

## XML 기반의 포괄적인 망 관리 시스템의 설계 및 구현

차 시 호\*, 김 규 호\*\*

### Design and Implementation of An XML-based Universal Network Management System

Si-ho Cha\*, Kyu-ho Kim\*\*

#### 요 약

현재의 망 관리자들은 관리 데이터를 검색하고 조작하기 위한 매우 고가의 특정 업체의 관리 콘솔을 사용하고 있다. 그러나, 점점 더 많은 시스템들이 웹 환경으로 변경되고 있기 때문에, 망 관리자들도 언제 어디서나 시스템을 관리하기를 원하게 되었으며, 관리 정보에 대한 통합적이며, 비용이 효율적이고, 사용하기 쉬운 접근을 원하게 되었다. 본 논문은 이러한 요구사항들을 만족시키기 위하여 관리 콘솔을 위한 여러 인터넷 장비들을 지원하는 XML(eXtensible Markup Language) 기반의 포괄적인 망 관리 시스템(XMAN)의 구현에 대하여 설명한다. XMAN은 3-계층 구조를 가지며, 웹, XML 및 서버 측 자바 기술들의 확장에 기반하여 구현된다.

#### Abstract

Currently, network managers are making use of expensive, vendor-specific management consoles to view and manipulate network data. And more and more systems are transferred to web environment. Therefore, network managers want to get information at anywhere, anytime. This requires an access of management information to be unified, cost-effective, and easy-to-use. This paper presents the implementation of an XML-based universal network management system (XMAN), which supports multiple internet devices as management console. XMAN has 3-tire architecture, and the implementation is based on an extensive use of emerging Web, XML, and Java technologies.

---

\* 광운대학교 컴퓨터과학과 박사과정  
\*\* 서울보건대학 사무자동화과 교수

## I. 서론

인터넷과 웹 기반 기술은 망 관리 분야에 있어서 커다란 변화를 일으키고 있다. 인터넷은 모든 종류의 응용들을 개발할 수 있는 훌륭한 환경을 제공함으로써, 개발 모델이 클라이언트/서버 구조에서 다중 계층(Multi-tier) 구조로 변환되도록 하였다. 전통적인 망 관리 시스템에서는 새로운 관리 기능을 추가하거나 기존의 관리 기능을 변경하려고 한다면, 망 관리자뿐만 아니라 에이전트들 모두를 새로운 버전으로 갱신시켜야 한다. 이것은 심각한 버전 제어의 문제점을 발생시키고 막대한 갱신 비용을 필요로 한다. 그러나, 다중 계층 구조를 갖는 망 관리 시스템에서는 모든 비즈니스 로직이 망 운영자들의 브라우저와 분리된 관리자 시스템에서 처리될 수 있다. 따라서, 관리 기능의 변경이나 추가 시에 나머지 시스템에 미치는 영향을 최소화할 수 있다.

웹 기술들은 사용하기 쉽고 운영 시스템에 독립적인 특성을 갖는다. 웹 기반 기술을 사용함으로써, 망 관리자들은 고가의 설치하기 힘들고, 사용하기 어려운 복잡한 툴을 사용하는 대신에, 웹 상에서 망과 시스템들을 효율적으로 관리할 수 있다. 또한, 점점 더 많은 시스템들이 인터넷과 웹 응용들로 전환되고 있기 때문에, 망 운영자들은 언제 어느 곳에서라도 다양한 인터넷 장비들을 통해서 관리 정보를 얻을 수 있기를 원하게 되었다. 그러나, 각 인터넷 장비들은 정보를 표현하기 위한 다양한 기술들을 사용하기 때문에, 각 장비들에 관리 정보를 보여줄 수 있는 단일 시스템을 개발하기가 쉽지 않다. 가까운 장래에는 망이나 시스템을 관리하기 위해 PDA(Personal Digital Assistant)(1)와 같은 이동 터미널이 사용될 것이다.

따라서, 본 논문에서는 다양한 인터넷 장비들을 통해서 망을 관리할 수 있고, 망 관리 시스템의 효율적인 갱신을 수행할 수 있는 망 관리 시스템을 개발하기 위해, 웹 기반 기술을 사용하는 3-계층 구조를 갖는 포괄적인 망 관리 시스템(XMAN)을 설계하고 구현하였다. XMAN은 웹 서버와 XML 관련 기술들인 XSLT(eXtensible Stylesheet Language Transformations)(8),

DOM(Document Object Model)(5)(7), WML(Wireless Markup Language), 자바 Servlet (5)(7), JDBC(6) 등의 기술들을 사용한다. 또한, XMAN은 PDA와 같은 이동형 전화기를 사용자 인터페이스로 갖는 WAP(Wireless Application Protocol) 인터페이스를 제공함으로써, 이동 중인 운영자들이 이동형 전화기를 통해 망을 관리할 수 있는 구조를 갖는다.

