

# SyncML 자료 동기화 방법을 이용한 이동무선통신 환경에서의 경매 시스템 구축

## Construction of a Bidding System Using SyncML Data Synchronization Mechanism

박 기 현 (Kee Hyun Park)    계명대학교 정보통신대학 컴퓨터공학과 교수  
유 상 진 (Sang Jin Yoo)    계명대학교 경영대학 경영정보학과 교수

### 요 약

PDA 혹은 핸드폰 등과 같은 이동 무선통신 단말기들의 수요가 급격히 증가함에 따라, 다양한 비즈니스 분야에서 이동 무선통신 단말기들이 중요한 역할을 한다. 본 논문에서는 이동 무선통신 환경에서의 경매시스템을 구축한다. 그러나, 많은 이동 무선통신 단말기들이 주어진 자료 항목의 값을 동시에 변경할 수 있기 때문에, 자료 일치성이 매우 중요하다.

그러므로, 본 논문에서는 SyncML 자료 동기화 방법을 사용한 경매시스템을 제안하고 구축한다. SyncML은 자료 동기화의 세계적인 표준을 위하여 이동 무선통신 기술 선두주자들이 컨소시엄을 구성한 SyncML Initiative에 의하여 제안된 자료 동기화 방법이다. 구축된 경매시스템은 LINUX, C++언어 및 PocketPC 2000 에뮬레이터 프로그래밍 환경에서 구현되고 테스트되었다.

**키워드:** 경매시스템, 이동무선통신단말기, 자료동기화, SyncML

## I. 서 론

이동무선통신의 발달과 함께, PDA(Personal Digital Assistant) 혹은 휴대폰과 같은 이동무선통신 단말기의 사용이 급격하게 늘고 있다. 바야흐로, 이동무선통신 단말기는 기업의 생산성을 높이기 위한 도구로 발전하면서 교육, 금융, 제조, 서비스, 물류 분야 등 많은 사업 분야에 적용되기 시작함으로써, 포스트 PC 시장을 선도할 품목으로 자리매김하고 있다. 최근 하드웨어 기술의 발달로 이동무선통신 단말기의 구입가격이 저렴해지고 통신의 발달로 효율성이 증가되면서 매우 급속하게 보급이 이루어지고 있다.

그런데 이동무선통신에서는 유선 인터넷통신과 달

리, 인터넷에 연결된 서버의 자료를 항상 이용할 수 없는 실정이지만, 사용자는 장소나 시간에 구애받지 않고 최신 자료의 이용을 원한다. 이동무선통신 단말기의 사용자가 인터넷에 연결된 서버의 자료를 사용하기 위해서는 먼저 무선 인터넷통신을 통하여 해당 자료를 검색한 후, 이를 이동무선통신 단말기에 저장하여야 한다. 단말기에 저장된 자료를 사용하는 동안에는 인터넷 서버와 연결이 끊어지기 때문에, 변경된 자료를 다시 무선 인터넷통신을 통하여 인터넷 서버에 전송하여야 한다. 뿐만 아니라, 주기적으로 인터넷에 재접속하여 서버의 변경된 자료를 전송받지 않으면 최신의 자료를 사용할 수 없다. 즉, 이동무선통신 단말기의 사용자가 항상 최신의 자료를 이용하기 위

해서는, 인터넷 서버의 자료와 이동무선통신 단말기의 자료를 일치시키는 작업이 필요하며, 이를 이동무선통신에서의 자료 동기화(Data Synchronization)라고 한다[1-4].

유선 혹은 무선 네트워크에서 동일한 자료가 인터넷 서버 및 다수의 이동무선통신 단말기들에게 중복 저장되어 있는 경우에, 한 곳의 자료 변경이 즉시 다른 곳에도 전달되어야 하는 요구가 최근에 급격히 늘어나고 있다. 예를 들면, 영업부서 사원들 모두가 동일한 일정 자료를 각자의 이동무선통신 단말기에 저장하고 오전에 회사를 출발하지만, 회사 내부 사정에 의한 일정 변경 혹은 영업부 사원의 거래처 사정에 의한 일정 변경 등으로 인하여 오후에는 서로 다른 일정 자료를 각자의 단말기에 저장하고 있을지 모른다. 이럴 경우에 영업부서의 전체적인 일정 자료에 대한 일치성 문제가 발생하여 혼란이 생길 수 있다. 따라서 각자의 일정 자료에 대한 변경이 일어날 때 마다, 다른 서버 혹은 단말기들에 중복 저장되어 있는 동일한 자료에 대한 동기화 작업이 필요하다. 이러한 문제들은 개인정보 관리(PIM: Personal Information Management), 이동무선통신 단말기 관리[1, 4, 12] 등의 분야에서 흔히 발견된다. 영업 분야에서 이동무선통신 단말기의 사용이 급증하는 현재의 추세로 볼 때, 문제 해결에 대한 요구가 급증할 것이라고 생각된다.

