A Management Paradigm in Large Distributed Computing Environment

Ronlon Tsai and Jiann-Liang Chen
Advanced Technology Center
Computer and Communication Research Laboratories
Industrial Technology Research Institute
Chutung, Hsinchu, Taiwan, R.O.C. 310

Abstract

The experience with the development and application of a Cooperation Multi-Agent Framework (CMAF) derived mainly from integrated technologies of distributed processing system, knowledge-based system, and object-oriented programming style is reported in this issue. At the same time, a two-level control mechanism which contains meta-planning and local-planning is also proposed to achieve the high degree of goodness in problem-solving process. At present, the developed prototyping of CMAF system is implemented based on two existed products: COSMIC's CLIPS and SUN's RPC. To demonstrate its effect, the fault management system is built using CMAF installed on the SUN network environment. It is found that using CMAF tool is a very efficient shell of the distributed problem solver. Moreover, based on the versatile properties of reusability shell, transparency facility, and adaptive knowledge base, the CMAF will be referred to as tools for large distributed management systems in the future.

KeyWords: Distributed Processing, Blackboard System, Distributed Artificial Intelligence, Distributed Management System

1 Introduction

Because of the low hardware cost in multicomputer systems and variant communication techniques, the demand of distributed processing is terribly increasing in our society. Many distributed applications have pervaded human life, such as automated teller machine networks, air-line reservation systems, and on-site validation of credit card [1]. For the need of more and more services for human being and the diversity of equipment interface strategies, the distributed system is continually increasing in size and complexity. However, the resource management has emerged as one of the most difficult and complex problems facing broad communication industry. To efficently share resource and exchange information, the management tools are becoming urgently necessary. For the last five years, the cons and pros of the management in large distributed system are controversial in the related conferences. Whatever, the discussion results may be that the powerful distributed system is needed.

Distributed computing environment is a cluster of computing systems or agents connected together through the communication network. And each agent could autonomously process local tasks and cooperatively interoperate with other agents. Hence, two basic problems should be easily overcame. One is the local optimal control of the individual resource; the other is the global optimization of local results. By doing so, we not only analyze and incorporate the concept from previous researches but also design a management paradigm, Cooperation Multi-Agent Framework (CMAF).

The research is a part of the project carried out in CCL/ITRI in developing a fault management system for multi-vendor based network environment. Because the procedure of managing network fault is exceedingly difficult and timeconsuming, a CMAF-based fault management system which provides a distributed peer-to-peer structure for the development and implementation of fault management and functions is proposed. The processing element architecture of agent in CMAF is shown in Figure 1. In the framework, meta-control manages a whole system by driving the CMIP defined in OSI Management Standards Documents [2]. And local-control manages the local managed resource. Both the control mechanisms share their information from management communication system, such as management information base (MIB). On account of this, we can find that the application of CMAF provides a platform and a set of primitives to be manageable and scalable [3,4].

2 The Proposed CMAF

Distributed environment consists of a collection of interacting problem solvers based on a set of workstations connected by LAN. Such an environment required to adapt changing circumstances, interoperates to share knowledge and data and cooperates to join problem-solving tasks. Four-layer distributed system is shown in Figure 2 and described as follows:

Network computing system (NCS). NCS includes computing elements/workstations, communication network, and operating systems.

Distributed computing environment platform (DCEP). DCEP provides basic services including remote procedure call, distributed file system, timing, naming, security,..., and thread control. In addition, for enhancing the functions of distributed

0-8186-2755-7/92 $03.00 © 1992 IEEE

system, some more popular services such as: application service, multimedia service, and distributed object service shown in Figure 3 will be also embedded into the platform. Since vendors will support users no matter what operating system and network backbone, the design of DCEP is also called middleware.

Distributed application paradigm (DAP). DAP provides distributed problem-solving facilities for users to composite their systems.

Distributed application (DA). On top layer of the structure, users could develop their customized systems by compositing the mentioned components.

In our work, the primary goal is to create the technologies and methodologies of DAP for building cooperative, intelligent systems with modular and heterogeneous components. And the principal to design DAP is derived from expert systems, blackboard systems, and object-oriented systems [1,5,6]. For this purpose, CMAF is designed for a distributed multi-agent system. Such a system connected by a common bus to form LAN. Each processing agent of a multi-agent system can compute autonomously and cooperate with other agents to reason a satisfactory solution for a large and complicate problem. The least-commitment approach is employed in the multi-agent system and the control mechanism is split into two levels: local-planning and meta-planning. The two-level control mechanism is iteratively and interchangeably executed to approach a suboptimal solution [7,8].

