Web-based Internet/Intranet Service Management
with QoS Support

Jong-Tae Park and Jong-Wook Baek
School of Electronic and Electrical Engineering
Kyungpook National University
1370, Sankyuk-Dong, Buk-Gu
Taegu, Korea 702-701,
Tel: +82-53-950-5543
Fax: +82-53-950-5505
park@ee.kyungpook.ac.kr

James Won-Ki Hong
Dept. of Computer Science and Engineering
POSTECH
jwkhong@postech.ac.kr

Abstract

There exist various application services such as e-mail, file transfer, news, telnet, archie, gopher and WWW on the Internet. New application services such as Web hosting, entertainment and electronic commerce are continuously being developed and deployed over an enterprise network infrastructure that spans multiple control domains. These services must be managed and controlled in order to provide secure, reliable and efficient services to users. In this paper, we present the design and implementation of a Web-based Internet/Intranet service management system with Quality of Service (QoS) support. We analyze the requirements of Internet service management and various Management Information Bases (MIBs). In particular, we have defined the QoS parameters for the management of Internet services and developed a new WWW service MIBs for QoS management. The specific functions for the management of Internet services have also been designed and implemented. The support of QoS management is extended into a multi-domain environment using the Service Level Agreement (SLA) concept. SLA among multiple control domains is measured, monitored and verified. We have built a prototype system in order to verify the functionality and performance of the proposed Web-based Internet service management system. The prototype system utilizes Java and CORBA technologies, so that various benefits of these technologies such as platform independence and scalability can be obtained.

[Keywords: Internet/Intranet Service Management, Enterprise Network Management, Web-based Management, QoS, SLA, CORBA, Java]
1. Introduction

There exist various Internet applications such as e-mail, file transfer, news, telnet, archie, gopher, World-Wide Web (WWW or Web) and so on. Internet services have been applied to improve the productivity of many organizations. The multimedia Internet services of Internet Phone, whiteboard, RealAudio and RealVideo are being widely used by ordinary users. Electronic commerce (EC), one of the new Internet services, is spreading rapidly throughout the world as well.

The need for a management system that can effectively manage various new application services and multimedia services on top of existing services has recently increases [Hon97]. Users ask for more reliable and efficient services on the Internet and Intranet. Effective management of Internet and/or Intranet services is needed to satisfy these requirements. However, the management of these services is not trivial because Internet services have diverse characteristics and they are sensitive to many network infrastructures.

Internet services such as Web hosting, EC, application hosting are being deployed over an enterprise network infrastructure that spans multiple control domains. These services require cooperation and internetworking among multiple organizations. Inter-domain management is critical to make these to be efficiently and reliably supported.

In this paper, we present the design and implementation of a Web-based Internet and Intranet service management system with Quality of Service (QoS) support. We analyze the requirements of Internet/Intranet service management and various Management Information Bases (MIBs). In particular, we have specified the QoS parameters for the management of Internet/Intranet services and developed a new WWW service MIBs for QoS management. The specific functions for the management of Internet/Intranet services have also been designed and implemented. The support for QoS management is extended into a multi-domain environment using the Service Level Agreement (SLA) [Nmf96] concept. SLA among multiple control domains is measured, monitored and verified. We have built a prototype system in order to verify the functionality and performance of the proposed Web-based Internet/Intranet service management system. The prototype system can not only monitor the performance of Internet/Intranet services and QoS delivered to users, but also modify the configuration of Internet/Intranet services.

We have applied Web technology for the management of Internet/Intranet services. As the proposed management system uses Web technology, it can provide platform independence, mobility and user-friendly interface. Using CORBA technology [Obj95], the architecture can offer security, portability and scalability. It can also support multiple network management protocols and allow easy integration of new Internet service management applications and the existing management applications. The Java technology is used to provide the user interface for Web-based Internet service management. Since most of the Internet services are used for Intranet, we will, for simplicity, use the term Internet services for Internet/Intranet services. When it is necessary to distinguish between the two concepts, we will differentiate explicitly.

