



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0097117  
(43) 공개일자 2011년08월31일

(51) Int. Cl.

G06F 9/44 (2006.01) G06F 15/16 (2006.01)  
H04W 12/06 (2009.01)

(21) 출원번호 10-2010-0016776

(22) 출원일자 2010년02월24일

심사청구일자 2010년02월24일

(71) 출원인

계명대학교 산학협력단

대구 달서구 신당동 1000번지

(72) 발명자

박기현

대구광역시 달서구 신당동 1000번지 계명대학교  
공학 1호관 1327호

박주건

대구광역시 달성군 다사읍 서재리 37-8번지

(74) 대리인

김일환

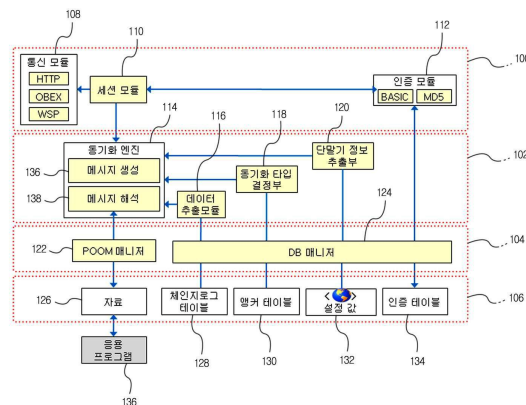
전체 청구항 수 : 총 4 항

(54) 자료 동기화를 위한 모바일 단말기 및 그 방법

### (57) 요약

본 발명에 따르는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 장치는, 자료 동기화 서버와의 통신을 담당하는 프로토콜 매니저; 응용 프로그램에 의한 변경된 데이터에 대한 LUID 및 데이터를 제공받아 데이터를 해쉬값으로 변환하고, 상기 LUID에 대응되는 해쉬값을 체인지 로그 테이블로부터 제공받아 비교하고, 두 해쉬값이 상이하면 동기화 명령 및 LUID 및 해쉬값으로 구성되는 동기화 메시지를 상기 프로토콜 매니저를 통해 상기 자료 동기화 서버로 전송하고, 상기 자료 동기화 서버로부터 동기화 메시지가 수신되면, 그 동기화 메시지에서 동기화 명령 및 LUID 및 데이터를 분석해내고, 상기 데이터를 해쉬값으로 변환한 후에, LUID 및 해쉬값을 상기 체인지 로그 테이블에 반영토록 출력하고, 상기 동기화 명령 및 LUID 및 데이터를 상기 응용 프로그램에 반영하는 동기화 매니저; 상기 체인지 로그 테이블로부터 LUID 및 해쉬값을 추출하여 상기 동기화 매니저에 제공하거나, 상기 LUID 및 해쉬값을 제공받아 상기 체인지 로그 테이블에 등록하는 데이터 스토어 매니저; 상기 체인지 로그 테이블 및 상기 응용 프로그램의 데이터 저장영역을 제공하는 데이터스토어; 를 구비함을 특징으로 한다.

대표도 - 도1



## 특허청구의 범위

### 청구항 1

모바일 단말기를 위한 임베디드 자료 동기화 에이전트 장치에 있어서,

자료 동기화 서버와의 통신을 담당하는 프로토콜 매니저;

응용 프로그램에 의한 변경된 데이터에 대한 LUID 및 데이터를 제공받아 데이터를 해쉬값으로 변환하고, 상기 LUID에 대응되는 해쉬값을 체인지 로그 테이블로부터 제공받아 비교하고, 두 해쉬값이 상이하면 동기화 명령 및 LUID 및 해쉬값으로 구성되는 동기화 메시지를 상기 프로토콜 매니저를 통해 상기 자료 동기화 서버로 전송하고,

상기 자료 동기화 서버로부터 동기화 메시지가 수신되면, 그 동기화 메시지에서 동기화 명령 및 LUID 및 데이터를 분석해내고, 상기 데이터를 해쉬값으로 변환한 후에, LUID 및 해쉬값을 상기 체인지 로그 테이블에 반영토록 출력하고,

