

# Task Coordination Organization Model and the Task Allocation Algorithm for Resource Contention of the Syncretic System

Danfeng Wu, Guangping Zeng\*, Di He, Zhaopeng Qian, and Qingchuan Zhang

**Abstract:** At present, robot embedded systems have some common problems such as closure and poor dynamic evolution. Aiming at resolving this situation, our paper focuses on improvements to the robot embedded system and sets up a new robot system architecture, and we propose a syncretic mechanism of a robot and SoftMan (SM). In the syncretic system, the structural organization of the SoftMan group and its modes are particularly important in establishing the task coordination mechanism. This paper, therefore, proposes a coordination organization model based on the SoftMan group, and studies in detail the process of task allocation for resource contention, which facilitates a rational allocation of system resources. During our research, we introduced Resource Requirement Length Algorithm (RRLA) to calculate the resource requirements of the task and a resource conformity degree allocation algorithm of Resource Conformity Degree Algorithm (RCDA) for resource contention. Finally, a comparative evaluation of RCDA with five other frequently used task allocation algorithms shows that RCDA has higher success and accuracy rates with good stability and reliability.

**Key words:** SoftMan; robot; syncretic system; organization model; task allocation; game theory

## 1 Introduction

In recent years, research on SoftMan<sup>[1-3]</sup> has developed rapidly in the soft computation and smart fields. SoftMan is a virtual robot in a network with anthropomorphic intelligence and emotion; it can imitate the functions and behavior of humans.

- Danfeng Wu is with the School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing 100083, China; School of Software, Liaoning Technical University, Huludao 125105, China; and Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China. E-mail: wudanfengby@163.com.
- Guangping Zeng, Zhaopeng Qian, and Qingchuan Zhang are with the School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing 100083, China, and Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China.
- Di He is with the Graduate Center, City University of New York, New York, NY 10016, USA.

\* To whom correspondence should be addressed.

Manuscript received: 2015-09-29; accepted: 2015-11-17

SoftMan also has anthropomorphic autonomy and the initiative to make decisions and produce behavioral responses according to the changes in task requirements and the environment, so as to actively provide a personalized service for users. From the perspective of theory structure and implementation, SoftMan is the sublimation of Agent and Object in the field of software; each SoftMan possesses different resources and capabilities. However, as the systems communication cost and resources are limited, providing solutions to problems is beyond the capacity of a single SoftMan. Therefore, a group of SoftMans is needed to solve problems through interactive communication and mutual cooperation.

This paper studies the task coordination problem in a syncretic system of a robot and SoftMan. In recent years, system task coordination problems have provided a research focus for domestic and foreign scholars. Karagiannis et al.<sup>[4]</sup> proposed a novel method for scheduling and allocating atomic and complex tasks in large-scale networks consisting of homogeneous or heterogeneous cooperative agents, where the task

allocation and the scheduling of complex tasks are accomplished by combining dynamic reorganization of agent groups and the allocation method of constraint optimization. Wang and Jiang<sup>[5]</sup> proposed a novel community-aware task allocation model for Social Networking Multi-Agent Systems (SN-MASs). Sarker et al.<sup>[6]</sup> presented a study on self-organized multi-robot task allocation, and the framework employed is a generic model of self-organized division of labor derived from observations of ant, human, and robot social systems. Capdepuy et al.<sup>[7]</sup> proposed an adaptive algorithm based on the informational theory that members of an organization can be adapted by their communications. However, these traditional task coordination models do not address the system framework, which can make the coordination process easy to implement, and the models do not fit the syncretic system of a robot and SoftMan.

Some research has been carried out on cooperation algorithms for task allocation. For example, Arsenyan et al.<sup>[8]</sup> proposed a method evaluating cooperation parameters based on the Nash Balance theory. Hermoso et al.<sup>[9]</sup> proposed a trust-based role coordination algorithm in task-oriented multi-agent systems. Hunt et al.<sup>[10]</sup> proposed a consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements.

In addition, under the limitation of computing resources, some related works have tried to solve the resource contention problem of task allocation. For example, Hyun et al.<sup>[11]</sup> proposed a decentralized Multi-Robot Task Allocation (MRTA) approach, taking into account the residual expendable resources of robots and their limited communication range. Lattuada and Ferrandi<sup>[12]</sup> modeled the resolution of resources contention in synchronous data flow graphs by adding a function member and communication channel. de Blanche and Lundqvist<sup>[13]</sup> proposed addressing characterization methods for memory contention aware co-scheduling. To calculate the computing complexity and performance of the system and solve the problem of autonomic resource contention-aware scheduling, a variable level framework based on resource management stack was proposed by Sheikhalishahi et al.<sup>[14]</sup> A set of tools to detect resource sensitivity and predict the performance improvements that can be achieved by resource-aware scheduling is provided by Tillenius et al.<sup>[15]</sup> Fink and Homberger<sup>[16]</sup> proposed an ant-based coordination mechanism for

resource-constrained project scheduling with multiple agents and cash flow objectives. However, these coordination models and task allocation algorithms seldom address the problem of reducing the waste of idle resources.

Therefore, in this paper, from the perspective of resource contention, we present a study on the task coordination mechanism, based on the SoftMan Group, in a syncretic system of robot and SoftMan. Our research is as follows:

(1) A cooperative organization model of the SoftMan Group is proposed, and the process of task coordination and allocation based on the cooperative organization model of the SoftMan Group, is thoroughly studied.