이러한 문제들을 해결할 수 있는 방법들 중의 하나로써, 앞에서 언급한 자료 동기화 방법이 주목받고 있다. 현재, 자료 동기화 문제를 해결하기 위하여 여러 회사들이 각자의 방법을 사용하고 있다. Compaq 회사에서는 Windows CE에서 동작하는 Active Sync[5]를 사용하고, Palm 회사의 경우에는 Palm OS에서 동작하는 HotSync[7]를 사용하는 등, 다양하고 많은 자료 동기화 프로토콜들과 방법들을 사용하고 있기 때문에 표준화에 대한 요구가 높아졌다. 따라서 Ericsson, IBM, Lotus, Motorola, Nokia, Palm Inc., Psion, Starfish Software 등의 주요 이동무선통신 회사들이 주축이 된 SyncML Initiative에서는 SyncML(Synchronization Markup Language) 자료 동기화 방법을 제안하여 공개

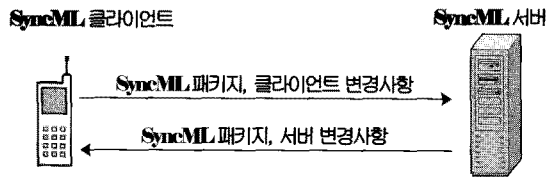
적인 표준화를 시도하고 있다[1, 4, 7-13].

본 논문에서는 자료 동기화 방법의 응용 분야를 넓히는 연구의 일환으로서, SyncML 자료 동기화 시스템을 이용하여 이동무선통신 환경에서의 경매 시스템을 구축한다. SyncML 자료 동기화 시스템을 이용한 경매 시스템은 현재의 경매 시스템보다 다음과 같은 유리한 점들이 있다. 첫째, 경매 자료를 위해서 소량의 자료만을 전송하므로, 통신비용 및 시간이 절감된다. 현재의 경매 시스템은 한 화면의 모든 자료들이 일제히 전송되므로, 전송하는데 많은 시간이 걸릴 뿐만 아니라, 통신비용도 많이 소요된다. 그러나 자료 동기화 시스템을 이용하면, 필요한 자료에 대해서만 전송이 일어나므로 유리하다. 둘째, 자료 동기화 도중에 통신 중단이 발생하더라도 (그 이후에) 자동으로 자료 동기화를 보장해 준다. 현재의 경매 시스템에서는 통신 중단 사고가 일어나면, 사용자 스스로 수동으로 조작하여 다시 서버에 접근하여야 하는 불편이 있다. 셋째, 본 논문에서 사용하는 SyncML 자료 동기화 방법은 현재 세계적인 표준으로 인정받고 있으므로, 향후 다양한 종류의 이동무선통신 단말기들에게 널리 사용될 수 있다. 넷째, 본 연구자가 소속된 팀이 개발한 SyncML 자료 동기화 시스템을 사용하기 때문에, 구입 및 유지비용이 거의 없을 뿐만 아니라, 상황에 맞게 변경이 적은 비용이 언제든 지 가능하다.

본 논문은 다음과 같이 구성 된다. 제 II장에서는 본 연구자가 소속된 팀이 개발한 SyncML 자료 동기화 시스템인 ThinkSync 시스템을 설명한다. 제 III장에서는 ThinkSync 시스템을 이용한 이동무선통신 환경에서의 경매 시스템의 구축 및 실험 결과에 대해서 논의한다. 마지막으로, 제 IV장에서는 결론 및 향후의 연구 과제에 대해서 언급한다.

## II. SyncML 시스템

<그림 1>은 SyncML을 기반으로 하는 클라이언트와 서버간의 자료 동기화를 나타내고 있다.

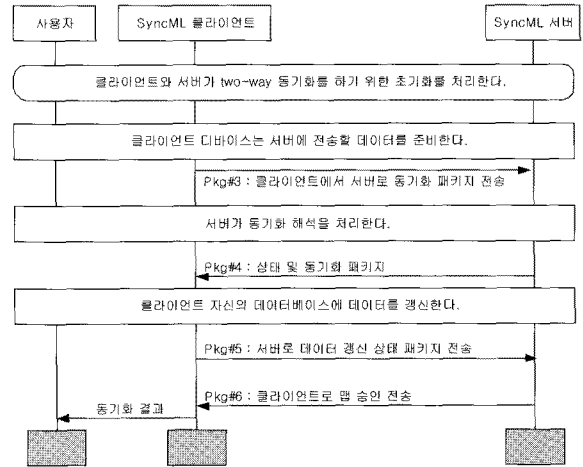


<그림 1> SyncML을 기반으로 하는 클라이언트-서버 구조

SyncML 클라이언트가 먼저 서버에게 갱신된 자료를 포함된 메시지를 전송하면, SyncML 서버는 클라이언트가 요청한 동기화 타입에 의해 서버 측 자료와 동기화 작업을 수행한 후, 다시 클라이언트에게 작업 결과 및 서버 자신의 변경된 자료도 함께 전송한다. 이러한 몇 번의 자료 교환을 수행함으로써, 자료 동기화가 이루어지며 클라이언트와 서버간의 해당 자료에 대하여 일치성을 보장하게 된다. 그림에서 패키지(Package)라 함은, 자료 동기화 과정에서 전달되는 패킷(Packet)의 기본 단위를 의미한다.

ThinkSync 시스템은 SyncML의 7가지 자료 동기화 모드(Mode)들을 모두 지원하도록 구현되었는데, 다음 표와 같다.

이들 중에서 가장 많이 사용되는 Two-way Synchronization 모드는 <그림 2>와 같이 진행된다.



