

웹 기반의 인트라넷 관리 보고서 작성기능 설계 및 구현

정연기[†] · 이상동^{**}

요 약

본 논문은 인트라넷 관리자가 웹 환경에서 인트라넷의 상황을 쉽게 분석할 수 있도록, 관리자의 요구에 따라 일별, 주별, 월별, 연도별로 인트라넷의 구성관리, 성능관리, 장애관리에 대한 현황을 일괄적으로(또는 선택적으로), 자동으로 생성되게 하는 방법에 관한 것이다. 본 연구의 특징은 웹을 기반으로 하여 분산 환경에서 개방형 통신망(TINA) 체계의 인트라넷 관리 보고서를 자동 생성할 수 있다는 것이다. 즉 코바(CORBA) 환경에서 사용자 에이전트(User Agent)에만 접속할 수 있게 함으로써 인트라넷 관리 시스템을 보호하고, 관리 시스템 접속 부분의 프로그램에 변경이 생기더라도 사용자 에이전트 프로그램만 수정하면 되므로 유지보수가 용이한 구조이다.

Design and Implementation of Web Based Intranet Management Report Publisher

Youn-Ky Chung[†] and Sang-Dong Lee^{**}

ABSTRACT

In this paper, we design and implement web based Intranet Management Report Publisher (IMRP) that generates automatically daily, weekly, monthly and yearly report for configuration, performance and fault management of the Intranet. Intranet managers can analyze status of the Intranet with IMRP easier. The IMRP publishes the Intranet management reports based on TINA (Telecommunication Information Networking Architecture) using CORBA(Common Object Request Broker Architecture). The advantages of this architecture are to protect the Intranet management system by being connected to the User Agent program and to raise the efficiency of maintenance by modifying only the User Agent program as accessing part of the Intranet management system is needed to change.

Key words: 인트라넷 관리 보고서, CORBA, IDL 인터페이스, 분산 환경, 구성관리, 성능관리, 장애관리

1. 서 론

통신망이 복잡해짐에 따라 다양한 업체들이 제공하는 이기종 통신망요소(Network Elements) 장치들을 통합하는 인트라넷 관리 체계가 요구되었다. TINA-C (TINA-Consortium)는 이기종 망 관리 기술들을 통합하기 위해서 ODP(Open Distributed Processing) 분산 환경 기반의 TINA (Telecommunications Information Networking Architecture) 체계를 제안하고 그 표준

화 작업을 진행하였다[1-5]. TINA 체계의 ODP 분산 환경은 일반적으로 OMG(Object Management Group)에서 표준화하는 CORBA(Common Object Request Broker Architecture)를 기반으로 하고 있으며, CORBA는 이기종 기술들의 상호연동 기능의 복잡성을 줄이고 분산처리의 장점을 가질 수 있게 한다[6,7].

TINA 체계의 인트라넷 관리 시스템에서 발생하는 정보는 그 양이 많고 복잡하기 때문에 인트라넷 관리 상태를 분석하기 위해 주기적으로 보고서를 만드는 작업은 시간이 많이 소요되고, 편집 실수로 인한 오류가 발생될 수 있다. 이러한 불편을 해소하기 위해 TINA 체계의 인트라넷 관리 시스템이 수집한

접수일 : 2001년 12월 31일, 완료일 : 2002년 6월 21일

[†] 중신회원, 경일대학교 IT대학 컴퓨터공학과 교수

^{**} 준회원, 밀양대학교 학술정보센터 근무

방대한 자료로부터 관리자의 요구에 따라 웹 기반으로, 주어진 특정 기간에 대한 인트라넷 관리 보고서를 자동으로 작성해 주는 기능이 필요하다.

이에 따라 본 논문에서는 웹 서버, 인트라넷 관리 시스템, 데이터베이스 시스템이 분산되어 있는 TINA 체계의 인트라넷 관리 시스템으로부터 CORBA와 자바(Java)의 서블릿(Servlet) 기능을 이용하여 웹 기반으로 인트라넷 관리 보고서를 자동으로 생성하는 기능(이하 IMRP: Intranet Management Report Publisher라 칭함)을 설계·구현하였다. IMRP는 인트라넷 관리자가 웹 환경에서 인트라넷의 상황을 쉽게 분석할 수 있도록, 관리자의 요구에 따라 일별, 주별, 월별, 연도별로 인트라넷의 구성관리, 성능관리, 장애관리에 대한 보고서를 일괄적으로(또는 선택적으로), 자동 생성하는 기능을 갖는다.

휴렛팩커드(Hewlett Packard)사의 망관리 시스템(HP Open View Network Node Manager I)도 망관리 보고서 생성 기능을 갖고 있지만[8], IMRP가 분산 환경에서 망관리 보고서 생성 기능을 제공하는데 비해 휴렛팩커드사의 것은 분산 환경에서 망 관리 기능을 제공하지 못 한다. 또한 휴렛팩커드사의 망관리 시스템의 보고서 형태는 이용률 보고서, 예외 보고서, 토폴로지 보고서, 성능 보고서 등으로 IMRP에 비해 다양한 망관리 현황을 제공하지 못 하는 차이점이 있다. 또한 휴렛팩커드사의 망관리 시스템은 중앙 집중식 시스템으로 규모면에서 관리 할 수 있는 최대 노드 수가 제한되기 때문에 다량의 망장비들로 구성된 네트워크를 관리하기에는 한계가 있으며, SNMP(Simple Network Management Protocol)를 이용하여 망 구성정보 및 성능정보를 수집하므로 SNMP가 탑재된 망장비들에 대해서만 관리 기능 수행이 가능하다.

