

SyncML기반 파일 동기화의 설계 및 구현*

고재두^o, 이훈희, 이윤선, 이상호, 한익주

한국산업기술대학교 컴퓨터공학과

{goldedit^o, solids80, nicelys0612, sangho. ijhan}@kpu.ac.kr

Design and Implementation of a File Synchronization based on SyncML

Jaedoo Go^o, Hunhee Lee, Yunsun Lee, Sangho Lee, Ikjoo Han

Dept. of Computer Engineering, Korea Polytechnic University

요 약

휴대기기의 보편화와 멀티미디어 기술 발달로 인해 MP3 파일 및 동영상 파일, 그리고 다른 여러 가지 파일들이 휴대기기에 저장 및 관리되고, 이러한 장비의 수가 증가함에 따라 개인의 정보 및 데이터는 여러 장치로 분산되었다. 본 논문에서는 최근 보편화되는 휴대기기와 서버간의 데이터 동기화 표준인 SyncML(Synchronization Markup Language)을 소개하고, SyncML에서 정의 되지 않은 파일전송 및 파일동기화 기술을 설계 및 구현한다. 본 논문에서 제안하는 기술은 멀티미디어 기술의 발달로 일반화가 되고 있는 멀티미디어 데이터의 공유에 크게 기여할 것이다.

1. 서 론

휴대기기와 무선 인터넷(Wireless Internet) 환경이 보편화 되면서 개인이 사용할 수 있는 장비의 종류도 다양해졌다. 이러한 장비의 수가 증가함에 따라 개인의 정보 및 데이터는 여러 장치로 분산되었고, 일관성 있게 유지해야 하는 정보 및 데이터도 증가하였다. 여러 장비 또는 이기종 장비에 분산 저장되면서 한쪽 데이터가 변경 될 경우 다른 분산 장치의 데이터에서도 변경 되어야 한다. 즉, 모든 장비의 데이터 동기화가 필요한 것이다. 이러한 데이터 동기화를 위해 개발된 표준 프로토콜이 SyncML(Synchronization Markup Language)이다. SyncML 프로토콜에는 동기화에 필요한 SyncML 동기화 명령어와 동기화 메커니즘이 정의되어져 있다. 그러나 SyncML은 데이터베이스를 기반으로 한 데이터 동기화만을 정의하고 파일에 대한 동기화에 관해서는 논의하지 않고 있다.

따라서, 본 논문에서는 SyncML에서 정의한 명령어들을 기반으로 파일 동기화를 위해 필요한 새로운 동기화 명령어를 정의하고 이를 활용한 동기화 메커니즘을 제안한다.

본 논문의 구성에서 2장은 SyncML에서 정의한 명령어에 관해서 논하고, 3장에서는 파일 동기화를 위한 새로운 명령어를 정의하고 그 기능에 관해서 설명한다. 4장에서는 제안하는 파일 동기화 알고리즘에 관해서 논하고, 5장에서 결론과 함께 향후 연구 과제를 제시한다.

2. SyncML 명령어들

본 장에서는 SyncML에서 정의한 명령어들의 분류와 데이터 동기화를 위한 명령어들에 관해서 논한다.

2.1 SyncML 명령어 분류

SyncML에서 정의한 명령어들[1]은 표1과 같이 분류된다. 요청 명령어는 그 역할에 따라 데이터 동기화 명령어, 초기화 명령어, 장치 정보 관리 명령어, 컨테이너로 다시 분류될 수 있다. 응답 명령어들은 요청 명령어의 수행 결과를 반환하기 위해서 사용되는 것이다.

표1. SyncML 명령어 분류표.

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

2.2 SyncML의 데이터 동기화 명령어들

데이터 동기화를 위해서 SyncML에서 정의한 명령어들은 표2와 같다. 표에서 보는 바와 같이 데이터 동기화에 사

* 본 연구는 산업자원부 공동핵심기술사업(과제번호: 10013815)의 지원을 받았음.

용되는 명령어들은 Add, Copy, Get, Put[4]이다. 이들 명령어들은 모든 데이터가 데이터베이스에 저장되어져 있다는 가정하에 정의되어져 있다. 따라서, 데이터베이스에 저장되어져 있지 않는 파일에 대한 동기화는 지원하지 않고 있다.

표2. SyncML 데이터 동기화 명령어들.

