



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년12월27일
(11) 등록번호 10-1099202
(24) 등록일자 2011년12월20일

(51) Int. Cl.

G06F 17/30 (2006.01)

(21) 출원번호 10-2004-0057574

(22) 출원일자 2004년07월23일

심사청구일자 2009년07월20일

(65) 공개번호 10-2005-0013932

(43) 공개일자 2005년02월05일

(30) 우선권주장

10/629,254 2003년07월29일 미국(US)

(56) 선행기술조사문헌

US06295541 B1

(73) 특허권자

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

우윤신

미국 98033 워싱턴주 커크랜드 노쓰이스트 65 스
트리트 12522

노비크레브

미국 98006 워싱턴주 벨레뷰 사우쓰이스트 45번
스트리트 14116

(뒷면에 계속)

(74) 대리인

제일특허법인

전체 청구항 수 : 총 12 항

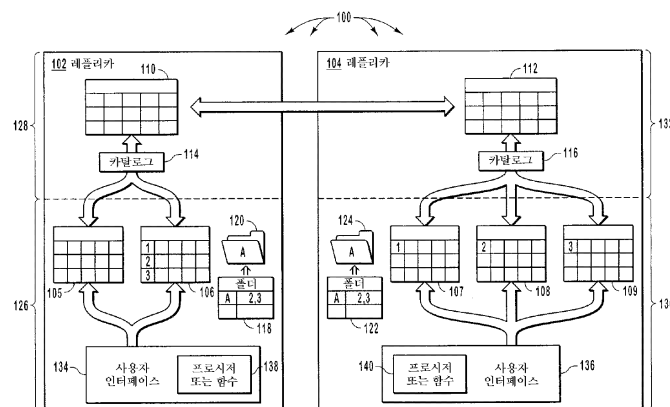
심사관 : 이명진

(54) 물리적 저장 표현에 독립적인 논리적 뷰의 동기화

(57) 요약

토폴로지(topology) 내의 레플리카(replica)에서 아이템을 동기화하기 위한 시스템, 방법, 및 컴퓨터 프로그램 제품이 개시된다. 아이템은 레플리카에서의 물리적 테이블에 저장된다. 각각의 레플리카에서의 물리적 테이블 레이아웃은 상이할 수 있으나 공통의 논리적 뷰(logical view)에 매핑된다. 레플리카에서 동기화가 수행되어야 한다는 것을 나타내는 지시자를 수신하는 경우에, 레플리카는 논리적 뷰로 매핑되어야 하는 변경이 레플리카에서 행해졌는지 여부를 결정한다. 레플리카는 물리적 테이블에서 행해진 변경에 대한 지역 변경 계수(local change enumeration)를 논리적 뷰에서의 변경에 대한 동기화 지역 변경 계수(synchronization local change enumeration)와 비교함으로써 이와 같은 결정을 용이하게 할 수 있다. 동기화될 필요가 있는 임의의 변경은 논리적 뷰를 통하여 토폴로지 내의 다른 레플리카에 송신된다.

대표도



(72) 발명자

조우샤오우

미국 98029 워싱턴주 아이작과 253번 코트 사우쓰
이스트 3250

휴디스이레나

미국 98007 워싱턴주 벨레뷰 145번 애비뉴 사우쓰
이스트 5607

샤아쉬스비.

미국 98074 워싱턴주 삼마미쉬 226번 레인 노쓰이
스트 319

특허청구의 범위

청구항 1

동일한 데이터의 레플리카들(replicas)을 저장하기 위한 복수의 상이한 유형의 플랫폼들을 포함하는 컴퓨터 네트워크에서 - 특정 플랫폼의 제한들(limitations) 또는 특징들(features)은 레플리카의 상기 데이터가 상기 특정 플랫폼의 데이터 저장 계층에서 상이한 물리적 배열로 저장되게 할 수 있으며, 상기 레플리카들을 동기화하는 경우 각각의 레플리카에 대한 상기 데이터는 각각의 플랫폼의 상기 상이한 물리적 배열로부터 다른 상이한 유형들의 플랫폼들 각각의 논리적 뷰 테이블과 공통된 레이아웃을 갖는 상기 특정 플랫폼의 동기화 계층의 논리적 뷰 테이블로 매핑되어야 함 - , 플랫폼의 데이터 저장 계층 내의 아이템들의 물리적 레이아웃을 상기 플랫폼의 동기화 계층 내의 논리적 뷰로 매핑하는 방법에 있어서,

특정 플랫폼의 컴퓨터 시스템에서 논리적 스키마를 카탈로그로 컴파일링하는 단계 - 상기 카탈로그는,

데이터 아이템들의 그룹의 임의의 부분이 수정되면, 데이터 아이템들의 전체 그룹은 다른 레플리카들을 동기화할 때 동기화되도록 데이터 아이템들의 그룹의 세분성(granularity)을 정의하는 변경 유닛, 및

최소 그룹의 임의의 아이템이 변경되면 동기화 동안에 보고되어야 하는 데이터 아이템들의 상기 최소 그룹을 정의하는 일관성 유닛

에 관하여 데이터 아이템들을 정의함으로써 상기 데이터 아이템들의 논리적 그룹핑을 포함하고 각각의 플랫폼에 대한 논리적 뷰와 공통된 레이아웃을 갖는 논리적 뷰를 정의하는 데 이용될 수 있음 - ;

상기 카탈로그를 이용하여 아이템들을 물리적 레이아웃으로부터 하나 이상의 다른 컴퓨터 시스템들의 상이한 플랫폼들의 논리적 뷰와 공통된 레이아웃을 갖는 논리적 뷰로 매핑하는 단계; 및

각각의 플랫폼에 대한 상기 동기화 계층의 논리적 뷰에 동기화 트랙커(tracker)를 저장하는 단계 - 상기 동기화 트랙커는 상기 컴퓨터 시스템의 상기 논리적 뷰의 데이터 아이템의 동기화된 버전에 대응하는 버전 변경 계수(version change enumeration), 및 상기 데이터 아이템의 동기화된 버전이 기초하는 특정 레플리카를 식별하는 소스 식별자를 보존함 -

를 포함하는 방법.

청구항 2

제1항에 있어서, 상기 컴퓨터 시스템의 동기화 계층은 하나 이상의 카탈로그들로 컴파일링되는 논리적 스키마를 포함하며, 상기 하나 이상의 카탈로그들은 XML(extensible mark-up language)로 되어 있는 방법.

청구항 3

제1항에 있어서, 각각의 컴퓨터 시스템에 대한 상기 데이터 저장 계층의 상기 물리적 배열은 하나 이상의 컴퓨터 시스템의 물리적 테이블들에 의해 정의되며, 상기 컴퓨터 시스템들 중 적어도 하나의 물리적 테이블들에 의해 정의되는 물리적 배열은 다른 컴퓨터 시스템들의 물리적 테이블들에 의해 정의되는 물리적 배열과는 상이한 것인 방법.

청구항 4

제3항에 있어서, 레플리카가 저장되어 있는 각각의 컴퓨터 시스템의 각각의 플랫폼의 상기 데이터 저장 계층은 프로시저들 또는 함수들 코드를 포함하는 사용자 인터페이스를 포함하며, 상기 프로시저들 또는 함수들 코드는 상기 레플리카에 대한 데이터를 상기 데이터 저장 계층의 상기 물리적 테이블들에 배열하도록 구성되는 방법.

청구항 5

제3항에 있어서, 각각의 플랫폼의 상기 데이터 저장 계층의 상기 하나 이상의 물리적 테이블들에 지역 변경 트랙커(local change tracker)를 저장하는 단계 - 상기 지역 변경 트랙커는 플랫폼의 상기 데이터 저장 계층에 저장된 아이템들의 지역 변경 계수들을 보존함 - 를 더 포함하며,

상기 지역 변경 트랙커와 상기 동기화 트랙커를 비교함으로써, 각각의 플랫폼의 상기 컴퓨터 시스템은 상기 컴

퓨터 시스템의 상기 데이터 저장 계층에 저장된 아이템이 전송되어 상기 컴퓨터 시스템의 상기 동기화 계층의 상기 논리적 뷰로 매핑되어야 하는지 여부를 결정할 수 있는 방법.

청구항 6

제5항에 있어서, 상기 지역 변경 계수 및 상기 버전 변경 계수가 상이한 값들을 포함하면, 상기 컴퓨터 시스템의 상기 데이터 저장 계층에 저장된 상기 아이템은 상기 논리적 뷰로 매핑되어야 하는 방법.

청구항 7

제5항에 있어서, 상기 지역 변경 계수 및 상기 버전 변경 계수가 동일한 값을 포함하면, 상기 컴퓨터 시스템의 상기 데이터 저장 계층에 저장된 상기 아이템은 상기 논리적 뷰로 매핑될 필요가 없는 방법.

청구항 8

제5항에 있어서,

또다른 플랫폼의 컴퓨터 시스템에 의해 생성된 레플리카의 상이한 버전에 의해 유발된 특정 플랫폼에 저장된 레플리카에 대한 데이터의 아이템의 변경은 다른 플랫폼의 컴퓨터 시스템에 대응하는 소스 식별자에 의해 식별되며,

특정 플랫폼의 상기 데이터 저장 계층의 상기 하나 이상의 물리적 테이블들에서 생성된 변경에 의해 유발된 상기 특정 플랫폼에 저장된 레플리카에 대한 데이터의 아이템의 변경은 상기 변경이 만들어진 연대적인 순서(chronological order)에 대응하는 변경 계수(change enumeration)에 의해 식별되는 방법.

청구항 9

제1항에 있어서, 레플리카가 저장되어 있는 각각의 컴퓨터 시스템의 상기 플랫폼은 각각의 플랫폼에 대하여 공통된 레이아웃을 갖는 논리적 뷰를 정의하기 위해 이용될 수 있는 카탈로그로 컴파일링되는 논리적 스키마를 상기 컴퓨터 시스템의 동기화 계층에 포함하는 방법.

청구항 10

제1항에 있어서, 레플리카가 저장되어 있는 각각의 컴퓨터 시스템에 대한 상기 플랫폼의 상기 데이터 저장 계층의 상기 물리적 배열은 상기 레플리카의 데이터를 저장하기 위한 하나 이상의 물리적 테이블들에 의해 정의되는 방법.

청구항 11

제1항에 있어서, 레플리카가 저장되어 있는 각각의 컴퓨터 시스템에 대한 상기 플랫폼의 상기 데이터 저장 계층은, 공통의 폴더에 그룹핑된 아이템들이 동기화될 수 있는 하나 이상의 폴더들을 포함하여, 상이한 플랫폼들의 동기화 계층들 간의 동기화의 범위를 정의하며,

동기화될 수 있는 공통의 폴더에 그룹핑된 상기 아이템들은 레플리카에 저장된 모든 아이템들보다 적은 방법.

청구항 12

제1항 내지 제11항 중 어느 한 항에 기재된 방법을 수행하기 위한 컴퓨터 프로그램 명령어들을 저장하는 컴퓨터 판독 가능 기록 매체.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- [0018] 본 발명은 일반적으로 데이터 처리 및 취급에 관한 것이다. 더욱 구체적으로, 본 발명은 컴퓨터 시스템 간의 데이터 동기화 분야에 관한 것이다.
- [0019] 컴퓨터 시스템 내에서, 데이터 동기화는 중요한 기능이다. 종종 동일한 데이터를 다수의 상이한 설정 및 지점에서 이용 가능하도록 할 필요성이 존재한다. 데이터를 동기화하는 것이 유용하게 되는 다수의 예 중에서, 한 가지 설명에 도움이 되는 실례는 디지털 주소록을 포함한다. 컴퓨터 사용자는 자신의 데스크탑 작업 컴퓨터에 저장된 디지털 주소록을 가질 수 있다. 직장에는 동안에, 이것은 주소, 전화 번호 및 일반 연락처 정보를 저장하기에 편리하고 접근 가능한 지점이다. 직장 위치에서 멀리 떨어져 있는 경우에, 컴퓨터 사용자는 PDA 또는 다른 이동 정보 저장 시스템과 같은 이동 형태의 주소록을 필요로 할 수 있다. PDA에 있는 연락처 정보는 데스크탑 작업 컴퓨터에 있는 연락처 정보와 일치하여야 한다.
- [0020] 더욱이, 동일한 컴퓨터 사용자는 연락처 정보가 저장되는 가정용 컴퓨터를 가질 수 있다. 이상적으로는, 가정용 컴퓨터, PDA 및 직장 컴퓨터에 있는 연락처 정보가 모두 동기화가 되어 있어야 한다. 다시 컴퓨터 사용자의

직장 위치를 살펴보면, 디지털 주소록에 저장된 데이터의 일부는 그 컴퓨터 사용자가 일하는 조직 내의 다른 컴퓨터 사용자에게 이용 가능한 정보일 수 있다. 그러므로, 이러한 데이터는 다수의 상이한 컴퓨터 사용자들에 의해 액세스 가능한 중앙 집중식 데이터베이스에 저장될 수도 있으며, 여전히 그 컴퓨터 사용자의 PDA, 직장 컴퓨터 및 가정용 컴퓨터와 동기화될 수 있다.