본 논문은 다음과 같이 구성된다. 2장에서는 웹 기반 망 관리 기술과 XML 관련 기술들에 대해 간략히 설명하고, 3장과 4장에서는 XMAN 시스템의 설계와 구현에 대해 설명한다. 5장에서는 실제로 WAP 시뮬레이터인 UP.SDK를 사용하여 망을 관리하는 실험을 수행하고 결과를 분석한다. 그리고, 6장에서는 본 논문에 대한 결론을 내리고 향후 연구에 대해 기술한다.

## II. 웹과 XML 관련 기술들

### 1. 웹 기반 관리 구조

최근의 인터넷과 웹 기반 기술들은 망을 관리하는 방법을 변화시키고 있다. 웹 기술을 사용하는 망 관리 시스템들은 객체 지향 모델로 개발될 수 있으며, 시스템의 구성 요소들을 각각의 모듈로 설계할 수 있다. 그러나, 정보를 표현하는 언어로는 단지 HTML 언어만을 지원한다. 웹 기반 관리 응용들을 개발하기 위해서는 주로 2-계층이나 3-계층의 클라이언트/서버 구조를 사용한다.

#### 1.1 2-계층 구조

2-계층 구조에서 관리자는 망 관리를 수행하기 위해 각각의 피 관리 자원들을 직접적으로 접근한다. 클라이언트 측은 인터페이스로 웹 브라우저를 사용하고 통신 프로토콜로 HTTP를 사용하여 관리 정보에 접근하고 관리 정보를 보여주게 된다. 서버 측은 웹 서버가 클라이언트의 요청을 처리하고 결과로 HTML 페이지의 형태를 전달할 수 있도록 하기 위해 각 피 관리 장비들에 내장된다. 이 경우에, 클라이언트들은 URL 주소를 사용하여 각 피 관리 장비들을 접근하게 된다. URL 주소에 의한 접근 기법은 현재의 망 관리 플랫폼들에 비하여 더 쉽고,

더 친숙하고, 더 효율적인 망 관리가 가능하게 해준다.

### 1.2 3-계층 구조

3-계층 구조에서는 그림 1.에서 보여지는 것처럼 웹 서버가 피 관리 자원들에 접근하는 중간 컴포넌트로 추가 되어 관리자에 의해 접근된다. 첫 번째 계층(First-tier)은 HTTP 프로토콜을 사용하여 서버와 통신을 수행하는 표준 브라우저로 대표되는 클라이언트 부분이다. 두 번째 계층(Second-tier)은 웹 서버와 관리 응용들을 가지며, 주어진 관리 프로토콜을 통해서 피 관리 자원들을 대표하는 세 번째 계층(Third-tier)인 에이전트들에 대한 접근을 수행한다. 이런 방법의 이점은 자신들의 관리 제품들에 웹 접근을 지원함으로써, 포괄적인 기능을 지원하는 웹 기반 관리 인터페이스를 제공할 수 있다는 것이다. 또한, 3-계층 구조는 서버 부분이 모든 피 관리 자원들과 통신하고 망 상태에 대한 표현을 제공하고 하나의 URL 주소를 사용하여 관리 응용들에 대한 유연한 접근을 가능하게 하는 장점을 갖는다.[2]

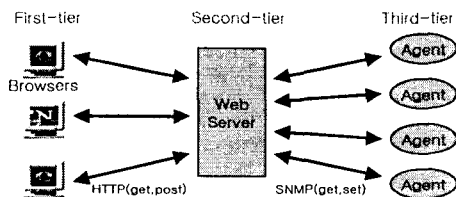


그림 1. 3-계층 구조

## 2. 서버 측 자바 기술과 XML 기술

### 2.1 자바 Servlet과 JDBC

자바는 망 응용들의 구현을 용이하게 할 수 있는 유효한 특성들을 갖는 프로그래밍 언어이다. 자바는 다양한 플랫폼 상에서 수행될 수 있기 때문에 단순하고 이식 가능한 언어이다. 자바 언어의 이식성은 현재의 이질적인 망 환경에 많은 이점을 제공한다.

웹 브라우저를 통한 클라이언트의 요청은 웹 서버를 거쳐서 CGI 프로그램에서 처리된다. 자바 Servlet은 자바를 이용해서 작성한 CGI 프로그램이지만, 일반 CGI 프로그램과는 다른 특성을 갖는다. Servlet은 클라이언트의 요청이 있을 때마다 프로세스를 생성하는 일반적인 CGI 프로그램과 달리 스레드를 생성한다. 스레드 생성은 프로세스 생성에 비해 작업 부하가 적게 걸리고, 속도 면에서도 상당히 빠르다. Servlet은 높은 성능 이외에도 플

랫폼 독립성, 서버 독립성, 확장성, 망을 통한 동적인 로딩 지원 등의 장점을 갖는다. Servlet을 사용하려면 Servlet 엔진이 필요한데, 주로 많이 사용되는 Servlet 엔진으로는 자카르타톰캣(Jakarta Tomcat)[10]과 JSDK 등이 있다. 톰캣은 아파치 그룹과 선, IBM 등의 업체들이 공동으로 개발한 Servlet 엔진으로 무료로 사용할 수 있다.