종류	의 함
Add	데이터 컬렉션에 데이터를 저장하는 명령어.
Copy	수신자 데이터베이스 내에서 데이터를 한 위치에서 또 다른 위치로 복사할 것을 요구하는 명령어
Get	동기화하려는 상대편 장치 정보를 자신에게 보내달라고 요청하는 명령어
Put	Put과 Get 명령어를 통해 동기화를 진행할 상대편 장치와 자신의 장치 정보를 교환 한다

3. 파일 동기화 명령어와 그 기능

기존의 SyncML 명령어만을 사용할 경우에는 데이터베이스에 저장된 내용만을 변경 하거나 삭제 할 수 있고, 파일 동기화는 수행할 수 없다[4]. 이런 문제점을 보완하기 위해 본 논문에서는 <File>이라는 명령어를 추가 하였다. <File>명령어는 서버 또는 클라이언트 데이터베이스의 동기화에 사용되는 명령어들과는 달리 실제 파일의 유무를 확인하고 동기화 하는 명령어이다.

<File>명령어는 상위 명령어로 Add, Delete, Copy, Replace[3]등의 명령어를 가질 수 있다. <File>명령어는 단독으로 사용되지는 않고, Add, Delete, Copy, Replace 명령어들과 항상 같이 사용된다. <File>은 상위 명령어의 속성에 따라 <File>명령어의 기능이 변경된다. 예를 들어, <Add>명령어 안에서 <File>명령어를 사용하면 파일명 및 경로는 데이터베이스에 저장되고, 해당 경로에 존재하는 파일이 저장된다. 이 명령어의 사용 예는 그림 1과 같다.

```

<Copy>
<CmdID>12345</CmdID>
<Cred>
  <Meta>
    <Type xmlns='syncml:metinf'>syncml:auth-md5</Type>
    <Format xmlns='syncml:metinf'>b64</Format>
  </Meta>
  <File> KPU.mp3 </File>
</Cred>
<Item>
  <Target><LocURI>mid:kpu1@kpu.com</LocURI></Target>
  <Source>
    <LocURI>10 </LocURI>
  </Source>

```

```

</Item>
</Copy>

```

그림 1. 파일 동기화 명령어 <File>의 사용 예.

그림 1은 SyncML의 명령어 <Copy>내에서 <File>을 사용한 예제이다. 일반적인 <Copy>명령어의 사용예제와 비슷하지만 보통 <Copy>명령어에는 <Cred>명령어 내에 <Data>만이 존재 한다. 그러나 파일 동기화를 위해 <File>명령어를 추가하여 파일동기화를 지원하도록 하였다. <File>은 동기화할 파일의 저장 경로(디렉토리명을 포함한 전체 경로)를 저장하고 있다.

4. 파일 동기화 알고리즘

본 장에서는 파일 동기화를 위한 시나리오와 동기화 과정에 관해서 먼저 논하고, 마지막으로 이를 기반으로 파일 동기화 알고리즘을 제안한다.

4.1 파일 동기화 메시지 및 동기화 과정

파일 동기화 과정에서 데이터베이스 내의 메시지 동기화 과정은 그림 2와 같다. 그림에서 클라이언트는 KPU.avi 파일에 대한 파일동기화를 서버측에 요청하였으며, 서버에서는 동기화 응답과 함께 KPU.zip 파일에 대한 파일동기화를 요청하여 최종 맵 테이블 동기화까지의 Two-Way Sync 과정을 보여주고 있다. <File>명령어를 사용하여 파일을 전송하면 클라이언트 데이터베이스에 저장하고 있는 파일 저장 경로를 서버 데이터베이스에 저장하고 있는 파일 저장 경로로 파일을 전송 하게 된다.

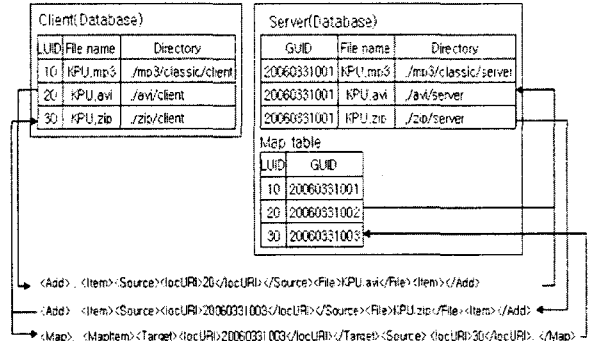


그림 2. 파일 동기화 과정.

4.2 파일 동기화 시나리오

그림 3에서는 Two-Way Sync의 메시지 교환에서 <File>명령어가 몇 번째 패킷에 담겨 동기화가 일어나는지 보여주고 있다. 패킷 1번과 2번에서는 동기화 과정이 수행되기 전 클라이언트에서 서버로 장치 정보에 대한 패킷1번을 보내면 서버에서는 패킷2번으로 1번 패킷에 대한 응답 정보와 함께 서버의 장치 정보를 클라이언트에게 전송한다. 1번과 2번 패킷은 동기화가 수행되기 전 서버와 클라이언트

사이의 인증을 위한 패킷이다. 실제 동기화가 수행되는 패킷은 3번과, 4번이다. 클라이언트에서 서버로 <Add> 명령어에 <File>KPU.avi</File>을 포함하여 보내면 서버는 데이터베이스에 <Add>에 대한 내용을 추가 하고 실제 데이터베이스에 저장하고 있는 파일 경로를 추가 시키며, 4번 패킷은 서버에서 클라이언트로 <Add>동기화를 할 내용(<File>KPU.zip</File>)과 3번 패킷에 대한 응답 메시지를 함께 보내게 된다. 마지막 5번, 6번 패킷은 변경된 내용을 Map Table에 적용시키기 위한 패킷이다. 6번의 메시지 교환으로 동기화 및 파일 전송이 완료된다.

