

Multi-Robot Remote Interaction with FS-MAS

Regular Paper

Yunliang Jiang², Xiongtao Zhang², Liang Tang³, Weicong Liu¹, Jing Fan² and Yong Liu^{1,4,*}¹ Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China² School of Information & Engineering, Huzhou Teachers College, Huzhou, China³ China Ship Development and Design Center, Wuhan, China⁴ State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China

* Corresponding author E-mail: yongliu@ipc.zju.edu.cn

Received 21 Jun 2012; Accepted 17 Oct 2012

DOI: 10.5772/54468

© 2013 Jiang et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract The need to reduce bandwidth, improve productivity, autonomy and the scalability in multi-robot teleoperation has been recognized for a long time. In this article we propose a novel finite state machine mobile agent based on the network interaction service model, namely FS-MAS. This model consists of three finite state machines, namely the Finite State Mobile Agent (FS-Agent), which is the basic service module. The Service Content Finite State Machine (Content-FS), using the XML language to define workflow, to describe service content and service computation process. The Mobile Agent computation model Finite State Machine (MACM-FS), used to describe the service implementation. Finally, we apply this service model to the multi-robot system, the initial realization completing complex tasks in the form of multi-robot scheduling. This demonstrates that the robot has greatly improved intelligence, and provides a wide solution space for critical issues such as task division, rational and efficient use of resource and multi-robot collaboration.

Keywords Network Interaction Service Model, FS-MAS, Multi-Robot System, Multi-Robot Collaboration

1. Introduction

Teleoperation has been widely applied in a variety of situations, ranging from space exploration, inspections, robotic navigation, surveillance, underwater operations and rescue activities. It has the promise and advantages of being able to provide replaceable surrogates for humans in hazardous or difficult working environments over long distances, potentially improving productivity and reducing costs. Regardless of whether the remote machine or robot is manually controlled by an operator, or semiautonomous, or even fully autonomous for some specific tasks, human observation, intervention and supervision still play integral roles in these teleoperated systems.^[1]

Multi-robot teleoperation is not a simple cumulative single robot teleoperation, the relationships in multi-robot system are extremely complex, and if not handled properly, it might not only malfunction, but also affect the performance of a single robot^[2,3,4]. In multi-robot systems, the inappropriate distribution or misuse of resources is likely to cause conflict or deadlock. Therefore, it is necessary to study network interaction

service models for multi-robot teleoperation, which ensure the exchange of information between robots and the sharing of resources. A well-designed interactive network service model is necessary for efficient cooperation in multi-robot systems.

Teleoperation systems have found many applications in past decades, however, these systems pose three challenges in the process of development^[5]:

- [1] Lack of a generic framework for multi-robot teleoperation systems
- [2] Low performance in limited bandwidth
- [3] The presence of a time delay in most teleoperation systems makes them unstable and/or demonstrates undesired performance.

Three strategies have been created for these challenges^[4,5]:

1. Evolution of existing systems
2. Adapt largely distributed systems to teleoperation
3. Design dedicated systems.

In order to overcome the current deficiencies, in this paper, we propose a novel finite state machine based on the network interaction service model, namely FS-MAS. This model imports Agent as a solution for resource sharing and use which can automatically search for resources and automatically run the tasks. The introduction of mobile GIS technology supports a mobile environment with mobile robot position detection framework. By introducing the idea of data-driven programming, using the basic tasks we defined before, the robot can combine them to complete more complex tasks, which have a certain independence and reusability. This model consists of three finite state machines, namely Finite State Mobile Agent (FS-Agent), which is the basic service module. Service Content Finite State Machine (Content-FS), we uses XML language to define workflow, to describe service content and service computation process. Mobile Agent computation model Finite State Machine (MACM-FS), used to describe the service implementation.

At the same time, we apply this service model to the multi-robot system, the initial realization completing complex tasks in the form of multi-robot scheduling. This demonstrates that robots have improved in intelligence, providing a wide solution space for critical issues such as task division, rational and efficient use of resource and multi-robot collaboration. This model is proved effectively.

The contributions of this article are:

- a) Importing mobile agent technology in teleoperation system, which provide good performance in limited bandwidth

- b) Importing the idea of data-driven programming which supports complex tasks.
- c) Designing and implementing a genetic framework and platform for multi-robot teleoperation.