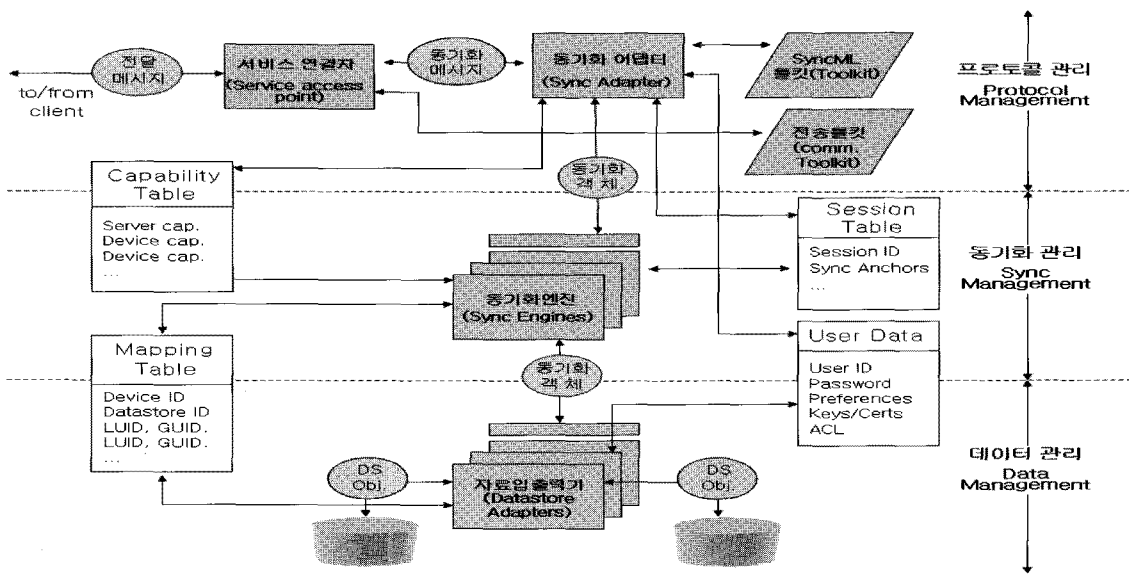
<그림 2> Two-way Synchronization 과정

본 연구자가 소속된 팀이 개발한 SyncML 자료 동기화 시스템은 가장 최근에 발표된 SyncML 1.1.1의 규정을 준수하도록 설계 구현되었으며, ThinkSync로 명명되었다. ThinkSync 시스템은 주로 유선으로 연결되어 있는 인터넷 서버에 탑재된 서버(Server)와 이동무선통신 단말기에 탑재된 클라이언트(Client)로 나뉘는데, <그림 3>은 ThinkSync 서버 구조를 보여준다.

ThinkSync 서버는 동기화 상대가 되는 ThinkSync 클라이언트로부터 수신한 메시지에 대한 처리와 서

<표 1> SyncML의 자료동기화 모드

자료 동기화 프로토콜	설 명
Two-way Synchronization	• 서버와 클라이언트가 변경된 자료에 대한 정보를 교환하는 가장 일반적인 프로토콜
Slow Synchronization	• 모든 자료 항목들에 대한 자료 동기화를 수행하는 프로토콜
One-way Synchronization from Client Only	• 클라이언트가 자신의 변경된 자료를 서버에게 일방적으로 전달하는 프로토콜
Refresh Synchronization from Client Only	• 클라이언트가 자신의 모든 자료를 서버에게 일방적으로 전달하는 프로토콜
One-way Synchronization from Server Only	• 서버가 자신의 변경된 자료를 클라이언트에게 일방적으로 전달하는 프로토콜
Refresh Synchronization from Server Only	• 서버가 자신의 모든 자료를 클라이언트에게 일방적으로 전달하는 프로토콜
Server Alerted Synchronization	• 서버가 클라이언트에게 지정된 프로토콜을 이용하여 자료 동기화 시작을 요청하는 프로토콜



〈그림 3〉 ThinkSync 서버 구조

버 측 변경 사항에 대한 메시지 송신 기능을 담당하며, 반영된 결과를 클라이언트에게 송신하는 기능을 갖는다.

ThinkSync 서버는 프로토콜 관리, 동기화 관리 그리고 자료 관리 모듈로 구성되어 있으며, 각 모듈의 기능은 다음과 같다. 프로토콜 관리 모듈은 메시지를 처리하고 생성하며, 연속적인 자료를 교환하는 기능을 담당한다. 또한 서비스 연결자(Service Access Point), 동기화 어댑터(Adapter), SyncML 툴킷(Toolkit), 전송 툴킷 등으로 구성되어 있으며, 세션 테이블, 사용자 자료, 장치 성능 테이블을 참조한다. 서비스 연결자는 전송 계층(Transport Layer)에서 신뢰성 있는 연결을 기반으로 특정 포트로부터 수신되는 메시지의 암호화/복호화를 담당하는 기능을 담당하며, 전송 툴킷을 이용하여 SyncML 메시지를 변환시킨다. 동기화 어댑터는 프로토콜 관리 모듈에서 가장 핵심적인 기능을 담당한다.

SYNC\_USERS 테이블에서 USER\_KEY, USERID, PASSWD, READONLY 등을 확인하여 인증하는 절차를 수행한 후, SyncML 툴킷을 이용하여 실제 동기화 객체로 변환하여 동기화 엔진으로 전송되어진다. 그리고 SYNC\_LOG, SYNC\_TARGET 등의 데이터베이스 테이블(Table)들을 이용하여 세션(Session) 경

로를 제어하고, 동기화 시점의 앵커(Anchor) 정보를 유지한다. 또한 클라이언트와 서버의 장치 처리 능력(Capability)에 대한 정보를 저장하고 있다. SyncML 툴킷은 클라이언트로부터 수신된 메시지를 디코딩하고, 서버 측에서 전송될 메시지를 인코딩하는 기능을 담당하며 Sync 어댑터를 돕는 역할을 한다. 통신 툴킷은 실제 메시지의 송수신을 담당하는 통신과 관련된 함수들의 집합체이다.