JDBC는 응용 프로그램들이 DBMS에 연결하고 데이터를 검색할 수 있도록 해주는 자바 클래스의 모임이다. JDBC는 응용 프로그램, 드라이버 매니저, 드라이버, DBMS의 4 부분으로 구성되어 있다. JDBC로 클라이언트 프로그램을 작성하면 DBMS를 변경하여도 응용 프로그램의 코드를 변경할 필요가 없어진다.

### 2.2 XML, DOM, XSL

XML은 데이터를 구조적인 방법으로 기술하기 위한 마크업 언어를 생성하기 위한 기술로서, 데이터가 표현되는 방법으로부터 콘텐츠와 구조를 분리함으로써, 데이터를 저장하고 교환하는데 있어서 매우 유연한 방법을 제공한다. XML은 플랫폼 독립적인 형식이기 때문에, 망 노드들 간에 데이터의 교환이 용이하다. 또한, XML로 표현되는 관리 정보들은 XSL 스타일 시트를 사용하여 인터넷 브라우저나 PDA와 같은 다양한 브라우저를 갖는 사용자 인터페이스를 지원할 수 있다.[13]

XML 문서를 처리하기 위해서는 XML 파서라고 하는 소프트웨어 프로그램이 필요하다. XML 파서는 XML 문서를 읽어서, 구문의 오류를 검사하며, XML 문서의 컨테츠에 대한 프로그램 접근을 가능하게 해준다. XML 파서에는 DOM 기반 파서와 SAX 기반 파서가 있다. DOM 기반 파서는 메모리 내에 XML 문서의 데이터를 포함하는 트리 구조를 생성하고, SAX 기반 파서는 문서를 파싱할 때, 문서 내의 태그나 텍스트 등을 만났을 때 문서로부터 데이터를 반환하는 이벤트를 발생시킨다. 주요 XML 파서로는 IE5에 내장된 XML 파서인 msxml, 아파치 XML 프로젝트의 파서인 Xerces[11], 선 마이크로시스템즈의 XML 파싱을 위한 자바 API인 JAXP, IBM의 자바를 위한 XML 파서인 XML4J 등이 있다. 실제 프로그래머는 라이브러리에 해당 jar 파일만 가져다 놓고 프로그램을 작성하게 된다.

XSL은 XML 문서의 포맷을 지정하기 위해 사용되는 기술로 XSLT, XPath 등의 기술을 사용한다. XSLT는 XML 문서의 포맷을 다른 형태로 변환하기 위해 사용되

는 언어로 XSLT 문서를 처리하기 위해서는 XSLT 프로세서가 필요하다. 아파치의 Xalan은 가장 많이 사용되는 XSLT 프로세서로 자바와 C++를 지원한다. XPath 언어는 XML 문서의 특정 부분에 접근하고 참조하기 위해 XSLT에 의해 사용되는 언어이다. XPath에서 XML 문서는 문서의 각 부분이 하나의 노드로 나타나는 개념적으로 트리로 보여진다. 그림 2.는 XSL의 프로세싱 절차를 보인 것이다. XML 파서는 XML 소스 문서를 소스 트리로 변환하고, XSL 스타일 시트를 해석하여 효율적인 룩업(lookup)을 위한 템플릿 규칙들을 생성한다. 그런 다음, XSL 프로세서는 소스 트리의 루트 노드로부터 탐색을 시작하여 각 노드를 해당하는 템플릿 규칙에 적합시킨다. 만일 정합이 이루어지면, 그 템플릿이 결과 트리로 복제되고, 소스 트리가 완전히 탐색될 때까지 이 작업을 계속 수행한다. 이 작업이 완료되면, XML 프로세서는 결과 트리를 탐색하여 xsl:stylesheet 컨테이너의 xmlns 속성에 의해 내포된 신택스(syntax)를 사용하여 출력 파일이나 스트림으로 복제한다.

