

SyncML 클라이언트 설계 및 구현

이지연^o, 김연수*, 최 훈
충남대학교 컴퓨터공학과

e-mail : {eunbi, hchoi}@ce.cnu.ac.kr, yskim@strauss.ce.cnu.ac.kr

Design and Implementation of a SyncML Client

JiYeon Lee^o, YounSoo Kim*, Hoon Choi

Dept. of Computer Engineering, Chungnam National University

요 약

PDA, 노트북, 데스크탑 등 다수의 단말 장치에 분산되어 있는 동일한 데이터들에 대해 가장 최신 값으로 일치시키는 것을 데이터 동기화(data synchronization)라고 한다. 기존의 동기화 솔루션들은 장치간 또는 응용 서비스간 호환성이 결여되었다. 이러한 문제점을 해결하기 위하여 SyncML 프로토콜 표준 규격이 발표되었다. 본 논문에서는 SyncML 표준 규격을 따르며 사용자 단말에서 실행되는 클라이언트를 설계하고 구현하였다. 또한, 기능을 검증하기 위하여 본 연구팀이 이미 구현한 SyncML 서버와의 테스트를 수행하였으며, PIMS 서비스에 대해 서버와 효과적으로 데이터 동기를 수행함을 확인하였다.

1. 서론

무선 인터넷이 발전함에 따라 PDA(Personal Data Assistant)나 이동 전화기 같은 단말기에서 네트워크 서버의 개인 정보 관리 서비스(PIMS: Personal Information Management Service)를 사용함으로써 자신의 일정을 관리하거나 전자 메일을 교환하는 것이 개인적인 용도뿐만 아니라 엔터프라이즈용으로, 비즈니스의 새로운 수단으로 발전하고 있다. 또한, 한 개인이 여러 개의 디바이스를 소유하는 경우도 증가하였다. 그러나, 현재 휴대형 단말기는 물리적 크기의 제한으로 인하여 소량의 메모리, 소형 프로세서를 탑재하기 때문에 서버에 비해서 상대적으로 자원이 제한적이며, 성능이 떨어진다. 따라서, 사용자는 개인 주소록, 일정 등의 정보를 포함하여 디바이스의 중요한 정보를 안전하게 보관하기 위해 신뢰성 있는 서버에 정보를 백업(backup)하기도 하며, 다른 디바이스에 존재하는 정보를 서로 교환하기도 한다. 이러한 경우, 데이터 동기화(data synchronization) 기능이 필요한데, 이는 갱신된

데이터를 변경하고 데이터간의 버전 차이를 해결하여 가장 최신 값으로 일치시키는 중재 동작을 의미한다. 기존의 각 PDA 메이커별 데이터 동기화 솔루션은 이 기종 디바이스간 상호 운용성을 보장하지 못하기 때문에, 모바일 관련 업체들을 중심으로 SyncML(Synchronization Markup Language) 그룹이 구성되었고 데이터 동기화를 위한 표준 프로토콜인 SyncML을 제정하였다[1][2].

본 논문의 2 장에서는 SyncML에 대한 소개와 SyncML을 지원하는 클라이언트 서버간 동작 방식에 대하여 설명하고, 3 장에서 본 논문에서 구현한 SyncML 클라이언트의 구조 및 개발 내용에 대하여 기술한다.

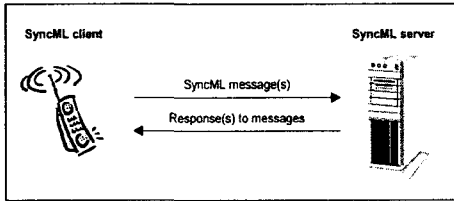
2. SyncML

표준 데이터 동기화 프로토콜의 필요성에 따라 IBM, Lotus, Motorola, Nokia, Palm, Psion, Starfish Software 등의 모바일 관련 업체를 중심으로 SyncML(Synchronization Markup Language) 그룹이 구성되었다[1][2]. SyncML 그룹은 서로 다른 컴퓨터 플랫폼, 네트워크, 응용 서비스에 이용될 수 있는 데이터 동기 방식의 개방형 표준 개발을 목적으로 하며, 2000년 12월에 SyncML 규격 1.0을 시작으로 현재 규격 1.0.1까지 발표하였다.

^o 이 연구는 정보통신부에서 지원하는 정보통신기초기술연구지원사업의 지원으로 수행
* 김연수는 BK21 충남대학교 정보통신인력양성사업의 지원을 받음

SyncML(Synchronization Markup Language)은 임의의 네트워크 상에서 서로 다른 디바이스 및 응용 서비스 간에 데이터를 일치시켜 주기 위한 국제 표준 언어이다. SyncML 은 XML((Extensible Markup Language) 형태로 표현된 데이터를 포함한 메시지를 서버와 디바이스 간에 주고 받음으로써 서버와 디바이스 간에 데이터 동기화를 수행한다[3]. 즉, 분산되어 있는 여러 단말기 내의 공통된 데이터들을 서로 일치시킨다.