(2) A Resource Conformity Degree Allocation algorithm (RCDA) for multi-task contention of resources as well as a Resource Requirement Length Algorithm (RRLA) for the task are introduced. After calculating the proximity degree between the resource requirement of the task and the resources of the system, we list the priorities of tasks needed to solve the problem of resource contention of multiple parallel tasks, and reduce the waste of idle resources. The above points highlight the differences compared with former research. The rest of this paper is organized as follows. Section 2 summarizes the syncretic system of the robot and SoftMan. Section 3 presents the cooperative organization model of the SoftMan group under the syncretic system. Section 4 studies the mechanism and algorithm for the dynamic coordinative allocation of tasks. Section 5 gives the simulation environment, verifying the validity of our method. Finally, Section 6 concludes our work.

## 2 Summary of SoftMan and the SoftMan System

In 2003, the SoftMan concept was put forward from the viewpoint of generalized artificial life<sup>[2]</sup>.

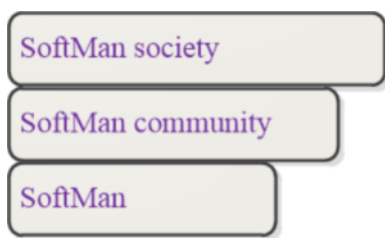
**Definition 1 SoftMan (SM).** The concept model of SoftMan can be expressed as a four-tuple,  $SM = (A, F, D, S)$ . Of these:  $A$  represents the personification properties of SoftMan, including autonomy, adaptability, sociality, etc.;  $F$  represents the personification functions of SoftMan, including learning, organization, working functions, etc.;  $D$  represents the personification behavior of SoftMan, including evolution, personification breeding, personification activities, etc.; and  $S$  represents

the personification structure of SoftMan, including humanoid-brain, humanoid-eye, humanoid-hand, etc.<sup>[1]</sup>

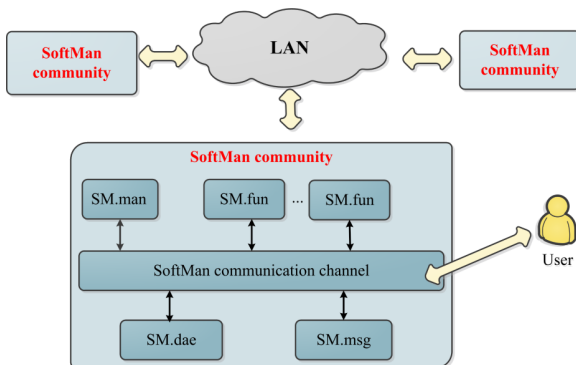
**Definition 2 SoftMan System (SMS).** The SoftMan system is a hierarchical, multi-level, harmonious, open, and loosely-coupled distributed system, composed of the SoftMan (the basic particle element), the SoftMan community (the medium particle element), and the SoftMan society (the large particle element)<sup>[1]</sup>. The hierarchical organization model of the SoftMan system is shown in Fig. 1 and its architecture is shown in Fig. 2.

**Definition 3 SoftMan community.** The SoftMan community is located in the computer node and is composed of SoftMan individuals. These SoftMan individuals consist of SoftMan for daemon (SM.dae), SoftMan for management (SM.man), SoftMan for messages (SM.msg), and SoftMan for executing functions (SM.fun). Next, we introduce the specific role of each SoftMan individual in the SoftMan community.

- **SM.man** is the manager of the SoftMan community and can stand for a node of the SoftMan community. It is responsible for the establishment and intenance of organization relationships, task allocation, community activity decisions, multi-SM coordination, and the transaction coordination among communities<sup>[1]</sup>.



**Fig. 1 Hierarchical organization model of the SoftMan system.**



**Fig. 2 Architecture of the SoftMan system.**

- **SM.fun** is a professional employee that accomplishes certain tasks. It is the carrier of the task logic and is also the final executor of the user’s request. This kind of SoftMan can “wander” in the network to reach the designated work destination (host). In the SoftMan community node, SM.fun generally refers to a function program<sup>[1]</sup>.
- **SM.msg** is not only the messenger in the SoftMan community, but also the founder of the communication channels inside the community. Through SM.msg, different SoftMans within a community or in different communities can communicate with each other<sup>[1]</sup>.
- **SM.dae** is the guardian of the SoftMan community. It makes the host node SoftManized and transforms the system by taking a thread, as the basic unit, into a SoftMan community that uses SoftMan for resource allocation and management. It is responsible for the creation of the SoftMan, fault tolerance in the community node, deployment of the system, and initialization of the community<sup>[1]</sup>.

### 3 Syncretic System and Its Task Coordination Organization Model Based on the Softman Group

The syncretic system of robot and SoftMan contains two aspects. On the one hand, there is the fusion of the SoftMan and robot control system, i.e., different SoftMans can attach to the robot control system as the robot control center to make it perform different functions. On the other hand, the unity of the SoftMan system and the robot system, i.e., after being attached to the SoftMan, the robots and the SoftMans in the SoftMan system can work together to complete the task.

#### 3.1 Syncretic system of robot and SoftMan

##### 3.1.1 Fusion of the SoftMan and the robot control system

At present, the robot widely adopts a two-level distributed open control system. This system structure obviously improves the operating speed and the control performance of the controller, but there are weaknesses such as the heavy computation burden and the poor real-time performance. When the robot is disturbed during running, it can not adjust in real time, and the task completion is affected. At the same time, though the upper-lower machine position method improves the

extendibility and reusability of the robot control system, it can not complete the dynamic configuration and on-line reconstruction of the robot function.

In order to solve this problem, we introduce the SM.host and the SM.app into the robot control system. By embedding them, the robot can be transformed into a sense carrier and a terminal actuator platform, and can implement various functions. A schematic of SoftMan and the robot control system fusion is shown in Fig. 3.

- **SoftMan for host (SM.host)** is the management and daemon center of the robot system and the interactive hub between the robot system and the SoftMan system. Meanwhile it is responsible for receiving SM.app and building its supporting environment. Specific duties include robot system initialization, message communication, SM.app reception and control, system fault tolerance control, and environment resource monitoring.
- **SoftMan for appendage (SM.app)** is essentially a special kind of SM.fun which can be migrated and it is the function control center of the robot system. By online migration and dynamic replacement of SM.app, the robot can realize real-time update functions.

