Adjusting Reconn		

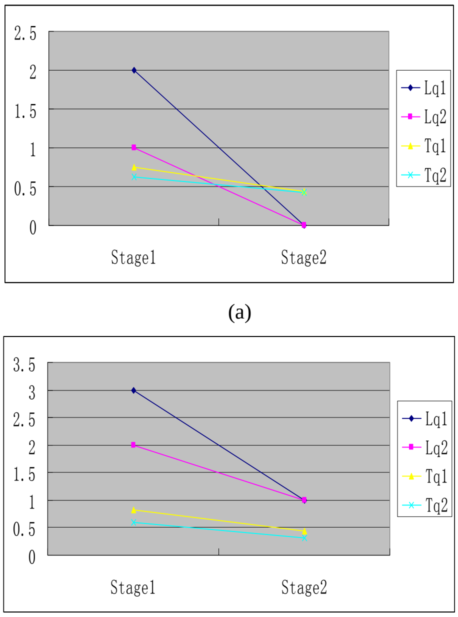
(a)

(b)

Fig.13. Simulation demonstration system.

Figure 13(a) presents initialization for entity agents situation setting for tactical reconnaissance simulation. In initialization, entity agents can be added and their initial tasks can be given according to real military requirements (See the central part in Figure 13(a)). Figure 13(b) presents partial, dynamic and real-time two-dimension battlefield situation information during agent-based TRS simulation. The real-time object information and state information of an agent (as a certain military role) during simulation (See the right parts in Figure 13(a) and Figure 13(b)) show that how a certain reconnaissance task is being performed by a certain appropriate entity.

The computational results proposed in Figure 12 can be testified by the agent-based TRS simulation. Here we take two main parameters, i.e., L_q and T_q, into account. For photo-reconnaissance vehicle agent(s) and information processing platform agent(s), the statistical simulation results in No.1 tactical reconnaissance stage and No.2 tactical reconnaissance stage are respectively illustrated in Figure 14(a) and Figure 14(b). We can see that the statistical simulation results and the computational results are consistent.



(a)

(b)

Fig.14. Contrastive results of interaction chain attributes in two stages.

Note that those data in the proposed model reflect a real life in warfare field, since the established simulation demonstration system can identify the main components in complex warfare system organization and discover their local interactions and behaviors.

We carry through Verification, Validation, and Accreditation (VV&A) for our simulation model to analyze these results. As far as the concept model, we check attributes description and interactions, e.g., whether the entities and their tasks are consistent with real force situation. As far as the program model, emphases are put in data to verify their correctness, dependability and performance.

By the evaluation, these results that we obtained from TRS simulation are accordant to real warfare situation. The fact proves that our models are reasonable and the approach is feasible and effective.

5. Conclusion

Analyzing the configuration and overall efficiency of interaction in a complex military multi-agent system, especially like tactical reconnaissance agents system, is a problem needing to be studied by researchers interested in this area. Markov chain method is a useful tool to analyze a discrete-state system. In this paper, based on qualitative description, according to military operational experience and the system framework we transformed tactical reconnaissance agents system organization into an interaction chain and thus obtained a corresponding interaction task request-service mechanism queuing system, and developed a Markov chain model on the task interaction process, since the state transition probability of the next state depends only on current state. The feasibility and efficiency of our model are proved by agent-based simulation demonstration results. The fact shows that the analyzing technique we proposed has the capability of optimizing the teamwork problems of practical complex warfare system organization.

In contrast with other current research results, our analyzing technique has some advantages: (1) presenting concept models of tactical reconnaissance agents system and conveniently implementing its subsystem-level representation by establishing a task-(role)-entity mapping mechanism; (2) transforming the system into an interaction task request-service mechanism queuing system, establishing a Markov chain model, and finding the inherent relationship of TRS entities and obtaining the optimized tactical reconnaissance agents system organization configuration by solving the state transition equations; (3) implementing performance demonstration for the model and validating the above results by the agent-based TRS simulation. Thus, it constructs a bridge from qualitative description to quantitative theoretical calculation and simulation demonstration, which are three related critical success factors in optimization of tactical reconnaissance agents system organization.

