

내장형 XML 데이터베이스를 기반으로 한 XML 문서 데이터 동기화 시스템

심명선[†], 배병진^{**}, 민정훈^{***}, 안병태^{****}, 강현석^{*****}

요 약

SyncML(Synchronization Markup Language) 프레임워크인 Sync4j와 내장형 데이터베이스 관리 시스템 인 버클리 DB XML을 기반으로 한 XML 문서 데이터 동기화 시스템 XDS4j(XML Document Data Synchronization System for java)를 제안한다. 기존의 데이터 동기화 시스템들은 파일 시스템 기반으로 데이터를 갱신하므로, Fast Sync때 데이터 동기화 시간을 매우 지연시킨다. 이에 비해 XDS4j에서는 SyncML 클라이언트와 SyncML 서버의 XML 문서 데이터를 버클리 DB XML을 이용하여 관리하며, Fast Sync때 특정 데이터의 갱신은 XPath를 적용하여 갱신될 XML 문서의 해당 요소만 접근하게 함으로써 데이터 동기화 시간을 줄인다.

An XML Document Data Synchronization System Based on Embedded XML Database

Myoung-Sun Sim[†], Byoung-Jin Bae^{**}, Jeong-Hoon Min^{***},
Byoung-Tae Ahn^{****}, Hyun-Syug Kang^{*****}

ABSTRACT

In this paper, we developed an XML Document Data Synchronization System for java(XDS4j) based on both SyncML(Synchronization Markup Language) framework Sync4j and embedded database system Berkeley DB XML. Because most existing data synchronization systems are based on file systems, data synchronization times at Fast Sync in those systems are very delayed. In the XDS4j, however, XML documents data produced at SyncML client and SyncML server are managed by embedded database system, and also synchronization time at Fast Sync is reduced because only partial elements are accessed by applying to XPath.

Key words: Data Synchronization(데이터 동기화), SyncML, Embedded Database(내장형 데이터베이스), XML

1. 서 론

최근 들어 PDA, 포켓 PC, 휴대폰과 같은 다양한

개인 이동용 기기가 빠르게 보급되고 있다. 이러한 이동용 기기들은 일반적으로 주소록 관리, 할일 관리, 스케줄 관리, 메모 관리 등의 PIMS(Personal

* 교신저자(Corresponding Author) : 강현석, 주소 : 경남 진주시 가좌동 900번지(660-701), 전화 : 055)751-5998, FAX : 055)762-1944, E-mail : hskang@gsnu.ac.kr

접수일 : 2005년 1월 14일, 완료일 : 2005년 5월 10일

[†] 준회원, 경상대학교 대학원 컴퓨터과학과 박사과정 (E-mail : energy6155@hanmail.net)

^{**} 준회원, 경상대학교 컴퓨터과학과 석사 (E-mail : bbj007@hanmail.net)

^{***} 준회원, 경상대학교 컴퓨터과학과 석사 (E-mail : acmask@hanmail.net)

^{****} 정회원, 경상대학교 대학원 컴퓨터과학과 박사과정 (E-mail : ahnbt@hanmail.net)

^{*****} 종신회원, 경상대학교 컴퓨터과학부 교수, 컴퓨터.정보통신 연구소 연구원

※ 이 논문은 2005 정보통신 기초 기술연구지원 사업에 의해 이루어졌습니다.

Information Management System) 응용을 지원하고 있으며 XML 문서 기반의 데이터 관리 기능을 제공한다[1].

그러나 개인이 들고 다니는 이동용 기기에는 항상 원하는 정보가 모두 저장되어 있는 것이 아니다. 즉, PDA에는 필요한 전화번호부만 저장되어 있고, 이와 관련된 텍스트 문서는 회사 PC(데이터 서버)에 저장된다. 이 경우 이동용 기기 사용자들은 무선 인터넷을 통해서 간헐적으로 원격지 데이터 서버에서 필요한 데이터를 이동용 기기로 가져와 데이터 갱신 등의 작업을 수행하게 된다. 즉, 분산된 다중 장치들 사이에서 필요한 정보를 얻고 갱신하는 작업이 필요하다[2].

이 경우 연결되지 않은 시간동안 원격지 데이터 서버에서 변경되었던 내용들을 필요에 따라 이동용 기기의 데이터 저장소에 반영하여야 한다. 이와 같이 두 데이터 집합들 간에 일관성을 유지하는 것을 데이터 동기화(Data Synchronization)라 한다[11].

현재 이동용 기기들에 많이 사용되는 데이터 동기화 프로토콜로는 Palm사의 HotSync, Pumattech사의 Intellisync, Microsoft사의 Active Sync 등이 있으며, 대부분의 경우에 기기 제조사들은 자사의 데이터 동기화 사양(Specification)만을 사용하고 있다[10]. 따라서 각 통신 서비스사들은 응용 서비스인 PIMS 등에 해당 데이터 동기화 프로토콜을 어쩔 수 없이 채택하게 된다.

그런데, 최근 들어 다양한 이동용 기기가 출현하면서 다양한 기기들 사이의 상호 운용성이 보장되는 동기화 프로토콜의 요구가 많아졌다. 이를 반영하여 최근 상호 운용성이 보장되는 데이터 동기화 프로토콜의 표준으로 XML 기반의 SyncML이 제안되었다[3].