This paper is organized as follows. In Section 2, we describe related work. We specify the requirements for the Internet/Intranet service management in Section 3, and we extend the existing MIBs by introducing various QoS concept in Section 4. In Section 5, we present the architecture of a Web-based Internet/Intranet service management system using CORBA and Java technologies. In Section 6, we describe a prototype implementation. Finally, we summarize the paper in Section 7.
2. Related Work

There are two main approaches related to Web-based management, Web-based Enterprise Management (WBEM) [Wbe96] and Java Management API (JMAPI) [Mic96]. The goal of WBEM is to develop industry standards that will allow administrators to use any Web browser to manage disparate networks, systems, and applications. WBEM describes an architecture, a management protocol, a management schema, and an object manager. Though WBEM is designed to address the failings of current management applications, it embraces existing management standards and protocols, allowing the integration of the distributed management services provided by different management platforms. The proposed standards integrate network and system management through the use of the Web technology without affecting the existing network infrastructure. JMAPI is a set of extensible objects and methods for the development of seamless system, network, and service management solutions for heterogeneous networks. This core set of application programming interfaces can be used across a diverse array of computing environments involving numerous operating systems, architectures, and network protocols, enabling the development of low maintenance, heterogeneous software from a single source. These Web-based network management approaches [Jon97] compared with the traditional network management [Cas90] have many advantages such as platform independence, mobility and user-friendly interface.

There have been a few research work on Internet services management. The UT-WWW project [Har96], sponsored by the Joint Research Center of the European Commission, investigated management of web-servers. The project designed three new MIBs that were presented as internet-drafts to Internet Engineering Task Force (IETF). These MIBs are the HTTP-MIB [Haz96], the Retrieval Service MIB [Haz96] and the Information Store MIB [Hen96]. The prototype of the agent system for the management of Web server has also been developed. The DESIRE project [Car96] developed a system for remote management of information servers, allowing their performance and availability to be monitored using standard network management protocols. Kong [Kon98] examined the issues of Internet service management and compared it with the service management approaches taken by the telecommunication industry. While the above mentioned research work offer a good start towards Internet service management, none of the research addresses the management framework considering multiple control domain environments. More efforts are required to achieve a standard Internet service management strategy to manage all types of Internet services.

Network Management Forum (NMF) has defined a set of common business processes for the management of telecommunication services [Nmf97]. The QoS Team in NMF has been working on the automation of the interface between service provider and customer for performance reporting with the Service Level Agreement (SLA) concept [Nmf98]. They have identified common terms and definitions, and created an industry-wide glossary for performance measurement and reporting. The work concentrates on the definition of service availability that is the key performance parameter in today's telecommunications world. There are a few products that allow customers to monitor the quality of service offered by service providers. These include Netcool [Mic98], InfoVista [Inf98], and VitalAnalysis [Vit98]. All of these work on telecommunication service management can be related to the management of Internet services.

The work by the Integrated Services Working Group of IETF has made a significant contribution to providing controlled QoS for multimedia applications over the Internet. The Group has defined a comprehensive integrated service architecture [Bra94] and a QoS framework [She97] for specifying the functionality of the Internet system elements which could
make multiple, dynamically selectable QoS available to applications. The Internet Protocol Performance Working Group of IETF has been working on the identification of Internet service metrics and measurement methodologies.

In addition to these, there have appeared several MIBs related to the management of Internet services, proposed by IETF. These include Host Resources MIB [Kru95], Network Services Monitoring MIB [Kil94], Application MIB, WWW Service MIB [Haz98]. The Host Resources MIB defines a uniform set of objects useful for the management of host computers independent of the operating system, network services, or any software application. The system application MIB has been defined to represent installed and running applications and their components. The WWW Service MIB defines a set of objects for managing WWW service.

3. Requirements of Internet/Intranet Service Management

3.1 User Requirements

Internet/Intranet service administrators require integrated monitoring and configuration tools for performance, fault, security and accounting management. Internet/Intranet service end-users also desire to have access to performance statistics and be notified of causes for service failures. The management system needs to produce output not only for Internet service administrators, but also for non-specialized Internet service end-users.

Visualizing the management information that might include QoS parameters and the result of management function execution is of great importance. The end-user would like to access the management information from anywhere on the Internet. Web-based management technology is suitable to satisfy these requirements.

3.2 Functional Requirements

Monitoring and configuration are two main management functional areas in Internet/Intranet service management. Monitoring focuses on performance, configuration, fault, security and access statistics. On the other hand, configuration functions provide the means for recovering from an error or preventing the fault or the degradation of performance. The management functional requirements are summarized below.