2.1 CMAF Architecture

According to the object-oriented design, a large sophisticated software system development should be approached by a systematic modularity. And a high-dimensional modular development methodology is good for uniformity and reusability in program structure. Hence, the functions of CMAF are structured in modulus in the research. Two major moduli in each processing element are knowledge sources and blackboards. The moduli and control mechanism used by the CMAF are described as follows.

2.1.1 Knowledge Sources

As the development of system is extended, it is envisioned that many local expert systems will be developed in agents to assist users in problem resolution and analysis in loosely-coupled operational areas. And each particular expert, knowledge source (KS), possesses its knowledge and reasoning philosophy. From this point of view, five categories of KS are classified as follows: local-planning KS, meta-planning KS, communication KS, problem domain KS, and constraint KS. Both features which is equally important to the above mentioned is the autonomous capability and interoperability in the framework.

Autonomous capability. For autonomous capability, the agent includes the local expertise and control function. The former consists of problem domain KS and constraint KS and the latter is composed of local-planning KS and meta-planning KS, shown in Figure 2. Problem domain KS is the definition, parameter, and conditions of problem itself. Through the problem-solving cycle, decomposition-allocation-synthesis [9], the problem domain could be described by heuristic rules. In brief, problem domain KS manipulates heuristic rules and facts in the application domain. However, constraint KS defines the restriction of local optimalization and handles partial solutions, based on the limitation of the problem domain as well as the confidence level of objects. Since the goal is to share some resources or solve the conflicts among them, the reasoning process will produce the best result with compromise and tradeoff. Due to the long reasoning cycle of match, conflict resolution, and action, efficiency should be taken into consideration. By doing so, it can be easily proved that local-planning KS, incrementally collecting partial solutions to local optimality, and meta-planning KS, organizing local optimality, maintaining global status, and producing the best result, will play important roles in the CMAF.

Interoperability. For interoperability, agents should be aware of the state knowledge and the role knowledge, and then receive and broadcast results through a common bus by providing the communication KS and facilities. The functions of KS include the user-interface, interrogation, error correction, interagents messages,... and so on shown in Figure 6.

2.1.2 Blackboard Modulus

The Blackboard system is shared by cooperating KSs which work together to achieve the goal. In the CMAF, the blackboard modulus consists of data blackboard and control blackboard. And for parallelism, communicativity, increment, and plurality, both blackboards are structured into several layers. Data blackboard which concerns about synthesizing solutions are separated into five layers: basic answer, hypothesis, partial solution, local optimality, and global optimality. In addition, to enhance the efficiency of inference process, control blackboard is organized in order of operation, model, policies, evaluation metrics, and network layers.

2.2 Control Mechanism

Facing CMAF, two basic problem should be considered. One is the optimal control of the individual resource, the other is the collection of the local optimality into the global suboptimal. To attack these problems in large distributed processing, the two-level control mechanism is employed in CMAF approach. The lower level, local-planning by result-sharing, is for local optimalization. And the upper level, meta-planning by task sharing, is for global optimalization [10]. The control mechanism is iteratively and interchangeably executed to approach a suboptimal solution and described as follows in details.

2.2.1 Local-Planning

In local optimality, the problem is the same as the space search problem. The process starts from the root of the search tree and incrementally collects the partial solutions until the goal is completed. The kind of optimality-type problems have been classified as

NP-complete. Therefore, AI heuristic technologies are introduced to obtain the satisfactory solution. Moreover, the best search and pruning techniques are also applied to reduce combinatorial explosion in search space. However, the local-planning will execute to obtain the optimal solutions according to blackboard system. In the system, local-planning KS processes the activated task scheduling and focuss attention to achieve local goal efficiently. The policy of scheduler is given by internal or external control priority, activated task life, and organizational significance. The focus of attention mechanism is to give local-planning KS the ability to decide which hypothesis it should pursue first through the activation of its available knowledge sources. Indeed, the focus of attention is determined by evaluating potential knowledge source activation against various criteria and then by selecting the most highly rated task for execution. During the processes, the KS applies conflict resolution reasoning methodology based on ultimate objectives and constraints. Hence, the result-sharing strategy could reason the partial results in different degrees concurrently as well as the incomplete information. The local-planning control mechanism is described in Figure 4.