상기 동기화 명령 및 LUID 및 데이터를 상기 응용 프로그램에 반영하는 동기화 매니저;

상기 체인지 로그 테이블로부터 LUID 및 해쉬값을 추출하여 상기 동기화 매니저에 제공하거나, 상기 LUID 및 해쉬값을 제공받아 상기 체인지 로그 테이블에 등록하는 데이터 스토어 매니저;

상기 체인지 로그 테이블 및 상기 응용 프로그램의 데이터 저장영역을 제공하는 데이터스토어;

를 구비함을 특징으로 하는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 장치.

### 청구항 2

제1항에 있어서,

상기 프로토콜 매니저에는,

상기 자료 동기화 서버에 대한 인증을 위한 인증 모듈이 더 구비됨을 특징으로 하는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 장치.

### 청구항 3

모바일 단말기를 위한 임베디드 자료 동기화 에이전트 방법에 있어서,

응용 프로그램에 의한 변경된 데이터에 대한 LUID 및 데이터를 제공받아 데이터를 해쉬값으로 변환하고, 상기 LUID에 대응되는 해쉬값을 체인지 로그 테이블로부터 제공받아 비교하고, 두 해쉬값이 상이하면 동기화 명령 및 LUID 및 해쉬값으로 구성되는 동기화 메시지를 상기 프로토콜 매니저를 통해 상기 자료 동기화 서버로 전송하는 단계;

상기 자료 동기화 서버로부터 동기화 메시지가 수신되면, 그 동기화 메시지에서 동기화 명령 및 LUID 및 데이터를 분석해내고, 상기 데이터를 해쉬값으로 변환한 후에, LUID 및 해쉬값을 상기 체인지 로그 테이블에 반영토록 출력하고,

상기 동기화 명령 및 LUID 및 데이터를 상기 응용 프로그램에 반영하는 단계;

를 구비함을 특징으로 하는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 방법.

### 청구항 4

제3항에 있어서,

상기 자료 동기화 서버에 대한 인증을 이행하는 단계가 더 구비됨을 특징으로 하는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 방법.

## 명세서

## 기술분야

- [0001] 본 발명은 모바일 단말기를 위한 동기화 기술에 관한 것으로, 더욱 상세하게는 OMA(Open Mobile Alliance)에서 제시한 표준 동기화 방식인 SyncML 자료 동기화 방식을 이용하여, 각각의 단말기 회사에서 제공되는 자료 동기화 방식을 그대로 사용하면서 상호간에 동기화가 가능하도록 하는 모바일 단말기를 위한 임베디드 데이터 동기화 에이전트 장치 및 방법에 관한 것이다.