SyncML은 기기나 데이터의 종류에 상관없이 다양한 전송 프로토콜과 연동되며 보안성을 가지는 XML 기반의 데이터 동기화 프로토콜로서, SyncML 정의의 일부로 개념적 SyncML 프레임워크가 제시되었다[3,4]. 따라서 특정 시스템에서 SyncML을 사용하기 위해서는 이 SyncML 프레임워크가 제공하는 개념적인 골격을 바탕으로 SyncML 메시지를 인코딩하고 디코딩하여, 데이터 동기화를 처리해야 한다. 현재 사용할 수 있는 대표적인 SyncML 프레임워크로는 SyncML Reference 툴킷[4], Sync4j[5] 등

이 있으며 이들은 공개 소스로 제공되고 있다.

한편 최근 이동용 기기의 보급률 증가와 더불어 제한된 리소스하에서 데이터를 효율적으로 관리하기 위해 내장형 데이터베이스 시스템(Embedded Database System)의 필요성이 크게 증가하였다. 이에 따라 C-ISAM[7], 버클리 DB[8], 버클리 DB XML[15], Infonbyte DB[16], 오라클 9i Lite[17] 등의 다양한 내장형 데이터베이스 시스템이 출현하였다.

이러한 내장형 데이터베이스 시스템을 이동용 기기에 채택하기 위해서는 다양한 플랫폼 지원, 응용과 메모리간의 리소스 로드의 적합성, 병렬성과 범위성에 대한 평가, 이동용 기기의 내장형 데이터베이스 탑재 공간 여부, 가격 등을 고려해야 한다[9].

본 논문에서는 이동용 기기에서 내장형 XML 데이터베이스 시스템인 버클리 DB XML을 기반으로 SyncML을 사용하여 XML 문서 데이터를 동기화할 수 있는 XDS4j(XML Document Data Synchronization System for java)를 제안한다. 이를 위해 우리는 Sync4j를 활용하였다.

논문의 구성은 다음과 같다. 2장에서는 SyncML과 내장형 데이터베이스 관리 시스템에 관련된 기술 동향을 살펴보고, 3장에서는 우리가 제안하는 XDS4j의 구조를 알아본다. 이 때 주로 내장형 XML 데이터베이스를 기반으로 설계된 SyncML 클라이언트와 SyncML 서버에 대해 설명한다. 4장에서는 XDS4j와 FUNAMBOL사의 SC4j(SyncClient API Java J2SE 2.5)를 비교한다. 특히 Slow Sync와 Fast Sync를 수행한 결과를 알아본다. 마지막으로, 5장에서는 결론 및 향후 연구 과제를 기술한다.

2. 관련 연구

이 장에서는 제안하는 XDS4j의 개발에 관련된 두 가지 연구 분야인 SyncML 프레임워크와 내장형 데이터베이스 관리 시스템에 대해 알아본다.

SyncML을 토대로 하여 데이터 동기화를 수행하는 SyncML 프레임워크에 대한 연구가 활발하다. 이러한 연구들은 SyncML Initiative에서 제시한 SyncML 개념적 프레임워크를 기반으로 주로 파일 시스템이나 관계형 데이터베이스[17]를 데이터 저장소로 이용한다. 현재 대표적인 SyncML 프레임워크로는 SyncML 레퍼런스 툴킷, ETRI SyncML 프레임워크, 그리고 Sync4j가 있다.

SyncML 레퍼런스 툴킷[5]은 크게 SyncML 코어 계층(SyncML Core Layer), SyncML 플러그인 계층(SyncML Plugin Layer), SyncML 라이브러리 계층(SyncML Library Layer)으로 구성되어 있다. 그런데 SyncML 레퍼런스 툴킷은 SyncML에서 제시한 Sync 프로토콜 항목을 모두 지원하지는 않는다. 따라서 응용 개발자가 SyncML 레퍼런스 툴킷을 확장 구현해야 한다. 그리고 그것은 C와 C++ 언어로 구현되어 있으므로 웹 환경과 플랫폼에서 독립적으로 동작하기 위해 자바를 이용할 때는 자바 네이티브 인터페이스(Java Native Interface) 모듈을 개발해야 하고 이 경우 동기화 처리 시간이 지연되는 문제를 가지고 있다.

ETRI SyncML 프레임워크[13]는 응용 계층, SyncML 프레임워크, SyncML 레퍼런스 툴킷으로 구성되어 있다. 이러한 ETRI SyncML 프레임워크는 응용에 독립적으로 동작할 수 있도록 SyncML PIMS 어댑터와 DB 어댑터를 C와 C++로 구현하였다. 그러나 다른 응용이 등록될 때 PIMS 어댑터는 다시 구성되어야 하고, ETRI SyncML 프레임워크의 공통 데이터 타입이 변한다면 DB 어댑터의 콜백(Callback) 함수에서 호출하는 부분도 변하게 된다.

Sync4j[6]는 트랜스포트 계층, 코어 계층, 프로토콜 계층, 프레임워크 계층으로 구성되어 있다. 이러한 Sync4j는 자바로 구현된 프레임워크로써 이질적인 플랫폼에서 독립적으로 동작한다는 큰 장점이 있는 반면, SyncML 서버는 J2EE 기반으로 구현되어 있기 때문에 큰 규모 응용에 적합하다.

