異質環境下之分散式系統管理

Distributed System Management in a Heterogeneous Environment

蕭明倫　林伸峰　劉安之
M. L. Hsiao, S. F. Lin, and A. C. Liu

Department of Information Engineering
Feng Chia University
Taichung, Taiwan

摘要

本文中我們提出一個三層分散式系統架構，這一個架構可以達到分離分散式系統中使用者介面，應用程式，與系統資源的目的，並且將其應用在我們所發展的分散式系統測試床 SD^2. SD^2 可以同時存在於 OSI 與 TCP/IP 的網路環境上，透過 OSI-Internet 閘道器而能同時管理 OSI 與 Internet 網路. 我們將 SD^2 的所有系統資源定義爲被管理物件. 並且利用 MIB 轉換工具完成了異質網路管理服務的模擬.

Abstract

In this paper, we propose a three-layer distributed architecture which isolates the interface, application, and resource of a distributed system. We apply this architecture to our distributed system testbed, SD^2. SD^2 exists in both OSI and TCP/IP network environments, and interworking with OSI and Internet management protocol through OSI-Internet management gateway. All resources of SD^2 are modeled as managed objects (MOs). In order to achieve heterogeneous network management, we utilize MIB translation tool and realize management services emulation.

Keywords: distributed system, network management, heterogeneous environment.

1. This work was supported by National Science Council, Republic of China, under contract NCS86-2213-E035-004.

1. Introduction

Distributed system based on client-server concepts provides a highly flexible system, but the management of the system is more complex and difficult. Furthermore, the distributed system contains nodes on a heterogeneous environment using different protocols. The network management applied to these networks should be integrated to support the overall functions.

In this paper, we introduce a three-layer distributed architecture. The distributed system management interface is achieved by OSI and Internet management services, and the interworking uses an OSI-Internet management gateway to perform the management service translation. We apply above to our distributed system testbed, SD^2. The management information of SD^2 is represented using the MO definitions of these two management protocols.

This paper is organized as follows. Section 2 introduces a three-layer distributed architecture. Section 3 presents the heterogeneous network management services translation. Section 4 shows an example to demonstrate our work with SD^2. Section 5 concludes the paper.

2. Three-Layer Distributed Architecture

In general, the network elements of a distributed system can be classified into three types: interface, application, and resource. A user employs the interface to obtain information on applications. User doesn't have to care what application they need to process, and where the application is located. The interacting applications make use of the system resource when necessary to complete the assigned task. Similar to the interface, the application doesn't care where the resource is supported. This architecture isolates the interface, application, and resource of a distributed system.

The distributed system that we manage in this paper is SD^2 (A System for Distributed Software Development) [1]. SD^2 provides a distributed processing environment in which the processes of a distributed application program allocated to various heterogeneous nodes can refer to each other in order to submit or receive messages. Furthermore, the users can observe the behavior of distributed application programs, and further use the tools provided by SD^2 to find out the bugs.

There are seven subsystems on SD^2: visualization tool, CSP transformation, Petri Net monitor, debugger, system server, system client, and OSI to Internet gateway. In SD^2, all of the messages between interface-application and application-resource are CMIP or SNMP [2]. Excluding the system server and system client, the relationship of SD^2 components that mapped with three-layer distributed architecture will be described later.

We support SD^2 to running on OSI and Internet network environment [3]. The system server is an OSI manager, which runs on an OSI network [4]. On the other hand, the agent is part of the system client that can be on OSI or Internet networks. The system server can manage its clients on Internet network through an OSI to Internet management gateway [5].

Fig. 1 shows our heterogeneous environment. The system server uses OSI management as the uniform management platform to integrate OSI and Internet management. It maintains a host table to record what network environment that system clients are on. When the system server sends a request to the client on Internet network, the system server should check the host table and sends this request to the management gateway, which will transform this request to Internet management target client. That is the same with events and traps.