동기화 관리 모듈은 동일한 자료를 대상으로 동시에 변경되는 자료로부터 발생하는 충돌(Conflict) 해결 및 탐지 기능을 담당한다. 동기화 엔진을 핵심으로 구성되어 있으며 장치별 처리 능력 테이블, 세션 테이블, 맵핑(Mapping) 테이블을 참조한다. 또한 동기화 엔진은 동기화 과정에서 발생 가능한 충돌 처리와 LUID(클라이언트 Item 식별자)와 GUID(서버의 Item 식별자)의 맵핑을 담당한다. 충돌 처리와 ID 맵핑이 완료되면 동기화 객체를 DS 어댑터에게 넘겨준다. 데이터 관리 모듈은 동기화 대상이 되는 객체를 데이터베이스 혹은 파일과 같은 형태로 Data Store에 저장한다. DataStore 어댑터로 구성되어 있으며, 맵핑 테이블, 사용자 데이터를 참조한다. 그리고 DataStore 어댑터는 동기화 엔진에서 넘겨준 동기화 객체를 데

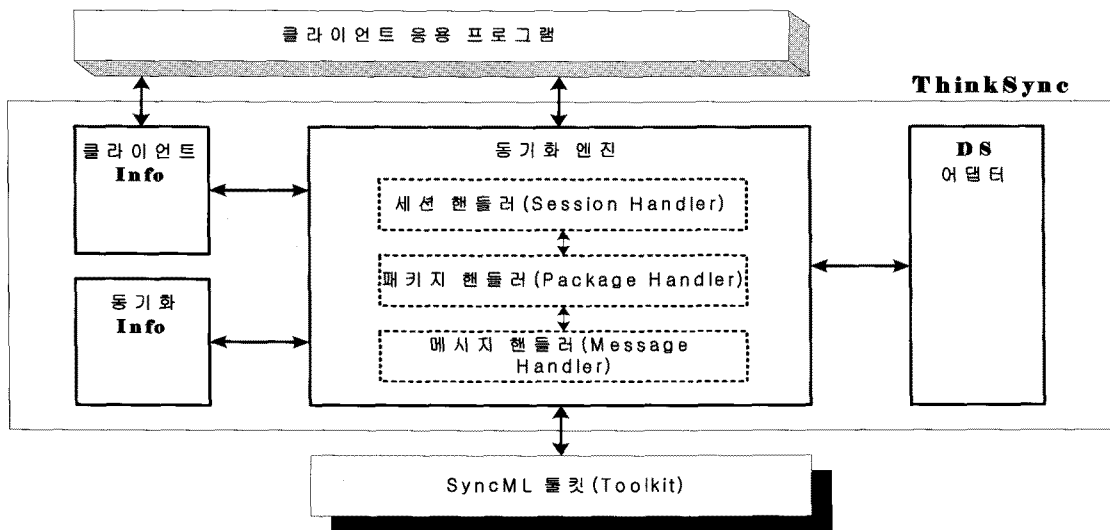
이터 스토어 객체로 변환하여 데이터베이스에 저장한 후 ID 맵핑을 수행한다.

ThinkSync 서버는 다음과 같은 진행 과정을 수행한다. 우선, 인코딩된 메시지가 서비스 연결자에 도착하면 서비스 연결자는 메시지 인증 절차를 수행하고 해당 메시지를 디코딩한 후, 메시지의 결과를 동기화 어댑터로 보낸다. 이 메시지는 클라이언트의 변경 정보를 담고 있으며, 동기화 어댑터는 세션 테이블을 갱신하고 메시지를 통하여 클라이언트가 데이터베이스에 접근 권한이 있는지 판별하고, 그리고 동기화 어댑터는 메시지를 툴을 이용하여 동기화 객체로 변환하고 동기화 엔진을 호출한다. 호출된 동기화 엔진은 자료의 충돌 여부를 파악하고 해결한다. 이러한 해결 과정에서 동기화 엔진은 데이터스토어 어댑터가 사용하는 자료 항목(Data Item)을 불러와서 사용하게 될 수도 있다. 또한 동기화 엔진은 클라이언트 혹은 서버에서 자료 항목의 생성과 변경이 발생했는지를 검사하고, 동기화 객체를 데이터스토어 어댑터가 받아서 데이터스토어 객체로 변경한 후, 데이터베이스에 저장하고 맵핑 테이블을 갱신한다.

<그림 4>는 ThinkSync 클라이언트의 구조를 보여준다. ThinkSync 클라이언트는 클라이언트 엔진, DS 어댑터(Adapter), 클라이언트 정보, 동기화 정보, SyncML

툴킷, 그리고 클라이언트 응용 부분 등으로 구성되어 있다. 클라이언트 엔진 부분은 동기화 할 자료를 포함하여 서버에게 전송하여야 할 동기화 메시지를 생성하고, 서버로부터 수신된 메시지를 처리 및 반영한다. 클라이언트 엔진은 동기화 단위에 따라 분류된 세 개의 모듈들로 구성되어 있다.