### 3.1.2 Unity of the SoftMan system and the robot system

As the SM.host and SM.app are introduced into the robot control system, a peer-to-peer, flexible, dynamic synergy pattern is constructed between the SoftMan system in the PC and the robot system. The SoftMan system can transfer different SM.apps to the robot control system according to the task requirement; at the same time, the SoftMans in the SoftMan system can cooperate with the SoftMans in the robot system

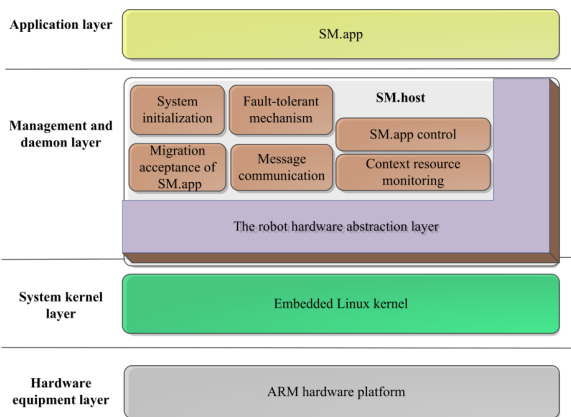
to complete the task. A diagram of the unity of the SoftMan and robot systems is shown in Fig. 4, which presents a new working scenario.

Figure 4 presents a new working scenario:

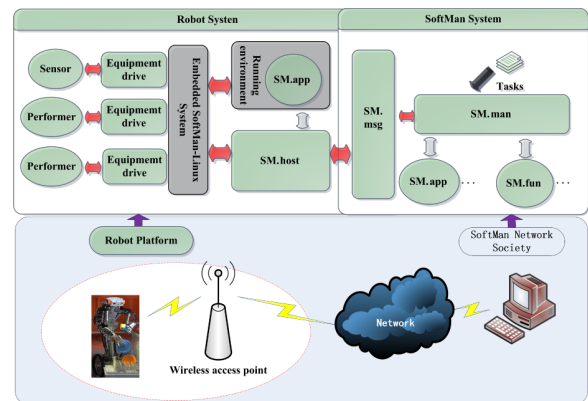
- (1) SM.host in the robot system loads and starts to complete the initialization of the robot system;
- (2) SM.host registers with the SM.man in the SoftMan community and provides the robot function information;
- (3) Users put forward the task requests to the SoftMan community which queries the relevant information, matches the robot which can complete the task, and sends the corresponding SM.app to execute the task;
- (4) The SM.host receives the SM.app and constructs a running environment for it, then executes the task;
- (5) After the completion of the task, SM.host reports the executing results to the SM.man, and after the confirmation, SM.host can choose to temporarily reserve the SM.app or delete it; if the robot needs to continue to perform the task, go to (3), otherwise go to (6);
- (6) SM.host enters an idle state.

It should be noted that, because of the limited computing resources of the robot with SM.host, we trust the decision-making of the SM.host in SM.man in the SoftMan community, thus, the decision-maker of the syncretic system is SM.man.

By unifying the SoftMan and robot systems, the robot system can receive different SM.apps, migrated from the SoftMan system, to realize the real-time replacement function. The fast intelligent decision and the large capacity storage characteristics of the SoftMan system in a PC can free the robot from the high



**Fig. 3** Fusion schematic diagram of SoftMan and the robot control system.



**Fig. 4** The schematic diagram of the unity of the SoftMan system and robot system.

requirements of real-time processing control, storage space, and so on. It can also dynamically expand the software system of the robot to adapt to dynamic task changes, so that the robot system's robustness and flexibility are enhanced, and its utility value is further improved.

### 3.2 Task coordination organization model based on the SoftMan group of the syncretic system

#### 3.2.1 Task coordination organization model design

We designed the task coordination organization model of the SoftMan group on the basis of hierarchical and layered concepts for generalized large systems<sup>[17, 18]</sup>. The model easily realizes the hierarchical and centralized management of the SoftMan group and highlights features such as active interaction, free migration, mutual communication, and cooperation among SoftMans. The task coordination organization model of the syncretic system is shown in Fig. 5.

**Definition 4 Syncretic System Social Network (SS-SN) and syncretic system community (ssc).** The syncretic system social network consists of several syncretic system communities and is defined as  $SS-SN = \{ssc_1, ssc_2, \dots, ssc_m\}, m \geq 1$ ; the syncretic system community consists of a SoftMan community in the computer node and a number of robots that have SM.man as the decision-maker.

**Definition 5 SoftMan for social coordination (SM.coor).** SM.coor is responsible for collecting the resource data and running status of the whole society and the SM.man of each community is obliged to report related data to it. It is the coordination liaison of SS-SN, but has no absolute right of control over communities<sup>[1]</sup>.

**Definition 6 Management SoftMan category and non-management SoftMan category.** In the syncretic system, according to the task function and management function, SoftMan is customized into five types: SM.fun, SM.app, SM.man, SM.host, and SM.coor. Of these, SM.fun and SM.app belong to the non-management SoftMan category, while the others

belong to the management SoftMan category.

**Definition 7 App-H.** As the management and guardian center of the robot, each SM.host can represent a robot. When SM.app and SM.host have a migrate and receive relationship, the robot with SM.host becomes the final executant of the user request after receiving the SM.app. Therefore, the App-H that represents the fusion of SM.app and SM.host can be regarded as a special SM.fun with a specific function.