From a scientific point of view, warfare entities make up a dynamic, non-linear, complex adaptive system in which the overall system behavior emerges from the aggregate interaction. Moreover, our research is still at an early stage, as there are only a few entities in model and it needs to be studied furthermore to be more practical. These result in extreme difficulty on problems test. The number of problems test may be increased so that statistical tests can be performed. The alternatives can be using problems in similar studies, test problems on web sites and generating random problems with different sizes. Future work may include more systematic validation of the model, more data collection and the application of our agent-based TRS simulation to the study of organization behaviors issues.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 70901075), Military Science Projects for Graduate Supervisors (Grant No. 2010JY0636-366, No. 2011JY001-015) and Innovation Foundation of AAFE (Grant No. 2011CJ057).

References

1. Zhongzhi Shi, *Intelligent Agents and Their Applications* (Science Press, Beijing, 2001).
2. Zhongzhi Shi and Nanning Zheng, Progress and challenge of artificial intelligence, *Journal of Computer Science and Technology*, **21**(5) (2006) 810-822.
3. Victor Lesser, *Autonomous Agents and Multi-agent Systems* (Kluwer, Boston, 1998).
4. Anita Raja and Victor Lesser, A Framework for meta-level control in multi-agent systems, *Autonomous Agents and Multi-agents Systems*, **15**(2) (2007) 147-196.
5. R. Kota, N. Gibbins, N. R. Jennings, Self-organizing agent organizations, in *Pro. Int. Conf. Autonomous Agents and Multi-agent Systems* (2009), pp. 797-804.
6. Lucian Busoniu, Robert Babuska, Bart De Schutter, A comprehensive survey of multiagent reinforcement learning, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, **38**(2) (2008) 156-172.
7. David M., Olivier S., Abderraa K., Simulation and evaluation of urban bus-networks using a multiagent approach, *Simulation Modelling Practice and Theory*, **15**(6) (2007) 659-671.
8. L. Yu, S. Wang, K. K. Lai, An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support, *European Journal of Operational Research*, **195**(3) (2009) 942-959.
9. A. Zunino, M.Campo, Chronos, A Multi-agent system for distributed automatic meeting scheduling, *Expert Systems With Applications*, **36**(3) (2009) 7011-7018.
10. B. Chen, H. H. Cheng , J. Palen, Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems, *Transportation Research Part C*, **17**(1) (2009) 1-10.
11. Xiong Li and Zhiming Dong, Platform-level distributed warfare model-based on multi-agent system framework, *Defence Science Journal*, **62**(3) (2012) 180-186.
12. Guoyin Jiang, Bin Hu, Youtian Wang, Agent-based simulation of competitive and collaborative mechanisms for mobile service chains, *Information Sciences*, **180**(2) (2010) 225-240.
13. Arthur Huang and David Levinson, Why retailers cluster: an agent model of location choice on supply chains, *Environment and Planning B: Planning and Design*, **38**(2011) 82-94.
14. Bikramjit B., Ahmed A., Landon K., Layered intelligence for agent-based crowd simulation, *Simulation*, **85**(10) (2009) 621-633.
15. Lance E. Champagne and Raymond R. Hill, A simulation validation method based on bootstrapping applied to an agent-based simulation of the bay of Biscay historical scenario, *Journal of Defense Modeling and Simulation*, **6**(10) (2009) 201-212.
16. Ibrahim Cil and Murat Mala, A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare, *Expert Systems with Applications*, **37**(2) (2010) 1331-1343.
17. Xiong Li, Gaotian Pan, Zhiming Dong, Dianbo Cui and Hongwei An, Designing of multi-agent-based of complex warfare system simulation model, *Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis*, **13**(S3) (2006) 953-959.
18. Farid Mokhati, Mourad Badri, Linda Badri, A formal framework supporting the specification of the interactions between agents, *Informatica*, **31**(2007) 337-350.
19. Krzysztof Juszczyszyn and Grzegorz Kolaczek, Attack pattern analysis framework for a multiagent intrusion detection system, *International Journal of Computational Intelligence Systems*, **1**(3) (2008) 215-224.
20. Guoyin Jiang, Bin Hu, and Youtian Wang, Agent-based simulation approach to understanding the interaction between employee behavior and dynamic tasks, *Simulation*, **87**(5) (2011) 407-422.