다음 <그림 1>은 SyncML 을 지원하는 클라이언트와 서버간의 데이터 동기화를 도식화 한 것이다.



<그림 1> SyncML 클라이언트와 서버

SyncML 클라이언트가 먼저 서버에게 갱신된 데이터를 포함한 메시지를 전송하면, SyncML 서버는 클라이언트가 요청한 동기화 타입에 의해 서버측 데이터와 동기화 작업을 수행한 후 다시 클라이언트에게 작업 결과를 송신한다[4]. 데이터 동기화 과정은 이러한 몇 번의 메시지 교환을 거치며, 정상적인 수행 후에는 클라이언트와 서버간 해당 응용 프로그램 데이터에 대하여 동일한 상태를 유지하게 된다.

SyncML 프로토콜은 다음의 7 가지 동기화 모드를 제공한다[4].

<표 1> 동기화 종류

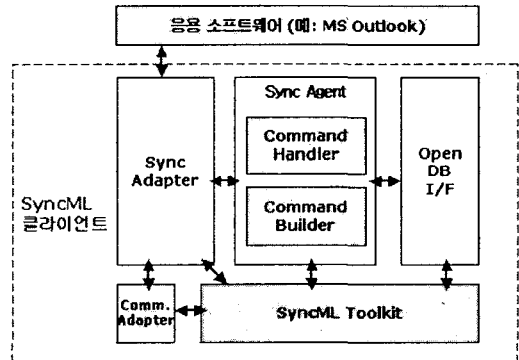
동기화 종류	설명
Two-way Sync	가장 일반적인 방법으로, 클라이언트와 서버 모두 변경 정보를 교환하여 동기화 수행
Slow Sync	응용 서비스 데이터 전체에 대하여 동기화 수행
One-way Sync from Client Only	클라이언트측의 변경 사항만 서버와 교환하여 동기화 수행
One-way Sync from Server Only	서버의 변경 사항만 클라이언트와 교환하여 동기화 수행
Refresh Sync from Client Only	서버의 모든 데이터를 클라이언트의 데이터로 변경하는 방식
Refresh Sync from Server Only	클라이언트의 모든 데이터를 서버의 데이터로 변경하는 방식
Server Alerted Sync	서버가 별도의 오프라인 통신에 의해 클라이언트에게 동기화를 시작하도록 요구하는 방식

3. SyncML 클라이언트

3.1 SyncML 클라이언트 구현

SyncML 클라이언트 모듈은 PDA, 휴대폰 등의 이동형 디바이스에 탑재되며, 이를 사용하여 한 사용자의 여러 디바이스 간이나 SyncML 서버와 데이터 동기화를 수행할 수 있다. 일반적으로 이동 디바이스는 물리적 크기의 제한으로 소량의 메모리, 소형 프로세서를 탑재하기 때문에 서버에 비해서 상대적으로 자원이 제한적이며, 성능이 떨어진다. 따라서, 디바이스에서 지원하는 각종 프로토콜이나 응용 소프트웨어 등은 경량(lightweight)이어야 한다. 이러한 특성을 고려하여 본 논문에서는 클라이언트를 구현함에 있어서, 최소한의 필요한 자원만을 사용하도록 하나의 프로세스 형태로 구현하였다. 또한 본 구현은 노트북에서 개발된 것이며 PDA 나 휴대폰과 같은 특정 디바이스에 탑재하기 위해서는 포팅 작업이 필요하다.

<그림 2>는 본 논문에서 구현한 SyncML 클라이언트의 구조를 보여 준다. 기능별로 5 개의 모듈로 구성되었으며, 각 모듈은 다음과 같은 기능을 갖는다. SyncML 그룹에서 제공하는 Palm OS 용 SyncML 클라이언트 데모 버전[7]을 참조하여 구현하였다.



<그림 2> SyncML Client 의 Framework

- Sync Adapter 는 응용 소프트웨어의 요청 또는 SyncML 서버로부터 전송되어지는 메시지를 받아 SyncML Toolkit 과 Sync Agent 로 전달하는 역할을 담당한다.
- SyncML Toolkit 은 SyncML 그룹에서 제공하는 툴킷(toolkit)을 사용하였는데, 서버로부터 전달되어진 메시지를 디코딩(decoding)하고, 클라이언트측에서 보내질 메시지를 인코딩(encoding) 하는 기능을 담당한다[5].
- Sync Agent 는 크게 Command Builder 와 Command Handler 로 구성된다. Command Builder 는 변경된 정보를 포함하여 서버에게 전송할 동기화 메시지를 생성하고, Command Handler 는 서버로부터 전송 받은 메시지를 처리하고 반영한다.

