3M−4

Noah: Environment for Distributed Agents (2) — Architecture Design —

Fumiaki SATO, Hiroshi KOZUKA, Kazuya MIYAZAKI,
Hisao FUKUOKA, Junichiro TSUJI

Computer & Information Systems Laboratory, Mitsubishi Electric Corporation

1 Introduction

Distributed computing system, in which several networked computers work cooperatively, has been getting more and more populer under the wave of down-sizing. In such environment, high cost performance and reliable services are expected. On the other hand, the distributed computing system has several new problems such as system administration, end-user's operation, and application development.

Noah (Network Oriented Applications Harmony) is a software platform for cooperation mechanism of applications in distributed computing environment. Noah provides programming environment for autonomous program modules which are called "agent".

This paper describes architecture and some characteristics of Noah.

2 Architecture

Noah provides hierarchical cooperation fields for the communication among agents. Having the inheritance mechanism in the communication field and cooperation function, Noah is able to configure the method and the scope of the applications harmony flexibly.

To realize these features, Noah architecture has the following components (see Fig. 1).

(1) Tuple Space Communication Layer:
The tuple space communication layer provides synchronization, data sharing and communication among agents.

(2) Cooperation Protocol Block:
The cooperation protocol block provides some common facilities to perform various types of cooperation among agents.

(3) Application Control & Monitoring Block:
The application monitoring & control block monitors and controls agents. Control means invocation and termination of applications, management of replication, modification of attributes, and so on.

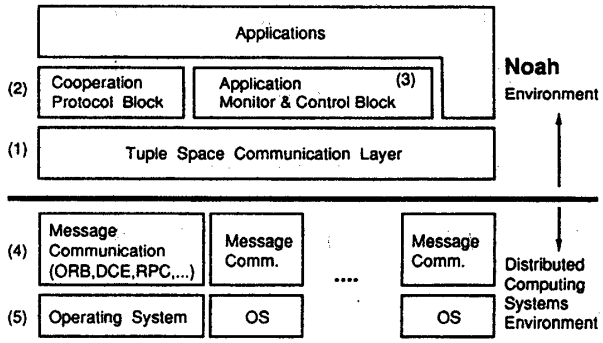

Figure 1: Noah architecture

3 Technology for each component

3.1 Tuple space communication layer

This layer provides transparent tuple space using lower message communication layer. This layer supports transparency for remote node access and remote tuple space management.

In the tuple space communication, any data exchanged among agents is treated as tuples. Agents communicate with each other by putting/getting tuples to/from the tuple space. Unlike RPC or object oriented message passing mechanisms, tuple space communication does not specify the receiver of the message. So it is very useful for broadcast, multicast and asynchronous communications.

The communication mechanism with a tuple space is suitable for the concept of Noah — applications harmony, because agents take part in the cooperation field autonomously in the Noah environment.

Noah uses an extended Linda model[1][2], which we call "LAKE" model. The LAKE model has a computational entity called "Ark" and a small tuple space. The Ark together with the tuple space composes a "Lake". Applications are composed of multiple Lakes. The Lake may create another Lake and purge it. The Ark can interact with the tuple space identified by the argument Lake. The feature of our LAKE computational model is a shared tuple space between a parent Ark and its child Ark. Fig.2 indicates the relation of the tuple space of the parent and child Lake.

Noah: Environment for Distributed Agents (2) — Architecture Design —
Fumiaki SATO, Hiroshi KOZUKA, Kazuya MIYAZAKI, Hisao FUKUOKA, Junichiro TSUJI
Computer & Information Systems Laboratory, Mitsubishi Electric Corporation.