실제 메시지의 송수신은 SyncML 툴킷 부분이 담당하며, 세션 핸들러(Session Handler)는 툴킷 부분에서 연결된 전송 프로토콜(HTTP, WAP, OBEX 등) 형태에 따라 선택된 프로토콜 세션을 열고 닫는다. 그리고 동기화해야 할 메시지의 헤더 부분을 생성하는 기능을 한다. 패키지 핸들러(Package Handler)는 SyncML 문서의 Syncbody 부분의 논리적 구성을 담당한다. 논리적 구성을 위해서 메시지 핸들러에게 서버로부터 받은 메시지의 종류를 확인하고, DS 어댑터에게 이전 동기화 이후의 변경사항을 확인한다. 메시지 핸들러(Message Handler)는 SyncML 클라이언트 엔진 최하부에 해당하는 부분으로 주로 이벤트 발생에 의해 호출되는 모듈이다. 동기화 시점에서 클라이언트가 담당하고 있는 자료의 변경이 발생되어 동기화를 해야 할 경우, 자료 동기화를 위해 필요한 XML 명령어를 실질적으로 만들어 주는 기능을 담당하고 있으며, 서버로부터 전송된 SyncML 프로토콜을 제일 먼저 받아 처리 및 분



<그림 4> ThinkSync 클라이언트 구조

〈표 2〉 SyncML을 구성하고 있는 DB테이블

테이블 이름	저장 정보
SYNC_CONTACTS	사용자의 주소록 정보
SYNC_EVENTS	일정에 관련된 정보
SYNC_FOLDERS	사용자의 폴더에 관한 정보
SYNC_PERM	사용에 관한 권한 설정 정보
SYNC_DEVICES	장치에 관한 정보
SYNC_USERS	사용자에 관한 정보
SYNC_MAPS	데이터베이스 테이블에 존재하는 자료 항목에 대한 GUID와 LUID의 맵핑 정보
SYNC_TASKS	사용자의 업무와 관련된 정보
SYNC_TARGETS	동기화시에 필요한 정보
SYNC_NOTES	메모 혹은 낙서장으로 활용될 시, 필요한 정보
SYNC_LOG	동기화에 대한 기록(Log) 정보

석하는 역할을 담당한다.

DS 어댑터는 외부로부터 실제 Data Store에 접근하여 자료를 삽입, 삭제, 갱신할 수 있는 방법을 제공하며, 동기화를 위해 이전에 변경되었던 사항에 대한 Log를 기반으로 새로 변경된 사항들을 검사하여 동기화 할 자료를 결정해준다. 클라이언트 정보 모듈은 동기화에 필요한 다음과 같은 정보를 포함하고 있다: 첫째, 동기화 대상이 되는 목적지 주소(Target URL) 및 vcard, vevents, vtodo, notes 등과 같은 PIMS 정보 형태에 대해서 기술하고 있다. 둘째, 클라이언트 장치의 제조사, 펌웨어(Firmware) 버전(Version), IMEI 등과 같은 물리적 정보와 허용 가능한 메모리 용량, 저장 가능한 최대 오브젝트 크기 등과 같은 장치의 성능 관련 정보를 담고 있다. 그리고 동기화 대상인 클라이언트 URL에 대해서 기술한다.

동기화 정보 모듈은 동기화 대상들 간에 교환되는 세션과 메시지의 참조 번호에 대한 정보를 담고 있으며, 클라이언트와 서버간의 역할과 기능을 결정하는 동기화 종류에 대한 정보를 담고 있다. 그리고 이전의 동기화 수행과 현재 수행하려 하는 동기화 수행간의 일치성을 검사하기 위한 앵커 정보를 담고 있다.

클라이언트 응용 모듈은 실제 동기화를 요구하는

사용자와 클라이언트 엔진 간의 인터페이스를 제공하며 SyncML 툴킷은 서버로부터 수신된 메시지를 디코딩하고, 클라이언트에서 전송할 메시지를 인코딩하는 기능을 담당하며 전송 프로토콜간의 실제 메시지의 송신과 수신을 담당한다.

ThinkSync 클라이언트의 동기화 과정은 다음과 같다.

- 사용자가 클라이언트 응용 프로그램을 통하여 자료 동기화를 요청하면 세션 핸들러는 SyncHeader를 생성하고 패키지 핸들러가 SyncBody 생성을 위한 메시지의 논리적인 구성을 담당한다.
- 실제 SyncBody를 생성하기 위해서 메시지 핸들러는 동기화 형태(Mode)를 규정짓는 Alert 명령, 동기화를 요청한 클라이언트의 장치 정보를 알려주기 위한 Put 명령, 그리고 동기화 대상 상대가 되는 서버의 장치 정보를 알기 위해 Get 명령어를 SyncBody에 담아 전송하고 전송한 명령어를 저장한다.
- SyncBody를 닫으면 클라이언트에서 자신의 장치 정보와 인증 정보를 서버로 전달하는 하나의 메시지가 완성되고 SyncML 툴킷을 통해 서버

로 전송한다. 서버는 이 메시지를 수신하여 단말기 관리 서비스를 수행할 적절한 대상인지를 인증하고, 수신한 클라이언트의 정보를 분석하여 자신의 정보와 비교한 후, 그 결과를 클라이언트에게 전송한다.