#### 3.2.2 Task coordination and allocation process based on the task coordination organization model

Contract net is an important kind of solution model for collaborative problem, and is widely used in the coordination of multi-agent systems. Therefore, our paper describes the process of the task coordination and distribution using the negotiation process of the contract net protocol, based on the structure of task coordination organization model SS-SN:

- (1) User generates the task requirement and takes the nearest local ssc as the request entrance.
- (2) SM.fun, which is responsible for the task decomposition, decomposes the task request according to the simplified social function information table provided by SM.coor. After decomposition, each subtask corresponds to a class of SM.fun or App-H with the same function.
- (3) SM.man filters out the local community's SM.apps or App-Hs which have the function of completing the corresponding task, and acquires the related information to make the task allocation decision within the community.
- (4) If the local community can not accomplish a certain task, SM.man sends the bidding document to SM.coor.
- (5) SM.coor inquires about the complete social function information table and sends the bidding document to the communities that have the demanded task function.
- (6) The SM.man of the community that has the cooperation intention, provides the original bid to SM.coor which returns all the original bids to the bidding community before the deadline for submission of tenders; if no bid community, turn to (9).
- (7) According to the task allocation decision model of the community, the SM.man of the bidding community determines and informs the winning community. After the winning community's

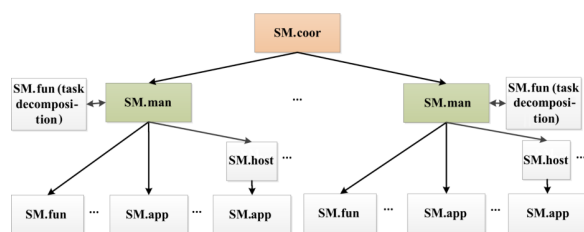


Fig. 5 Task coordination organization model of the syncretic system.

confirmation, the SM.man of the bidding community completes the process of task allocation. If the winning community does not make a confirmation in time or revokes the original bid, the SM.man of the bidding community will choose a suboptimal community and issue a notice of winning, and so on; if none of them make a confirmation, turn to (9).

- (8) After confirmation, the winning community carries out (3), and returns the task execution information regularly to the bidding community.
- (9) When task allocation fails, the task entrance community returns the failure information to the user.

From the above, we find that the task coordination of SS-SN puts the local community as a priority and finally plays the role of global coordination to avoid large consumption of communication resources.

### 3.2.3 Comparison of our model structure and other task allocation structure

#### 3.2.3.1 Comparison of the system task managers

In recent years, much research<sup>[19–23]</sup> related to the task allocation of the multi-agent system has widely used the structure of the manager/contractor or principal/assistant, that is, an agent is assigned as the manager of tasks. If there are not enough resources for the tasks needed, it will negotiate with other agents for contractors. In this structure, the manager needs to perform tasks in addition to searching for contractors and making the task allocation decision, so its reliability is low. If the manager fails during execution, the task fails.

In the task coordination model of SS-SN, SoftMan is divided into two categories: management class and non-management class. These are intended to liberate the SM.man of each community from decomposition and execution of the task and make it responsible for the task allocation decision, system resources management, and task coordination within the community or inter-communities. This can improve the success rate of task allocation and execution, and when SM.man fails, the tasks already assigned can continue to perform, thus improving the reliability of system.

#### 3.2.3.2 Comparison between the centralized controller in the centralized task allocation structure and SM.coor in our model

In Refs. [24–26], a centralized control structure is generally adopted. In their systems, there is a special

controller to observe and control global resources. Although the process is simple and it is easy to get a global optimal solution, the communication concentration easily causes network congestion and the computation complexity makes it difficult to meet the real-time requirements. When the centralized controller fails, the task allocation and execution of the entire system is affected. That is, the centralized system structure brings the problems of fault concentration, risk concentration, communication concentration, and poor real-time capability.

SM.coor in our organization model structure is not a centralized controller and does not receive tasks, it just coordinates, so we can say that it also indirectly plays a role in centralized control. If it fails, each community node can run normally, but in this case, the SS-SN turns into a pure decentralized structure and the horizontal communication inter-community becomes intensive.

#### 3.2.3.3 Comparison between hierarchical and decentralized coordinations

In Refs. [20, 27], the decentralized task coordination structure was adopted as shown in Fig. 6. The task allocation in the decentralized coordination system is oriented to the individual agent; if the task manager does not have sufficient resources for the task, it will use the distributed method to seek a contractor in the local community or other communities associated with it. This decentralized coordination is largely influenced by whether there is a relevant inter-community, however, it is easy to fall into the local optimum and the quality of communication will greatly affect the effect of the task distribution.

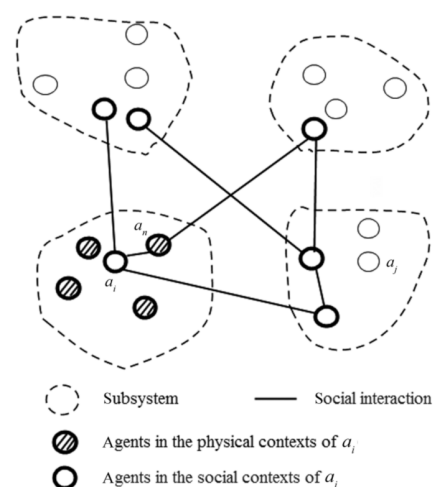


Fig. 6 Decentralized task coordination structure.

The task coordination of our organization model is hierarchical. The task entrance is the community. The SM.man of the community allocates the task to the non-management SoftMan of the local community at first. If the local community cannot complete the task, SM.man, on behalf of its community, seeks help from SM.coor which then asks the SM.mans of the other communities, so there is a hierarchical coordination. In addition, the allocation of tasks is no longer limited to whether there is a relationship between communities, so the success rate of task allocation increases.

