韓國 컴퓨터情報學會 論文誌
第8卷 第3號, 2003. 9.

2003-8-3-1-8

디지털 콘텐츠 저장 및 유통을 위한 분산 지능형 플랫폼에 관한 연구

장 연 세*, 임 승 린**, 나오키 엔도***

A Study on Distributed & Intelligent Platform for Digital Contents

Yeun-Sae JANG*, Seung-In LIM**, Naoki ENDOH***

요 약

현재 각 기업 및 기관에서 구축하고 있는 디지털 콘텐츠는 HTML을 비롯한 다양한 형식의 멀티미디어로 작성되어 있다. 따라서 타 기관에 전시된 콘텐츠 정보를 검색하거나 연동하기는 불가능하다. 뿐만아니라 시스템을 확장하기 위해서는 전체 시스템을 교체해야하는 문제도 파생될 수 있다. 그러나 본 연구에서 제안하는 플랫폼은 기존 레거시(Legacy) 어플리케이션을 수정하지 않더라도 쉽게 신 시스템과 연동이되고 타 시스템과도 연동이된다. 또한 CORBA에 기반한 부하 분산 기법과 분산 처리기법을 도입하여 하나의 시스템에 과부하가 발생하여 전체 시스템의 성능이 저하되는 문제를 방지하고 안정성을 보장하기 위하여 Fail-over 기능을 제공한다. 또한 SyncML 기법을 도입하여 여러 종류의 DBMS를 사용하는 타 기관과의 콘텐츠 실시간 공유를 지원한다. 이로써 콘텐츠를 쉽게 저장 및 유통할 수 있는 기반 환경을 제공한다.

Abstract

In these day, Digital Contents is authorized in forms of HTML and various kind of multi-media by an enterprise and/or organization. At this circumstances, it is very hard or impossible to search a contents between them. Futhermore, you may have to change your contents service systems to extend to cover more user services needs. On this paper, we propose a new architecture to get extendability and empowered interoperability without any maintenance, update or changes. And we introduce the OMG's CORBA to make fault resilient system and to provide a load balancing using a distributed process mechanism. Even more, we adopted the SyncML specification. It can help us to make it real to synchronize between DBMS at real-time. So all these are support easy content synchronization and user-friendness.

▸ Keywords : SyncML, CORBA, 컴포넌트

* 쿠도커뮤니케이션(주) 기술연구소 소장
** 수원과학대학 인터넷정보과 부교수
*** TOSHIBA SI Tech. Center 센터장
※ 본 연구는 한국문화콘텐츠진흥원 개발 과제의 연구비 지원에 의하여 연구되었음.

Ⅰ. 서 론

Web을 통한 인터넷의 활용이 다양한 비즈니스의 영역으로 확대됨에 따라 e-Business라는 새로운 패러다임이 등장하였다.

B2C(Business to Customer) 또는 B2B(Business to Business)라는 용어로 대변되는 이 새로운 패러다임에 대응하기 위해 각종 문화 콘텐츠로 충만한 전시 기관들은 기존의 시스템을 웹으로 통합하기 위한 노력을 지속적으로 수행하고 있다. 하지만 기존 시스템을 Web으로 통합하는데는 여러 문제가 있으며, 특히 다양한 플랫폼을 갖는 레거시(Legacy) 시스템들 간의 연동 및 서로 다른 컴포넌트 기술을 바탕으로 작성된 분산 컴포넌트들 간의 통합은 해결하기 어려운 문제이다[3][4].

현대의 네트워크 규모는 그 규모가 엄청나게 크며 전 세계 네트워크에는 개인용 PC뿐만 아니라 PDA에 내장된 초소형 컴퓨터에서 군사, 항만 시설 등의 초대형 프레임워크까지 연결되어 있다.

