



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년06월25일
(11) 등록번호 10-1278785
(24) 등록일자 2013년06월19일

- (51) 국제특허분류(Int. C1.)
G06F 17/40 (2006.01) *G06F 17/30* (2006.01)
G06Q 50/30 (2012.01)
- (21) 출원번호 10-2006-0011131
(22) 출원일자 2006년02월06일
심사청구일자 2011년01월31일
(65) 공개번호 10-2006-0103828
(43) 공개일자 2006년10월04일
(30) 우선권주장
11/091,079 2005년03월28일 미국(US)
(56) 선행기술조사문헌
US20030212705 A1*
Typing a multi-language intermediate code,
Andrew D. Gordon, POPL '01 Proceedings of the
28th ACM SIGPLAN-SIGACT symposium on
Principles of programming languages (2001)*
*는 심사관에 의하여 인용된 문헌
- (73) 특허권자
마이크로소프트 코포레이션
미국 위싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
(72) 발명자
슈클라, 아밋
미국 98052 위싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내
노리, 아닐 쿠마
미국 98052 위싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내
(뒷면에 계속)
- (74) 대리인
제일특허법인

전체 청구항 수 : 총 14 항

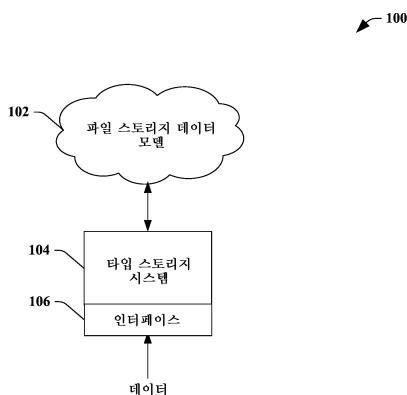
심사관 : 안지현

(54) 발명의 명칭 파일 시스템 모델의 데이터베이스 객체로의 맵핑을 용이하게 하는 시스템 및 방법

(57) 요약

본 발명은 데이터베이스 모델을 데이터베이스 객체로 맵핑하는 것을 용이하게 하는 시스템 및/또는 방법을 제공한다. 타입 스토리지 시스템은 파일 스토리지 데이터 모델의 스토리지 맵핑을 이용할 수 있다. 맵핑은 스키마에 적어도 부분적으로 기초하여 생성된 데이터베이스 객체, 및 스키마 내에 기술된 타입의 인스턴스가 저장되고 액세스되는 방법을 기술할 수 있다. 또한, 적어도 하나의 기준을 만족시키는 항목, 문서, 및/또는 콘택 중 적어도 하나를 발견하기 위한 질의가 제공될 수 있다. 타입 스토리지 시스템은 스키마, 데이터 모델, 타입, 질의 및 질의 기준 중 적어도 하나인 데이터를 인터페이스를 통해 수신하여 저장 및 질의를 제공할 수 있다. 또한, 타입 스토리지 시스템은 적어도 하나의 타입 인스턴스를 노출시키는 뷰를 생성할 수 있다.

대 표 도 - 도1



(72) 발명자

테미로스키, 베컴

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

프리드만, 그레고리 에스.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

현터, 제이슨 티.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

피어스, 제프리 티.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

뉴먼, 마이클 제이

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

엘리스, 나이젤 알.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

아차리아, 스리니바스무시 피.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트
웨이마이크로소프트 코포레이션 내

특허청구의 범위

청구항 1

데이터 모델의 맵핑을 용이하게 하는 시스템- 상기 시스템은 컴퓨터 판독가능 매체에 기록되고 컴퓨터에 의해 실행될 수 있음 -으로서,

맵핑 스키마를 정의하는 데 사용되고 정보의 저장, 발견 및 연관을 가능하게 하는 파일 스토리지 데이터 모델;

관련된 상기 맵핑 스키마에 기초하여 스키마 객체를 데이터베이스 객체로 맵핑하는 탑입 스토리지 컴포넌트 - 상기 맵핑 스키마 내의 탑입의 인스턴스가 저장되고, 상기 탑입 스토리지 컴포넌트는, 각각의 기준을 만족시키는 상기 시스템 내의 적어도 하나의 항목, 각각의 기준을 만족시키는 상기 시스템 내의 적어도 하나의 문서, 각각의 기준을 만족시키는 적어도 하나의 콘택(contact) 중 적어도 하나를 발견하기 위한 질의(query)를 더 제공하고, 상기 콘택은 개인, 조직 및 그룹 중 적어도 하나를 포함하며,

상기 스키마 내의 상기 탑입은 상기 파일 스토리지 데이터 모델의 스토리지 내의 공통 언어 실행 시간 (CLR: common language runtime) 클래스로 맵핑되고, 상기 탑입은 항목, 항목 확장, 항목 프래그먼트, 및 링크 중 적어도 하나임 - ;

1) 상기 각각의 탑입의 적어도 하나의 인스턴스를 포함하는 열 및 2) 상기 탑입의 인스턴스를 표현하는 상기 CLR 클래스 인스턴스의 직렬화된 표현(serialized representation)을 포함하는 행 중 적어도 하나를 구비한 테이블 - 상기 항목 확장, 항목 프래그먼트, 및 상기 링크 탑입의 인스턴스는 유사한 테이블 구조로 표현되고 인라인(inline) 탑입의 인스턴스는 개별 테이블 및 열에 저장되는 것이 아니라 부모 객체 인스턴스(parent object instance) 내부에 저장됨 - ; 및

특정 컨택스트 또는 사용자의 액션을 추론하고 데이터와 이벤트에 대한 고려에 기초하여 대상 상태들(states of interest)에 대한 확률 분포를 생성하는 것을 통해 저장되어야 하는 사용자 정의 탑입을 결정하는 지능 컴포넌트

를 포함하는, 데이터 맵핑을 용이하게 하는 시스템.

