



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년04월19일
(11) 등록번호 10-1137101
(24) 등록일자 2012년04월09일

(51) 국제특허분류(Int. Cl.)

H04L 7/02 (2006.01)

(21) 출원번호 10-2005-0010522

(22) 출원일자 2005년02월04일

심사청구일자 2010년01월28일

(65) 공개번호 10-2006-0041737

(43) 공개일자 2006년05월12일

(30) 우선권주장

10/771,896 2004년02월04일 미국(US)

(56) 선행기술조사문헌

US20020133508 A1

EP1130511 A2

US20020120688 A1

WO2003042863 A1

전체 청구항 수 : 총 19 항

(73) 특허권자

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

라지브, 비리세티

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

홀, 그레이

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

(뒷면에 계속)

(74) 대리인

제일특허법인

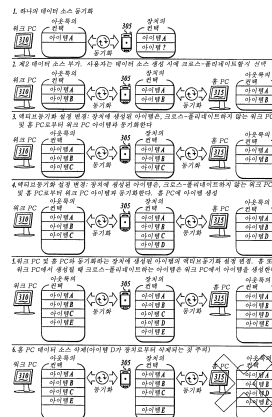
심사관 : 이형일

(54) 발명의 명칭 다중 동기화 소스의 크로스-폴리네이션

(57) 요약

본 발명은, 사용자가, 크로스-폴리네이트하는 적어도 2개의 데이터 소스들과 장치를 동기화할 수 있도록 한다. 사용자의 장치는 소스들 간의 변경을 전달하고, 변경이 다중 소스의 아이템에서 동시에 이루어질 때 충돌을 해결 하는데 사용된다. 사용자의 장치는 각 동기화 소스가 어느 버전의 아이템을 갖는지를 계속 추적하고, 이 소스들 각각을 최종 버전의 아이템으로 동기화한다.

대표도 - 도4



(72) 발명자

슬레이퍼, 제이슨, 에반

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

발두치, 주안 브이. 에스테브

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

포스터, 마이크

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

라마니, 프리디

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

양, 지동

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이 마이크로소프트 코포레이션 내

특허청구의 범위

청구항 1

장치를 데이터 소스들과 동기화하고, 상기 데이터 소스들을 크로스-폴리네이션(cross-pollination)하도록 하는 방법으로서,

제1 데이터 소스 및 제2 데이터 소스를 생성하는 단계;

상기 제1 데이터 소스, 상기 제2 데이터 소스 및 상기 장치 간의 동기화를 위한 아이템들을 자동으로 결정하는 단계 - 상기 제1 데이터 소스, 상기 제2 데이터 소스 및 상기 장치는 특정 사용자와 연관된 사용자 장치들이며, 상기 아이템들은 사용자가 생성한 아이템들(user created items)을 포함하고, 상기 장치는 상기 제1 데이터 소스 및 상기 제2 데이터 소스 상의 각 아이템의 버전을 결정하고 상기 제1 데이터 소스 및 상기 제2 데이터 소스 각각을 아이템 각각의 최신 버전에 동기화하도록 구성됨 - ;

상기 장치를 상기 제1 데이터 소스에 접속시키는 단계;

상기 장치를 상기 제1 데이터 소스와 동기화하는 단계 - 상기 동기화 이후에 상기 장치 및 상기 제1 데이터 소스 각각은 동일한 버전의 아이템들을 포함함 -;

상기 장치를 상기 제2 데이터 소스에 접속시키는 단계;

상기 장치를 상기 제2 데이터 소스와 동기화하는 단계 - 상기 장치는 상기 제1 데이터 소스와 상기 제2 데이터 소스 간에서 크로스-폴리네이트(cross-pollinate)하는 데 사용되며, 따라서 상기 제1 데이터 소스, 상기 제2 데이터 소스 및 상기 장치 각각은 상기 동기화 및 크로스-폴리네이트 이후에 동일한 버전의 아이템들을 포함함 - ; 및

아이템이 이미 동기화되었는지를 판정하는 중복 검출 검사(duplicate detection check)를 수행하는 단계를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 중복 검출 검사를 수행하는 단계는 특성 비교를 수행하는 단계를 더 포함하는 방법.

청구항 3

제1항에 있어서,

상기 중복 검출 검사를 수행하는 단계는 동기화 해시값(sync hash value)을 계산하는 단계를 더 포함하는 방법.

청구항 4

제1항에 있어서,

상기 아이템이 이미 동기화되었을 때, 상기 아이템을 갱신하는 단계를 더 포함하는 방법.

청구항 5

제1항에 있어서,

삭제 명령을 수신하고 상기 삭제 명령을 수행하는 단계를 더 포함하고,

상기 삭제 명령은 소프트 삭제(soft delete) 및 하드 삭제(hard delete)로부터 선택되는 방법.

청구항 6

제1항에 있어서,

상기 데이터 소스들 간의 크로스-폴리네이션을 제한하는 단계를 더 포함하는 방법.

청구항 7

제1항에 있어서,

상기 제1 데이터 소스 및 상기 제2 데이터 소스를 생성하는 단계는

데이터 소스 타입을 지시하는 단계, 및

상기 제1 데이터 소스 및 상기 제2 데이터 소스 각각과 연관된 식별자를 저장하는 단계를 더 포함하는 방법.

청구항 8

제1항에 있어서,

상기 장치를 상기 제1 데이터 소스와 동기화하는 단계는 제1 동기화 프로토콜을 사용할 수 있고,

상기 장치를 상기 제2 데이터 소스와 동기화하는 단계는 제2 동기화 프로토콜을 사용할 수 있는 방법.

청구항 9

프로세서에 의해 실행될 때 데이터 소스들을 크로스-폴리네이트하기 위한 명령어들을 저장하는 컴퓨터 판독가능 기억 매체로서,

상기 명령어들은

장치와 동기화할 제1 데이터 소스를 생성하고 상기 장치와 동기화할 제2 데이터 소스를 생성하는 명령어 - 상기 제1 데이터 소스, 상기 제2 데이터 소스 및 상기 장치는 특정 사용자에게 의해 사용되는 장치들임 - ;

상기 장치를, 상기 장치와의 동기화를 위한 제1 아이템들을 가지는 상기 제1 데이터 소스에 접속시키는 명령어;

상기 제1 데이터 소스와 상기 장치 간의 동기화를 위한 제1 아이템들을 결정하는 명령어 - 상기 제1 아이템들은 사용자가 생성한 아이템을 포함함 - ;

상기 장치를 상기 제1 데이터 소스와 동기화하는 명령어 - 상기 동기화 이후에 상기 장치 및 상기 제1 데이터 소스 각각은 동일한 버전의 제1 아이템들을 포함함 - ;

상기 장치를 상기 제1 데이터 소스와 동기화한 후에, 상기 장치를 제2 데이터 소스에 접속시키는 명령어 - 상기 장치는 상기 제1 데이터 소스 및 상기 제2 데이터 소스 상의 각 아이템의 버전을 결정하고 상기 제1 데이터 소스 및 상기 제2 데이터 소스 각각을 아이템 각각의 최신 버전에 동기화도록 구성됨 - ;

상기 제2 데이터 소스와 상기 장치 간의 동기화를 위한 제2 아이템들을 결정하는 명령어 - 상기 동기화를 위한 제1 아이템들을 결정하는 명령어 및 상기 동기화를 위한 제2 아이템들을 결정하는 명령어는 각 아이템에 대하여 계산되고 각 아이템들과 함께 저장되는 SyncHash 값을 검사하는 명령어를 포함하고, 각 아이템에 대하여 계산된 상기 SyncHash 값은, 아이템이 중복될 경우를 고려하도록 비교되는 주 특성들(primary properties)로서 정의되는 필드 세트인 주 키셋(primary keyset), 및 두 아이템들 간에 상기 주 키셋들이 매칭되더라도, 상기 두 아이템들은 서로 중복되지 않는다는 특성의 데이터가 존재하는지 검사하는 데에 사용되는 더 큰 필드 세트인 부 키셋(second keyset)으로 구성되는 상기 SyncHash 값이 계산될 경우에 2 레벨의 특성 레벨 매칭(two levels of property level matching)을 포함함 - ;

상기 장치를 상기 제2 데이터 소스와 동기화하는 명령어 - 상기 동기화 이후에 상기 장치 및 상기 제2 데이터 소스 각각은 동일한 버전의 제2 아이템들을 포함하고, 상기 제2 데이터 소스 각각은 상기 장치와 상기 제1 데이터 소스 간에 동기화되었던 상기 제1 아이템들에 대한 갱신들을 포함하고, 상기 장치가 상기 제1 데이터 소스와 다시 동기화되는 경우, 상기 장치와 동기화되는 상기 제2 아이템들은 상기 제1 데이터 소스와 동기화됨 - ; 및

아이템이 이미 동기화되었는지를 판정하는 중복 검출 검사를 수행하는 명령어

를 포함하는 컴퓨터 판독가능 기억 매체.

청구항 10

제9항에 있어서,

상기 중복 검출 검사를 수행하는 명령어는 동기화 해시값을 계산하는 명령어를 더 포함하는 컴퓨터 판독가능 기억 매체.

청구항 11

제10항에 있어서,

삭제 명령을 수신하고 상기 삭제 명령을 수행하는 명령어를 더 포함하고,

상기 삭제 명령은 소프트 삭제 및 하드 삭제로부터 선택되고,

상기 하드 삭제는 상기 아이템을 물리적으로 삭제하며, 상기 아이템이 상기 장치, 상기 제1 데이터 소스 및 상기 제2 데이터 소스로부터 제거되도록 상기 하드 삭제가 각각의 상기 데이터 소스들 및 장치 전반에 걸쳐 전파되고,

소프트 삭제는 필터 밖으로 나간 아이템(item that has gone out of filter)이며, 상기 소프트 삭제는 상기 장치 및 상기 데이터 소스들 중 하나로부터 상기 아이템을 제거하지만 상기 데이터 소스들 중 다른 하나로부터는 상기 아이템을 제거하지 않는 컴퓨터 판독가능 기억 매체.

청구항 12

제11항에 있어서,

상기 데이터 소스들 간의 크로스-폴리네이션을 제한하는 명령어를 더 포함하는 컴퓨터 판독가능 기억 매체.