문제는 이러한 네트워크가 서로 다른 언어, 플랫폼과 이기종의 기기를 사용하고 있다는 데에 있다. 이러한 문제점을 해결하기 위해 OMG(Object Management Group)로부터 제정된 규격이 CORBA(Common Object Request Broker Architecture)이다. CORBA를 사용하는 경우에 분산 컴퓨팅을 위한 응용 프로그램 개발이 쉬워질 뿐만 아니라, 하드웨어와 운용체계 및 프로그래밍 언어와 무관하게 분산 객체들간에 커뮤니케이션이 이루어진다. CORBA는 어떠한 언어로 구성되어 있건 간에 클라이언트와 서버는 서로 커뮤니케이션을 할 수 있다. 즉, 클라이언트는 자바 애플릿이고, 서버는 C++로 작성되어 있다고 해도, 클리이언트는 서버에 있는 객체의 메소드를 호출 할 수 있다.

또한 CORBA는 서로 다른 언어와 OS로 설계된 프로그램조각(컴포넌트) 들의 통합을 가능하게 함으로써, 여러 가지 컴포넌트 들을 이용하여 하나의 완성된 Enterprise Application을 구축하는 Component Based Development를 가능하게 해 준다.

무선 인터넷이 발전함에 따라 이동 데이터 단말기를 통해 인터넷(Internet)을 검색하거나, 서버의 개인 정보 관리자(PIMS : Personal Information Management Service) 서비스에 액세스하여 자신의 업무 일정을 관리하거나 전자메일을 교환하는 것과 같이, 이동 데이터 단말기 이용이 개인적인 용도 뿐만 아니라 비즈니스의 새로운 수단으로 발전하고 있다. 또한 무선 이동 통신 환경이 다양해지면서 각 벤더마다 제공하고 있는 이동 단말 장치의 종류가 다양해지고, 개별 유저 당 보유하고 있는 단말기의 수도 늘어나고 있다. 이러한 다수의 단말 장치에 분산되어 있는 동일한 데이터에 대해서 서로 동기(synchronization), 즉 콘텐츠의 일치화가 이루어져야 하는데, 이를 위해 사용되는 동기화 프로토콜은 현재 단말기 벤더마다 각기 독자적인 방법을 개발하여 사용하고 있다.

따라서 이기종 시스템 간, 서비스간 데이터 호환이 불가능한 실정이다. 호환성 없는 동기화 프로토콜이 점점 늘어나면서 데이터 동기화 표준의 필요성이 대두되었고, 이에 2000년 2월 IBM, Lotus, Motorola, Nokia, Palm, Psion, Starfish Software 등의 모바일 관련 업체를 중심으로 SyncML(Synchronization Markup Language) 컨소시엄이 구성되었다. SyncML 그룹은 서로 다른 컴퓨터 플랫폼, 네트웍, 응용서비스에 이용될 수 있는 데이터 동기 방식의 개방형 표준 개발을 목적으로 한다.

본 연구에서는 CORBA에 기반하여 이기종의 전산 자원들을 통합하고, 향후 전개될 컴포넌트의 자원화를 지원하며 SyncML을 이용하여 이 기종의 DBMS간 자료 공유 및 동기화를 이룰 수 있는 아키텍처를 제안한다. 이렇게 함으로써 각 기관에 구축되었거나 구축될 디지털 문화 컨텐츠 관련 전산 시스템간의 연동을 지원하여 사용자에게 다양한 정보에 대한 접근성을 제공하고 컨텐츠간 동기화를 이루어 최신의 정보를 다수의 기관이 동시에 제공하는 기반을 확보할 수 있다.

Ⅱ. 관련 연구

본 연구에서 제안하는 아키텍처는 XML (eXtensible Markup Language)에 기반한 SyncML을 이용하여 데이터를 동기화 하고 CORBA를 이용한 통합형 분산 시스템이다. 전체 아키텍처의 이해를 돕기 위하여 각 요소 기술을 정리한다.