청구항 2

제1항에 있어서, 상기 탑입의 인스턴스는 문서, 이미지, 음악, 비디오, 콘택, 메시지, 사람, 조직, 이메일, 팩스, 음성 및 오디오 클립 중 적어도 하나인, 데이터 맵핑을 용이하게 하는 시스템.

청구항 3

제1항에 있어서, 관계 스토리지 및 관계 질의 능력(capability) 중 적어도 하나를 이용하는 관계 컴포넌트를 더 포함하는 데이터 맵핑을 용이하게 하는 시스템.

청구항 4

제3항에 있어서, 상기 관계 컴포넌트는 형식적으로 기술된 한 세트의 테이블로서 구성된 데이터 항목들의 집합인 관계형 데이터베이스 기술을 이용하며, 데이터는 재구성 없이 액세스되는 것과 재구성 없이 재결합되는 것 중 적어도 하나일 수 있는, 데이터 맵핑을 용이하게 하는 시스템.

청구항 5

제1항에 있어서, 상기 맵핑은 상기 스키마에 기초하여 생성된 상기 데이터베이스 객체, 및 상기 스키마에 기술된 상기 탑입의 인스턴스가 저장되고 액세스되는 방법 중 적어도 하나를 기술하는, 데이터 맵핑을 용이하게 하는 시스템.

청구항 6

제1항에 있어서, 상기 탑입의 적어도 하나의 인스턴스를 노출시키는 뷰(view)를 생성하는 뷰 컴포넌트를 더 포함하고, 상기 뷰는 베이스 탑입과 연관된 각각의 뷰 탑입의 서브 세트를 투영하는, 데이터 맵핑을 용이하게 하는 시스템.

청구항 7

제1항에 있어서, 상기 타입은 타입 계층 구조 및 상속성 중 적어도 하나를 포함하는, 데이터 맵핑을 용이하게 하는 시스템.

청구항 8

데이터 모델의 맵핑을 용이하게 하는 컴퓨터 구현 방법으로서,

맵핑 스키마를 정의하는 데 사용되고 정보의 저장, 발견 및 연관을 가능하게 하는 파일 스토리지 데이터 모델을 취득하는 단계;

스키마, 타입, 기준, 및 질의 정보 중 적어도 하나를 수신하는 단계;

상기 파일 스토리지 데이터 모델에 연관된 상기 스키마에 기초하여 스키마 객체를 데이터베이스 객체로 맵핑하는 단계;

상기 스키마로부터 상기 타입의 인스턴스를 저장하는 단계;

상기 스키마 내의 상기 타입을 상기 파일 스토리지 데이터 모델의 스토리지 내의 공통 언어 실행 시간(CLR) 클래스로 맵핑하는 단계 - 상기 타입은 항목, 항목 확장, 항목 프래그먼트, 및 링크 중 적어도 하나임 -;

상기 각각의 타입의 적어도 하나의 인스턴스를 포함하는 열, 및 상기 타입의 인스턴스를 표현하는 상기 CLR 클래스 인스턴스의 직렬화된 표현을 포함하는 행 중 적어도 하나를 구비한 테이블을 제공하는 단계 - 상기 항목 확장, 항목 프래그먼트, 및 상기 링크 타입의 인스턴스는 유사한 테이블 구조로 표현되고, 인라인(inline) 타입의 인스턴스는 개별 테이블 및 열에 저장되는 것이 아니라 부모 객체 인스턴스 내부에 저장됨 - ; 및

특정 컨텍스트 또는 사용자의 액션을 추론하고 데이터와 이벤트에 대한 고려에 기초하여 대상 상태들에 대한 확률 분포를 생성하는 것을 통해 저장되어야 하는 사용자 정의 타입을 결정하는 지능 컴포넌트를 제공하는 단계

를 포함하는 데이터 모델의 맵핑을 용이하게 하는 컴퓨터 구현 방법.

청구항 9

제8항에 있어서, 각각의 기준을 만족시키는 시스템 내의 적어도 하나의 항목, 각각의 기준을 만족시키는 상기 시스템 내의 적어도 하나의 문서, 및 각각의 기준을 만족시키는 적어도 하나의 콘택 중 적어도 하나를 발견하기 위해 질의하는 단계를 더 포함하는 데이터 모델의 맵핑을 용이하게 하는 컴퓨터 구현 방법.

청구항 10

제8항에 있어서, 관계형 데이터베이스 엔진을 이용하여 관계 스토리지 및 관계 질의 능력을 제공하는 단계를 더 포함하는 데이터 모델의 맵핑을 용이하게 하는 컴퓨터 구현 방법.

청구항 11

제8항에 있어서, 상기 타입의 적어도 하나의 인스턴스를 노출시키는 뷰를 생성하는 단계를 더 포함하고, 상기 뷰는 베이스 타입과 연관된 각각의 뷰 타입의 서브 세트를 투영하는, 데이터 모델의 맵핑을 용이하게 하는 컴퓨터 구현 방법.

청구항 12

제8항의 방법을 실행하는 명령어(instructions)를 저장한 컴퓨터 판독가능 기록 매체.