- Providing an up-to-date view of service configuration, identifying changes and discovering configuration automatically
- Monitoring service performance and utilization, which include system resource utilization, service throughput & delays, error rates, system and service availability, service utilization accessibility
- Maintaining historical performance data for planning purposes
- Monitoring and analysis of errors and faults
- Monitoring of server security mechanisms and access control
- Gathering published information access statistics. This can be used to analyze user access patterns and improve QoS
- Checking the consistency of published information
- Remote configuration of the server parameters, including run-time, access control and limitation parameters.
- Change the service entity running status, in order to recover from a crash or to change the server configuration
3.3 Generic QoS Parameters for Internet/Intranet Services

We derive a set of QoS parameters that are classified into three layers: the network, the system and the application. Network level parameters provide estimates for the network status and health. System level parameters provide system health information that could affect the performance and reliability of Internet services. Application level parameters provide the security, access, configuration, current status and resource utilization of the Internet service entity and dependent components. Some of the application level parameters for Internet service management are described below.

- **Number of connections** - This is a single value parameter that is the number of the current connections for the Internet services entity. It can be calculated as the sum of inbound associations and outbound associations:
  \[(Inbound\ Association) + (Outbound\ Association)\]

- **Internet service component load** - It is the number of the current connections divided by the maximum number of connections allowed for the Internet service component, whose formula is as follows:
  \[\frac{(Inbound\ Association) + (Outbound\ Association)}{Maximum\ Connection}\]

- **Internet service entity status** - This indicates the current status of an Internet services entity. This is useful to both the manager and the end-user. Possible values are as follows: *up, down, halted, congested, restarting, unknown.*

- **Service availability** - This is an indicator for service availability and meaning downtime in minutes (or hours) per week (or month). Downtime is the duration of service unavailability due to system or Internet services entity faults. This is calculated according to the following simple formula:
  \[(System\ up\ Time) \div (Application\ Last\ Change)\]

- **Fault frequency** - This indicates the faulty behavior of the Internet services entity and is the number of errors per day (or weekly, monthly, annually). This is calculated according to the following formula:
  \[\frac{Error\ Count\ Number}{(System\ Up\ Time) \div (Error\ Count\ Since)}\]

- **Reliability** - This indicates the number of errors per total number of requests and responses generated or received by the Internet services entity. This is calculated as the following formula:
  \[\frac{(Summary\ Request\ Errors) + (Summary\ Response\ Errors)}{(Summary\ Requests) + (Summary\ Responses)}\]

- **Delay** - This indicates delay between service entity or dependent application.

- **Internet service entity error table** - This table contains request / response errors and discards / unknown messages handled by the Internet services entity.

- **Internet service entity throughput table** - This table shows the throughput values for request / response errors and discards / unknown messages handled by the Internet services entity. The throughput is calculated through a sliding-window mechanism:
  \[Average\ of\ N\ throughput\ values\ X(t), X(t+s), ..., X(t+N*s),\]
  \[where\ X(t) = \frac{V(t) - V(t-s)}{s},\ V = MIB\ counter,\ s = sampling\ period.\]
Document last access statistics - This includes an average time and variance between subsequent accesses for a document of an Internet services entity. These two parameters indicate the frequency of documents being accessed for a specific Internet services entity and the diversity of accessed documents. A dual value indicator are calculated as follows: 
Average of [(Document Last Access Time) ? (current Time)], and 
Variance of [(Document Last Access Time) ? (current Time)]

Document accesses - The total number of accesses per document and number of accesses since last update for top N documents. The manager who constructs the table defines the number N.

User/domain accesses - These indicate statistics on N recent users and domains accessing the Internet services entity and top N users and domains with a configurable N.

Figure 1. Structure of Extended WWW Service MIB

4. Extensions to WWW Service MIB

Driven by the analysis of Internet services and its management requirements, we have formulated a set of QoS parameters described in Section 3.2. WWW Service MIB [Haz96] is missing these QoS parameters such as the throughput of Internet service entities, the error of Internet service entities and the delay between Internet service entities. We extend WWW Service MIB to satisfy the requirements of Internet service management described in Section 3.1. Extended WWW Service MIB consists of WWW service information, WWW protocol
statistics, WWW document statistics, WWW access statistics, WWW error statistics, and WWW QoS. We have added the QoS group, access statistics group and error statistics group. Figure 1 shows the structure of extended WWW Service MIB. Newly defined groups are described below in more detail.