본 논문에서는 XDS4j의 개발을 위해 플랫폼의 독립성을 고려하여 Sync4j를 사용하였다.

많은 내장형 XML 데이터베이스 시스템은 응용과 직접적으로 연결될 수 있는 소프트웨어 라이브러리 형태로 사용된다. 예를 들어 자바로 작성되는 응용에서 간단한 import 문과 메소드를 사용하여 데이터베이스에 접근할 수 있게 한다. XML 문서를 관리하기 위해 개발된 이러한 내장형 XML 데이터베이스 시스템들은 일반적으로 기존의 관계형 또는 객체 지향형 데이터베이스와 달리 특정 데이터 모델을 갖지 않고 키/값의 쌍으로 데이터를 저장함으로써 처리 속도를 빠르게 하고 데이터베이스 서버 관리자와 다중 프로세스간의 통신에 대한 필요성을 없애 시스템의 오버헤드를 줄인다.

대표적인 내장형 XML 데이터베이스 시스템으로 Infonbyte DB와 버클리 DB XML이 있다.

내장형 환경을 고려해 개발된 Infonbyte DB[17]는 XML 문서를 네이티브(Native) 형태로 저장하며 이러한 형태를 효과적으로 처리하고 변환하는 기능과 W3C에서 제시한 XML 질의 언어 XQuery를 지원한다. Infonbyte DB는 자바로 구현되어 있으며 플랫폼에 독립적으로 동작한다. 그러나 주기억장치에서 PDOM 트리를 생성하여 모든 XML 문서를 처리하므로 메모리 공간을 많이 차지한다. 따라서 특히 PDA, 포켓 PC 등의 소형 단말기의 경우는 적은 메모리와 성능상의 제약성이 있기 때문에 처리 속도에 큰 영향을 미치게 된다. 이러한 문제점을 해결하기 위해 보조기억장치에서 DOM 트리를 유지하고 필요한 노드들만을 주기억장치에서 관리하는 연구도 나타나고 있다[12].

버클리 DB XML[15]은 컨테이너(Container)에 네이티브 형태로 XML 문서를 저장하며, XPath를 이용한 인덱싱 기법과 안정된 쓰레드 기반의 처리를 통해 질의 처리를 효과적으로 수행한다. 그리고 버클리 DB를 하부에 두고 C++ 라이브러리, 자바 라이브러리를 이용해 XML 문서를 다룬다. 따라서 버클리 DB에서 다루는 키/값의 쌍으로 구성되는 레코드를 자료구조로 이용할 수 있으며 XML 문서 저장소인 컨테이너를 이용하여 XML 문서를 관리한다.

그런데, 이러한 내장형 데이터베이스 관리 시스템들은 데이터 동기화 기능이 있지 않거나[15] 파일 시스템을 기반으로 개발되어 있으며, 파일내의 일부 데이터를 갱신할 때도 Active Sync와 같은 일부 시스템[10]을 제외한 대부분의 시스템에서 모든 데이터를 클라이언트와 서버간에 교환해야 한다. 따라서 동시성 제어, 회복 등의 기능을 이용할 수 없고 데이터 전송시간 및 동기화 시간을 크게 지연시키게 된다.

본 논문에서는 XDS4j의 개발을 위해 PDA, 포켓 PC 등 소형 이동 기기의 메모리 성능과 데이터 동기화, 네이티브 XML 저장 방식, C와 C++ 기반의 자바 라이브러리 제공, 데이터 처리 방법 등을 고려하여 XML 기반의 내장형 데이터베이스 시스템인 버클리 DB XML을 사용하였다.

3. XML 문서 데이터 동기화 시스템

Sync4j는 SyncML 개념적 프레임워크를 바탕으

로 SyncML을 구현한 공개소스 SyncML 프레임워크이다[6]. 이러한 Sync4j는 다양한 데이터 저장소를 이용하여 데이터 동기화를 수행할 수 있으며, 파일 시스템 기반의 데이터 저장소에 클라이언트에서 사용되는 XML 문서의 특정 내용을 저장한다. 그러나 우리는 파일 시스템 대신에 내장형 데이터베이스 시스템을 이용하고 데이터 동기화를 위해 XML 문서의 필요한 부분만을 교환할 수 있도록 하는 XDS4j(XML Data Synchronization for java)을 개발하였다. 즉, XDS4j는 Sync4j와 버클리 DB XML을 이용하여 XML 문서의 특정 부분만을 갱신할 수 있도록 XPath 질의를 구성해 XML 문서 데이터에 대해 데이터 동기화를 수행한다. 이러한 XDS4j는 크게 XDS4j 클라이언트와 XDS4j 서버로 구성된다.

3.1 XDS4j 클라이언트

XDS4j 클라이언트는 XML 문서 데이터 동기화를 수행하는 SyncML 클라이언트로서 클라이언트 응용, Sync4j 클라이언트, 클라이언트 DB XML 어댑터, 그리고 버클리 DB XML로 구성되어 있다. 그림 1은 이러한 구조를 나타낸다.