- Open DB Interface 는 SyncML 클라이언트 서버 간 데이터 동기화 작업에 필요한 데이터를 저장하거나 클라이언트 내 저장소와의 인터페이스를 제공한다.

일반적으로 SyncML 은 개인 주소록, 일정 정보와 같은 PIMS 정보 동기화에 이용되는데, 사용자는 디바이스의 아웃룩(Outlook)과 같은 응용 소프트웨어에서 동기화를 요청할 수 있다. SyncML 클라이언트는 사용자 요청을 받아 SyncML 서버와 동기화를 수행하게 된다. 데이터 동기화 수행을 위하여 사용자는 동기화 서버, 인증 방식, 동기화 모드 등에 대하여 선택할 수 있으며, 기본 지정 값으로 수행할 수도 있다.

SyncML 클라이언트는 다음과 같은 개발 환경에서 구현되었다.

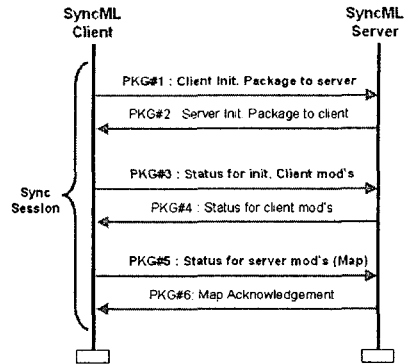
- 운영 체제 : Windows 2000
- 데이터베이스 : MS Access
- 개발 언어 : Visual C++
- 전송 프로토콜 : HTTP

3.2 SyncML 클라이언트 동작

SyncML 클라이언트는 서버와의 데이터 동기화를 위해 ① 메시지 생성, ② 교환, ③ 메시지 처리의 절차를 수행한다.

사용자가 디바이스의 응용 소프트웨어에서 동기화 요청을 하면, SyncML 클라이언트의 Sync Adapter 가 사용자가 지정한 동기화 옵션과 함께 요청을 전달 받게 된다. 사용자는 기본(default)으로 지정된 값으로 요청할 수 있고, 제공되는 UI(user interface)를 통해 동기화 하려는 서버, 동기화 모드, 인증 방식 등을 지정하여 요청할 수도 있다. Sync Adapter 는 서버에게 보내기 위한 메시지를 생성하기 위하여 초기화와 전역 데이터를 생성하고, Sync Agent 의 Command Builder 는 SyncML Toolkit 과 ODBI 인터페이스를 이용하여 메시지에 포함될 커맨드를 생성한다. Sync Adapter 에서 완전한 메시지를 생성하여 Communication Adapter 를 통해 HTTP 전송 프로토콜[6]을 이용하여 서버에게 전달하고, 응답 메시지를 기다린다. Sync Adapter 는 서버로부터 메시지를 받아 Sync Agent 로 보내면 Sync Agent 의 Command Handler 는 변경 정보를 처리하게 된다. 이후 클라이언트는 상태 정보와 변경 정보를 위와 같은 방식으로 서버와 교환함으로써 데이터 동기화를 수행하게 된다.

다음 <그림 3>은 Two-way Sync 모드에서의 SyncML 클라이언트와 서버간 데이터 동기화를 수행에 대한 다이어그램(diagram)이다. 우선 클라이언트와 서버 각각 지정된 동기화 옵션으로 수행하기 위한 초기화를 하고, 양측에서 변경된 데이터와 그 처리 상태 정보를 교환한다.



<그림 3> 동기화 절차 (Two-way Sync)

본 논문에서 구현한 SyncML 클라이언트는 SyncML 프로토콜이 제시한 7 가지 동기화 모드 중 Server Alerted Sync 를 제외한 나머지 모든 동기화 모드를 구현하였다. 기본 모드는 Two-way Sync 이며 또한, 원하는 동기화 모드를 지정하여 수행할 수 있다. 사용자 인증을 위해 Basic 방식과 MD5 방식을 모두 지원한다. 대상이 되는 응용 서비스로서 vCard 2.1 포맷을 따르는 주소록 정보, vCalendar 1.0 포맷을 따르는 일정 정보에 대하여 동기화를 지원한다.

SyncML 메시지는 여러 가지 커맨드들로 구성되어 동기화 정보를 표현할 수 있다. 다음 <표 2> 는 커맨드에 대한 SyncML 그룹의 요구 사항과 구현된 SyncML 클라이언트가 지원하는 정보를 보여 준다. 구현된 클라이언트는 필수적으로 수행해야 하는 모든 커맨드 항목에 대하여 지원 한다.