인티(Inti)의 Mona Lisa-web은 네트워크 관리 분석 시스템인 Mona Lisa를 멀티미디어 프리젠테이션이 가능한 Web버전으로 개선한 시스템으로 SNMP 프로토콜을 기반으로 TCP/IP 네트워크에 접속된 장치와 트래픽의 장애, 성능에 대한 감시, 분석 관리 기능을 제공한다[9]. 휴렛 팩커드사의 망관리 시스템과 마찬가지로 Mona Lisa 또한 SNMP를 기반으로 SNMP 프로토콜이 운용중인 망장비 관리에 관리기능이 집중되어 있으며, 중앙집중식 관리 체계로 많은 수의 관리 대상으로 구성된 대규모 네트워크에는 적용이 어렵다는 단점이 있다.

본 논문과 유사한 기능을 수행하는 기존에 연구된 웹 기반의 망관리 시스템들은 주로 Java Applet으로 개발되어 수행속도, 적재시간과 같은 성능상의 문제와 Applet download 시의 보안문제등이 나타나고 있다[10,11]. 또한 이러한 문제점의 대안으로 PHP(Professional HTML Preprocessor)등과 같은 스크립트 언어를 통하여 직접 데이터베이스에 접근하여 통계데이터를 수집하는 방법도 제시되고 있으나 실시간의 정보를 나타낼 수 없고, 데이터베이스화 할 수 없는 정보는 표현이 불가능하다는 단점이 있다[11].

서론에 이어 2장에서는 IMRP의 기능을 분석하고 3장에서는 IDL 인터페이스, IMRP의 동작 과정 등을 설명한다. 4장에서 실행 결과를 분석하고 5장에서 결론을 맺는다.

2. IMRP의 기능 분석

IMRP는 CORBA 환경에서 사용자 에이전트(User Agent)에만 접속하도록 함으로써 인트라넷 관리 시스템에 미치는 영향을 최소화하도록 하고, 관리 시스템에 변경이 생기더라도 사용자 에이전트 프로그램만 수정하면 되도록 시스템 유지 보수가 용이하게 한다[12-15]. 또 자바 언어의 서블릿 기능을 이용함으로써 웹 기반으로 망 관리 보고서를 생성할 수 있게 한다.

2.1 구성관리 보고서 작성

인트라넷 관리 시스템은 자신이 관리하는 인트라넷의 정보를 저장하는 방법으로 크게 다음과 같이 두 가지를 사용한다. 우선 현시점의 인트라넷에 대한 정보는 분산 MIB(Management Information Base)에 보관하며 수시로 변하는 현시점의 인트라넷 정보이다. 다른 한 부류의 정보는 과거의 누적된 인트라넷 정보들로서 데이터베이스에 저장된다.

구성관리 현황은 보고서 종류(일별, 주별, 월별, 연도별)에 관계없이 인트라넷 보고서 작성 시점에서의 구성관리 현황을 출력한다. 따라서 구성관리 보고서를 작성하기 위해서는 인트라넷 관리 시스템의 분산 MIB에 접근하여 필요한 자료를 받아와야 한다. 즉 구성관리 보고서 작성 시 인트라넷 관리 시스템이 가지고 있는 당시의 인트라넷 정보를 직접 받아와야 하고, 이를 위해 OMG의 IDL(Interface Definition

Language) 인터페이스를 통하여 인트라넷 관리 시스템과 연동되어야 한다.

분산 MIB를 통하여 수집된 인트라넷 정보는 웹 기반의 사용자 인터페이스로 제공되며, 이는 웹 브라우저가 설치된 모든 컴퓨터에서 인트라넷에 대한 구성관리 보고서를 살펴볼 수 있도록 하여 사용자 접근성과 편의성을 높인다. 망 관리자는 전세계 어디에서나 웹브라우저가 설치된 컴퓨터만 있으면 자신이 관리하는 인트라넷 망 장비의 물리적인 연결 구성에 대한 보고서를 받아 볼 수 있다.

2.2 성능관리 및 장애관리 보고서 작성

IMRP의 성능관리 및 장애관리 보고서에 포함될 현황은 누적된 과거의 통신망 상태에 대한 성능 정보 및 장애정보들이므로 과거의 누적 정보를 담고 있는 데이터베이스로부터 그 정보를 읽어와야 한다. 이를 위하여 인트라넷 관리 시스템은 IMRP가 요구하는 형식의 레코드를 주기적으로 데이터베이스에 저장해야 한다. 그리고 IMRP는 성능관리 및 장애관리 보고서를 작성하기 위해 관리 시스템을 경유하여 데이터베이스에 접근하는 것이 아니라, 데이터베이스에 직접 접근하도록 설계함으로써 인트라넷 관리 시스템의 부하를 줄이도록 한다.

