

다양한 이동통신 단말기들 간의 상호 운용을 위한 SyncML 기반의 통합 자료 동기화 게이트웨이 시스템

(A Gateway System for Integrated Data Synchronization
Based on SyncML for Interoperability Between Various
Mobile Terminals)

장 대 진 [†] 박 기 현 ^{**}
(DaeJin Jang) (KeeHyun Park)

요 약 무선 이동통신 단말기와 중앙 서버에 각각 저장되어 있는 자료에 대한 동기화는, 효율적인 이동통신 환경에서는 반드시 필요한 작업이다. 그러나, 이동통신 단말기 제조회사들은 자기 고유의 자료 동기화 방식을 사용하고 있기 때문에, 제조회사가 다른 단말기들 간의 자료에 대한 상호 운용을 보장할 수 없다.

본 논문에서는 OMA(Open Mobile Alliance)의 공개 표준안인 SyncML(Synchronization Markup Language) 자료 동기화 방식을 기반으로 하는 통합 자료 동기화 게이트웨이 시스템을 설계하고 구현하였다. 즉, 서로 다른 동기화 방식의 자료를 SyncML 방식의 자료로 변환하는 통합 자료동기화 게이트웨이 시스템을 구현함으로써, 서로 다른 이동통신 단말기들 간의 상호 운용을 가능하게 하였다. 본 논문에서 구현한 통합 자료 동기화 게이트웨이 시스템을 사용하면, WinCE 기반 ActiveSync 방식의 자료, PalmOS 기반 HotSync 방식의 자료 및 OMA의 SyncML 방식의 자료들 간의 상호 운용이 가능하다.

키워드 : 자료 동기화, 자료 변환, 통합 게이트웨이 시스템, OMA, SyncML, ActiveSync, HotSync

Abstract Data synchronization between data stored in mobile communication terminals and data in a central server is one of essential tasks for efficient mobile communication environments. Since, however, manufacturers of mobile terminals have their proprietary data synchronization mechanisms, data interoperability can hardly be achieved.

In this paper, an integrated data synchronization gateway based SyncML is designed and implemented. SyncML data synchronization is one of open standards proposed by OMA(Open Mobile Alliance). In other words, by constructing a data synchronization gateway which can transform various proprietary data into SyncML data and vice versa, data interoperability between mobile terminals which use different data synchronization mechanisms can be achieved. With an integrated data synchronization gateway system implemented in this work, interoperability among WinCE-based ActiveSync data, PalmOS-based HotSync data and OMA SyncML data can be achieved.

Key words : Data Synchronization, SyncML, Data Transformation, Integrated Gateway System, OMA, SyncML, ActiveSync, HotSync

· 본 연구는 산업자원부 지방기술혁신사업(RTI04-03-02) 지원으로 수행되었음

[†] 정 회 원 : 계명대학교 컴퓨터공학과
djjang19@kmu.ac.kr

^{**} 종신회원 : 계명대학교 컴퓨터공학과 교수
khp@kmu.ac.kr

논문접수 : 2006년 10월 26일

심사완료 : 2008년 1월 24일

Copyright © 2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 테러 제14권 제2호(2008.4)

1. 서론

최근 이동컴퓨팅 기술과 하드웨어 기술의 발달로 인해 무선 이동통신 단말기가 매우 급속하게 보급되고 있다. 또한, 무선 이동통신 단말기는 일정관리나 전자메일 교환과 같은 개인적인 용도에서 기업 운영이나 업무의 효율성을 향상시키는 비즈니스 영역에 대한 새로운 수단으로 그 역할이 확대되고 있다[1,2].

하지만, 무선통신 환경이 가지고 있는 네트워크 서비스 범위의 한계 및 연결 비용에 따른 경제적인 효율성 때문에 각각의 단말기들은 통합 관리서버와 항상 연결 상태를 유지할 수 없다. 또한, 무선 이동통신 단말기 사용자는 시간이나 장소에 구애받지 않고 원하는 최신의 정보를 이용하기 위해서는 중앙 통합 서버의 자료와 무선 이동통신 단말기의 자료를 일치시키는 작업이 필요하며, 이를 자료 동기화(Data Synchronization)라고 한다. 현재 주요 단말기 제조 회사에서 제공하는 자료 동기화 방식이 존재하지만, 각각의 제조회사별 단말기와 응용 서비스 간의 호환성이 결여되어 있다. 따라서, 각 단말기 회사에서 제공되는 동기화 방식의 상호 운용성을 보장하기 위해서 2000년 2월 여러 단말기 회사들이 주축이 된 OMA(Open Mobile Alliance)에서 SyncML 방식을 제안하여 공개적인 표준화를 시도하고 있다. 하지만, 각 단말기 제조회사들이 기술적인 경쟁을 통해서 시장점유율을 높이고 있기 때문에, 각각의 동기화 방식에 대한 기술적인 공개를 하지 않고 있다. 따라서, SyncML이 자료 동기화 방식에 대한 표준화를 시도하고 있지만, 각각의 동기화 방식을 탑재한 무선 이동통신 단말기들이 모두 SyncML 방식을 동기화 표준화로 대체하기는 어려운 실정이다[1-3].

이러한 현황에 비추어 서로 다른 동기화 방식이 탑재된 단말기들 간의 자료 동기화에 대한 상호 운용성을 보장하기 위한 해결 방안으로써 다음과 같은 방법을 제시할 수 있다. 첫째, 통합 중앙 서버의 동기화 대상 자료를 관리하는 데이터베이스 시스템에 특화된 단말기용 솔루션을 탑재함으로써, 동일한 솔루션을 탑재하고 있는 각각의 단말기들은 상호간에 자료 동기화를 수행할 수 있게 된다. 둘째, 각 단말기 제조회사에서 제공하는 서로 다른 동기화 방식을 수용할 수 있는 변환 방식의 통합 게이트웨이 시스템을 들으로써, 각각의 단말기들이 상호간에 자료 동기화를 수행할 수 있게 된다.

본 논문에서는 각 단말기 제조회사에서 제공하는 서로 다른 동기화 방식을 수용할 수 있는 변환 방식의 통합 게이트웨이 시스템을 제안하였다. 본 논문에서 제안한 시스템은 SyncML 프로토콜을 근간으로 어플리케이션 변환 방식의 통합 게이트웨이 시스템을 설계하고 구

현하였다. 즉, OMA에서 공개 표준안으로 제안한 SyncML 자료 동기화 방식을 기준으로 하여, 다른 자료 동기화 방식들(ActiveSync 및 HotSync)의 자료를 SyncML 자료로 변환시키는 통합 자료 동기화 게이트웨이 시스템을 구현함으로써, 서로 다른 단말기들 간의 상호 운용을 가능하게 하였다. 또한, 본 논문에서 제안된 시스템은 새롭게 추가되는 자료 동기화 방식뿐만 아니라, 일반적인 휴대폰에도 적용할 수 있는 시스템 확장성을 가지고 있다.

본 논문에서 제안된 시스템이 해당 연구 분야에서 기여하는 점은 다음과 같다. 첫째, 기존의 학술적인 연구 사례 및 상업적인 용도로써 변환 게이트웨이 방식의 SyncML 기반 통합 자료 동기화 시스템에 대한 구체적인 개발 사례를 제시한 것이다. 동기화 대상 서버에 특화된 각각의 단말기용 솔루션을 탑재함으로써, 이기종 단말기들 간의 자료 동기화를 수행할 수 있는 제품들은 출시되고 있지만, 이에 대한 기술적 및 성능적인 측면에 대한 분석 결과는 아직까지 학술적으로 발표되지 않았다. 따라서, 서로 다른 방식을 탑재하고 있는 무선 단말기 자료 동기화 방식에 대한 상호 호환성을 보장하기 위한 또 다른 방법으로써 그 기초를 제공하고 있다. 둘째, OMA에서 자료 동기화 방식에 대한 표준으로 SyncML 방식을 제안하였다. 본 논문에서는 SyncML 방식을 근간으로 한 통합 자료 동기화 게이트웨이 시스템의 개발 사례를 제시함으로써, SyncML 방식에 대한 기능 및 성능적인 적합성에 대해서 검증할 수 있다. 따라서, 실제로 사용 가능한 SyncML 기반의 통합 자료 동기화 게이트웨이 시스템을 개발함으로써, SyncML 방식의 확산 및 국제 표준화에 기여할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구 및 사례 연구에 대한 결과를 제시하고 있으며, 3장에서는 본 논문에서 제안한 시스템의 구조에 대해서 제시하고, 4장에서는 구현 및 검증 결과에 대해서 제시한다. 마지막으로, 5장에서는 결론을 제시하고 향후 연구 방향에 대해서 논의한다.