## 배경기술

- [0002] 최근 이동 컴퓨팅 기술과 하드웨어 기술의 발달로 인해 모바일 단말기가 매우 급속하게 보급되고 있다. 상기 모바일 단말기의 사용자가 시간이나 장소에 구애받지 않고 원하는 정보를 이용하기 위해서는 중앙 통합 서버의 자료와 모바일 단말기의 자료를 동기화시키는 자료 동기화(Data Synchronization)가 요구된다.
- [0003] SyncML은 각 제조회사의 단말기가 가지고 있는 서로 다른 플랫폼, 통신 프로토콜, 자료 형태, 응용 서비스에 이용될 수 있는 자료 동기화 방식과 장치 관리에 대한 개방형 표준 인터페이스 개발을 목적으로 제안되었다. 또한 상기 SyncML은 개방형 표준과 다수의 구현 방식을 따르는 것을 채택함으로써 자료 동기화에 대한 사실상의 표준이 된다.
- [0004] 상기 SyncML은 모바일 단말기와 서버간의 원격 동기화를 목표로 설계되었으나, 로컬 동기화와 유선 네트워크로 연결된 장치들 간의 동기화에도 사용될 수 있다. 상기 SyncML을 기반으로 하는 클라이언트와 서버간에 동기화가 수행되는 동안, 동기화 대상이 되는 객체는 논리적으로 패키지를 교환한다.
- [0005] 또한 각 패키지는 모바일 단말기의 제한된 자원과 무선 통신상의 낮은 대역폭으로 인해 다수의 메시지로 분리되어 교환된다. 한 번의 완전한 자료 동기화를 위해 클라이언트와 서버 간에 주고받은 모든 패키지를 세션이라고 하며, 세션은 여러 개의 패키지를 포함하고 하나의 패키지는 여러 메시지를 포함한다.
- [0006] SyncML 자료 동기화 규격은 XML 기반의 SyncML 자료 표현(Data Representation) 프로토콜, SyncML 동기화(Synchronization) 프로토콜 그리고 SyncML 전송 바인딩(Transport Bindings) 프로토콜로 구성되어 있다. 상기 SyncML 자료 표현 프로토콜은 자료 동기화를 위해 교환되는 SyncML 메시지의 논리적인 구조와 형태를 XML 형식으로 정의하고 있다. 즉, 각각의 필드가 어떠한 정보를 담고 있으며 해당 정보가 어떤 의미를 내포하는 것인지에 대한 약속을 정의하고 있다. 상기 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. 마지막으로, SyncML 전송 바인딩 프로토콜은 SyncML이 동기화 메시지를 전송하기 위해서 사용하는 HTTP, WSP, OBEX 등과 같은 프로토콜인데, 상기 SyncML 규격에서는 SyncML 전송 바인딩 프로토콜에 대해서 새로운 프로토콜을 정의하는 것이 아니라, 전송 프로토콜과 바인딩 규칙만을 정의한다.
- [0007] HotSync 프로토콜은 Palm OS를 기반으로 하는 단말기에서 동작하는 자료 동기화 방식이다. Conduit이라는 코드 모듈을 이용하여 Palm OS 기기와 데스크 탑 컴퓨터 간의 자료 동기화를 수행하는 방식이다. Palm OS 기반의 HotSync 프로토콜은 하위 Bearer에 대한 기술적인 규격에 대하여 공개하고 있지 않다. 따라서, Palm OS 기반의 응용 소프트웨어는 HotSync 솔루션을 가지고 있는 회사에서 개발 및 공급을 하고 있다. XcelleNet, Mobile Automation, Pumatech, AdantGo 등과 같은 회사가 HotSync에 대한 솔루션을 제공하고 있다.
- [0008] HotSync 클라이언트는 HotSync 매니저, Conduit 모듈, Notifier DLL 모듈, Handheld App. 모듈, Desktop App. 모듈, Sync 매니저 API와 같은 6개의 컴포넌트로 구성되어 있다. Hotsync 프로토콜은 SyncML 프로토콜의 Two-way와 유사한 Slow-sync와 Fast-sync 두 가지 자료 동기화 모드를 지원한다. Fast Sync가 Slow Sync보다 속도가 빠르기 때문에, Slow Sync보다 Fast Sync를 우선적으로 수행한다.
- [0009] 현재 주요 단말기 제조 회사에서 제공하는 자료 동기화 솔루션이 존재하지만, 각각의 단말기와 응용 서비스 간의 호환성이 결여되어 있다. 따라서, 각 모바일 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.
- [0010] 하지만 각 단말기 제조회사들이 기술적인 경쟁을 통해서 시장점유율을 높이고 있기 때문에, 각각의 동기화 방식에 대한 기술적인 공개를 하지 않고 있다. 따라서, SyncML이 동기화 솔루션에 대한 표준화를 시도하고 있지만, 각각의 동기화 방식을 탑재한 모바일 단말기들이 모두 SyncML 방식을 동기화 표준화로 대체하기는 어려운 실정

이다.

## 발명의 내용

### 해결하려는 과제

- [0011] 본 발명은 OMA(Open Mobile Alliance)에서 제시한 표준 동기화 방식인 SyncML 자료 동기화 방식을 이용하여, 모바일 단말기 회사에서 제공되는 다양한 자료 동기화 방식을 그대로 사용하면서 상호간에 동기화를 가능하게 할 수 있는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 장치 및 방법을 제공하는 것을 그 목적으로 한다.