IMRP의 성능관리 및 장애관리 보고서 생성기능 역시 웹 기반의 사용자 인터페이스를 가지게 한다. 따라서 망 운용자는 웹 브라우저를 통하여 위치에 상관없이 자신이 관리하고 있는 인트라넷의 트래픽 부하와 같은 인트라넷 성능 정보와 장애가 발생한 장비들의 목록과 같은 인트라넷 장애관리 보고서를 받아 볼 수 있다.

3. IMRP의 설계 및 구현

3.1 IDL 인터페이스

IMRP는 인트라넷 관리 시스템에서 수집한 과거의 데이터 혹은 인트라넷 관리 시스템이 보유하고 있는 현재의 데이터를 수집하여 가공한 후 자동으로 보고서를 생성해내는 시스템이다. 이를 위하여 IMRP는 인트라넷 관리 시스템이 수집한 정보와 현재 관리 시스템이 보유하고 있는 정보에 대한 수집 기능이 필요하다. 이를 위하여 본 논문에서는 두 가지 다른

형태의 관리 시스템 접근 방법을 사용하고 있다.

그림 1은 IMRP와 인트라넷 관리 시스템간의 연동 구조를 나타내고 있다. IMRP는 크게 두 가지 인터페이스를 통하여 인트라넷 관리 시스템과 연동한다. 과거의 인트라넷 트래픽(traffic) 변화 추세나 장애 발생률과 같은 일정 시간 누적된 데이터에 대한 통계정보를 얻기 위한 JDBC (Java Database Connectivity) 인터페이스가 있다. 또한 IMRP는 네트워크의 현 상태에 대한 보고서를 생성하기 위하여 과거 정보의 수집이 아니라 인트라넷 관리 시스템에서 현재 보유하고 있는 망 정보를 직접 받아들일 필요가 있다. 이때는 OMG의 IDL 인터페이스를 통하여 인트라넷 관리 시스템과 연동한다[6,7].

OMG IDL 인터페이스를 통한 연동에서 고려되어야 할 중요사항은 연동에 사용되는 컴포넌트(component)의 개수이다. 연동에 사용되는 컴포넌트 개수가 많아질수록 연동 구조는 복잡해지고 부하가 커져 전체 시스템의 성능을 저하시킬 수 있다. 본 논문에서 제안하는 IMRP도 CP(Connection Performer), TC(Topological Configurator), PM(Performance Manager), FM(Fault Manager)과 같은 다수의 인트라넷 관리 컴포넌트와 연동하여야 한다. 만약 이들 모두와 IMRP가 직접 1:N 형태로 연동을 한다면 인트라넷 관리 시스템에서 특정 컴포넌트가 변경되면 IMRP의 변경도 같이 일어나게 된다.

본 논문에서는 이러한 문제를 해결하기 위하여 그림 1에서와 같이 GUA(Generic User Agent)를

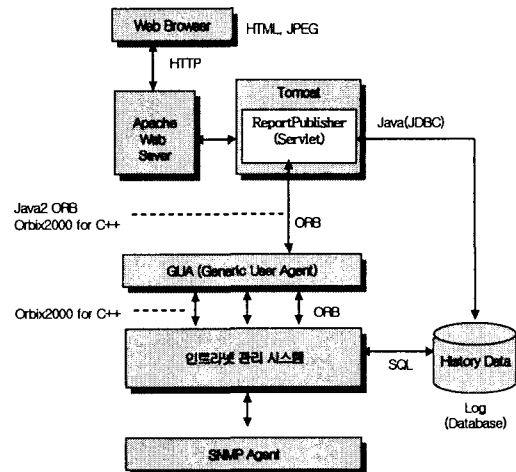


그림 1. IMRP와 인트라넷 관리 시스템간의 연동

IMRP와 인트라넷 관리 시스템 사이에 추가하는 방법을 사용한다. 이는 GUA와 인트라넷 관리 시스템 간의 인터페이스를 추가로 정의하여야 하며 GUA라는 부가적인 컴포넌트 하나를 더 추가하는 것이 되어 성능상의 부하로 작용할 수 있으나 시스템 유지 보수의 측면에서 커다란 장점을 제공한다.

IDL 인터페이스를 통하여 두 개 이상의 컴포넌트가 연동하기 위해서는 OMG의 권고에 따른 IDL 문서 작성이 필요하다. 표 1은 이때 사용되는 IDL 인터페이스들을 나타내고 있다. GUA와 IMRP 간의 연동을 위하여 i_GenericUserAgent라는 IDL 인터페이스가 사용되고 있다. 해당 IDL은 망 관리 컴포넌트가 사용자 에이전트에게 제공 가능한 모든 서비스들을 포함하고 있다. 즉, IMRP는 i_GenericUserAgent 인터페이스로의 접속만 유지하면 망관리 시스템의 모든 기능들을 호출할 수 있다. i_IPCP, i_IPTC, i_IPPM, i_IPFM은 각각 인트라넷 연결관리, 구성관리, 성능관리 및 장애관리 컴포넌트와 GUA간의 연동을 위한 IDL이다. 그림 2는 표 1에서 정리한 IDL 인터페이스간의 연관 관계를 나타내고 있다.