- 클라이언트는 전송받은 메시지를 툴킷을 통해 세션 핸들러에서 각 필드의 적합성 및 버전, 인증 등을 고려해서 SyncHeader를 처리하고, 메시지 핸들러에서 Syncbody의 Alert에 대한 Status 처리 및 Put에 대한 Status 처리 그리고 Get에 대한 Status 처리를 한 후, 세션을 닫고 결과를 처리한다. 또한 처리된 내용에 대한 응답을 서버로 전송하기 위해서 세션 핸들러는 응답 메시지에 대한 SyncHeader를 생성하며, 패키지 핸들러는 SyncBody를 생성한다.
- 메시지 핸들러에서는 SyncHeader 처리를 위한 Status 명령어를 요청하며, Alert 처리를 위한 Status 명령어를 요청한다. DS Adaptor에서 앵커 정보를 참조하여 이전 동기화 후 변경된 사항을 확인하고, 변경된 사항이 있다면 Sync 태그(Tag)를 처리한다. 이 후, DS 어댑터에서 삽입, 갱신, 삭제에 대한 자료 조작 명령어를 구분하면, 패키지 핸들러는 적합한 자료 조작 명령어를 메시지 핸들러에게 요청하게 된다. 그리고 Sync 태그와 SyncBody를 닫으면 세션 핸들러는 툴킷을 통해 서버로 변경된 내용을 전송한다.
- 클라이언트에 변경사항이 있다면 서버에 반영하고 서버에서 변경된 사항이 있다면 변경된 사항을 클라이언트로 전송하게 된다. 클라이언트는 툴킷을 통해 세션 핸들러에서 SyncHeader를 처리하고 메시지 핸들러는 각 명령어에 대한 Status를 처리한 후, DS 어댑터에 변경사항을 저장하고 세션을 닫고 결과를 처리한다.
- 처리 후에도 변경 사항이 발생하면 위의 동기화 과정이 반복적으로 수행된다.

ThinkSync는 서로 다른 장치와 응용 프로그램 간

의 자료 동기화를 위해 이기종 간의 상호 운용성을 보장하기 위한 메시지 기술 언어로서 XML을 사용하며, 메시지 내의 명령어 및 자료 동기화를 위한 정보는 XML DTD로 정의된 정보들을 다음과 같은 데이터베이스 테이블들로 구성하여 해당 필드에 전송되어 온 자료들을 저장한다.

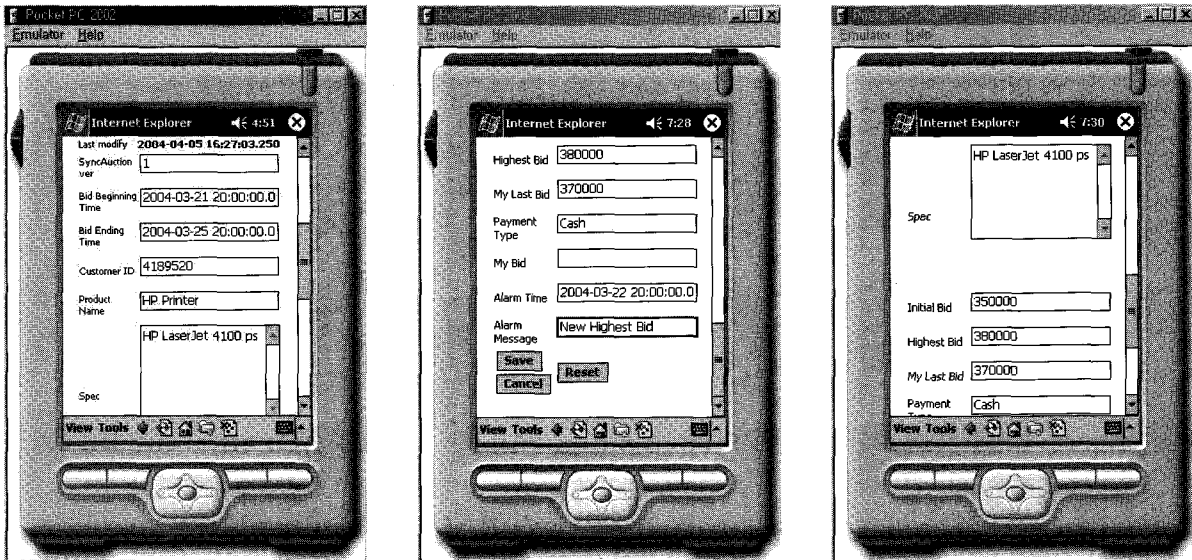
### III. 경매 시스템 구축 및 실험

본 연구에서는 ThinkSync 시스템을 이용한 이동무선통신 환경에서의 경매 시스템을 구축하기 위하여 앞에서 언급한 데이터베이스 테이블들 이외에, 아래와 같은 별도의 테이블 SYNC\_BID를 추가하였다.

〈표 3〉 SYNC\_BID 테이블

속성(Attribute)	설 명
LstMod	자료가 수정된 최근 시간
SyncVer	SyncML 버전(Version)
BidBegin	경매 시작시간(yyyy-mm-dd hh:mm:ss.s)
BidEnd	경매 종료시간(yyyy-mm-dd hh:mm:ss.s)
CustID	사용자 ID
ProdName	경매 물품 이름
Spec	경매 물품 내역
InitBid	초기 입찰가
HighBid	최고 입찰가
MyLstBid	사용자의 최근 입찰가
PayType	지불 방식
MyBid	사용자의 입찰가(입력)
AlarmType	알람 종류

SYNC\_BID 테이블의 각 속성은 굳이 설명하지 않아도 명백하다. 다만, AlarmType 속성에 대해서만 별도로 언급한다. 사용자가 입찰 도중에 계속 단말기의 화면을 주시할 수 없는 상황도 발생하므로, 만약 사용자가 입력한 입찰가 보다 높은 입찰가가 (다른 사용



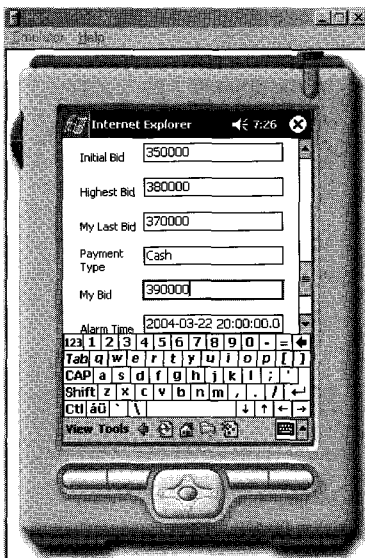
〈그림 5〉 SyncML 시스템을 이용한 경매 시스템 전체 창

자에 의해) 입력되면 경고신호와 함께 알람(Alarm) 화면이 나타나도록 설계했다. 현재는 Alarm Type이 1가지 밖에 없으나, 원한다면 여러 형태의 알람들을 포함시킬 수 있다.