4.1 WWW QoS Group

We have defined the WWW Service MIB for QoS management. The QoS group which is shown in Table 1 contains network management information about the quality of the retrieval service. This group consists of wwwQoSDelayTable and wwwQoSThroughputTable. The wwwQoSDelayTable has information for the delay between source (WWW server) and destination (WWW client). The wwwQoSThroughputTable has information related to the throughput with a specific client.

<table>
<thead>
<tr>
<th>MIB variable name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>wwwQoSDelayTable</td>
<td></td>
</tr>
<tr>
<td>wwwDelaySource</td>
<td>The DNS name of the source</td>
</tr>
<tr>
<td>wwwDelayDestination</td>
<td>The DNS name of the destination</td>
</tr>
<tr>
<td>wwwDelayTime</td>
<td>The delay which occurred during transport of information from source to destination</td>
</tr>
<tr>
<td>wwwQoSThroughputTable</td>
<td></td>
</tr>
<tr>
<td>wwwThroughputClient</td>
<td>The DNS name of the client</td>
</tr>
<tr>
<td>wwwThroughput</td>
<td>The throughput of data with the client</td>
</tr>
</tbody>
</table>

Table 1. WWW QoS Group

4.2 WWW Service Information Group

The WWW service information group contains information about the WWW services known by the SNMP agent. We newly define the wwwVirtualHostTable that contains information about the virtual host configuration. Table 2 shows the WWW service information group.

<table>
<thead>
<tr>
<th>MIB variable name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>wwwVirtualHostTable</td>
<td></td>
</tr>
<tr>
<td>wwwVirtualHostIndex</td>
<td>An integer used to uniquely identify a virtual host</td>
</tr>
<tr>
<td>wwwVirtualHostStatus</td>
<td>Indicates the status of the virtual host</td>
</tr>
<tr>
<td>wwwOriginalAddress</td>
<td>The original address of the virtual host</td>
</tr>
</tbody>
</table>

Table 2. WWW Service Information Group

4.3 WWW Access Statistics Group

The WWW access statistics group contains statistics about accessing users. The group consists of wwwUserAccessCtrlTable, wwwUserAccessLastNTable, and wwwUserAccessTopNTable. The wwwUserAccessCtrlTable provides the administrator a means to limit the access statistics tables in size. The wwwUserAccessLastNTable provides the manager information about the users of last N accesses. The number of accesses in the wwwUserAccessLastNTable is controlled by the wwwUserAccessCtrlLastNSize in the wwwUserAccessCtrlTable. Table 3 shows the MIB variable names with associated definition of the WWW access statistics group.
### Table 3. WWW Access statistics group

<table>
<thead>
<tr>
<th>MIB variable name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>wwwUserAccessCtrlTable</td>
<td></td>
</tr>
<tr>
<td>wwwUserAccessCtrlLastNSize</td>
<td>The maximum number of entries in the wwwUserAccessCtrlLastNTable</td>
</tr>
<tr>
<td>wwwUserAccessCtrlLastNStatus</td>
<td>Allow a manager to suspend the update process in order to retrieve the wwwUserAccessLastNTable</td>
</tr>
<tr>
<td>wwwUserAccessCtrlTopNSize</td>
<td>The maximum number of entries shown in the wwwUserAccessTopNTable</td>
</tr>
<tr>
<td>wwwUserAccessLastNTable</td>
<td></td>
</tr>
<tr>
<td>wwwUserAccessLastNIndex</td>
<td>The index for wwwUserAccessLastNTable</td>
</tr>
<tr>
<td>wwwUserAccessLastNName</td>
<td>The name of user of last N access</td>
</tr>
<tr>
<td>wwwUserAccessLastNTime</td>
<td>The last time the user has accessed this server</td>
</tr>
<tr>
<td>wwwUserAccessLastNBytes</td>
<td>The number of content bytes of document that last user accesses</td>
</tr>
<tr>
<td>wwwUserAccessTopNTable</td>
<td></td>
</tr>
<tr>
<td>wwwUserAccessTopNIndex</td>
<td>The index for wwwUserAccessTopNTable</td>
</tr>
<tr>
<td>wwwUserAccessTopNName</td>
<td>The name of user of top N access</td>
</tr>
<tr>
<td>wwwUserAccessTopNCount</td>
<td>The number of accesses made by the user on this table</td>
</tr>
<tr>
<td>wwwUserAccessTopNBytes</td>
<td>The number of content bytes of document that user accesses</td>
</tr>
</tbody>
</table>

### 4.4 WWW Error Statistics Group

The WWW error statistics group contains statistics about errors that occurred during accesses of request or response. This group abstracts the network errors and the errors resulting from information access. Table 4 shows MIB variable names with associated definition for WWW error statistics group.

