A Management Framework for CORBA-based Distributed Services and Applications

Ji-Young Kong
Dept. of Computer Science and Engineering
POSTECH
Email: konga@postech.ac.kr
http://www.postech.ac.kr/~konga
Contents

• Introduction
• Related Work
• Mgmt Architecture
• Mgmt Interface Object
• CDS MIB
• Mgmt Services
• Prototype Implementation
• Conclusion & Future Work
Introduction

• Explosive growth of CORBA-based Distributed Services (CDS) on Internet & telecommunication networks
• CDSs need to be monitored and controlled to provide reliable and efficient services.
• There does not yet exist international standards for managing CORBA-based services or applications.
Goal of Thesis

• Design a mgmt framework for managing CORBA-based distributed services and applications, including
  – Mgmt Architecture
  – Mgmt Interface
  – Mgmt Information
  – Mgmt Services
Related Work (1)

- IETF’s MIBs for applications
  - Network Services Monitoring MIB
  - System Application MIB
  - Application MIB
  - Mail Monitoring MIB
  - X.500 Directory Monitoring MIB
  - WWW MIB
Related Work (2)

• MAScOTTE Project
  – Part of ESPRIT project
  – Aims to provide a set of management services for object-oriented distributed systems.
  – Focuses on
    • Extension of CORBA-based systems for management
    • Management facilities
    • External management platform to CORBA Gateway
CORBA-based Service Model

Object Request Broker

Remote object method call
Intra-server method call
Mgmt Architecture

A Management Domain

Management Server

Mgmt Application

Management API

Application Interface

App

Application Interface

MIB

Other Mgmt Server

management communication application communication
Mgmt Interface Object (1)
Mgmt Interface Object (2)

- Enables MSO to monitor and control servers
- Instrumented in every managed server (SO)
- Registered by server and maintained by cMSO
- 2-Level MIOs
  - Sever Level MIO: general information about server
  - Object Level MIO: specific information about each object
- Functions
  - Allowing MSO to access mgmt information
  - Allowing MSO to update mgmt information if necessary
  - Reporting events or faults to MSO
## CDS MIB

<table>
<thead>
<tr>
<th>Group</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Information Group</td>
<td>Server Name, Host Name, Executable, Owner, Protocol, Version, Last Update Time, Last Start Time, Status, Platform, Language</td>
</tr>
<tr>
<td>Operational Statistics Group</td>
<td># of Current In-bound Associations, Requests, Errors, Received/Sent Bytes</td>
</tr>
<tr>
<td>Object Information Group</td>
<td># of Objects, ObjMIO Reference Table</td>
</tr>
<tr>
<td>Resource Information Group</td>
<td>CPU Utilization, Memory Space, # of Child Processes, Threads, IPCs</td>
</tr>
</tbody>
</table>

- Provided through MIO interfaces
- Gathered from OS, ORB, and SO by MIO
- Can be specialized using the inheritance feature of the object-oriented technique
Mgmt Service Architecture

Management Application

mgmt requests

replies

MSOs

cMSO

fMSO

sMSO

pMSO

Event Channel Object

requests

ev event flow

server Level MIO

Application Interface
Configuration Mgmt Service

• cMSO is responsible for managing containment, relationships among servers, objects and clients, and attributes of each server and each object.

• Functions
  – Finding servers in a management domain
  – Providing a server’s current status
  – Providing information of servers
  – Changing information of servers
  – Initializing or terminating the operation of a server
  – Providing relationships between servers and clients
Object Model for MIO & cMSO

BasicMIO

MIO
server
has
ObjMIO
elements

Application Service Object

MIO

server_id

manage

cMSO

server_id

ObjMIO

Event Service

- Adopted the OMG CORBA Event Service Specification
- Every object can act as supplier/consumer by including supplier/consumer objects.
- Event Channel creates proxies and manages proxy supplier’s event queue lists.
- Incoming events are multicasted through event queue lists.
Object Model for Event Service

- EventChannel
  - Consumer Admin
  - Supplier Admin
  - ProxyPull Supplier
  - ProxyPush Consumer

- Pull Supplier
- Push Consumer

- Pull Consumer
- Push Supplier

Relationships:
- Consumer Admin creates Supplier Admin
- Supplier Admin creates ProxyPull Supplier
- Supplier Admin creates ProxyPush Consumer
- ProxyPull Supplier connects Pull Consumer
- ProxyPush Consumer connects Push Supplier
Fault Mgmt Service

• fMSO is responsible for providing a reliable service environment by handling events and faults.

• Functions
  – Providing logging/reporting event filter objects
    • Allowing users to set scope of events and information logged or notified, and name of a logging file
    • Allowing users to start and stop logging or notifying process
  – Fault Diagnosis
    • Analyzing events and detects failures in servers
    • Localizing the causes of the failure
    • Notifying the suspicious causes of the failure
Object Model for fMSO

- **fMSO** creates **Efilter**
  - **Efilter** has **ELfilter** and **ERfilter**
  - **Pull Consumer**
Performance Mgmt Service

- pMSO is responsible for providing an efficient service environment by monitoring the behaviors of servers.
- Functions
  - Providing performance related information
    - Average response time, requests, errors, throughput, bytes
    - Minimum, Maximum, Current etc
  - Logging performance data
  - Setting performance thresholds
  - Notifying when a dangerous condition happens
Security Mgmt Service

• sMSO is responsible for providing a secure environment by controlling the access on services and information.

• Functions
  – Authentication
    • Checking valid users of the management system (adding user, deleting user, login, unlog)
  – Authorization
    • Confining access on services and information to authorized users (setting ACL, getting ACL)
Prototype Implementation Architecture

Web Browser

Management API

Web Server

OrbixWeb

CDS

MIB

MSOs

Naming

Storage/Retrieval

Communication

Session

Remote MAESTRO Services

MAESTRO Services

APIs
Implementation Environment

- **Management Servers**
  - Orbix 2.2 & Sparc C++ 4.0
  - Solaris 2.4, 2.5

- **Management Application**
  - written in Java
  - OrbixWeb 2.0.1
  - Web-based Interfaces
  - Platform independent management application
Configuration Mgmt Tool
Event & Fault Mgmt Tool
Performance & Security Mgmt Tool
Conclusion & Future Work

• Management Framework for CORBA-based Distributed Services and Applications
  – CDS MIB, MIO, MSOs, MA

• Our work can be easily extended to manage other CORBA-based services & applications.

• We developed a prototype mgmt system.

• Future Work
  – Integrated web-based management
    • SNMP, CMIP, DMI gateways
  – Management of OMA Components
  – Management of other CORBA-based applications