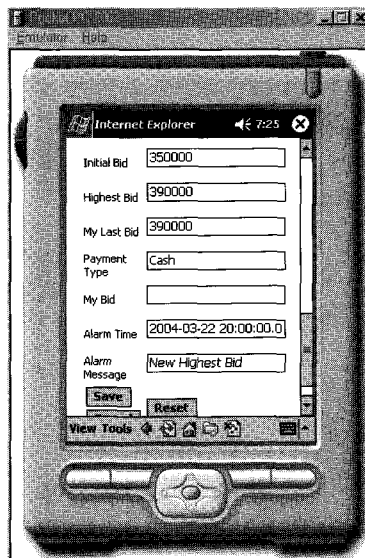
본 연구의 경매 시스템은 ThinkSync 자료 동기화 시스템을 기반으로 LINUX와 C++ 환경에서 구축되었다. 이동무선통신 단말기로서는 PocketPC 2002

Emulator[6]를 사용하였다. <그림 5>는 ThinkSync 시스템을 이용한 경매 시스템의 전체 창(Window)을 나누어서 보여준다. 그림의 단말기에서 보이는 모든 자료들은 ThinkSync 시스템의 서버 및 동작하는 다른 단말기들에게도 동일하게 중복 저장되어 있다.

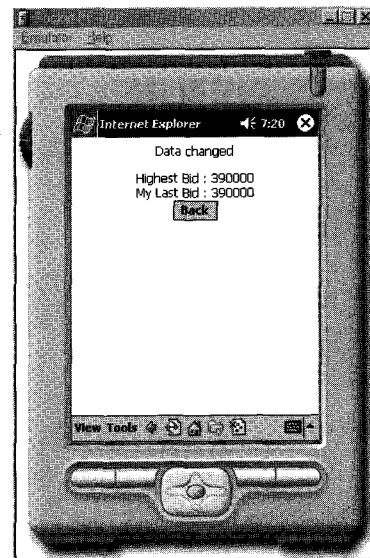
<그림 5>는 사용자가 입찰가를 입력하는 창을 보여준다. 입력 창은 그림에서 굵게 표시되었다. 창 아



〈그림 6〉 입찰가 입력중인 화면



〈그림 7〉 입찰가 입력후의 변환된 화면



〈그림 8〉 알람팝업(Pop-up) 창



래쪽에 보이는 것은 입력 시에 나타나는 키보드 창이다.

또한 <그림 6>은 입찰가를 입력한 후의 창을 보여준다. 이 그림에서 'My Last Bid' 항목과 'Highest Bid' 항목의 값이 바뀌었음을 알 수 있다. 입력된 입찰가는 SyncML의 Two-way Synchronization 모드(제 II장 참조)를 이용하여 서버에게 전달된다. 만약, 어떤 단말기가 무선 신호가 약한 지역으로 이동하거나 전원이 중단되어서 제때에 변경된 자료를 수신하지 못하는 경우에는, Slow Synchronization 모드가 작동하여 모든 자료에 대한 전송이 서버로부터 이루어진다.

<그림 7>은 알람 팝업(Pop-up) 창을 보여준다. 현재 이 알람 창은 사용자의 입찰가 보다 높은 입찰가가 다른 사용자에 의해서 입력될 경우에만 나타나게 되어 있다. 이 알람 창에는 최고 입찰가와 사용자 자신의 최근 입찰가가 모두 나타난다.

#### IV. 결 론

본 논문에서는 자료 동기화 방법의 응용 분야를 넓히는 연구의 일환으로서, SyncML 자료 동기화 시스템을 이용하여 이동무선통신 환경에서의 경매 시스템을 구축하였다. SyncML 자료 동기화 시스템을 이용한 경매 시스템은 현재의 경매 시스템보다 다음과 같은 유리한 점들이 있다: 첫째, 경매 자료를 위해서 소량의 자료만을 전송하므로, 통신비용 및 시간이 절감된다. 현재의 경매 시스템은 한 화면의 모든 자료들이 일제히 전송되므로, 전송하는데 많은 시간이 걸릴 뿐만 아니라, 통신비용도 많이 소요된다. 그러나 자료 동기화 시스템을 이용하면, 필요한 자료에 대해서만 전송이 일어나므로 유리하다. 둘째, 자료 동기화 도중에 통신 중단이 발생하더라도 (그 이후에) 자동으로 자료 동기화를 보장해 준다. 현재의 경매 시스템에서는 통신 중단 사고가 일어나면, 사용자 스스로 수동으로 조작하여 다시 서버에 접근하여야 하는 불편이 있다. 셋째, 본 논문에서 사용하는 SyncML 자료 동기

화 방법은 현재 세계적인 표준으로 인정받고 있으므로, 향후 다양한 종류의 이동무선통신 단말기들에게 널리 사용될 수 있다. 넷째, 본 연구자가 소속된 팀이 개발한 SyncML 자료 동기화 시스템을 사용하기 때문에, 구입 및 유지비용이 거의 없을 뿐만 아니라, 상황에 맞게 변경이 적은 비용이 언제든지 가능하다.