표 1. HTML/SGML/XML의 비교
Table 1. Comparison HTML and SGML with XML

항 목	HTML	SGML	XML
태그	□내장된 DTD 사용 □사용자정의 불가능	□사용자 정의 태그사용	□사용자 정의 태그사용
문서재사용	□불가능	□가능	□가능
응용	□문서의 단순 표현	□복잡한 구조의 문서 및 방대한 내용을 요구하는 문서	□SGML과 동일 □웹에서 정보교환
난이도	□쉬움	□복잡하고 어려움	□SGML의 단순화로 인하여 비교적 용이
검색	□별도의 검색 엔진이 필요하여 검색이 어렵다	□자료의 표현과 내용이 분리되어 정확한 검색 가능	□SGML과 동일
출력형식	□CSS	□DSSSL	□XSL
Data교환	□부가되는 작업량이 방대하여 거의 불가능함	□표현부와 내용부가 분리되어 교환이 용이	□SGML과 동일

1. ML (Markup Language)

이기종의 응용시스템의 통합, 서로 다른 포맷의 데이터 통합, 상황에 따른 웹 페이지의 레이아웃 수정, 상호 연결 가능한 파일 등의 문제는 정보의 검색, 전자상거래, 데이터 처리와 공유 등의 부분에서 웹을 수용하고 활용하는데 문제점으로 대두 되었다. 이러한 문제는 HTML의 언어적 특성으로 인하여 디스플레이에 관해서는 다양한 편의를 제공하지만, 데이터 처리와 관련한 표준 기반 방식은 제공하지 않음으로 인하는데 기인한다[5].

이와 같은 문제들을 해결하기 위한 근본적인 대책으로 정보 교환의 자동화에 문제의 초점이 맞추어 졌다. 이를 위해 HTML의 모체인 SGML(Standard Generalized Markup Language)처럼 MetaData의 사용과 표준이 되는 Shared Context의 정립과 활용이 제시되는 XML의 표준 제정이 급격히 진행되었다.

2.1 SGML

SGML(Standard Generalized Markup Language)이란 ISO(International Standard Organization, 국제표준화 기구)에서 1986년에 제정된 규격으로써 각종 문서 정보를 기술하기 위해 필요한 요소를 구적적으로 정의한 표준이다(ISO 8879). 이 규격은 문헌이나 문서의 표준포맷을 규정하는 것이 아니라 문서의 다양성을 정의할 수 있도록 설계되었다. 즉, 문서가 표현하고 있는 논리적인 구조정보들을 Markup을 이용하여 표현함으로써 정보의 공유와 유통을 극대화한다.

즉 SGML은 S/W패키지나 S/W벤더,H/W기종과 관계없이 자료를 저장하고 정보를 교환하기 위한 개념에서 출발하였다[7].

HTML은 일종의 SGML의 Sub-Set으로써 Application의 한 종류이다. 그러나 실질적 산업표준이라 할 수 있는 SGML이 웹 환경에서 확산되지 못하는 이유는 지나치게 구조가 복잡하고 문서를 정의하기 위하여 고가의 툴을 사용하여야 하는등 투자비용이 높기 때문이다.

2.2 XML

XML이란 eXtensible Markup Language의 약자로 HTML(Hyper Text Markup Language)과 같은 부류의 Markup Language이다. 그러나 HTML과는 달리 사용자가 Tag를 정의할 수 있고 데이터를 기술할 수 있다. 그래서 1998년에W3C에서는 인터넷에서 기존에 사용하던 HTML의 한계를 극복하고 SGML의 문제를 해결하는 방안으로써 XML을 발표하였다. 즉, XML은 웹상에서 구조화된 문서를 전송 가능하도록 설계된, 표준화된 텍스트 형식의 마크업 언어로써 SGML의 Subset이며 SGML보다 훨씬 간결하고 인터넷에서 바로 사용가능한 문서를 표현하는 표준이다.

2.3 HTML

SGML에서 웹의 사용을 위해 규정된 DTD를 만들어 표현위주로 재정의한 것이 HTML이다. SGML의 어플이케이션의 일종으로써 데이터를 표현하는 기능에만 초점이 맞추어져 데이터 교환은 불가능하다.

그래서 XML은 Data의 관점에서 기술하고 구조화하며 HTML은 처리된 데이타를 표현하는 관점에서 이해할 수 있다[5][6][7].

<표 1>에서 HTML, SGML과 XML의 관계를 정리한다.

2. SyncML (Synchronization Markup Language)

SyncML은 현재의 데이터 동기화 프로그램의 문제인 호환성 부재, 환경에 종속적인 운영 문제를 해결하기 위해 만들어졌다.