The horizontal flow of information among different communities is dense and complex, see Fig. 1, and the decentralized control structure makes the global state of large systems unobservable. Therefore, decentralized coordination is suitable for occasions when the demand for the coordination of large systems is not very high or mutual communication is more convenient. In the syncretic system, SM.coor can quickly obtain information on the change in the community topology structure and can indirectly observe the global state of a large system in the structure. When a community needs help, it can quickly find out the communities where the task-required non-management SoftMans is located, and sends help information accordingly. This reduces the number of inter-community communications and the communication density. Hierarchical longitudinal flow through SM.coor–SM.man–SM.fun or SM.app also reduces the density of transverse information flow.

In addition, the syncretic system also has the following characteristics:

- (1) The syncretic system makes task coordination between the Agent and the robot transform into coordination between SM.fun and SM.app, resulting in a seamless combination of hardware and software.
- (2) All kinds of SoftMan performs their duties, enhancing the robustness of the system.
- (3) The model structure of SS-SN is more similar to the real social structure. SM.coor is only a coordination liaison and does not have absolute right of control. Each community under the centralized control of SM.man has full autonomy. The task allocation of SS-SN chooses the local community first, which avoids large consumption of communication resources and achieves remote cooperation, expanding the physical space.

In short, the SS-SN task coordination model has direct and timely local control of SM.mans and centralized, indirect, and overall coordination of SM.coor, so it has the advantage of centralized and decentralized coordination. The social structure and hierarchical relationship also make task coordination relatively simple and the amount of information for coordination control and observation is smaller.

#### **4 Mechanism and Algorithm of Dynamically Coordinative Task Allocation for Resource Contention**

In the syncretic system, the task coordination of the SoftMan group specifies the distribution of dynamic tasks. The multi-task flow brings many new challenges, mainly involving the influence of the current task on the distribution of a future task and the cross effect of the distribution between two task flows. In addition, in the distributed open environment of the syncretic system, the conflict arising from obtaining the task load status, the structure of the network topology, and the traffic of the entire system become more acute. Therefore, task allocation requires a reasonable distribution of system resources in the final analysis, namely, uses an appropriate distribution algorithm to lower the occupancy rate of the system resources, reduce the total execution time, lower the cost of the communication, reduce the conflict probability when the underlying non-management SoftMan group executes tasks, and improve the system's self-adaption to the dynamic environment.

In the syncretic system, the coordination organization model is designed from the hierarchical and layered concept of large system cybernetics, therefore, in the whole system, we adopted a layered design concept to pursue the optimal distribution of resources. Namely, the physical allocation of tasks corresponds to the bottom SM.fun and APP-H, the policymakers are SM.man and the logical allocation corresponds to the middle SM.man. Because of the limited computing resources, the SM.host transfers its decision-making operation to the SM.man on the network platform. This SM.man then uses the SM.host to extend and complement its ability for mobile applications. Therefore, SM.man and SM.host have a managed and symbiotic relationship. We need to point out that the fulfillment of each task corresponds to SM.fun and APP-H and the resources required by the task are the

system resources required by SM.fun and APP-H that perform the task. The goal of task allocation in the syncretic system is optimal resource allocation for the whole system.

During the actual running of the system, a shortage of resources causes contention among multi-tasks. To describe the resource contention behavior quantitatively, we introduce a multi-player game mode, namely, the auction model. To facilitate construction of this model we make the following assumptions.

**Assumption 1** Task set  $T$  is a game set, whose members compete for the system resources,  $T = \{t_1, t_2, \dots, t_l\}$ . The user-defined priorities of the tasks involved in the game set are the same, and all the tasks are in parallel.

**Assumption 2** The execution of each task  $T$  corresponds to an SM.fun and APP-H, the tasks compete for system resources and the system is where the corresponding SM.fun and APP-H belong.

**Assumption 3** The resource conflicts in the same class cannot be occupied by different tasks at the same time.

Now, assume that  $R$  represents the collection of system resources,  $m$  represents the number of system resource types,  $R^r$  represents the system resource type  $r$ ,  $R_T^r$  denotes the resource requirement of  $T$  for resource  $r$  and  $R_i^r$  denotes the resource requirement of  $t_i$  for resource  $r$ .

Actually, the resource requirement of the task is discrete and to calculate it, this paper applies a linearization method to the nonlinear resource requirement, which we call the RRLA shown in Eqs. (1) and (2).

The calculation equation of  $R_i^r$  is

$$R_i^r = \text{select}_{j=1}^{n_r} \left( \int_{x_0}^{x_0+\delta} (a_r^j \cdot x + b_r^j) dx \right) \quad (1)$$

where “select” means that the resource labeled  $r$  is divided into  $n_r$  segments and the task requests resource  $r$  from a segment as  $j$ ;  $(a_r^j \cdot x + b_r^j) dx$  is the approximate value of the request resources in the segment  $j$ ,  $a_r^j$  is the slope of the linear approximate function,  $b_r^j$  is the intercept, and  $x$  is the request time zone for  $r$ , whose initial value is  $x_0$ .

It should be emphasized that the number of intervals of resource  $r$  means the partition granularity of  $r$ , and the real-time state change of the system resources is the basis of the number of intervals. This forms in an adaptive way to realize a dynamical real-time

adjustment of the number of intervals. If the resources requested by the task are much less than the existing resources of the system, the number of the intervals will increase self-adaptively, otherwise they decrease.