그림 3은 IMRP가 직접적으로 사용하는 i_GenericUserAgent의 IDL 유사 코드(pseudo code)를 나타내고 있다. 그림 3의 IDL 인터페이스에서 제공하는 기능은 모두 4가지 종류로 각각 IPCP(IPConnection Performer), IPTC(IP Topological Configurator), IPPM

표 1. IMRP와 관련된 IDL 인터페이스

연동 컴포넌트	IDL 인터페이스
UserAgent - IMRP	i_GenericUserAgent.idl
UserAgent - IPCP	i_IPCP.idl
UserAgent - IPTC	i_IPTC.idl
UserAgent - IPPM	i_IPPM.idl
UserAgent - IPFM	i_IPFM.idl

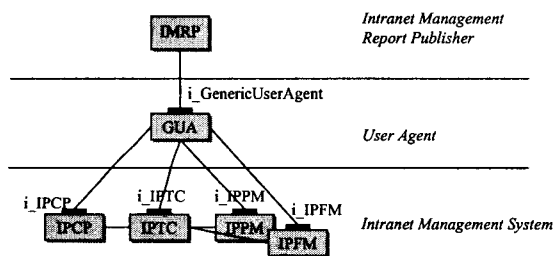


그림 2. IDL 인터페이스들간의 관계

(IP Performance Manager), IPFM(IP Fault Manager)의 기능을 이용할 수 있도록 하고 있다. 표 2는 각 기능들을 요약한 것이고, 표 2에 나타난 메소드(method)들의 기능은 표 3과 같다.

```

...
module GenericUserAgent
{
    interface i_UAService;

    interface i_UserAgent : i_Entity
    {
        attribute t_InstanceName inst_name;
        void create_UAService(
            in t_InstanceName inst_name, out i_UAService uas);
        void destroy_UAService(
            in t_InstanceName inst_name, out string result);
    };

    interface i_UAService : i_Entity
    {
        attribute t_InstanceName inst_name;

        long getAttribute(
            in string com_type,
            in t_InstanceName inst_name,
            out t_Attributes attr );

        long setAttribute(
            in string com_type,
            in t_InstanceName inst_name,
            out t_Attributes attr );

        void convertNameToObject(
            in t_InstanceName name, out Object obj_ref);

        long get_topology_info(
            in t_InstanceName tc_name,
            out t_Attributes attr );

        long createSNW(
            in t_InstanceName tc_name,
            in string rdn,
            in t_Attributes attr );
        ...
        long create_setup_Connectivity(
            in t_InstanceName cp_name,
            in t_InstanceName cnn_name,
            in t_InstanceName root_nwctp,
            in t_InstanceName leaf_nwctp );
    };
}
    
```

그림 3. IMRP와 인트라넷 관리 시스템간의 IDL 인터페이스

```

long check_Delay(
    in t_InstanceName cp_name,
    in t_InstanceName cnn_name,
    out long delay);
...
oneway void requestMonitoring(
    in t_InstanceName pm_name,
    in t_MOInfo MOInfo,
    in Object objectRef,
    in t_MonitoringMethod m_method,
    in t_MonitoringType m_type,
    in t_MeasureDirect direct,
    in t_PeriodType period_t,
    in t_GIAType gia_type,
    in t_ITMA_T itma_type,
    in unsigned short m_period,
    in unsigned short m_duration );

oneway void responseMonitoring(
    in t_MOInfo MOInfo,
    in t_m_result monitoring_result );

void endMonitoring(
    in t_MOInfo MOInfo );
....
oneway void requestEvent(
    in t_InstanceName inst_name,
    in t_Attributes event );

oneway void responseEvent(
    in string EventInfo,
    in t_Attributes event );
....
};
};

```

그림 3. IMRP와 인트라넷 관리 시스템간의 IDL 인터페이스 (계속)

그림 3에 나타난 IDL 인터페이스 정의는 GUA에서 구현되며 IMRP와 인트라넷 관리 시스템간의 연동 기능을 제공한다. GUA는 그림 3의 IDL 인터페이스로 요구된 IMRP의 요구를 인트라넷 관리 컴포넌트에게 다시 요구하여야 하며, 그 결과를 다시 IMRP로 전달하여야 한다. 표 2와 표 3에 나타낸 기능은 각 관리 기능별 대표 기능을 나열한 것으로 필요시 다른 기능을 추가할 수 있다. 그림 3의 IDL 인터페이스는 Microsoft Windows 2000 Server 환경 하에서