본 연구의 경매 시스템은 ThinkSync 자료 동기화 시스템을 기반으로 LINUX와 C++ 환경에서 구축되었다. 이동무선통신 단말기로서는 PocketPC 2002 Emulator[6]를 사용하였다. 앞으로, 다양한 인증 방법을 구현하여 보다 안전한 경매가 이루어 질 수 있도록 하여야 하겠고, 여러 대의 단말기들이 동시에 입력을 시도할 때에 일어날 수 있는 자료 충돌에 대한 효율적인 해결책을 모색하기 위한 연구가 더 필요하다고 생각된다.

#### 참 고 문 헌

- 장대진, 김현정, 주홍택, 박기현, "SyncML 기반의 단말기 관리시스템 설계", 한국정보처리학회 추계학술발표논문집, 제10권, 제2호, 2003년 11월, pp. 1003-1006,
- S. Agarwal, D. Starobinski and A. Trachtenberg, "On the Scalability of Data Synchronization Protocols for PDAs and Mobile Devices," *IEEE Networks*, Vol. 16, No. 4, June 2002, pp.22-28.
- U. Hansmann, R. Mettala, A. Purakayastha, P. Thompson, SYNCML: Synchronizing and Managing Your Mobile Data, Prentice Hall, 2002.
- Hong Taek Ju, D. J. Jang, KeeHyun Park, B. H. Ha, M. C. Lee, Sung-Chae Bae, "Design of ThinkSync DM based on SyncML Device Management," The 3rd Asia Pacific International Symposium on Information Technology, KIPS, December 2004, pp. 569-574.
- Microsoft Co., <http://msdn.microsoft.com/library>
- Microsoft Co., <http://www.microsoft.com/windows>

- mobile/resources.
- Palm Inc., Palm Developer Knowledgebase Manuals, <http://palmos.com/dev/support/docs/palmos/ReferenceTOC.html>.
- SyncML Initiative Ltd., "Building an Industry-Wide Mobile Data Synchronization Protocol," SyncML White Paper, November 2001.
- SyncML Initiative Ltd., "SyncML Representation Protocol, Device Management Usage SyncML," <http://www.syncml.org>, 2002.
- SyncML Initiative Ltd., "SyncML Device Management Protocol," <http://www.syncml.org>, 2002.
- SyncML Initiative Ltd., "SyncML Device Management Tree and Description Session," <http://www.syncml.org>, 2002.
- SyncML Initiative Ltd., "SyncML Device Management Standardised Objects," <http://www.syncml.org>, 2002.
- SyncML Initiative Ltd., "SyncML Representation Protocol v1.1," <http://www.syncml.org>, 2002.

Information System Review  
Volume 6 Number 1  
June 2004

## Construction of a Bidding System Using SyncML Data Synchronization Mechanism

Kee Hyun Park\* · Sang Jin Yoo\*\*

### Abstract

As the necessity of such mobile wireless terminals as PDAs or Cellular phones drastically increases, the terminals play important role in various business areas. In this paper, a bidding system in mobile wireless environments is constructed. However, with many mobile wireless terminals modifying at the same time their (duplicated) bidding values for a given data item, data coherency for the data item becomes essential.

Therefore, in this paper, a bidding system using SyncML data synchronization mechanism is proposed and constructed. SyncML is one of data synchronization mechanisms proposed by the SyncML Initiative, a consortium of mobile wireless mobile technology leaders for universal data synchronization. The bidding system is implemented and tested in LINUX, C++, and PocketPC 2000 emulator programming environments.

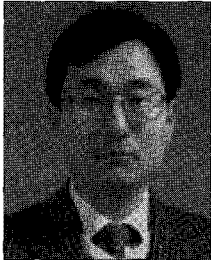
**Keywords:** *Bidding System, Mobile Wireless Terminals, Data Synchronization, SyncML*

---

\* Professor, Dept. of Computer Science, Keimyung University

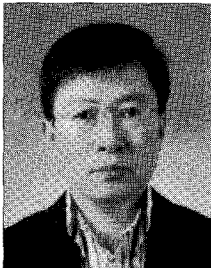
\*\* Professor, Dept. of MIS, Keimyung University

## ○ 저 자 소 개 ○



**박 기 현 (khp@kmu.ac.kr)**

경북대학교에서 전자공학과 컴퓨터를 전공하였고, 한국과학기술원 전자계산학과에서 석사학위를 받았으며, 미국 Vanderbilt 대학교에서 전자계산학과 학위를 취득하였다. 현재 계명대학교 컴퓨터 및 전자공학부 교수로 재직 중이며, 주요 관심분야는 네트워크 소프트웨어, 병렬/분산처리 시스템, 운영체제, 성능분석 등이다.



**유 상 진 (yoosj@kmu.ac.kr)**

서강대학교에서 물리학(이학사), 경영학(경영학사)을 복수 전공하였으며, 미국 University of Nebraska-Lincoln에서 MIS전공으로 박사학위를 취득하였다. 현재 계명대학교 경영정보학과 교수로 재직 중이며, 현직에 오기 전에는 미국 Bowling Green State University에서 조교수로 재직하였다. 한국경영정보학회 부회장, 한국정보시스템학회 회장을 역임하였으며, 현재 대구경북 CIO협의회 회장, 대구경북 ECRC전문위원, 대한상사중재원 중재인으로 활동하고 있다. 주요 관심분야는 IS/IT의 전략적 활용, 경영혁신, 지식경영, 인터넷창업 등이다.