청구항 13

데이터 모델의 맵핑을 용이하게 하는 시스템으로서, 상기 시스템은 컴퓨터 판독가능 매체에 기록되고 컴퓨터에 의해 실행될 수 있으며,

스키마, 타입, 기준, 및 질의 기준 중 적어도 하나를 수신하기 위한 수단;

파일 스토리지 데이터 모델을 사용하는 정보의 저장, 발견 및 연관을 가능하게 하는 맵핑 스키마를 정의하기 위한 수단;

파일 스토리지 데이터 모델과 관련된 스키마에 기초하여 스키마 객체를 데이터베이스 객체로 맵핑하기 위한 수단 - 상기 스키마 내의 타입의 인스턴스가 저장되고, 상기 맵핑하기 위한 수단은 각각의 기준을 만족시키는 상기 시스템 내의 적어도 하나의 항목, 각각의 기준을 만족시키는 상기 시스템 내의 적어도 하나의 문서, 및 각각의 기준을 만족시키는 적어도 하나의 콘택 중 적어도 하나를 발견하기 위한 질의를 더 제공하고 상기 콘택은 개인, 조직, 및 그룹 중 적어도 하나를 포함함 - ;

상기 스키마 내의 상기 타입을 상기 파일 스토리지 데이터 모델의 스토리지 내의 공통 언어 실행 시간(CLR) 클래스로 맵핑하기 위한 수단 - 상기 타입은 항목, 항목 확장, 항목 프래그먼트, 및 링크 중 적어도 하나임 - ; 및

상기 각각의 타입의 적어도 하나의 인스턴스를 포함하는 열, 및 상기 타입의 인스턴스를 표현하는 상기 CLR 클래스 인스턴스의 직렬화된 표현을 포함하는 행 중 적어도 하나를 구비한 테이블을 제공하기 위한 수단 - 상기 항목 확장, 항목 프래그먼트, 및 상기 링크 타입의 인스턴스는 유사한 테이블 구조로 표현되고, 인라인 타입의 인스턴스는 개별 테이블 및 열에 저장되는 것이 아니라 부모 객체 인스턴스 내부에 저장됨 - ; 및

특정 컨텍스트 또는 사용자의 액션을 추론하고 데이터와 이벤트에 대한 고려에 기초하여 대상 상태들에 대한 확률 분포를 생성하는 것을 통해 저장되어야 하는 사용자 정의 타입을 결정하는 지능 컴포넌트를 제공하기 위한 수단

을 포함하는 데이터 모델의 맵핑을 용이하게 하는 시스템.

청구항 14

제6항에 있어서, 상기 뷰 컴포넌트는 사용자와 상기 타입 스토리지 컴포넌트 사이에서 상호작용하는 사용자 인터페이스를 호출하는, 데이터 모델의 맵핑을 용이하게 하는 시스템.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0038] 본 발명은 일반적으로 데이터베이스에 관한 것으로서, 보다 구체적으로는 타입 인스턴스의 저장 및/또는 데이터 질의를 용이하게 하는 시스템 및/또는 방법에 관한 것이다.

- [0039] 일반적으로 컴퓨터 기술의 진보(예를 들어, 마이크로프로세서 속도, 메모리 용량, 데이터 전송 대역폭, 소프트웨어 기능 등)는 여러 산업에서의 향상된 컴퓨터 응용에 기여해 왔다. 서버들의 어레이로서 종종 구성되는 훨씬 더 강력한 서버 시스템이 일반적으로 제공되어, 예를 들어 월드 와이드 웹과 같은 외부 소스로부터 발생하는 요구에 서비스하고 있다.
- [0040] 이용 가능한 전자 데이터의 양에 따라, 사용자에게 친숙하고 빠른 데이터 검색 및 수취를 용이하게 하는 관리 가능한 방식으로 데이터를 저장하는 것이 더욱 중요해지고 있다. 오늘날, 일반적인 접근법은 전자 데이터를 하나 이상의 데이터베이스에 저장하는 것이다. 일반적으로, 전형적인 데이터베이스는 예를 들어 컴퓨터 프로그램이 원하는 데이터를 빠르게 검색하고 선택할 수 있도록 데이터를 구축한 체계화된 정보 집합이라고 말할 수 있다. 일반적으로, 데이터베이스 내의 데이터는 하나 이상의 테이블을 통해 체계화된다. 테이블은 행 및 열로 배열될 수 있다.
- [0041] 테이블은 하나 이상의 레코드를 포함할 수 있고, 레코드는 한 세트의 필드를 포함할 수 있다. 레코드는 일반적으로 테이블 내의 행으로서 인덱스되며, 레코드 필드는 일반적으로 열로서 인덱스되어, 인덱스들의 행/열 쌍이 테이블 내의 특정 데이터를 참조할 수 있게 된다. 예를 들어, 행은 세일즈 트랜잭션, 사람 또는 프로젝트에 관한 완전한 데이터 레코드를 저장할 수 있다. 마찬가지로, 테이블의 열은 동일한 일반 데이터 포맷을 가진 행들의 개별 부분들을 정의할 수 있으며, 여기서 열은 레코드의 필드를 정의할 수 있다.
- [0042] 일반적으로, 각각의 개별 데이터는 독립적으로는 별로 유익하지 못하다. 데이터베이스 애플리케이션은 사용자가 데이터를 체계화하고 처리하는 것을 돋기 때문에 데이터를 더 유용하게 한다. 데이터베이스 애플리케이션은 사용자가 데이터를 비교, 정렬, 순서화, 병합, 분리 및 상호연결하는 것을 가능하게 하여, 데이터로부터 유용한 정보가 생성될 수 있게 한다. 지금도, 데이터베이스의 성능 및 다기능성은 믿을 수 없을 정도로 계속 성장하여, 가상적으로 끝없는 데이터베이스를 이용한 저장 능력을 가능하게 하고 있다. 더욱이, 전형적인 데이터베이스 시스템은 시간, 파일 확장, 위치 및 크기에 기초한 제한된 질의 능력을 제공한다. 예를 들어, 데이터베이스와 연관된 막대한 양의 데이터를 검색하기 위해서 전형적인 검색은 파일명, 파일 크기, 생성 일자로 제한되고 있는데, 이러한 기술은 부족하고 부적절하다.
- [0043] 최종 사용자들로부터의 계속적이고 증가하는 데이터의 생성과 더불어, 데이터를 발견하고, 관련시키고, 저장하는 것을 둘러싼 문제 및 어려움이 극에 달하고 있다. 최종 사용자들은 문서를 기록하고, 사진을 저장하고, 컴팩트 디스크로부터 음악을 립(rip)하고, 이메일을 수신하고, 전송된 이메일의 사본을 유지하는 등등을 행한다. 예를 들어, 음악 컴팩트 디스크를 생성하는 간단한 프로세스에서 최종 사용자는 메가 바이트의 데이터를 생성할 수 있다. 컴팩트 디스크로부터 음악을 런핑하고, 파일을 적절한 포맷으로 변환하고, 주얼 케이스 커버를 생성하고, 컴팩트 디스크 라벨을 설계하는 작업 모두는 데이터의 생성을 필요로 한다.
- [0044] 사용자를 둘러싼 문제뿐만 아니라, 개발자들도 데이터와 관련된 유사한 문제를 갖고 있다. 개발자들은 개인적인 애플리케이션에서 고도로 개발된 기업 애플리케이션에 이르는 수많은 애플리케이션을 생성하고 작성한다. 생성 및/또는 개발 동안, 개발자들은 항상은 아니지만 빈번하게 데이터를 수집한다. 데이터를 수집하면서 데이터를 저장할 필요가 있다. 즉, 데이터를 발견하고, 관련시키고, 저장하는 것을 둘러싼 문제 및 어려움이 개발자 및 최종 사용자 양자에게 영향을 미친다. 이러한 관점에서, 종래의 시스템 및 데이터베이스와 연관된 결함을 완화하는 시스템 및/또는 방법을 제공하고/거나 개량할 필요성이 있다.