2. 관련 연구

2.1 자료 동기화 프로토콜

2.1.1 SyncML(Synchronization Markup Language)

SyncML은 이동무선통신 단말기와 서버간의 원격 동기화를 목표로 설계되었으나, 로컬 동기화와 유선 네트워크로 연결된 장치들 간의 동기화에도 사용될 수 있다. SyncML을 기반으로 하는 클라이언트와 서버 간에 동기화가 수행되는 동안, 동기화 대상이 되는 객체는 논리적으로 패키지를 교환한다. 또한, 실제로 각 패키지는 이동무선통신 단말기의 제한된 자원과 무선 통신상의

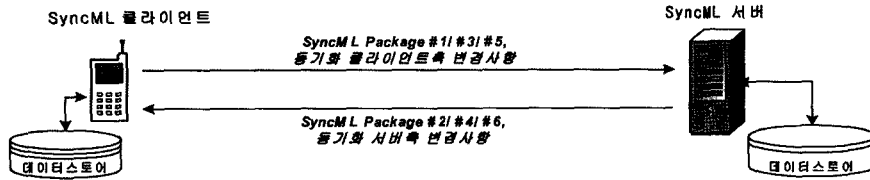


그림 1 SyncML 기반의 클라이언트-서버간의 자료 동기화

낮은 대역폭으로 인해 다수의 메시지들로 분리되어 교환된다. 한 번의 완전한 자료 동기화를 위해 클라이언트와 서버 간에 주고받은 모든 패키지를 세션(Session)이라고 하며, 세션은 여러 개의 패키지를 포함하고 하나의 패키지는 여러 메시지를 포함한다. 그림 1은 SyncML을 기반으로 하는 클라이언트와 서버 간의 자료 동기화를 도식화하고 있다[1,2].

그림 1에서 SyncML 클라이언트가 먼저 서버에게 갱신된 자료를 포함한 SyncML 메시지를 전송하면, SyncML 서버는 클라이언트가 요청한 동기화 타입에 의해 서버측 자료와 동기화 작업을 수행한 후, 다시 클라이언트에게 작업 결과 및 서버 자신의 변경된 자료도 전송한다. 이러한 수차례의 패키지 교환을 수행함으로써 자료 동기화가 이루어지며, 클라이언트와 서버 간의 해당 자료에 대하여 일치성을 보장하게 된다.

SyncML 자료 동기화 규격은 자료 동기화를 위해 교환되는 XML 기반의 자료 표현(Data Representation) 프로토콜, SyncML 동기화(Synchronization) 프로토콜 그리고 전송 바인딩(Transport Bindings) 프로토콜로 구성되어 있다. SyncML 자료 동기화 규격의 중요한 두 부분은 SyncML 자료 표현 프로토콜과 동기화 프로토콜이다. SyncML 규격을 구성하고 있는 각 부분의 기능은 다음과 같다[1,4,5].

첫째, SyncML 자료 표현 프로토콜은 자료 동기화를 위해 교환되는 SyncML 메시지의 논리적인 구조와 형태를 XML 형식으로 정의하고 있다. 또한, 각각의 필드가 어떠한 정보를 담고 있으며 해당 정보가 어떤 의미를 내포하는 것인지에 대한 약속을 정의하고 있다.

둘째, SyncML 동기화 프로토콜은 SyncML 클라이언트와 서버 간에 이루어지는 자료 표현 프로토콜 규격에 의해 생성된 자료의 추가, 삭제, 갱신과 같은 동기화 명령과 그 밖의 상태 정보에 대한 메시지가 교환되는 방법에 대해서 정의하고 있다.

셋째, 전송 바인딩 프로토콜은 여러 네트워크 프로토콜을 이용해 메시지를 전송할 때 필요한 요구사항에 대한 규격을 제시하고 있다.

2.1.2 ActiveSync 프로토콜

ActiveSync 프로토콜은 WinCE를 기반으로 하는 무

선 단말기에서 동작하는 자료 동기화 방식이다. ActiveSync 제공자와 ActiveSync 서비스 매니저로 구성되어 있으며, 투웨이 싱크(Twoway-Sync)와 슬로우 싱크(Slow-Sync)에 대한 동기화 모드를 지원하고 있다. 하지만, ActiveSync 프로토콜은 하위 프로토콜에 대한 기술적인 내부 규격에 대하여 공개하고 있지 않다[6,7].

2.1.3 HotSync 프로토콜

HotSync 프로토콜은 Palm OS를 기반으로 하는 단말기에서 동작하는 자료 동기화 방식이다. 또한, HotSync 프로토콜은 SyncML 프로토콜의 투웨이 싱크와 유사한 슬로우 싱크와 패스트 싱크(Fast-Sync) 두 가지 자료 동기화 모드를 지원하며, 패스트 싱크가 슬로우 싱크보다 속도가 빠르기 때문에, 슬로우 싱크보다 패스트 싱크를 우선적으로 수행한다. 또한, HotSync 프로토콜은 ActiveSync 프로토콜과 마찬가지로 하위 프로토콜에 대한 기술적인 내부 규격에 대하여 공개하고 있지 않다[8,9].

2.2 변환 게이트웨이 방식(10,11)

2.2.1 프로토콜 변환 게이트웨이 방식

SyncML 프로토콜 기반의 동기화 서버와 각 단말기 제조업체의 동기화 방식을 탑재한 클라이언트 간의 송·수신되는 프로토콜을 변환한다. 즉, 양 끝단의 중간 매개체인 게이트웨이는 송신측으로부터 수신된 메시지에 대해서 메시지를 구성하고 있는 순수한 자료에 대해서 다른 한 쪽의 목적지 프로토콜 규격에 맞게 메시지가 재구성되기 전에 송신측 전송 방식의 프로토콜 헤더 정보를 분석한 후, 게이트웨이에서 전송 메시지 재구성에 필요한 정보만을 추출한 뒤, 프로토콜 헤더를 제거한다. 그 후, 추출된 정보를 바탕으로 최종 목적지의 새로운 전송 메시지 형태로 재구성한 후 전송한다. 이러한 프로토콜 변환 방식은 서로 다른 통신 프로토콜을 사용하고 있는 양 끝단의 장치들 간의 자료 교환을 위한 변환 방식 중에서 가장 효율적이고 적합한 방식이다.

본 논문에서 제안된 통합 게이트웨이 시스템 방식에 이러한 프로토콜 변환을 위해서는 SyncML 프로토콜 방식과 같이 각각의 동기화 프로토콜에 대한 하위 Bearer의 기술적인 공개가 요구된다. 하지만, ActiveSync 프로토콜 및 HotSync 프로토콜 방식은 프로토콜

내부 계층에 대한 공개가 이루어져 있지 않으므로, 본 논문에서 제안한 통합 게이트웨이 시스템의 적용 방식에는 적합하지 않다.

2.2.2 자료 변환 게이트웨이 방식

자료 변환 게이트웨이 방식은 자료 동기화 방식이 서로 다른 두 시스템들 간의 주고받는 데이터의 변화를 통하여 게이트웨이 역할을 수행한다. 교환되는 자료의 표준은 연락처 정보의 포맷을 규정한 vCard, 일정 정보의 포맷을 규정한 vCalendar 등과 같은 동기화 대상의 데이터 타입에 대한 표준이 존재한다. 변환 게이트웨이는 각 단말기에서 전송되는 모든 규격 문서에 대해서 분석 및 새로운 문서를 생성할 수 있는 모듈이 필요하다.

이러한 변환 방식은 각 동기화 방식이 동작하는 플랫폼에 의거한 자료의 규격을 알고 있어야 한다. 또한, 서로 다른 플랫폼을 탑재하고 있는 모든 단말기에서 생성된 자료 규격에 대한 분석 및 새로운 규격에 적합한 자료의 생성에 대한 모든 처리를 변환 게이트웨이에서 수행함으로써, 자료 변환에 따른 동기화 수행 시간의 증가로 인하여 변환 게이트웨이의 오버헤드 현상이 발생할 수 있다.

2.2.3 어플리케이션 변환 게이트웨이 방식

어플리케이션 변환 방식은 서로 다른 동기화 방식을 수용하기 위해서, 무선 단말기 및 변환 게이트웨이에 개별적인 기능을 수행하기 위한 독립적인 프로그램을 탑재하고 있다. 단말기에 탑재된 프로그램은 동기화 대상이 되는 자료를 추출하기 위한 모듈과 변환 게이트웨이와 메시지를 교환하기 위한 통신 모듈을 탑재하고 있다. 변환 게이트웨이 측에는 단말기와 자료 교환을 위한 통신 모듈 및 단말기에서 추출한 자료를 바탕으로 생성된 메시지에 대한 분석 및 생성 모듈을 가지고 있다. 또한, 자료 교환 대상 서버와 통신을 하기 위한 클라이언트를 기본 모듈로 가지고 있다.

