A Loosely Coupled Control Architecture Based on Agent and CORBA for Multiple Robots

Wu Shandong ( ), Chen Yimin, He Yongyi
(Shanghai University, Shanghai 200072, P. R. China)

Abstract

With the rapid development of information technology, adopting advanced distributed computing technology to construct robot control system is becoming an effective approach gradually. This paper proposes a distributed loosely coupled software architecture based on Agent and CORBA to control multiple robots. This model provides the robot user with agent control units at the semantic level and CORBA provides function interfaces to agent at the syntax level, which shows a good adaptability, flexibility and transparency.

Key words: agent, CORBA, robot, loosely coupled, distributed computing technology

0 Introduction

In general, there are two solutions to control multiple robots: concentrated control and distributed control. For the control PC, concentrated control requires adequate computing ability and as the number of robot increases, it may result in a computing bottleneck on real time performance. In addition, because the environment of multiple robots may be an unfixed physical model and dynamically changing, the user has to reorganize robots and re-setup the software relations when the environment changes. Therefore the concentrated system has a poor adaptability, and it is difficult to be extended.

The distributed control system can resolve well the shortcomings above mentioned. What is more, using the advanced distributed computing technology we can construct a control resolution with high performance and effectively boost the controlling performance.[1,2]

In general, an excellent multiple robot distributed system should have at least two characteristics, adaptability and transparency. A flexible system can meet those unfixed robot control models. In operating robots transparency is also an important property, for the robot user would not be obliged to care about the difference of robot or control operating system. On the base of analyzing the distributed computing technique[3,1], CORBA and Agent, this paper proposes a loosely coupled control architecture for multiple robots.

1 Loosely Coupled Architecture

1.1 CORBA

CORBA is a kind of object-oriented distributed computing specification that can also be regarded as a communication middleware. One CORBA application can share and access the objects of other CORBA application, which not only provides extendibility, reuse ability but also is convenient for optimizing the software framework and constructing distributed application software.

Independent of software and hardware platforms, network protocols and programming languages, CORBA has been the most powerful cross-platform computing technology.

1.2 Agent

Agent is a physical or abstract entity that can act on itself and environment, and can react to the changing environment. It’s autonomous, reactive, deliberative, intentional and persistent.

Agent is deemed as the third generation of distributed computing technology. It can describe the system characteristics naturally and can be applied to set up a software model naturally. Agent can provide a high software re-use advantage. Agent based distributed system is composed of a group of agents which affect each other and work with collaboration according to some rules.

MultiAgent System (MAS) and Mobile Agent (MA) are two of the most important research branches of agent. MAS, set of agents, divides complex software or hardware system into little communicationable, collaborative and easily managed agents. MAS provides all kinds of agent groups and their collaboration to support a user’s tasks.

MA is based on the concept of mobile code and
can autonomously move to other computer nodes to get the necessary resources. So the MA is not limited to the original computer. When it moves, its states and code move attached, which could make it continue its interrupted works in new node.

1.3 A Agent and CORBA

Although there has been a mass of research and application on agent, it still lacks the support of standards, including the implementation mechanism. Now agent based software engineering is a research hotspot, and as far as programming is concerned, object-oriented method is still the best choice.

The advantages of CORBA include the transparency among heterogeneity operating systems and integrating different programming languages. From another view, that is the distributed multiple robots control system view, the“ object” of CORBA is syntax based, and has little granularity in the user's view. In fact, a robot user prefers a semantic based operating and controlling entity, so the agent based system is a very fit choice.

1.4 Architecture Model

According to the characteristics of agent and CORBA, the loosely coupled control architecture proposed is illustrated as Fig. 1.

![Loosely coupled architecture based on agent and CORBA](image)

CORBA is an effective approach to implement the agent based distributed system. In different layer, Agent and CORBA are complementary to each other. In this model, CORBA provides function interfaces to agent at syntax level, and agent provides interfaces to user at semantic level. Combining their advantages, the loosely coupled architecture represents a higher performance such as flexibility and transparency.

What the robot user faces is a group of agents and the top layer of the entire system is based on agent. Each agent is able to move freely and responsible for controlling a robot. The bottom layer is ORB and the agent implementation is CORBA applied. Agents accomplish their tasks by invoking the CORBA interfaces. This loosely coupled vertical architecture provides friendly and easy using robot control entity and user interface.

While exerting the location-awareness advantage of CORBA, in the multiple robot control view, there still exists the concept of location and mobility. Because CORBA is static, the agent, which is located between user and CORBA, should be mobile freely. Location-awareness mobile agent technology can make up for the limitation of CORBA appropriately. Because what the agents care for are the location-awareness CORBA interfaces and the mobility of agent drives the CORBA client to move together, and no matter where it moves, it still could invoke the CORBA interfaces correctly. In order to reduce the network load effectively, the CORBA servers could reside at relatively fixed computer nodes to control a corresponding robot.