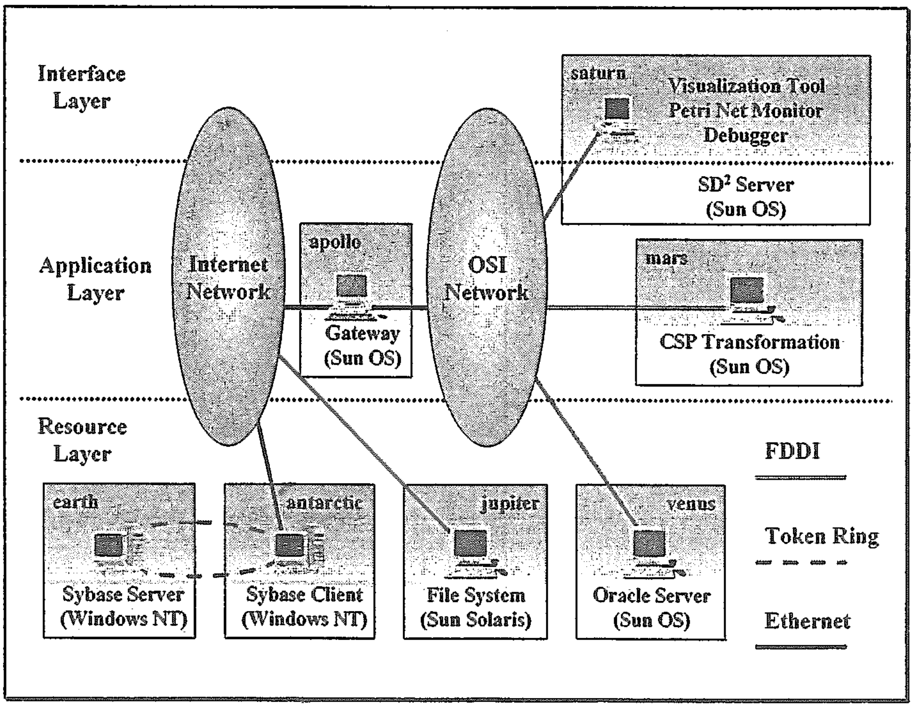

Fig. 1 Our Heterogeneous Environment

3. Heterogeneous Network Management Services Translation

An important issue among different network elements is their heterogeneity. For the purpose of network management, we integrate two major network management protocols: CMIP and SNMP. Interworking is defined as the ability for OSI and Internet management to coexist in the same distributed management environment with some level of integration between them. The objective of interworking is to provide a single, uniform end-to-end view of the managed network, no matter what the underlying protocols are.

In order to implement the interworking strategy, three methodologies are defined: protocol translation, MIB translation, and service emulation [6]. Because in many cases protocol syntax cannot be directly translated (e.g., CMIP scope Get, SNMP GetNext), protocol translation requires some aspect of MIB translation and service emulation in a complex and multi-step interworking strategy. In this paper, we focus on service emulation and utilize a translation tool: "imibtool" [7] to achieve MIB translation [8].

OSI	*Internet*
M-Get	Get (non-table entry) Get-Next (table entry + scope)
M-Cancel-Get	No equivalent service
M-Set	Set
M-Action	No equivalent service
M-Create & M-Delete	Set (table entries)
M-Event-Report	Trap

Table 1 Operation of Service Emulation

Service emulation involves mapping OSI management services into Internet management services, or vice versa. Performing OSI management service emulation requires the Internet MIB's schema information that is described in OSI GDMO templates, and requires some functional mapping. NMF Forum 028 [9] has defined some elements of CMIS service emulation (Table 1).

The management gateway is responsible for emulating CMIS requests by generating SNMP requests, using SNMP responses to generate CMIS responses, and mapping SNMP traps to CMIS notifications. There should be an emulation procedure, and the Internet MIB has to be translated into OSI GDMO providing the MIB schema for service emulation and naming mapping.

To perform the CMIS emulation, the proxy or gateway can use either of the approaches, stateless or stateful approach, described by IIMC Strategy [6] to retrieve or modify Internet MIB information.

4. An Example

The managed resources that SD^2 uses should be modeled as managed objects. The status of an object is a measure of its behavior at a discrete point in time and is presented by the current value of a set of status attributes. On the other hand, the event of an object is the reflection of a change in its status.