청구항 13

제12항에 있어서,

상기 제1 데이터 소스 및 상기 제2 데이터 소스를 생성하는 명령어는

데이터 소스 타입을 지시하는 명령어, 및

상기 데이터 소스들 각각과 연관된 식별자를 저장하는 명령어를 더 포함하는 컴퓨터 판독가능 기억 매체.

청구항 14

제11항에 있어서,

상기 장치를 상기 제1 데이터 소스와 동기화하는 명령어는 제1 동기화 프로토콜을 사용하고,

상기 장치를 상기 제2 데이터 소스와 동기화하는 명령어는 제2 동기화 프로토콜을 사용하는 컴퓨터 판독가능 기억 매체.

청구항 15

데이터 소스들을 크로스-폴리네이팅하기 위한 프로세서를 가지는 시스템으로서,

서로 크로스-폴리네이트할 수 있는 제1 데이터 소스 및 제2 데이터 소스를 포함하는 적어도 2개의 데이터 소스; 및

크로스-폴리네이트하기 위한 상기 적어도 2개의 데이터 소스들 간의 셔틀(shuttle)로서 작용하도록 구성되고, 상기 적어도 2개의 데이터 소스들과 동기화하도록 구성되는 장치 - 동기화 및 크로스-폴리네이트 이후에 상기 장치와 상기 적어도 2개의 데이터 소스는 동기화되도록 선택된 동일한 버전의 아이템들을 포함함 -

를 포함하고,

상기 데이터 소스들과 상기 장치는 특정 사용자에게 연관된 사용자 장치들이고,

상기 아이템들은 사용자가 생성한 아이템들을 포함하고,

상기 장치는 상기 데이터 소스들 상의 아이템 각각의 버전을 결정하고 상기 데이터 소스들 각각을 아이템 각각의 최신 버전에 동기화하도록 구성되고,

상기 장치는 먼저 상기 제1 데이터 소스와 동기화되며, 따라서 상기 동기화 이후에 상기 장치 및 상기 제1 데이터 소스 각각은 동일한 버전의 아이템들을 포함하고,

상기 장치를 상기 제1 데이터 소스와 동기화한 후에, 상기 장치를 상기 제2 데이터 소스와 동기화하며, 따라서

상기 제1 데이터 소스, 상기 제2 데이터 소스 및 상기 장치 각각은 상기 동기화 이후에 동일한 버전의 아이템들을 포함하고,

상기 제1 데이터 소스와 상기 제2 데이터 소스의 동기화 중에, 아이템이 이미 동기화되었는지를 판정하는 중복 검출 검사를 수행하는 시스템.

청구항 16

제15항에 있어서,

상기 중복 검출 검사의 수행은 동기화 해시값의 계산을 더 포함하는 시스템.

청구항 17

제15항에 있어서,

상기 장치는 소프트 삭제 명령 및 하드 삭제 명령을 처리하도록 구성되는 시스템.

청구항 18

제17항에 있어서,

상기 장치는 상기 적어도 2개의 데이터 소스 간의 크로스-폴리네이션을 제한하도록 더 구성된 시스템.

청구항 19

제18항에 있어서,

상기 장치와 상기 적어도 2개의 데이터 소스의 동기화는 2개 이상의 동기화 프로토콜을 사용할 수 있는 시스템.

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0040] 사용자는 일 장치를 다른 장치와 동기화할 수 있다. 예를 들어, 사용자는 자신의 작업 컴퓨터를 자신의 이동 장치에 동기화할 수 있다. 이동 장치와의 동기화 후에, 컴퓨터는 동기화되도록 선택된 동일한 정보를 포함한다. 동기화의 기본적인 목적은 2개의 시스템을 서로 동시에 유지하는 것이다. 일 소스에서 이루어지는 변경은 장치가 동기화될 때 다른 소스로 전파된다.

[0041] 예를 들어, 사용자는 다른 이동 장치에 대한 연락처를 자신의 홈 PC와 동기화하고, 그 다음 홈 PC와 이동 장치는, 동기화가 완료된 후에 동일한 연락처를 가질 것이다. 다수의 사용자는 그들의 문서 뿐만 아니라 이메일, 연락처, 달력, 아이템을 포함하는 아이템들을 동기화한다. 그러나, 2개 이상의 서로 다른 장치들 사이의 정보를 동기화하는 것은 매우 어렵다. 따라서, 다중 동기 소스들을 동기화하는 방법이 요구된다.

발명이 이루고자 하는 기술적 과제

- [0042] 본 발명은 장치를 2개 이상의 컴퓨팅 장치("동기 소스")와 동기화하는 것에 관한 것이다.
- [0043] 본 발명의 일 양상에 따르면, 장치는 그들의 데이터를 "크로스-폴리네이션(cross-pollination)"한다. 이동 장치와 같은 컴퓨팅 장치들 중 하나는 각 데이터 소스가 어느 버전의 아이템을 갖는지 추적하고, 데이터 소스를 최종 버전의 아이템으로 동기화한다. 사용자는 아이템 및 데이터 소스에 대해 선택적으로 크로스-폴리네이션할 수 있거나, 크로스-폴리네이션할 수 없다.
- [0044] 본 발명의 다른 양상에 따르면, 중복 검출 알고리즘은 장치들에 대해서 중복 데이터의 발생을 피하도록 돕는다.
- [0045] 본 발명의 또 다른 양상에 따르면, 아이템의 삭제는 삭제된 아이템과 더이상 사용자의 필터에 있지 않은 아이템을 비교해서 그 차이에 따라 다르게 처리된다.
- [0046] 본 발명의 또 다른 양상에 따르면, 크로스-폴리네이션은, 동기화 소스가 서로 다른 버전의 동기화 프로토콜을 사용하더라도 발생할 수 있다.

발명의 구성 및 작용

- [0047] 명세서 및 청구항에 걸쳐서, 다음의 용어들은, 그 내용이 명확하게 명시되어 있지 않은 한, 본 명세서와 결합됨으로써 그 의미가 분명해지는 것이다. "크로스-폴리네이션"이라는 용어는 다중 동기화 소스들로부터 데이터를 믹싱 및 함께 혼합하는 것을 의미한다.
- [0048] 예시적 동작 환경
- [0049] 도 1을 참조하면, 본 발명을 구현하기 위한 하나의 양호한 시스템은 컴퓨팅 장치(100)와 같은 컴퓨팅 장치를 포함한다. 매우 기본적인 구성에서, 컴퓨팅 장치(100)는 적어도 하나의 프로세싱부(102) 및 시스템 메모리(104)를 일반적으로 포함한다. 컴퓨팅 장치의 정확한 구성 및 유형에 따라서, 시스템 메모리(104)는 (RAM과 같은) 휘발성 메모리, (ROM, 플래시 메모리, 등과 같은) 비휘발성 메모리, 또는 이 둘의 조합일 수 있다. 시스템 메모리(104)는 일반적으로 오퍼레이팅 시스템(105), 하나 이상의 애플리케이션(106)을 포함하고, 프로그램 데이터(107)도 포함할 수 있다. 일 실시예에서, 애플리케이션(106)은 동기화 애플리케이션(120)을 포함할 수 있다. 이러한 기본 구성은 점선 108에 포함된 성분들로서 도 1에 도시된다.
- [0050] 컴퓨팅 장치(100)는 추가적인 특성 및 기능성을 갖는다. 예를 들어, 컴퓨팅 장치(100)는 또한, 예를 들어, 자기 디스크, 광학 디스크, 또는 테이프와 같은 추가의 (삭제가능 및/또는 삭제불가능) 데이터 저장 장치를 포함할 수 있다. 이러한 추가의 저장 장치는 삭제가능 스토리지(109) 및 삭제불가능 스토리지(110)로서 도 1에서 도시된다. 컴퓨터 저장 매체는, 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈, 또는 다른 데이터와 같은 정보를 저장하기 위한 소정의 방법 또는 기술로서 구현되는 삭제가능 및 삭제불가능 매체, 휘발성 및 비휘발성 매체를 포함한다. 시스템 메모리(104), 삭제가능 스토리지(109), 및 삭제불가능 스토리지(110)는 모두 컴퓨터 저장 매체의 예들이다. 컴퓨터 저장 매체는, RAM, ROM, EEPROM, 플래시 메모리, 또는 다른 메모리 기술의 메모리, CD-ROM, 디지털 다기능 디스크(DVD) 또는 다른 광학 스토리지, 즉, 자기 카세트, 자기 테이프, 자기 디스크 스토리지 또는 다른 자기 저장 장치, 또는 희망 정보를 저장하는 데 사용될 수 있으며, 컴퓨팅 장치(100)에 의해 액세스될 수 있는 소정의 다른 매체를 포함하지만, 이것들로 제한되는 것은 아니다. 소정의 이러한 컴퓨터 저장 매체는 장치(100)의 일부를 구성한다. 컴퓨팅 장치(100)는 또한, 키보드, 마우스, 펜, 음성 입력 장치, 터치 입력 장치, 등과 같은 입력 장치(들)(112)를 포함한다. 또한, 디스플레이, 스피커, 프린터, 등과 같은 출력 장치(들)(114)도 포함된다.
- [0051] 컴퓨팅 장치(100)는 또한, 장치가 네트워크 등을 통해서 다른 컴퓨팅 장치(118)와 통신할 수 있도록 하는 통신 접속(116)을 포함한다. 통신 접속(116)은 통신 매체의 일 예이다. 통신 매체에는 일반적으로, 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈, 또는, 반송파 또는 다른 전송 메카니즘 등으로 변조된 데이터 신호의 기타 데이터가 내장되고, 통신 매체는 소정의 정보 전달 매체를 포함한다. "변조된 데이터 신호"라는 용어는 하나 이상의 특성 세트를 포함하거나, 신호의 정보를 암호화하는 방식으로 변경된 신호를 의미한다. 예시적으로, 통신 매체는, 유선 네트워크 또는 직접-배선 접속과 같은 유선 매체, 및 음향, RF, 적외선과 같은 무선 매체, 및 기타 무선 매체를 포함하지만, 이것으로 제한되는 것은 아니다. 본 명세서에서 사용된 컴퓨터 판독가능 매체라는 용어는 저장 매체 및 통신 매체 둘 다를 포함한다.
- [0052] 도 2는 본 발명의 양호한 실시예에 따라 사용될 수 있는 이동 컴퓨팅 장치를 도시한다. 도 2를 참조하면, 본 발명을 구현하기 위한 양호한 일 시스템은 이동 컴퓨팅 장치(200)와 같은 이동 컴퓨팅 장치를 포함한다. 이동 컴퓨팅 장치(200)는 프로세서(260), 메모리(262), 디스플레이(228), 및 키패드(232)를 포함한다. 메모리(262)

는 일반적으로 휘발성 메모리(예를 들어, RAM) 및 비휘발성 메모리(예를 들어, ROM, 플래시 메모리, 등) 둘 다를 포함한다. 이동 컴퓨팅 장치(200)는 마이크로소프트사의 윈도우 CE 오퍼레이팅 시스템과 같은 오퍼레이팅 시스템(264), 또는 메모리(262)에 상주하며, 프로세서(260)에서 실행되는 다른 오퍼레이팅 시스템을 포함한다. 키패드(232)는 (일반적인 전화기와 같은) 푸쉬 버튼 숫자식 다이얼링 패드, (종래의 키보드와 같은) 멀티-키 키보드일 수 있다. 디스플레이(228)는 액정 디스플레이일 수 있고, 또는 이동 컴퓨팅 장치에서 통상 사용되는 다른 타입의 디스플레이일 수 있다. 디스플레이(228)는 접촉-감지식일 수 있고, 이는 입력 장치로서의 역할을 할 수도 있다.