Now, we can get  $R_T^r$ , as shown in Eq. (2)

$$R_T^r = \sum_{i=1}^l R_i^r \quad (2)$$

Game behavior is generated under the condition of different tasks contention for the same resource. If each type of system resource  $R^r$  fully satisfies  $R_T^r$ , some resources  $R^k$  do not satisfy any  $R_i^k$ , the game behavior will not occur. If some resources  $R^k$  do not satisfy  $R_T^r$ , and each type of system resource  $R^r$  fully satisfies any  $R_i^k$ , the game behavior will occur. Therefore, this paper uses a condition controlled function to control whether group game behavior happens or not, as shown in Eq. (3).

$$C(f) = \begin{cases} 0, & R^r \geq R_T^r \text{ or } R^k < R_i^k; \\ 1, & R^k < R_T^k \text{ and } R^r \geq R_i^r \end{cases} \quad (3)$$

where  $r = 1, 2, \dots, m; k = 1, 2, \dots, m; i = 1, 2, \dots, l$ .  $m$  represents the type number of system resources and  $l$  represents the task number of the game set  $T$ . As the control function of game behavior,  $C(f)$  is defined as 1 or 0, where 0 represents no game, and 1 represents a game occurrence.

Now, we can get the degree of conformity of the resources required by each task and the existing resources of the system. This is shown in Eq. (4).

$$G(i) = \begin{cases} \sqrt{\sum_{i=1}^m (R^r - R_i^r)}, & C(f) = 1; \\ \infty, & C(f) = 0 \end{cases} \quad (4)$$

**Theorem 1** In the process of the game, the more the task requirement for resources conforms to the existing resources of the system, the higher priority the task has.

**Proof:** The more the task requirement for resources conforms to the existing resources of the system, the closer is the requirement to existing resources, resulting in less wastage. According to the Pareto principle of resource allocation, we validate that the higher the degree of coincidence, the better the allocation of resources.

According to Theorem 1, the smaller  $G(i)$ , the higher the degree of resource conformity and the higher resource contention priority. The task that has the highest priority is allocated first. We call this the RCDA.



## 5 Experiments

First, several frequently-used multi-task coordination and allocation methods are selected for the comparative test, which include Rate Monotonic Algorithm (RMA)<sup>[28]</sup>, Earliest Deadline First Algorithm (EDFA)<sup>[29]</sup>, First in first out Algorithm (FiA), Complexity Algorithm (CoA), and Random Algorithm (RaA)<sup>[30]</sup>.

RMA and EDFA are coordinated allocation algorithms based on periodic task models. RMA depends on the length of the task execution cycle to determine the scheduling priority; the task with a small implementation cycle has a high priority. This algorithm satisfies the requirement of general systems, and its implementation is simple. EDFA is a dynamic algorithm, the priority of tasks is dynamically allocated and based on a task deadline; the closer the deadline, the higher the priority. CoA depends on the task complexity to determine the scheduling priority; the task with a high complexity has a high priority. All of the above algorithms were realized with Matlab2010 before integrating into the distributed SoftMan system based on Linux in the form of mixed programming.

### 5.1 Experiment design

We compared the RCDA proposed in this paper with five task coordinated allocation algorithms. In addition, we selected two indicators, success rate and accuracy, to evaluate their validity and stability. The success rate of an algorithm is the probability of it successfully giving priority to each task and successfully allocating tasks when multi-tasks compete for resources at the same time. The accuracy refers to the probability of the task execution results being in accordance with the actual task logic after the task allocation.

We put into operation 50 task sequences in a distributed SoftMan system based on Linux with the task sequence consisting of 300 parallel tasks. These 300 parallel tasks had the same initial priority and each task sequence constituted a game set. The tasks in the task sequence involved the typical tools and algorithms of the different task types. These task types include: data encryption and decryption, data compression, document recognition, network information crawl, video encoding conversion, data sorting processing, tree traversal, graph traversal, image identification, feature extraction, neural networks, clustering analysis, classification analysis, correlation analysis in data

mining, and other algorithms which are common in the SoftMan system. We used the above six types of allocation algorithm for test, recording their operation and calculating their success rate and accuracy.

### 5.2 Results of experiment

The results of the comparative experiment are graphically presented in Figs. 7 and 8.

### 5.3 Data analysis of the results

The cumulative success rate of 50 task sequences of an algorithm divided by 50 is the average success rate. From statistical analyses of the comparative tests, the average success rates of RMA, EDFA, FiA, CoA, RaA, and RCDA are 0.97, 0.94, 0.84, 0.81, 0.83, and 0.96, respectively, with respective standard variances of 0.0155, 0.0215, 0.0374, 0.0291, 0.0464, and 0.0170. The cumulative accuracy of 50 task sequences of an algorithm divided by 50 is the average accuracy. The average accuracies of the algorithms are 0.81, 0.82, 0.68, 0.70, 0.73, and 0.91, respectively, with respective

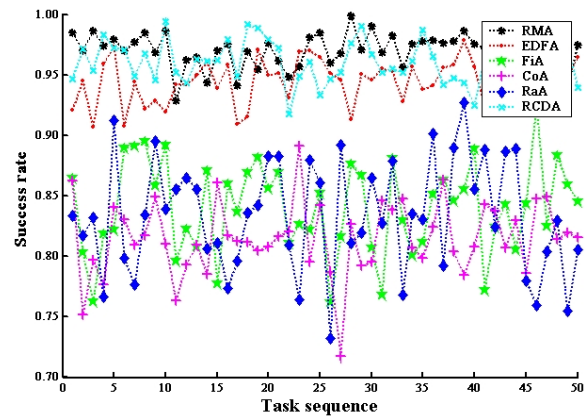


Fig. 7 Success rate of task allocation.