Based on the features of managed resources, we define four managed objects: *distributed_system*, *compiler*, *database*, and *process* (Fig. 2). The *distributed_system* object records the status of SD^2 such as whether the local system is ready or the data will be dispatched to process by remote node. The *compiler* object corresponds to compiler of each target node. SD^2 uses compiler object to set the compiler environment and control the compiler. The *database* object is used to access the database of SD^2 which stores the information of processes decomposed from user's application program in every target node. The meaning of *process* object is to model the behavior of processes running on each node of SD^2. The behavior of a process that we concern with is the communication between processes.

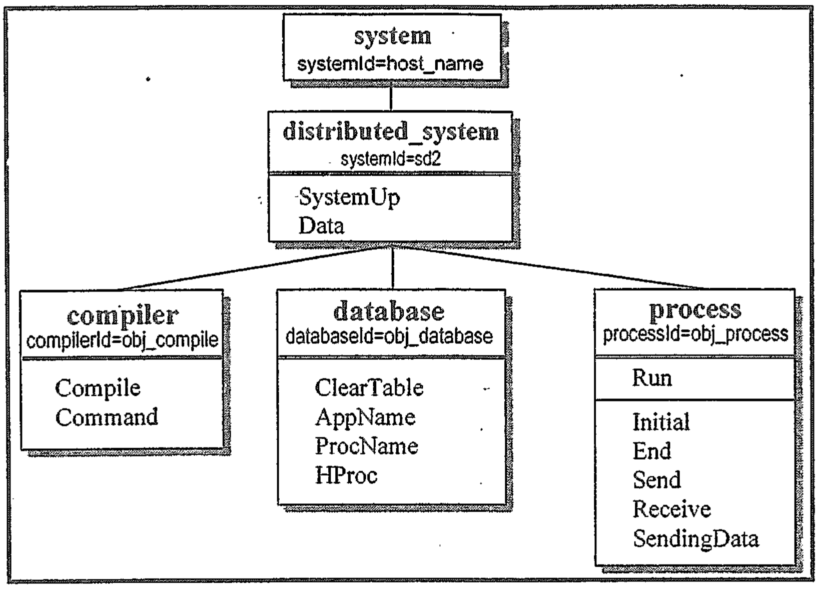

Fig. 2 Containment Tree of SD[2]

CMIS Service	*SNMP Service*
M-Get (MOC, systemId=gateway@systemId=host_name @MOCId=NULL, Attribute, baseObject)	Get (host_name, Object)
M-Get (MOC, systemId=gateway@systemId=host_name @MOCId=NULL, NULL, wholeSubtree)	n×Get-Next (host_name, Table or Group)
M-Set (MOC, systemId=gateway@systemId=host_name @MOCId=NULL, Attribute, Set_Value, baseObject)	Set (host_name, Object, Set_Value)

Table 2 Management Service Translation

The management services translation that performed in gateway is shown in Table 2. In the CMIS services, the *MOC* parameter indicates the managed class, which is translated from SNMP MIB. The *systemId=gateway@systemId=host_name@MOCId=NULL* parameter is a distinguished instance name of the *MOC*. "systemId=gateway" points out the gateway host, "systemId=host_name" presents the SNMP agent host name, and "MOCId=NULL" is a name of the instance of *MOC*. The *Attribute* parameter is the attribute of the *MOC* that the service is accessing. The *Set_Value* parameter of the M-Set service is the value that will be set to the attribute. The *baseObject* and *wholeSubtreel* parameters are scopes the service applying to the *MOC*. *baseObject* shows that the service only operates the *MOC* object. *wholesubtreel* means that the service operates the objects in whole subtree from the *MOC* object.

In the SNMP services, the *host_name* parameter is the host where the service request is delivered. The *Object*, *Table*, and *Group* are the targets that the service applies to. The MIB definition for SNMPv1 does not contain a macro for representing traps. In SNMPv1, traps are considered part of the protocol but not part of the MIB. Therefore, the *trap* of SNMPv1 cannot be translated to *notification* of CMIP being emitted by specific managed object in MIB translation step directly.