[0053] 하나 이상의 애플리케이션 프로그램(266)은 메모리(262)로 로드되어, 오퍼레이팅 시스템(264)에서 실행된다. 동기화 애플리케이션은 이동 컴퓨팅 장치(200)에 상주하고, 동기화 애플리케이션이 수행되도록 프로그램된다. 동기화 애플리케이션은 장치의 하드웨어 또는 소프트웨어에 상주할 수 있다. 이동 컴퓨팅 장치(200)는 또한 메모리(262)내에 비휘발성 스토리지(268)를 포함한다. 비휘발성 스토리지(268)는, 이동 컴퓨팅 장치(200)가 파워 다운되었을 경우 손실되면 안되는 지속 정보를 저장하는데 사용된다.

[0054] 이동 컴퓨팅 장치(200)는, 하나 이상의 배터리로서 구현되는 전력 공급기(270)를 포함한다. 전력 공급기(270)는 배터리를 공급하거나 충전하는 AC 어댑터 또는 전력공급된 도킹 요람과 같은 외부 전력 소스를 더 포함할 수 있다.

[0055] 이동 컴퓨팅 장치(200)는 2개 타입의 선택적 외부 공지 메카니즘, 즉 LED(240) 및 오디오 인터페이스(274)를 포함하는 것으로 도시된다. 이들 장치는 전력 공급기(270)에 직접 연결되어, 활성화되었을 때, 프로세서(260) 및 다른 구성 요소들이 배터리 전력 유지를 중단하더라도, 공지 메카니즘에 의해 규정된 기간동안 온 상태를 유지한다. 오디오 인터페이스(274)는 사용자에게 청취가능 신호를 제공하고, 사용자로부터 청취가능 신호를 수신하는데 사용된다. 예를 들어, 오디오 인터페이스(274)는, 전화 통화를 용이하게 하기 위해서 청취가능 출력을 제공하기 위한 스피커, 및 청취가능 입력을 수신하기 위한 마이크로폰에 접속될 수 있다.

[0056] 이동 컴퓨팅 장치(200)는 또한, 통신을 전송하고 수신하는 기능을 수행하는 무선 인터페이스층(272)을 포함한다. 무선 인터페이스층(272)은 이동 컴퓨팅 장치(200)와 외부 세계와의 무선 접속을 가능하게 한다. 일 실시예에 따르면, 무선 인터페이스층(272)으로/으로부터의 송신은 오퍼레이팅 시스템(264)의 제어하에서 수행된다. 즉, 무선 인터페이스층(272)에 의해 수신된 통신은 오퍼레이팅 시스템(264)을 통해 애플리케이션 프로그램(266)으로 전달될 수 있고, 그 역도 가능하다.

[0057] 다중 동기화 소스들의 크로스-폴리네이션

[0058] 도 3은 본 발명의 양상에 따른 크로스-폴리네이션 동기화 시스템을 나타낸다. 사용자는 장치 1(305)과 같은 컴퓨팅 장치를 2개 이상의 컴퓨팅 장치들과 동기화할 수 있고, 이 컴퓨팅 장치들은 각각 "동기화 소스"로 고려된다.

[0059] 양호한 크로스-폴리네이션 시나리오에서, 사용자는 2개의 동기화 소스와 동기화되는 단일의 장치를 갖는다. 사용자는 장치를 두 소스와 동기화하고, 크로스-폴리네이션은 소스들이 서로 동기화를 유지하도록 한다. 사용자의 장치는 소스들 사이의 변경을 전달하고, 변경이 다중 소스의 아이템에서 동시에 이루어질 때 충돌을 해결하는데 사용된다. 사용자의 장치는, 각 동기화 소스가 어느 버전의 아이템을 갖는지를 계속 추적하고, 이 소스들 각각을 최종 버전의 아이템으로 동기화한다.

[0060] 크로스-폴리네이션의 예로서, 사용자는 자신의 PDA 장치(305)를 홈 PC(315)와 동기화시키고, 또한 워크 PC(310)와도 동기화시킨다. 일단 동기화되면, 각 장치(PDA 장치(305), 워크 PC(310), 및 홈 PC(315))는 아이템 A(320)를 포함할 것이다. 사용자가 자신의 PDA를 워크 PC와 동기화시키면, 아이템 A는 PDA에 의해 수신된다. 사용자가 PDA를 홈 PC와 동기화시키면, 워크 PC로부터 PDA로 나온 아이템 A는 자동적으로 홈 PC로 보내질 것이다. 또한, 홈 PC에서 생성되는 소정의 아이템은, 크로스-폴리네이션이 가능할 때 PDA를 통해 자동적으로 워크 PC로 보내질 것이다.

[0061] 일 실시예에 따르면, 사용자는 소스들 간의 크로스-폴리네이션을 가능하게 하거나, 불가능하게 한다. 즉, 사용자는 소스들 사이의 크로스-폴리네이션을 할 수 있거나, 아이템을 분리하여 유지할 수 있다. 이 크로스-폴리네이션 설정은 어느때나 온 오프 전환될 수 있다. 예를 들어, 사용자는 홈 PC(315)와 크로스-폴리네이션할 수 있는 능력을 턴 오프할 수 있다. 크로스-폴리네이션이 홈 PC에서 불가능한 경우, 워크 PC로부터의 아이템은 홈 PC로 이동되지 않을 것이다. 또한, 사용자는 하나의 소스를 분리되게 유지할 수 있고, 2개의 다른 소스들을 크로스-폴리네이션할 수 있다. 이러한 유동성은 사용자가 어떻게 데이터를 동기화할 지를 전체적으로 제어할 수

있도록 돕는다.

- [0062] 본 발명의 일 실시예에 따라서, 장치가 단일 소스와 더이상 동기화하지 않으면, 모든 데이터는 장치로부터 제거된다. 데이터는 또한 장치에 남겨질 수도 있다. 장치가 2개 이상의 소스와 동기화하는 경우, 데이터는 다른 소스와 동기하고 있기 때문에 장치에 남겨진다. ID 컬럼은 동기화되지 않은 데이터 소스에 대해 클리어되고, 이 소스에 대한 추적 데이터는 제거된다.
- [0063] 도 4는 본 발명의 양상에 따라서 크로스-폴리네이션의 가능 및 불가능을 도시한다. 크로스-폴리네이션 동기화 장치는 자신이 동기화하고 있는 소스들을 알고 있다. 일 실시예에 따라서, 동기화 소스 구성 정보는 이동 장치에 저장되고, 이 장치는 AirSync 프로토콜을 사용한다. 구성 정보는 다른 위치에도 저장될 수 있고, 장치는 다른 동기화 프로토콜을 사용할 수 있다.
- [0064] 일 실시예에 따라서, 관리자는, 장치가 동기화할 수 있는 소스의 수를 제한하도록 장치에 대한 방침을 설정한다. 이러한 특성은 장치가, 동기화할 수 있는 소스들을 선정된 수로 제한함으로써 크로스-폴리네이션을 차단하는데 사용될 수 있다. 다른 실시예에 따르면, 소정의 인증된 사용자가 장치에 대한 방침을 설정할 수 있다.
- [0065] 예 1은 하나의 데이터 소스(워크 PC(310))와 PDA(305)의 동기화를 나타낸다. 도시된 바와 같이 아이템 A는 두 장치 모두에서 복사된다.
- [0066] 예 2는 제2 데이터 소스(홈 PC(315))의 부가를 도시한다. PDA(305)는 워크 PC 및 홈 PC 둘 다와 동기화되고, 이들 각 장치는 모든 장치의 각 아이템을 포함한다. 원래 워크 PC에 있던 아이템 A는 PDA(305) 및 홈 PC에 복사된다. 원래 홈 PC에 있던 아이템 B는 PDA(305) 및 워크 PC에 복사된다.
- [0067] 예 3은, 장치에 생성된 아이템을 워크 PC에 동기화하며, 워크 PC 및 홈 PC가 크로스-폴리네이트하지 않도록 하는 동기화 설정의 변경을 도시한다. 도시된 바와 같이, 장치에 생성된 아이템 C는 워크 PC와 동기화되지만, 홈 PC와는 크로스-폴리네이션되지 않는다.
- [0068] 예 4는 워크 PC에 동기하는 장치에 생성된 아이템, 및 워크 PC와 크로스-폴리네이트하지 않는 홈 PC에서 새롭게 생성된 아이템을 도시한다. 이 예에서, 아이템 D는 홈 PC에서 생성되고, 장치와 동기화되지만, 워크 PC와는 동기화되지 않는다.
- [0069] 예 5는 모든 장치에서 새롭게 생성된 아이템을 크로스-폴리네이트하는 동기화 설정의 변경을 도시한다. 이 예에서, 아이템 E는 워크 PC에서 생성되고, 장치 및 홈 PC와 동기화된다. 일 실시예에 따르면, 서로 다른 동기화 설정에서 생성된 아이템들은 그들의 동기화 설정을 유지한다. 예를 들어, 도 3 및 도 4에 도시된 서로 다른 동기화 설정 아래서 생성된 아이템 C 및 아이템 D는 크로스-폴리네이트되지 않는다.
- [0070] 예 6은 홈 PC 동기화 소스의 삭제를 도시한다. 일 실시예에 따르면, 제거된 장치에서 생성된 소정의 아이템은, 제거된 장치에만 있는 아이템이라면, 다른 동기화 장치로부터도 제거된다. 이 예에서, 아이템 D는 장치로부터 제거되지만, 아이템 B는 워크 PC에도 동기화되기 때문에 제거되지 않는다.
- [0071] 도 5는 본 발명의 양상에 따라서 양호한 동기화 표 구조를 도시한다. 일 실시예에 따르면, SQL-CE는 동기화 상태를 유지하는데 사용된다.
- [0072] DB 파일(510)은 동기화되는 각 데이터의 타입에 대한 단일 표를 포함한다. 이 예는 컨택, 달력, 및 업무의 동기화를 나타낸다. 아이템에 대한 변경이 다수의 표에서 발생하기 때문에, 아이템과 연관된 최종변경된 필드 (LastModified field;540)는 아이템이 편집될 때마다 갱신된다. 이는 모든 변경들이 공지되고 동기화되도록 한다. 아이템 표, 카테고리 결합 표, 또는 소정의 다른 관련 표에서의 아이템에 대한 변경들은 최종변경된 필드를 갱신하도록 한다. 일 실시예에 따르면, 이 값은 현재의 장치 날짜 및 시간으로 설정된다. 이 플래그는, SQL-CE 변경 추적 시스템이 소정의 아이템 변경을 적절하게 추적하는 것을 보장하도록 하는 아이템에 대한 각각의 갱신에 따라서 일부 수정된다. 도시된 바와 같이, 최종변경된 컬럼은 컨택(524), 달력(520), 및 업무 표(522)에 추가된다. 최종변경된 컬럼은 아이템이 변경될 때마다 갱신된다.
- [0073] 다음은 표에 있는 아이템의 양호한 설명이다.