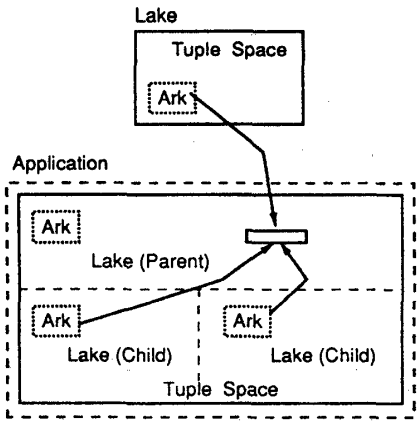

Figure 2: Cellula and LAKE model

3.2 Cooperation protocol block

To perform the cooperation among applications, many functions should be executed — choice of the application to be executed, selection of the cooperation protocol, data exchange among applications, estimation of the result caused from applications' cooperation, and so on. If we provide these procedures to each application packages, efficiency of the software development will be low expansion or configuration of the system will not be achieved easily. So we concentrate the common functions to be used in cooperative communication in a single cooperation protocol block. This approach will result in an efficient software development and easy customizing of applications.

For example, Contract net[3] is one of the basic cooperation protocols. It is a procedure to invite servers by sending a request specification to them. User can select the most suitable server by receiving response specifications from the servers and can ask the server to do the request.

3.3 Application monitor & control block

Sometimes dynamic invocation and termination of a particular agent (application), or the modification of the agent's (application's) behavior and monitoring are required to reuse the software developed in other environment. Noah can support those requirements using the function of application control and monitoring block. This block provides the invocation, termination, dynamic attribute modification of agents, and monitoring of the state of agents and systems.

Noah provides "**Probe**" and "**Activator**" which are extensions of Sensor and Actuator in Meta[4] respectively. Probe and activator re the door to monitor and control interface of agent. Monitor and control request are processed through probe and activator. The relation of agents is shown in Fig.3.

The probe and activator can be constructed or attached to the agents. The probe gets the state of internal data of the agent and field, and sends it to the monitor interface periodically. The Activator receives messages from the control interface aperiodically, and changes the state of the agent. If agents in Noah have hierarchy, parent agent can monitor and control its children with probe and activator.

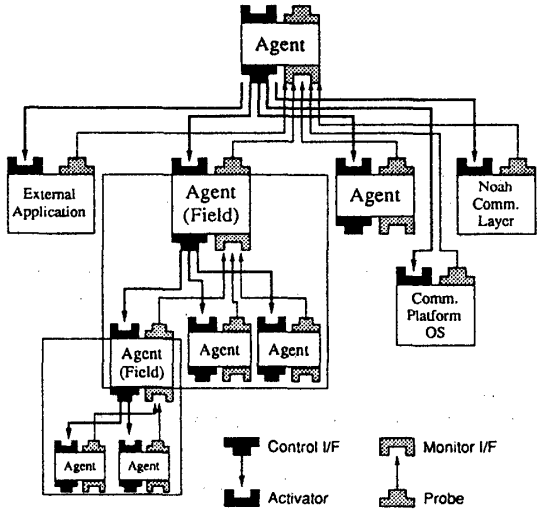

Figure 3: Probe & activator of agents

4 Conclusion

This paper described an architecture of Noah (Network Oriented Applications Harmony) for distributed environment. Noah is suitable for resource allocation, fault-tolerant applications, system management, and applications for cooperative works.

References

[1] Sudhir Ahuja, Nicholas Carriero, David Gelernter, "Linda and Friends," IEEE COMPUTER, Vol.19, No.8, 1986.

[2] Yoshida, Narasaki, "A Parallel Cooperation Model 'Cellula' Composed of 'Process + Field'," Trans. of IPSJ, Vol.31, No.7, 1990 (In Japanese).

[3] R. G. Smith, "The contract net protocol: high level communication and control in a distributed problem solver", IEEE Trans. on Comput., Vol.C-29, No.12, 1980.

[4] Keith Marzullo, Robert Cooper, Mark D. Wood, Kenneth P. Birman, "Tools for Distributed Application Management," IEEE COMPUTER, Vol.24, No.8, 1991.

[5] H. Kozuka et al. "Noah: Environment for Distributed Agents (1) — Basic Concepts —," Proc. IPSJ 46th meeting, 1993.

[6] K. Miyazaki et al. "Noah: Environment for Distributed Agents (3) — Applications —," Proc. IPSJ 46th meeting, 1993.