[0021] 따라서, 상술된 예에서, 디지털 주소록을 저장하기 위한 적어도 네 개의 상이한 유형의 플랫폼이 존재한다. 즉, 작업 컴퓨터, PDA, 가정용 컴퓨터 및 중앙 집중식 데이터베이스가 그것이다. 이들 플랫폼의 각각은 데이터 기억 장치(data store)에 디지털 주소록의 사본을 저장하는 레플리카(replica)를 가질 수 있다.

[0022] 각각의 데이터 기억 장치에서의 데이터는 상이한 물리적 배열, 예컨대, 물리적 테이블 및/또는 물리적 테이블의 그룹으로 유지될 수 있다. 물리적 테이블은 데이터베이스 대용량 저장 어레이(database mass storage array), 컴퓨터 하드 드라이브 또는 플래시 메모리와 같은 컴퓨터 저장 장치에 있는 데이터의 실제적인 배열이다. 인식되는 바와 같이, 상이한 플랫폼들의 각각은 다른 레플리카들에 있는 정확히 동일한 데이터를 레플리카에 저장할 수 있다. 그러나, 특정 플랫폼의 제한이나 특징들로 인하여, 데이터는 그 특정 플랫폼에서의 상이한 물리적 배열(즉, 상이한 물리적 테이블 배열들 또는 파일들)에 저장될 수 있다. 데이터를 동일한 토폴로지(topology) 내에서 상이한 레플리카들에 다르게 물리적으로 저장하는 것은 상이한 레플리카들을 서로 간에 동기화시키는 경우에 다수의 과제를 야기한다.

[0023] 일반적으로 레플리카의 데이터는 종종 "아이템(item)"이라고 불리는 데이터의 이산적인 그룹들로 분할된다. 예를 들어, 디지털 주소록에 있어서, 아이템은 성명, 주소, 전화 번호, 전체 연락처, 또는 임의의 다른 이산적인 그룹핑이 될 수 있다. 다른 예에 있어서, 아이템은 파일, 이미지, 폴더 등이 될 수 있다. 레플리카의 아이템들은, 예컨대, 아이템의 추가, 삭제, 및 수정에 의해 변경될 수 있다. 상이한 물리적 배열들로 인하여, 레플리카들 간의 변경들을 동기화하는 것이 어려울 수 있다.

[0024] 상이한 레플리카들에 있는 데이터를 동기화하는데 있어서 발생하는 다른 과제는 동기화 데이터 세분성(synchronization data granularity)의 문맥과 관련된다. 앞서 언급된 바와 같이, 레플리카에서의 데이터는 아이템들로 분할될 수 있다. 이들 아이템은 동기화되는 정보의 이산적인 단편들을 나타낸다. 통상적으로, 아이템의 세분성은 특정 레플리카 토폴로지에 대하여 정의되며, 변하지 않는다. 디지털 주소록의 예에서, 아이템이 연락처, 예컨대 성, 성명, 전화 번호, 거리 주소(street address), 주(state), 또는 ZIP 코드의 단일 필드인 경우에, 아이템은 세밀한 세분성을 갖는다. 반면에, 아이템이 이를테면 성명(성과 성명 모두), 연락처 번호들 또는 주소 중의 하나인 경우에, 아이템은 보통의 세분성을 갖는다. 거친 세분성(coarse granularity)을 갖는 아이템은 레플리카에서의 아이템으로서의 전체 연락처를 포함할 수 있다.

[0025] 레플리카 내의 아이템들의 동기화는 대개 각각의 아이템에 연관되는 메타데이터(metadata)를 필요로 한다. 메타데이터는 아이템이 최후에 변경되었던 때를 나타내는 타임 스탬프와 같은 정보를 포함할 수 있다. 각각의 아이템은 관련 메타데이터를 가져야 하기 때문에, 만약 아이템의 세분성이 지나치게 세밀하다면, 과도한 메타데이터로 인해 불필요하게 특정 레플리카의 자원(예컨대, 저장 장소 및 시스템 메모리)이 낭비될 수 있다. 예를 들어, 앞서 논의된 디지털 주소록에 있어서, 만약 아이템 세분성이 거리 주소, 시 및 ZIP 코드를 포함한다면, 이 세 개의 아이템의 각각에 대한 메타데이터가 유지될 필요가 있을 것이다. 그러나, 거리 주소에 있어서의 변경은 또한 시와 ZIP 코드에 있어서의 변경을 가져오기 쉬운데, 이는 시와 ZIP에 대한 메타데이터는 거리가 갱신되는 경우에 대체로 변경된다는 것을 의미한다.

[0026] 반면에, 만약 세분성이 지나치게 넓다면, 적어도 두 개의 문제점, 즉 동기화 중에 지나치게 많은 동기화 데이터가 전송될 필요가 있을 수 있으며, 불필요한 충돌이 나타날 수 있는 문제점이 발생한다. 예를 들어, 앞서 논의된 디지털 주소록에 있어서, 만약 아이템이 전체 연락처에 의해 정의된다면, 그 연락처의 임의의 부분에서의 변경이 있는 경우에 전체 연락처가 동기화 중에 송신되어야 한다. 그러므로, 동기화 중에 두 개의 레플리카 사이에 잉여 데이터(redundant data)가 송신된다. 예컨대, 연락처에서의 전화 번호에 대한 변경은, 대응하는 연락처를 동기화하기 위하여 그 성명과 주소 정보가 송신되도록 요구하는 것은 아니다. 그러나, 아이템이 전체 연락처로 정의되는 경우에는, 그럼에도 불구하고 전화 번호가 변경되는 경우, 동기화 중에 성명과 주소가 송신되도록 한다. 따라서, 이미 동기화되어 있는 데이터를 전달하는데 통신 자원이 낭비된다.

[0027] 또한, 아이템 정의가 지나치게 거친 경우에는, 레플리카가 레플리카에 있는 데이터 간의 충돌을 부적절하게 검출할 수 있다. 예를 들어, 만약 제1 레플리카에서 연락처의 전화 번호가 변경되고 제2 레플리카에서 그 연락처의 주소가 변경된다면, 그 아이템 세분성이 전체 연락처인 경우 이 제1 및 제2 레플리카들은 충돌 상태에 있는 것으로 보일 수 있다. 그러나, 전화 번호에 있어서의 변경은 완전하게 유효할 수 있고, 주소에 있어서의 변경

과 무관하기 때문에 아무런 실제적인 충돌도 존재하지 않을 수 있다.

[0028] 상용 디지털 저장 애플리케이션에 있어서, 최적화는 특정 애플리케이션이 판매되어 다수의 사용자에게 의해 사용되기 전까지는 대개 발견되지 않으며, 최적화는 물리적 저장에 변경을 가져올 수 있다. 그러므로, 디지털 저장 애플리케이션의 초기 버전에 데이터를 포함하는 물리적 테이블은 그 디지털 저장 애플리케이션의 최근 버전에 동일한 데이터를 저장하는 물리적 테이블과 동일한 레이아웃을 갖지 않을 수 있다. 디지털 저장 애플리케이션의 상이한 버전들 간의 데이터를 동기화시키기 위해서는, 최근 버전의 최적화를 이용하지만 여전히 동기화 능력을 제공하도록 새로운 코드가 작성될 필요가 있을 수 있다.

[0029] 상기 예들은 디지털 주소록과의 관계에서 고안되었지만, 데이터 동기화를 이용하는 다수의 다른 환경이 존재한다. 어떤 예들은 문서 버전화(versioning), 파일 및 정보의 공유, 소프트웨어 업데이트 등을 포함한다. 이들 환경뿐만 아니라 그 밖의 환경들의 각각은 전술된 과제들로 어려움을 겪을 수 있다. 따라서, 컴퓨터 시스템과 통신 자원들을 보다 효율적으로 이용하는 동기화 메커니즘이 이로울 것이다. 데이터 충돌을 보다 적절하게 검출하는 동기화 메커니즘 또한 이로울 것이다.

발명이 이루고자 하는 기술적 과제

[0030] 본 발명의 원리는 레플리카의 동기화를 제공한다. 컴퓨터 시스템은 토폴로지(예컨대, 컴퓨터 네트워크 토폴로지)에 포함되는 다수의 레플리카 중의 하나가 되는 소스 레플리카(source replica)를 포함한다. 컴퓨터 시스템은 소스 레플리카가 수신지 레플리카(destination replica)와 동기화되어야 한다는 것을 나타내는 지시자(예컨대, 요청 동기화 메시지(request synchronization message))를 수신한다. 컴퓨터 시스템은 아이템이 소스 레플리카에 있는 (예컨대, 데이터베이스 테이블 내의) 물리적 레이아웃에서 변경되었는지 여부를 결정한다. 일례에 있어서, 컴퓨터 시스템은 지역 변경 계수(local change enumeration)를 동기화 지역 변경 계수(synchronization local change enumeration)와 비교한다.

[0031] 상술한 비교가 수행되는 경우에, 예컨대 지역 변경 계수와 동기화 지역 변경 계수가 상이하다면, 물리적 레이아웃에서의 아이템이 변경된 것으로 결정된다. 컴퓨터 시스템은 소스 레플리카에 있는 물리적 레이아웃에서의 임의의 변경된 아이템을 논리적 뷰에 매핑하는데, 이 논리적 뷰는 토폴로지 내의 하나 이상의 다른 레플리카에서의 논리적 뷰와 실질적으로 유사하다. 각각의 레플리카에서의 카탈로그(catalog)는 논리적 뷰가 실질적으로 유사하도록 하기 위해 물리적 레이아웃으로부터 논리적 뷰로의 매핑을 제어한다. 컴퓨터 시스템은 논리적 뷰로 매핑된 적어도 하나의 아이템을 수신지 레플리카로 송신한다.

[0032] 본 발명의 원리는 또한 컴퓨터 시스템이 토폴로지 내의 다른 컴퓨터 시스템과 데이터를 동기화할 수 있도록 한다. 컴퓨터 시스템은 다수의 아이템 및 이 다수의 아이템에 대한 지역 변경 계수를 유지하는 지역 변경 트래커(local change tracker)를 포함하는 데이터 저장 계층(data store layer)을 포함한다. 컴퓨터 시스템은 또한 토폴로지 내의 하나 이상의 다른 컴퓨터 시스템에서의 논리적 뷰와 실질적으로 유사한 논리적 뷰를 포함하는 동기화 계층(synchronization layer)을 더 포함한다. 논리적 뷰는 다수의 아이템으로부터의 하나 이상의 아이템의 매핑을 나타낸다. 동기화 계층은 또한 논리적 뷰로 매핑된 하나 이상의 아이템에 대한 버전과 동기화 지역 변경 계수를 유지하는 동기화 변경 트래커(synchronization change tracker)를 더 포함한다.