[0074]	카테고리	내용	값
	데이터소스 타입	alAirSyncServer	1
		alAirSyncDesktop	2

[0075]

데이터 소스 - 각 데이터 소스에 대해 추적하는 데이터		
데이터소스ID	PK, GUID	데이터 소스에 대한 특정 식별자
데이터소스타입	long	어느 타입의 데이터 소스인지를 지시한다. 성능을 추적하는데 사용된다.
친근한이름	nvarchar	스토어를 식별하기 위해 애플리케이션에 의해 사용되는 데이터 소스의 이름
네트워크어드레스	nvarchar	데이터 소스에 대한 어드레스. 데이터 소스 타입에 따라 값이 설정된다.

[0076]

본 발명의 일 실시예에 따르면, 각 데이터 소스에 대한 관습 특성(Sync Custom Props 5560)은 등록부(Registry)에 저장된다. 이러한 관습 특성은 동기화 엔진이 추적하는 상세한 내용을 포함한다. 일 실시예에 따르면, 서버는 사용자이름, 패스워드, 도메인, 및 룰(Rule)을 계속 추적한다. 데스크탑 소스는 사용자이름, 패스워드(선택적), 도메인, 룰, 및 기기명을 계속 추적한다.

[0077]

다음은, 본 발명의 양상에 따른 양호한 데이터 소스 애플리케이션 프로그램 인터페이스의 리스트이다.

[0078]

데이터 소스 APIS	
기능성	설명
<p>데이터 소스 생성</p> <p>HRESULT CreateDataSource([in]UNIT cProps, [in]CEPROPVALS*rgVals, [out]GUID*pDataSourceID);</p>	<p>파라미터; DataSourceType FriendlyName NetworkAddress ExchangeServer</p> <p>새로운 데이터 소스가 생성되고, 성공적이라면, 새로운 데이터소스ID가 OUT 파라미터로 되돌려진다.</p> <p>통과된 친근한 이름은 유일한 것이다. 요구된 FriendlyName이 이미 존재하면, 예러가 호출 함수로 반환된다.</p>
<p>데이터 소스 편집</p> <p>HRESULT GetDataSourceProps([in]GUID DataSourceID, [in]UNIT cProps, [in]CEPROPVALS*rgVals);</p>	<p>어느 데이터 소스가 갱신될지를 나타내는 DataSourceID에서 통과. 생성 동작에서의 모든 특성 세트가 편집될 수 있다.</p>
<p>HRESULT SetDataSourceProps([in]GUID DataSourceID, [in]UNIT cProps, [in]CEPROPVALS*rgVals);</p>	<p>이들 필드 중 임의의 필드를 편집하는 것은 동기화 키의 리셋을 요구하지 않는다.</p>
<p>데이터 소스 삭제</p> <p>HRESULT DeleteDataSource([in]GUID DataSourceID);</p>	<p>요구된 데이터 소스를 삭제하고, 표로부터 이 데이터 소스에 대한 컬럼을 삭제한다.</p> <p>아이템이 동기화하지만, 이제 더이상 동기화 소스에 접속되지 않으면, 표로부터 아이템을 삭제한다.</p>
<p>데이터 소스 획득</p> <p>HRESULT EnumDataSourceFirst([out]HANDLE*phEnum, [out]GUID*pDataSourceID);</p> <p>HRESULT EnumDataSourceNext([in]HANDLE hEnum, [out]GUID*pDataSourceID);</p>	<p>데이터 소스의 리스트를 통해 검색 및 계산</p>

[0079]

아이템에 대한 셋팅으로 동기화 설정/검색 <pre>struct SyncDataSource{ GUID DataSourceID; BOOL fSyncing }</pre>	
<pre>HRESULT GetDataSourceForItem([in]GUID itemID, [in]GUID itemType,[out] SyncDataSource**ppSDS); HRESULT SetDataSourceForItem([in]GUID itemID, [in]GUID itemType,[in] SyncDataSource*ppSDS);</pre>	<p>아이템에 대한 데이터 소스 획득</p> <p>새로운 아이템에 대해 이 API는, 동기화될 수 있는 아이템 타입의 데이터 소스 및 디폴트 값의 리스트 (어느 데이터 소스가 디폴트에 의해 동기화되어야 하는 새로운 아이템을 갖는지)를 결정하도록 호출된다.</p> <p>기존 아이템에 대해 이 API는 유효 데이터의 리스트를 결정하도록 호출된다.</p> <p>아이템이 동기화될 수 있는 소스, 및 (아이템이 현재 동기화하고 있는) 현재 동기화 값</p> <p>특정 아이템ID에 대해 동기화할(Sync To) 값을 설정하도록 이 API를 사용한다.</p> <p>아이템에 대한 데이터 소스 설정</p> <p>이 함수는 여러 상황들에 대응한다.</p> <p>1: 새로운 아이템 - 관련되는(pending) 플래그를 설정</p> <p>2: 동일한 값으로 편집 - 아무것도 하지 않음</p> <p>3: 새로운 값으로 편집 - 관련된 삭제 또는 관련되는 적합 장소에 대해 표시</p>

[0080]

하드 및 소프트 삭제 사이의 차이점

[0081]

도 6은 본 발명의 양상에 따라 소프트 삭제 및 하드 삭제를 도시한다. 일 실시예에 따르면, 동기화 프로토콜은, 아이템이 필터 밖으로 나가거나(즉, 더이상 동기화하지 않으면), 또는 물리적으로 제거되면, 아이템에 대한 삭제를 아래로 보낸다. 일 실시예에 따르면, 크로스-폴리네이팅하는 아이템에 대한 필터 설정은 장치 범위에서의(device-wide) 설정이다. 크로스-폴리네이팅하는 소스들에 대한 삭제가 필터 외부로 전달됨으로써 데이터의 손실이 가능하기 때문에, Hard Delete 명령은 소프트 삭제와 하드 삭제를 차별화하도록 하는데 사용된다.

[0082]

"Hard Delete"라는 용어는 각 스토어로부터 아이템을 물리적으로 삭제하는 것을 의미한다. 하드 삭제는 모든 크로스-폴리네이팅하는 소스들에 대해 전파된다. "Soft Delete"는 아이템을 필터밖으로 내보냈다는 것을 의미한다. 소프트 삭제는 장치로부터 아이템을 삭제하지만, 이 삭제가 어느 다른 소스로 전파되는 것은 아니다.

[0083]

동기화 프로토콜의 버전이 소프트 삭제를 지원하지 않는다면, 삭제 명령이 수신될 때, 장치는 그 스토어로부터만 아이템을 삭제할 것이다. 즉, 이 삭제는 크로스-폴리네이션되지 않는다. 아이템이 최종 동기화후에 필터에서 완료될 수 있기 때문에, Soft Delete는 Sync 세션의 끝에서 발생한다.

[0084]

예를 들어, 아이템(610)은 서버로부터 소프트 삭제되고, 그 다음, 아이템이 장치로부터 삭제될 것이지만, 삭제(620)로 도시된 바와 같이, 홈 PC에서는 삭제되지 않는다. 반대의 경우라면, 아이템은 워크 PC에서 하드 삭제되어, 삭제(625)로 도시된 바와 같이, 아이템(610)은 장치 및 홈 PC로부터 삭제된다. 삭제 명령은, 장치가 존재하지 않는 아이템에 대한 삭제 명령을 수신하는 경우 무시된다.

[0085]

중복 검출

[0086]

도 7은 본 발명의 양상에 따라서 장치에 아이템을 추가하라는 명령의 처리를 도시한다. 아이템이 장치에 추가되기 전에, 중복 검출 검사가 완료된다. 중복 검출 검사를 수행하는 것은, 크로스-폴리네이팅되고 있는 서로 다른 장치에서 중복이 발생하지 않도록 보장하도록 한다. 일 실시예에 따르면, 중복 검출은 아이템 특성 비교에 기초하고, 중복 검출 코드는 (도 3에 도시되어 있는 장치(305)와 같은) 장치에서 구현된다.

[0087]

블록 710에서, 장치는 동기화되고 있는 데이터 소스로부터 추가 명령을 수신한다. 블록 715로 진행하여, 장치는 추가될 아이템에 대한 해시값을 계산하고, 동일한 해시값을 갖는 소정의 다른 아이템에 대해서 스토어를 검

색한다. 해시값은 아이템을 유일한 것으로서 식별하는데 사용된다. 아이템을 유일한 것으로서 식별하는 다른 방법들도 사용될 수 있다. 판단 블록 720으로 진행하여, 장치가 크로스-폴리네이팅되는지 여부가 결정된다.