통합 게이트웨이는 무선 단말기를 위한 서버 기능의

모듈과 동기화 서버를 위한 클라이언트 기능의 모듈을 모두 가지고 연결고리 역할을 수행한다. 따라서, 본 논문에서 제안된 SyncML 기반의 통합 자료 동기화 게이트웨이 시스템은 SyncML 클라이언트 모듈 및 무선 단말기를 위한 서버 모듈을 개발하여 탑재하고, 무선 단말기 측에는 자료 추출 기능을 수행하도록 개발한 모듈을 탑재함으로써, 각각의 모듈이 독립적으로 동작하여 변환 기능을 수행하는 어플리케이션 변환 방식을 수용할 수 있다.

2.3 사례 연구

2.3.1 IntelliSync 프로토콜

Pumatech사에 의해서 개발된 IntelliSync 방식은 동기화 시간을 최소화하기 위하여 FastSync 모드만 지원한다. 이 방식은 단일의 중앙 관리 서버가 모든 동기화 클라이언트들과 동기화를 수행하며, 각 단말기에 대한 계정 및 보안 설정에 대해서도 관리를 담당한다[3].

IntelliSync 방식은 항상 동일한 서버와 동기화를 수행하기 때문에, 패스트 싱크(Fast-Sync) 모드에 특성화되어 있다. 따라서, 다수의 단말기들이 단말기들 간의 동기화를 수행할 수는 없지만, 각각의 단말기들은 하나의 중앙 서버에 연결되어 있으므로, 각 단말기들의 변경 사항에 대한 기록을 가질 수 있다. 그림 2는 IntelliSync 방식의 동기화 서비스 개념에 대한 구조를 보여주고 있다.

IntelliSync 방식은 Microsoft Outlook, Exchange 서버 그리고 Palm PDA, Pocket PC 및 Symbian Handheld가 포함된 무선 단말기들과 동기화 서버 간의 동기화가 가능하다. 하지만, 이 방식은 일반적인 중앙 집중 모델의 단점을 가지고 있다. 즉, 중앙 서버의 고장 및 통신 두절로 인하여 모든 동기화 클라이언트의 동기화 서비스 불능, 그리고 동기화 클라이언트 수의 증가에 따른 네트워크 트래픽의 가중과 같은 취약점을 가지고 있다.

2.3.2 오라클(Oracle) Partners 솔루션

오라클사와 기술 제휴를 체결한 국내·외 업체에서

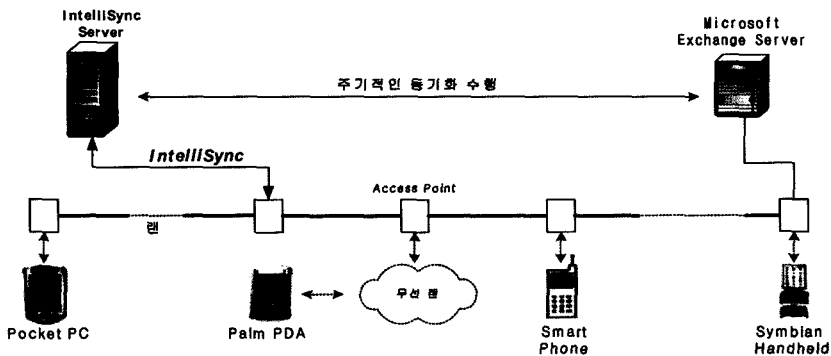


그림 2 IntelliSync 방식의 서비스 개념

오라클 DBMS를 이용하여 구축된 동기화 대상 서버와 다양한 무선 단말기 간의 자료 동기화 기능을 제공하고 있다. 본 솔루션은 변환 방식의 게이트웨이를 통하여 대상 서버와 자료 동기화를 수행하지 않고, 오라클을 이용하여 구축된 통합 솔루션 서버와 연동할 수 있는 특화된 프로그램을 각 단말기에 탑재하여 자료 동기화를 수행하고 있다. 따라서, 게이트웨이 변환 방식에 따른 불필요한 오버헤드로 발생할 수 있는 자료 동기화 시간에 대한 지연 현상이 없으며, 통신의 양 끝단 장치인 동기화 클라이언트와 서버가 직접적으로 정보교환을 수행함으로써, 더 높은 통신 신뢰성을 가지고 있다[17].

하지만, 본 솔루션은 오라클 DBMS로 구축된 동기화 서버와 자료 동기화를 수행해야 하는 한계점을 가짐으로써, 다른 플랫폼으로 구축된 동기화 서버와 서로 다른 방식의 무선 단말기 간의 동작은 지원하지 않는다. 또한, WinCE 운영체제를 탑재하고 있는 PDA, 스마트폰과 같은 무선 단말기에 탑재할 수 있는 솔루션만을 제공하고 있으므로, 다양한 플랫폼에서 동작할 수 있는 단말기용 솔루션의 확장이 요구된다.

2.3.3 Synchronica SyncML 게이트웨이

SyncML 방식이 탑재된 무선 단말기에서 동작하는 SyncML 기반 통합 솔루션이며, 자료 동기화 기능과 DM (Device Management) 기능도 포함하고 있다. SyncML 게이트웨이 서버는 Microsoft Exchange 서버 혹은 Sun JES 서버와 무선 단말기 간의 SyncML 방식의 동기화 서비스를 제공한다. 그림 3은 Synchronica SyncML 게이트웨이의 서비스 개념에 대해서 보여주고 있다[13].

Synchronica 솔루션은 PalmOS 기반의 PDA 및 WinCE 기반의 Pocket PC에서 동작하기 위한 단말기용 SyncML 클라이언트 모듈을 제공하며, SyncML 게이트웨이를 통하여 MS Exchange 서버와 Sun JES 서

버에 저장된 연락처 정보 및 일정 정보와 같은 PIMs (Personal Information Management에 대해서 자료 동기화가 가능하다.

본 솔루션은 동기화 대상이 되는 무선 단말기 및 동기화 서버에 SyncML 규격을 준수하는 특화된 프로그램을 탑재하고 있다. 또한, Synchronica SyncML 게이트웨이는 다양한 플랫폼을 탑재하고 있는 무선 단말기와 SyncML 동기화 서버 간의 변환 기능을 수행하지 않고, 다양한 플랫폼으로 구축된 동기화 서버에 저장된 대상 자료를 SyncML 문서로 변환함으로써, 다양한 플랫폼 상에서 SyncML 방식을 탑재하고 있는 무선 단말기 간의 자료 동기화를 수행한다.

3. 제안 시스템

제조회사별 무선 이동통신 단말기에 탑재된 동기화 방식들은 하위 Bearer 계층의 프로토콜에 대한 규격 및 원리를 공개하고 있지 않음으로써, 각각의 동기화 프로토콜에 대한 변환에 어려움이 있다. 따라서, 본 논문에서는 각 단말기의 동기화 프로토콜의 운영 기반의 되는 PalmOS와 WinCE 운영체제에서 제공하는 API를 이용하여, 어플리케이션 수준의 변환 게이트웨이 방식인 SyncML 기반의 통합 자료동기화 게이트웨이를 설계 및 구현하였다. 본 논문에서 제안된 SyncML 기반의 통합 자료 동기화 게이트웨이 시스템은 무선 단말기 영역과 변환 게이트웨이 영역으로 구성되어 있다. 그림 4는 SyncML 기반의 통합 자료 동기화 게이트웨이 시스템 구조를 나타내고 있다.