데이터 동기화를 수행하기 위해 명령어 라인에서

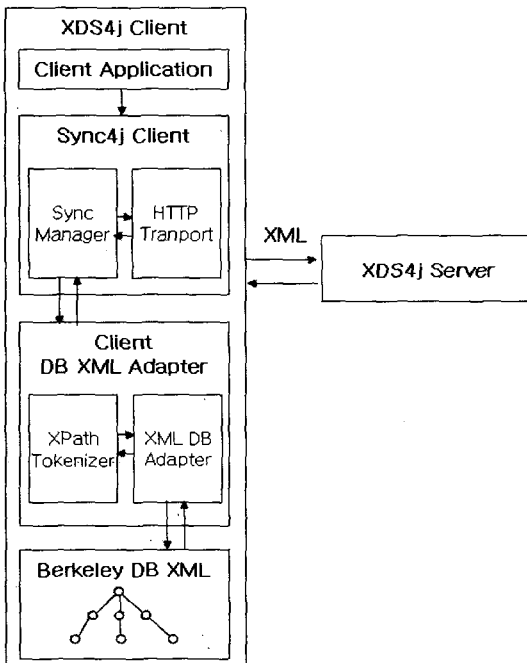


그림 1. XDS4j 클라이언트의 구조

XDS4j 클라이언트를 최초로 시동하는 프로그램은 클라이언트 응용이다. 이 프로그램은 Sync4j 클라이언트의 동기화 수행 관리기인 Sync 관리자를 호출하여 데이터 동기화를 시도한다. 이 클라이언트 응용에 의해 호출된 Sync 관리자는 사용자 이름, 비밀번호, 서버 URI, 서버측 DB 정보 등을 보낸다. 이러한 정보에 대한 상태(Status) 정보를 XDS4j 서버로부터 수신한 후 Slow Sync와 Fast Sync를 구분한다. Slow Sync와 Fast Sync의 동기화 타입의 분류에 따라 데이터 수집은 클라이언트 DB XML 어댑터에서 버클리 DB XML에 접근하여 데이터를 Sync 관리자에게 전달한다.

이러한 데이터를 추출하는 방법은 초기화 과정에 모든 데이터를 동기화 시키는 Slow Sync와 응용 수행중에 갱신되는 데이터만 동기화 시키는 Fast Sync가 각각 다르다. Slow Sync에서는 클라이언트 DB XML에 있는 요소를 루트로 XML 문서의 모든 데이터를 추출한다. 반면에 Fast Sync의 경우는 클라이언트 DB XML 어댑터의 XPath 토큰화기가 특정 요소로 접근할 수 있는 XPath를 생성한 후, XML DB 어댑터가 버클리 DB XML에 있는 해당 XML 문서에서 특정 요소를 루트로 갖는 자식 요소들만을 접근하여 추출한다. 이러한 XML 문서 생성 및 갱신 구조는 3.3절에서 자세하게 설명한다.

이렇게 생성된 XML 문서를 XDS4j 서버에 전달하여 응답을 받은 후 동기화 처리를 종료한다.

3.2 XDS4j 서버

XDS4j 서버는 XML 문서 데이터 동기화를 수행하는 SyncML 서버로서 Sync4j 서버, 서버 DB XML 어댑터, MySQL, 그리고 버클리 DB XML로 구성되어 있다. 그림 2는 이러한 구조를 나타낸다.

XDS4j 서버는 Sync4j 서버를 기반으로 하고 있으나 데이터 저장소는 버클리 DB XML을 이용한다. Sync4j 서버는 HTTP 프로토콜의 POST 방식으로 SyncML 클라이언트와 통신을 담당하는 트랜스포트 계층(Transport Layer), 데이터 동기화 타입에 따라 SyncML 메시지를 생성하는 프로토콜 계층(Protocol Layer), SyncML 표현 프로토콜에서 정의한 SyncML 메시지를 디코딩 및 인코딩하는 코어 계층(Core Layer), XDS4j 클라이언트로부터 받은 SyncML 메시지를 디코딩하여 데이터를 Server DB

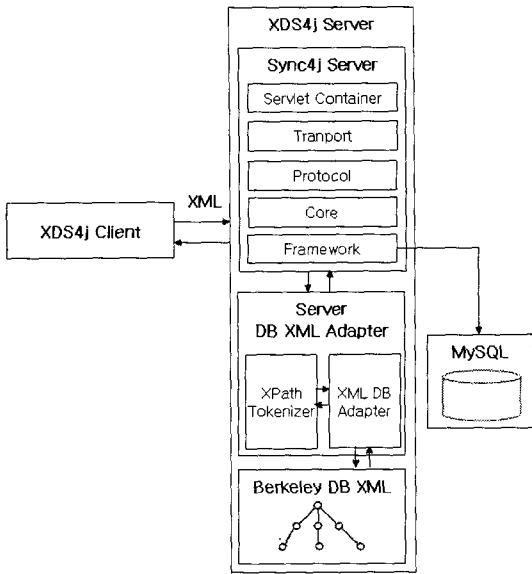


그림 2. XDS4j 서버의 구조

XML 어댑터를 통해 버클리 DB XML에 저장하고 SyncML 클라이언트로 보낼 메시지를 인코딩하는 프레임워크 계층(Framework Layer)으로 구성되어 있다.