[0088] 해시가 판단 블록 725에 위치하면, 장치는 아이템이 중복된 것으로 고려하여, 판단 블록 735로 이동한다. 동기화 해시가 발견되지 않으면, 프로세스는 블록 730으로 진행하고, 여기에서 아이템은 장치에 부가되고, 동기화 팬딩(pending) 비트가 설정된다. 판단 블록 735에서, 발견된 아이템이 현재 이 데이터 소스와 동기화된다면, 아이템이 생성되고, 동기화 팬딩 필드가 블록 730에서 설정된다. 일 실시예에 따르면, 동기화 해시와 매칭되는 장치상의 아이템이 이미 소스와 동기화된다면, 장치는 어느것도 중복되는 것으로 고려하지 않을 것이다.

[0089] 아이템이 데이터 소스와 현재 동기화하지 않으면, 프로세스는 블록 740으로 진행하고, 여기에서 아이템은 중복된 것으로 검출된다. 이 시점에서, 장치는 이미 가졌던 버전을 유지하며, 새로운 아이템에 대한 ID를 기존의 기록에 단순히 부가할 것이다.

[0090] 도 8은 본 발명의 양상에 따라서 양호한 중복 검출 시나리오를 도시한다. 일반적으로, 아이템이 장치에 부가되면, SyncHash(810)는 아이템 특성의 서브셋에서 계산된다. 아이템 A는 HASH101(815)의 동기화 해시값을 갖는다. 장치가 부가될 아이템을 수신하면, 장치는 새로운 아이템의 SyncHash와 기존 아이템을 비교하여, 장치가 중복을 검출하도록 한다.

[0091] 단계 1로 돌아가서, 아이템 A는 서버 1로부터 장치에 이미 동기화되었다. 이 시점에서, 아이템 A는 서버 1에 있고, 장치는 HASH101의 SyncHash 값을 갖는다. 단계 2에서, 사용자는 자신의 장치로부터 서버 1로 아이템 A를 복사하여, 장치의 서버 1에서 아이템 B를 생성한다. 아이템 B는 아이템 A와 동일한 SyncHash 값을 갖는다. 장치가 단계 3에서 서버와 동기화될 때, 아이템 B는 이미 서버와 동기화하기 때문에, 장치에서 생성될 것이다.

[0092] 해시 계산

[0093] 2개의 특성 레벨 매칭이 SyncHash를 계산하는데 사용된다.

[0094] 정확한 데이터 비교(주(primary) 키셋): 필드 세트는 아이템이 중복된 것으로 고려하도록 비교되는 주 특성으로서 정의된다. 이 필드 세트는 각 데이터 타입에 대해 다를 것이다. 선택된 필드 세트는 주요 코드 변경없이 코드에 적용될 수 있어야 한다. 이는 헤더 파일의 필드 리스팅을 포함할 수 있고, 이에 따라 알고리즘은 테스트 프로세스 동안에 조정될 수 있다.

[0095] 특성 존재(부(secondary) 키셋): 거의 완전하게 포함할 수 있는 더 큰 필드 세트는 이러한 특성들의 데이터가 단순히 존재하는 지에 대한 검사에 사용된다. 이는, 필드가 일 아이템에 설정되어 있고, 또 다른 아이템에는 비어있다면, 주 키가 매칭되더라도, 아이템들이 서로 중복된 것으로 고려되지 않을 것이라는 것을 보장한다.

[0096] 해시 저장/갱신

[0097] 아이템이 처음으로 동기화되면, SyncHash는 계산되어 아이템에 저장된다. SyncHash 값은 이미 동기화된 아이템이 편집될 때 갱신된다. 다음은 본 발명의 일 실시예에 따른 컨택, 달력, 및 업무에 대한 양호한 주 및 부 키셋이다. 다른 특성들은 동기화될 다른 아이템들에 대한 주 및 부 키셋을 생성하는데 사용된다.

컨택 동기화 해시 알고리즘	특성
주 키셋 (정확한 데이터 비교)	이름(FirstName) 성(LastName) 집전화번호의 최종 4개 숫자 회사전화번호의 최종 4개 숫자 (4 디짓보다 작다면, 소정의 존재하는 것 사용)

부 키셋 (특성 존재)	기념일, 점원이름, 점원전화번호, 생일, 바디, 회사2전화번호, 회사도시, 회사국가, 회사우편코드, 회사스테이트, 회사스트리트, 회사팩스번호, 차량전화번호, 회사명, 부서, 이메일주소, 이메일2주소, 이메일3주소, 파일에즈(FileAs), 집2전화번호, 홈도시, 홈국가, 홈우편코드, 홈스테이트, 홈스트리트, 홈팩스번호, 직업명, 중간이름, 이동전화번호, 회사위치, 다른도시, 다른국가, 다른우편코드, 다른스테이트, 다른스트리트, 무선호출기번호, 무 선전화번호, 배우자, 부가물, 타이틀, 웹페이지, 요미회사명(YomicompanyName), 요미이름, 요미성(YomiLastName)
-----------------	--

[0099]

달력 동기화 해시 알고리즘	특성
주 키셋 (정확한 데이터 비교)	UID, 개시시간, 위치, 서브젝트
부 키셋 (특성 존재)	하루이벤트(AllDayEvent), 이메일, 이름, DtStamp, 종료시간, 삭제됨, 예외개시시간, 회의스테이터스, 주최자이메일, 주최자이름, 반복(Recurrence), 타입, 기한(Until), 발생(Occurrences), 간격, 주간날짜(DayOfWeek), 월간날짜(DayOfMonth), 월간주(WeekOfMonth), 년간달((MonthOfYear)), 리마인더

[0100]

업무 동기화 해시 알고리즘	특성
주 키셋 (정확한 데이터 비교)	서브젝트, 기한일, 카테고리
부 키셋 (특성 존재)	개시날짜, 완료날짜, 중요성, 완료(IsComplete), 반복(IsRecurring), 민감성, 리마인더설정, 리마 인더시간, 바디

[0101]

아이템이 크로스-폴리네이션에 대해 표시되면, 장치에 사용가능한 데이터는 다른 데이터 소스들과 크로스-폴리 네이트될 것이다. 아이템의 단절된 버전들은 장치 제한때문에 크로스-폴리네이트할 것이다. 일 실시예에 따르 면, 텍스트[생략된 메시지]는, 단절이 발생할 때, 주의 필드의 말단에 추가된다. 따라서, 사용자는 완전한 오 리지널 사본을 갖지 않았다는 것을 알게 된다.

[0102]

장치의 동기화 엔진은 먼저 데이터 소스로부터의 모든 부가들(Adds)을 요구한다. 다음으로, 데이터 소스로 아 직 남아 있는 모든 부가들을 보낸다(중복된 부가들은 표시되고, 이에 따라 발행되지 않는다).

[0103]

일 실시예에 따라서, 장치의 아이템은 3개의 특성을 추적한다. 서버ID 값 및 SyncHash는 각 아이템에 관습 특 성으로서 저장될 것이다.

[0104]

특성	설명
아이템ID	이는 아이템을 식별하는데 사용된다.
서버ID1...N	이는 서버/데스크탑에서 아이템을 식별하는 ID이 다. 아이템은 아이템에 관습 특성으로서 저장된 다중 서버 ID들을 가질 수 있다.
SyncHash	이는 중복을 검출하는데 사용되는 동기화 해시값 이다. 이 값은 각 변경에 따라 계산 및 재계산된 다.

[0105] 서버ID 특성은 다음 4개의 상태를 갖는다:

동기화 상태	설명
동기화 펜딩	아이템은 장치에 있지만, 목표 데이터 소스로 아직 동기화되지 않는다. 아이템은 다음 동기화에 따라서 데이터 소스에서 생성된다.
유효 서버ID	아이템 서버ID는 데이터 소스와의 적절한 매핑을 가능하게 하도록 저장된다.
펜딩 삭제	아이템은 이 데이터 소스에 대해 과거에 동기화되었지만, 사용자는 이 아이템이 더이상 동기화되지 않을 것을 요구한다. 다음 동기화에서, 데이터 소스로부터 이 아이템을 삭제하여, 그 특성을 클리어한다.
비동기화	아이템은 이 데이터 소스와 동기화하지 않는다.

[0107] 도 9는 본 발명의 양상에 따라서 장치에 있는 기존의 아이템의 중복에 대한 부가 명령의 처리를 도시한다. 중복된 것으로 고려되는 아이템에 대해서, 프로세스는, 부가 명령이 변경 명령으로 전환되는 블록 910으로 진행된다. 전체 아이템을 장치로 복사하는 대신에, 아이템이 갱신된다. 블록 920에서, 특성 리스트는 공지된 도표와 비교된다. 블록 930에 따르면, 확실한 삭제는 제공되지 않은 공지된 특성들에 대한 변경 명령에 부가된다. 블록 940은 갱신된 명령을 도시한다. 블록 950에서, 변경 명령이 처리된다.

[0108] 서로 다른 동기화 버전들에 대한 크로스-폴리네이션

[0109] 특성 레벨 변경 추적은, 다중의 동기화 프로토콜 버전의 크로스-폴리네이팅을 지원하는 것을 돕고, 불필요한 기입을 최소화하기 위해 구현된다. 각 동기화 제공자는 동기화하는 특성 세트를 알아야 하고, 제공자가 관심을 갖는 특성에 변경이 발생하는 경우 동기화를 개시해야만 한다. 동기화 제공자는 자신이 알고 있는 특성들만을 변경하기 때문에, 장치의 아이템이, 병합된 특성 세트를 포함하는 것이 가능하고, 여기에서 일부 특성들은 v1 서버로부터 나온 것이며, 다른 특성들은 v3 서버만이 동기화하는 것이다.

[0110] 도 10은 본 발명의 양상에 따라서 v1 및 v3 서버의 크로스-폴리네이션을 도시한다. 서로 다른 버전의 동기화 프로토콜은 서로 다른 특성을 지원한다. 도시된 예에서와 같이, V1 프로토콜은 특성 A 및 B를 지원하지만, V3 프로토콜은 특성 A, B, 및 Z를 지원한다.

[0111] 처음에, 장치는 V1 서버 또는 V3 서버와 동기화되지 않는다. 901에서, 장치는 V3 서버와 동기화한다. 아이템 A 및 B는 중복된 것으로 검출되고, 아이템 A, B, 및 Z는 장치에 동기화된다. 930에서, 장치가 V1 서버와 다시 동기화할 때, 아이템 Z만이 변경되고, 이는 V1 서버에 의해 지원되지 않기 때문에 동기화는 발생하지 않는다.