발명이 이루고자 하는 기술적 과제

- [0045] 다음은 본 발명의 몇몇 양태의 기본적인 이해를 제공하기 위하여 본 발명의 개요를 설명한다. 이 개요는 본 발명의 외연적인 개요는 아니다. 본 발명의 주요 또는 결정적인 요소들을 식별하거나 본 발명의 범위를 정하고자 하는 의도는 없다. 그 유일한 목적은 후술하는 상세한 설명에 대한 서론으로서 본 발명의 몇몇 개념들을 간단한 형태로 설명하고자 하는 것이다.
- [0046] 본 발명은 데이터 모델의 데이터베이스 객체로의 맵핑을 용이하게 하는 시스템 및/또는 방법에 관한 것이다. 타입 스토리지 시스템은 타입 인스턴스의 저장 및/또는 질의를 제공할 수 있다. 타입의 저장은 데이터베이스 객체, 및 타입 인스턴스가 저장 및/또는 액세스되는 방법을 기술할 수 있다. 질의는 각각 소정의 기준을 만족시키는 항목, 문서, 및 콘택 중 적어도 하나를 발견할 수 있다. 더욱이, 타입 스토리지 시스템은 인터페이스를 이용하여 데이터를 수신할 수 있는데, 데이터는 스키마, 타입, 기준, 질의 기준 등을 포함할 수 있다.
- [0047] 본 발명의 일 양태에 따르면, 타입 스토리지 시스템은 적어도 하나의 타입 인스턴스를 저장하는 스토어 컴포넌트를 포함할 수 있다. 스토리지는 데이터 모델의 맵핑일 수 있으며, 데이터 모델은 파일 스토리지 시스템을 표

현할 수 있다. 또한, 타입 스토리지 시스템은 질의 컴포넌트도 포함할 수 있다. 질의 컴포넌트는 기준을 만족시키기 위하여 파일 스토리지 시스템 데이터 모델 내에 적어도 하나의 질의를 제공할 수 있다. 본 발명의 다른 양태에 따르면, 타입 스토리지 시스템은 관계 스토리지 및/또는 관계 질의 능력을 이용할 수 있는 관계 컴포넌트를 포함할 수 있다. 관계 컴포넌트는 데이터베이스 엔진을 호출하여 관계 기술을 제공할 수 있으며, 이러한 기술은 타입 인스턴스의 저장 및/또는 질의를 용이하게 할 수 있다.

[0048] 더욱이, 타입 스토리지 시스템은 주어진 타입의 모든 인스턴스를 노출시키는 뷰를 제공할 수 있는 뷰 컴포넌트를 더 포함할 수 있다. 타입은 계층 구조 및/또는 상속 구조일 수 있다. 뷰는 타입들의 서브 세트를 베이스 타입으로 투영하는 특정 타입과 연관될 수 있다. 본 발명의 다른 양태들에서, 데이터 모델의 데이터베이스 객체로의 맵핑을 용이하게 하는 방법이 제공된다.