In our control architecture, the agent model has both MAS and MA meanings. On the one hand, multiple robots are mapped multiple intelligent agents of MAS. The intelligent agent focuses on the individual function of a robot to accomplish its task. On the other hand, the agent is a mobile agent. MA is directly faced to a user, which shows the semantic level control unit. The free mobile ability of an agent in network is the base of flexibly manipulating robot and constructing robot work domain flexibly.

In fact, a mobile agent is the shell or carrier of an intelligent agent. What the user can operate are groups of registered agents with their own functions, but they do not care for the inner details of an agent. The agent groups divide the whole control system into flexible controllable units with an appropriately distributed granularity. For a robot task, the robot user first sets up a work model, then schedules agents to construct an agent work group. After assigning sub-tasks to each agent, the user sends mobile agents to the specified nodes to start work, while a mobile agent can move on its own initiative to other nodes according to the specified itineraries. After finishing the
task, the mobile agent may return to user automatically. Of course, the user can retract any agent freely at any time.

2 Agent Implementation

According to control requirements, the agent model in this paper is composed of four parts, core function, communication machine, reasoning engine and mobility, which are illustrated as Fig. 2.

![Agent model in robot control](image)

Fig. 2 Agent model in robot control

In the multiple robot control view, agent is an control unit for robot user. Following is an introduction of agent implementation in detail.

2.1 Agent Core Function

The core function of an agent is its specified task, which is the key point in distinguishing different agents. In our control system there are two types of agent to be implemented. One is control agent that is responsible for controlling robot, for example, the instruction interpreter-agent. The other is assistant agent, which is responsible for the subsidiary work in robot control process, for instance, message service agent and simulation agent.

In a distributed robot control system, the instruction interpreter is the main robot control part in executing a task, whose function consists of assigning, pausing, resuming and stopping a task. According to the CORBA programming method, we can define such IDL CORBA interface object as follows. CORBA client could invoke the CORBA server of instruction interpreter via these implemented CORBA interfaces.

```java
Interface PT500_Robot_Interpreter {
    void PT500_Assign_Task () ;
    void PT500_Interpreter_Start () ;
    void PT500_Interpreter_Pause () ;
    void PT500_Interpreter_Resume () ;
    void PT500_Interpreter_End () ;
}
```

2.2 Reasoning Engine

Intelligence makes an agent run reasonably. Knowledge base and reasoning engine provide the agent with intelligence. Each agent itself has an individual function, which needs a local rules base and reasoning engine to guide the control of corresponding robot. Besides, for the whole multiple robots control system, the collaboration among agents also requires a global reasoning and decision engine. During the running process of robots, the global reasoning engine is responsible for making decisions in time, in line with the states of agents, the global states, outer sensors data and user operations. We choose Clips, a Forward Chaining based object-oriented language for expert system, to set up the local and global reasoning machines.

2.3 Agent Communication

Communication is the intercourse and collaboration requirement among agents. In this paper, KQML is adopted as the communication protocol. KQML is a famous language that is used to interchange information and knowledge so that it can take the communication task in the format of knowledge. KQML involves a series of extensible performatives, which specify all kinds of operations to knowledge and goals so that agent could set up high-level inter-operation model above these performatives.

According to the designed communication rules in robot control, communication contents mainly include position data and control commands. The position data represent the intending robot joint position, which is used, by the global reasoning engine, to forecast collision event. There exist two types of command. One type of command is the order from user or global reasoning and decision engine to agent, such as BEGIN and PAUSE command. The other type is the interchanging signals among agents, which are used to synchronize interactive actions.

2.4 Agent Mobility

In a distributed control system based on network, the mobility of an agent is very important. Each mobile agent is able to fit in with different platform environments, which can greatly improve the flexibility of the agent. In this paper, the Aglets Workbench of IBM is selected to implement the mobile agents. Aglet is an excellent and widely used mobile agent platform in connection with the Tahiti environment.

In accordance with the loosely coupled architecture described above, the Java based CORBA client
generated from defined IDL object is invoked by the agent. At the top layer, the user manages a registered mobile agent pool[11]. The function and life cycle of MA is illustrated as Fig. 3.

![Fig. 3 Function and life cycle of Mobile Agent in robot control](image)

3 Conclusion

A multiple robots distributed control system requires high performance. The loosely coupled distributed software architecture based on agent and CORBA is a practical idea. Mature CORBA technology is the base of the whole system. And the agent, as a new distributed computing model, provides a higher-level abstract in the designing and developing robot control system. This architecture provides the user with a strong and simple agent operation interface. Agent loosely couples CORBA very well and represents a higher performance.

Under the circumstance of CORBA being widely used and agent deeply researched, this loosely coupled architecture, with an easy understandability, strong adaptability and high degree modularity, is a useful experiment and has nice practicality.

References

[5] Sebastian A, Andreas K, Jochen S. A Management Archi-