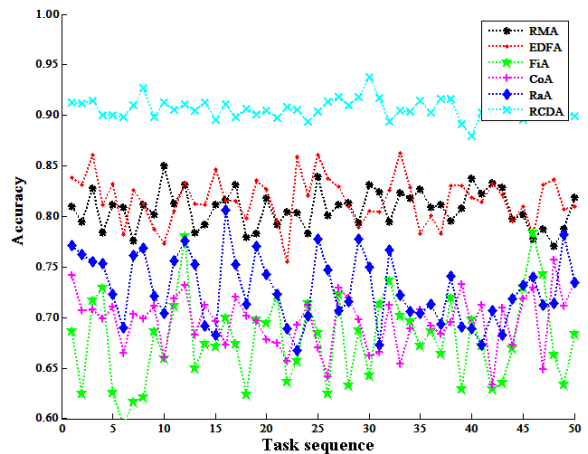


Fig. 8 Accuracy of task execution.

standard variances of 0.0218, 0.0222, 0.0409, 0.0282, 0.0389, and 0.0095, as shown in Table 1.

Next, we present comprehensive evaluation indexes (Table 2) consisting of two values: a comprehensive evaluation value and an average value of the standard variances. The evaluation value is defined as the product of the success rate and accuracy, and the average value of the standard variances is the arithmetic mean value of the standard variances of success rate and accuracy.

The comparative experiment showed that RCDA's success rate in task allocation is 0.96, ranked second after the RMA success rate of 0.97; the accuracy of task allocation for RCDA is 0.91, ranked first, 0.10 higher than the RMA; the standard variance of success rate is 0.0170, the second lowest value; the standard variance of the accuracy is 0.0095, which was the lowest. Finally, we carried out a comprehensive evaluation of these algorithms. RCDA has the highest comprehensive evaluation value at 0.87, 0.08 higher than the next value, and the minimum average value of the standard variances is 0.0133. These values demonstrate that RCDA is more stable and reliable than the other five algorithms.

## 6 Conclusion

In a syncretic system of a robot and SoftMan, SoftMan is customized into six types and divided into two classes: non-management SoftMan and management

SoftMan. The coordination organization model of the SoftMan group is designed according to hierarchical and layered concepts, which benefits the multi-level, hierarchical, and centralized management of the SoftMan group. The coordination process in the organization model prioritizes the local community coordination at first, resulting in global coordination that avoids excessive consumption of communication resources.

The problem of the dynamic coordination allocation of tasks for the entire system is translated into a problem of contention for resources. This paper uses a linearization method, based on a nonlinear function, to calculate the resource requirements of the task, namely, the RRLA, then uses the RCDA to allocate the task. Following a comparative experiment between the RCDA and five other frequently-used task allocation algorithms, we conclude that the RCDA shows the optimal comprehensive evaluation value and the lowest standard variance, indicating that it is a reliable and stable algorithm.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61404069) and the National High-Tech Research and Development (863) Program of China (No. 2015AA015403).

## References

- [1] G. P. Zeng, X. Y. Tu, and H. B. Wang, *SoftMan Research and Applications* (in Chinese), Beijing, China: Science Press, 2007, pp. 10–19.
- [2] G. P. Zeng and X. Y. Tu, The concept model and construction frame of the SoftMan, *Computer Science*, vol. 32, no. 25, pp. 135–143, 2005.
- [3] Q. Liu, G. P. Zeng, W. Zhang, and X. Y. Tu, Research on SoftMan individual model, *Computer Engineering and Applications*, vol. 30, pp. 49–51, 2005.
- [4] P. Karagiannis, G. Vouros, K. Stergiou, and N. Samaras, Overlay networks for task allocation and coordination in large-scale networks of cooperative agents, *Autonomous Agents and MultiAgent Systems*, vol. 24, no. 1, pp. 26–68, 2012.
- [5] W. Y. Wang and Y. C. Jiang, Community-aware task allocation for social networked multiagent systems, *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1529–1543, 2014.
- [6] M. O. F. Sarker, T. S. Dahl, and E. Arcaute, Local interactions over global broadcasts for improved task allocation in self-organized multi-robot systems, *Robotics & Autonomous Systems*, vol. 62, no. 10, pp. 1453–1462, 2014.
- [7] P. Capdepu, D. Polani, and C. L. Nehaniv,

**Table 1 Average success rate, average accuracy, and standard deviation calculation of task allocation under different algorithms.**

Algorithm	Average success rate	Std of success rate	Average accuracy	Std of accuracy
RMA	0.97	0.0155	0.81	0.0218
EDFA	0.94	0.0215	0.82	0.0222
FiA	0.84	0.0374	0.68	0.0409
CoA	0.81	0.0291	0.70	0.0282
RaA	0.83	0.0464	0.73	0.0389
RCDA	0.96	0.0170	0.91	0.0095

**Table 2 Comprehensive evaluation indexes.**

Algorithm	Comprehensive evaluation value	Average value of standard variances
RMA	0.79	0.0187
EDFA	0.77	0.0216
FiA	0.57	0.0391
CoA	0.57	0.0287
RaA	0.61	0.0426
RCDA	0.87	0.0133