[0049] 아래의 설명 및 첨부 도면은 본 발명의 소정의 예시적인 양태들을 상세히 설명한다. 그러나, 이들 양태는 단지 본 발명의 원리들이 이용될 수 있는 몇몇 다양한 방법을 나타내는 것이며, 본 발명은 이러한 모든 양태 및 이들의 균등물을 포함하는 것으로 의도된다. 본 발명의 다른 이점 및 신규 특징들은 도면과 관련하여 고려될 때 아래의 발명의 상세한 설명으로부터 명확해질 것이다.

발명의 구성 및 작용

[0050] 본 명세서에 사용되는 용어들, "컴포넌트", "시스템", "인터페이스", "스키마" 등은 컴퓨터 관련 엔티티, 하드웨어, 소프트웨어(예를 들어, 실행중인) 및/또는 펌웨어를 지칭하고자 한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 프로세스, 프로세서, 객체, 실행 가능물, 프로그램 및/또는 컴퓨터일 수 있다. 예를 들어, 서버 상에서 실행되는 애플리케이션 및 서버는 컴포넌트일 수 있다. 하나 이상의 컴포넌트가 프로세스 내에 있을 수 있으며, 하나의 컴포넌트가 하나의 컴퓨터 상에 로컬화되거나 둘 이상의 컴퓨터 사이에 분산될 수 있다.

[0051] 본 발명은 도면들을 참조하여 설명되는데, 도면 전체에서 동일 참조 번호는 동일 요소를 지칭하도록 사용된다. 아래의 설명에서는 설명의 목적으로 본 발명의 철저한 이해를 제공하기 위해 다양한 특정 상세가 설명된다. 그러나, 본 발명은 이러한 특정 상세 없이도 실시될 수 있다는 것은 명백할 수 있다. 다른 예에서는 본 발명의 설명을 용이하게 하기 위하여 공지된 구조 및 장치들이 블록도의 형태로 도시된다.

[0052] 이제, 도면을 참조하면, 도 1은 데이터 모델과 연관된 타입 인스턴스의 저장을 용이하게 하는 시스템(100)을 나타낸다. 데이터 모델은 정보의 저장, 발견 및 연관을 가능하게 하는 파일 스토리지 데이터 모델(102)일 수 있다. 예를 들어, 정보 타입은 문서, 이미지, 비디오, 콘택, 메시지, 이메일, 오디오 클립 등일 수 있지만 이에 한하지 않는다. 정보 타입은 상속을 지원하는 타입 시스템의 일부인 복합 타입들의 인스턴스들로서 표현될 수 있는 정보의 단위들로 간주될 수 있다. 상속은 소정의 특성이 하나의 컨텍스트에서 다른 컨텍스트로 전달되는 상황으로서 정의될 수 있다. 객체 지향 프로그래밍에서 특히, 객체는 다른 객체로부터 속성 및/또는 거동을 상속한다. 상속은 계층 구조 및/또는 포맷으로 간주될 수 있다는 것을 이해해야 한다.

[0053] 파일 스토리지 시스템(104)은 타입 인스턴스를 저장하고, 항목, 문서 및/또는 콘택 중 적어도 하나를 효과적으로 그리고 효율적으로 발견하기 위해 질의할 수 있다. 타입 스토리지 시스템(104)은 데이터를 수신할 수 있으며, 데이터는 타입, 기준, 스키마, 질의 기준 등을 포함할 수 있다. 구체적으로, 정보 타입(예를 들어, 문서, 이미지, 비디오, 콘택, 메시지, 이메일, 오디오 클립)에 관한 타입 인스턴스의 저장은 1) 소정 기준을 만족시키는 파일 스토리지 데이터 모델(102) 내의 적어도 하나의 항목의 발견, 2) 소정 기준을 만족시키는 파일 스토리지 데이터 모델(102) 내의 적어도 하나의 문서의 발견, 및 3) 소정 기준을 만족시키는 적어도 하나의 콘택(예를 들어, 개인, 조직 및 그룹을 포함)의 발견 중 적어도 하나를 제공할 수 있다.

[0054] 타입 스토리지 시스템(104)은 관계 스토리지 및 관계 질의와 연관된 관계형 데이터베이스 기술을 이용할 수 있다. 이러한 능력은 데이터베이스 엔진(도시되지 않음)에 의해 제공될 수 있다는 것을 이해해야 한다. 관계형 데이터베이스 기술은 관계형 데이터베이스와 연관될 수 있는데, 관계형 데이터베이스는 형식적으로 기술되는 한 세트의 테이블로서 체계화된 데이터 항목들의 집합이다. 테이블 내의 데이터는 데이터베이스 테이블의 재구성 없이 다양한 방식으로 액세스 및/또는 재결합될 수 있다. 또한, 관계형 데이터베이스는 기존 애플리케이션 및/또는 데이터에 대한 수정 없이 새로운 카테고리의 추가와 같이 쉽게 확장될 수 있다. 본 발명은 관계형 데이터베이스 및/또는 관련 기술로 한정되지 않으며, 임의의 적절한 기술이 이용될 수 있다는 것을 이해해야 한다.

[0055] 시스템(100)은 타입 스토리지 시스템(104)을 가상적으로 임의의 운영 체제로 통합하기 위해 다양한 어댑터, 커넥터, 채널, 통신 경로 등을 제공하는 인터페이스 컴포넌트(106)를 더 포함한다. 또한, 인터페이스 컴포넌트(106)는 데이터 및 타입 스토리지 시스템(104)과의 상호작용을 제공하는 다양한 어댑터, 커넥터, 채널 통신 경

로 등을 제공할 수 있다. 인터페이스 컴포넌트(106)가 타입 스토리지 시스템(104) 내에 포함되지만 이러한 구현은 제한적인 것은 아니라는 것을 이해해야 한다. 예를 들어, 인터페이스 컴포넌트(106)는 시스템(100)과 관련된 데이터를 송수신할 수 있는 독립 컴포넌트일 수 있다.