[0033] 본 발명의 원리는 또한 데이터 저장 장소에서의 아이템의 물리적 레이아웃에서 논리적 뷰로의 매핑을 생성할 수 있도록 한다. 컴퓨터 시스템은 논리적 스키마의 컴파일을 용이하게 하는데 사용될 수 있는 코어 코드(core code)에 액세스한다. 컴퓨터 시스템은 변경 유닛과 일관성 유닛의 정의를 포함하는 논리적 스키마에 액세스한다. 변경 유닛은 개발자가 가용한 자원에 기초하여 변경 유닛의 크기를 할당할 수 있도록 설정 가능하다. 예를 들어, 낮은 대역폭 또는 높은 대기 시간 네트워크 접속에 대해서는 네트워크 상에서 더 작은 아이템과 더 적은 데이터를 가져오는 하위의 세분성(granularity)이 사용될 수 있다. 또한, 일관성 유닛은 개발자가 동시에 동기화되어야 하는 아이템을 함께 그룹핑할 수 있도록 설정될 수 있다. 예를 들어, 공동으로 하나의 주소를 구성하는 아이템들 중의 임의의 아이템에 대한 변경은 그들 모든 아이템에 대한 변경을 가져올 수 있기 때문에, 개발자는 이들 아이템들을 그룹핑할 수 있다.

[0034] 컴퓨터 시스템은 아이템을 물리적 레이아웃으로부터 논리적 뷰로 매핑하는 적어도 하나의 카탈로그로 논리적 스키마를 컴파일하기 위해 코어 코드를 이용한다. 컴파일은 결국 프로시저나 함수가 생성되도록 할 수 있다. 프로시저 또는 함수는 아이템이 어떻게 물리적 레이아웃에 저장되는지를 지시한다. 프로시저나 함수로의 액세스를 통해, 카탈로그는 데이터의 물리적 레이아웃을 하나 이상의 다른 컴퓨터 시스템에서의 논리적 뷰와 실질적으로 유사한 논리적 뷰에 매핑할 수 있다. 다수의 컴퓨터 시스템 간에 유사한 논리적 뷰를 사용함으로써, 레플리

카드 간에 보다 효율적인 동기화를 촉진할 수 있다.

[0035] 본 발명의 부가적인 특징은 이하의 설명에서 기술될 것인 바, 부분적으로는 설명으로부터 명백하거나 본 발명의 실시예에 의해 알게될 수 있다. 본 발명의 특징 및 장점은 첨부된 청구범위에서 특히 지적된 장치나 조합에 의해서 실현되고 달성될 수 있다. 본 발명의 이들 특징 및 다른 특징은 이하의 설명과 첨부된 청구범위로부터 보다 완전하게 분명해질 것이며, 이하에 설명된 본 발명의 실시예에 의해 알 수 있게 될 것이다.

발명의 구성 및 작용

[0036] 본 발명의 상술된 특징 및 다른 특징을 얻을 수 있는 방식을 설명하기 위하여, 앞서 간략하게 설명된 본 발명이 첨부된 도면에 도시된 특정한 실시예를 참조하여 보다 상세하게 설명될 것이다. 이들 도면은 본 발명의 전형적인 실시예만을 나타내므로, 본 발명의 범위를 제한하는 것으로 간주되어서는 안 된다는 점을 이해하여야 하며, 본 발명은 첨부된 도면의 사용을 통해 보다 구체적이고 상세하게 기술되고 설명될 것이다.

[0037] 본 발명은 논리적 뷰(logical view)를 통하여 레플리카의 아이템들을 동기화하기 위한 시스템, 방법 및 컴퓨터 프로그램 제품까지 확장된다. 레플리카는 물리적 레이아웃에서 논리적 뷰로 아이템을 매핑하기 위하여 카탈로그(catalog)를 이용한다. 카탈로그는 상이한 물리적 레이아웃(예컨대, 복수의 상이한 레플리카에 있음)에서 실질적으로 유사한 논리적 뷰(예컨대, 그 복수의 상이한 레플리카 간에 공유됨)로의 데이터의 매핑을 용이하게 하는 논리적 스키마로부터 컴파일될 수 있다. 이하에서 보다 상세하게 논의되는 바와 같이, 본 발명의 실시예들은 컴퓨터 하드웨어의 다양한 아이템을 포함하는 특수 목적용 컴퓨터나 범용 컴퓨터를 포함할 수 있다.

[0038] 이제 도 1을 참조하면, 본 발명의 원리에 따라 동기화될 수 있는 레플리카를 포함하는 예시적인 컴퓨터 아키텍처가 도시되어 있다. 도 1은 토폴로지(100)를 나타낸다. 토폴로지(100)는 레플리카(102) 및 레플리카(104)를 포함하는데, 이들은 일반적인 네트워크(예컨대, LAN, WAN, 또는 인터넷)에 접속된 컴퓨터 시스템들에 있을 수 있다. 레플리카(102)는 데이터 저장 계층(data store layer; 126) 및 동기화 계층(synchronization layer; 128)으로 분할된다. 레플리카(104)도 데이터 저장 계층(130) 및 동기화 계층(132)으로 분할된다. 레플리카(102)는 데이터 저장 계층(126)에 저장되는 레플리카(102)의 데이터를 포함하는 물리적 테이블(105 및 106)을 포함한다. 레플리카(104) 또한 데이터 저장 계층(130)에 저장되는 레플리카(104)의 데이터를 포함하는 물리적 테이블(107, 108 및 109)을 포함한다.

[0039] 물리적 테이블(105 및 106)과 물리적 테이블(107, 108 및 109)은 동일한 정보를 포함할 수 있다. 그러나, 이 테이블들은 물리적으로 상당히 다르게 배열될 수 있다. 이들 차이점은 테이블이 존재하는 레플리카의 유형의 결과일 수 있다. 예를 들어, 서버 레플리카에 대한 최적화는 클라이언트 레플리카에 대한 최적화와 다를 수 있는데, 이는 상이한 물리적 테이블 레이아웃을 가져올 수 있다. 대안적으로, 물리적 테이블은 동일한 상용 소프트웨어의 상이한 버전들에 존재할 수 있다. 대개 상용 소프트웨어 애플리케이션의 최근 버전은 최적화를 포함하고 물리적 테이블의 이전 버전들과는 상이한 물리적 테이블 레이아웃을 가질 것이다.

[0040] 아이템은 사용자 인터페이스(134)를 통하여 레플리카(102)에 있는 테이블(105 및 106)에 입력될 수 있다. 사용자 인터페이스(134)는 아이템의 테이블 레이아웃이 테이블(105 및 106)에 추가되거나 변경되도록 하는 코드를 포함한다. 본 발명의 일 실시예에 있어서, 이러한 코드는 프로시저 또는 함수 코드(138)이다. 레플리카(104)는 레플리카(102)에서의 사용자 인터페이스(134)와 유사하게 기능하는 사용자 인터페이스(136)를 포함한다. 사용자 인터페이스(136) 또한 프로시저 또는 함수 코드(140)를 이용할 수 있다.

[0041] 레플리카(102)는 논리적 뷰(110)도 포함한다. 물리적 테이블(105 및 106)은 논리적 뷰(110)에 매핑된다. 레플리카(104) 또한 논리적 뷰(112)를 포함한다. 물리적 테이블(107, 108 및 109)은 논리적 뷰(112)에 매핑된다. 논리적 뷰(110 및 112)는 실질적으로 유사한 논리적 뷰가 될 수 있는데, 예를 들어 논리적 뷰(110 및 112)는 모두 연락처 정보를 나타낼 수 있다. 아이템은 논리적 뷰(110 및 112) 모두에서 유사하게 나타날 수 있다. 본 발명의 일 실시예에 있어서, 레플리카(102)에서 테이블(105 및 106)로부터 논리적 뷰(110)로의 매핑은, 레플리카(102)에 있는 아이템의 물리적 저장 위치에 관한 정보를 포함하는 카탈로그(114){동기화 계층(128)에 있음}를 이용하여 달성된다. 레플리카(104)에 대한 매핑은 레플리카(102)에 대한 매핑과 유사하다. 레플리카(102 및 104)는 논리적 뷰(110 및 112)를 통하여 동기화된다. 본 발명의 일 실시예에 있어서, 물리적 스키마 매핑 문서(이는 이하에서 보다 상세하게 설명될 것임)에 대한 XML 또는 다른 논리적 스키마가 카탈로그 및 프로시저나 함수를 생성하기 위해 컴파일된다.

[0042] 예시적인 일 실시예에 있어서, 동기화는, 동기화가 개시되어야 하는 레플리카(102)에 의해 수신되는 지시자에 응답하여 수행된다. 지시자는, 레플리카(102)가 사용자 인터페이스(134)를 통하여 변경되었거나 추가된 테이블

(105 및 106)에 아이템을 갖고 있다는 레플리카(102)에 의한 인식, 레플리카(104)로부터 동기화에 대한 요청의 수신, 또는 임의의 다른 적합한 지시자를 포함할 수 있다. 그 후, 레플리카(102)는 카탈로그(114)를 통하여 논리적 뷰(110)로 매핑될 필요가 있는 물리적 테이블(105 및 106)에서의 임의의 아이템을 매핑한다. 그 후, 아이템은 레플리카(102)의 논리적 뷰(110)로부터 레플리카(104)의 논리적 뷰(112)로 송신된다. 그 후, 레플리카(104)는 논리적 뷰(112)로부터 송신된 아이템을 카탈로그(116)를 통하여 물리적 테이블(107, 108 및 109)로 매핑할 수 있다.

[0043] 본 발명의 일 실시예에 있어서, 동기화의 범위는 폴더(folder)의 사용을 통해 제어될 수 있다. 폴더는 아이템을 함께 그룹핑하기 위해 사용되는 논리적 구조이다. 그러므로 폴더를 동기화함으로써, 공통의 폴더에 그룹핑된 아이템들이 동기화될 수 있다. 폴더를 이용하는 경우에, 폴더에 그룹핑된 아이템들만을 동기화함으로써 레플리카 상에 저장된 모든 아이템들 보다 적은 아이템들에 대해서 동기화가 수행될 수 있다. 폴더 그룹핑의 예가 도 1에 도시되어 있다. 폴더 그룹핑은 레플리카에서의 데이터 저장 계층에 있는 폴더 테이블을 이용함으로써 달성될 수 있다. 도 1은 데이터 저장 계층(126)에 테이블(106)을 도시함으로써 이러한 개념을 설명한다. 테이블(106)은 1, 2 및 3으로 라벨링된 데이터 아이템을 포함한다. 폴더 테이블(118)은 식별 번호 2와 3에 대응하는 엔트리 A를 포함한다. 이 테이블 엔트리는 2와 3으로 라벨링된 데이터 아이템을 포함하는 A로 라벨링된 폴더(120)를 나타낸다. 레플리카(104)는 1, 2 및 3으로 라벨링된 데이터 아이템을 저장하는 테이블(107, 108 및 109)을 포함하는 대응 논리적 구조를 갖는다. 이들 아이템은 물리적으로 다르게 저장되더라도, 레플리카(102)에서의 1, 2 및 3으로 라벨링된 아이템들에 대응한다. 폴더 테이블(122)은 레플리카(104)의 데이터 저장 계층(130)에 저장된다. 폴더 테이블(122)은 A로 라벨링된 폴더(124)에 매핑된다. 따라서, 레플리카(102)는 폴더 수준에서 레플리카(104)와 동기화될 수 있다.

[0044] 논리적 뷰는 XML 스키마 또는 소정의 다른 트리와 같은 데이터 구조에 의해 정의될 수 있다. 이러한 데이터 구조의 예는 도 2a에 도시되어 있다. 논리적 뷰를 이해하기 위해서는, 먼저 다수의 다른 용어들이 이해되어야 한다. 예를 들어, 일관성 유닛(consistency unit)은 동기화 중에 보고될 수 있는 아이템 변경의 최소 그룹을 나타낸다. 일관성 유닛에 의해 정의될 수 있는 데이터의 일례는 주소이다. 그리고 주소는 거리, 시, 주 및 ZIP 코드를 포함하는 몇 개의 개별적인 컴포넌트로 이루어져 있다. 주소가 갱신되는 경우에는, 대체적으로 이들 아이템을 중의 몇 가지가 변경된다. 그러므로, 만약 동기화 중에, 변경된 이들 아이템들 중의 일부만이 전송되어 이들을 수신한 레플리카에서 변경된다면, 결과적으로 불일치하고 의미 없는 주소가 레플리카에 발생될 수 있다. 그러한 것으로서, 오직 일관성 유닛의 변경된 컴포넌트들, 즉 거리, 시, 주 및 ZIP 코드의 각각이 동기화 중에 송신되는 경우에만, 일관성 유닛으로 정의된 주소가 동일한 동기화 중에 레플리카에서 갱신될 것이다.