표 2. GenericUserAgent module에서 제공하는 연동 기능

연동 모듈	Method 이름	기능
UserAgent - IPCP	create_setup_Connectivity check_Delay	연결관리 정보수집
UserAgent - IPTC	getAttribute setAttribute convertNameToObject get_topology_info createSNW	구성관리 정보수집
UserAgent - IPPM	requestMonitoring responseMonitoring endMonitoring	성능관리 정보수집
UserAgent - IPFM	requestEvent responseEvent	장애 관리 정보수집

표 3. GenericUserAgent module에서 제공하는 method들의 기능

Method 이름	기능
create_setup_Connectivity	연결성(Connectivity) 생성 요구
check_Delay	생성된 연결성을 통한 end-to-end 지연 검사
getAttribute	관리객체(MO:Managed Obejct)의 값 획득
setAttribute	관리객체(MO:Managed Obejct)의 값 설정
convertNameToObject	문자열로 구성된 name을 IOR(Interoperable Object Reference)로 변환
get_topology_info	망의 위상정보 획득
createSNW	Subnetwork 생성
requestMonitoring	망 성능 모니터링 시작
responseMonitoring	망 성능 모니터링 수신
endMonitoring	망 성능 모니터링 종료
requestEvent	망 장애 사건보고 시작
responseEvent	망 장애 사건보고 수신

IONA Orbix 2000 v1.2의 IDL 컴파일러를 통하여 컴파일 되었으며 Microsoft Visual C++ 6.0을 통하여 구현되었다[1-7,14].

3.2 IMRP의 동작

그림 1은 분산 환경에서 인트라넷 관리 보고서를 자동으로 생성하기 위한 동작 구조를 나타내며, 그림 4는 인트라넷의 관리 현황을 보고서로 자동 작성하는 예를 설명하기 위해, 성능관리 보고서 생성 시에

```

<htm>
<body>
:
<p><font size="5">
<b>2. 인트라넷의 성능 관리 분석</b></font></p>
:
<p>Subnetwork별 Traffic 현황</p>
<table>
:
</table>
①

<img src=http://bisdn1.kyungil.ac.kr/servlet
/PM_Subnetwork_Traffic_TotaL_Graph?date1
=20010401&date2=20010401>
②
</body>
</htm>
    
```

그림 4. IMRP의 보고서 생성과정 예

작성되는 HTML 문서의 일부를 나타낸다.

구성관리 현황을 작성하는 방법과 성능관리 및 장애관리 현황을 작성하기 위한 방법은 다르게 구현하였다. 구성관리 현황은 누적된 정보를 필요로 하는 것이 아니라 보고서 작성 시점에 망의 상태가 어떠한가를 나타내야 하므로, 보고서 종류(일별, 주별, 월별, 연도별)에 관계없이 인트라넷 보고서 작성 시점에서의 구성관리 현황을 출력한다. 따라서 구성관리 보고서를 작성하기 위해 인트라넷 관리 시스템에 접근하여 필요한 자료를 받아와야 한다.

그림 1과 4에 따라 IMRP의 기능 중 구성관리 현황을 생성하는 과정을 설명하면 다음과 같다.

- 1) 웹 브라우저에서 아파치 웹 서버(Apache Web Server)에게 인트라넷 관리 보고서 생성을 요청하면, 웹 서버는 톰캣(Tomcat)으로 하여금 ReportPublisher를 실행하도록 요청한다.
- 2) ReportPublisher가 실행되면 Java2ORB를 사용하여 Orbix 2000 for C++로 작성된 GUA에 구성 관리 현황에 대한 내용을 요청한다.
- 3) 그러면 GUA는 ReportPublisher로부터 요청 받은 내용을 Orbix 2000 for C++로 작성된 인트라넷 관리 시스템에게 요청한다.
- 4) 인트라넷 관리 시스템은 요청받은 내용에 해당하는 결과를 GUA에게 리턴하고, GUA는 인트라넷 관리 시스템으로부터 넘겨받은 내용을 ReportPublisher에게 리턴한다.
- 5) 이 때 ReportPublisher가 실행되어 그림 4의 ①, ②와 같은 인트라넷 관리 현황(구성관리)을 웹 서버로 리턴한다.

- 6) 웹 서버는 톰캣으로부터 넘겨받은 내용을 웹 브라우저로 리턴한다. 이 때 웹 브라우저는 그림 4의 ①과 같은 내용을 화면에 출력함으로써 보고서의 장, 절에 대한 제목과 각종 현황 테이블을 출력한다.
- 7) 이어서 그림 4의 ②와 같은 이미지 작성 부분이 나타나면, 웹 브라우저는 웹 서버에게 이것을 실행하도록 요청한다.
- 8) 웹 서버는 다시 톰캣으로 하여금 네트워크 토폴로지를 생성하는 서블릿 프로그램을 실행하도록 요청한다. 톰캣이 실행한 서블릿의 실행 결과를 리턴 받은 웹 브라우저는 토폴로지가 나타나야 할 부분에 이것을 출력함으로써 구성관리 보고서를 작성하게 된다.

그림 1에서 성능 관리 및 장애 관리 현황을 작성할 경우, ReportPublisher는 JDBC를 이용하여 데이터 베이스에 직접 접근·질의하고, 그 결과를 가지고 그림 4와 같은 방법으로 성능 관리 및 장애 관리 현황을 작성한다. 그림 1과 4를 보고 인트라넷 관리 보고서 중 성능관리 및 장애관리 보고서의 생성 과정을 설명하면 다음과 같다.