XDS4j 클라이언트에서 받은 XML 문서를 버클리 DB XML에서 관리하는 방법은 XDS4j 클라이언트에서와 유사하다. 즉, XDS4j 클라이언트와 마찬가지로 서버 DB XML 어댑터에서 버클리 DB XML을 이용하여 XML 문서를 저장, 수정, 삭제하는 관리 기능을 수행한다. 그리고 XDS4j 서버에서는 사용자 정보, 매핑 테이블 정보를 사용자들이 쉽게 접근할 수 있게 일반 데이터베이스 관리에 사용되는 MySQL로 관리한다. 이는 사용할 DBMS가 변경될 경우, 변경될 수 있다.

3.3 XDS4j 클라이언트와 XDS4j 서버의 XML 문서 갱신 구조

XDS4j 클라이언트는 변경된 XML 문서를 수신하여 변경된 데이터에 대해 특정 요소만을 접근하여 버클리 DB XML에 반영하고 새로운 XML 문서가 추가(Add) 요소일 때는 버클리 DB XML에 데이터를 저장한 후 맵정보를 이용하여 다시 XDS4j 서버에 전송한다. 그리고 XDS4j 서버 또한 XDS4j 클라이언트와 같은 방법으로 버클리 DB XML에 데이터를 저장한다.

그림 3은 XDS4j 시스템에서 갱신된 데이터를 전송받은 XDS4j 클라이언트와 XDS4j 서버가 버클리 DB XML에 그것을 반영시키는 구조를 나타낸다.

우선 SyncML 메시지를 클라이언트(서버) XML DB 어댑터에서 처리할 수 있도록 Sync4j에서 디코딩한다. 이렇게 디코딩된 데이터를 내장형 XML DB에 반영한다. 즉, 추가(Add) 요소일 경우는 XML 문서를 그대로 저장하고, 삭제(Delete) 요소일 경우는 전송받은 XML 문서 ID와 일치하는 XML 문서를 삭제한다. 그리고 갱신(Update) 요소일 경우는 XPath 토큰화기에 의해 변경할 요소에 대한 XPath와 그 요소에 대한 텍스트를 각각 키/값으로 해시 테이블에 저장한 후 XML 문서 ID와 일치하는 요소에 대해 텍스트를 갱신시킨다.

이러한 과정에서 XPath 토큰화기의 역할이 중요하다. 우선 XDS4j 클라이언트의 데이터를 갱신(Update)시키기 위해서 XDS4j 서버에서 전송하는 SyncML 메시지의 일부는 그림 4와 같다.

그림 4에서 “<Target><LocURI>7</LocURI></Target>”은 XDS4j 클라이언트의 버클리 DB XML에 저장되어 있는 XML 문서 ID를 가리키고,

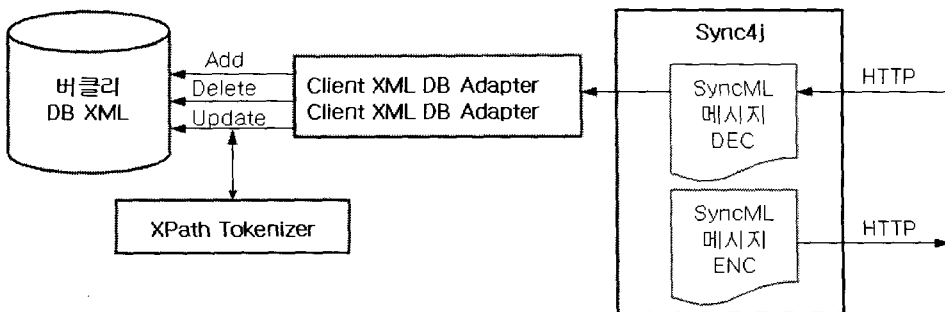


그림 3. XML 문서 갱신 구조

```

<Sync>
.....
<Replace>
<CmdID>5</CmdID>
<Item>
<Target><LocURI>7</LocURI></Target>
<Data><XPath>/xmltest/f1=sudata71</XPath><XPath>/xmltest/f2=sudata72</XPath></Data>
</Replace>
.....
</Sync>
    
```

그림 4. XDS4j 서버에서 전송한 Replace 명령어 메시지

“<Data><XPath>/xmltest/f1= sudata71</XPath><XPath>/xmltest/f2=sudata72</XPath></Data>” 는 XML 문서 ID가 7인 /xmltest/f1과 /xmltest/f2 요소의 텍스트가 각각 sudata71, sudata72로 서버측에 변경되었으니, 클라이언트에서도 변경하라는 의미이다.

이를 위해 XPath 토큰화는 <Data> 요소의 지식 요소를 임의의 XML 문서 xmltest_temp.dbxml로 하여 버클리 DB XML에 저장한 후 “/xmltest/xidid[text()='7]”인 XML 문서에서 “/xmltest/XPath/text()”로 “/xmltest/f1=sudata71”과 “/xmltest/f2=sudata72”를 추출한 후 “=”를 구분자로 구분하여 토큰화시킨다.

그리고 이렇게 토큰화된 결과값을 해시 테이블에 저장하는데 키는 “/xmltest/f1”, “/xmltest/f2”가 저장되고, 대응되는 값은 “sudata71”, “sudata72”가 저장된다. 표 1은 키/값이 저장된 해시 테이블의 상태를 나타낸다.

