Bul. Ştiinţ. Univ. Baia Mare, Ser. B,
Matematică-Informatică, Vol. XVIII(2002), Nr. 2, 165 - 170

A DECENTRALIZED CONTROL REAL-TIME SYSTEM USING AGENTS TECHNOLOGY

Ozten CHELAI

Abstract. The process control systems are real-time systems that have to deal with unexpected situations and have to be rebuilt to meet new requirements. The paper presents how an autonomous decentralized control system can be build-using agents. System architecture is based on the multi-agent system paradigm.

MSC 2000:68M14
Keywords: Real-time Systems, Software Agents, Distributed Systems

1. Problem description

The context offered by the evolution of the system development offers new possibilities to the system design. In this context the multi-agent system paradigm has increased it's usability.

An agent brings to the system flexibility with its properties:
reactive – answering to the questions
proactive – the agent possibility to fulfill his own plans
social – the communication with other agents.

The agents are used to build dynamic and distributed applications. The agents are autonomous and execute specific tasks.

Many real-time systems are embedded in sensors and actuators and function as digital controllers. The term plant refers to a controlled system that can be an aircraft, an engine, a patient, etc. The state of the plant is monitored by sensors and can be changed by actuators. The real-time system estimates from sensor readings the current state of the plant and computes a control output based on the difference between the current state and the desired state called reference input. The computation is named the control-low computation of the controller. The generated output activates the actuators and brings the plant closer to the desired state.

In a control system we can have also controllers with functions to achieve a higher-level goals. Let say, in a flight control system, we have a flight control at physical layer and a trajectory control for the most desirable trajectories at the high level.

Using agent-based systems we have the advantages over centrally controlled systems when achieving multiple goals and managing multiple sensory input.

The problem solving that the multi agent systems can achieve is often determined by the amount of co-operation or collaboration that individual agents actively participate in and what extent information is shared [2]. The degree of co-operation or collaboration of the agency is generated by the adoption of a particular agent interaction strategy. There are *co-operative agents*, *self-interested agents* and *hostile agents*. Co-operative agents work together with the intention of solving joint problem and will be used in our architecture. Self-interested are agents that try to maximize their own good without concern for the global good, doing services to the other agents only for compensation. Hostile agents have a utility that increases with its own gains and also with competitor's losses.

Agents can pass messages to each other of changes to the environment or information about 'plans' to change the system's state.

Communication enables agents in a multi-agent environment to exchange information (resources, state information, etc.).

The agents offer to the system more flexibility.

The mobile agent technology is concerned with the ability to move executable code from one computer to another. The main benefit from adopting this approach is that available computational resources on another computer could be utilized when resources on the original computer became scarce.

In a control system using autonomous agents, we identify **local agents** that are local autonomous agents that stores the local data from the machines and **control agents** that selects a machine to process and moves on processing or controllers according to the specific procedures. The control agent acts as autonomous entities in the system. In a real-time control system the local agents can be acquisition agents that get data from acquisition machines or controllers.

The control agent selects an appropriate local agent (in real-time control systems, an acquisition agent) and directs it on how to process the product. That means that the control agent directs the local agent in charge of the machine and the local agent controls the controllers.

The control agent acts autonomously knowing system status and deals with the situations dynamically.

The agents negotiate with other agents if necessary.

The control agents are mobile agents and the local agents are stationary agents.

Local agents run on controllers who control acquisition or on computers connected to the controllers. The control agents are mobile agents and move to the computer where the target local agent is running. After moving to the target computer, the control agent sends a process request to the local agent and waits until the local agent makes notification of the end of the process.

In the system there are and other agents as: management agent and user dialog management agent that offers support to the two main agent's types.

This architecture doesn't have a centralized control mechanism and in case of a big number of machines we don't have a great load as in the case of centralized systems.

System architecture

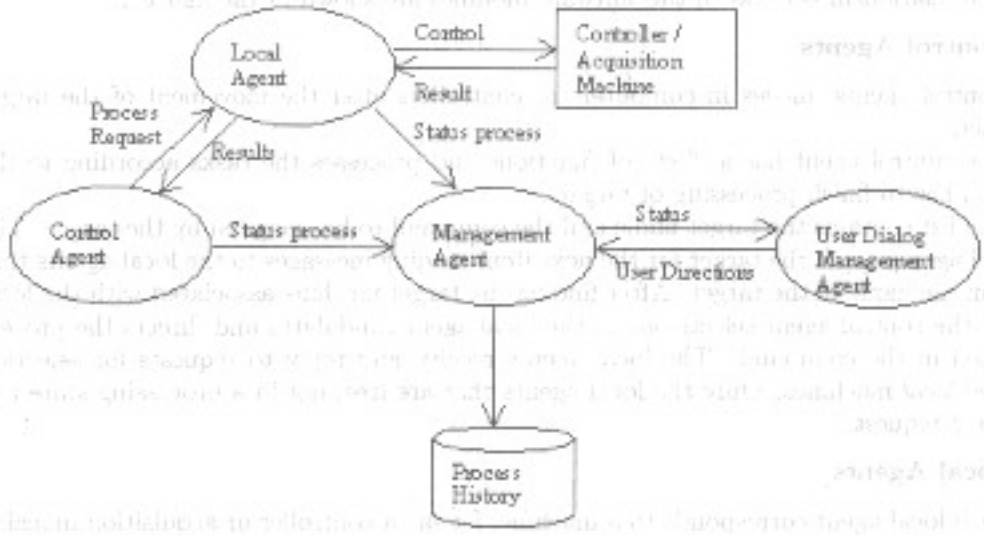

Figure : Software Architecture

There are four different high level architectures for the control systems: centralized, hierarchical, distributed local control and hybrid architecture. Each with its advantages and disadvantages.