We define the possible notifications mapping to SNMP traps. The number of trap is enterpriseSpecific(6) in an OID table. The internetSystem object in the management gateway emits this notification. We use two examples in managing SD² to explain how CMIS M-Set and SNMP Trap are translated into SNMP Set and CMIS M-Event-Report. Before the management service translation, the SNMP MIB should be translated into CMIP GDMO MIB. According to this GDMO MIB, we will generate a Simple Input File (SIF) table and an OID table providing the information for translation. The SIF table records the structure of the SNMP MIB in the view of OSI MIB. The OID table maps OSI managed objects and attributes to the object identifiers of SNMP groups and objects.

The first example shown in Fig. 3 is to set the *Data* object belonged to the *distributed_system* group of SNMP MIB with a value "24" in the host "jupiter". We should give a CMIS M-Set request to the management gateway executed in the host "apollo". The M-Set request we give is:

M-Set (distributed_system, systemId =apollo
@systemId=jupiter@systemId=NULL,
Data, 24, baseObject)

The management gateway looks up the SIF table to check the MIB structure in the CMIP view. Then, in the OID table, it finds out the OID of the SNMP object (1.3.6.1.4.1.300.1.2) mapping to the attribute (*Data*) that we assigned. Finally, the Set request of SNMP is sent to the agent in host "jupiter".

For the second example, we translate a SNMP trap that is emitted when a process (in a host whose IP address is 140.134.24.26) sends a message(24) to the other process (Fig. 4). This trap is an enterprise trap(6) and the specific trap number is 2. The enterprise OID of SD² is {1.3.6.1.4.1.300}; however, this OID is an experimental one.

First, we define a *snmpDataSending* notification of the *internetSystem* object in the management gateway to map to it. According to the enterprise OID and specific trap fields of the trap, we give the notification a registration identifier, {1.3.6.1.4.1.300.0.2}, recorded in the OID table. So, when a trap arrives at the gateway, we can use the enterprise OID and specific trap fields of the trap to find out which notification mapping to it in the OID table. By M-Event-Report service of CMIS, the *internetSystem* object emits this notification containing the information that consists of the IP, OID, and Value fields of the SNMP trap to OSI manager.