표 1의 키를 이용하여 그림 5와 같이 XML 문서의 ID가 텍스트값과 일치하는 XML 문서를 추출하여, 키로 갱신되기 전의 요소의 텍스트값을 삭제한 후, 해당 요소의 시작 태그의 다음에 해시 테이블에 저장되어 있는 값으로 대체한 후 XML 문서를 갱신시킨다.

표 1. 해시 테이블(요소와 텍스트의 값)

Index	Key	Value
0	/xmltest/f1	sudata71
1	/xmltest/f2	sudata72

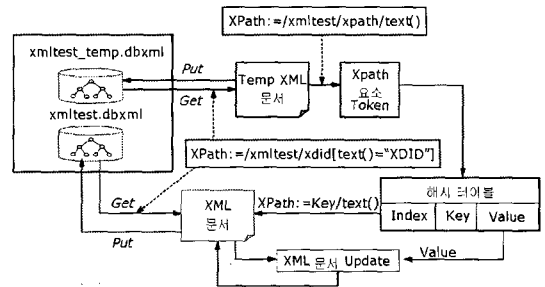


그림 5. XPath 토큰화기의 XML 문서 갱신 과정

4. 성능 비교

현재 내장형 XML 데이터베이스 시스템을 이용하여 데이터 동기화를 수행하는 시스템이 없기 때문에 비교 대상을 찾기가 쉽지 않다. 따라서 이 장에서는 제안한 XDS4j를 파일 시스템을 기반으로 개발된 데이터 동기화 시스템과 비교하였다. 우리는 비교 대상으로 FUNAMBOL사의 SC4j(SyncClient API J2SE 2.5)[14]를 선택하였다. SC4j는 데이터 형태를 데이터베이스가 아닌 텍스트 파일을 사용하지만 XDS4j와 같이 SyncML과 Sync4j를 사용한다. 따라서, 파일 시스템 기반이지만 XML 문서 데이터 동기화 목적은 XDS4j와 같다. 여기서는 XDS4j와 SC4j의 XML 문서 데이터 관리에서 Slow Sync와 Fast Sync를 실행하여 데이터 동기화 처리 시간에 대한 효율성을 비교하였다.

XDS4j와 SC4j는 실제로 XML 문서 데이터의 내부 처리 방법에 있어서 서로 차이를 보인다. 우선 XDS4j는 XML 문서 ID를 버클리 DB XML에서 관리하며 동기화 단위를 XML 문서의 일부로 삼을 수 있고, 해당 요소의 추출을 위해 XPath를 이용한다.

그리고 동기화 시간 정보를 그림 6에서 보는 것과 같이 XML 문서에서 종합적(<last> 태그안)으로 다룬다.

```

C:\WINDOWS\System32\cmd.exe
<xmltest>
<f1 f="N">data1</f1>
<f2 f="N">data2</f2>
<f3 f="N">data3</f3>
<last>1071500761004</f3>
<xmltest>
<xmltest>
<f1 f="N">data11</f1>
<f2 f="N">data22</f2>
<f3 f="N">data33</f3>
<last>1071500761004</f3>
<xmltest>
    
```

그림 6. XDS4j의 XML 문서

이에 반하여 SC4j에서는 그림 7과 그림 8에서 보는 것과 같이 XML 문서 파일과 동기화 시간 정보를 별개로 관리한다. 우선 그림 7은 그림 6과 동일한 문서 내용을 관리하기 위해 SC4j에서 사용하는 파일들과 ID를 나타낸다.

이름	크기	종류
1	11KB	파일
2	4KB	파일
3	21KB	파일
4	4KB	파일
5	1KB	파일
6	2KB	파일
7	3KB	파일
8	14KB	파일
9	7KB	파일
10	3KB	파일

그림 7. SC4j의 파일과 ID

그림 7에서 보는 것과 같이 파일 이름을 ID로 사용하여 데이터가 이중으로 저장되는 것을 방지하고 XDS4j 서버에 파일 이름을 매핑 정보로 전송한다. 다음 그림 8은 데이터에 대한 동기화 상태 정보와 마지막 동기화 시점을 총괄하는 파일을 나타낸다.

```

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
#FileSystemSyncSource file database
Wed Jul 07 20:45:51 KST 2004
9=S1089200740984
8=S1089200740984
7=S1089200741000
6=S1089200741000
5=S1089200741000
4=S1089200741015
3=S1089200741015
2=S1089200741015
1=S1089200741015
10=S1089200741031
    
```

그림 8. SC4j의 상태 정보와 동기화 시간 정보

그림 8의 파일은 데이터 동기화를 수행할 때 java.util.Properties 객체에 "="를 구분자로 구분하여 프로퍼티(Property) 파일에 저장한다. 이렇게 저장되는 Key 값은 파일 이름이고, 대응되는 Value 값은 상태 정보와 마지막 동기화 시점이다. 따라서 마지막 동기화 시점 이후의 데이터를 추출하기 위해서 그림 7의 파일 이름과 그림 8의 Key 값을 비교하여 일치하는 파일을 SyncML 서버측에 전송한다. 이는 텍스트 파일로 저장되기 때문에 보안에 취약하고 데이터가 많을 경우 파일에 접근하는 시간과 데이터를 추출하는데 많은 시간이 걸리므로 데이터를 동기화하는 시간이 길어진다. 그러나, 버클리 DB XML과 같이 데이터 관리 환경을 설정할 필요가 없고 텍스트 파일로 데이터를 저장하므로 메모리를 차지하는 비율이 낮아 시스템의 오버헤드를 줄일 수 있다.