The paper is organized as follows: section 2 introduces the related technique used in multi-robot teleoperation and the related interaction service model. The architecture of the multi-robot system is presented in section 3. In section 4 we introduce a novel finite state mobile agent model and its application in the multi-robot remote interaction system. Section 5 concludes and outlines the future directions of our work.

2. Related work

Mobile agent technology has been proven to be an efficient and effective tool for sensor networks and distributed systems. Mobile agents can be dynamically created during runtime and dispatched to remote systems to perform tasks with the most up-to-date code. Therefore, the mobility of mobile agents provides large-scale distributed applications with significant flexibility and adaptability to deal with unforeseen system perturbations. Agent technology is also used to communicate and synchronize the robots to perform complex tasks that require interagent coordination.

In the last few years a great number of agent-based methodologies have been proposed, but only some of them have been applied to develop robotic applications. As far as we know, the only agent-based methodologies used to analyse and design a robotic system are Cassipeia^[7], MaSE^[8] and PASSI^[9].

It is well-known that during robot teleoperation, the operator's performance is seriously degraded due to large time delay. Recently, teleoperation based on virtual reality (VR) has been developed to tackle the problem of large time delays. Virtual reality technology is an effective way to overcome the large delay during space robot teleoperation^[10], however, it is highly dependent on the accuracy of the model. Model errors between the virtual and real environment exist inevitably. The existing way to deal with the problem is by means of either model matching or robot compliance control.

Today, there have been numerous network interaction service models in multi-robot interactions, such as the early RPC, RMI, the more mature CORBA^[11], COM/DCOM and Web Service. But the existing network interaction service models cannot provide a dynamic service, the service is manual or semi-automatic, and with the lack of interoperability and reusability, more importantly, these models do not support multi-robot teleoperation well.

3. Architecture of the Multi-Robot

As shown below, robots communicate using a monitoring station through the wide area network. The robots communicate directly in the form of blackboard communication instead of communicating directly. Each robot can read or write messages in the MessageList in the blackboard. At the beginning, the robots retrieve tasks from the TaskList on the blackboard, afterwards they communicate and resource schedule, cooperating to execute tasks.

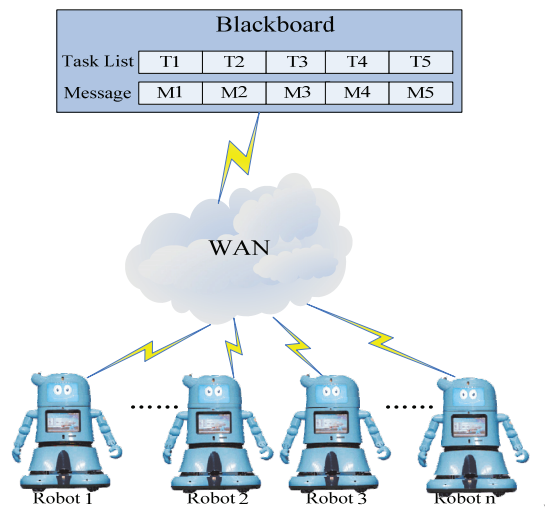


Figure 1. Architecture of multi-robot remote interaction

The Blackboard system is the expansion of the agenda in traditional artificial intelligence and expert systems, which supports distributed problem solving through the use of appropriate structures. Blackboard in multi-robots systems provides a public work area, where robots can exchange information, data and knowledge. The robot starts to by writing information item on the board, and then the other robots can use this for the system. The robot may access to the blackboard at any time to see if there is new information coming. It does not need to read all the information, but can filter out the information needed to for the current work. The robot must be authorized to access the central site. Each robot independently completes their promise to solve the sub-task.

4. Network Interaction Service Model And Its application in a multi-robot system

The capability to communicate makes it possible for agents to work together to solve complex problems which cannot be dealt with by a single agent, this being the essence of multi-agent systems (MAS)^[17]. MAS are noted for the fact that they are made up of collections of potentially independent and autonomous agents, usually heterogeneous, which work together to solve a problem which goes beyond their individual capabilities. MAS are appropriate within the domain in which the necessary

knowledge to solve a problem is distributed along different places.