2.2.2 Meta-Planning

The local level which is individually optimized may not be mutually optimal in global result. The other main concept in two-level control mechanism is the delay-binding based on the least commitment methodology. The methodology defers the commitment until the agent is needed. To reach the final goal, the meta-planning provides the following steps. Firstly, collect the local best results. Secondly, find crucial solutions, bind partial resources to the most critical case, and manipulate the incomplete or uncertainty events. Thirdly, broadcast the binding results and issue another action to local optimalization. Finally, repeat and execute the above steps until no further improved result is provided. The mechanism of meta-planning is shown in Figure 5.

3 Implementation

The design of CMAF not only presents the idea and method of the framework but also completes the prototyping system. The developed CMAF is implemented based on two existed products: COSMIC's CLIPS and SUN's RPC. Since CLIPS is based on C language, this expert system tool was selected as the basis to develop the autonomous functions of agents in framework. Then the functions of interoperability are implemented by distributively processing the CLIPS, which modified the source code of CLIPS by adding RPC function call facility. In addition, each agent is also endowed with six states. The initial state begins on "start" state (state 1). After running, the agent compares the list of facts with preconditions in rules (state 2). Two situations will happen no matter whether the match was found or not. If there is no firable rule, this agent will go to "sleep" state until being awaken by another agent (state 3). If there are some firable rules, the most critical one will be selected by predefined criterions (state 4). And the rule will be fired in "action" state (state 5), then action should be carried out and the effect should be broadcasted to other agents (state 6).

3.1 Autonomous Mechanism

CLIPS is developed by NASA for evaluating the potential use of AI technology. And CLIPS supports not only fast execution and low cost but also simplicity and portability between different hardware platforms. Furthermore, the new vision, CLIPS 5.0, with the object-oriented property can provide users the transparent facilities to develop software system through rule-based statement easily. In addition, the inference engine drives the reasoning of forward chaining as well as the backward chaining. Based on RETE algorithm for efficiency, the inference cycle consists of match, conflict resolution, and action.

3.2 Cooperative Mechanism

There are two daemons in each agent: autonomous process and communication process. Autonomous process described in the previous section performs the reasoning cycle. And communication process is responsible for intercommunication through the asynchronous RPC primordial client-server IPC in distributed computing environment. Although SUN's RPC provides sufficient functions hidding details of complicate knowledge in communication network, asynchronous facilities are replaced to reduce length blocking time in client cite. In CMAF, the functions of interoperability achieved by the communication process are shown in Figure 6.

4 Application

The functions of network management is to control and supervise the resources in network. The major goals are to exchange information and share resources in the manner of high performance, reliability, and availability. According to ISO draft international standards, network management systems include five functional managements, the basic management frameworks, and structures [11,12]. In the network management system, the procedure of managing network fault is exceedingly difficult and timeconsuming because of the size and complexity of the network problems. Hence, the paper concentrates on the issue of fault management. The fault management is a collection of activities needed to dynamically maintain the services in network level. These activities are divided into four categories: fault detection, fault diagnosis, fault collection, and fault administration [13]. Once an anomaly condition occurs, fault management system should detect faults, isolate, recovery, and record the information. In brief, the fault management system is described to accept external events, operator data, and other management function as input command. Moreover, the system shares the fault historical database, diagnosis rules, and fault event modeling to process event reports, trend analysis, and issue correction command sequence. The block diagram is shown in Figure 7.

To manage the large and multi-vendor networks, the integrated network management systems (INMS) with continuous network operations become critical. In the application, we proposed an approach of CMAF. A logical connection between an INMS implemented by CMAF and a number of element management systems implemented by CMAF is presented in Figure 8. Network managed by the INMS includes modems, DSU/CSUs, multiplexers, bridges, and routers, just naming a few. Furthermore, the CMAF-based expert systems can share communication KS, meta-planning KS, and local-planning KS. Therefore, by substituting the resource domain knowledge, the adjustment fault management system can exchange to perform the resource management.

5 Conclusions

Because of the low hardware cost in multicomputer systems and variant communication techniques, the need of applications in distributed computing system have increased. To fit any heterogeneous system, software system vendors provide DCSP with various services. Although those systems are based on open and interoperability system, it needs a management paradigm to coordinate in heterogeneous environment. Therefore, a development tool, Cooperated Multi-Agent Framework, is designed to alleviate the programming burden. The CMAF includes five independent reasoning modules, knowledge sources, and two global work area, data and control blackboards. And the CMAF is also embedded into a two-level control mechanism: local-planning and meta-planning. The prototyping of CMAF is not only provided with a fault management system but also designed as a distributed problem solving paradigm. However, this primitive project stimulated some other researches and the results will be issued in the future. Above all the versatile properties of reusability shell, transparency facility, and adaptive knowledge base in CMAF will be the future trend of large distributed management system.