[0045] 이해되어야 하는 또 다른 개념은 변경 유닛(change unit)의 개념이다. 변경 유닛은 아이템의 경계를 정의하는 구조이다. 다시 말해서, 만약 변경 유닛의 임의의 부분이 수정된다면, 전체의 변경 유닛이 후속의 동기화에서 동기화될 것이다. 일례에 있어서, 변경 유닛은 전화 번호로서 정의될 수 있다. 지역 코드에 대한 변경은 전체의 전화 번호가 후속의 동기화 중에 송신되도록 할 것이다.

[0046] 이제 도 2a를 참조하면, 일관성 유닛과 변경 유닛이 논리적 스키마(200)의 일반화된 뷰에 도시되어 있다. 논리적 스키마(200)는 최상위 레벨 일관성 유닛(202)을 포함한다. 이 최상위 레벨 일관성 유닛(202)에 두 개의 하위 레벨 일관성 유닛이 추가된다. 제1 하위 레벨 일관성 유닛(204)은 변경 유닛(206 및 208)을 포함한다. 제2 하위 레벨 일관성 유닛(210)은 변경 유닛(212, 214, 216 및 218)을 포함한다. 논리적 스키마(200)의 보다 구체적인 예는 연락처(252){최상위 레벨 일관성 유닛(202)에 대응됨}를 나타내는 도 2b에 도시되어 있다. 성명(254){추가된 제1 일관성 유닛(204)에 대응됨}은 연락처(252) 내에 도시된다. 성명(254)은 이름(256)과 성(258)을 포함한다. 주소(260){제2 일관성 유닛(210)에 대응됨}가 도시되어 있는데, 이것은 거리(262), 시(268), 주(266) 및 ZIP 코드(268)를 포함한다.

[0047] 본 발명의 실시예들은 정의 가능한 일관성 유닛 및 변경 유닛을 고려한다. 예를 들어, 일관성 유닛 및 변경 유닛은 각각의 특정한 애플리케이션에 대한 XML 스키마에서 정의될 수 있다. 본 발명의 일 실시예에 있어서, 개발자는 XML 스키마를 생성한다. 이러한 XML 스키마는 코어 코드를 포함하는 컴퓨터 시스템에 저장된다. 컴퓨터 시스템에서의 코어 코드는 XML 스키마를 컴파일한다. XML 스키마를 컴파일함으로써, {도 1에 도시된 레플리카(102)에서의 논리적 뷰(110)와 물리적 테이블(105 및 106)과 같은} 논리적 뷰와 물리적 테이블 사이에서 아이템을 매핑하는데 사용되는 {도 1에 도시된 레플리카(102)에서의 카탈로그(114)와 같은} 카탈로그가 생성된다. 컴파일은 또한 프로시저나 함수 코드가 생성되도록 할 수 있다. 프로시저나 함수 코드는 {도 1에 도시된 레플리카(102)에서의 데이터베이스 계층(126)과 같은} 레플리카의 데이터베이스 계층에 저장될 수 있다. 프로시저나 함수 코드는, 아이템이 테이블 또는 다른 저장 배열에 어떻게 저장될 것인지를 지시하는 것과 같은, {도 1에

도시된 물리적 테이블(105 및 106)과 같은 물리적 저장 장소에 아이템을 저장하기 위한 기능을 제공한다.

- [0048] 본 발명의 일 실시예에 있어서, 사용자 인터페이스를 통하여 행해진 변경과 같은, 지역적으로 행해진 변경은 동기화에 기인한 변경과 구별될 수 있다. 이제 도 1 및 도 3a를 참조하여 그러한 기능이 설명된다. 앞서 언급된 바와 같이, 레플리카(102)는 데이터 저장 계층(126)에 저장된 물리적 테이블(105 및 106)을 포함한다. 도 3a는 사용자 인터페이스(134)를 통하여 행해진 갱신이 물리적 테이블(105 및 106)과 논리적 뷰(110)를 이용하여 어떻게 동기화되는지를 나타내는 시간 방향 경과를 도시한다. 도 3a에 도시된 물리적 테이블(105)은 두 개의 열, 즉 아이템 열(302) 및 지역 변경 계수(local change enumeration; LOCAL_CN) 열(304)을 갖는다. 아이템 열(302)은 레플리카(102)에 저장된 아이템을 포함한다. 지역 변경 계수 열(304)은 데이터 저장 계층(126)이 아이템에 라벨로서 할당한 지역 변경 계수를 포함한다. 각각의 아이템에 연관된 지역 변경 계수는 아이템에 변경이 행해질 때마다 증가적으로 갱신된다. 데이터 저장 계층(126)은 그러한 변경들이 사용자 인터페이스를 통하여 행해지는지 아니면 동기화를 통하여 수신되는지 여부에 관계없이 모든 변경에 지역 변경 계수를 할당한다. 지역 변경 트랙커(local change tracker)는 지역 변경 계수 열(304)을 포함할 수 있다.
- [0049] 논리적 뷰(110)는 세 개의 열, 즉 카탈로그(114)를 통하여 물리적 테이블(105 및 106)로부터 매핑되는 아이템을 저장하는 아이템 열(306), 아이템에 할당된 지역 변경 계수의 동기화 형태를 저장하는 동기화 지역 변경 계수 열(308), 및 아이템에 대해 변경을 수행하였거나 버전을 할당했던 토폴로지 내의 레플리카와 아이템이 추가되었거나 변경된 연대 순서에 관한 정보를 포함하는 버전을 포함하는 버전 열(310)의 세 개의 열을 갖는 테이블을 포함한다. 동기화 변경 트랙커는 동기화 지역 변경 계수 열(308) 및 버전 열(310)의 정보를 포함할 수 있다.
- [0050] 시간 (1)에서는 레플리카(102)가 안정한 상태에 있다. 시간 (2)에서는, 사용자 인터페이스(134)를 통하여 I2로 라벨링된 아이템에 대해 변경이 수행된다. 지역 변경 계수가 I2에 할당되는데, 이 지역 변경 계수는 이용 가능한 다음 연대순의 변경 계수가 되는데, 이를테면 이 경우에는 12가 된다.
- [0051] 시간 (3)에서, 레플리카(102)는 동기화가 수행되어야 한다는 것을 나타내는 지시자를 수신한다. 동기화 계층(128)에서, 레플리카(102)는 동기화 지역 변경 계수(308)를 열(304)에서의 지역 변경 계수와 비교함으로써 논리적 뷰 테이블(110)을 검사한다. 열(304)에서의 지역 변경 계수가 열(308)에서의 동기화 지역 변경 계수와 일치하지 않기 때문에, 레플리카(102)는 물리적 테이블(105 및 106)에 변경이 행해졌다는 사실을 발견한다. 즉 지역 변경 계수는 12의 값을 가지며 동기화 지역 변경 계수는 5의 값을 가진다. 그러므로, 레플리카(102)는 열(310)에서의 버전을, 변경이 레플리카(102)에서 행해졌으며{이 예에서, 레플리카(102)는 토폴로지에서의 레플리카 A로 라벨링됨}, 레플리카(102)는 12의 지역 변경 계수를 할당받았다는 것을 나타내는 A12로 갱신한다. 이와 거의 동시에, I2로 라벨링된 아이템이 지역 테이블(105 및 106)로부터 논리적 뷰 테이블(110)로 매핑된다. 그 후 시간 (4)에서는, 열(304)에서의 지역 변경 계수와 일치하도록 열(308)에서의 동기화 지역 변경 계수가 갱신된다. 그 후, 물리적 테이블(105 및 106)로부터 매핑된 아이템을 포함하는 논리적 뷰(110)에서의 아이템은 토폴로지(100) 내에서 다른 레플리카들과 동기화될 수 있다.
- [0052] 이제 도 3b를 참조하면, 동기화의 결과로서, 지역 테이블을 포함하는 레플리카의 갱신에 대한 예가 도시되어 있다. 시간 (1)에서, 레플리카(102)는 토폴로지 내의 다른 레플리카들로부터 아무런 새로운 아이템 또는 변경도 수신하지 않은 안정한 상태에 있다. 시간 (2)에서, 아이템 I3에 대한 변경이 논리적 뷰 테이블(110)에 수신된다. 아이템 I3에 대한 변경은 버전 C101을 갖는다. 이러한 변경은 논리적 뷰 테이블(110)로부터 카탈로그(114)를 통하여 물리적 테이블(105)로 송신된다. 이 변경을 수신하는 경우에, 데이터 저장 계층(126)은 데이터 저장 계층(126)에 의한 할당을 위해 이용 가능한 다음 변경 계수가 되는 지역 변경 계수를 할당한다. 이러한 경우에, 시간 (3)에서 열(304)에 도시된 바와 같이, 지역 변경 계수는 13이 된다. 그 후에 즉시 또는 곧, 시간 (4)에 도시된 바와 같이, 열(304)에서의 변경에 할당된 지역 변경 계수를 알아내기 위한 검사가 동기화 계층(128)에서 수행된다. 열(308)에서의 동기화 지역 변경 계수는 열(304)에서의 지역 변경 계수로 갱신된다. 이러한 방법으로, 동기화 계층은, 열(304)에서의 지역 변경 계수와 열(308)에서의 동기화 지역 변경 계수가 일치하는 한, 동기화되지 않은 어떠한 지역 변경도 사용자 인터페이스(134)를 통하여 아이템 I3에 대해 행해지지 않았음을 알게 될 것이다. 또한, 열(310)에서의 버전은 어느 레플리카가 변경을 행하였는지 및 그 변경이 행해진 연대순에 관한 정보를 포함한다{이 경우에는 시간 101에 C로 라벨링된 레플리카}.
- [0053] 본 발명의 소정의 실시예들에 있어서는, 앞서 논의된 논리적 뷰와는 다른 커스텀 뷰(custom view)를 제공할 필요성이 존재할 수 있다. 즉, 앞서 논의된 논리적 뷰는 동기화 효율이 극대화되도록 모든 레플리카에 대하여 동일한 레이아웃을 갖는다. 그러나, 논리적 뷰로 설계되지 않은 레플리카가 토폴로지에 추가되어 그들을 포함하도록 갱신될 수 없는 경우가 있을 수 있다. 그러나, 본 발명의 실시예들은 이들 커스텀 레플리카와 호환 가능

한 것으로 의도된다. 이제 도 4를 참조하면, 이들 커스텀 레플리카를 수용하는 한 가지 방법이 도시되어 있다. 도 4는 {도 1에 도시된 계층(128)과 같은} 동기화 계층에 할당된 열(310)에서의 버전과 커스텀 레플리카에 의해 일반적으로 할당된 열(410)에서의 커스텀 변경 계수를 커스텀 뷰(404)에 매핑하는 시스템을 나타낸다. 그 후, 커스텀 뷰(404)를 통해 아이템을 송수신하는 부가적인 동작으로 전술한 바와 같이 동기화가 발생할 수 있다. 예를 들어, 레플리카에서 커스텀 레플리카로 아이템이 송신되는 경우에, 그 아이템은 송신되기 전에 적절한 커스텀 변경 계수를 할당받을 수 있다.

[0054] 아이템을 수신하는 경우에는, 커스텀 뷰(404)를 참조함으로써, 적절한 버전이 할당될 수 있어, 아이템이 {도 1에 도시된 논리적 뷰(110)와 같은} 논리적 뷰에서의 적절한 위치에 매핑될 수 있다.