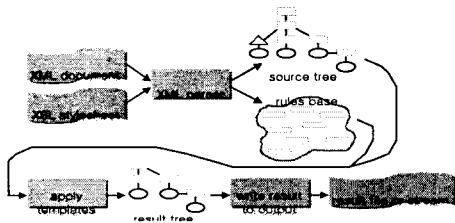


그림 2. XSL 프로세싱 절차

### 3. WAP 기술

WAP[14]은 W3C(World Wide Web Consortium)로부터 XML 1.0 권고안에 추가된 WML을 사용한다. 현재, 많은 디지털 셀룰러 폰들이 웹 브라우징, e-mail 검색, 인터넷 쇼핑 등을 위해 WAP을 지원하고 있다. WAP을 사용하는 것은 망 관리자가 이런 WAP을 지원하는 폰을 사용하여 망을 관리할 수 있도록 함으로써 이동형 관리 스테이션을 가능하게 해준다.

WAP 규격은 디지털 데이터 네트워킹 표준들과 같은 이동 네트워킹 기술들과 XML, URL, 스크립팅, 그리고 다양한 콘텐츠 형식 등과 같은 인터넷 기술들을 확장하여 사용한다.

그림 3.은 WAP 프로그래밍 모델을 보여주고 있다. WAP의 환경에서는 콘텐츠가 WML 형식으로 보여진다.

셀룰러 폰이나 PDA 혹은 다른 인터넷 장비들은 개인용 컴퓨터만큼 강력하지 않다. 이들 모든 장비들은 낮은 계산 능력과 작은 크기의 스크린을 갖는다. 따라서, WAP을 위한 특정한 마크업 언어를 정의해야만 한다. WAP 모델에서는 휴대 단말기(Client)와 인터넷 서버 사이에 WAP 프로토콜이라 불리는 WAP 게이트웨이를 두도록 하고 있다. WAP 게이트웨이의 주요 역할은 WAP 프로토콜과 인터넷 TCP/IP 프로토콜을 중간에서 변환해 주는 것이다. 즉, WML 요청을 HTML 요청 형식으로 변환시켜서 웹 서버에 서비스를 요청하며 서비스가 종료되면 HTML 형식을 WML 형식으로 변환시켜준다.

WAP과 WML을 사용하는 무선 장비들은 인터넷 접속으로 한정되기 때문에, WML은 덱(Deck)이라 불리는 단일의 WML 문서 내에 여러 콘텐츠 페이지들을 포함한다. 이것은 여러 페이지들을 한 번에 다운로드하는 것과 같은 기능을 가짐으로써, 무선 장비가 정보를 다운로드하기 위해 무선 망에 접속해야 하는 접속 횟수를 줄여준다. 덱에 포함되는 각 페이지는 카드(card) 요소에 저장된다. 즉, 각 덱은 하나 이상의 카드로 구성된다.

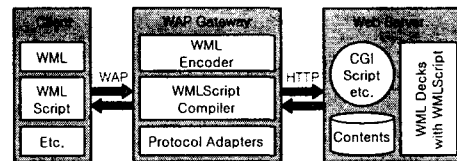


그림 3. WAP 프로그래밍 모델

### III. XMAN의 설계

본 논문에서 설계하고 구현한 XMAN 시스템은 HTML, XML 뿐만 아니라 다양한 마크업 언어를 지원할 수 있다. XMAN 시스템의 전체 관리 구조는 그림 4.와 같이 3-계층 응용 모델을 사용한다. 클라이언트 계층(Client-tier)은 기본적으로 웹 브라우저이며, PDA와 같은 이동형 단말기도 지원된다. 중간 계층(Middle-tier)은 웹 서버를 통해서 클라이언트 계층에 대한 서버의 역할을 수행한다. 중간 계층은 망 관리자의 역할을 수행하기 위한 관리 비즈니스 로직뿐만 아니라 클라이언트의 브라우저에 관리

정보를 보여주기 위한 관리 프리젠테이션 로직을 수행한다. 세 번째 계층(Third-tier)은 실제 피 관리 망상에서 동작하는 SNMP 에이전트이다. 클라이언트 계층과 중간 계층은 HTTP나 WAP 프로토콜을 통해서 통신을 수행하고, 중간 계층과 세 번째 계층 간에는 표준 SNMP 프로토콜을 사용하여 관리 오퍼레이션을 수행한다.

관리 응용에 대한 접근은 웹 브라우저나 PDA를 통해 클라이언트에 의해 시작된다. 중간 계층에 존재하는 관리자는 클라이언트들로부터의 HTTP/WAP 요청을 수신하기 위해 하나의 주어진 통신 포트를 계속적으로 검사한다. 관리자의 관리 서비스 모듈은 웹 서버 모듈에 의해 특정한 처리에 요청이 수신될 때 시작된다. 관리 서비스 모듈들의 주요 업무는 크게 클라이언트의 요청을 해석하여 SNMP 인터페이스 모듈을 통해 장애 관리, 구성 관리, 과금 관리, 성능 관리, 보안 관리를 수행하기 위한 SNMP 오퍼레이션을 수행하는 것과 관리 정보를 클라이언트에 맞게 표현하기 위해 XML 형식을 HTML 형식이나 WML 형식으로의 변환을 수행하는 것이다. 이 관리 서비스 모듈들은 CGI 보다 더 효율적이며 이식성의 장점을 갖는 자바 Servlets로 구현된다. 관리자는 클라이언트들의 요청을 자체적으로 처리하거나 SNMP 프로토콜을 사용하여 SNMP 에이전트를 통해서 요청을 처리하게 된다. 관리자의 XML DB 변환기 또한 자바 Servlet으로 구현되며, XML 파일을 데이터 베이스로 저장하고 데이터 베이스로부터 데이터를 검출하여 XML 문서로 변환하는 기능을 수행한다.