- 1) 웹 브라우저에서 웹 서버에게 인트라넷 관리 보고서 생성을 요청하면, 웹 서버는 톰캣으로 하여금 ReportPublisher를 실행하도록 요청한다.
- 2) ReportPublisher는 JDBC를 이용하여 데이터 베이스에 직접 접근·질의하고 그 결과를 가지고 온다.
- 3) 이 때 ReportPublisher가 실행되어 그림 4의 ①과 같은 인트라넷 관리 현황(성능관리, 장애관리)과 그림 4의 ②와 같은 이미지(그래프 생성용)의 실행 루틴을 작성하여 이를 웹 서버로 리턴한다.
- 4) 웹 서버는 톰캣으로부터 넘겨받은 내용을 웹 브라우저로 리턴한다. 이 때 웹 브라우저는 그림 4의 ①과 같은 내용을 화면에 출력함으로써 보고서의 장, 절에 대한 제목과 각종 현황 테이블이 출력된다.
- 5) 이어서 그림 4의 ②와 같은 그래프 작성 부분이 나타나면, 웹 브라우저는 웹 서버에게 이것을 실행하도록 요청한다.
- 6) 웹 서버는 다시 톰캣으로 하여금 그래프를 생

성하는 서버릿 프로그램을 실행하도록 요청한다. 톱캣이 실행한 서버릿의 실행 결과를 리턴 받은 웹 브라우저는 그래프가 나타나야 할 부분에 이것을 출력함으로써 보고서의 일부를 작성하게 된다.

위와 같은 방법으로 구성관리, 성능관리, 장애관리 보고서가 순차적으로 생성되어 웹브라우저에 출력된다. 이것을 한글 97이나 마이크로소프트 워드 문서로 변환하여 보고서 형식을 좀 더 세련되게 편집할 수 있다.

4. 실행 및 분석

IMRP의 초기화면에서 양식메뉴(전체, 구성관리, 성능관리, 장애관리)와 기간메뉴(일일, 주별, 월별, 연도별 보고서) 내용 중 각각 하나를 선택하여 보고서의 종류를 지정하고, 확인 버튼을 누르면 망 관리 보고서가 자동으로 생성된다. 인트라넷 관리 보고서는 표지, 목차, 구성관리 현황, 성능관리 현황, 장애관리 현황 순서로 구성된다. 지면 관계상 본 논문에서는 보고서 내용중 일부만 설명하기로 한다.

4.1 구성관리 현황

구성관리 현황에는 네트워크 토폴로지, 서브네트워크 사용 현황, 서브네트워크별 IP 사용 현황, 서브네트워크별 중단점 구성정보 등이 포함된다. 그림 5는 서브네트워크의 토폴로지를 나타내고, 표 4는 165.229.11 서브네트워크에 대한 중단점 구성정보를 나타낸다. 여기서 ifMTU는 인터페이스의 최대 전송 단위를 나타내고 ifSpeed는 인터페이스의 전송대역 폭을 나타낸다. 또 ifAdmin Status는 특정 호스트가 관리 가능한 상태에 있는지의 여부를 나타내며,

표 4. 서브네트워크의 중단점 구성정보

IP Address	ifType	ifMtu (octets)	ifSpeed (Kbps)	ifAdmin Status	ifOper Status
165.229.11.3	Ethernet-csmacd	1500	100,000	up	up
165.229.11.13	Ethernet-csmacd	1500	100,000	up	up
165.229.11.16	Ethernet-csmacd	1500	100,000	up	up
165.229.11.22	Ethernet-csmacd	1500	100,000	up	up
165.229.11.101	propPointToPointSerial	4470	44,210	up	up

1.1 네트워크 토폴로지

- 총 도메인 수 : 2개
- 사용 IP수:6개

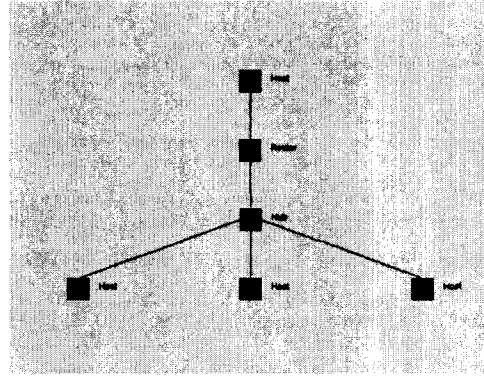


그림 5. 네트워크 토폴로지

ifOper Status는 호스트 장비가 동작상태에 있는지 여부를 나타낸다.

4.2 성능관리 현황

성능관리 현황에는 IP계층 성능분석 현황과 물리계층 성능분석 현황이 포함된다. IP계층 분석현황에는 서브네트워크별 트래픽 현황, 각 서브네트워크의 서비스별 트래픽 현황, 서브네트워크간의 트래픽과 지연현황이 포함되고, 물리계층 분석현황에는 물리노드별 부하현황이 포함된다.