### 과제의 해결 수단

- [0012] 상기한 목적을 달성하기 위한 본 발명에 따르는 모바일 단말기를 위한 임베디드 자료 동기화 에이전트 장치는, 자료 동기화 서버와의 통신을 담당하는 프로토콜 매니저; 응용 프로그램에 의한 변경된 데이터에 대한 LUID 및 데이터를 제공받아 데이터를 해쉬값으로 변환하고, 상기 LUID에 대응되는 해쉬값을 체인지 로그 테이블로부터 제공받아 비교하고, 두 해쉬값이 상이하면 동기화 명령 및 LUID 및 해쉬값으로 구성되는 동기화 메시지를 상기 프로토콜 매니저를 통해 상기 자료 동기화 서버로 전송하고, 상기 자료 동기화 서버로부터 동기화 메시지가 수신되면, 그 동기화 메시지에서 동기화 명령 및 LUID 및 데이터를 분석해내고, 상기 데이터를 해쉬값으로 변환한 후에, LUID 및 해쉬값을 상기 체인지 로그 테이블에 반영토록 출력하고, 상기 동기화 명령 및 LUID 및 데이터를 상기 응용 프로그램에 반영하는 동기화 매니저; 상기 체인지 로그 테이블로부터 LUID 및 해쉬값을 독출하여 상기 동기화 매니저에 제공하거나, 상기 LUID 및 해쉬값을 제공받아 상기 체인지 로그 테이블에 등록하는 데이터 스토어 매니저; 상기 체인지 로그 테이블 및 상기 응용 프로그램의 데이터 저장영역을 제공하는 데이터스��어; 를 구비함을 특징으로 한다.

### 발명의 효과

- [0013] 상기한 본 발명은 WinCE 또는 Window Mobile 기반의 모바일 단말기에 탑재된 ActiveSync 자료 동기화 개체들은 서로 다른 동기화 방식을 탑재하고 있는 모바일 단말기와는 상호 호환성이 결여되어 있다. 그러나 본 발명에서 개발한 임베디드 자료 동기화 에이전트는 서로 다른 동기화 방식들 간의 상호 호환성을 보장할 수 있다.
- [0014] 또한 본 발명에서는 OMA에서 제시한 OMA DS SyncML 프로토콜 스펙을 기반으로 개발하였으므로, 국제 표준을 준수하고 있으며, 또한 향후 OMA DS SyncML 프로토콜이 자료 동기화 방식에서 세계적인 표준기술로 채택되는 점에 기여한 점이 있다.
- [0015] 또한 본 발명과 유사한 기술로는 특허등록 100844361호(2008.07.01)의 자료동기화 게이트웨이 시스템이 있지만 이 시스템은 여러 대의 단말기들로부터 수신한 동기화 요청에 의해 동시에 처리할 데이터의 양이 증가함으로, 게이트웨이에 병목 현상이 발생할 수 있다. 또한 게이트웨이 내부에서 각 단말기 또는 동기화 서버로부터 수신한 데이터를 변환하기 위해서는 암호화된 데이터를 복호화하는 과정이 필요하다. 만약 이 과정에서 보안 공격을 당한다면 민감한 개인 자료들이 외부에 노출될 수 있다. 본 발명의 에이전트는 이러한 단점을 보완하기 위해 임베디드 방식으로 개발되었으며 각 사용자의 모바일 단말기 내에 탑재되고, 단말기 내에서 자료를 추출하여 변환하므로, 보안상의 취약점과 병목현상을 완화할 수 있다.

### 도면의 간단한 설명

- [0016] 도 1은 본 발명에 따른 SyncML 기반의 자료 동기화 게이트웨이 시스템 구성도.  
 도 2는 본 발명에 따른 SyncML 클라이언트와 Sync. 프로토콜 매니저 간의 처리 흐름도.  
 도 3은 본 발명에 따른 자료 동기화 게이트웨이 시스템의 처리 흐름도.  
 도 4는 본 발명에 따른 SyncML 기반의 자료 동기화 게이트웨이 동작 메커니즘을 도시한 도면.  
 도 5는 본 발명에 따른 전체 동기화 시간과 변환 게이트웨이에서 소비 시간에 대한 그래프를 도시한 도면.