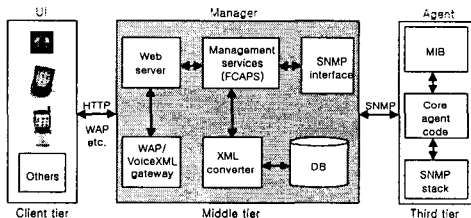


그림 4. XMAN 시스템의 전체적인 구조

### 1. Client-tier(사용자 인터페이스)

XMAN 내의 클라이언트 계층은 웹 브라우저, 셀룰러 폰, PDA, 그리고 그 밖의 인터넷 장비들을 지원할 수 있도록 설계되었지만, 현재의 시스템은 HTML과 WML 데이터 형식만을 지원할 수 있으므로, HTML과 WML을 보여줄 수 있는 인터넷 장비들을 지원할 수 있다. 따라

서, 현재의 시스템에서, 클라이언트 계층은 HTML을 지원하는 모든 인터넷 브라우저와 WML을 지원하는 WAP 장비들로 구성된다.

클라이언트가 관리 응용에 접근하기 위해서는 관리자 웹 서버의 URL 주소를 알아야 하며, 구현에 따라서는 통신 포트 번호도 알아야 한다. 관리자의 웹 서버로 접속이 설정된 이후에는 비 인증된 접근으로부터 관리 응용들을 보호하기 위해 클라이언트는 관리 응용에 대한 접근을 제어하기 위해 패스워드에 의해서 인증된다. 인증은 개방된 인터넷 환경에서 중요한 관리 데이터를 보호하기 위한 기본 기능이다. 따라서, 클라이언트는 망 운영자들의 브라우저로서 단지 망 관리 서비스를 요청하고 관리자 시스템으로부터의 결과를 받아서 보여주는 역할만을 수행한다.

### 2. Middle-tier(관리자)

관리자의 주요 모듈은 브라우저들로부터 발생된 요청들을 수신하기 위해 통신 포트를 계속적으로 검사하는 웹 서버 모듈이다. 웹 서버 모듈은 HTTP/WAP 요청을 수신해서 응답을 처리하는 웹 서버로서 동작한다. 그림 5.는 그림 4.의 XMAN 시스템의 전체적인 구조에서 관리자의 웹 서버 모듈과 관리 서비스 모듈들 및 XML DB 변환 모듈을 세부적으로 보인 것이다.

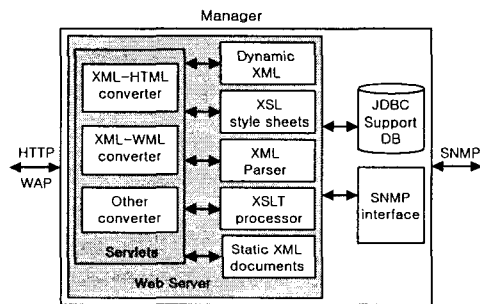


그림 5. XMAN의 관리자 세부 구조

#### 2.1 Web Server

웹 서버 모듈은 관리자의 핵심적인 구성 요소이다. 웹 서버는 자바 Servlets의 컨테이너의 역할을 수행한다. Servlet 컨테이너는 자바 Servlets이 클라이언트들로부터의 요청에 대해 응답할 수 있도록 하기 위한 웹 서버의 주요 부분이다. 웹 서버 모듈은 멀티스레딩에 기반한 자바 프로그램이기 때문에 여러 클라이언트의 요청을 동시에 처리할 수 있다.

### 2.2 Servlets

Servlet들은 망 관리를 위한 비즈니스 로직과 관리 정보를 표현하는 프리젠테이션 로직을 수행한다. 각 Servlet은 특정한 관리 응용을 지원하기 위해 데이터 베이스로부터 저장된 관리 정보를 추출하거나, 직접 SNMP 인터페이스를 통해 SNMP 에이전트로부터 원하는 정보를 얻어서 동적으로 XML 문서를 생성한다. 각 인터넷 장비들은 다양한 정보 표현 방식을 가지기 때문에, Servlet들은 클라이언트의 장비 특성에 맞게 XML 문서 형식을 HTML이나 WML 페이지로 변환하여 클라이언트로 전송하게 된다. XML-HTML과 XML-WML 페이지의 변환 기능은 XSLT 프로세서를 통해서 적절한 XSL 스타일 시트를 적용함으로써 성취할 수 있다.