XDS4j와 SC4j를 표 2와 같은 환경에서 Slow Sync와 Fast Sync를 수행하여 데이터 동기화 처리 시간의 효율성을 비교하였다.

표 2. XDS4j와 SC4j의 비교 환경

항목	버전
컴퓨터	펜티엄 IV, 2.6GHz
운영체제	윈도우즈 서버 2003
SyncML 프레임워크	Sync4j 2.0
내장형 데이터베이스	버클리 DB XML 1.2.1
관계형 데이터베이스	MySQL 4.0.18

그림 9는 Slow Sync의 데이터 동기화 처리 시간을 보여준다.

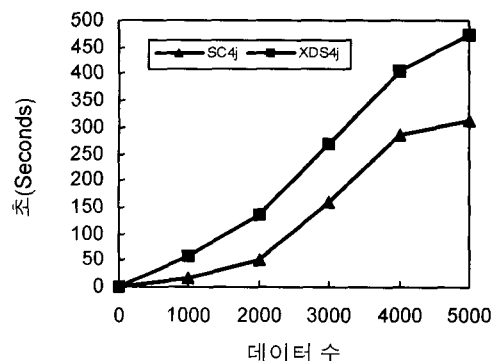


그림 9. Slow Sync 수행 시 데이터 동기화 처리 시간 비교

그림 9에서 SC4j보다 XDS4j의 처리 시간이 많이 걸리고 있다. 이는 SC4j의 경우는 처음 Slow Sync를

수행할 때에 같은 데이터라 할지라도 XML 요소와 같은 태그(Tag) 부분이 없고 특정 디렉토리에 파일 자체를 저장하지만, XDS4j의 경우는 버클리 DB XML의 컨테이너와 버클리 DB 환경을 구성한 후 XML 문서를 저장하기 때문이다. 따라서 XDS4j는 이러한 환경 설정 요소와 같은 데이터가 추가 적용되기 때문에 Slow Sync에서는 SC4j보다 데이터 동기화 시간이 많이 걸린다.

그림 10은 Fast Sync의 데이터 동기화 처리 시간을 보여준다.

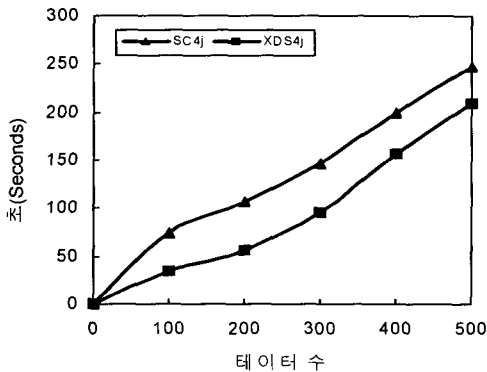


그림 10. Fast Sync 수행 시 데이터 동기화 처리 시간 비교

그림 10에서 가로축은 클라이언트와 서버측에 각각 1000개의 데이터가 저장되어 있고 각각의 변경된 데이터 수를 나타낸다. 예를 들어, 가로축 100은 클라이언트와 서버가 각각 100개가 변경되었다는 것으로 교환할 데이터의 총합계는 200개가 됨을 나타낸다. 이러한 Fast Sync를 수행할 때 SC4j는 파일의 내용 중 변경된 정보만 갱신시키는 것이 아니라 변경된 파일 전체를 전송하여 갱신시킨다. 이에 반해 XDS4j는 변경된 요소에 대한 XPath 정보와 관련된 요소 텍스트만을 전송하므로 전송 시간과 갱신 시간이 더 빠르다.

그림 9와 그림 10을 종합해 보면 Slow Sync의 경우는 SC4j가, Fast Sync의 경우는 XDS4j가 데이터 동기화를 효율적으로 처리함을 알 수 있다. Slow Sync의 경우, 같은 데이터에 대해 XDS4j는 XML 문서를 기반으로 하기 때문에 XML 문서 요소 태그가 추가되어 데이터 크기가 크다는 점과 버클리 DB XML 환경을 구성해야 된다는 점에서 SC4j 보다 길어졌다. 반면에 Fast Sync의 경우, XDS4j는 파일 전체를 갱신시키는 것이 아니라 변경된 요소만을 추출

하여 XML 문서를 갱신시킨다. 그러나 SC4j는 이외에도 데이터에 대한 ID를 파일 이름으로 사용해야 하는 제약성이 있고, 특정 ID로 데이터를 추출하기 위해 순차적으로 검색해야 하는 단점이 있다.

5. 결론 및 향후 과제

본 논문에서는 XML 기반 데이터 동기화 표준인 SyncML을 이용하여 자바로 구현된 XML 문서 데이터 동기화 시스템 XDS4j(XML Document Data Synchronization System for java)를 제안하였다. XDS4j는 내장형 데이터베이스 관리 시스템인 버클리 DB XML과 SyncML 프레임워크 공개 소스인 Sync4j를 이용하여 클라이언트와 서버간의 XML 문서에 대해 데이터 동기화를 수행하는 시스템이다.

본 논문에서 제안한 XDS4j가 갖는 특징은 다음과 같다.