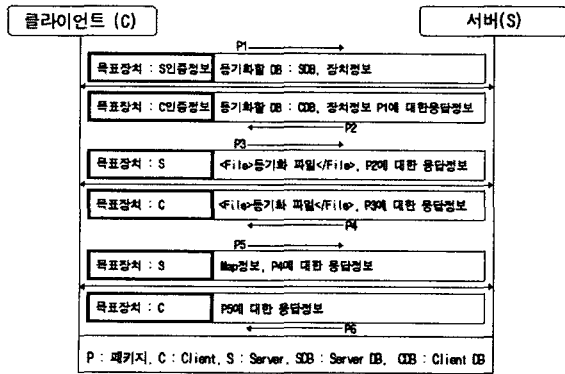


그림 3. 파일 동기화 시나리오.

4.3 파일 동기화 알고리즘

그림 4는 파일 동기화 알고리즘을 보이고 있다. 그림에서와 같이 <File>명령어는 <Add>, <Delete>, <Copy>, <Replace>[3] 명령어 안에서 상위 명령어의 동작에 따라 기능이 결정된다. 클라이언트의 상위<Copy>명령어 내에서 복사할 파일을 찾아 <File>명령어를 이용해 파일의 전송을 지시한다.

<File>명령어를 실행하면 LUID, GUID, MapTable, ChangeLog를 이용하여 클라이언트, 서버의 데이터베이스에 기록된 내용이 각각 복사, 수정되며, 그에 따른 실제 파일이 전송되어 데이터베이스 및 파일의 동기화가 완료되며, MapTable 과 ChangeLog의 내용을 수정하는 것을 마지막으로 동기화 과정을 종료한다.

5. 결론

본 논문에서는 SyncML에서 정의 되지 않은 파일 동기화 명령어 <File>과 파일 동기화 알고리즘을 제안하였다. <File>명령어는 서버 측, 또는 클라이언트 측의 저장장치에 저장되어 있는 미디어 파일, 문서 파일등 동기화 하고자 하는 파일의 유무를 판단한 뒤 <File>명령어를 이용하여 동기화 한다.

논문에서 제안하는 <File>명령어는 기존 명령어들이 데이터베이스의 내용을 검색, 추가, 수정, 삭제 하여 동기화 하는 명령어들로 구성되어 있는 반면, 실제 파일을 동기

화 할 수 있다는 점에서 현재 데이터베이스의 동기화에만 사용가능한 SyncML의 활용 범위를 확대할 수 있다.

향후 SyncML 통합개발 환경의 필요성에 대한 연구가 필요할 것이며, SyncML에 대한 지속적인 표준화 작업이 필요할 것이다.

```
File (SyncML File)
{
  클라이언트 to 서버 장치 및 C데이터베이스파일 정보전송
  서버 to 클라이언트 장치 및 S데이터베이스파일 정보전송

  if(클라이언트 데이터베이스 정보 변경)
  {
    변경된 내용을 서버로 전송 및 File전송
    if(File == New File)
    {
      서버 데이터베이스에 MapTable 생성
      적용된 MapTable 내용 및 GUID 클라이언트로 전송
    }
  }

  if(서버 데이터베이스 정보 변경)
  {
    변경된 내용을 클라이언트로 전송 및 File전송
    If(File == New File)
    {
      클라이언트 데이터베이스에 내용 저장 retrun LUID
    }else{
      클라이언트 데이터베이스 내용 변경
    }
  }

  변경된 파일 없음(종료)
}
```

그림 4. 파일 동기화 알고리즘.

6. 참고 문헌

- [1] 하인숙, 조재혁, 양지현, "데이터 동기화의 표준 SyncML 기초 다지기" 마이크로 소프트웨어 2001년 5월, p.330-340, May, 2001.
- [2] 하인숙, 조재혁, 양지현, "SyncML 레퍼런스 툴킷 그 내부를 보자" 마이크로 소프트웨어 2001년 7월, p.324-336, Jul, 2001.
- [3] www.syncml.org의 Syncml Representation Protocol 1.1
- [4] www.syncml.org의 Syncml Representation Protocol 1.0