### 2.3 SNMP 인터페이스

SNMP 인터페이스 모듈은 자바 Servlets과 SNMP 에이전트들 간의 인터페이스를 수행한다. SNMP 인터페이스 모듈은 Servlets에 의해 요청된 SNMP 오퍼레이션을 수행한다. 에이전트로부터 수신된 정보들은 Servlet을 통해 데이터 베이스에 저장되거나 클라이언트의 브라우저에 따라 XML-HTML 혹은 XML-WML 변환 등을 수행하여 클라이언트로 전송된다.

### 3. Third-tier(에이전트)

에이전트는 피 관리 망 장비 내에서 관리 대행자로 동작하는 프로그램이다. 관리자는 에이전트를 통해서 실제 망 장비를 관리할 수 있게 된다. SNMP는 표준 프로토콜이므로 에이전트는 프로그래밍 언어에 독립적으로 개발될 수 있다. XMAN 구조 내에서의 세 번째 계층인 에이전트 시스템은 표준 SNMP를 사용하여 관리될 수 있는 모든 장비들의 에이전트를 의미한다. 현재는 표준 SNMP 프로토콜을 통해서 제어될 수 있는 기존의 모든 SNMP 에이전트를 포함하지만, 향후에는 기존의 SNMP 에이전트를 XML MIB를 지원하는 SNMP 에이전트로 변경함으로써, SNMP MIB를 실시간으로 확장할 수 있는 구조를 지원할 계획이다.

클라이언트 계층은 HTML이나 WML을 지원할 수 있는 모든 종류의 인터넷 장비가 사용될 수 있다. 또한, 세 번째 계층은 실제 관리 대상 망 장비 내의 SNMP 에이전트가 된다. 따라서, 현재의 XMAN 시스템에서의 구현은 중간 계층은 망 관리자에만 국한된다.

망 운영자의 모든 행위는 중간 계층인 관리자를 통해서 에이전트로 전달된다. 관리자는 피 관리 망 장비를 관리하기 위한 비즈니스 로직과 관리 정보를 망 운영자의 브라우저에 표현하기 위한 프리젠테이션 로직을 구현한다.

### 1. 구현 구조

제안된 XMAN 시스템의 세부적인 구현 구조는 그림 6.과 같으며, 3장에서 설명된 설계 구조에 기반하여 모든 모듈들을 자바 언어를 사용하여 구현하였다. 또한, XML 관련 기술들인 XSLT, DOM, WML, 자바 Servlet, JDBC 등의 기술들을 사용하였다. 구현 구조로는 클라이언트의 관리 요청을 받기 위한 웹 서버, XML-WML 혹은 XML-HTML 변환기능을 수행하기 위한 XSLT 프로세서, DB 접근을 위한 JDBC, SNMP 오퍼레이션을 통해서 에이전트와 통신하기 위한 SNMP 인터페이스 등으로 구성된다

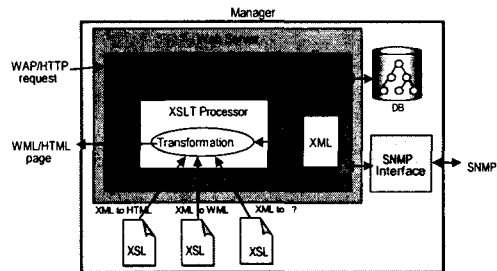


그림 6. XMAN의 구현 구조

### 2. Web Server

웹 서버는 자바 Servlet 컨테이너를 갖는다. 자바 Servlets은 웹 서버 내의 Servlet 컨테이너에서 수행한다. XMAN의 구현에서는 JSP 환경을 지원하는 Servlet 컨테이너인 자카르타 톰캣(Jakarta Tomcat)을 사용하였다. 자카르타 톰캣은 아파치 그룹과 썬, IBM 등의 업체들이 공동으로 개발한 Servlet 엔진으로, 썬의 JSWDK와 아파치의 JServ의 뒤를 잇는 아파치 그룹의 JSP/Servlet 컨테이너이다. 따라서 자카르타를 설치하면 별도로 JSDK나 JSWDK, 혹은 JServ를 설치할 필요가

## IV. XMAN의 구현

없다. 톰캣은 독립적으로 또는 웹 서버에 추가되어 사용 가능하다. 본 구현에서는 jakarta-tomcat 4.0 b3 버전을 사용하여 독립형 서버로 구축하였다. 자카르타는 8080 포트를 사용하기 때문에 URL에 웹 서버(관리자)의 주소와 8080 포트 번호를 포함해서 웹 서버에 접근하게 된다.