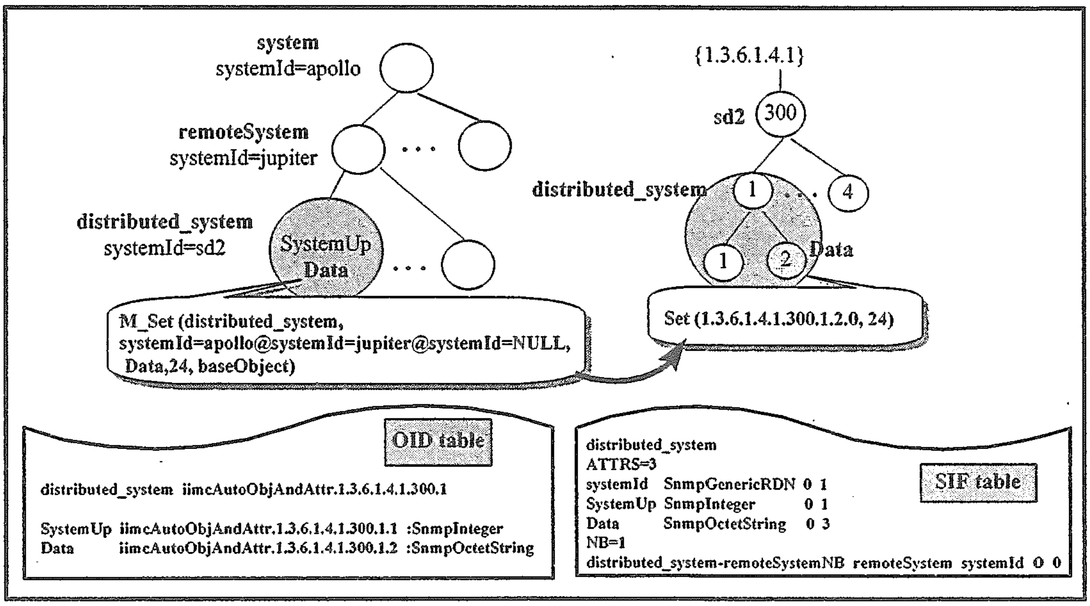

Fig. 3 Translate CMIS M-Set into SNMP Set

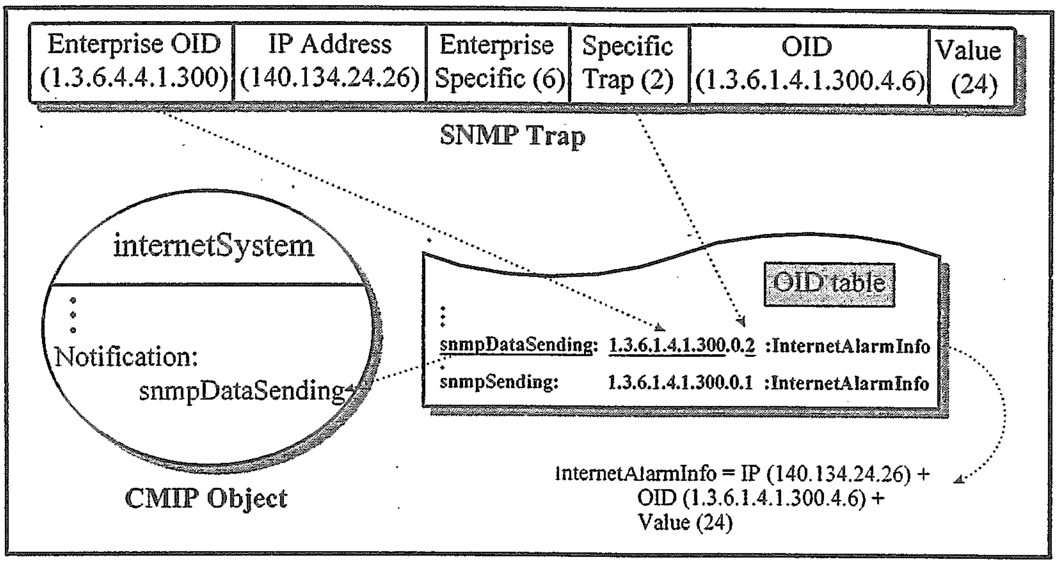

Fig. 4 Map SNMP Trap to CMIS Notification

5. Conclusion

In this paper, we propose a three-layer distributed architecture and apply to our SD[2] system with a heterogeneous environment. We have achieved two major tasks. First, we have a consistent interface to manage a distributed system. In other words, we can access the management information of a distributed system through the standardized management services. Second, the distributed system can exist in a heterogeneous network, i.e., OSI and Internet network environments. An OSI to Internet management gateway is adopted to achieve the interworking between the two environments.

Reference

[1] A. C. Liu, *分散式軟體發展測試床 (III)*, 行政院國家科學委員會專題研究計畫成果報告, Jul. 1995.

[2] W. Stallings, *SNMP, SNMPv2, and CMIP - The Practical Guide to Network Management Standards*, Addison - Wesley, 1993.

[3] M. L. Hsiao, *Distributed System Management in a Heterogeneous Environment*, Master Thesis, Institute of Information Engineering, Feng Chia University, Jul. 1997.

[4] R. H. Stratman, "Development of an Integrated Network Manager for Heterogeneous Network Using OSI Standards and Object-Oriented Techniques," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 6, pp. 1110-1120, Aug. 1994.

[5] K. McCarthy, G. Pavlou, S. Bhatti, and J.N.D. Souza, "Exploiting the Power of OSI Management for the Control of SNMP-capable Resources Using Generic Application Level Gateways," *Integrated Network Management IV*, pp. 480-493, Chapman & Hall, 1995.

[6] Network Management Forum, *ISO/CCITT and Internet Management: Coexistence and Interworking Strategy - Issue 1.0*, Oct. 1992.

[7] J. Reilly, *Modifications to SMIC 'SNMP MIB Compiler' to Produce GDMO MIB Definitions*, Technical Research Center of Finland Telecommunications Lab. (VTT/TEL), May 1993.

[8] Network Management Forum, Forum 026: *Translation of Internet MIBs to ISO/CCITT GDMO MIBs*, Mar. 1994.

[9] Network Management Forum, Forum 028: *ISO/CCITT to Internet Management Proxy*, Mar. 1994.