<table>
<thead>
<tr>
<th>MIB variable name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>wwwErrorTable</td>
<td></td>
</tr>
<tr>
<td>wwwErrorIndex</td>
<td>The index indicating the error</td>
</tr>
<tr>
<td>wwwErrorDescription</td>
<td>The textual description of the error</td>
</tr>
<tr>
<td>wwwErrorCount</td>
<td>The number of errors that occurred</td>
</tr>
<tr>
<td>wwwErrorLastTime</td>
<td>The time of the last error occurred</td>
</tr>
</tbody>
</table>

### Table 4. WWW Error Statistics Group

### 5. Design of Web-based Internet/Intranet Service Management System

In this section, we present the design of a Web-based Internet/Intranet service management system. First, we present an overall management architecture. We then describe the manager and managed system architecture. Finally, we present the design of an SLA management system.

#### 5.1 Web-based Management System Architecture for Internet/Intranet Service

The Web-based Internet service management system consists of Web server, service managing system, SLA managing system and managed system. The service managing system is composed of management application, SNMP manager, and gateway. SNMP manager communicates with agent that collects or changes information and remotely controls the resources. Gateway plays the role of message conversion between management applications and Web browser.

The SLA managing system consists of a contract manager, customer SLA repository, service
model and SLA manager. Contract Manager is responsible for contracting a service level agreement among multiple control domains. Service Model includes various components that enable a service, such as Post Office Protocol (POP) server, IP address, name server and dependencies that exist among the different elements of the service. SLA Manager is responsible for verifying a service level agreement between customer and service provider. Figure 2 shows the architecture of our Web-based Internet/Intranet service management system.

Figure 2. Architecture of Web-based Internet/Intranet Service Management System

Web server is needed for downloading Java applets, which are executed on the Web browser. Java applets allow the service manager to perform management functions through a Web browser. They also allow the end-users to access performance statistics and be notified of causes for service failures. Java applets communicate with Gateway through CORBA ORB. As CORBA ORB handles all requests from Web browser to Gateway, it provides transparency of information exchange and makes an easy integration of new Internet service management applications with the existing management applications, providing effective extensibility of management applications. Java applets also allow the Contract Manager to interact with the customer and SLA manager to verify the SLA between the customer and service provider through a Web browser. The subsequent sections explain the components of the architecture in more detail.

5.2 Service Management System

5.2.1 SNMP Manager

Figure 3 shows the architecture of SNMP Manager. SNMP Manager is composed of Manager and Trap Handler. Manager supports management functions and collects the managed information from agent. Trap Handler forwards event reports from agent to Message Handler in Gateway.
5.2.2 Gateway

Figure 4 shows the architecture of Gateway which is composed of Message Handler and Service Handler. Gateway communicates with Java applets in Web Browser through CORBA ORB. Message Handler coordinates the requests from Java applets to various Managers and Trap Handler. It parses the request and forwards it to the appropriate destination. Service Handler controls and maintains the Internet service defined by management systems. The procedure for providing Internet application service management is as follows.

1. First, Web Browser tries to connect the Web Server in Management System.
2. After connection, Web server sends HTML document to Web browser and Java applets are downloaded.
4. Java applets then communicate with the Message Handler in the Gateway.
5. Agent performs management using operation primitives (such as Get or Set).
6. Service Handler communicates with management application service module and provides the management service.

![Figure 4. Architecture of Gateway](image-url)
Figure 5 shows the message flows and connection procedures between Web Browser, Managing System and Managed System for Get, Set and Trap operations.

Table 5 shows an example of collecting and controlling the information of operational status of a WWW Service (wwwServiceOperStatus). The message of ‘running’ indicates that the service is operational and available. Manager may have down, halted, congested or restarting as values of wwwServiceOperStatus.