6 Acknowledgement

The authors would like to express their gratitude to the Advanced Technology Center of Computer and Communication Research Laboratories of Industrial Technology Research Institute for computer facility support.

This paper is a partial result of the project no. 35N5000 conducted by the ITRI under sponsorship of the Minister of Economic Affairs, R.O.C..

7 Abbreviation

CMAF : Cooperation Multi-Agent Framework

CLIPS : C Language Integrated Production System

RPC : Remote Procedure Call

CCL : Computer and Communication research Laboratories

ITRI : Industrial Technology Research Institute

CMIP : Common Management Information Protocol

OSI : Open Systems Interconnection

MIB : Management Information Base

NCS : Network Computing System

DCEP : Distributed Computing Environment Platform

DAP : Distributed Application Paradigm

DA : Distributed Application

KS : Knowledge Source

IPC : InterProcess Communication

ISO : International Standards Organization

INMS : Integrated Network Management System

EMS : Element Management System

MO : Managed Object

References

[1] Popescu-Zeletin R. et al., "'Y' Distributed Application Platform," Computer Communication, Sept. 1991.

[2] ISO/IEC DIS 10164-4, OSI Management Standards Documents, June 1990.

[3] S. Goldsmith and U. Vizcaino, "Enterprise Network Management," Integrated Network Management, I, Elsevier Science Publishers B.V., IFIP, pp. 541-552, 1989.

[4] J. Pasquale, "Using Expert System to Manage Distributed Computer Systems," IEEE Network, Sept. 1988.

[5] C. Horn and V. Cahill, "Supporting Distributed Application in the Amadeus Environment," Computer Communication, Sept. 1991.

[6] B. Maitre and H. Laasri, "Cooperating Expert Problem-Solving in Blackboard Systems: ATOME case study," Decentralized A.I., pp. 251-263, 1990.

[7] E.C. Ericson, L.T. Ericson, and D. Minoli, Expert Systems Applications in Integrated Network Management, Artech House, Inc., Norwood, MA, 1989.

[8] M.J. Haw, B. Harrow, and S. Herman, Distributed Artificial Intelligence for Multi-Agent Problem Solving and Group Learning, IEEE, 1991.

[9] J.L. Chen and R. Tsai, "The Manager's Consultant: An Intelligent Framework for Network Management," Proc. of the IEEE/ACM International Conference on Developing and Managing Expert System Programs, pp. 380-387, 1991.

[10] R.G. Smith and R. Davis, "Frameworks for Cooperation in Distributed Problem Solving," IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-11, No. 1, pp. 61-70, 1981.

[11] K. Terplan, Communication Networks Management, Prentice-Hall International Editions, 1992.

[12] J. Malcolm and T. Wooding, "IKBS in Network Management," Computer Communication, Nov. 1990.

[13] "OSI-Based Network Management," DataPro, C05-010-651, June, 1989.