3.1 변환 게이트웨이 영역

변환 게이트웨이는 통신 모듈을 통해서 클라이언트와 메시지를 교환하고, 서버 기능 모듈을 통해 수신된 메시지를 분석 및 송신할 메시지를 생성하게 된다. 이후, 동기화 프로토콜 매니저에 의해 송수신 메시지의 대상

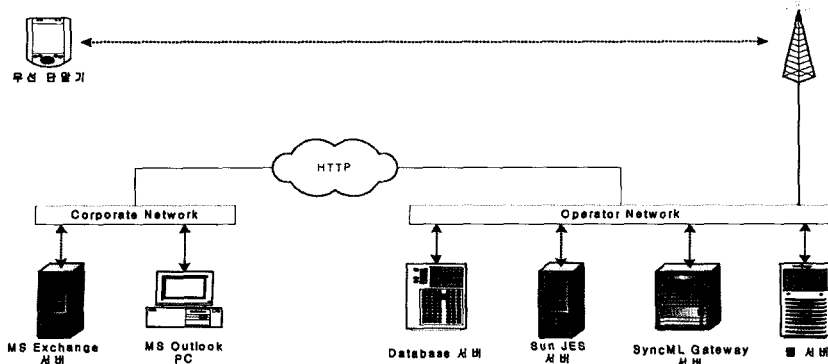


그림 3 Synchronica SyncML 게이트웨이의 서비스 개념

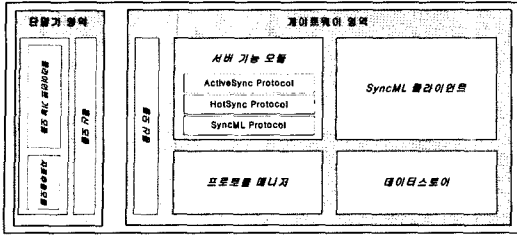


그림 4 통합 자료 동기화 게이트웨이 시스템 구조

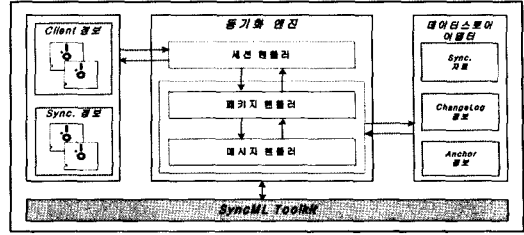


그림 6 SyncML 클라이언트의 구조

이 되는 동기화 자료에 대해서 추가, 삭제 및 수정 명령을 데이터스토어에 요청하고, SyncML 클라이언트에 의해 해당 메시지를 SyncML 서버와 교환하게 된다.

서버 기능 모듈은 동기화 클라이언트에서 요청한 동기화 메시지를 분석하고, 이에 대응하는 송신 메시지를 생성하며, 메시지 분석기와 메시지 생성기로 구성되어 있다. 그림 5는 서버 기능 모듈의 두 구성 요소에 대한 처리 흐름을 보여주고 있다.

자료 동기화를 요청하는 무선 단말기의 자료 추출 모듈에 의해 추출된 동기화 대상 자료는 클라이언트 기능 모듈에 의해 전송 메시지로 구성되어 통합 게이트웨이로 전송된다. 이러한 메시지를 통신 모듈을 통해 수신한 서버 기능 모듈은 메시지 분석기에 SyncML 문서로 변환하기 위한 정보와 실제 데이터를 조작하기 위한 정보로 분석 한 후, 메시지 생성기로 넘겨주게 된다. 이와는 반대로, SyncML 동기화 서버에서 수신된 SyncML 문서를 수신 한 후, 메시지 분석기는 해당 단말기에 적합한 메시지로 생성하기 위한 정보로 분류하고, 이 정보를 메시지 생성기로 넘겨주게 된다. 메시지 생성기는 메시지 분석기로부터 받은 정보를 바탕으로 SyncML 문서를 생성하기 위한 메시지 및 단말기 적합한 메시지 형태로 생성하게 된다.

SyncML 클라이언트는 동기화 엔진, 데이터스토어 어

댑터, 클라이언트 정보, 동기화 정보 그리고 SyncML 툴킷 부분으로 구성되어 있다[14]. 그림 6은 SyncML 클라이언트의 구조를 보여주고 있다.

SyncML 클라이언트는 변환 게이트웨이 영역에서 가장 핵심적인 역할을 담당하고 있다. SyncML 클라이언트는 동기화 대상 데이터가 포함된 동기화 명령어를 생성 및 SyncML 서버로부터 수신된 동기화 명령어를 처리하고, 해당 데이터를 데이터스토어에 반영한다. 또한, SyncML 클라이언트의 동기화 엔진 부분은 동기화 단위에 따라 세션 핸들러(Session Handler), 패키지 핸들러(Package Handler), 메시지 핸들러(Message Handler) 세 부분으로 구성되어 있다. 세션 핸들러는 통신 세션에 대한 제어 기능과 동기화 메시지 헤더 부분을 생성하는 기능을 담당하며, 패키지 핸들러는 SyncML 메시지의 Syncbody 부분에 대한 논리적 구성을 담당한다. 또한, 메시지 핸들러는 자료 동기화를 위해 필요한 XML 명령어를 실제로 생성해 주는 기능을 담당하고 있으며, 서버로부터 수신된 SyncML 메시지를 받아 처리 및 분석하는 역할을 담당한다.

데이터스토어 어댑터는 동기화 엔진 모듈의 각 핸들러와 자료 사이를 연결하는 인터페이스 역할을 한다. 클라이언트 정보 모듈은 동기화 대상이 되는 목적지 주소 및 클라이언트 장치의 정보와 동기화 대상인 클라이언

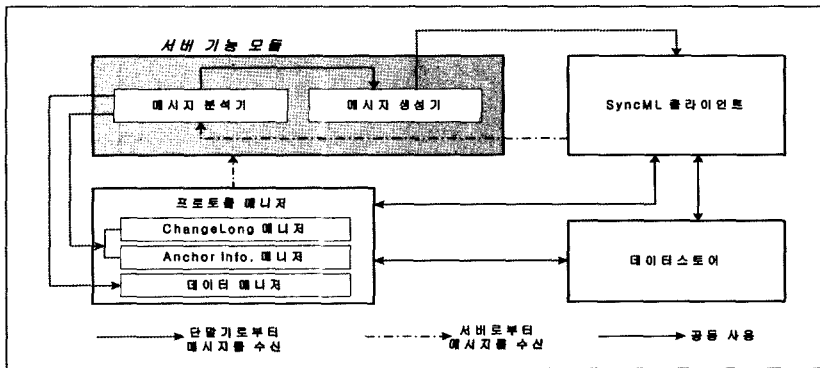


그림 5 서버 기능 모듈의 처리 흐름

트 주소에 대한 정보도 포함하고 있으며, 동기화 정보 모듈은 동기화 종류에 대한 정보 및 이전의 동기화 수행과 현재 수행하려 하는 동기화 간의 상호 일치성을 검사하기 위한 앵커(Anchor) 정보를 담고 있다. 또한, SyncML 툴킷은 서버로부터 수신된 메시지를 디코딩하고, 클라이언트에서 전송할 메시지를 인코딩하는 기능을 담당하며 전송 프로토콜간의 실제 메시지의 송신과 수신을 담당한다.

프로토콜 매니저는 해당 장치의 현재 데이터와 이전 동기화의 상태를 비교하여 ChangeLog를 갱신하고 그 정보를 반환한다. 동기화가 완료되면 Anchor 정보를 갱신한 후, Sync. Protocol Manager는 실제 데이터스토어에 접근하여 동기화의 대상이 되는 데이터를 제어한다. 그림 7은 프로토콜 매니저의 처리 흐름을 보여준다.

SyncML 클라이언트가 사용자 정보와 앵커 정보를 데이터스토어 어댑터에게 요청하면 데이터스토어 어댑터는 프로토콜 매니저에게 해당 장치의 사용자 정보와 앵커 정보를 요청하여 인증 과정을 거치게 된다. 또한,

현재 동기화 대상이 되는 아이템을 결정하기 위해 변경 기록 로그(ChangeLog)를 갱신하기 위하여 프로토콜 매니저는 해당 장치의 현재 데이터와 이전 동기화의 상태를 비교하여 변경 기록 로그를 갱신하고, 그 정보를 반환한다.

프로토콜 매니저는 해당 디바이스의 현재 데이터와 이전 동기화의 상태를 비교하여 변경 기록 로그를 갱신하고 그 정보를 반환한다. 동기화가 완료되면 앵커(Anchor) 정보를 갱신한 후, 프로토콜 매니저는 실제 데이터스토어에 접근하여 데이터를 삽입, 삭제, 갱신함으로써 동기화의 대상이 되는 데이터를 제어한다. 그림 8은 동기화 수행결과로써 반영되는 자료에 대한 프로토콜 매니저와 데이터스토어 간의 데이터 제어 원리를 보여주고 있다.