SyncML 컨소시엄의 스폰서들을 살펴보면 대부분 모바일 계열의 유수한 업체가 많다는 것을 알 수 있다. 실제로 데이터 동기화가 이슈로 부상하는 것은 모바일 장치 사용자의 수가 일정 수준을 돌파하면서부터라고 할 수 있다. 또한 기업 환경에서 기동성이 뛰어난 모바일 장치가 시스템의 한 축으로 사용되면서 PC와 모바일 장치간의 데이터 동기화가 핵심 기능으로 떠오른 것이다.

각 모바일 장치의 운영체제는 SyncML 이전부터 이미 운영체제 레벨에 탑재한 자사의 동기화 스펙이 있었다. 팜 계열의 핫싱크(HotSync)나 윈도우 CE 계열의 액티브싱크(ActiveSync)가 바로 이런 벤더 스펙을 구현한 제품이다. 또한 각 통신 서비스사를 중심으로 개인정보관리(PIMS)에 동기화 프로그램이 활발이 사용되고 있다.

모바일 사용자가 증가하면서 호환성이 없는 벤더 스펙의 동기화 프로그램이 점점 늘어나자 모바일 관련 업체를 중심으로 통일적인 동기화 표준을 작성하기 위해 SyncML 컨소시엄이 구성됐다[8].

자사의 폐쇄적인 스펙을 사용하던 개발사들은 상호호환성 보장을 위한 표준을 도출했고, 그 결과 2000년 12월 SyncML 1.0 스펙이 탄생했다.

모바일 환경에 대한 고려가 스펙에서 눈에 자주 띄는 것은 모바일 관련 업체가 SyncML 컨소시엄의 구성과 스펙 작성을 주도한 것과 무관하지 않다. 하지만 SyncML은 단순히 모바일과 PC간의 데이터 동기화에 사용이 국한된 것은 물론 아니다. PC와 PC, PC와 메임프레임간의 데이터 동기화를 포함한 모든 분산 데이터간의 동기화에 SyncML을 이용할 수 있다[7].

3. CORBA(Common Object Request Broker Architecture)

1989년 당시에 현존하는 객체지향 기술을 바탕으로 응용 프로그램들을 결합하고 객체지향 표준을 제정하기 위해 OMG(Object Management Group)가 탄생하였다.

OMG는 600개 이상의 컴퓨터 관련 단체들이 참가하여 객체지향 기술을 기반으로 이종의 분산된 환경하에서 응용 프로그램들을 통합하고 상호연동할 수 있는 표준기술을 제정했는데, 바로 이 표준이 OMA(Object Management Architecture)이다[1].

OMA는 응용 프로그램간의 통합뿐만 아니라 객체의 생성, 소멸에서부터 저장, 트랜잭션(transaction) 기능에 이르기까지 분산 객체 환경에서 필요한 모든 서비스를 총칭하는 것이다. 이들 기능중 CORBA는 컴퓨터 내부의 버스처럼 서로 다른 프로그램들 사이의 버스 역할을 하는 모듈로서 프로그래머가 원하는 메소드(methods) 등의 위치에 관계없이 마치 로컬에 있는 것처럼 사용할 수 있도록 하는 기능을 제공한다. 따라서 CORBA는 OMA에서 가장 중요한 요소가 된다[1][2].

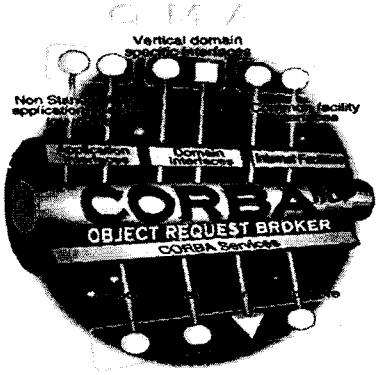

그림 1. CORBA의 구조
Fig 1. CORBA Architecture

CORBA 서비스는 거의 모든 객체가 필요로 하는 기본적인 기능을 제공한다. 이러한 서비스에는 객체의 생명주기 서비스(Object Lifecycle Service), 명명 서비스(Naming Service)와 디렉토리 서비스(Directory Service)등이 있으며 또한 객체지향의 접근 방법인 온라인 트랜잭션 처리(On-line Transaction Processing)와 트레이더 서비스(Trader Service)등을 포함하고 있다.