[0112] 도 11은 본 발명의 양상에 따라서 양호한 동기화 시나리오를 나타낸다. 단계 1110은, 아이템 A가 홈 PC 및 워크 서버와 사전에 크로스-폴리네이트된 것을 나타낸다. 1120에서, 아이템 A는 아이템 A의 편집된 생성 버전 2(v2)이다. 장치는 1130에서 워크 서버와 동기화하고, 여기에서 버전 2는 장치에 동기화된다. 다음으로, 블록 1140에서, 장치는 홈 PC와 동기화한다. 장치가 서버 윈(win)으로 설정되기 때문에, 아이템 A(버전 1)는 이제 장치에 있게 된다. 장치가 워크 서버와 다시 동기화할 때, 여기에 충돌은 없고, 아이템 A - 버전 1은 워크 서버(1150)에 있게 된다.

[0113] 도 12는 본 발명의 양상에 따라서 양호한 동기화 시나리오를 도시한다. 단계 1210은 아이템 A가 홈 PC 및 워크 서버와 사전에 크로스-폴리네이트된 것을 나타낸다. 1220에서, 동기화 키가 워크 서버에서 0의 값을 갖는다고 식별된다. 사용자는 워크 서버 ID가 블록 1230에서 삭제되도록(1224) 크로스-폴리네이팅한다. 블록 1240으로 진행하여, 동기화는 스크래치(scratch)로부터 시작되고, 장치는 아이템 A를 장치에 부가하라고 요구하는 부가 명령을 수신한다. 아이템은 중복된 것으로 검출되고, 홈 PC로부터의 아이템은 유지된다. 워크 서버 ID는 아이템 태그에 부가된다.

[0114] 데이터 소스 생성

[0115] 도 13은 본 발명의 양상에 따라서 데이터 소스 생성을 도시한다. 개시 블록 후에, 프로세스는 블록 1310으로 진행하고, 여기에서 새로운 데이터 소스에 대한 계수 정보가 획득된다. 블록 1320에 따라서, 새로운 데이터 소스를 시스템에 등록하라는 요청이 이루어진다. 블록 1330에서, 데이터 소스 타입은 데이터 소스와 결합된 타입

으로 설정된다. 일 실시예에 따르면, 데이터 소스 타입은 장치의 타입, 즉, 데스크탑동기화 및 서버동기화에 관련된다. 다음으로, 블록 1340에서, 기기명이 기록된다. 이는, 데스크탑 동기화 정보가 손실된다면, 장치가 데스크탑(대부분의 경우)에 재접속할 수 있도록 한다. 다음에 프로세스는 종료 블록으로 진행하여, 다른 동작들을 처리하도록 돌아간다.

[0116] 데이터 소스 제거

[0117] 도 14는 본 발명의 양상에 따라서 데이터 소스의 제거 프로세스를 도시한다. 데이터 소스의 삭제는 데이터 소스 및 데이터 소스가 동기화하는 모든 아이템을 삭제한다. 아이템이 2개 이상의 데이터 소스와 동기화하거나, 동기화하고 있는 중이면, 아이템은 삭제되지 않을 것이다.

[0118] 개시 블록 후에, 프로세스는 블록 1410으로 진행하여, 얼마나 많은 소스들이 장치와 동기화하고 있는지를 판단한다. 판단 블록 1420으로 진행하여, 여기에 데이터 소스가 단지 하나 있다면, 프로세스는 블록 1430으로 진행하고, 여기에서 삭제된 데이터 소스와 동기화되는 아이템은 장치로부터 제거된다. 2개 이상의 장치가 있다면, 프로세스는 블록 1440으로 진행하고, 여기에서 장치의 크로스-폴리네이션되지 않은 아이템은 제거된다. 블록 1450으로 이동하여, 삭제된 소스의 ID가 장치로부터 제거된다. 일 실시예에 따르면, 이 데이터 소스에 대한 관습 컬럼은 애플리케이션 표로부터 제거된다. 일 실시예에 따르면, 소정 데이터를 제거하기 전에 경고 메시지가 사용자에게 제공된다.

[0119] 다음은 보다 구체화하는 예이다. 장치가 그의 컨택 및 달력을 제1 소스로서 동기화하고, 장치가 달력 및 업무를 제2 소스로서 동기화한다고 가정한다. 그 다음, 사용자는 제1 소스를 삭제한다. 제1 소스를 삭제한 결과로서, 모든 컨택들은 장치로부터 제거된다. 달력 아이템 및 업무는 장치에 남아 있고, 제1 소스에 대한 ID들은 제거된다.

[0120] 동기화되지 않은 데이터 소스 정보 조정

[0121] 도 15는 본 발명의 양상에 따라서 동기화되지 않은 데이터 소스 정보의 조정을 도시한다.

[0122] 다음 예들은 설명하는데 사용될 것이다. 데이터 소스(워크 PC)는, 워크 PC가 더이상 알지 못하는 장치에 존재한다. 이러한 경우는 다수의 이유로서 발생할 수 있다. 예를 들어, 사용자가 동기화 프로그램을 재설치하거나, 또는 기기를 재구축하는 경우 발생할 수 있다. 장치가 데스크탑에 접속될 때, 데스크탑은 장치가 동기화하는 데이터 소스 셋업을 이미 갖는 것을 감지한다. 이는 장치의 데이터 소스의 네트워크 기기명과 워크 PC의 명칭을 비교함으로써 결정된다. 기존의 데스크탑 소스는 삭제되고, 새로운 데스크탑 소스가 생성된다.

[0123] 데이터 소스는 장치가 더이상 알지 못하는 데스크탑에 존재한다. 이 경우는 예를 들어, 장치가 콜드 부트(cold boot)할 때 발생할 수 있다. 데이터 소스에 대해 데스크탑에 저장된 장치ID와 장치의 장치ID를 비교함으로써, 장치가 데스크탑과 동기화하도록 사용되는지 결정될 수 있고, 이는 장치가 데스크탑과 동기화하도록 다시 설정되어야 하는지 여부에 따라서 결정된다. 사용자가 '예'라고 답하면, 사용자에게 대해 새로운 데이터 소스가 생성된다.

[0124] 상술한 설명, 예, 및 데이터는 본 발명의 제조 및 구성의 사용에 대해 전체적인 설명을 제공한다. 본 발명의 다수의 실시예는 본 발명의 사상 및 범위를 벗어나지 않는 한 이루어질 수 있으며, 본 발명은 다음에 첨부된 청구항에 따른다.

발명의 효과

[0125] 다중 동기 소스들을 동기화하는 방법을 제공함으로써, 2개 이상의 서로 다른 장치들 사이의 정보를 동기화할 수 있도록 한다.

도면의 간단한 설명

[0001] 도 1 및 2는 본 발명의 양호한 일 실시예에서 사용될 수 있는 양호한 컴퓨팅 장치를 도시하는 도면.

[0002] 도 3은 크로스-폴리네이션 동기화 시스템을 도시하는 도면.

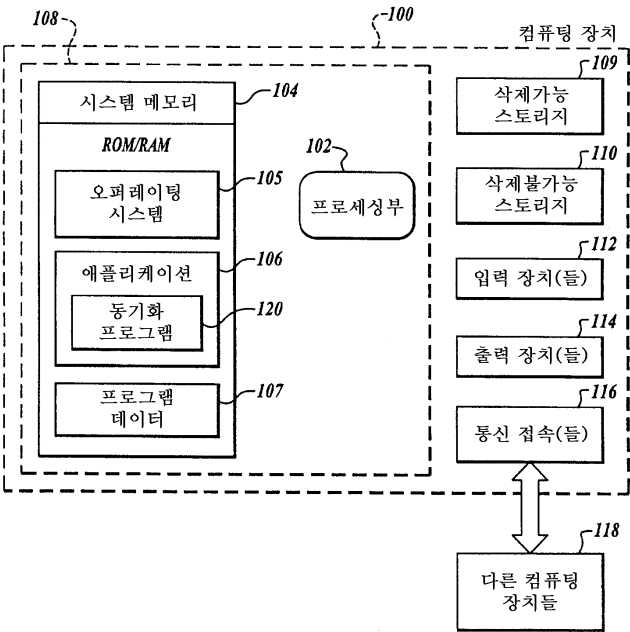
[0003] 도 4는 크로스-폴리네이션의 가능 및 불가능을 도시하는 도면.

[0004] 도 5는 양호한 동기화표 구조를 도시하는 도면.

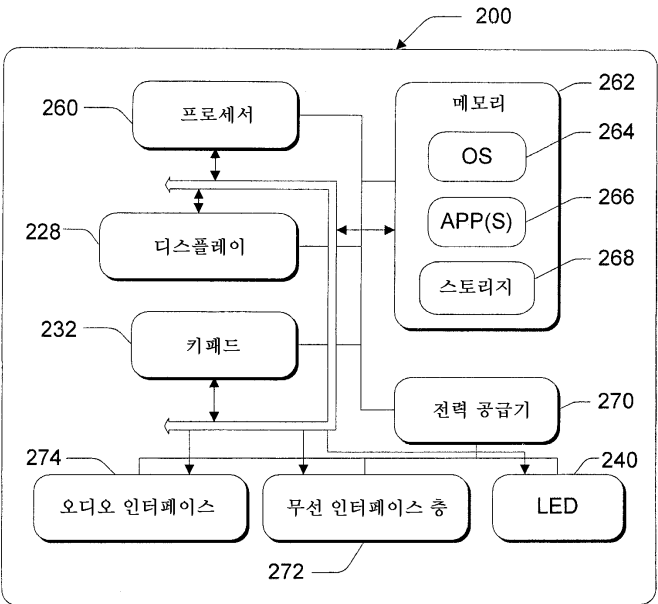
[0005]	도 6은 소프트 삭제 및 하드 삭제를 도시하는 도면.
[0006]	도 7은 아이템을 장치에 부가하라는 명령의 처리를 도시하는 도면.
[0007]	도 8은 양호한 중복 검출 시나리오를 도시하는 도면.
[0008]	도 9는 장치에 있는 중복된 기존 아이템에 대한 부가 명령의 처리를 도시하는 도면.
[0009]	도 10은 v1 및 v3 서버의 크로스-폴리네이션을 도시하는 도면.
[0010]	도 11 및 12는 양호한 동기화 시나리오를 나타내는 도면.
[0011]	도 13은 데이터 소스 생성을 도시하는 도면.
[0012]	도 14는 데이터 소스를 삭제하는 프로세스를 나타내는 도면.
[0013]	도 15는 본 발명의 양상에 따라서, 동기화되지 않은 데이터 소스 정보의 조정을 도시하는 도면.
[0014]	<도면의 주요 부분에 대한 부호의 설명>
[0015]	100 : 컴퓨팅 장치
[0016]	102 : 프로세싱부
[0017]	104 : 시스템 메모리
[0018]	105 : 오퍼레이팅 시스템
[0019]	106 : 애플리케이션
[0020]	107 : 프로그램 데이터
[0021]	109 : 삭제가능 스토리지
[0022]	110 : 삭제불가능 스토리지
[0023]	112 : 입력 장치
[0024]	114 : 출력 장치
[0025]	116 : 통신 접속
[0026]	118 : 다른 컴퓨팅 장치
[0027]	120 : 피닝 애플리케이션
[0028]	260 : 프로세서
[0029]	262 : 메모리
[0030]	228 : 디스플레이
[0031]	232 : 키패드
[0032]	305 : PDA 장치
[0033]	310 : 워크 PC
[0034]	315 : 홈 PC
[0035]	320 : 아이템 A
[0036]	510 : DB 파일
[0037]	520 : 달력
[0038]	522 : 업무 표
[0039]	524 : 컨택