[0055] 도 5는 본 발명의 소정의 실시예들의 형태를 이용하여 레플리카들을 동기화하기 위한 예시적인 방법(500)을 도시한다. 상기 방법(500)은, 도 1에 도시된 것과 같은 컴퓨터 아키텍처에서 실시될 수 있다. 따라서, 도 5의 방법은 도 1의 구성 요소들을 참조하여 상세히 설명될 것이다. 상기 방법(500)은, 논리적 뷰에 매핑되는 물리적 레이아웃으로부터 아이템들을 식별하기 위한 기능적 결과 지향 단계(functional result oriented step)를 포함한다{단계(502)}. 단계(502)는 논리적 뷰에 매핑되는 물리적 레이아웃으로부터 아이템들을 확인하기 위한 임의의 대응 동작들(acts)을 포함할 수 있다.

[0056] 그러나, 도 5의 예시적 방법에서, 단계(502)는 동기화 지시자를 수신하는 대응 동작을 포함하는데{동작(504)}, 동기화 지시자는 소스 레플리카가 목적지 레플리카와 동기화되어야 한다는 것을 나타낸다. 동작(504)은 소스 레플리카가 목적지 레플리카에 동기화되어야 한다는 것을 나타내는 지시자를 수신하는 컴퓨터 시스템을 포함한다. 예를 들어, 레플리카(102)를 포함하는 컴퓨터 시스템은, 레플리카(102)가 레플리카(104)와 동기화되어야 한다는 지시자를 수신할 수 있다.

[0057] 동작(504)은 도 1에 도시되는 레플리카(102)의 동기화 계층(128)과 같은 레플리카의 동기화 계층에서 발생할 수 있다. 레플리카(102)는, 데이터를 동기화하는 다른 레플리카들을 포함하는 토폴로지(100) 내에 있다. 동작(504)에서 설명한 지시자는 다수의 상이한 소스들로부터 발생하는데, 이러한 소스들에 포함되는 동기화 요구는 토폴로지(100) 내의 다른 레플리카로부터 또는 레플리카(102) 내의 타이머 만료로부터 또는 데이터베이스 계층(126) 내에 저장된 데이터에 대한 일부 변경의 결과로서 플래그를 설정하는 것에 의해 발생된다. 본 명세서에서는 구체적으로 설명되지 않더라도, 다른 다양한 지시자들도 또한 수신되어 동기화가 수행되어야 한다는 것을 지시하는 적합한 지시자들이 될 수 있다.

[0058] 단계(502)는 또한 소스 레플리카에 있는 물리적 레이아웃 내의 아이템들이 변경되었는지 여부를 결정하는 대응 동작을 포함한다{동작(506)}. 동작(506)은 소스 레플리카에 있는 물리적 레이아웃 내의 아이템들이 변경되었는지 여부를 결정하는 컴퓨터 시스템을 포함할 수 있다. 예를 들어, 레플리카(102)를 포함하는 컴퓨터 시스템은 테이블(105 및 106)에 물리적으로 저장된 아이템들이 변경되었는지 여부를 결정할 수 있다.

[0059] 도 3a에 도시된 동작(506)의 일 실시예가 시간 (2)에서 도시되는데, 여기에서는, 앞서 설명한 바와 같이, 열(308)에서의 동기화 지역 변경 계수가 열(304)에서의 지역 변경 계수와 비교된다.

[0060] 열(304)의 지역 변경 계수는 레플리카(102)의 데이터 저장 계층(126) 내에 있고, 데이터 저장 계층(126)에 의해, 예컨대 테이블(105) 내의 아이템과 관련된다. 열(308)의 동기화 지역 변경 계수는 레플리카(102)에 있는 논리적 뷰(110)에 매핑된 아이템과 연관되는데, 레플리카(102)에서, 매핑된 아이템들은 테이블(105 및 106)의 물리적 레이아웃 내에 있는 대응 아이템들을 가진다. 열(304)의 지역 변경 계수는 열(308)의 동기화 지역 변경 계수에 독립적으로 변경될 수 있다. 따라서, 지역적인 변경이 수행되는 경우에, 열(304)의 지역 변경 계수는 열(308) 내의 동기화 지역 변경 계수의 값과는 다른 값으로 변경된다.

[0061] 이와 같이, 동작(506)을 수행하는 경우에, 소스 레플리카에 있는 물리적 레이아웃 내의 아이템들에 대해 변경이 이루어졌는지 여부를 결정하기 위해, 지역 변경 계수들과 동기화 지역 변경 계수들이 비교될 수 있다. 예를 들어, 지역 변경 계수 및 동기화 지역 변경 계수가 상이한 값들을 가진다면, 이는 물리적 레이아웃 내의 변경을 나타낼 수 있다. 아이템이 저장된 레플리카가 아닌 다른 레플리카에서 아이템이 변경되었거나, 토폴로지 내의 다른 레플리카에 대해 아이템이 동기화되었다면, 지역 변경 계수 및 동기화 변경 계수는 동일한 값이 될 수 있다. 상술된 도 3a 및 3b에 관한 논의는 본 발명의 일 실시예에서 이것이 어떻게 달성되는 지에 대한 일례를 설명한다.

[0062] 레플리카의 동기화 계층에 있는 동기화 변경 트랙커는 논리적 뷰들에 매핑된 아이템들과 연관된 동기화 지역 변경 계수들과 버전들을 유지한다. 예를 들어, 테이블(307)은 동기화 계층(128)에 위치할 수 있다. 동기화 변경

트랙커 내의 버전은 레플리카 ID들을 포함할 수 있는데, 이 레플리카 ID들은 대응 버전과 연관된 아이тем들을 변경시키는 레플리카를 확인한다. 버전은 레플리카들이 변경된 시간 순서를 식별하는 레플리카 변경 계수들을 포함할 수도 있다. 도 3a에서, {열(306) 내의} 아이тем I1은 버전 A4와 연관되는데, 버전 A4에서 A는 레플리카 ID(예를 들어, 레플리카(102))이고 4는 레플리카 변경 계수이다. 따라서, 아이тем I1에 대한 변경은 시간 순서 4의 레플리카(102)에서 행해진다. 본 발명의 소정의 실시예에서, 레플리카 ID는 변경을 행하는 레플리카가 아닌 레플리카를 식별한다. 이러한 실시예들에서는, 토폴로지 내의 상이한 몇몇의 레플리카들에 의해 이루어진 변경들에 버전들을 할당하는데 하나의 레플리카가 사용될 것이다. 이는 대리 저작(surrogate authoring)이라고 불린다.

[0063] 본 발명의 방법(500)은 또한 물리적 레이아웃 내의 각각의 아이тем을 위한 변경 유닛을 정의하는 단계를 포함할 수 있는데, 상기 물리적 레이아웃에서 변경 유닛은 아이тем의 세분성을 정의한다. 변경 유닛은, 도 2a 및 2b에 도시된 바와 같이, 논리적 스키마 또는 논리적 구조에서 정의될 수 있다.

[0064] 본 발명의 방법(500)은 또한 논리적 스키마에서 일관성 유닛을 정의하는 단계를 포함할 수 있다. 일관성 유닛은 정의된 세분성의 복수의 아이тем을 포함하는데, 이 복수의 아이тем은 임의의 변경 유닛들이 레플리카에 의해 적용될 수 있도록 그 레플리카에 의해 수신되어야 한다. 예를 들어, 도 2b는 주소 일관성 유닛(260)을 도시하는데, 주소 일관성 유닛은 거리(262), 시(264), 주(266) 및 ZIP(268) 변경 유닛들을 포함한다. 동기화 중에, 거리(262), 시(264), 주(266), 및 ZIP(268) 변경 유닛들 중의 임의의 변경된 아이тем들이 레플리카에 의해 수신되지 않는다면, 아이тем들 중의 어느 것도 그 레플리카에서 갱신되지 않을 것이다. 일관성 유닛 내의 모든 아이тем이 수신되었을 때, 모든 아이тем들이 레플리카에서 동시에 갱신될 수 있다.

[0065] 본 발명의 방법(500)은 또한 논리적 스키마를 컴파일하여, 예컨대 카탈로그(114)와 유사한 카탈로그를 생성하는 단계를 포함할 수 있다. 카탈로그는 아이тем들이 물리적 테이블(105 및 106)에서 물리적으로 놓여있는 위치에 대한 정보를 포함한다. 더욱 구체적으로, 본 발명의 일 실시예에서, 카탈로그는, 물리 저장 장소 내의 아이тем들을 논리적 뷰들 내의 아이тем들로 매핑하는 프로시저 및 함수를 생성하기 위한 명령어와 상세한 메타데이터(metadata)를 포함한다. 따라서, 물리 테이블(105 및 106) 내의 아이тем들을 논리적 뷰(110)에 매핑하기 위해 카탈로그가 사용된다. 논리적 스키마가 컴파일될 때, 프로시저 및 함수 코드가 또한 생성될 수 있다. 프로시저 또는 함수 코드는 {레플리카(102)의 데이터베이스 계층(126)과 같은} 레플리카의 데이터 기반 계층에 저장될 수 있다. 프로시저 또는 함수 코드는 아이тем들을 {물리 테이블들(105 및 106)과 같은} 물리 테이블들의 물리적 레이아웃에 저장하기 위해, 아이тем을 테이블에 저장하는 방법을 지시하는 등의 기능을 제공한다.

[0066] 본 발명의 방법(500)은 또한 동작(504) 전에 카탈로그를 설치하는 단계를 포함할 수 있다. 카탈로그는 개발자 또는 카탈로그가 상술한 컴퓨터 시스템이 아닌 다른 컴퓨터 시스템에서 개발되고 컴파일된 다른 소스에 의해 제공될 수 있다. 그 후, 카탈로그는 대상 컴퓨터 시스템에서 실행될 수 있는 설치 가능한 코드로 제공된다. 마찬가지로, 프로시저 또는 함수 코드도 컴퓨터 시스템에 제공되어 설치될 수 있다.

[0067] 본 발명의 방법(500)은 또한 레플리카에 폴더를 정의하는 단계를 포함할 수 있는데, 폴더는 레플리카에서의 아이тем들의 일부를 포함한다. 토폴로지 내의 다른 레플리카는 또한 대응 아이тем들과 함께 대응 폴더를 포함할 수 있다. 도 1은 레플리카(102)에서의 폴더(120)를 도시하는데, 상기 폴더(120)는 레플리카(104)에서의 폴더(124)에 대응한다. 레플리카는, 폴더 내에 있고 동기화될 필요가 있는 아이тем들만 송신함으로써, 폴더의 동기화를 최적화할 수 있다.

[0068] 본 발명의 방법(500)은 또한 소스 레플리카에 있는 물리적인 레이아웃 내에서의 임의의 변경된 아이тем들을 논리적 뷰로 매핑시키는 동작{동작(508)}을 포함한다. 동작(508)은 소스 레플리카에 있는 물리적 레이아웃 내의 임의의 변경된 아이тем들을 논리적 뷰로 매핑하는 컴퓨터 시스템을 포함할 수 있다. 예를 들어, 레플리카(102)를 포함하는 컴퓨터 시스템은 물리적 테이블(105 및 106) 내에 저장된 아이тем들을 카탈로그(114)를 통해 논리적 뷰(110)로 매핑할 수 있다. 하나의 레플리카에서의 논리적 뷰는 토폴로지 내의 하나 이상의 다른 레플리카에서의 논리적 뷰들과 실질적으로 유사할 수 있다. 예를 들어, 논리적 뷰(110)는 논리적 뷰(112)와 실질적으로 유사할 수 있다. 이는 토폴로지(100)에서 레플리카들을 동기화하는 효율을 향상시킬 수 있다.

[0069] 본 발명의 방법(500)은 또한 적어도 하나의 매핑된 아이тем을 소스 레플리카로부터 목적지 레플리카로 송신하는 동작을 포함한다{동작(510)}. 동작(510)은 논리적 뷰(110)로부터 논리적 뷰(112)로 아이тем들을 전송하는 레플리카(102)를 포함할 수 있다. 예를 들어, 레플리카(102)는 연락처(252)와 유사한 연락처로부터의 하나 이상의 아이тем을 레플리카(104)로 전송할 수 있다.