The authors show the suitability of using MAS for a teleoperation robot system because (1) the loose coupling nature of a multi-agent architecture allows more flexibility in the communication process and (2) the ability to assign responsibilities to each agent is ideal for solving complex tasks in a teleoperation system. On the other hand, intelligence distribution in MAS allows dealing with questions that turn up in the development of a teleoperation system (bandwidth, productivity, speed, robustness, autonomy and scalability)^[18].

Here is an example of multi-robots finishing a work by robot scheduling and task division.

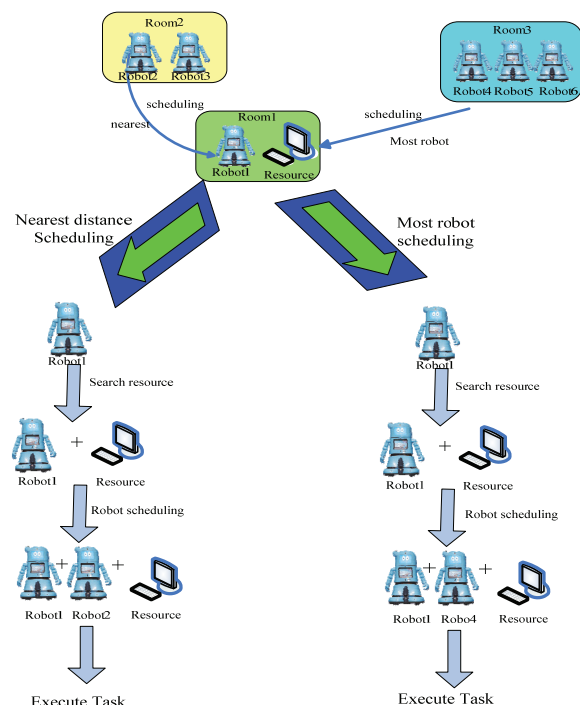


Figure 2. Two types of scheduling strategies for multi-robot

Robot1 read task T1, found the resources required for the task T1 in Room1, and then to obtain the resources from Room1, when carrying out their task, it found itself alone and could not go to achieve the task T1, so it needed to call robots in another room. Most robots use the priority algorithm in robot scheduling algorithms, however, the graph above shows two kinds of scheduling scheme. We use the nearest distance priority algorithm, that is, call for the nearest robot, collaborative tasks T1.

In the paper, we present a novel finite state mobile agent interaction service model. This model consists of three finite state machines, namely Finite State Mobile Agent (FS-Agent), which is the basic service module. The Service Content Finite State Machine (Content-FS), using XML language to define workflow, describes the service

content and service computation process. The Mobile Agent computation model Finite State Machine (MACM-FS) is used to describe the service implementation.

4.1 FS-Agent

The FS-Agent provides an external mobility detection framework, as shown in Figure 3, which is always running in the kernel of agents. Within the framework, there is a finite state machine for the mobile agent, and it is the same as the FS-Agent model in our previous work^[12]. The external mobility detection framework can ascertain whether the hosts that agents deploy are moved and then collects the new resources of the hosts in their current positions. This is especially important for the position-aware services, such as choosing different control targets when users move in smart home cases.

The life cycles of all the actual service agents in FS-MAS can be defined as an auto-motioning finite state machine driven by resources. Here resources represent the general designation of data, devices and software in remote nodes that will potentially affect the state transition of the mobile agents. That is why we call those agents FS-Agent. As shown in Figure 1, the finite state machine includes five states and one transition relationship between those states:

- Request state - each agent begins its life in this state and tries to request the resources and execution time slot which are required for the task.
- Blocked state - once an agent obtains the necessary service time, it transits into this state. All the service mobile agents who obtain enough service time while without enough service resources at certain nodes will be put into the blocked state queue.
- Suspended state - an agent will transit into this status once it obtains all the necessary resources. The system maintains a queue for the agents in this state which have enough resources while without enough service time at certain nodes.
- Migration state - this state represents that an agent tries to obtain the necessary resources from a remote node after it fails to acquire those resources from the current node.
- Serving state - if an agent obtains enough resources and service time, it will execute its service.