도면

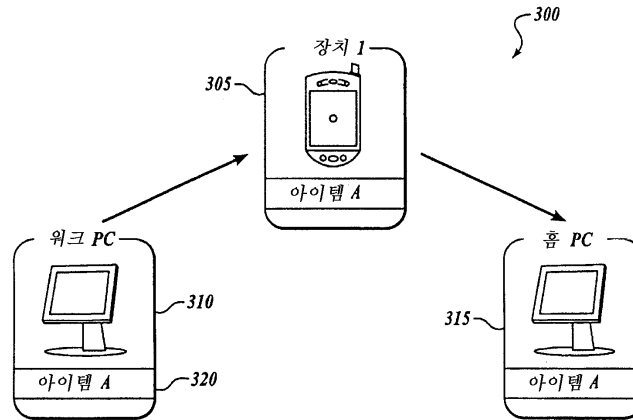
도면1



도면2

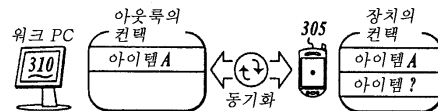


도면3

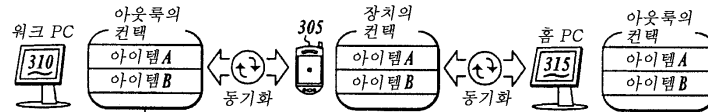


도면4

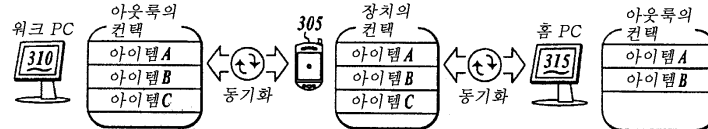
1. 하나의 데이터 소스 동기화



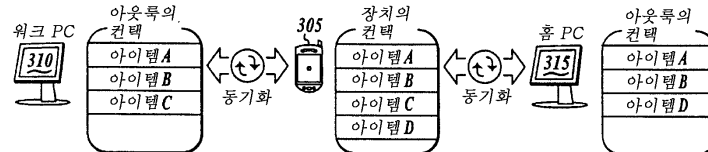
2. 제2 데이터 소스 부가. 사용자는 데이터 소스 생성 시에 크로스-폴리네이트할지 선택



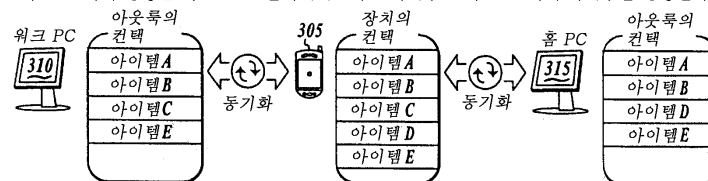
3. 액티브동기화 설정 변경: 장치에 생성된 아이템은, 크로스-폴리네이트하지 않는 워크 PC 및 홈 PC로부터 워크 PC 아이템과 동기화한다



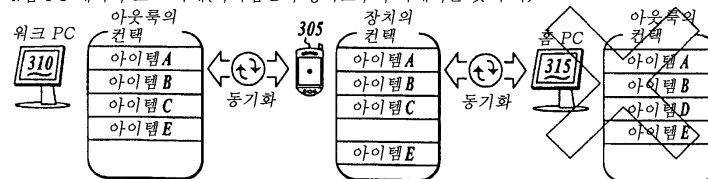
4. 액티브동기화 설정 변경: 장치에 생성된 아이템은, 크로스-폴리네이트하지 않는 워크 PC 및 홈 PC로부터 워크 PC 아이템과 동기화한다. 홈 PC에 아이템 생성



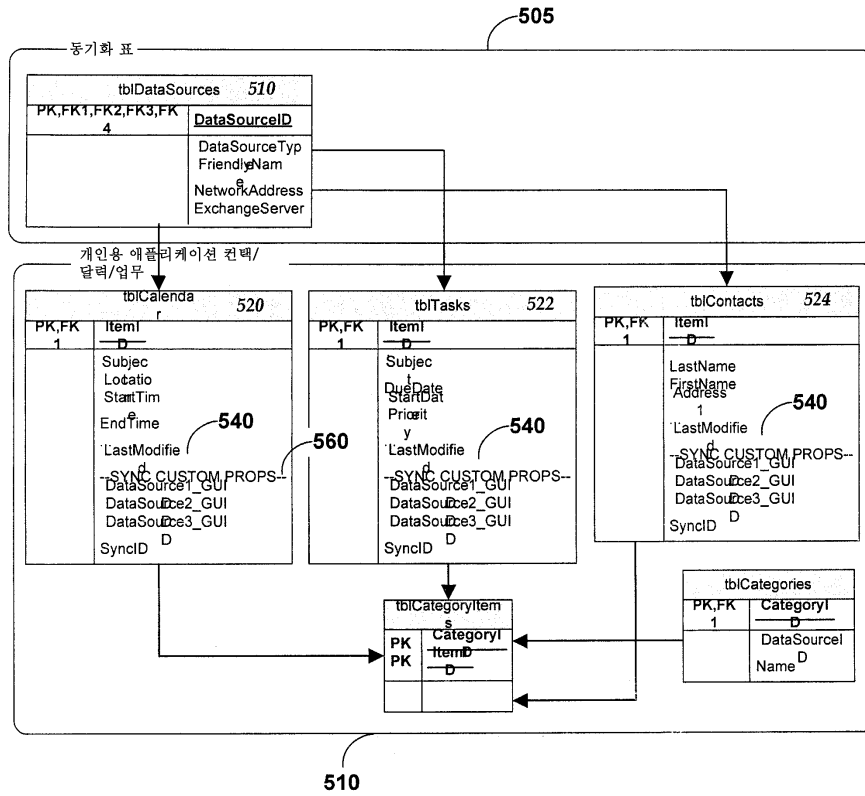
5. 워크 PC 및 홈 PC와 동기화하는 장치에 생성된 아이템의 액티브동기화 설정 변경. 홈 또는 워크 PC에서 생성될 때 크로스-폴리네이트하는 아이템은 워크 PC에서 아이템을 생성한다.



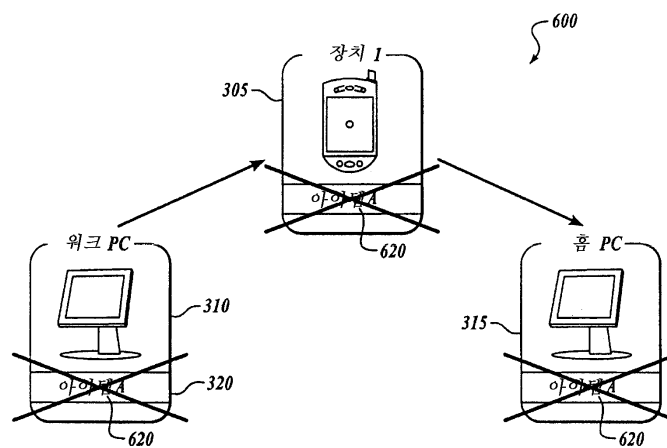
6. 홈 PC 데이터 소스 삭제(아이템 D가 장치로부터 삭제되는 것 주의)