CORBA 서비스가 객체에게 필요한 서비스를 제공하는 반면, CORBA 편의(CORBA Facilities)는 응용에게 필요한 서비스를 제공한다.

예를 들어 복합문서 관리(Compound Document Management) CORBA 편의는 응용이 복합문서의 컴포넌트를 액세스할 때 표준화된 방법을 제공한다. 이런 CORBA 편의는 두 가지 주요 컴포넌트로 구성된다.

첫째는 수평 편의(Horizontal Facilities)로 복합문서 서비스가 이에 속하며 모든 사업 영역에서 사용될 수 있는 서비스를 말한다. 둘째는 수직 편의(Vertical Facilities)로 특정 산업 그룹에서만 이용될 수 있는 서비스를 말한다[2].

III. 분산 지능형 콘텐츠 플랫폼

현재 각 기업 및 기관에서 구축하고 있는 디지털 콘텐츠는 HTML을 비롯한 다양한 형식의 멀티미디어로 작성되어 있다. 따라서 타 기관에 전시된 콘테츠 정보를 검색하거나 연동하기는 불가능하다. 뿐만 아니라 시스템을 확장하기 위해서는 전체 시스템을 교체해야하는 문제도 파생될 수 있다.

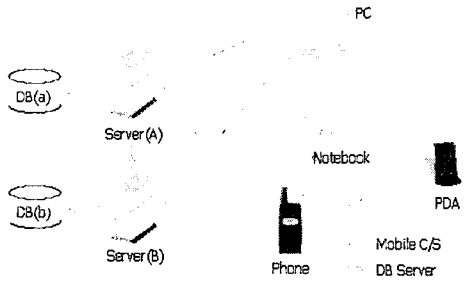

그림 2. 플랫폼의 운영
Fig 2. Operation case

그러나 본 연구에서 제안하는 플랫폼은 기존 레거시(Legacy) 어플리케이션을 수정하지 않더라도 쉽게 신 시스템과 연동이 되고 타 시스템과도 연동이 된다.

또한 CORBA에 기반한 부하 분산 기법과 분산 처리기법을 도입하여 하나의 시스템에 과부하가 발생하여 전체 시스템의 성능이 저하되는 문제를 방지하고 안정성을 보장하기 위하여 Fail-over 기능을 제공한다. 또한 SyncML 기법을 도입하여 여러 종류의 DBMS를 사용하는 타 기관과의 콘텐츠 실시간 공유를 지원한다. 이로써 콘텐츠를 쉽게 저장 및 유통할 수 있는 기반 환경을 제공한다.

1. 플랫폼의 아키텍쳐

콘텐츠 서비스를 제공하는 특정 시스템에 과도한 사용자가 동시에 서비스를 요청하는 경우 CORBA에 기반한 에이전트(Agent)가 타 시스템과의 플러그-앤드-플래이(Plug-&-Play)를 지원한다. 따라서 재 프로그래밍이나 별도의 튜닝과정을 필요로 하지 않는다. 또한 CORBA 에이전트는 콘텐츠 동기화 엔진의 이상 유무를 모니터링 한다.

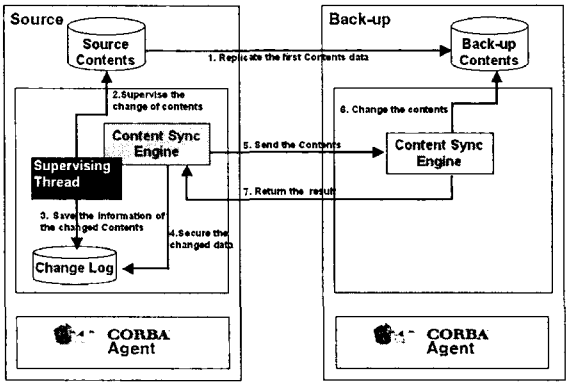

그림 3. 플랫폼의 아키텍처
Fig 3. Proposed Architecture

2. 콘텐츠 동기화 엔진

콘텐츠 동기화를 수행하는 엔진을 (1) 동기화에 필요한 정보를 담는 메시지 구조체, (2) 메시지의 핸들링에 필요한 프로토콜, (3) 구성된 메시지의 네트워크 송수신을 위한 트랜스포트 바인딩과 (4) 상호 운영성을 보장하기 위한 규격 등의 네 가지 카테고리에 따라 분류한다.