- [0070] 본 발명의 방법(500)은 또한 논리적 뷰들을 커스텀 뷰들에 매핑하는 단계를 포함할 수 있다. 논리적 뷰들을 커스텀 뷰들에 매핑하는 단계는 {예를 들어, 열(310) 내의} 버전들을 {예를 들어, 열(410) 내의} 커스텀 변경 계수들에 매핑하는 단계를 포함할 수 있다. 커스텀 레플리카는 커스텀 변경 계수들을 커스텀 레플리카 내의 커스텀 아이템들에 할당할 수 있다. 커스텀 변경 계수들은 {도 4에 도시된 커스텀 뷰(404)와 같은} 버전들과 상호 관련될 수 있다. 커스텀 변경 계수들은 지역 변경들과는 다른 포맷(예를 들어, 다른 데이터 포맷)일 수 있다.
- [0071] 그 후 상술한 바와 같이, 커스텀 뷰를 통해 아이템들을 전송하고 수신하는 추가의 동작과 더불어 동기화가 발생할 수 있다. 예를 들어, 아이템이 레플리카로부터 커스텀 레플리카로 송신되는 경우에, 아이템은 송신 전에 적절한 커스텀 변경 계수를 할당받을 수 있다. 아이템들을 수신하는 경우에, 커스텀 뷰(404)를 참조함으로써, 적절한 버전이 할당되고 아이템이 {도 1에 도시된 논리적 뷰(110)와 같은} 논리적 뷰 내의 적절한 위치로 매핑될 수 있다.
- [0072] 이제 도 6을 참조하면, 논리적 뷰로 데이터 저장 장소에서의 아이템의 물리적 레이아웃의 매핑을 생성하기 위한 방법이 방법(600)으로 도시된다. 방법(600)은 논리적 스키마의 컴파일을 용이하기 위하여 사용될 수 있는 코어 코드에 액세스하는 동작을 포함한다(동작(602)). 동작(602)은 논리적 스키마의 컴파일을 용이하기 위하여 사용될 수 있는 코어 코드에 액세스하는 (예를 들어 컴퓨터 시스템에서의) 레플리카를 포함할 수 있다. 예를 들어, 레플리카(102)은 물리적 스키마 매핑으로의 논리적 스키마의 컴파일을 용이하게 하는 코어 코드에 액세스할 수 있다. 동작(602)은, 적절한 기능을 갖는 코드를 만들고 이 코드를 컴파일링 컴퓨터 시스템에 설치하는 개발자에 의하여, 본 발명의 일 실시예에서 구현될 수 있다. 그 후, 컴파일링 컴퓨터 시스템은 코드에 액세스한다.
- [0073] 방법(600)은 논리적 스키마에 액세스하는 동작을 더 포함한다(동작(604)). 동작(604)은 논리적 스키마에 액세스하는 (예를 들어 컴퓨터 시스템에서의) 레플리카를 포함할 수 있다. 예를 들어, 레플리카(102)는 논리적 스키마에 액세스할 수 있다. 본 발명의 일 실시예에서, 논리적 스키마는 아이템의 세분성을 정의하는 변경 유닛을 포함한다. 논리적 스키마는 또한 정의된 세분성의 하나 이상의 아이템을 정의하는 일관성 유닛을 포함하는데, 이들 아이템들의 변경은, 상기 하나 이상의 아이템 중의 임의의 아이템이 다른 컴퓨터 시스템에서 업데이트 되도록 하기 위하여 다른 컴퓨터 시스템에 수신되어야 한다.
- [0074] 본 발명의 일 실시예에서, 스키마는 설계시 개발자에 의하여 설계 된다. 특정 타입의 데이터 저장 장소를 설계하는 개발자는 변경 유닛을 정의하여 특정 어플리케이션에 대하여 세분성을 최적화할 수 있다. 예를 들어, 변경 유닛은 더 큰 변경 유닛을 정의함으로써, 유지되어야 할 필요가 있는 메타데이터의 양을 감소시키도록 정의되거나, 더 작은 변경 유닛을 정의함으로써 충돌 발생을 감소시키도록 설계되거나, 또는 상기 두 가지를 절충할 수 있다. 변경 유닛 세분성은 논리적 스키마에 정의될 수 있으므로, 개발자는 변경 유닛을 정의하기 위하여 새로운 컴퓨터 실행 가능 코드를 작성할 필요가 없다.
- [0075] 일관성 유닛을 정의하는 것은 설계시 개발자에 의하여 또한 수행될 수 있다. 일관성 유닛을 정의하는 것은 도 2a 및 2b의 설명과 관련하여 앞서 보다 상세히 설명되었다. 일관성 유닛을 정의하는 것은 함께 갱신되어야 하는 아이템들이 함께 갱신된다는 것을 보증하는데 도움이 된다. 일관성 유닛은 논리적 스키마에서 정의될 수 있으므로, 개발자는 일관성 유닛을 정의하기 위하여 새로운 컴퓨터 실행 가능 코드를 작성할 필요가 없다.
- [0076] 방법(600)은 논리적 스키마를 카탈로그로 컴파일하는 코어 코드를 사용하는 동작을 더 포함한다(동작(606)). 동작(606)은 논리적 스키마를 카탈로그로 컴파일하는 코어 코드를 사용하는 (예를 들어 컴퓨터 시스템에서의) 레플리카를 포함한다. 예를 들어, 레플리카(102)은 액세스된 코드를 사용하여 액세스된 논리적 스키마를 카탈로그(114)로 컴파일할 수 있다. 논리적 스키마를 물리적 스키마 매핑으로 컴파일함으로써, {도 1의 레플리카(102)와 같은} 레플리카의 {도 1의 계층(128)과 같은} 동기화 계층에 저장될 수 있는 {도 1에 도시된 카탈로그(114)와 같은} 카탈로그를 생성할 수 있다.
- [0077] 카탈로그는 {도 1의 물리적 테이블(105 및 106)과 같은} 물리적 테이블로부터 아이템을 {도 1의 논리적 뷰(110)와 같은} 논리적 뷰로 매핑하기 위한 정보를 포함한다. 논리적 뷰는 실질적으로 토폴로지에서의 하나 이상의 다른 컴퓨터에서 유사할 수 있다. 동작(606)은 또한 프로시저 또는 함수 코드를 생성할 수 있다. 프로시저 또는 함수 코드는 {도 1에 도시된 레플리카(102)에서의 데이터 저장 계층(126)과 같은} 레플리카의 데이터 저장 계층에 위치한다. 프로시저 또는 함수 코드는 {도 1에서의 테이블(105 및 106)과 같은} 물리적 테이블에서의 아이템의 물리적 저장 공간을 지시하는 컴퓨터 실행 가능 명령어를 포함한다.
- [0078] 방법(600)은 또한 논리적 뷰를 통하여 토폴로지 내의 다른 컴퓨터 시스템으로 아이템을 송신하도록 컴파일링 컴퓨터 시스템을 설정하는 단계를 포함할 수 있다. 토폴로지 내의 다른 컴퓨터 시스템은 컴퓨터 시스템에서의 논

리적 뷰와 실질적으로 유사한 논리적 뷰를 갖는다. 논리적 뷰를 통하여 아이টে를 송신하는 예는 도 1의 설명에서 보다 상세하게 논의되었다.

[0079] 방법(600)은 또한 물리적 레이아웃에 설정된 데이터 저장 장소에서 사용하기 위하여 컴퓨터 시스템에 카탈로그를 설치할 수 있다. 데이터 저장 장소는 컴퓨터 시스템에 설치되는 카탈로그보다 먼저 컴퓨터 시스템에 설치되거나, 시스템에 카탈로그가 설치된 후에 설치되거나 또는 동시에 설치될 수 있다. 설치하는 네트워크를 통하거나, 컴퓨터 디스크 또는 CD-ROM과 같은 휴대용 저장 매체 상에 카탈로그를 물리적으로 위치시키고 컴퓨터 시스템에 운반하는 것을 포함하는 다양한 방식으로 행해질 수 있다.

[0080] 방법(600)은 또한 논리적 뷰에 따라 데이터를 동기화하도록 설정된 레플리카에서 사용하기 위하여 컴퓨터 시스템에 카탈로그를 송신할 수 있다. 레플리카는 카탈로그 송신에 앞서, 또는 카탈로그 송신 후, 또는 송신과 동시에 설정될 수 있다.

[0081] 컴퓨터 시스템은 데이터 저장 계층 및 동기화 계층으로 나누어 질 수 있다. 데이터 저장 계층은 상술한 도 1의 데이터 저장 계층(126)과 유사할 수 있다. 동기화 계층은 상술한 동기화 계층(128)과 유사할 수 있다.

[0082] 방법(600)은 또한 컴퓨터 시스템의 {도 1의 계층(126)과 같은} 데이터 저장 계층에 지역 변경 트랙커를 저장할 수 있다. 지역 변경 트랙커는 소스 레플리카에 저장된 아이টে에 대한 지역 변경 계수를 유지한다. 지역 변경 트랙커의 예시적인 실시예는 도 3a 및 3b에 도시되어 있다. 방법(600)은 또한 {도 1의 계층(128)과 같은} 동기화 계층에서 동기화 변경 트랙커를 저장할 수 있다. 동기화 변경 트랙커는 {도 3a 및 3b의 열(310)에 도시된 것과 같은} 버전 및 {도 3a 및 3b의 열(308)에 도시된 것과 같은} 동기화 지역 변경 계수를 유지한다. 버전은 토폴로지에서의 컴퓨터 시스템에 대응하는 레플리카 ID 및 변경이 이루어진 연대 순에 대응하는 변경 계수를 포함할 수 있다. 소정의 실시예에 있어서, 레플리카 ID는 버전에 연관된 변경을 행한 컴퓨터 시스템에 대응한다. 다른 실시예에서는, 레플리카 ID가 버전을 할당하는 컴퓨터 시스템에 대응되도록, 하나의 컴퓨터 시스템이 여러 컴퓨터 시스템에 대하여 버전을 할당할 수 있다.

[0083] 지역 변경 계수 및 동기화 지역 변경 계수는, 변경된 아이টে들이 송신되어야 하며, 이에 따라 물리적 레이아웃에서 논리적 뷰로 매핑되어야 하는지 여부를 식별하는데 사용될 수 있다. {도 1에서 도시된 사용자 인터페이스(134)와 같은} 사용자 인터페이스를 통해 변경이 행해지는 경우에, 지역 변경 계수가 그 변경에 할당된다. 따라서, {도 1에서의 계층(126)과 같은} 데이터 저장 계층에서의 아이টে는 {도 1에서의 계층(128)과 같은} 동기화 계층에서의 아이টে에 연관된 동기화 지역 변경 계수와는 다른 지역 변경 계수를 가질 것이다. 그 후, 컴퓨터 시스템은 사용자 인터페이스를 통해 변경된 아이টে이 유효한 버전 번호를 할당받아 다른 컴퓨터 시스템과 동기화될 필요가 있다는 것을 동기화 중에 인식할 수 있다. 이러한 프로세스는 도 3a와 도 3b와 관련하여 앞서 더욱 상세하게 논의되었다.

[0084] 방법(600)은 또한 폴더가 아이টে들을 다같이 그룹핑할 수 있는 데이터 저장 계층에 폴더를 저장하는 단계를 포함한다. 그러한 폴더의 예는 폴더(120)로서 도 1에 도시된다. 폴더를 저장하는 단계는 아이টে들을 폴더와 상호 관련시키기 위해 {도 1에서의 테이블(118)과 같은} 테이블을 사용하는 단계를 포함할 수 있다. 아이টে는 폴더를 사용하여 다른 컴퓨터 시스템에 송신될 수 있다. 이러한 방법으로, 컴퓨터 시스템에 존재할 수 있는 모든 아이টে들이 아니라 선택된 아이টে들만이 송신된다.