21. Menggao Dong, Xinjun Mao, Zhiming Chang, Ji Wang, Zhichang Qi, Running mechanism and strategy description language SADL for self-adaptive MAS, *Journal of Software*, **22**(4) (2011) 609-624.
22. Ralph Depke, Reiko Heckel, Jochen M. K., Roles in agent-oriented modeling, *International Journal of Software Engineering and Knowledge Engineering*, **11**(3) (2001) 281-302.
23. Gyoo Gun Lim, Kun Chang Lee, Won Jun Seo and Dae Chul Lee, Multi-agent based simulation for evaluation of mobile business models, *Information–An International Journal*, 14 (2011), 3063-3080.
24. Mariusz Jacyno, Self-organizing agent communities for autonomic computing, Ph. D. thesis, (Department of Electronics and Computer Science, University of Southampton, 2010).
25. Limin Chen and Zhongzhi Shi, A behavior strategy for agents in the semantic web using dynamic description logics, *Information–An International Journal*, 14 (2011), 993-998.
26. V. K. Murthy and E. V. Krishnamurthy, Collaborating agents in distributed networks and emergence of collective knowledge, *Lecture Notes in Computer Science*, **3213**(2004) 95-102.
27. Xiong Li, Kai Wang, Xianggang Liu and Jiuting Duo, Analysis on negotiation in platform-level armored force combat entity agents system, *Lecture Notes in Artificial Intelligence*, **4114**(2006) 346-354.
28. Haibin Zhu and Mengchou Zhou, Role-based collaboration and its kernel mechanisms, *IEEE Transactions on Systems, Man and Cybernetics-Part C*, **4**(36) (2006) 578-588.
29. Ilachinski, A., *Land Warfare and Complexity, Part II: An Assessment of the Applicability of Nonlinear Dynamics and Complex Systems Theory to the Study of Land Warfare* (Center for Naval Analyses, Virginia, 1996).
30. L. Eronen, F. Geerts, and H. Toivonen, A Markov chain approach to reconstruction of long haplotypes, in: *Pro. Pacific Symposium on Biocomputing* (2004), pp. 126-139.
31. Wilson, D. B., Mixing times of Lozenge tiling and card shuffling Markov chains, *Ann. Appl. Probab*. **14**(1) (2004) 274-325.
32. Seneta, E., *Non-negative Matrices and Markov Chains, Springer Series in Statistics* (Springer, New York, 2006).
33. Haijing Hou, Lisheng Jin, Qingning Niu, Yuqin Sun, Meng Lu, Driver intention recognition method using continuous hidden Markov model, *International Journal of Computational Intelligence Systems*, **4**(3) (2011) 386-393.
34. D. Gamerman, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference* (CRC Press, Boca Raton, FL, 1997).
35. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (eds.), *Markov Chain Monte Carlo in Practice* (Chapman & Hall, Boca Raton, FL, 1996).
36. W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains* (Princeton University Press, Princeton, NJ, 1995).
37. M. T. Jacobson and P. Matthews, Generating uniformly distributed random Latin squares, *Journal of Combinatorial Design*, **4** (1996) 405-437.
38. Pemantle, R., A survey of random processes with reinforcement, *Probab*. *Surv*. **4** (2007) 1-79.
39. O. Kallenberg, *Foundations of Modern Probability* (Springer-Verlag, New York, 1997).