<표 2> SyncML 클라이언트의 커맨드 지원

커맨드	요구 사항		처리	
	생성	처리	생성	처리
ADD	Should	Must	Yes	Yes
ALERT	Must	Must	Yes	Yes
ATOMIC	May	May	No	No
COPY	May	May	No	No
DELETE	Must	Must	Yes	Yes
EXEC	May	May	No	No
GET	Should	Must	Yes	Yes
MAP	Must	May	Yes	No
MAPITEM	Must	May	Yes	No
PUT	Should	Must	Yes	Yes
REPLACE	Must	Must	Yes	Yes
RESULT	Must	Should	Yes	Yes
SEARCH	May	May	No	No
SEQUENCE	May	May	No	No
STATUS	Must	Must	Yes	Yes
SYNC	Must	Must	Yes	Yes

구현된 SyncML 클라이언트의 기능 검증을 위하여 SyncML 그룹이 정의한 SyncML 적합성 시험 (Conformance test) [8][9]을 수행하였으며, 본 연구팀이 이미 구현한 CNU SyncML Server 3.1 과 상호 연동 테스트를 수행하였다. 클라이언트가 지원 가능한 동기화 모드, 응용 서비스 타입, 인증 방식, 커맨드를 포함하는 다양한 동기화 시나리오를 작성하여 테스트 하였으며, 각 경우에 대하여 모두 정상적으로 기능을 수행하는 것을 확인하였다.

다음의 <그림 4>는 성공적으로 수행된 동기화 테스트에서 생성된 메시지의 일부이다. 클라이언트는 ADD, REPLACE, DELETE 커맨드를 사용하여 변경 정보를 서버에게 전달하는 경우의 예제이다.

```

- <Sync>
  <CmdID>3</CmdID>
  <Target>
    <LocURI>./Address</LocURI>
  </Target>
  <Source>
    <LocURI>./Contacts</LocURI>
  </Source>
  <Add>
    <CmdID>4</CmdID>
    <Meta>
      <Type xmlns="syncml:metinf">text/x-vcard</Type>
    </Meta>
    <Item>
      <Source>
        <LocURI>./1</LocURI>
      </Source>
      <Data>BEGIN:VCARD VERSION:2.1 N:Kim;Youn Soo
        FN:student EMAIL;INTERNET:skellion@lycos.co.kr
        ADR;WORK::cnu;Daejeon;;107-100;Korea
        TEL;WORK;VOICE:822-6025 TEL;HOME;VOICE:042-
        583-7319 TEL;HOME;CELL:017-404-3493
        END:VCARD</Data>
    </Item>
  </Add>
  <Replace>
    <CmdID>5</CmdID>
    <Meta>
      <Type xmlns="syncml:metinf">text/x-vcard</Type>
    </Meta>
    <Item>
      <Source>
        <LocURI>./4</LocURI>
      </Source>
      <Data>BEGIN:VCARD VERSION:2.1 N:Kim;Jin
        FN:student END:VCARD</Data>
    </Item>
  </Replace>
  <Delete>
  </Delete>
</Sync>
<Final />
</SyncBody>
</SyncML>

```

<그림 4> 동기화 메시지 (일부)

4. 결론

본 연구에서는 SyncML 규격을 준수하는 SyncML 클라이언트를 설계하고 구현하였다. 구현된 클라이언트는 디바이스의 특성을 고려하여 하나의 프로세스 형태로 된 경량의 구조를 갖도록 효율적으로 구현되었다. 또한, 기능을 검증하기 위하여 본 연구팀이 이미 구현한 CNU SyncML Server 3.1 과의 다양한 동기화 시나리오에 대한 연동 테스트를 수행하였으며, PIMS 서비스에 대해 효과적으로 데이터 동기를 수행함을 확인하였다.

추후 연구 과제로 구현된 SyncML 클라이언트를 실제 PDA, 휴대용 단말기가 사용하는 Windows ME, Palm 과 같은 운영 체제(Operating System) 기반에서 실행될 수 있도록 변환하여 상용화를 대비할 수 있다.

참고문헌

- [1] SyncML Initiative, <http://www.syncml.org>
- [2] SyncML Initiative, Building an Industry-Wide Mobile Data Synchronization Protocol, SyncML White paper, Mar. 20, 2000
- [3] SyncML Initiative, SyncML Representation Protocol version 1.0.1c, June 15, 2001
- [4] SyncML Initiative, SyncML Synchronization Protocol version 1.0.1, June 15, 2001
- [5] SyncML Initiative, SyncML Architecture, version 0.2, May 10, 2001
- [6] SyncML Initiative, SyncML HTTP Binding, version 1.0.1, June 15, 2001
- [7] SyncML Initiative, SDA2 Specification Version 0.2, Aug. 21, 2000
- [8] SyncML Initiative, SyncML Conformance Test Suite version 0.2, 2001.1.15
- [9] SyncML Initiative, SyncML Manual Test Cases version 0.3, 2001.2.2