In a real FS-MAS system, any FS-Agent is actually executed in a container called agent home. The agent home is deployed at all the nodes and essentially provides the following services to all the agents:

- All the functions of an agent are invoked by the agent home indirectly
- Queue management and agent Execution scheduling - it maintains several different queues (e.g., blocked

queue, suspended queue, etc.) for the agents at the current node and is responsible for allocating service time for them if necessary.

- Resource searching - it will proactively search resources on behalf of blocked agents.
- Physical mobility detection for hosts, and updates the resources available in the new location when the hosts are moving.

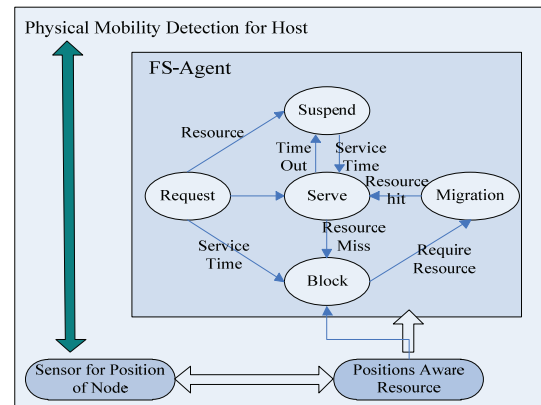


Figure 3. Transition relationship of the FS-Agent

Scheduling: Intelligent scheduling is used to determine the best combination, it is one of the most interesting research areas in artificial intelligence and intelligent control. In decision-making sub-systems of multi-robot scheduling and collaboration, in order to be able to complete the task efficiently, the general role of each robot is not fixed. Therefore, the object-oriented intelligent scheduling is set :

Here are the sets of tasks, robots, rooms and resources.

Task = {T1, T2, T3, T4, T5}

Robot = {Robot1, Robot2, Robot3, Robot4, Robot5, Robot6}

Room1 = {Robot1, Computer}

Room2 = {Robot2, Robot3}

Room3 = {Robot4, Robot5, Robot6}

Resource = {Computer}

4.2 Content-FS

The Finite state machine mobile agent-based service model is mainly used to describe the situation in the actual network topology and path of the process of implementation. As for the expression of network services and description of the content itself, this paper presents a state machine to describe the service processes, a finite state service content machine, so you can be free from complex network topologies and the implementation of technical details which is specifically application- oriented, and allowing service designers to focus more on services content. A Sservices content finite state machine (Content-FS) is defined as follows:

$$SF = (S_0, Q, R, E, \Theta, F)$$

here, Q is the implementation of all services in the state set, Q is representative of each fragment during the operation of the specific resources, driven by the implementation of the finite state machine model, also known as the service status (equivalent to the state in MACM-FS); S_0 is the initial implementation service state of the machine, which is the entrance of service state machine, there is $S_0 \in Q$; E is the end set of states, i.e. all the network service can be stopped, there is $E \subset Q$; R is the resource set which drives all state transitions in the network service; Θ is the state transition function driven by resource:

$$\Theta: Q \times R \rightarrow Q$$

F is a collection of state machine constraint rules - constraint rules defined by the specific services are to meet the specific service.

In fact, from the above definition, we can obtain the logical control relationship in the executive process of MACM-FS, the definition of the services of each state machine is driven by the model of mobile computing to start and compute. In different network topologies and distribution of information and network resources, the computing process of the service computing model driven by the same service content state machine (mobile agent service and the migration path) may not be the same, but its implementation of the service functions are the same - defined and described by the service content machine.

We describe all resources around the robot using XML language, i.e., using the state machine of Content-FS, describe the state of the robot with the state machine of FS-Agent, and finally using MACM-FS, according to different strategies for reasoning, guide the robot to complete tasks.

For Figure 5, the Content-FS is described by XML, the process is shown in Figure 5 (see Appendix).

In the mobile robot network interaction services model, the mobile robot system describes the logical flow of its concrete implementation through Content-FS. This paper uses XML technology to achieve the process of mobile robot description and expression. Then analyses through Parser is embedded in the robot home and schedule, and offers the mobile service according to logic based on Content-FS processes. In the example above, this paper presents a simple XML format Content-FS description. The description file in the graph consists of three parts, namely the state of Content-FS (STATE label), necessary resources implementation for each state (RDF description) and the state of implementation of the function.