### 3. Servlets

각 Servlet은 특정한 관리 응용을 지원하기 위해 데이터 베이스로부터 저장된 관리 정보를 추출하거나, 직접 SNMP 인터페이스를 통해 SNMP 에이전트로부터 원하는 정보를 얻어서 동적으로 XML 문서를 생성한다. 동적으로 생성된 XML 문서들은 XSLT 프로세서를 통해서 적절한 XSL 스타일 시트를 적용함으로써 클라이언트의 장비 특성에 맞게 XML 문서 형식을 HTML이나 WML 페이지로 변환하여 클라이언트로 전송하게 된다. 그림 7.은 Servlet들에 관련된 주요 클래스 다이어그램이다.

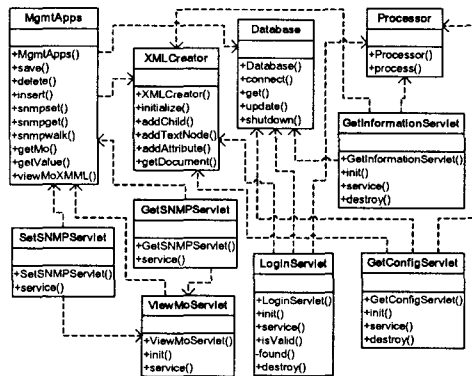


그림 7. XMAN의 주요 클래스 다이어그램

LoginServlet 클래스는 관리 응용으로의 login을 처리하는 Servlet이다. LoginServlet 클래스는 망 운영자에게 사용자 ID와 패스워드를 입력받아서 운영자의 login을 수행한다. XMLCreator 클래스는 DOM을 사용하여 XML 문서를 생성하기 위해 사용되는 유틸리티 클래스이다. DOM 파서로는 Xerces XML 파서[11]를 사용한다. Processor 클래스는 XML 문서를 변환하기 위해 XSL 스타일 시트를 적용하는 유틸리티 클래스이다. XSLT 프로세서로는 Xalan XSLT 프로세서[12]를 사용한다. Database 클래스는 JDBC를 사용하여 데이터 베이스에 대한 연결과 데이터 검색과 갱신을 위한 질의를

위해 사용된다. GetInformationServlet 클래스는 망 요소 데이터베이스로부터 특정 피 관리 노드에 대한 정보를 포함하는 XML 문서를 생성하며 GetConfigServlet 클래스는 망 요소 데이터베이스로부터 피 관리 노드의 구성 정보를 포함하는 XML 문서를 생성한다.

### 4. XSLT 프로세서

XML 문서를 HTML로 변환하거나 WML로 변환하는 등의 XSLT 문서를 처리하기 위해서는 XSLT 프로세서가 필요하다. 본 구현에서는 아파치의 자바를 위한 Xalan XSLT 프로세서를 사용하였다.

그림 7.에서 보인 클래스 다이어그램의 Processor 클래스가 Xalan XSLT 프로세서[12]를 사용하여 XSL 문서 변환을 수행한다. Processor 클래스는 여러 Servlet들에 의해 동적으로 생성된 혹은 정적인 XML 요소 문서와 XSL 스타일 시트를 입력으로 받아서 XSLT 변환을 수행한 결과를 클라이언트인 망 운영자에게 전달하게 된다. 전달되는 데이터의 유형은 클라이언트의 브라우저에 따라 다르게 적용된 XSL 스타일 시트에 따라서 HTML이나 WML로 변환된다.

### 5. SNMP 인터페이스

SNMP 인터페이스의 구현에는 AdventNet에서 자유롭게 사용할 수 있도록 제공하는 자바로 작성된 AdventNet SNMP 패키지 v2c(9)를 사용하였다. AdventNet의 SNMP API는 SNMP 기반 관리 응용들을 구축할 수 있도록 해주며 관리자와 에이전트들 간의 동기 및 비동기 SNMP 통신을 지원해 준다.

## V. 실험 및 결과 분석

본 논문에서는 XMAN 시스템이 WAP 장비를 지원하는지 증명하기 위해 UP.Simulator[13]를 사용하여 실험하였다. UP.Simulator는 Openwave사의 WAP 시뮬레이터이다. XMAN 시스템은 URL을 사용하여 접근된다. 이 실험에서, XMAN 시스템은 Windows 2000에서 실행되었고 관리 정보를 저장하기 위하여 Microsoft 액

세스 2000을 사용하였다. 그림 8은 XMAN 시스템에 대한 시뮬레이터의 결과를 보인 것이다.