그림 6은 특정 서브네트워크에 대한 일일보고서의 시간대별 트래픽 현황을 나타낸다. 이와 같이 그

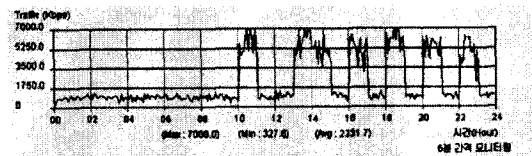


그림 6. 서브네트워크의 트래픽 현황

표 5. 서브네트워크의 서비스별 트래픽 현황

Domain Addr	Service Type	Traffic (Kbps)		
		Max	Min	Avg
165.229.63	HTTP	6955	0	1225
165.229.63	FTP	4731	2	665
165.229.63	SMTP	1968	0	215
165.229.63	TELNET	982	0	114
165.229.63	S_Total	7000	60	2221

표 6. 물리노드별 부하현황

IP Addr	Processor Load			PerfMem		
	Max	Min	Avg	Max	Min	Avg
165.229.63.198	99.9	0.9	48.1	99.9	0.7	52.4
165.229.63.199	99.9	0.1	51.3	98.9	0.2	48.9
165.229.63.21	99.9	0.9	48.1	99.9	0.7	52.4
165.229.63.232	99.7	0.1	46.8	99.6	0.4	50.5
165.229.63.27	99.9	0.1	51.3	98.9	0.2	48.9
165.229.63.28	99.7	0.1	46.8	99.6	0.4	50.5

래프로 나타낼 경우, 일별, 주별, 월별, 연도별 보고서 종류에 따라 각각 데이터를 5분, 30분, 2시간 그리고 하루 간격으로 그 평균값을 구하여 그래프를 작성하도록 구현하였다.

표 5는 특정 서브네트워크의 서비스별 트래픽 현황을 나타내고, 표 6은 물리노드의 부하현황을 나타내며 그래프 형태로도 작성하게 된다. 여기서 Processor Load는 프로세서 이용률을, PerfMem는 메모리 이용률을 나타낸다.

4.3 장애관리 현황

장애관리 현황에는 장애 형태에 따른 서브네트워크별 장애 발생현황, 장애 정도에 따른 서브네트워크별 장애발생 현황, 중요 서버별 장애발생 현황 등이 포함된다. 표 7은 장애 종류에 따른 서브네트워크별 장애 발생 현황을, 표 8은 특정 서버의 장애 발생 현황을 나타낸다. 여기서 Communication 정보는 통신

표 7. 서브네트워크별 장애 발생 현황

Subnetwork	Communication	QoS	Processing	Equipment	Environment
165.229.11	455	489	421	483	420
165.229.63	59	50	68	50	61
165.229.64	65	61	57	52	53

표 8. 특정 서버의 장애 발생 현황

날짜	시간	장애 유형	세부 내용
2001-05-01	00:00:00	Communication Alarm	Connection Down
2001-05-01	00:05:21	Environment Alarm	Fan Down
2001-05-01	00:07:35	Environment Alarm	Fan Down
2001-05-01	00:08:03	Equipment Alarm	Board Failure
2001-05-01	00:09:23	Communication Alarm	Connection Down
2001-05-01	00:23:47	Environment Alarm	Fan Down

생략

링크 다운과 같은 경우, Equipment 정보는 라우터 포트 다운과 같은 경우, Environment 정보는 송풍팬의 고장과 같은 경우에 발생한다.

4.4 기능 비교 분석

본 논문에서 구현한 IMRP와 상용 망관리 시스템 간의 기능 및 구조상의 차이점을 표 9에 나타내었다. 우선 설계 및 구현 시에 적용된 국제 표준에서 IMRP는 기존의 상용 시스템에 비해 보다 다양한 국제 표준안을 수용한 종합적인 설계 구조를 가지고 있다. IMRP는 정보 모델링이 비교적 상세히 정의되어 있는 TMN 체계에서 관리 객체 구조를 수용하였고, 실제 망장비 접속을 위해 SNMP를 사용하였으며, 분산 관리 시스템 구조 설계 시에는 TINA 체계를 준수하였다. 이에 비하여 휴렛 팩커드사와 인티사의 관리 시스템은 SNMP 관리 체계에 기반을 두고 제작되었다. 따라서 타 관리 체계를 따르는 관리 시스템과의 연동이 힘든 단점이 나타난다. 그리고 관리 시스템의 주요 관리 대상 면에서도 IMRP는 Intranet 뿐만이 아니라 Host 관리 기능 역시 강조 되고 있으나 기존의 사용제품들은 네트워킹 장비 관리에 집중하고 있다. 또한 IMRP는 CORBA를 기반으로 하는 분산 망관리 구조로서 대규모 네트워크에 적용이 가능한 확장성을 가지고 있으나 기존의 망관리 제품들은 중앙 집중식 구조를 기반으로 하고 있으므로 관리시스템의 부하를 효율적으로 분산하지 못하며 확장성 문제로 인해 대규모 네트워크에 적용이 힘든 문제가 있다.