2.1 XML Representation

XML Representation 카테고리는 SyncML 메시지 구조체를 정의한다. 각각의 필드가 어떠한 정보를 담고 있으며 해당 정보가 어떤 의미를 내포한 것인지에 대한 약속을 정의하고 있다. SyncML은 메시지 구조체를 정의하기 위해서 세 가지 스펙을 제시하고 있다.

SyncML Representation 프로토콜은 SyncML의 XML 기본 표현을 기술한 핵심 스펙이다. 본 연구에서는 장치 정보 표현을 위한 SyncML Device Information DTD 스펙과 추가적인 동기화 정보와 해석을 표현하기 위한 SyncML Meta Information DTD 스펙은 SyncML Representation 프로토콜에 캡슐화해 사용한다.

2.2 Sync Role

Sync Role 카테고리는 XML Representation 표준을 이용해 동기화를 수행하는 일련의 프로세스 규칙을 정의한다.

SyncML Sync 프로토콜 스펙에 따라 데이터 동기화를 수행하기 위해 필요한 각종 정보의 개념과 동기화 프로세서 등을 일곱 가지 동기화 타입으로 나누고 각각에 대한 메시지 교환 규칙을 정의하였다.

2.3 Transport Binding

트랜스포트 바인딩 카테고리에는 여러 네트워크 프로토콜을 이용해 메시지를 전송할 때 필요한 요구사항을 명세하였다.

2.4 Interoperability

앞서 세 가지 카테고리가 실제로 구현한 콘텐츠 동기화 엔진의 규격이라면 본 카테고리에서는 타 벤더가 구현한 SyncML 엔진과 호환성을 위하여 SyncML 규격을 유지하고 있는지 테스트할 수 있는 절차와 주의사항을 명시하였다.

3. SyncMessage구성

SyncML 메시지에 대한 구조체 규약(동기화 형식)을 SyncML Representation 프로토콜로써 정의했다.

Representation 프로토콜을 내부 기술 언어로 XML을 사용하고 있고 최종적으로 XML 문서로 변환되어 네트워크로 전송된다. 각각의 콘텐츠 메시지는 인터넷의 다목적 메시지 전송을 위해 정의된 MIME 타입으로 정의되어 있다.

따라서 SyncML 메시지는 MIME을 지원하는 애플리케이션이나 트랜스포트 층에서 쉽게 전송될 수 있다.

표 2. SyncML 지원을 위한 명령
Table 2. SyncML Command

종류	역할	커맨드
요청 커맨드	데이터 동기화	Add, Copy, Delete, Exec, Map, Replace, Search
	초기화	Alert
	장치 정보	Get, Put
	컨테이너	Atomic, Sequence, Sync
응답 커맨드		Status, Results

3.1 SyncML 패키지와 메시지

Sync 프로토콜 스펙에서 동기화의 기본이 되는 단위는 패키지(Package)이지만, 패키지의 크기가 해당 장치의 한계 메모리를 넘어서는 경우가 있다. 대용량의 데이터를 동기화할 경우 상대편 장치가 소량의 메모리로 동작하는 PDA와 같은 모바일 환경이라면 메모리 오버플로우가 발생할 수 있다. 따라서 패키지는 상대편 장치의 자유 메모리보다 작은 메시지로 분할해 전송할 수 있어야 한다. 다중 메시지를 지원하기 위해 'Final' 엘리먼트를 사용하는데, 이를 이용해 마지막 메시지인지를 판단한다.