[0085] 본 발명의 범위 내의 실시예들은 또한 저장된 컴퓨터 실행 가능 명령어 또는 데이터 구조를 갖거나 수록하기 위한 컴퓨터 판독 가능 매체를 포함한다. 이러한 컴퓨터 판독 가능 매체는 범용 또는 특수 목적용 컴퓨터가 액세스할 수 있는 임의의 가용 매체가 될 수 있다. 제한이 아닌 예로서, 이러한 컴퓨터 판독 가능 매체는 RAM, ROM, EEPROM, CD-ROM이나 다른 광 디스크 저장 장치, 자기 디스크 저장 장치나 다른 자기 저장 장치, 또는 컴퓨터 실행 가능 명령어나 데이터 구조의 형태로 원하는 프로그램 코드 수단을 수록하거나 저장하는데 사용되며 범용 컴퓨터나 특수 목적용 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 정보가 네트워크나 다른 통신 접속(유선이나 무선, 또는 유/무선의 조합)을 통하여 컴퓨터에 전송되거나 제공되는 경우에, 컴퓨터는 당연히 그 접속을 컴퓨터 판독 가능 매체로 간주한다. 따라서, 임의의 이러한 접속은 당연히 컴퓨터 판독 가능 매체라고 한다. 상기한 것들의 조합도 또한 컴퓨터 판독 가능 매체의 범위 내에 포함되어야 한다. 컴퓨터 실행 가능 명령어는, 예컨대 범용 컴퓨터, 특수 목적용 컴퓨터, 또는 특수 목적용 처리 장치로 하여금 소정의 기능이나 한 그룹의 기능을 수행하도록 하는 명령어와 데이터를 포함한다.

[0086] 도 7 및 이하의 논의는 본 발명이 구현될 수 있는 적합한 컴퓨팅 환경에 대한 간략하고 일반적인 설명을 제공하고자 의도된다. 필수적이지는 않더라도, 본 발명은, 네트워크 환경에서 컴퓨터에 의해 실행되는, 프로그램 모

들과 같은 컴퓨터 실행 가능 명령어의 일반적인 문맥에서 설명될 것이다. 일반적으로, 프로그램 모듈은 특정 작업을 수행하거나 특정 추상 데이터 타입을 구현하는 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등을 포함한다. 컴퓨터 실행 가능 명령어, 관련 데이터 구조 및 프로그램 모듈은 본 명세서에서 개시된 방법의 단계를 수행하기 위한 프로그램 코드 수단의 예를 나타낸다. 이러한 실행 가능 명령어나 관련 데이터 구조의 특정 시퀀스는 이러한 단계에서 기술되는 기능들을 구현하기 위한 대응 동작들의 예를 나타낸다.

[0087] 당해 기술 분야의 당업자는, 본 발명이 개인용 컴퓨터, 핸드 헬드(hand-held) 장치, 멀티프로세서 시스템, 마이크로프로세서 기반 또는 프로그램 가능 소비자 전자 제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터 등을 포함하는 다수 유형의 컴퓨터 시스템 구성을 구비한 네트워크 컴퓨팅 환경에서 실시될 수 있다는 점을 이해할 것이다. 본 발명은 또한 통신 네트워크를 통하여 (유선 링크나 무선 링크에 의해, 또는 유/무선 링크의 조합에 의해) 링크된 로컬 및 원격 처리 장치에 의해 작업이 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다. 분산 컴퓨팅 환경에서는, 프로그램 모듈이 로컬 및 원격 메모리 저장 장치 모두에 위치할 수 있다.

[0088] 도 7을 참조하면, 본 발명을 구현하기 위한 예시적인 시스템은, 처리 유닛(721), 시스템 메모리(722), 및 시스템 메모리(722)를 포함하는 다수의 시스템 컴포넌트를 처리 유닛(721)에 결합하는 시스템 버스(723)를 포함하는 종래의 컴퓨터(720)의 형태로 범용 컴퓨팅 장치를 포함한다. 시스템 버스(723)는 메모리 버스나 메모리 컨트롤러, 주변 장치 버스, 및 임의의 다수의 버스 아키텍처를 사용하는 로컬 버스를 포함하는 몇 개의 유형의 버스 구조 중의 임의의 것이 될 수 있다. 시스템 메모리는 ROM(724) 및 RAM(725)을 포함한다. 시동 중과 같은 때에 컴퓨터(720) 내의 구성 요소 간의 정보를 전달하는데 도움이 되는 기본 입/출력 시스템(BIOS; 726)은 ROM(724)에 저장될 수 있다.

[0089] 컴퓨터(720)는 또한 자기 하드디스크(739)로부터 판독하고 이에 기록하기 위한 자기 하드디스크 드라이브(727), 착탈식 자기 디스크(729)로부터 판독하거나 이에 기록하기 위한 자기 디스크 드라이브(728), CD-ROM이나 다른 광 매체와 같은 착탈식 광 디스크(731)로부터 판독하거나 이에 기록하기 위한 광 디스크 드라이브(730)를 포함할 수도 있다. 자기 하드디스크 드라이브(727), 자기 디스크 드라이브(728), 및 광 디스크 드라이브(730)는 하드디스크 드라이브 인터페이스(732), 자기 디스크 드라이브 인터페이스(733), 및 광 드라이브 인터페이스(734) 각각에 의해 시스템 버스(723)에 접속된다. 드라이브 및 관련 컴퓨터 판독 가능 매체는 컴퓨터 실행 가능 명령어, 데이터 구조, 프로그램 모듈 및 컴퓨터(720)에 대한 다른 데이터의 비휘발성 저장 장소를 제공한다. 본 명세서에서 설명된 예시적인 환경이 자기 하드디스크(739), 착탈식 자기 디스크(729) 및 착탈식 광 디스크(731)를 사용하더라도, 자기 카세트, 플래시 메모리 카드, DVD, 베르누이 카트리지, RAM, ROM 등을 포함하는, 데이터를 저장하기 위한 다른 유형의 컴퓨터 판독 가능 매체가 사용될 수 있다.

[0090] 운영 체제(735), 하나 이상의 애플리케이션 프로그램(736), 다른 프로그램 모듈(737), 및 프로그램 데이터(738)를 포함하는 하나 이상의 프로그램 모듈을 포함하는 프로그램 코드 수단은 하드디스크(739), 자기 디스크(729), 광 디스크(731), ROM(724) 또는 RAM(725)에 저장될 수 있다. 사용자는 키보드(740), 포인팅 장치(742), 또는 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너 등과 같은 다른 입력 장치(도시되지 않음)를 통해 컴퓨터(720)에 명령 및 정보를 입력할 수 있다. 이들 입력 장치 및 다른 입력 장치는 대개 시스템 버스(723)에 결합된 직렬 포트 인터페이스(46)를 통해 처리 유닛(721)에 접속된다. 대안적으로, 입력 장치는 병렬 포트, 게임포트 또는 USB와 같은 다른 인터페이스에 의해 접속될 수 있다. 모니터(747) 또는 다른 디스플레이 장치도 비디오 어댑터(748)와 같은 인터페이스를 통해 시스템 버스(723)에 접속된다. 모니터 외에, 개인용 컴퓨터는 전형적으로 스피커와 프린터와 같은 다른 주변 출력 장치(도시되지 않음)를 포함한다.

[0091] 컴퓨터(720)는 원격 컴퓨터(783 및 793)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 이용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(783 및 792)는 각각 다른 개인용 컴퓨터, 서버, 라우터, 네트워크 PC, 피어(peer) 장치 또는 다른 공통 네트워크 노드일 수 있으며, 전형적으로 컴퓨터(720)와 관련되어 상술된 구성 요소 중의 다수 또는 모두를 포함한다. 도 7에 도시된 논리적 접속은 예로서 본 명세서에서 제시된 지역망(LAN; 751) 및 광역망(WAN; 752)을 포함하는데 이에 한정되지 않는다. 이러한 네트워킹 환경은 사무실 광역 또는 기업 광역 컴퓨터 네트워크, 인트라넷 및 인터넷에서 일반적인 것이다.

[0092] LAN 네트워킹 환경에서 사용되는 경우에, 컴퓨터(720)는 네트워크 인터페이스 또는 어댑터(753)를 통하여 지역망(751)에 접속된다. WAN 네트워킹 환경에서 사용되는 경우에, 컴퓨터(720)는 모뎀(754), 무선 링크, 또는 인터넷과 같은, 광역망(752)을 통해 통신을 설정하기 위한 그 밖의 수단을 포함할 수 있다. 내장형이나 외장형일 수 있는 모뎀(754)은 직렬 포트 인터페이스(746)를 통해 시스템 버스(723)에 접속된다. 네트워킹화된 환경에서, 컴퓨터(720)와 관련되어 기술된 프로그램 모듈 또는 그것의 부분은 원격 메모리 저장 장치에 저장될

수 있다. 도시된 네트워크 접속은 예시적인 것이며, 광역망(752)을 통해 통신을 설정하는 다른 수단이 사용될 수 있다는 점이 이해될 것이다.

[0093] 본 발명은, 본 발명의 사상 또는 필수적인 특성을 벗어나지 않고 다른 특정한 형태로 구현될 수 있다. 설명된 실시예들은 모든 점에 있어서 단지 예시적이며 제한적이지 않은 것으로 간주되어야 한다. 따라서, 본 발명의 범위는 전술된 설명에 의해서라기 보다는 오히려 첨부된 청구범위에 의해 지시된다. 청구범위의 균등의 의미와 범위 내에서 발생한 모든 변경은 청구범위의 범위 내에 포함되어야 한다.

발명의 효과

[0094] 본 발명에 따라 논리적 뷰를 통하여 레플리카의 아이템들을 동기화할 수 있다. 레플리카는 물리적 레이아웃에서 논리적 뷰로 아이템을 매핑하기 위하여 카탈로그를 이용한다. 카탈로그는 상이한 물리적 레이아웃에서 실질적으로 유사한 논리적 뷰로의 데이터의 매핑을 용이하게 하는 논리적 스키마로부터 컴파일될 수 있다.

도면의 간단한 설명

[0001] 도 1은 본 발명의 원리에 따라 동기화될 수 있는 레플리카를 포함하는 예시적인 컴퓨터 아키텍처를 나타낸 도면.

[0002] 도 2a는 본 발명의 원리에 따른 일관성 유닛 및 변경 유닛을 정의하기 위한 예시적인 구조를 나타낸 도면.

[0003] 도 2b는 본 발명의 원리에 따른 일관성 유닛 및 변경 유닛을 정의하기 위한 예시적인 구조를 나타낸 도면.

[0004] 도 3a는 본 발명의 원리에 따라 아이템, 변경 계수 및 버전을 저장하는데 유용한 예시적인 테이블을 나타낸 도면.

[0005] 도 3b는 본 발명의 원리에 따라 아이템, 변경 계수 및 버전을 저장하는데 유용한 예시적인 테이블을 나타낸 도면.

[0006] 도 4는 본 발명의 원리에 따라 논리적 뷰를 커스텀 뷰로 매핑하기 위한 예시적인 테이블을 나타낸 도면.

[0007] 도 5는 본 발명의 소정의 실시예의 형태를 이용하여 레플리카를 동기화하기 위한 예시적인 방법을 나타낸 도면.

[0008] 도 6은 본 발명의 소정의 실시예의 형태를 이용하여 데이터 저장 장소에 있는 아이템의 물리적 레이아웃에서 논리적 뷰에서의 아이템으로의 매핑을 생성하기 위한 예시적인 방법을 나타낸 도면.

[0009] 도 7은 본 발명의 원리를 위해 적합한 운영 환경을 나타낸 도면.