표 9. IMRP와 상용시스템간의 기능 비교

	IMRP	HP Openview	Inti Mona Lisa
적용된 국제표준	TMN, SNMP, TINA	SNMP	SNMP
관리 대상	Intranet 및 Host	Intranet	Intranet
CORBA지원여부	지원	일부 지원	일부 지원
시스템 구조	분산형	중앙집중형	중앙집중형
NMS 부하분산	분산	분산 못함	분산 못함

5. 결 론

본 논문에서는 망 관리자가 웹 환경에서 인트라넷의 상황을 쉽게 분석할 수 있도록, 관리자의 요구에

따라 일별, 주별, 월별, 연도별로 인트라넷의 구성관리, 성능관리, 장애관리에 대한 현황을 일괄적으로 또는 선택적으로 자동 생성하는 IMRP를 구현하였다.

IMRP는 웹을 기반으로 하며, 크게 두 가지 인터페이스를 통하여 분산 환경에서 TINA 체계의 인트라넷 관리 보고서를 자동 생성한다. 즉 과거의 인트라넷 트래픽 변화 추세나 장애 발생률과 같은 일정 시간 누적된 데이터에 대한 통계정보를 얻기 위해 JDBC 인터페이스를 이용하고, 네트워크의 현 상태에 대한 보고서를 생성하기 위해 인트라넷 관리 시스템에서 현재 보유하고 있는 망 정보를 직접 받아들일 때에는 OMG의 IDL 인터페이스를 통하여 인트라넷 관리 시스템과 연동한다.

IMRP는 기존의 휴렛팩커드사나 인티사의 보고서 작성 기능보다 다양한 종류의 인트라넷 관리 현황을 제공하고, 분산 환경을 기반으로 하여 인트라넷 관리 보고서를 생성한다는 것이 특징이다. 특히 IMRP는 IP 계층 관점과 물리 계층 관점에서 구성관리, 성능관리, 장애관리 현황을 보고서로 작성해주기 때문에 인트라넷의 상황을 더 정확하게 분석할 수 있게 한다.

참 고 문 헌

[1] Yuji Inoue, Deb Guha, and Hendrik Berndt, TINA Consortium, 1998.
 [2] TINA-C, Management Architecture Ver 2.0, Dec. 1994.
 [3] TINA-C, Network Resource Architecture, Ver. 3.0, Feb. 1997.
 [4] TINA-C, Network Resource Information Model Specification, Ver. 3.0, Dec. 1997.
 [5] TINA-C, Network Components Specification, Ver. 2.2 Final Draft, Dec. 1997.
 [6] OMG, The Common Object Request Broker: Architecture and Specification, 2001.
 [7] IONA Technologies, Orbix Programmers Guide, IONA Technology Ltd, 1997.
 [8] Hewlett-Packard, HP Open View Network Node Manager I, Hewlett-Packard Co., 1999.
 [9] 인티, Mon Lisa White Paper, Available web server from <http://www.inti.co.kr>, 2002.
 [10] 강건호, 오양훈, 송왕철, "Java Applet을 이용한

SNMP 망 관리 사용자 인터페이스", 제주대학교 산업기술연구소 논문집, 제9권 2호, pp135-141, 1998.

[11] 공병철, "Java를 이용한 웹기반의 네트워크 트래픽 모니터링 시스템의 구현", 순천향대학교 석사학위논문, 1999.
 [12] Danny Ayers, Professional JAVA Server Programming, Wrox, 2000.
 [13] George Pavlou, Telecommunication Management Network: A Novel Approach Towards its Architecture and Realisation Through Object-Oriented Software Platforms, Thesis of the degree of Doctor, Mar., 1998.
 [14] ITU-T Rec. X.710, Information Technology - Open Systems Interconnection - Common Management Information Service, 1997.
 [15] ITU-T Rec. G.805: Generic Functional Architecture of Transport Networks, ITU-T, 1996.



정 연 기

1982년 2월 영남대학교 전자공학과 졸업 (공학사)
 1984년 2월 영남대학교대학원 전자공학과 졸업 (공학석사)
 1996년 2월 영남대학교대학원 전자공학과 졸업 (공학박사)
 1985년 3월 ~ 1990년 2월 가톨릭상지대학 전산정보처리과 조교수
 1990년 3월 ~ 현재 경일대학교 IT대학 컴퓨터공학과 교수
 1998년 1월 ~ 1998년 12월 호주 뉴캐슬대학교 전기 및 컴퓨터공학과 교환교수
 관심분야 : 멀티미디어 통신, LAN/WAN 기술, TMN/TINA 체계의 통신망 운용관리, 차세대 인터넷



이 상 동

1992년 2월 경일대학교 컴퓨터공학과 졸업 (공학사)
 2002년 8월 경일대학교산업대학원 컴퓨터공학과 졸업 (공학석사)
 1992년 11월 ~ 현재 밀양대학교 학술정보센터 근무

관심분야 : 멀티미디어 통신, TMN/TINA 체계의 통신망 운용관리, 차세대 인터넷