[0056] 도 2는 타입 인스턴스의 저장 및/또는 항목, 문서 및 데이터 모델(202)과 연관된 콘택을 발견하기 위한 질의를 용이하게 하는 시스템(200)을 나타낸다. 데이터 모델(202)은 계층적 특성 및/또는 상속성을 이용하는 파일 스토리지 시스템의 모델 표현일 수 있다. 타입은 문서, 이미지, 비디오, 콘택, 메시지, 이메일, 오디오 클립 등을 포함할 수 있다. 그러나, 타입은 데이터 모델(202)에 의해 표현되는 시스템 내에 저장된 전형적인 정보 타입일 수 있다는 것을 이해해야 한다. 타입 스토리지 시스템(204)은 타입 인스턴스를 저장할 수 있으며, 데이터 모델(202) 내의 항목, 데이터 모델(202) 내의 문서, 및 데이터 모델(202) 내의 콘택 중 적어도 하나를 효율적으로 발견할 수 있는 질의를 제공할 수 있다. 타입 스토리지 시스템(204)은 데이터를 수신할 수 있는데, 여기서 데이터는 질의 기준, 스키마, 기준, 스키마 정의, 데이터 모델, 타입 등일 수 있다. 시스템(200)은 파일 스토리지 시스템(204)을 가상적으로 임의의 운영 체제로 통합하기 위한 다양한 어댑터, 커넥터, 채널, 통신 경로 등의 제공을 용이하게 하고 통신을 용이하게 하기 위해 인터페이스(206)를 사용할 수도 있다.

[0057] 타입 스토리지 시스템(204)은 타입 인스턴스를 저장할 수 있는 스토어 컴포넌트("스토어(208)")를 더 포함할 수 있다. 스토리지는 데이터 모델(202)로의 맵핑일 수 있는데, 스토리지 맵핑은 스키마 정의에 기초하여 생성되는 데이터베이스 객체 및 스키마에 기술된 타입 인스턴스가 저장 및/또는 액세스되는 방법을 기술할 수 있다. 즉, 스토어(208)는 타입 인스턴스 및 타입 선언의 데이터베이스 객체로의 맵핑과 연관된 적어도 하나의 규칙을 저장할 수 있다.

[0058] 타입 스토리지 시스템(204)은 데이터 질의를 제공하는 질의 컴포넌트(210)(("질의(210)")를 포함할 수 있다. 질의(210)는 데이터 모델(202)에 의해 표현되는 시스템 내의 항목, 데이터 모델(202)에 의해 표현되는 시스템 내의 문서, 및 데이터 모델(202)에 의해 표현되는 시스템 내의 콘택(예를 들어, 개인, 조직 및 그룹을 포함) 중 적어도 하나를 취득할 수 있다. 질의는 적어도 인터페이스(206)를 통해 얻어지는 소정의 기준 및/또는 질의 기준에 기초할 수 있다는 것을 이해해야 한다. 또한, 질의(210)는 항목, 문서 및 콘택에 한정되는 것이 아니라, 데이터 모델(202)에 의해 표현되는 시스템 내에 저장된 임의의 적절한 정보 타입이 이용될 수 있다.

[0059] 도 3은 관계 스토리지 및 관계 질의 능력의 이용을 용이하게 하는 시스템(300)을 나타낸다. 타입 스토리지 시스템(304)은 타입 인스턴스를 저장하고 항목, 문서 및 콘택을 효율적으로 그리고 효과적으로 발견하기 위해 질의 할 수 있는데, 이러한 요소들은 데이터 모델(302)에 의해 표현되는 파일 시스템과 연관된다. 파일 시스템은 계층 구조 및/또는 상속성을 이용하는 스토리지 파일 시스템일 수 있다는 것을 이해해야 한다. 타입 스토리지 시스템(304)은 도 1 및 2에 각각 도시된 타입 스토리지 시스템(104, 204)과 거의 유사할 수 있다. 타입 스토리지 시스템(304)은 본 발명에 따라 이용될 수 있는 통신 및/또는 데이터 수신을 용이하게 하기 위해 인터페이스(306)를 호출할 수 있다.

[0060] 타입 스토리지 시스템(304)은 스토어 컴포넌트(308)(스토어(308)) 및 질의 컴포넌트(질의(310))를 포함할 수 있다. 스토어(308)는 타입 인스턴스를 저장하기 위한 임의의 적절한 저장 기술을 제공할 수 있다. 질의(310)는 기준을 만족시키는 시스템 내의 적어도 하나의 항목, 기준을 만족시키는 시스템 내의 적어도 하나의 문서, 및 기준을 만족시키는 시스템 내의 적어도 하나의 콘택 중 적어도 하나를 효율적으로 그리고 효과적으로 취득할 수 있는 질의를 제공할 수 있다. 스토어(308) 및 질의(310)는 도 2에 도시된 스토어(208) 및 질의(210)와 거의 유사할 수 있다는 것을 이해해야 한다.