4.3 MACM-FS

The Content-FS only describes how a specific service will be carried out logically. However, it does not provide any information about how the service agent will be executed and migrated in reality. In our previous works^[12], we have proposed a finite state mobile agent computation model (MACM-FS), however, it obviously does not support the mobility of hosts, so here an improved mobile agent computation model is required. The Improved MACM-FS offers users a unified protocol to deploy and execute the mobile agent in a real system. The protocol can be viewed as a service computing process in a real network and it is also the mobile agent trace route. The Improved MACM-FS includes four key elements:

- The network nodes set R , $R_i \in R$, in the computation environment. Any mobile agent will possibly migrate from one node to another node due to the next step's resources, the node list can trace out the mobile agent migration route.
- The nodes set with physical mobility, $R' = \{P(R_i) | R_i \in R\}$, here $P(R_i)$ is a mapping of the real node R_i after physical migration.
- Finite states in Improved MACM-FS. The states in Improved MASC-FS are constituted by two parts, one part is from the sub-service states of Content-FS, S_i , and the others are the four special operation states: message sending state M_s , message receiving state M_r , migration state Λ and the null operation state ε .
- Transition rules for the Improved MACM-FS. In Improved MACM-FS, each state transition can be presented as the form of $R \times R' \times (S \cup \{\Lambda, \varepsilon, M_s, M_r\}) \rightarrow R'$, normally the transition rules are:

$$\Phi: (R_i, P(R_i)) \times (S \cup \{\Lambda, \varepsilon, M_s, M_r\}) \rightarrow R',$$

where

1. To all the $R_i, R_j \in R$, if $\Phi(R_i, P(R_i), \varepsilon) = P(R_j)$, then $R_i = R_j$,
2. To all the $R_i, R_j \in R$, if $\Phi(R_i, P(R_i), \Lambda) = P(R_j)$, then $R_i \neq R_j$,
3. To all the $R_i, R_j \in R$, $S_k \in S$, if $\Phi(R_i, P(R_i), S_k) = P(R_j)$, then $R_i = R_j$,

To all the $R_i \in R$, if $\Phi(R_i, P(R_i), M_s) = P(R_i)$, then the next transition state relation is $\Phi(P(R_i), M_s) = P(R_i)$.

In MACM-FS, all the migration, communication (message method) and remote execution have been regarded as a corresponding state in the computation model.

In the multi-robot system in our scenario, the job is to find the optimal combination in the set of Task, Robot, Room and Resource.

In the above example, for task T1, in this paper, we adopt the nearest distance priority scheduling policy, that is, the set of completing task T1.

T1 = {Robot1, Robot2, Computer}

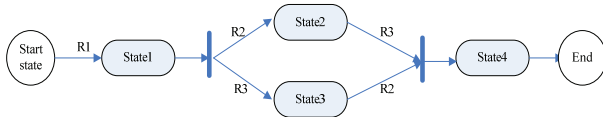


Figure 4. Robot state flow (finite state of mobile robot)