### 발명을 실시하기 위한 구체적인 내용

- [0017] 본 발명은 OMA(Open Mobile Alliance)에서 제시한 표준 동기화 방식인 SyncML 자료 동기화 방식을 이용하여, 모

바일 단말기 회사에서 제공되는 다양한 자료 동기화 방식을 그대로 사용하면서 상호간에 동기화를 가능하게 한다.

- [0018] 도 1은 본 발명의 바람직한 실시예에 따른 임베디드 자료 동기화 에이전트 장치의 구성을 도시한 것이다.
- [0019] 상기 임베디드 자료 동기화 에이전트 장치는 자료 변환 방식을 가지고 있으며 모바일 단말기 내에 탑재된다. 상기 임베디드 자료 동기화 에이전트 장치는 ActiveSync 방식의 메시지를 OMA DS SyncML 기반의 메시지로 변환하거나, OMA DS SyncML 기반의 메시지를 ActiveSync 방식의 메시지로 변환한다. 상기 임베디드 자료 동기화 에이전트 장치는 프로토콜 매니저(100), 동기화 매니저(102), 데이터스토어 매니저(104) 그리고 데이터스토어(106)로 구성된다.
- [0020] 상기 프로토콜 매니저(100)는 통신 모듈(108), 세션 모듈(110), 인증 모듈(112)로 구성된다.
- [0021] 상기 통신 모듈(108)은 자료 동기화 에이전트와 미도시된 자료 동기화 서버간의 동기화 메시지 교환을 담당한다.
- [0022] 상기 세션 모듈(110)은 자료 동기화가 완전히 종료될 때까지 서버와의 세션을 유지하며, 동기화 메시지의 헤더(header) 부분을 해석하고 메시지의 적합성을 검증한다. 여기서, 자료 동기화가 시작되면 모바일 단말기와 자료 동기화 서버는 동기화 메시지를 교환하게 되는데, 한 번의 완전한 자료 동기화를 위해 모바일 단말기와 자료 동기화 서버 사이에 주고 받은 모든 패킷을 세션(Session)이라고 한다.
- [0023] 상기 인증 모듈(112)은 자료 동기화를 수행하기에 앞서 권한이 있는 자료 동기화 서버로부터 온 동기화 요청인지 체크한다. 상기 인증 모듈(112)은 데이터스토어(106)의 인증 테이블(134)에 저장된 자료 동기화 서버 인증 정보를 가져와 동기화 시, 자료 동기화 서버가 전송한 인증 정보와 비교한다. 상기 인증 모듈(112)의 인증 방법으로는 OMA DS에서 사용하는 BASIC과 MD5 해쉬 알고리즘이 채용될 수 있다.
- [0024] 상기 동기화 매니저(102)는 실제 자료 동기화를 수행하며, 동기화 엔진(114), 데이터 추출모듈(116), 동기화 타입 결정부(118), 단말기 정보 추출부(120)로 구성된다.
- [0025] 상기 동기화 엔진(114)은 자료 동기화 서버로부터 수신한 동기화 메시지를 해석하여 동기화 명령 및 동기화 자료를 추출하고, 데이터스토어 매니저(104)를 통해 데이터스토어(106)에 반영한다. 상기 동기화 엔진(114)에 의해 해석된 메시지 구성은 도 2와 같다.
- [0026] 상기 동기화 메시지는 모바일 단말기 자료의 고유한 ID인 LUID, 추가(ADD), 삭제(DELETE), 수정(REPLACE)을 나타내는 동기화 명령, 그리고 연락처, 일정과 같은 실제 동기화 데이터로 구성된다.
- [0027] 그리고 모바일 단말기측 변경사항이 있을 때, 또는 자료 동기화 서버측 동기화 요청에 대한 응답 메시지를 송신할 때 동기화 엔진(114)은 동기화 메시지를 생성한다. 상기 동기화 메시지는 현재 널리 사용되고 있고, OMA DS에서도 사용하는 XML 형태로 생성한다.
- [0028] 상기 데이터 추출모듈(116)은 변경사항 추출을 위해 체인지 로그 테이블을 사용한다. 상기 체인지 로그 테이블에는 각 자료의 상태 플래그, 변경 시각 등이 기록되어 있다. 상기 체인지 로그 테이블을 바탕으로, 마지막 동기화 이후 변경된 자료만을 추출하여 동기화 엔진(114)에게 전달한다.
- [0029] 상기 동기화 타입 결정부(118)는 전체 자료를 전송(Slow Sync)할 것인지, 변경된 자료만을 전송(Fast Sync)할 것인지를 결정하기 위해 앵커 테이블을 사용한다. 상기 앵커 테이블에는 이전 동기화 시각이 저장되어 있다. 만약 자료 동기화 에이전트와 자료 동기화 서버의 앵커 정보가 다르다면 전체 자료를 전송하고, 같다면 변경된 자료만을 전송한다.
- [0030] 상기 단말기 정보 추출부(120)는 모바일 단말기가 수용할 수 있는 동기화 타입, 단말기의 메모리 용량 등을 자료 동기화 서버에 알려주기 위한 단말기 설정 값을 추출한다. 상기 단말기 설정 값은 XML 문서로 저장되어 있다.
- [0031] 상기 데이터스토어 어댑터(104)는 POOM(Pocket Outlook Object Model) 매니저(122)와 DB 매니저(124)로 구성된다.
- [0032] 상기 POOM 매니저(122)는 응용 프로그램에서 사용하는 데이터를 추출 및 저장, 수정한다. 상기 POOM은 윈도우 모바일 또는 윈도우 CE 기반의 단말기 응용 프로그램의 PIMS(Personal Information Management System) 정보를 나타내는 클래스들의 집합이다. 상기 POOM 매니저(122)가 POOM 라이브러리를 사용하여 데이터를 추출, 저장, 삭