- Perception-action loops of multiple agents: Informational aspects and the impact of coordination, *Theory in Biosciences*, vol. 131, no. 3, pp. 149–159, 2012.
- [8] J. Arsenyan, G. Büyükcikan, and O. Feyziolu, Modeling coordination formation with a game theory approach, *Expert Systems with Applications*, vol. 42, no. 4, pp. 2073–2085, 2015.
- [9] R. Hermoso, H. Billhardt, and S. Ossowski, Trust-based role coordination in task-oriented multi-agent systems, *Knowledge-Based Systems*, vol. 52, pp. 78–90, 2013.
- [10] S. Hunt, Q. G. Meng, C. Hinde, and T. W. Huang, A consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements, *Cognitive Computation*, vol. 6, no. 3, pp. 338–350, 2014.
- [11] L. D. Hyun, S. A. Zaheer, and K. J. Hwan, Ad hoc network-based task allocation with resource-aware cost generation for multi-robot systems, *IEEE Transactions on Industrial Electronics*, vol. 61, no.12, pp. 6871–6881, 2014.
- [12] M. Lattuada and F. Ferrandi, Modeling resolution of resources contention in synchronous data flow graphs, *Journal of Signal Processing Systems*, vol. 80, no. 1, 39–47, 2014.
- [13] A. de Blanche and T. Lundqvist, Addressing characterization methods for memory contention aware co-scheduling, *The Journal of Supercomputing*, vol. 71, no. 4, pp. 1451–1483, 2015.
- [14] M. Sheikhalishahi, L. Grandinetti, R. M. Wallace, and J. L. Vazquez-Poletti, Autonomic resource contention-aware scheduling, *Software: Practice and Experience*, vol. 45, no. 2, pp. 161–175, 2015.
- [15] M. Tillenius, E. Larsson, R. M. Badia, and X. Martorell, Resource-aware task scheduling, *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 14, no. 1, p. 5, 2015.
- [16] A. Fink and J. Homberger, An ant-based coordination mechanism for resource-constrained project scheduling with multiple agents and cash flow objectives, *Flexible Services and Manufacturing Journal*, vol. 25, nos. 1&2, pp. 94–121, 2013.
- [17] X. Y. Tu, Z. Wang, and Y. H. Guo, *Large Scale Systems Control Theory*, (in Chinese). Beijing, China: Beijing University of Posts and Telecommunications Press, 2005.
- [18] X. Y. Tu, Generalized model, intelligent simulation, soft-man, *Computer Simulation*, vol. 28, no. 7, pp. 224–228, 2011.
- [19] M. M. D. Weerdt, Y. Zhang, and T. Klos, Multiagent task allocation in social networks, *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 1, pp. 46–86, 2012.
- [20] Y. Jiang and J. Jiang, Contextual resource negotiation-based task allocation and load balancing in complex software systems, *IEEE Transactions on Parallel & Distributed Systems*, vol. 20, no. 5, pp. 641–653, 2008.
- [21] Y. Jiang, Y. Zhou, and W. Wang, Task allocation for undependable multiagent systems in social networks, *IEEE Transactions on Parallel & Distributed Systems*, vol. 24, no. 8, pp. 1671–1681, 2013.
- [22] Y. Jiang and Z. Li, Locality-sensitive task allocation and load balancing in networked multiagent systems: Talent versus centrality, *Journal of Parallel & Distributed Computing*, vol. 71, no. 6, pp. 822–836, 2011.
- [23] Y. Jiang and Z. Huang, The rich get richer: Preferential attachment in the task allocation of cooperative networked multiagent systems with resource caching, *Systems Man & Cybernetics Part A Systems & Humans IEEE Transactions on*, vol. 42, no. 5, pp. 1040–1052, 2012.
- [24] L. Wang, Sofa: An expert-driven, self-organization peer-to-peer semantic communities for network resource management, *Expert Systems with Applications*, vol. 38, no. 1, pp. 94–105, 2011.
- [25] S. Fjuita and V. R. Lesser, Centralized task distribution in the presence of uncertainty and time deadlines, in *Proc. 2nd ICMAS*, Kyoto, Japan, 1996, pp. 87–94.
- [26] H. Kose, U. Tatldede, C. E. MeriCli, K. Kaplan, and H. L. Ak, Q-learning based market-driven multi-agent collaboration in robot soccer, in *Proceedings of the Turkish Symposium on Artificial Intelligence & Neural Networks*, 2004, pp. 219–228.
- [27] W. Wang and Y. Jiang, Community-aware task allocation for social networked multiagent systems, *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1529–1543, 2014.
- [28] M. Park and H. Park, An efficient test method for rate monotonic schedulability, *IEEE Transactions on Computers*, vol. 63, no. 5, pp. 1309–1315, 2014.
- [29] E. Okuyan and B. Kayayurt, Earliest deadline first scheduling algorithm and its use in ANKA UAV, *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st IEEE*, 2012, pp. 8B1-1–8B1-8.
- [30] Y. Biletskiy, G. R. Ranganathan, and O. Vorochek, Identification and resolution of conflicts during ontological integration using rules, *Expert Systems*, vol. 27, no. 2, pp. 75–89, 2010.



**Danfeng Wu** received the BS degree in 2004 and MS degree in 2009 from Liaoning Technical University, China. She is currently working towards her PhD degree at University of Science and Technology Beijing, China. Her current research interests include intelligent robotics and SoftMan technology and

distributed computing.



**Zhaopeng Qian** is currently working towards the MS degree at University of Science and Technology Beijing, China. He received the BS degree in 2012 from Jiangxi Agriculture University, China. His current research interest includes intelligent robotics, SoftMan technology, artificial intelligence & artificial life, and

communication cooperation.



**Guangping Zeng** received the PhD degree from University of Science and Technology Beijing, China, in 1999. He is currently a professor and PhD supervisor of School of Computer and Communication Engineering at University of Science and Technology Beijing, and the Director of the Center of Smart

System and Soft Computing (CSSSC). His research areas include distributed and migrating computing, Linux operating system and embedded systems, intelligent robotics and SoftMan technology, intelligent network and communications, and smart systems and soft computing.



**Qingchuan Zhang** received the BS degree from Hebei University of Science and Technology in 2004 and the MS degree from Hebei University of Science and Technology in 2008. In 2013, he received the PhD degree from University of Science and Technology Beijing, China. He is currently doing postdoctoral research at

University of Science and Technology Beijing, China. His research interests focus on intelligence software and distributed computing.



**Di He** is currently a PhD student at the Graduate Center of the City University of New York. He received the master degree from University of Science and Technology Beijing in 2015. His currently research focuses on machine learning algorithms for analyzing binding promiscuity and specificity using chemical

genomics data, development of multi-instance, multi-target regression algorithms for individualized anti-cancer drug response prediction, and multi-scale modeling of drug-target binding/unbinding kinetics using molecular dynamics simulation and machine learning.