5. Appendix

```

<?xml version="1.0"?>
<ContentFS>
  <STATE stateid="SA2" call="read_the_blackboard" nstatenum="1">
    <rdf:RDF>
      <rdf:Description>
        rdf:about="http://zju.8866.org/multi-robot/read_the_blackboard">
          <TaskList>
            <Task:element>T1</Task:element>
            <Need Resource>Computer</Need Resource>
            <Task:element>T2</Task:element>
            <Task:element>T3</Task:element>
            <Task:element>T4</Task:element>
          </TaskList>
        </rdf:Description>
      </rdf:RDF>
    </STATE>
    <STATE stateid="SP3" call="search_resource" nstatenum="2">
      <rdf:RDF>
        <rdf:Description>
          rdf:about="http://zju.8866.org/multi-robot/search_resource">
            <Resource>
              <Room1>
                <Coord>
                  <X>X1_coordinate</X>
                  <Y>Y1_coordinate</Y>
                </Coord>
                <Resource:element>Robot1</Resource:element>
                <Resource:element>Computer</Resource:element>
                <RobotNumber>1</RobotNumber>
              </Room1>
              <Room2>
                <Coord>
                  <X>X2_coordinate</X>
                  <Y>Y2_coordinate</Y>
                </Coord>
                <Resource:element>Robot2</Resource:element>
                <Resource:element>Robot3</Resource:element>
                <RobotNumber>2</RobotNumber>
              </Room2>
              <Room3>
                <Coord>
                  <X>X3_coordinate</X>
                  <Y>Y3_coordinate</Y>
                </Coord>
                <Resource:element>Robot4</Resource:element>
                <Resource:element>Robot5</Resource:element>
                <Resource:element>Robot6</Resource:element>
                <RobotNumber>3</RobotNumber>
              </Room3>
              .....
            </Resource>
          </rdf:RDF>
        </STATE>
        <STATE stateid="SF4" call="robot_scheduling" nstatenum="3">
          <rdf:RDF>
            <rdf:Description>
              rdf:about="http://zju.8866.org/multi-robot/robot_scheduling">
                <SchedulingStrategy>
                  <NearestDistancePriority>
                    <Element>Robot2</Element>
                    <Element>Robot3</Element>
                  </NearestDistancePriority>
                  <MostNumberPriority>
                    <Element>Robot4</Element>
                    <Element>Robot5</Element>
                    <Element>Robot6</Element>
                  </MostNumberPriority>
                  .....
                </SchedulingStrategy>
              </rdf:Description>
            </rdf:RDF>
          </STATE>
          <STATE stateid="SM5" call="execute_task" nstatenum="4s">
            <rdf:RDF>
              <rdf:Description>
                rdf:about="http://zju.8866.org/multi-robot/execute_task">
                  <ExecuteTask>
                    <Execute:Task>T1</Execute:Task>
                    <Execute:Robot>Robot1</Execute:Robot>
                    <Execute:Robot>Robot2</Execute:Robot>
                    <Execute:Robot>Robot3</Execute:Robot>
                    <Execute:Resource>Computer</Execute:Resource>
                    .....
                    <Execute:Task>T2</Execute:Task>
                    <Execute:Robot>Robot4</Execute:Robot>
                    <Execute:Robot>Robot5</Execute:Robot>
                    <Execute:Robot>Robot6</Execute:Robot>
                    .....
                  </ExecuteTask>
                </rdf:Description>
              </rdf:RDF>
            </STATE>
          </ContentFS>
  
```

Figure 5. Content-FS describe the process of task executing

The figure above is a flow chart of the robot state in the initial state. State1, the robot reads the task. State2, robot obtains the resources needed to implement the task, but has lack of execution time. State3 is opposite to State2, the robot accesses State4 after acquiring resources and time for the task.

6. Summary

This article we develop an FS-MAS model and apply it to the multi-robot teleoperation system. We conduct multi-robot remote scheduling and resource management in a room - working together to complete complex tasks. FS-MAS enable multi-robot remote interaction to be more effective and productive. In particular, we believe that that model will enable humans and robots to work efficiently and effectively together, regardless of spatial distribution, communication channel and user interface. Further research work will focus on building a reusable network services state library, and on developing an appropriate constraint rule for some specific network systems (such as the Ad Hoc network, mobile communication network, etc.) or specific types of network services. In addition, as the popularity and development of the Semantic Web increases, how to improve robots intelligence, to better understand and identify the network resources and improve service quality, is also a very challenging research direction, but also full of practical and theoretical research value.

7. Acknowledgments

This work was partly supported by the Natural Science Foundation of China (61173123, 61203324), Zhejiang Provincial Natural Science Foundation of China (R1090244, Z1090630, Z1101048, Y1100070, Y1101237, Y1110944, Y12F02012), Zhejiang Province Community Technology Research Projects (2011C23132). We are grateful to the anonymous referees for their insightful comments and suggestions, which clarified the presentation.