그림 8의 첫 번째 그림은 관리자의 로그 인을 처리하는 화면을 보인 것이며, 두 번째 그림은 수행할 망 관리 기능을 선택하는 화면을 보인 것이다. 그리고 마지막 그림은 특정 피 관리 노드인 infotel1에 대한 노드 정보를 보여주는 화면이다.

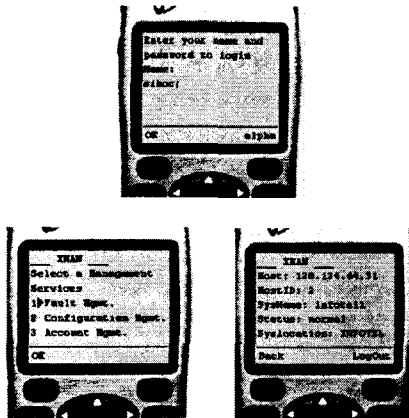


그림 8. XMAN에 대한 WAP 시뮬레이터 결과

## VI. 결론 및 향후 연구

본 논문은 XMAN이라 불리는 XML 기반의 포괄적인 망 관리 시스템의 설계와 구현에 대하여 설명하였다. XMAN 시스템은 망 관리자들이 다양한 인터넷 장비들을 통해서 망을 관리할 수 있도록 하기 위하여 또한 망 관리 시스템의 확장성과 이식성을 제공하기 위하여 웹, XML, 서버 측 자바 기술 등을 사용하여 3-계층 모델을 갖는 구조로 설계되고 구현되었다. 본 XMAN 시스템은 J2EE (Java 2 Enterprise Edition) 플랫폼 기반의 서비스 관리 시스템의 프론트엔드(Front-End) 시스템으로 통합이 가능할 것으로 기대된다. 따라서, 향후 과제로는 XMAN 시스템을 바탕으로 J2EE 플랫폼 상에서 컴포넌트 기반의 서비스 관리 시스템을 설계하고 구현하는 것이다.

## 참고문헌

- [1] Chi-Hsing Chu, Chien-Hsuan Huang, Michael Lee, "Building an XML-Based Unified User Interface System under J2EE Architecture", MSE 2000, December 2000.
- [2] A. Ghlamallah and R. Boutaba, "Implementing a Distributed Web-based Management System in Java", ITS98, vol. 2, 1998.
- [3] H. M. Deitel, P. J. Deitel, T. R. Nieto, T. M. Lin, P. Sadhu, "XML How To Program", Prentice Hall, 2001.
- [4] W3C, "Extensible Markup Language (XML) 1.0 (Second Edition)", <http://www.w3.org/TR/>.
- [5] SUN Microsystems, "Java servlet API 2.3", <http://java.sun.com/products/servlet/>.
- [6] SUN Microsystems, "JDBC API 3.0", <http://java.sun.com/products/jdbc/>.
- [7] W3C, "Document Object Model (DOM) Level 2 Specification", <http://www.w3.org/TR/>.
- [8] W3C, "XSL Transformations (XSLT) version 1.0", <http://www.w3.org/TR/>.
- [9] AdventNet Inc., "AdventNet SNMP API v2c", <http://www.adventnet.com/products/>.
- [10] Apache, "Jakarta Tomcat 4.0", <http://jakarta.apache.org/tomcat/>.
- [11] Apache, "Xerces Java Parser 1.4.2", <http://xml.apache.org/xerces-j/>.
- [12] Apache, "Xalan-Java Version 1.2.2", <http://xml.apache.org/xalan/>.
- [13] Openwave Systems Inc., "UP.SDK 4.1", <http://developer.openwave.com/download/>.
- [14] WAP Forum, "WAP-191-WML", <http://www.wapforum.org/what/technical.html>.
- [15] Gilda Pour, "Enterprise JavaBeans, JavaBeans & XML Expanding the



Possibilities for Web-Based Enterprise Application Development", TOOLS 31, 1999.

- [16] Jae-Oh Lee, "Enabling Network Management Using Java Technologies", IEEE Communications Magazine, January 2000.

### 저 자 소개



차 시 호

1995년: 순천대학교 컴퓨터과  
학과 졸업(이학사)  
1997년: 광운대학교 대학원 컴  
퓨터과학과 졸업(이학석사)  
1997년 ~ 2000년: 대우통신  
종합연구소 선임연구원  
2000년 ~ 현재: 광운대학교  
대학원 컴퓨터과학과 박사  
과정  
관심분야: 서비스 관리, 액티브  
네트워크, 소프트웨어 컴포  
넌트 기술



김 규 호

광운대학교 전자계산학과 졸업  
(이학사)  
광운대학교 대학원 전자계산학  
과(이학석사)  
광운대학교 대학원 전자계산학  
과(이학박사)  
서울보건대학 사무자동화과 교수  
한국OA학회 이사  
관심분야 : ATM 망관리, 사이  
버교육