표 3. SyncML 엘리먼트
Table 3. SyncML Element

Common Use Elements	데이터 동기화에 필요한 각각의 정보를 제공하는 필드 구조체
	Archive, Chal, Cmd, Cred, Final, Lang, LocName, LocURI, MsgID, NoResp, NoResults, RespURI, SessionID, Source, Target 등
Message Container	메시지를 이루는 구조체
Elements	SyncML, SyncHdr, SyncBody
Data Description	동기화할 데이터 자체를 담고 있는 구조체
Elements	Data, Item, Meta
Protocol Management	실행을 요청한 커맨드의 성공/실패를 반환
Elements	Status
Protocol command	SyncML 바디에 들어가는 커맨드들
Elements	Add, Alert, Atomic, Copy, Delete, Exec, Get, Map, MapItem, Put, Replace, Results, Search, Sequence, Sync

3.2 Command

SyncML 메시지는 하나하나 개별적인 well-formed XML 문서가 되며 메시지는 SyncHdr 태그로 기술된 헤더와 SyncBody 태그로 기술된 바디로 구성된다. SyncML 헤더는 SyncML 메시지가 전송될 타겟 장치를 찾을 수 있는 라우팅 정보가 있다. 라우팅 정보 이외에더 SyncML 버전 정보, 인증 절차에 필요한 정보를 담고 있다. SyncML 바디는 하나 또는 그 이상의 SyncML 커맨드들을 담는 컨테이너 역할을 한다. 헤더가 동기화 대상이 위치한 장치의 접근 정보를 담고 있다면 SyncML 바디의 커맨드는 동기화 대상이 되는 데이터를 식별하고 실제 동기화에 사용할 정보를 표현하고 있다.

Atomic은 재미있는 커맨드인데 Atomic 안에 존재하는 커맨드들은 어느 하나가 실제 동기화에 실패할 경우 기존에 동기화한 모든 것이 다시 복귀(rollback)하는 트랜잭션의 단위 영역을 나타내 준다.

Sequence는 Sequence 안에 포함된 커맨드들을 순차적으로 처리하도록 요구한다. SyncML의 경우 각각의 커맨드를 성공/실패의 최소단위로 처리한다. 따라서 사용자는 컨테이너 커맨드들을 이용해 성공/실패의 최소 단위를 조절할 수 있다.

3.3 SyncML Element

SyncML 헤더와 SyncML 바디의 각 커맨드들은 엘리먼트의 집합체다 SyncML은 엘리먼트를 조합해 데이터 동기화에 필요한 여러 정보를 전송한다.

4. Well-formed XML 문서 처리기

W3C에서는 여러 마크업 언어(Mark-up Language)간의 데이터 형식과 처리에 관한 아키텍처를 정의하였다.

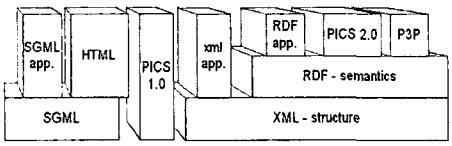

그림 4. W3C 데이터 형식 아키텍처
Fig 4. W3C Data Format Architecture

본 연구에서는 〈표 4〉와 같이 Java를 이용하여 표준 규격을 준수하는 Well-formed XML 문서 처리기를 구현하였다.