[0061] 타입 스토리지 시스템(304)은 관계 컴포넌트("관계(312)")를 더 포함할 수 있다. 관계(312)는 데이터베이스 기술을 이용하여(예를 들어, 데이터베이스 엔진을 이용하여) 관계 스토리지를 구축하고, 그리고/또는 타입 인스턴스의 저장 및/또는 질의를 용이하게 하기 위한 관계 질의 능력을 제공한다. 관계(312)는 관계형 데이터베이스와 연관된 기술을 포함할 수 있는데, 여기서 관계형 데이터베이스는 전술한 바와 같이 형식적으로 기술된 한 세트의 테이블로서 체계화된 데이터 항목들의 집합이다. 본 발명은 관계형 데이터베이스 및 관련 기술로 한정되는 것이 아니라 임의의 적절한 기술이 이용될 수 있다는 것을 이해해야 한다.

[0062] 도 4는 데이터 모델(402)과 관련한 맵핑 및/또는 뷰잉을 용이하게 하는 시스템(400)을 나타낸다. 데이터 모델(402)은 정보의 저장, 발견 및 연관을 가능하게 하는 파일 스토리지 시스템을 표현할 수 있다. 시스템에 저장될 수 있는 전형적인 정보 타입은 문서, 이미지, 음악, 비디오, 콘택, 메시지 등을 포함할 수 있다. 정보 타입은 상속을 지원하는 타입 시스템의 일부인 복합 타입들의 인스턴스들로서 표현될 수 있다. 타입 스토리지 시스템(404)은 타입 인스턴스를 제공하고 항목, 문서, 및 소정의 기준을 만족시키는 콘택을 발견하기 위한 질의를

제공할 수 있다. 시스템(400)은 또한 인터페이스(406)를 이용하여, 통신 및/또는 기준, 타입, 스키마, 모델 및 질의 기준을 포함할 수 있는 데이터의 수신을 용이하게 할 수 있다.

[0063] 타입 스토리지 시스템(104)은 타입 인스턴스의 저장을 용이하게 하기 위한 맵 컴포넌트(408)를 더 포함할 수 있다. 맵 컴포넌트(408)는 스키마에 기술된 타입들의 정의된 타입들 및 데이터베이스 객체들로의 맵핑을 제공한다. 맵 컴포넌트(408)는 스키마 정의에 기초하여 생성될 수 있는 적어도 하나의 데이터베이스 객체 및/또는 스키마에 기술된 타입 인스턴스가 저장 및/또는 액세스되는 방법을 기술하는 스토리지 맵핑일 수 있다. 즉, 타입 인스턴스는 저장될 수 있으며, 규칙들이 이용되어 타입 선언을 데이터베이스 객체로 맵핑할 수 있다. 스키마 내의 각각의 타입은 스토리지 내의 클래스(예를 들어, 공통 언어 실행 시간(CLR))로 맵핑된다.

[0064] 타입 스토리지 시스템(404)은 뷰 투영을 제공하는 뷰 컴포넌트(410)를 포함할 수 있다. 뷰 투영은 보여질 타입 인스턴스를 노출시킬 수 있다. 타입은 적어도 상속성을 이용하는 계층 구조 내에 있을 수 있다. 즉, 각각의 타입은 타입 계층 구조의 일부이다. 뷰는 주어진 타입과 연관될 수 있으며, 그의 베이스 타입과 연관된 뷰의 각각의 타입의 서브 세트를 투영할 수 있다. 뷰는 특정 타입과 연관된 인스턴스를 투영할 수 있다. 예를 들어, 타입 "메시지"에 대해, 어느 메시지가 부모인지에 대한 인스턴스만이 적어도 계층 구조에 기초하여 보여질 수 있다.

[0065] 뷰 컴포넌트(410)는 사용자와 타입 스토리지 시스템(404)에 결합된 임의의 컴포넌트 사이의 상호작용을 용이하게 하기 위한 다양한 타입의 인터페이스를 제공할 수 있다. 도시된 바와 같이, 뷰 컴포넌트(410)는 타입 스토리지 시스템(404)으로 통합된 개별 엔티티이다. 그러나, 뷰 컴포넌트(410) 및/또는 유사한 뷰 컴포넌트는 타입 스토리지 시스템(404)과 별개의 컴포넌트 및/또는 독립 유닛일 수 있다는 것을 이해해야 한다. 뷰 컴포넌트(410)는 하나 이상의 그래픽 사용자 인터페이스(GUI), 명령 라인 인터페이스 등을 제공할 수 있다. 예를 들어, 사용자에게 데이터의 적재, 수입, 판독 등을 위한 영역 또는 수단을 제공하는 GUI가 제공될 있으며, 그 결과를 제공하기 위한 영역을 포함할 수 있다. 이들 영역은 공지된 텍스트 및/또는 대화 상자, 정적 제어, 드롭 다운 메뉴, 리스트 상자, 팝업 메뉴, 편집 제어, 콤보 상자, 무선 버튼, 체크 상자, 부시 버튼 및 그래픽 상자를 포함하는 그래픽 영역을 포함할 수 있다. 또한, 네비게이션을 위한 수직 및/또는 수평 스크롤 바 및 영역이 볼 수 있는 것인지를 판정하기 위한 툴바 버튼과 같이 프리젠테이션을 용이하게 하기 위한 유ти리티들이 사용될 수 있다. 예를 들어, 사용자는 타입 스토리지 시스템(404)에 결합된 컴포넌트들 중 하나 이상과 상호작용할 수 있다.