데이터 체크 모듈과 데이터 제어 모듈은 실제 동기화에 이용되는 데이터에 접근하여 데이터의 비교, 추가, 삭제 및 수정과 같은 동작을 수행하게 된다. 데이터 체크 모듈은 변경 기록 로그에 기록된 데이터의 동기화

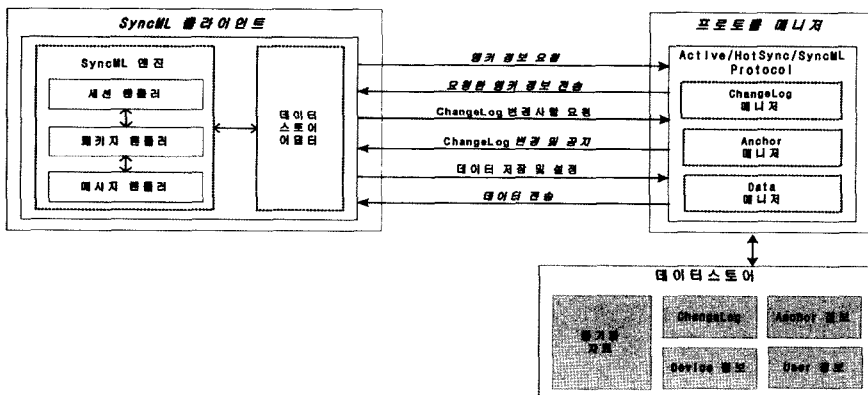


그림 7 프로토콜 매니저의 처리 흐름

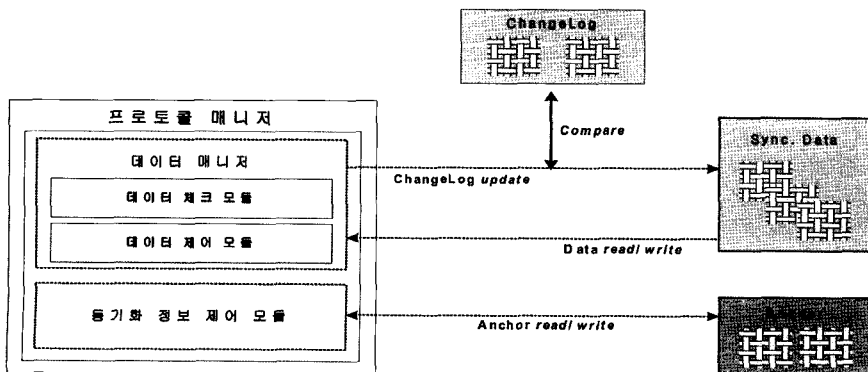


그림 8 프로토콜 매니저의 데이터 제어 원리

정보와 실제 저장되어 있는 데이터의 변경 내역을 비교하여 수신측에 추가, 삭제 및 수정해야 할 동기화 자료의 목록을 작성한다. 이후 자료 동기화가 완료된 후 처리 결과를 변경 기록 로그에 반영하게 되고, 그 변경 기록 로그는 다음 동기화 시점에 이용하게 된다. 또한, 데이터 제어 모듈은 데이터 체크 모듈에서 작성한 동기화 대상 데이터에 직접 접근을 한다. 즉, 전송해야 할 데이터를 가져오고, 수신한 데이터에 대해서 추가, 삭제 및 수정을 하게 된다.

3.2 무선 단말기 영역

동기화 클라이언트 기능을 수행하는 무선 단말기 부분은 응용 프로그램 수준에서 동기화 대상이 되는 자료를 추출하기 위한 모듈과 클라이언트 기능 모듈, 그리고 통신 모듈로 구성되어 있다. 그림 9는 무선 단말기 영역에서 기능별로 분류한 세부적인 모듈들을 보여주고 있다.

무선 단말기는 통신 모듈을 통해서 변환 게이트웨이와 메시지를 교환하고, 클라이언트 기능 모듈을 통해 송신할 메시지를 생성 및 수신된 메시지를 분석한다. 무선 단말기에 저장된 동기화 대상 자료는 PIMs 응용프로그램(App.)에 의해서만 제어가 가능하므로, PIMs 응용 프로그램에서 접근하는 자료를 추출하기 위해 각 무선 단말기의 플랫폼에서 제공하는 API를 이용하여 자료 추출

모듈을 설계하였다.

자료 추출 모듈에서 WinCE 기반의 ActiveSync 프로토콜을 사용하고 있는 단말기의 경우에는 POOM (Pocket Outlook Object Model) 라이브러리를 이용하여 Window Mobile Pocket Office Outlook으로부터 PIMs 데이터인 vCard를 추출한다. 또한, PalmOS 기반의 HotSync 프로토콜을 사용하고 있는 단말기의 경우에는 PIMs 자료에 접근하기 위하여 Pack/Unpack이라는 동작을 통하여 접근하여, 자료를 추출한다. 추출된 자료는 클라이언트 기능 모듈에서 통합 게이트웨이로 전송하기 위한 메시지 형태로 생성되며, 통합 게이트웨이에서 수신된 메시지는 메시지 분석기에 의해 무선 단말기에 필요한 정보를 걸러낸 후, 자료 추출 모듈에 의해 데이터스토어에 반영된다.

그림 10은 동기화 클라이언트 단말기에서 동기화 요청이 발생했을 경우 SyncML 기반의 자료 동기화 게이트웨이 시스템의 동작 메커니즘이다.

무선 단말기에서 동기화가 시작되면 자료 동기화 게이트웨이로 이전 동기화 이후로 변경된 데이터를 하나씩 보내게 된다. 자료 동기화 게이트웨이는 단말기에서 수신된 데이터를 추출하여 처리한다. 이후, 단말기에서 동기화 요청 데이터를 모두 전송하면 자료 동기화 게이

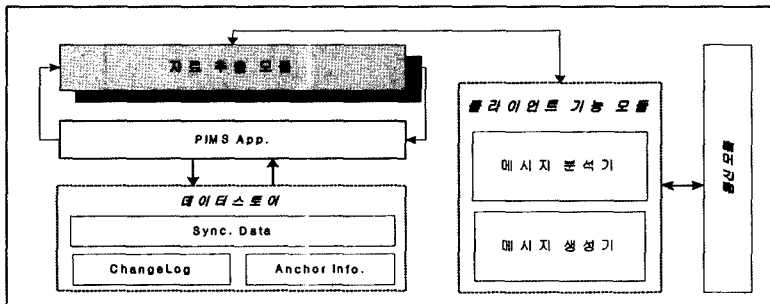


그림 9 무선 단말기 영역의 모듈 구조

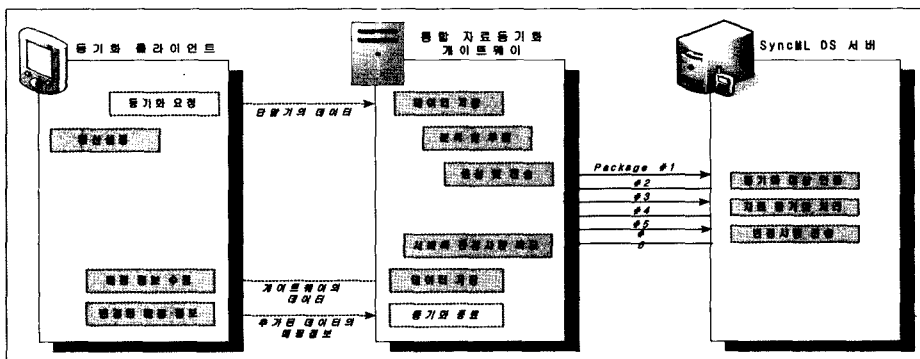


그림 10 SyncML 기반의 통합 자료 동기화 게이트웨이 동작 메커니즘

트웨이에서는 변경 기록 로그를 갱신하고, SyncML 서버와 동기화를 시작한다. 이때 SyncML 서버와의 동기화는 SyncML 규격에 의해 1번 패키지에서 6번 패키지까지 송수신하게 된다. 게이트웨이는 SyncML 서버에서 동기화 수행을 마친 후, 서버측 변경 데이터를 단말기로 전송한다. 무선 단말기는 게이트웨이로부터 수신된 데이터를 받아서, 현재 매핑 테이블과 비교하여 각 데이터에 대한 매핑 정보를 수정한다. 이때, 새로 추가된 데이터에 대해서는 게이트웨이에 저장된 데이터와 단말기에 저장된 데이터의 ID가 다르기 때문에 자료 동기화 게이트웨이로 매핑 정보를 보낸다.

4. 구현 및 검증

본 논문의 통합 자료 동기화 게이트웨이 시스템은 WinCE 기반의 ActiveSync 방식 및 PalmOS 기반의 HotSync 방식으로 클라이언트에서 서버로 전달되는 문서를 중간에서 OMA의 SyncML 방식으로 변환하거나, 그 반대로 변환하여 서버에서 클라이언트로 전달하도록 구현되었다.