제, 수정하는 과정은 다음과 같이 3단계로 구성된다.

[0033] 1) OutlookSession 객체를 생성하여 POOM 세션 시작

[0034] 2) ItemID 객체를 사용하여 응용프로그램이 사용하는 데이터의 고유한 ID인 LUID 추출

[0035] 3-1) 데이터 추가 시 데이터를 매개변수로 하는 Add() 메소드 호출, 새롭게 할당된 LUID를 반환

[0036] 3-2) 데이터 삭제 시 LUID를 매개변수로 하는 Remove() 메소드 호출

[0037] 3-3) 데이터 수정 시 LUID와 데이터를 매개변수로 하는 Update() 메소드 호출

[0038] 상기 DB 매니저(124)는 체인지 로그, 앵커, 설정 값, 인증값을 추출 및 저장, 수정한다. 상기 DB 매니저(124)는 상기 POOM 매니저(122)가 추출한 데이터의 LUID를 참고하여 쿼리문을 사용하여 체인지 로그에 저장된 해쉬값을 추출한다. 상기 체인지 로그는 윈도우 모바일에서 제공하는 CEDB를 사용하며 응용 프로그램이 사용하는 데이터와 쌍을 이루는 LUID와 해쉬값을 가지고 있다. 상기 CEDB는 윈도우 모바일 기반의 단말기에서 사용하는 데이터베이스이다. 상기 DB 매니저(124)가 CEDB 쿼리를 사용하여 해쉬값을 추출, 저장, 삭제, 수정하는 과정은 다음과 같다.