[0010] <도면의 주요 부분에 대한 부호의 설명>

[0011] 102, 104 : 레플리카

[0012] 114, 116 : 카탈로그

[0013] 118, 122 : 폴더

[0014] 128, 132 : 동기화 계층

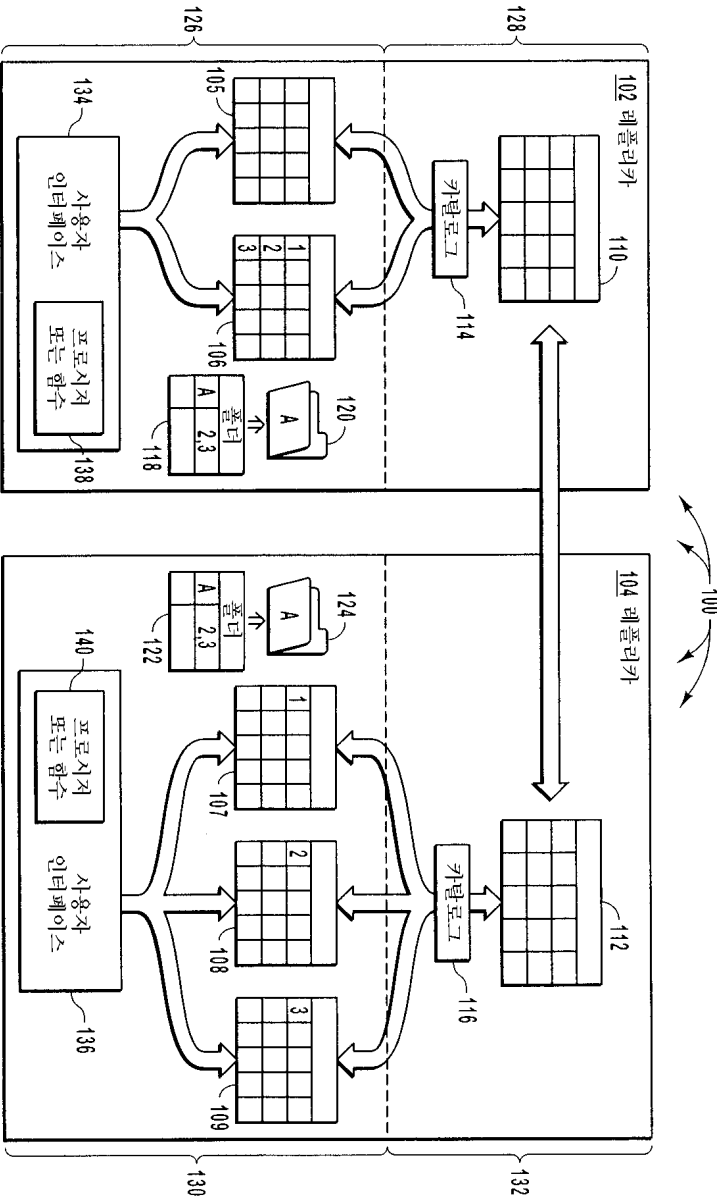
[0015] 126, 130 : 데이터 저장 계층

[0016] 134, 136 : 사용자 인터페이스

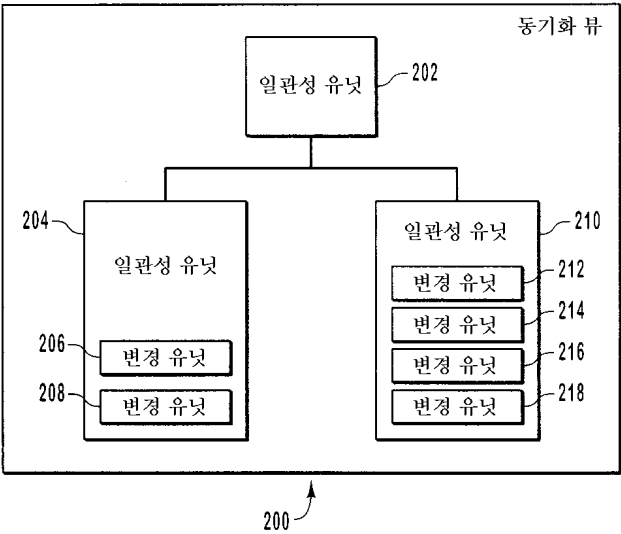
[0017] 138, 140 : 프로시저 또는 함수

도면

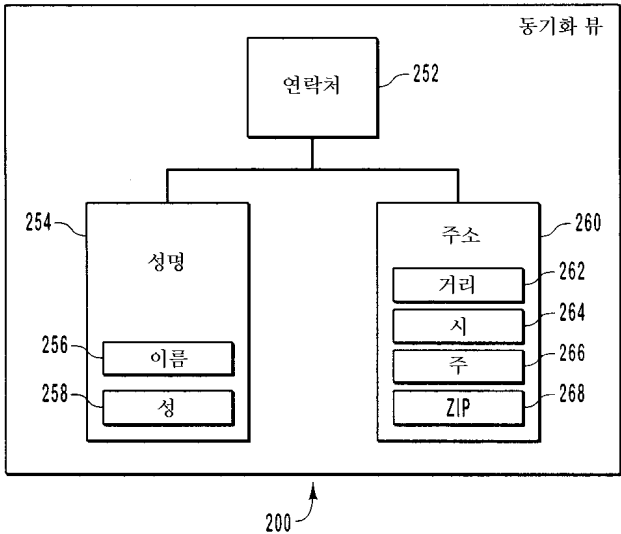
도면1

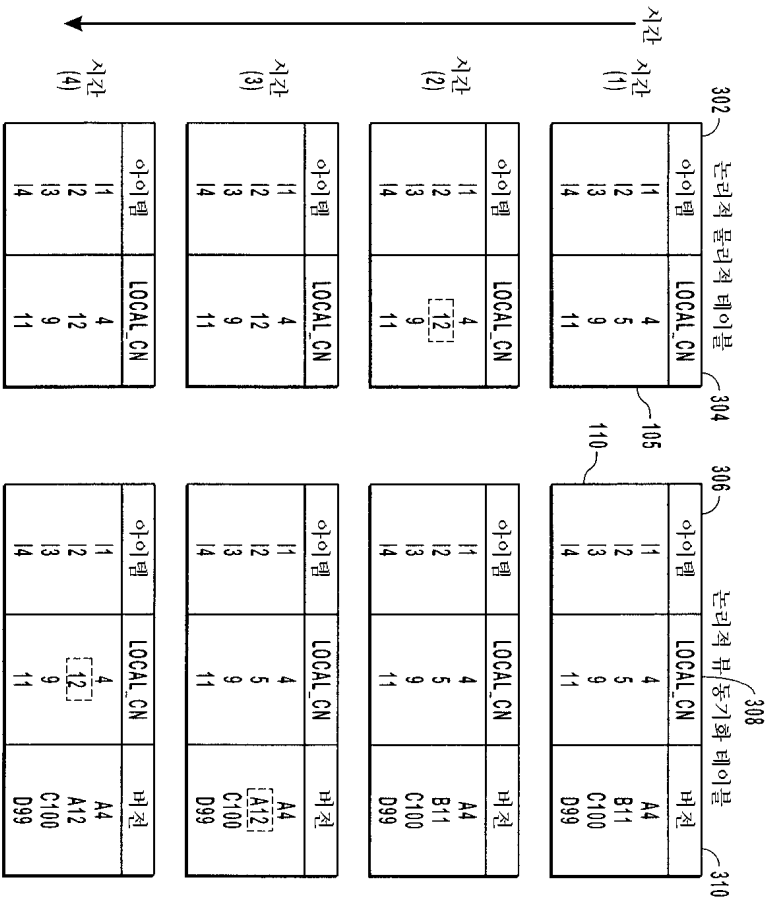


도면2a



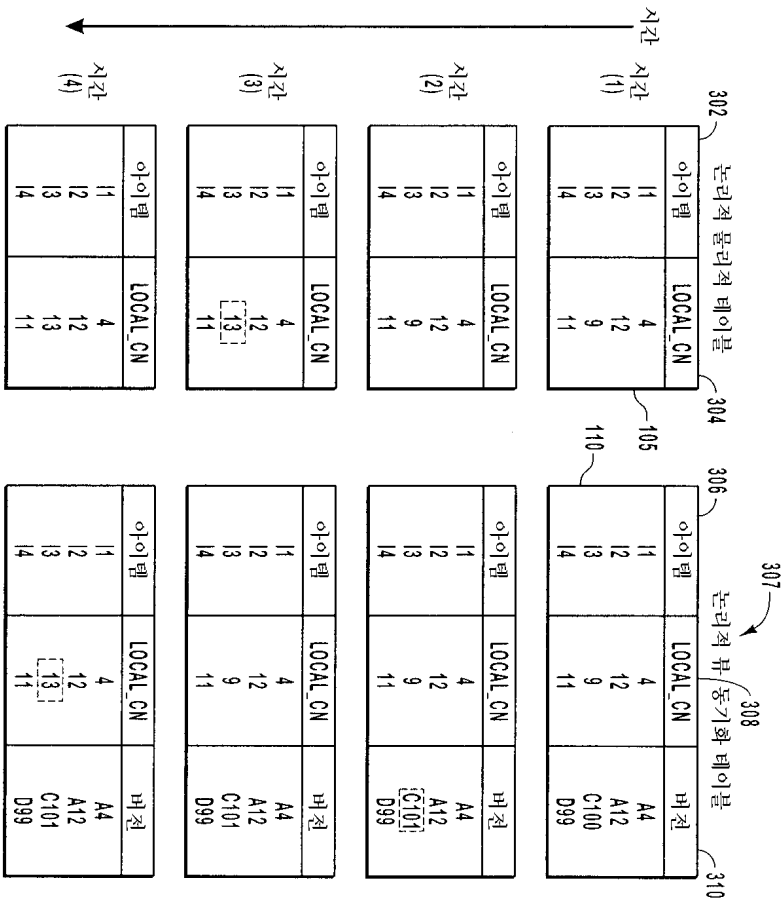
도면2b



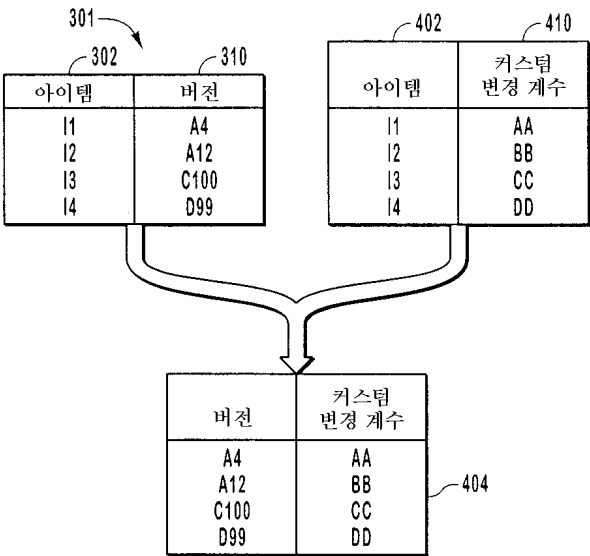


도면3a

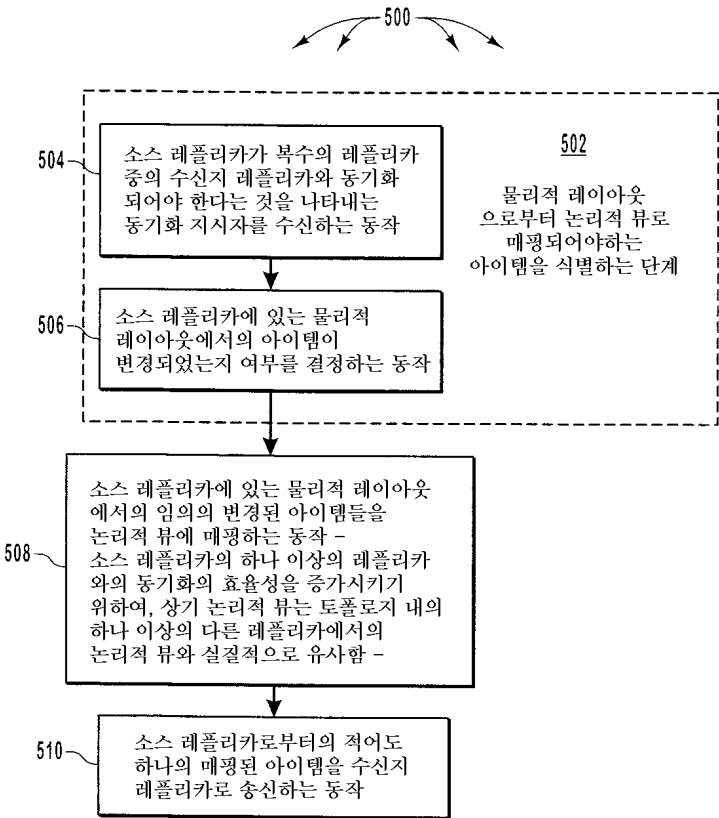
도면3b



도면4



도면5



도면6