4.1 자료 동기화 클라이언트에서의 자료 추출 방식

4.1.1 WinCE 기반의 자료 추출

WinCE 기반의 단말기에 저장된 동기화 대상이 되는 자료는 POOM(Pocket Outlook Object Model)으로 불리는 COM 기반의 라이브러리를 이용하여 접근할 수 있다. 또한, POOM은 Events, Todo-lists, Contacts와 같은 아이템에 대해서 추가, 삭제 및 갱신을 할 수 있는 객체기반의 프레임워크이다[10.11]. 따라서, 자료 동기화에 참여하는 WinCE 기반의 클라이언트에서 PIMs 자료를 추출하기 위해서는 POOM을 이용한 세션 단계가 필요하다.

Step 1. POOM 파트 연결을 위한 COM 오브젝트를 생성

Step 2. Pocket Outlook COM Server에 접근

Step 3. PIMS 리소스에 접근하여 해당 정보를 읽음

POOM을 이용하여 추출된 자료는 클라이언트 기능

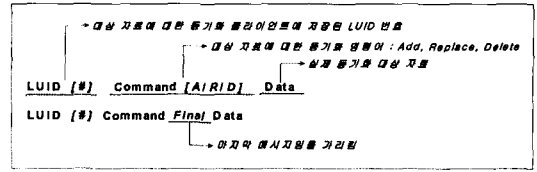


그림 12 동기화 클라이언트에서 생성된 메시지 의미

모듈에서 메시지로 생성되어 통합 게이트웨이 측으로 전송된다. 그림 11은 동기화 대상이 되는 자료를 추출한 후, 생성된 메시지가 클라이언트에서 통합 게이트웨이 측으로 전송되는 과정을 보여주고 있다.

그림 11에서 알 수 있듯이, 추출된 자료는 클라이언트 기능 모듈에서 동기화 대상이 되는 자료, 각 자료에 대한 LUID(Local Unique ID) 및 해당 동기화 명령어로 구성된 메시지로 생성되어 통신 모듈에 의해 통합 게이트웨이 측으로 전송된다. 생성된 메시지에 대한 의미는 그림 12와 같다.

4.1.2 Palm OS 기반의 자료 추출

PalmOS를 기반으로 하는 동기화 클라이언트는 동기화 대상 자료를 데이터베이스 형태로 관리 및 저장된다. 또한, PalmOS 기반의 데이터베이스는 Pack/Unpack이라는 동작을 통하여 접근할 수 있다. Pack이란 모든 자료들을 메모리상에 순차적으로 로딩하는 것을 말하며, Unpack이란 Pack 상태로 메모리에 상주된 자료들에 대해서, 처리 대상이 되는 자료에 대해서 접근하는 것을 Unpack이라 한다[13,14]. 따라서, 자료 동기화 대상 정보를 접근하고 추출하기 위하여, 다음과 같은 과정이 요구된다.

Step 1. 자료 동기화 대상 영역의 데이터베이스를 오픈

Step 2. 오픈된 데이터베이스는 메모리상에 순차적으로 저장 - Pack 상태

Step 3. 처리 대상의 레코드 영역에 대해서 Pack 상태를 해제 - Unpack 상태

Step 4. record_struct에 저장된 레코드의 각 필드 데

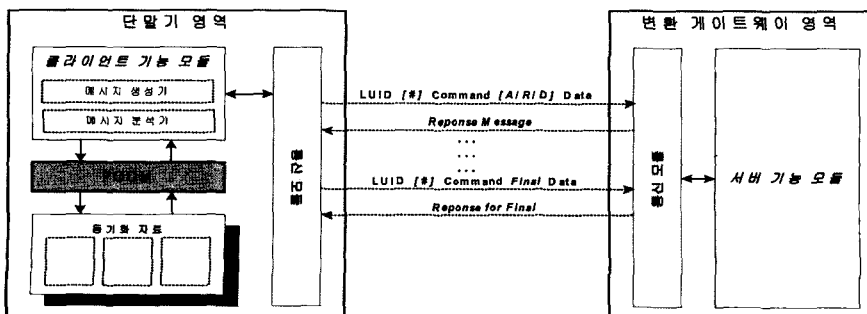


그림 11 POOM을 이용한 Windows CE 기반의 자료 추출 방식

이타를 읽음

Step 5. 각 필드의 데이터를 다 읽은 후, Pack 상태로 전환

위의 과정을 통해 추출된 자료는 클라이언트 기능 모듈에서 메시지로 생성되어 통합 게이트웨이 측으로 전송된다. 그림 13은 PalmOS 기반의 클라이언트에서 동기화 대상이 되는 PIMs 정보를 추출한 후, 클라이언트에서 통합 게이트웨이 측으로 전송되는 형식을 보여주고 있다.

추출된 PIMs 자료는 위의 그림과 동일한 메시지로 생성된 후, 통합 게이트웨이 측으로 전송된다. 자료 동기화 클라이언트로부터 통신 모듈을 통해 수신한 메시지는 통합 게이트웨이 내의 서버 기능 모듈에 의해 메시지를 분석하고, 동기화 클라이언트 측으로 응답 메시지를 전송하게 된다.

4.2 통합 게이트웨이 측에서 자료 추출 및 SyncML 메시지 생성 방식

각각의 자료 동기화 클라이언트로부터 추출된 정보를 수신한 통합 자료 동기화 게이트웨이는 다음과 같은 순서로 수신된 자료를 파싱한 후 SyncML 메시지로 생성하는 과정을 거치게 된다.

Step 1. 수신된 메시지는 서버 기능 모듈에 의해 수신 메시지를 분석한 후, 추출된 자료를 프로토콜 매니저로 전송

Step 2. 프로토콜 매니저는 이전 단계로부터 추출된 LUID, 동기화 명령어 및 동기화 대상 자료에 대해서 이전 동기화의 정보와 비교하여 변경 기록 로그를 갱신하고, 그 정보를 SyncML 클라이언트로 반환

Step 3. 반환된 정보를 바탕으로 SyncML 클라이언트는 SyncML 메시지를 생성한 후, 동기화 서버와 SyncML 프로토콜 규격에 의해 자료 동기화를 수행

Step 4. SyncML 동기화 서버로부터 해당 자료 동기화 완료 메시지를 수신하게 되면, 프로토콜 매니

저는 Anchor 정보를 갱신한 후, 실제 데이터 스토어에 접근하여 동기화 대상이 되는 데이터를 제어

Step 5. 프로토콜 매니저는 갱신된 데이터스토어의 정보를 바탕으로 서버 기능 모듈에게 해당 동기화 클라이언트로 전송할 메시지 생성을 요청

Step 6. 생성된 각각의 클라이언트 메시지는 전송된 후, 해당 클라이언트의 데이터베이스에 게이트웨이 측에서 전송된 데이터로 추가

그림 14는 자료 동기화 클라이언트로부터 요청된 동기화 자료에 대해서 통합 게이트웨이에서 수신된 메시지를 분석하고, SyncML 메시지 생성 및 전송하는 과정을 보여주고 있다.

4.3 검증

본 논문에서 구현된 SyncML 기반의 자료 동기화 게이트웨이 시스템의 정상적인 동기화 수행 여부를 검증하기 위해서 SyncML 기반의 Synthesis DS(Data Synchronization) 서버와 연동하여 시험하였다. Synthesis DS 서버에는 기본적으로 PIMs 정보가 저장되어 있으므로, 동기화 검증에 사용된 동기화 대상 자료로 사용하였으며, HTTP 프로토콜을 통해서 SyncML 메시지를 교환한다[15]. 그림 15는 본 논문에서 성능 평가 및 검증에 적용한 네트워크 구성에 대해서 보여주고 있다.