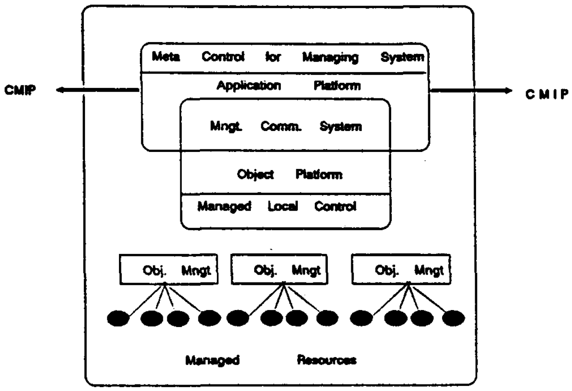

Fig. 1 Processing Element Architecture

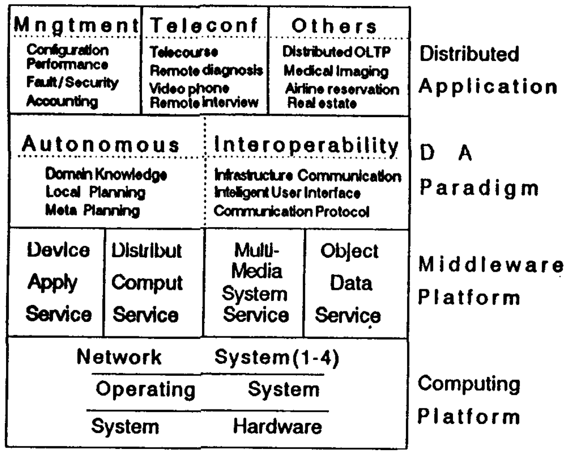

Fig. 2 Four-layer Distributed System

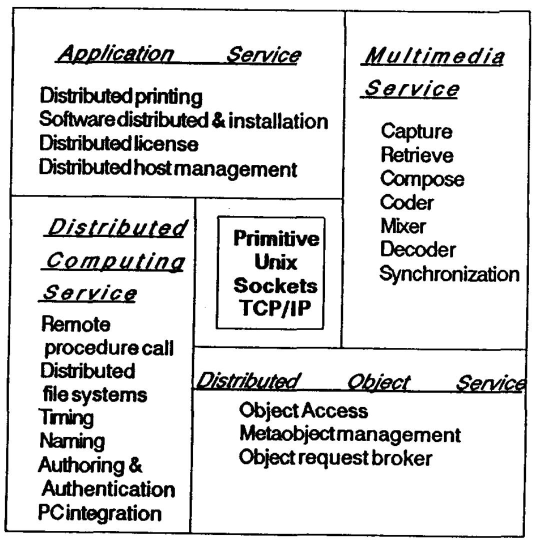

Fig. 3 Middleware Platform

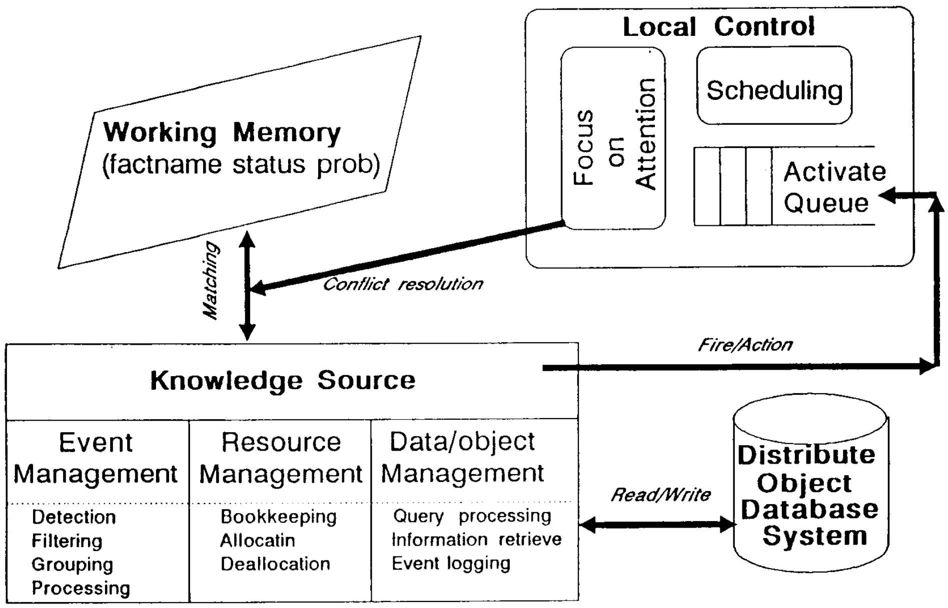

Fig. 4 Local-planning