<table>
<thead>
<tr>
<th>The management information in Applet</th>
<th>The management information in Web-based managing system</th>
</tr>
</thead>
<tbody>
<tr>
<td># MIB OID</td>
<td># MIB OID</td>
</tr>
<tr>
<td>mib_oid = wwwServiceOperStatus</td>
<td>mib_oid = wwwServiceTable</td>
</tr>
<tr>
<td># set wwwServiceOperStatus to running</td>
<td># set wwwServiceOperStatus to running</td>
</tr>
<tr>
<td>set_value = running</td>
<td>set_value = running</td>
</tr>
<tr>
<td># The period of Monitoring service</td>
<td># The period of Monitoring service</td>
</tr>
<tr>
<td>period = NULL</td>
<td>period = NULL</td>
</tr>
<tr>
<td># After Set operation, the value is</td>
<td># After Set operation, the value is</td>
</tr>
<tr>
<td>nm_information = running</td>
<td>nm_information = running</td>
</tr>
</tbody>
</table>

Table 5. Management Messages in Applet and Web-based Managing System
5.2.3 Internet/Intranet Service Management Functions

Internet/Intranet Service management system provides manager with Internet/Intranet service management based on collected result of management information. Figure 6 illustrates the structure that Web browser gets the management service through network management service interface.

![Diagram of Web-based Internet Service Management System](image)

Figure 6. Web-based Internet Service

The management functions for Internet/Intranet service management are briefly described below.

- **MIB Browser**: User accesses MIB by using browser interface. MIB browser shows the management information about Internet service management.
- **Document Usage Monitoring**: This module shows the document statistics with graph.
- **Request and Response Monitoring**: This module shows the request and response of Internet/Intranet services.
- **The performance of system and application server**: This includes service throughput, error and timeout rate, service availability, server and system down time, system resource utilization and others.
- **Server errors and faulty conditions**: This module monitors and tries to analyze the errors and faulty condition of the system and servers. Errors could be categorized according to temporal parameters or service types and be used to identify error patterns, sources and causes.
- **Server security**: The manager is able to monitor server security at the system on the information level.
- **Access statistics**: This module offers to the manager access statistics for published information.
- **Quality of service delivered to users**: The manager monitors various QoS metrics to record user satisfaction.
- **User access pattern**: The manager monitors the user accesses to the servers to identify user access patterns for providing input to planning procedures which includes caching policy, bandwidth planning, etc.
Configuration of the server parameters: The manager reads or modifies server configuration parameters through the agent.

Server process operational status: The manager may kill or restart a server process.

5.3 Managed System

Figure 7 shows the architecture of managed system. This managed system is composed of Multi-agent Handler and two agents, which are WWW Service Agent and MIB-II Agent. Each agent has management information about related real resource. We use UNIX system call and log file to build management information.

Multi-agent Handler has the access information of agents, such as UDP port number, OID (Object Identifier), community and so on. Multi-agent Handler forwards Trap message received from agent to Trap Handler in Web-based Managing System. Diverse agents can be integrated into the Managed System through Multi-agent Handler.

5.4 SLA Management System

The SLA concept provides the realization of QoS support mechanism in multi-domain environments. This can be used to measure, monitor and verify a service level agreement among multi-domain environments. An SLA management system is composed of Contract Manager, SLA Manager and Customer SLA Repository.

5.4.1 Contract Manager

Contract Manager is responsible for contracting a service level agreement among multiple control domains. It analyzes a contract data and a service model and generates customer/service/system dependent information. This information is stored in a Customer SLA Repository. A SLA includes the feature of the service, the expected behavior of the service and the parameters of quality of service. Typically, a SLA includes the following information.

System and service availability
Time to identify the cause of a customer reported problem
Time to repair a customer reported fault
Provisioning-related time
- Tracking delivered service level quality against service level agreement
- Linking the performance monitoring and reporting to other service management processes, such as billing and trouble ticketing management
- Service capability to be increased or modified
- Reporting at regular intervals

Figure 8 shows the procedure of SLA data generation.

Figure 8. Customer SLA Data Generation

A description method of contract contents is necessary to treat the result of contracts efficiently. We have developed a simple Contract Description Language (CDL) [Bae98] for defining service contracts. CDL is a simple, declarative language and has syntax similar to C++.