본 논문에서 구현된 자료 동기화 게이트웨이 시스템에서 동기화 클라이언트측 모듈은 Embedded VC++ 4.0(ActiveSync 클라이언트)과 CodeWarrior 9.0(HotSync 클라이언트)을 사용하였으며, 통합 게이트웨이측 모듈은 리눅스 기반의 gcc 3.3.5를 사용하였다. 그림 16과 그림 17은 자료 동기화 게이트웨이 시스템이 정상적인 자료 동기화가 수행됨을 보여주기 위하여, WinCE 기반의 ActiveSync 클라이언트 및 PalmOS 기반의 HotSync 클라이언트와 SyncML 기반의 Synthesis 동기화 서버간의 자료 동기화를 수행한 결과 화면을 보여준다. 또한, SyncML 방식의 클라이언트와 ActiveSync 및 Hot-

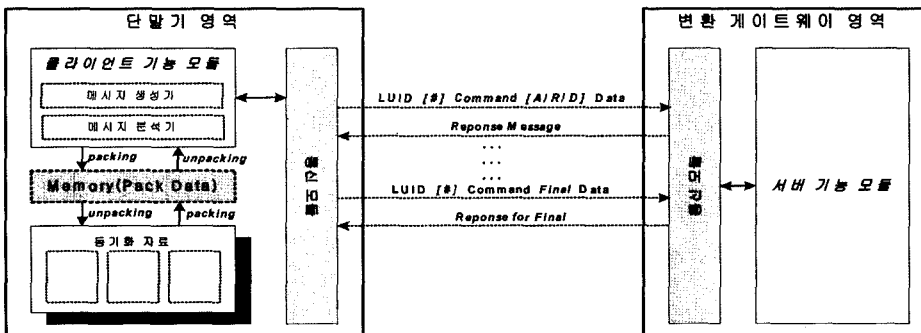


그림 13 packed/unpack을 이용한 Palm OS 기반의 자료 추출 방식

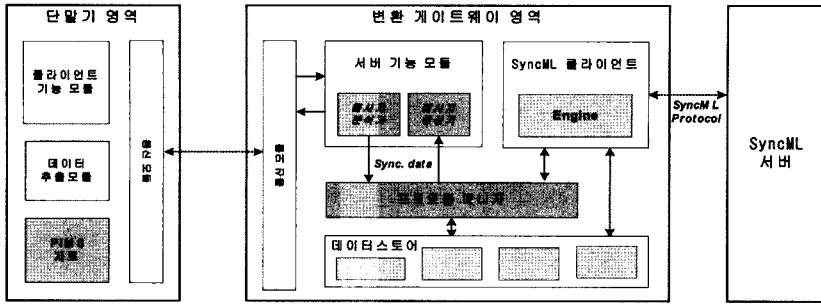


그림 14 통합 게이트웨이에서 SyncML 자료 변환

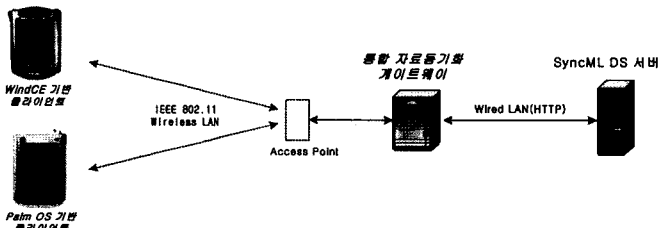


그림 15 검증 및 성능평가 네트워크 환경

Sync 방식의 동기화 클라이언트와의 자료 동기화 시간에 대한 성능평가를 수행하였다.

클라이언트 화면이고, 우측 화면은 정상적인 동기화 연산이 수행된 후의 SyncML DS 서버 내의 PIMs 객체 정보를 보여주고 있다. 위의 그림에서 알 수 있듯이, ①

① Data를 Add(추가)했을 경우

	<table border="1"> <tr><td>2825</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:14:25.647</td><td>Gun</td><td>Jang</td></tr> <tr><td>2827</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:14:44.193</td><td>Donghun</td><td>Ha</td></tr> <tr><td>2831</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:15:14.210</td><td>Hyodor</td><td>Emilianenko</td></tr> <tr><td>2837</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:42:22.863</td><td>James</td><td>Michael</td></tr> <tr><td>2838</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:42:46.020</td><td>Lee</td><td>Bruce</td></tr> <tr><td>2839</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:43:00.397</td><td>Peter</td><td>bob</td></tr> <tr><td>2840</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:45:39.210</td><td>daejin</td><td>Jang</td></tr> <tr><td>2841</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:46:10.223</td><td>nakamura</td><td>Yoski</td></tr> <tr><td>2842</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:46:27.583</td><td>monkey</td><td>Heho</td></tr> <tr><td>2843</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:46:39.570</td><td>Add</td><td>Item</td></tr> </table>	2825	EDIT	DELETE	2006-09-12 18:14:25.647	Gun	Jang	2827	EDIT	DELETE	2006-09-12 18:14:44.193	Donghun	Ha	2831	EDIT	DELETE	2006-09-12 18:15:14.210	Hyodor	Emilianenko	2837	EDIT	DELETE	2006-09-12 18:42:22.863	James	Michael	2838	EDIT	DELETE	2006-09-12 18:42:46.020	Lee	Bruce	2839	EDIT	DELETE	2006-09-12 18:43:00.397	Peter	bob	2840	EDIT	DELETE	2006-09-12 18:45:39.210	daejin	Jang	2841	EDIT	DELETE	2006-09-12 18:46:10.223	nakamura	Yoski	2842	EDIT	DELETE	2006-09-12 18:46:27.583	monkey	Heho	2843	EDIT	DELETE	2006-09-12 18:46:39.570	Add	Item
2825	EDIT	DELETE	2006-09-12 18:14:25.647	Gun	Jang																																																								
2827	EDIT	DELETE	2006-09-12 18:14:44.193	Donghun	Ha																																																								
2831	EDIT	DELETE	2006-09-12 18:15:14.210	Hyodor	Emilianenko																																																								
2837	EDIT	DELETE	2006-09-12 18:42:22.863	James	Michael																																																								
2838	EDIT	DELETE	2006-09-12 18:42:46.020	Lee	Bruce																																																								
2839	EDIT	DELETE	2006-09-12 18:43:00.397	Peter	bob																																																								
2840	EDIT	DELETE	2006-09-12 18:45:39.210	daejin	Jang																																																								
2841	EDIT	DELETE	2006-09-12 18:46:10.223	nakamura	Yoski																																																								
2842	EDIT	DELETE	2006-09-12 18:46:27.583	monkey	Heho																																																								
2843	EDIT	DELETE	2006-09-12 18:46:39.570	Add	Item																																																								

② Data를 Delete(삭제)했을 경우

	<table border="1"> <tr><td>2825</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:14:25.647</td><td>Gun</td><td>Jang</td></tr> <tr><td>2826</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:14:37.867</td><td>James</td><td>Michael</td></tr> <tr><td>2827</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:14:44.193</td><td>Donghun</td><td>Ha</td></tr> <tr><td>2828</td><td>EDIT</td><td>DELETE</td><td>2006-09-11 18:23:11.643</td><td>Bruce</td><td>Lee</td></tr> <tr><td>2831</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:15:14.210</td><td>Hyodor</td><td>Emilianenko</td></tr> <tr><td>2832</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:15:29.897</td><td>Get</td><td>Item</td></tr> <tr><td>2836</td><td>EDIT</td><td>DELETE</td><td>2006-09-12 18:15:58.740</td><td>George</td><td>Benson</td></tr> </table>	2825	EDIT	DELETE	2006-09-12 18:14:25.647	Gun	Jang	2826	EDIT	DELETE	2006-09-12 18:14:37.867	James	Michael	2827	EDIT	DELETE	2006-09-12 18:14:44.193	Donghun	Ha	2828	EDIT	DELETE	2006-09-11 18:23:11.643	Bruce	Lee	2831	EDIT	DELETE	2006-09-12 18:15:14.210	Hyodor	Emilianenko	2832	EDIT	DELETE	2006-09-12 18:15:29.897	Get	Item	2836	EDIT	DELETE	2006-09-12 18:15:58.740	George	Benson
2825	EDIT	DELETE	2006-09-12 18:14:25.647	Gun	Jang																																						
2826	EDIT	DELETE	2006-09-12 18:14:37.867	James	Michael																																						
2827	EDIT	DELETE	2006-09-12 18:14:44.193	Donghun	Ha																																						
2828	EDIT	DELETE	2006-09-11 18:23:11.643	Bruce	Lee																																						
2831	EDIT	DELETE	2006-09-12 18:15:14.210	Hyodor	Emilianenko																																						
2832	EDIT	DELETE	2006-09-12 18:15:29.897	Get	Item																																						
2836	EDIT	DELETE	2006-09-12 18:15:58.740	George	Benson																																						