[0039] 1) SqlConnection 객체를 생성하여 데이터베이스에 접속

[0040] 2) POOM 매니저(122)가 추출한 데이터의 LUID와 동일한 레코드를 선택하기 위해 ExecuteReader() 메소드를 사용하여 SELECT 쿼리문을 실행

[0041] 3) ExecuteNonQuery() 메소드를 사용하여 INSERT, DELETE, UPDATE 쿼리문 실행

[0042] 도 3은 데이터 추출 모듈(116)의 세부 구조도를 도시한 것이다. 상기 데이터 추출 모듈(116)은 POOM 매니저(122)와 DB 매니저(124)로부터 전달받은 한 쌍의 데이터를 비교하여 변경 상태(동기화 명령)와 LUID, 데이터를 동기화 엔진(114)에 전달한다. 예를 들어 LUID가 3인 데이터 ABC가 응용 프로그램에서 추가되었다면 데이터 추출 모듈(116)은 <ADD, LUID#3, ABC> 라는 메시지를 동기화 엔진에 전달한다. 또한 자료 동기화 서버로부터 동기화 대상 데이터와 동기화 명령을 수신하면 추가(ADD)/삭제(DELETE)/수정(REPLACE)과 같은 동기화 명령과 LUID, 데이터를 POOM 매니저(122)에게 전달한다. 또한 데이터의 해쉬값을 계산하여 LUID와 함께 DB 매니저(124)에게 전달한다.

[0043] 도 4는 본 발명에 따른 임베디드 자료동기화 에이전트의 흐름도를 도시한 것이다. 상기 도 4를 참조하여 본 발명에 따른 임베디드 자료 동기화 에이전트의 동작을 설명한다.

[0044] POOM 매니저(122)는 응용 프로그램에서 변경된 데이터를 추출하여 데이터의 고유 ID인 LUID와 데이터를 결합하여 데이터 추출모듈(116)로 제공한다. 그리고 DB 매니저(124)는 POOM 매니저(122)가 추출한 데이터의 LUID를 참고하여 체인지 로그에 저장된 해쉬값을 추출하여, 상기 LUID와 해쉬값을 결합하여 데이터 추출모듈(116)로 제공한다.

[0045] 상기 데이터 추출모듈(116)은 상기 POOM 매니저(122)가 송신한 LUID와 데이터에 대해서는 상기 데이터를 해쉬값으로 변환한다. 이후 상기 데이터 추출모듈(116)은 상기 LUID와 해쉬값과 상기 DB 매니저(124)로부터의 LUID와 해쉬값을 비교하여, 두 값이 다르면 동기화 명령과 LUID와 해쉬값을 결합하여 메시지 생성모듈(136)로 제공한다. 상기 메시지 생성모듈(136)은 상기 동기화 명령과 LUID, 해쉬값을 결합하여 동기화 메시지를 생성하여 자료 동기화 서버로 전송한다.

[0046] 그리고 자료 동기화 서버로부터의 동기화 메시지가 수신되면, 메시지 분석 모듈(138)은 상기 동기화 메시지를 동기화 명령, LUID, 데이터로 분석한 후에 데이터 추출모듈(116)에 제공한다.

[0047] 상기 데이터 추출모듈(116)은 상기 동기화 메시지에서 분석해낸 데이터를 해쉬값으로 변환한 후에 DB 매니저(124)로 제공하며, 상기 DB 매니저(124)는 상기 동기화 메시지에 포함된 LUID와 해쉬값을 체인지 로그에 저장한다.

[0048] 이후 상기 데이터 추출모듈(116)은 상기 동기화 메시지에 포함된 동기화 명령, LUID, 데이터를 POOM 매니저(122)에 제공한다. 상기 POOM 매니저(122)는 상기 동기화 명령, LUID, 데이터에 따라 데이터의 저장/삭제/수정 등의 동기화 동작을 이행한다.