Figure 9 shows a formal description of the CDL grammar.

```
<contract> ::= <declaration list> {<party Information>} {<assertion set>}
<declaration list> ::= ContractName <identifier> ;
                   ServiceName <identifier> ;
                   PartyInformationName <identifier> ;
                   AssertionName <identifier> ;
<identifier> ::= <letter> {<letter> | <digit>}
<letter> ::= a | b | c | y | z | A | B | C | Y | Z
<digit> ::= 0 | 1 | 2 | 3 | 8 | 9
<party information> ::= PartyInfo <identifier> ;
                       person <identifier> ;
                       organization <identifier> ;
                       [phone <identifier>] ;
<assertion set> ::= Assertion <identifier> ;
                   {<assertion>} ;
<assertion> ::= <identifier> ; "(,<expression>[<relational operator><number><unit>])"
<expression> ::= <identifier> [<relational operator><number><unit>]
<relational operator> ::= > | < | = | <= | == | !=
<number> ::= {digit}[ , {digit}]
<unit> ::= % | sec
```

Figure 9. Formal Description of CDL Syntax
The details on CDL are out of the scope of this paper, but can be obtained by referring to the paper [Bae98]. Figure 10 shows a part of CDL to describe a Web hosting service.

```
ContractName exampleContract;
ServiceName webHosting;
PartyInfo customer, provider;
Assertion dialupServer;
Assertion webServer;

PartyInfo Customer {
    Person Jong-Wook;
    Organization company_A;
}
PartyInfo Provider {
    Person Jeong-Hwan;
    Organization ISP_1;
}
Assertion DialupServer {
    DialupServerAvailability ( dialupSrv1 > 90%, dialupSrv2 > 95%); }
Assertion WebServer {
    WebServerAvailability ( webSrv1, > 95%);
    WebResponseTime ( webSrv1, > 3sec );
    ReportAccessingHost ( daily, weekly, monthly);
    ReportHtmlStatistics ( daily, weekly, monthly );
}
```

Figure 10. CDL Example for Web Hosting Service

5.4.2 SLA Manager

Figure 10 shows the architecture of SLA manager. Customers are able to verify a SLA through SLA verification interface.

![Architecture of SLA Manager](image)

Figure 11. Architecture of SLA Manager
When a customer requests to verify an SLA, SLA manager authenticates the customer using authentication information in Customer SLA Repository. If the customer is identified as a legal user, SLA manager dynamically loads a proper plug-in program to perform the customer's request. The plug-in programs can gather data from local management system or other domains, and evaluate the SLA. After the evaluation of the SLA, SLA manager reports the results to the customer.

6. Prototype Implementation

We have implemented a prototype system in order to verify the functionality and performance of the proposed architecture. Service managing system and SLA managing system have been implemented by using JDK1.1 and Orbix 2.3c [Ion98], and agents implemented by using Tcl/Tk8.0 and Scotty package 2.1.5 [Sco98].

Figure 12 shows MIB Tree and MIB Browser of WWW MIB. User can access the MIB by using MIB browser interface. MIB browser shows the management information about Internet service management. In Figure 12, MIB Tree shows the structure of WWW MIB. OID is \texttt{wwwServiceOperStatus}, Host is \texttt{ain.kyungpook.ac.kr}, Version is \texttt{SNMPv2c} and Community is \texttt{public}. If the Description button is clicked, MIB Browser will show the information of \texttt{wwwServiceOperStatus}.

Figure 13 shows the statistics of WWW service request. Figure 14 shows the access statistics of WWW documents. OID is \texttt{wwwSummaryInRequests.1}, Host is \texttt{ain.kyungpook.ac.kr}, graph color is \texttt{Black} and Scaling is \texttt{Auto Scale}. Click the \texttt{Add Graph} button, then graph shows the WWW Request statistics of \texttt{wwwSummaryInRequests.1}.
Figure 13. Implementation of WWW Request Statistics

Figure 14. Implementation of WWW Document Statistics
7. Conclusion

In this paper, we have presented the design and implementation of a Web-based Internet and InTRANet service management system. We developed a new, extended WWW service MIB through the analysis of requirements, QoS parameters for the management of Internet/InTRANet services. We have also described our effort on the prototype implementation and demonstrated how to manage WWW server. The architecture can be used not only for measuring and monitoring the service reliability, performance and QoS of Internet/InTRANet services, but also for managing a service level agreement between a customer and service provider. Furthermore, it will ensure an early diagnosis of faults or lack of performance through the collection of information related to the usage of various resources. We have applied WWW technology to provide mobility, platform independence and useful user interface. In particular, we have applied CORBA technology integrated with Java to provide scalable, distributed, and extensible management of legacy system as well as new Internet services.

For future work, we plan to analyze the performance of our system and apply the same management architecture and techniques to other types of Internet/InTRANet Services. We envision that this should be fairly simple since the groundwork that is the most difficult part has been already done.

References