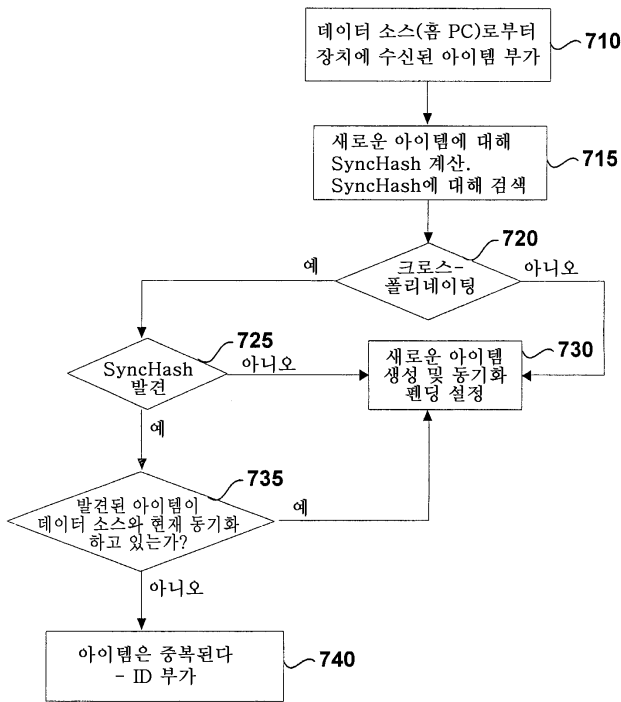
도면5



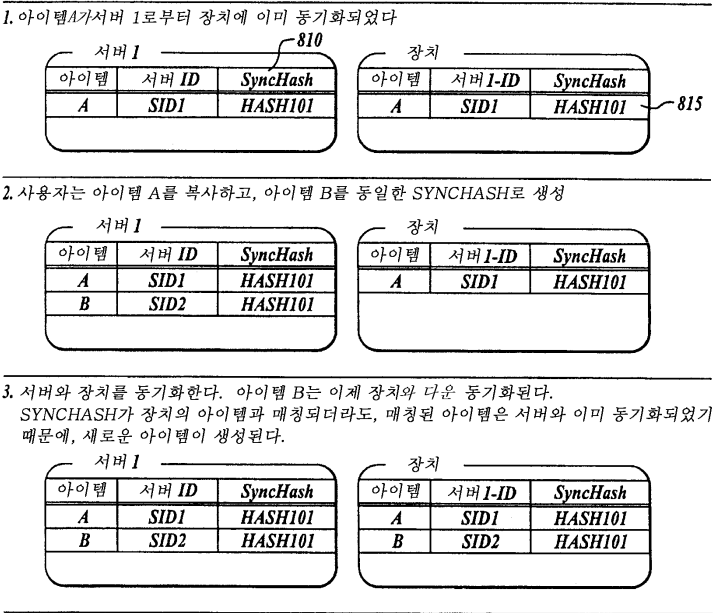
도면6



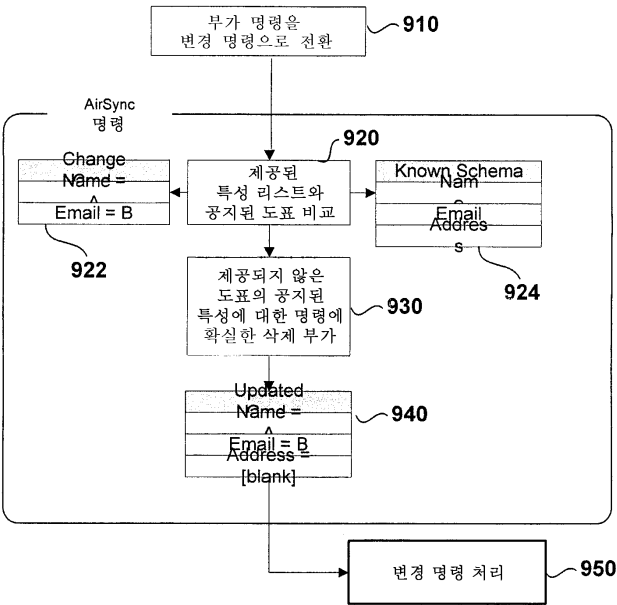
도면7



도면8

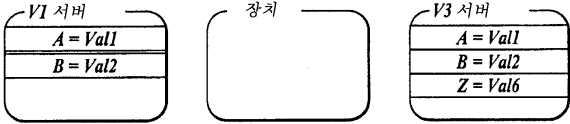


도면9

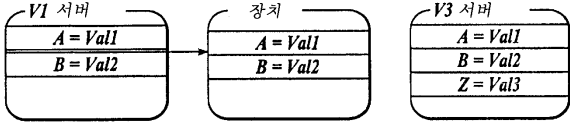


도면10

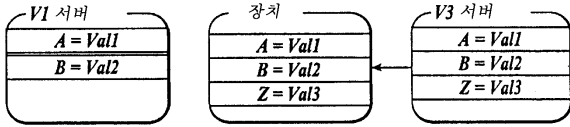
1. V1 프로토콜은 특성 A 및 B를 지원. V3 프로토콜은 특성 A, B, 및 Z를 지원.



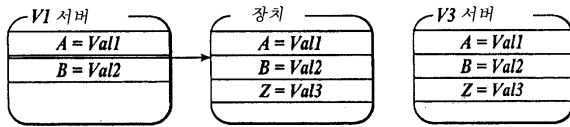
2. V1 서버 특성 A 및 B와 제1 동기화



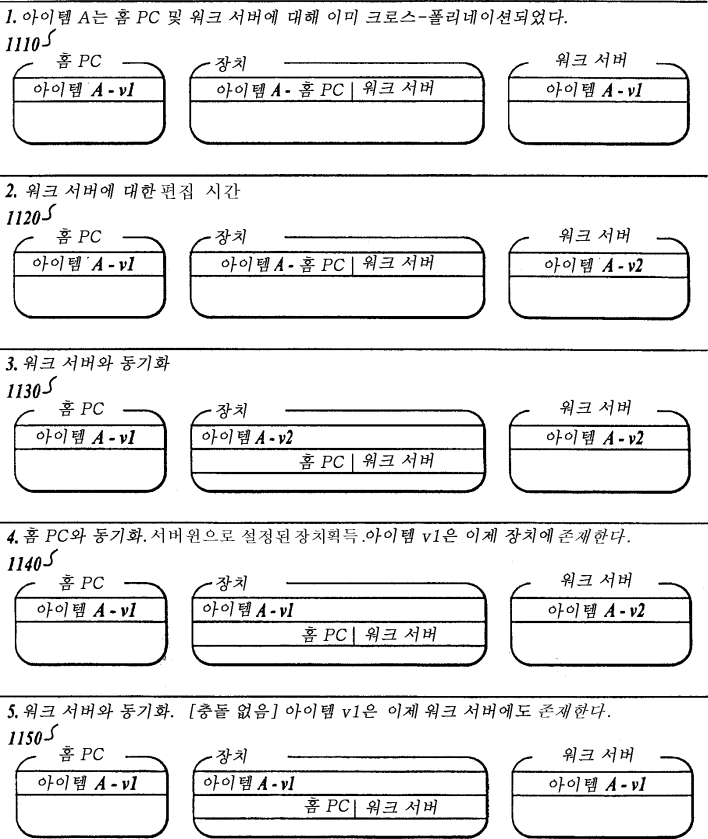
3. V3 서버와 동기화. 아이템은 특성 A 및 B의 비교 후에 중복된 것으로 검출됨
특성 A, B, Z를 다운 동기화



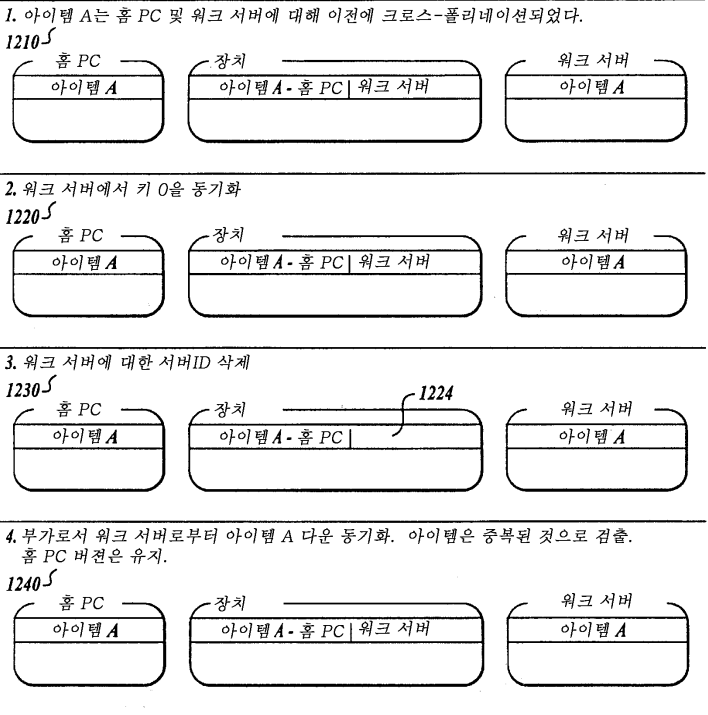
4. 장치에서 추적하는 SQL-CE 변경은 특성 Z에 대한 변경만을 기록한다.
장치가 다음에 V1과 동기화하면, 장치는 아이টে에 대한 변경은 감지하지만, 동기화하는 다른 특성들에 대해서는 변경을 감지하지 않는다(Z만이 변경됨).
추가 동기화는 발생하지 않는다.



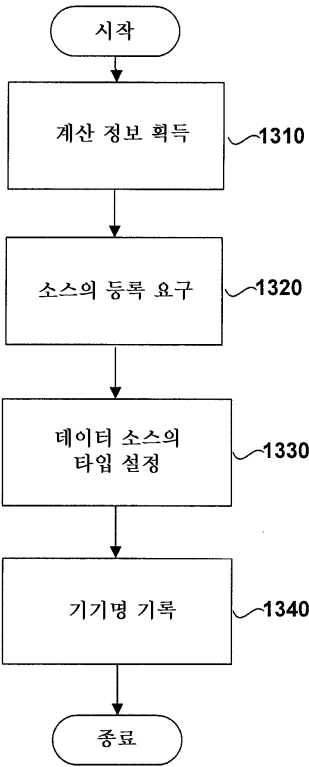
도면11



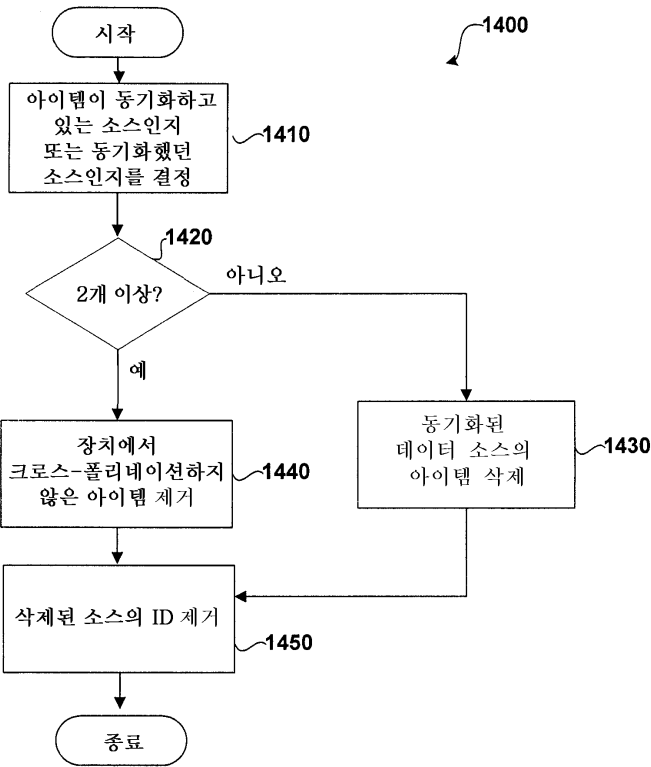
도면12



도면13



도면14



도면15