그림 16 WinCE 기반의 동기화 클라이언트의 정상적인 자료동기화 수행화면

① Data를 Add(추가)했을 경우						
	2825	EDIT DELETE	2006-09-11 18:22:08.630	daejin	Jang	
	2826	EDIT DELETE	2006-09-11 18:22:03.333	James	David	
	2827	EDIT DELETE	2006-09-11 18:22:36.817	Donghun	Ha	
	2828	EDIT DELETE	2006-09-11 18:23:11.643	Bruce	Lee	
	2829	EDIT DELETE	2006-09-11 18:23:33.847	Lotem	Petar	
	2830	EDIT DELETE	2006-09-11 18:23:49.737	Crocoo	Mirco	
	2831	EDIT DELETE	2006-09-11 18:24:38.190	Jun	Jo	
	2832	EDIT DELETE	2006-09-11 18:24:55.110	Sync	Mental	
	2833	EDIT DELETE	2006-09-11 18:25:16.657	namia	Han	
	2834	EDIT DELETE	2006-09-11 18:25:35.000	Thunder	Master	
	② Data를 Delete(삭제)했을 경우					
		2825	EDIT DELETE	2006-09-11 18:22:08.630	daejin	Jang
		2826	EDIT DELETE	2006-09-11 18:22:03.333	James	David
		2827	EDIT DELETE	2006-09-11 18:22:36.817	Donghun	Ha
2828		EDIT DELETE	2006-09-11 18:23:11.643	Bruce	Lee	
2829		EDIT DELETE	2006-09-11 18:23:33.847	Lotem	Petar	
2830		EDIT DELETE	2006-09-11 18:23:49.737	Crocoo	Mirco	
2831		EDIT DELETE	2006-09-11 18:24:38.190	Jun	Jo	
2832		EDIT DELETE	2006-09-11 18:24:55.110	Sync	Mental	

그림 17 PalmOS 기반의 동기화 클라이언트의 정상적인 자료동기화 수행화면

번 화면은 ActiveSync 클라이언트 및 HotSync 클라이언트에서 다수의 PIMs 데이터를 추가하였다. 그 후, 통합 게이트웨이 시스템을 통하여 자료 변환이 발생한 후, SyncML DS 서버와 데이터 추가 연산에 대한 자료 동기화가 수행됨으로써 새롭게 생성된 데이터를 보여주고 있다. ②번 및 ③번 화면은 ①화면과 동일한 과정을 거침으로써, PIMs 데이터에 대해서 삭제 및 추가 연산에 대한 정상적인 자료 동기화 화면을 보여주고 있다.

또한, WinCE, PalmOS 기반의 자료동기화 클라이언트 및 SyncML 자료동기화 클라이언트에 각각 10개, 20개, 40개, 60개 80개, 100개의 동기화 대상 자료에 대해서 통합 자료 동기화 게이트웨이 시스템을 통해 자료 동기화를 수행했을 경우, ActiveSync 및 HotSync를 탑재한 동기화 클라이언트가 SyncML 방식을 탑재한 클라이언트보다 각각 평균 31%, 47%의 동기화 시간을 더 소비하게 된다. ActiveSync 및 HotSync를 탑재한 동기화 클라이언트에서는 각각의 동기화 방식에 따른 대상 자료를 SyncML 통합 자료 동기화 게이트웨이 시스템에서 SyncML 프로토콜에 적합한 메시지 형태로 변환시켜주는 데 대한 수행시간이라고 볼 수 있다.

5. 결론

이동통신 컴퓨팅 환경의 발달과 이동통신 단말기 보

급의 확대에 의해 다양한 플랫폼 기반의 단말기들이 사용되고 있으나, 각각의 서로 다른 자료 동기화 방식을 사용하기 때문에 여러 가지 불편함과 오버헤드가 발생하고 있다. 따라서, 자료 동기화 기술의 표준화에 대한 중요성이 부각되었고, OMA에서는 자료 동기화의 표준으로서 SyncML을 제안하였다. 하지만, SyncML이 각각의 동기화 방식을 탑재한 무선이동 단말기들이 모두 SyncML 방식을 동기화 표준화로 대체하기는 어려운 실정이다.

본 논문의 통합 자료 동기화 게이트웨이 시스템은 WinCE 기반의 ActiveSync 방식 및 PalmOS 기반의 HotSync 방식으로 클라이언트에서 서버로 전달되는 문서를 중간에서 OMA의 SyncML 방식으로 변환하거나, 그 반대로 변환하여 서버에서 클라이언트로 전달하도록 구현되었다. 또한, SyncML 방식의 클라이언트와 ActiveSync 및 HotSync 방식의 동기화 클라이언트와의 자료 동기화 시간에 대한 성능평가를 수행하였다.

또한, SyncML 클라이언트와 ActiveSync 및 HotSync 자료 동기화 클라이언트의 비교를 통해서 통합 자료 동기화 게이트웨이 시스템의 성능평가를 하였다. Windows CE 및 Palm OS 기반의 동기화 클라이언트가 SyncML 방식을 탑재한 클라이언트보다 클라이언트에서 서버로의 자료 동기화 수행의 경우에는 각각 평균

31%, 47%의 동기화 시간을 더 소비하게 된다. 이는 각각의 동기화 방식에 따른 대상 자료를 SyncML 통합 자료 동기화 게이트웨이 시스템에서 SyncML 프로토콜에 적합한 메시지 형태로 변환시켜주는 데 대한 수행시간이라고 볼 수 있다.

향후 과제로는 구현된 통합 자료 동기화 게이트웨이 시스템의 전체 자료 동기화 수행시간에 대한 불필요한 오버헤드를 줄이기 위한 개선이 필요하다. 또한, 다양한 플랫폼에서 동작할 수 있는 통합 게이트웨이 시스템의 확장이 요구된다.

참 고 문 헌

[1] Hansmann, U., Mettala, M., Purakayastha, A., Thompson, P., *SyncML synchronizing and managing your mobile data*, New York, Prentice Hall, pp. 21-34, 2003.

[2] 장대진, 박기현, 주홍택, "SyncML 기반의 자료 동기화 클라이언트 개발", *한국정보과학회 논문지: 컴퓨팅의 실제*, 제 11권, (4호), pp. 357-367, 2005.

[3] Agarwal, S., Starobinski, D., Trachtenberg, A., "On the scalability of data synchronization protocols for PDAs and mobile devices," *IEEE: Journal of Network*, Vol.16, (Issue 4), pp. 22-28, 2002.

[4] SyncML Representation Protocol, OMA Technical Section, [2003년 8월 검색], http://www.openmobile-alliance.org/tech/syncml_represent_v11.pdf.

[5] SyncML Synchronization Protocol, OMA Technical Section, [2003년 8월 검색], http://www.openmobile-alliance.org/tech/syncml_protocol_v11.pdf.

[6] Fox, D., Box, J., *Building solutions with the Microsoft .NET compact framework: Architecture and best practices for mobile development*, Massachusetts, Addison Wesley, pp. 243-282, 2003.

[7] Grattan, N., Brain, M., *Windows CE 3.0 Application Programming*, New York, Prentice Hall, pp. 187-215, 2000.

[8] Rbodes, N., Mckeeban, J., *Palm OS Programming The Developer's Guide*, California, O'REILLY, pp. 320-368, 2002.

[9] Winton, G., *Palm OS Network Programming*, California, O'REILLY, pp. 255-260, 2001.

[10] Ahmed, M., Krishnamurthy, S., Katz, R., Dao, S., "An architecture for providing range extension by deploying mobile gateways in ad hoc networks," *IEEE: The 13th International Symposium on Personal, Indoor and Mobile Ratio Communications*, Vol.4, pp. 1660-1664, 2002.

[11] Lee, J. K., Kim, Choi, Y. H., Choi, J. H., Nam, S. W., "Hybrid gateway advertisement scheme for connecting mobile ad hoc networks to the Internet," *IEEE: The 57th Semiannual Vehicular Technology Conference(VTC 2003-Spring)*, Vol.1, pp. 191-195, 2003.

[12] Oracle Partners-Solutions, ORACLE, [2006년 11월 5일 검색], <http://www.oracle.com/global/partners.html>.

[13] Synchronica Mobile Synchronization and Device Management, Synchronica, [2005년 10월 23일 검색], <http://www.synchronica.com>.

[14] SyncML C Toolkit, SourceForge.net, [2003년 8월 1일 검색], <http://sourceforge.net/projects/syncml-ctoolkit>.

[15] Synthesis AG - SyncML Client & Server Solutions, Synthesis AG, [2003년 10월 검색], <http://www.synthesis.ch>.



장 대 진

1998년 2월 계명대학교 정보통신학부 컴퓨터공학 학사. 2001년 8월 계명대학교 정보통신학부 컴퓨터공학 석사. 2007년 2월 계명대학교 정보통신학부 컴퓨터공학 박사. 관심분야는 Mobile Data Synchronization, Wireless System, 임베디

스 시스템



박 기 현

1979년 2월 경북대학교 전자계산학 학사. 1981년 2월 한국과학기술원 전자계산학 석사. 1990년 8월 미국 Vanderbilt 대학교 전자계산학 박사. 1981년 3월~현재 계명대학교, 정보통신대학, 교수. 관심분야는 Parallel Processing System,

모바일 소프트웨어, 임베디드 소프트웨어