In a real-time control system the system must be robust and must deal with run-time exceptions. The centralized architecture is often used for the control systems, but with the evolution of the new technologies, distributed control architecture can bring new advantages for these systems as:
improving flexibility.
improving robustness
improving response-time
reducing communication overhead
distributing tasks

Reactive agents are suited for making real-time and local control decisions because their internal architecture supports the requirement that they should be able to generate control actions within the available time interval. Unexpected occurrences must be handled using only local information until the strategic levels can asses the new situation.

In the real-time control system architecture we can identify four software modules:
Control Agents
Local Agents
Management Agents

User Dialog Management Agent (GUI)

The relationships between the software modules are shown in the figure 1.

Control Agents

Control agents' moves in computers or controllers after the movement of the target product.

The control agent has a "list" of functions and processes the tasks according to the list and has to finish processing of targets.

The list contains the target name and the command to be executed by the target. The control agent search the target for the next item sending messages to the local agents that contain the name of the target. After finding the target machine associated with the local agent, the control agent selects one of the local agent candidates and directs the process specified in the command. The local agents receive and reply to requests for selection not the local machines. Only the local agents that are free, not in a processing state can receive a request.

Local Agents

Each local agent corresponds to a machine, let say a controller or acquisition machine for a real-time control system. It has to know and control the status of the machine. The local agent has a name that corresponds to its function. Of course there are many local agents with the same name because they do the same function. The function description is in the list of the control agent.

The local agent can do only one job at a time. Local acquisition agent is running and gets the status of the machine.

The local agent receives directions from the control agent (only one at a time) and controls the machine after it.

Management Agents

The management agents start the control and local agents. The local agents are created at system startup, and product agents at the beginning of the processing indicated by system operators.

The management system receives status data, process histories sent by control and local agents, and stores them in a persistent manner. The management agents can monitor the activities of the machines and agents.

The management agent provides access to the database, along with the communication with external software.

Agent Communication

The agents communicate with messages.

A standard that can be used is FIPA-ACL standard ("Foundation for Intelligent Physical Agents").

The protocol used to recover information about acquisition current state asking for it from the agent in charge of maintain it can be FIPA-query. The message must allow the specification of the language used for the query (SQL, etc.)

To ask an agent for the execution of a specific action can be used FIPA-request.

The notification that provides information when a condition is satisfied or an event is triggered can use the protocol FIPA-request-when.

For the coordination and negotiation between agents can be used FIPA-contract-net protocol.

2. Implementation

An agent-based application is a dynamic, potentially large-scale distributed application in an open context. The agent application is not typical. It has a number of characteristics and requirements. Agent based applications should be autonomous, heterogeneous, open, dynamic, robust and secure. These characteristics are difficult to implement and use.

The Agent pattern [1] for mobile agent systems can be used for the control agent.

The control system can be implemented as a distributed system on an agent support system or as a CORBA implementation. The agent support systems provide a framework to develop applications based on agent paradigm and ORBs on the object-oriented paradigm. Another new alternative is using J2EE platform in multi-tier architecture.

For control agents can be used aglets developed by IBM Japan. Aglets extend the model of network-mobile code by allowing the class files to maintain state when migrating from node to node.

3. Conclusions

An agent's technology with a distributed local control is useful in building flexible real-time control systems. This architecture increases the efficiency of the system.

A multi-agent system for control brings the advantages of building decentralized system that work so good in system with a great load and the independence of the system parts.

The system architecture can be used in many fields that have a centralized control.

REFERENCES

[1] Alberto Silva, Jose Delgato: The Agent Pattern for Mobile Agent Systems. 3rd European Conference on Pattern Languages of Programming and Computing EuroPLoP'98

[2] Alex L.G.Hayzelden, John Bigham: Agent Technology in Communications Systems: An Overview. Intelligent System Application Group, University of London, January 1999

[3] Jane W. S. Liu: Real-time Systems. Prentice Hall 2000

[4] Miguel Rebollo, Vicente Julian, Carlos Carrascosa, Vicente Botti: A Multi-Agent System for the Automation of a Port Container Terminal.

[5] Hidehiko Wada, Yuichi Sakuraba, Masako Negishi: Machinery Control System Using Autonomous Agents.

Received: 2.09.2002

"Ovidius" University of Constantza, Romania
E-mail:ochelai@univ-ovidius.ro