첫째, 레코드 기반이 아닌 XML 문서로 데이터를 관리함으로써 레코드에 저장하기 위해 데이터를 파싱해야 되는 과정없이 XML 문서를 버클리 DB XML에 그대로 저장한다.

둘째, XML 문서를 갱신할 때에는 특정 요소를 XPath로 접근하여 갱신함으로써 클라이언트와 서버간의 데이터 전송 시간을 줄일 수 있다.

셋째, 버클리 DB XML을 이용한 응용의 개발은 내장형 시스템의 특징 플랫폼에 독립적으로 동작할 수 있게 한다.

현재 XDS4j는 XML 문서를 저장할 때 XML 문서에 대한 DTD 정보없이 그대로 저장한다. 따라서 SyncML 클라이언트와 SyncML 서버의 XML 문서 구조가 완전히 같아야 하는데, 처음에 XML 문서가 잘못 입력되면 버클리 DB XML의 컨테이너를 다시 생성해야 한다.

이를 극복하기 위해서는 앞으로 XML DTD 생성기에 관한 연구가 추가로 필요하다. 그리고 내장형 시스템에 버클리 DB XML과 SyncML 클라이언트를 탑재시켜 SyncML 서버와 데이터 동기화를 수행할 수 있도록 확장해야 한다.

참고 문헌

[1] A. Abeta and K. Kakizaki, "Operation Record

Based Workflow Extracting Method for Personal Information Management System," *IEEE Computer*, pp. 224-231, July 1998.

[2] D. Starobinski, A. Trachtenberg, and S. Agarwal, "Efficient PDA Synchronization," *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, pp. 40-51, Jan. 2003.

[3] SyncML Initiative, "http://www.syncml.org".

[4] SyncML Sync Protocol, "http://www.syncml.org/docs/syncml_sync_protocol_v1_1_1_2002_1002.pdf".

[5] SyncML Reference ToolKit Manual, "http://www.syncml.org".

[6] Sync4j Server Manual, "http://sync4j.funambol.com".

[7] C-ISAM, "http://www.informix.com".

[8] Berkeley DB, "http://www.sleepycat.com".

[9] M. Olson, "Selecting and Implementing an Embedded Database System," *IEEE Computer*, Vol 33, No. 9, pp. 27-34, Sept. 2000.

[10] S. Agarwal, D. Starobinski, and A. Trachtenberg, "On the Scalability of Data Synchronization Protocols for PDAs and Mobile Devices," *Special Issue of IEEE Networks on Scalability in Communication Networks*, 2002.

[11] A. Trachtenberg, D. Starobinski, and S. Agarwal, "Fast PDA Synchronization Using Characteristic Polynomial Interpolation," *Proceedings of the IEEE Infocom*, 2002.

[12] 강동완, 제권엽, 홍영표, 한동원, 강현석, 배종민, "내장형 XML 데이터베이스 시스템을 위한 P-DOM의 설계 및 구현," *한국멀티미디어학회*, 제6권, 제2호, pp. 183-193, 2003. 4.

[13] ETRI 휴대클라이언트팀, "SyncML 기반의 데이터 동기화 프레임워크의 세부 구현," *기술 보고서*, 2002.

[14] Sync4j Projects, SyncClient API 2.5 for Java Programmer Guide, "http://sync4j.funambol.com".

[15] Berkeley DB XML, "http://www.sleepycat.com/products/pdfs/ds_dbxml2.pdf".

[16] Infonyte DB, "http://www.infonyte.com".

[17] Oracle 9i Lite Database, "http://www.oracle.com/ip/deploy/database/oracle9i/9lite/".



심 명 선

1995년 경상대학교 컴퓨터과학과 졸업
 2000년 경상대학교 컴퓨터과학과 졸업(석사)
 2002년~현재 경상대학교 컴퓨터과학과 박사과정
 2002년~현재 마산 한일전산여자고등학교 재직

관심분야 : 관계형 데이터베이스, 데이터 동기화, MPEG-7



배 병 진

2002년 경상대학교 컴퓨터과학과 졸업
 2004년 경상대학교 컴퓨터과학과 졸업(석사)
 2004년~현재 한국기계연구원 경영정보팀 재직

관심분야 : 내장형 데이터베이스, 그리드 컴퓨팅



민 정 훈

2003년 경상대학교 컴퓨터과학과 졸업
 2003년~현재 경상대학교 컴퓨터과학부 졸업(석사)

관심분야 : MPEG-7, 멀티미디어 데이터베이스



안 병 태

1999년 2월 국민대학교 전산학부 졸업
 2001년 2월 경남대학교 컴퓨터공학부 석사 졸업
 2003년~현재 경상대학교 컴퓨터과학과 박사과정
 2001년~현재 Best Click Computer 대표

관심분야 : 내장형 데이터베이스, XML, MPEG7, MPEG21



강 현 석

- 1981년 동국대학교 전자계산학과 졸업
- 1981년 서울대학교 전산학과 졸업(석사)
- 1986년 서울대학교 전산학과 졸업(박사)
- 1981년~1984년 한국전자통신연

구원

1984년~1993년 전북대학교 전자계산학과 부교수
1993년~현재 경상대학교 컴퓨터과학부 교수
관심분야 : MPEG-7, 내장형 데이터베이스