8. References

- [1] Hughes S, Lewis M, Robotic camera control for remote exploration. In: Proceeding of 22nd ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2004), ACM Press.2004. 511-517
- [2] Eason G, Noble B and Sneddon I N, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (references)
- [3] Sheridan T B. Space Teleoperation through Time Delay: Review and Prognosis. IEEE Transaction on Robotics and Automation. 1993.9(5).592-606.
- [4] Weisbin C R, Rodriguez G. NASA robotics research for planetary surface exploration. IEEE Robotics & Automation Magazine.2000.7(4).25-34.
- [5] Jenkins L. Telerobotic work system-space robotics application. Proceedings of IEEE International Conference on Robotics and Automation1986.3.804-806.
- [6] Singer N C, Seering W P. Using a causal shaping techniques to reduce robot vibration. Proceedings of IEEE International Conference on Robotics and Automation.1998.3.1434-1439.
- [7] Collinot A, Drogoul A and Benhamou P. Agent oriented design of a soccer robot team. In second international conference on multi-agent systems, ICMAS'96.41-47
- [8] Deloach S, Matson E and Li Y. Applying agent oriented software engineering to cooperative robotics. In Fifteenth international Florida artificial intelligence research society conference.2002.391-396.
- [9] Cossentino M, Sabatucci L and Chella A. A possible approach to the development of robotic multi-agent systems. In IEEE/WIC conference on intelligent agent technology, IAT'03.13-17
- [10] Kelly A, Capstick E, Huber D, Herman H, Rander P and Warner R. Real-Time Photorealistic Virtualized Reality Interface for Remote Mobile Robot Control. Robotics Research.2011(70).211-226
- [11] Pytasz M, Granosik G. Applying CORBA Technology for the Teleoperation of Wheeler. Robot Motion and Control.2007.311-318
- [12] Yong LIU, Jian-hua YANG. Content Oriented Finite State Based Network Service Model. ACTA ELECTRONICA SINICA.2008.10(36).1897-1902.
- [13] Wooldridge M, Jennings N R. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review.1995.10(2).115-152.
- [14] W3C. Web Ontology Language: OWL[EB/OL]. [2004-2-10]. <http://www.w3.org/TR/owl-features/>.
- [15] Kai Qin, Huipeng Xu. Remote Moved Video Surveillance System Based on GIS. Geospatial Information. 208.6(1).87-89.
- [16] Zhu D, Gedeon T, Taylor K. "Moving to the centre": A gaze-driven remote camera control for teleoperation. Interacting with Computers. 23(2011)85-95.
- [17] Manuel Gascuena J, Fernandez-Caballero A. Agent-oriented modeling and development of a person-following mobile robot. Expert Systems with Applications.38(2011)4280-4290.
- [18] Huhns, M N and Stephens L M. Multiagent systems and societies of agents. In G. Weiss (Ed), Multiagent systems. The MIT Press. 2009.
- [19] Jun-Zhou LUO, Wen-Jia WU, Ming YANG. Mobile Internet: Terminal Devices, Networks and Services. Chinese Journal of Computers. 2011.11(34).2029-2051
- [20] Ao-Ying ZHOU, Bin YANG, Che-Qing JIN, Qiang MA. Location-Based services: Architecture and Progress. Chinese Journal of Computers. 2011.7(34).1155-1172
- [21] Sreenath R M, Singh M P. Agent-based service selection. Journal of Web Semantics, 2004.(3).261-279
- [22] Aguero C and Veloso M. Transparent Multi-Robot Communication Exchange for Executing for

- Executing Robot Behaviors. Highlights on Practical Applications of Agents and Multi-Agent Systems. *Advances in Intelligent and Soft Computing*.2012(156).215-222
- [23] Bradshaw, J M, Dignum V and Jonker C. Human-agent-robot teamwork. Publisher by the IEEE Computer Society. *IEEE Intelligent System*.2012.2(4).2-7
- [24] Bradshaw J M, Feltovich P J, Johnson M, Bunch L, Breedy M, Jung H, Lott J and Uszok A. Coordination in Human-Agent-Robot Teamwork. *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*.2008.5.467-476
- [25] Vazaios E, Narukom. A distributed, cross-platform, transparent communication framework for robotic teams. Diploma thesis, Technical University of Crete, Greece(2010)
- [26] Risler M. Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines. PhD thesis, Technische Universität Darmstadt, Germany(2009)
- [27] Andriatrimoson A, Abchiche N, Galerne S and Colle E.A Robot-Environment Cooperation Architecture for the Safety of Elderly People at Home. *Impact Analysis of Solutions for Chronic Disease Prevention and Management. Lecture Notes in Computer Science*. 2012(7251).91-98