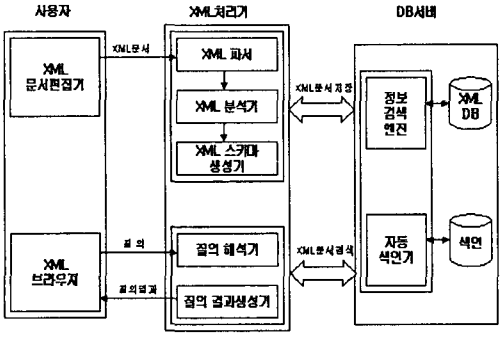

그림 5. XML문서 처리기
Fig 5. XML Processor

표 4. XML 파서
Table 4. XML Parser

```
import org.xml.sax.*;
public class NotifyStr extends HandlerBase {
    static public void main(String[] argv) {
      try {
        Class c = Class.forName(argv[0]); //SAX
        // create instance of the class
        Parser parser = (Parser)c.newInstance();
        NotifyStr notifyStr = new NotifyStr();
        parser.setDocumentHandler(notifyStr);
        parser.parse(argv[1]);
      } catch (Exception e) {
        e.printStackTrace();
      }
    }
    public NotifyStr() {  //dummy constructor
    }
    public void startDocument() throws SAXException {
      System.out.println("startDocument is called:");
    }
    public void endDocument() throws SAXException {
      System.out.println("endDocument is called:");
    }
    public void startElement(String name, AttributeList amap) throws SAXException {
      System.out.println("startElement is called: element name=" + name);
      for (int i = 0; i < amap.getLength(); i++) {
        String attname = amap.getName(i);
        String type = amap.getType(i);
        String value = amap.getValue(i);
        System.out.println("  attribute name="+attname+" type="+type+" value="+value);
      }
    }
    public void endElement(String name) throws SAXException {
      System.out.println("endElement is called: " + name);
    }
    public void characters(char[] ch, int start, int length) throws SAXException {
      System.out.println("characters is called: " + new String(ch, start, length));
    }
}
```

Ⅳ. 결 론

본 연구를 통하여 콘텐츠를 서비스함에 있어 모바일로부터 엔터프라이즈급에 이르는 다양한 시스템 환경에서 단일 플랫폼을 활용할 수 있게 되었다.

또한 기존 레거시(Legacy) 시스템을 인터넷과 연동시킬 뿐만 아니라 일반 엔터프라이즈 시스템에서 DB동기화 분야에 적용 가능하다.

제안된 플랫폼을 통하여 콘텐츠 자동 동기화에 따른 중복 투자 방지, 모바일 시스템과 연동시 안정적 콘텐츠 서비스 및 핵심 기술과 관련한 국산 제품의 간접 수출등의 효과를 얻을 수 있을 것이다.

향후 플랫폼의 응용 범위와 콘텐츠 처리 자동화에 관한 연구가 후속되어야 한다.

참고문헌

[1] CORBA와 DCOM의 통합, 장연세,금영욱, 한국정보과학회, 1999년 7월호

[2] CORBA 3 프로그래밍 바이블, 장연세, 금영욱, 도서출판 그린, 2000년5월

[3] 참조 컴포넌트 아키텍처 모델과 UML 명세화에 대한 연구, 장연세, 한국OA학회지, 2001년 6권 3호

[4] 이형분산 컴포넌트 플랫폼간 상호 운영성 보장에 대한 연구, 한국OA학회지, 2002년 7권 4호

[5] W3C XML Working Group Extensible Markup Language(XML), http://www.w3c.org/XML

[6] XML 문서 저장 시스템, 데이터베이스연구회지, 김훈한상웅, 홍의경, 데이터베이스연구회지, 16권 2호

[7] Generalized SGML Repositories : Requirements and Modeling, Computer & Interface, P. Francois, 18:11-24, 1996

[8] Reusing MS-Windows Software Applications Under CORBA Environment, Re-Chi Lin, IEEE, 1998

저 자 소 개

장 연 세

1993 서울산업대학교 전자계산학과(공학사)

1995 수원대학교 전자계산학과(이학석사)

1998 아주대학교 컴퓨터공학과(박사과정 수료)

1998. 2 ~ 1999. 2 (주)필컴 기술연구소 팀장

1999. 3 ~ 2001. 4 (주)한국정보컨설팅 기술연구소장

2001. 5 ~ 현재 한국컴포넌트 컨소시엄 위원

2002. 5 ~ 현재 쿠도커뮤니케이션(주) 기술연구소장

임 승 린

1979 숭실대학교 전자계산학과(학사)

1987 숭실대학교 전자계산학과(공학석사)

1999 숭실대학교 컴퓨터학과(공학박사)

1978 ~ 1989 (주)대우/대우통신 전산실/과장

1989 ~ 현재 수원과학대학 인터넷정보과 부교수

〈관심분야〉 컴퓨터 응용, 가상교육시스템, 데이터베이스, 웹기반 프로그래밍, 컴포넌트 기반 시스템

나오키 엔도

TOSHIBA SI Tech. Center 센터장