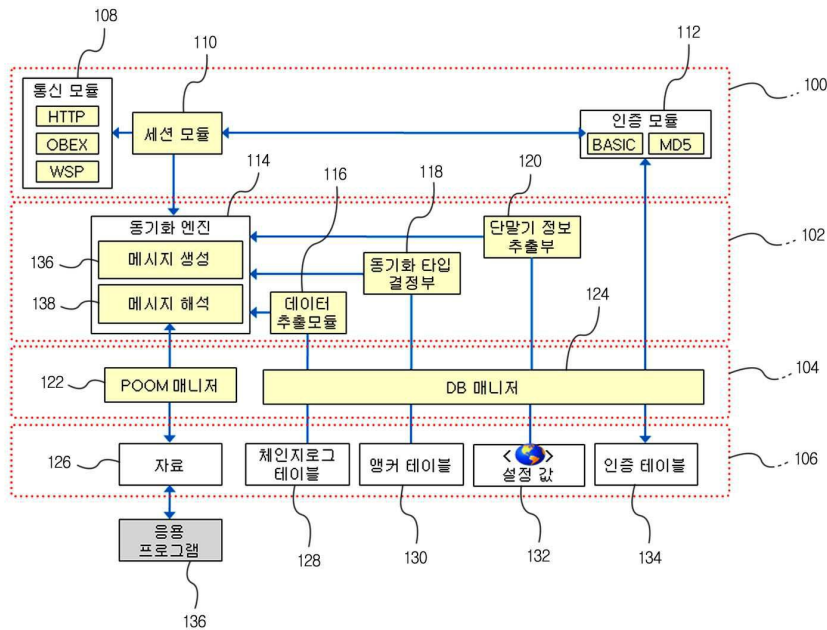
## 부호의 설명

[0049]

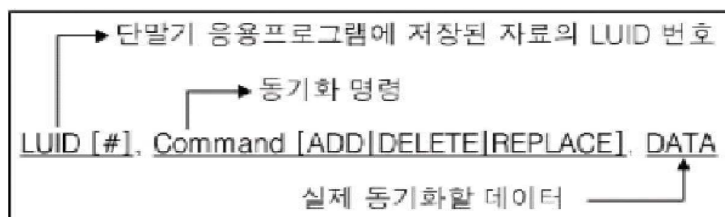
- 100: 프로토콜 매니저
- 102 : 동기화 매니저
- 104 : 데이터스토어 매니저
- 106 : 데이터스토어

## 도면

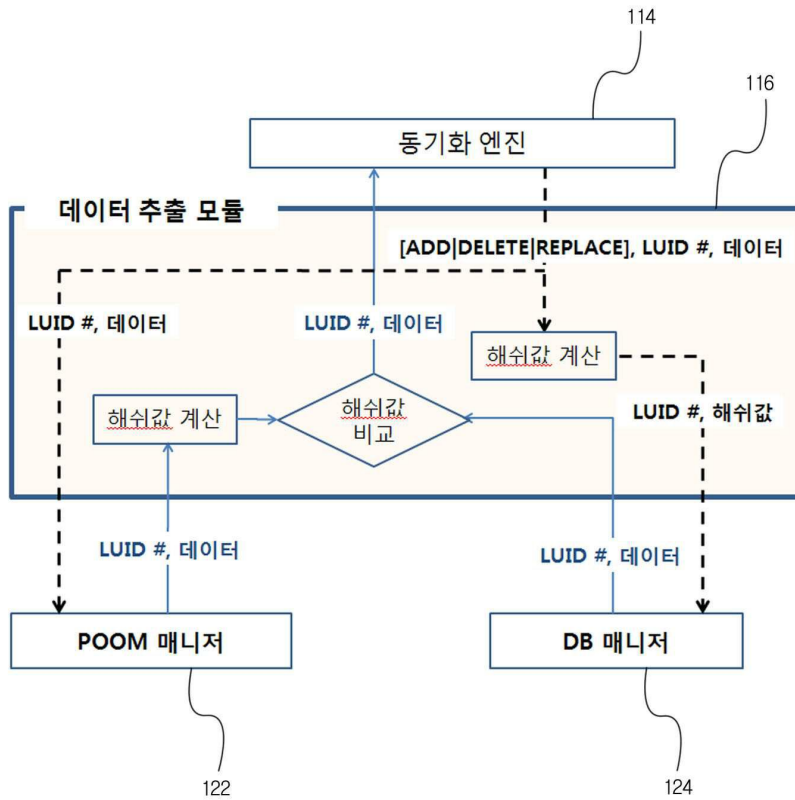
### 도면1



### 도면2



도면3



도면4