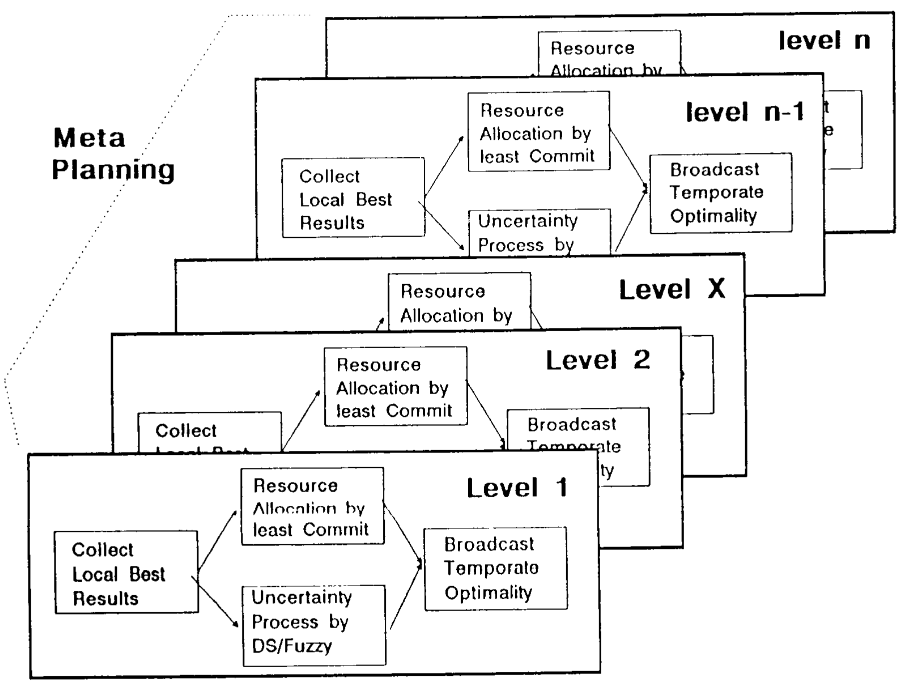

Fig. 5 Meta-planning

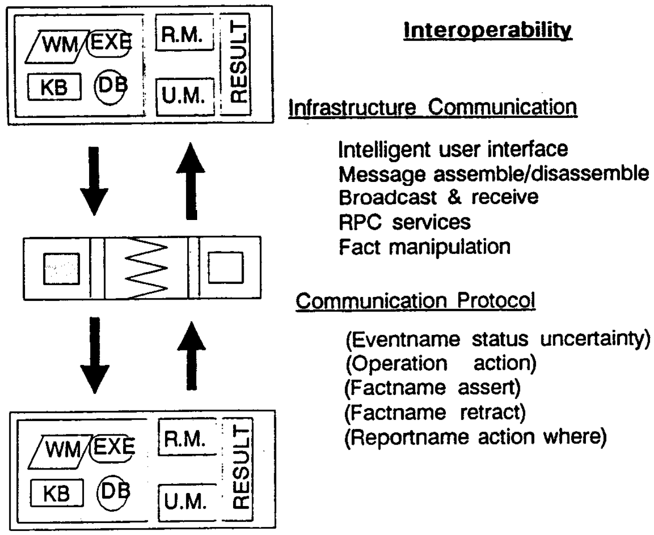

Fig. 6 The Functions of Interoperability

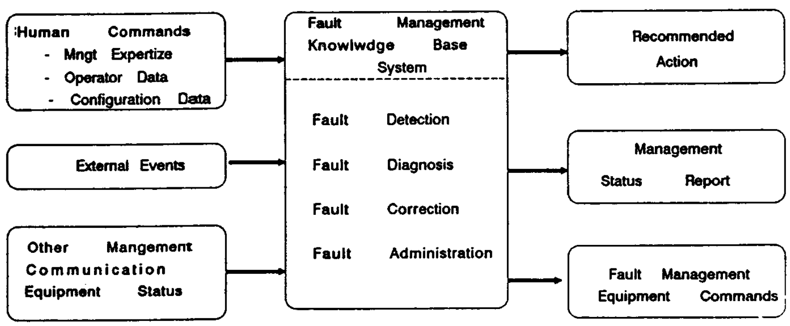

Fig. 7 The Functions of Fault Management

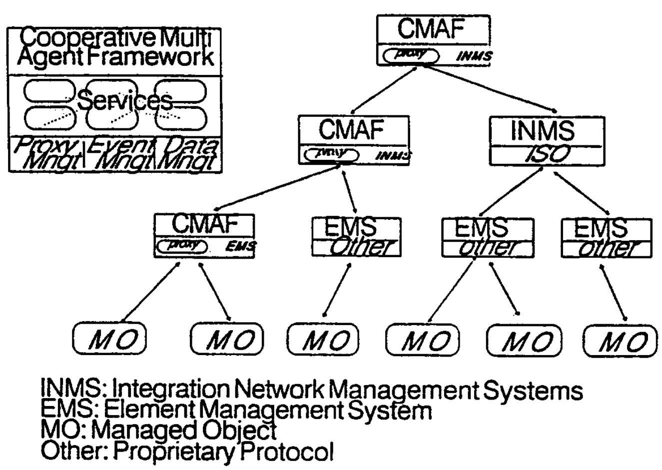

Fig. 8 Logical Connection Between an INMS and a number of EMSs