[0066] 사용자는 또한 영역들과 상호작용하여, 예를 들어 마우스, 롤러 볼, 키패드, 키보드, 웹 및/또는 음성 동작과 같은 다양한 장치를 선택하고 정보를 제공할 수 있다. 일반적으로, 푸시 버튼 또는 키보드 상의 입력 키와 같은 메카니즘은 검색을 개시하기 위한 정보의 계속적인 입력에 이용될 수 있다. 그러나, 본 발명은 이에 한정되지 않는다는 것을 이해해야 한다. 예를 들어, 단지 체크 상자를 하이라이팅하여 정보 전송을 개시할 수 있다. 다른 예에서, 명령 라인 인터페이스가 사용될 수 있다. 예를 들어, 명령 라인 인터페이스는 텍스트 메시지의 제공을 통해 사용자에게 정보를 재촉할 수 있다(예를 들어, 디스플레이 상의 텍스트 메시지 및 오디오 톤을 통해). 그러면, 사용자는 인터페이스 프롬프트 내에 제공되는 옵션에 대응하는 영수자 입력과 같은 적절한 정보, 또는 프롬프트에서 제시된 질문에 대한 응답을 제공할 수 있다. 명령 라인 인터페이스는 GUI 및/또는 API와 결합하여 이용될 수 있다는 것을 이해해야 한다. 또한, 명령 라인 인터페이스는 제한된 그래픽 지원 및/또는 저대역폭 통신 채널을 가진 하드웨어(예를 들어, 비디오 카드) 및/또는 디스플레이(예를 들어, 흑백 및 EGA)와 결합하여 사용될 수 있다.

[0067] 도 5는 데이터 모델(502)과 연관된 타입 인스턴스의 저장을 용이하게 하기 위한 지능을 이용하는 시스템(500)을 나타낸다. 시스템(500)은 앞의 도면들에 도시된 컴포넌트들과 거의 유사할 수 있는 타입 스토리지 시스템(504), 인터페이스(506) 및 데이터 모델(502)을 포함한다. 인터페이스(506)는 기준, 타입, 스키마, 데이터 모델, 및 질의 기준을 포함할 수 있는 데이터와 연관된 통신을 용이하게 할 수 있다. 시스템(500)은 타입의 저장, 질의 및/또는 뷰를 제공할 수 있다. 데이터베이스 엔진이 시스템(500)에 관계 스토리지 및 관계 질의를 제공할 수 있다는 것을 이해해야 한다.

[0068] 시스템(500)은 지능 컴포넌트(508)를 더 포함할 수 있다. 지능 컴포넌트(508)는 시스템(500)에 대한 저장 및/또는 질의를 용이하게 하기 위해 타입 스토리지 시스템(504)에 의해 이용될 수 있다. 예를 들어, 지능 컴포넌트(508)는 저장될 사용자 정의 타입의 결정을 용이하게 하는 데 사용될 수 있다. 사용자 프로파일과 결합된 이력 데이터는 지능 컴포넌트(508)가 타입의 저장 및/또는 소정 기준을 이용한 질의를 결정하는 것을 가능하게 할 수 있다.

- [0069] 지능 컴퓨포넌트(508)는 이벤트 및/또는 데이터를 통해 캡쳐된 한 세트의 관측으로부터 시스템, 환경 및/또는 사용자의 상태의 추리 또는 추론을 제공할 수 있다는 것을 이해해야 한다. 추론은 특정 컨텍스트 또는 액션을 식별하는 데 이용되거나, 예를 들어 상태들에 대한 확률 분포를 생성할 수 있다. 추론은 확률, 즉 데이터 및 이벤트의 고려에 기초한 대상 상태들에 대한 확률 분포의 계산일 수 있다. 추론은 또한 한 세트의 이벤트 및/또는 데이터로부터 보다 높은 레벨의 이벤트를 구성하는 데 사용되는 기술을 말할 수 있다. 이러한 추론은 이벤트들이 시간적으로 근접하여 상관되어 있는지의 여부에 관계없이, 그리고 이벤트 및 데이터가 하나 또는 여러 이벤트 및 데이터 소스로부터 발생했는지에 관계없이 한 세트의 관측된 이벤트 및/또는 저장된 이벤트 데이터로부터 새로운 이벤트 또는 액션을 구성한다. 다양한 분류(명시적으로 그리고/또는 암시적으로 훈련됨) 스kip 및/또는 시스템(예를 들어, 지원 벡터 머신, 신경망, 전문가 시스템, 베이시안 신뢰망, 퍼지 이론, 데이터 퓨전 엔진 등)이 본 발명과 관련된 자동 및/또는 추론 액션을 수행하는 것과 관련하여 이용될 수 있다.
- [0070] 분류자는 입력 속성 벡터 $x=(x_1, x_2, x_3, x_4, x_n)$ 를 입력이 클래스에 속하는 컨피던스로 맵핑하는 함수, 즉 $f(x)=\text{confidence}(\text{class})$ 이다. 이러한 분류는 사용자가 자동으로 수행되기를 원하는 액션을 예측 또는 추론하기 위해 확률 및/또는 통계 기반 분석(예를 들어, 분석 유ти리티 및 비용으로의 인수분해)을 이용할 수 있다. 지원 벡터 머신(SVM)은 이용될 수 있는 분류자의 일례이다. SVM은 가능한 입력들의 공간에서 초과 표면을 발견함으로써 동작하는데, 이러한 초과 표면은 논-트리거링 이벤트로부터 트리거링 기준을 분할하도록 시도한다. 직관적으로, 이것은 훈련 데이터에 가깝지만 동일하지는 않은 데이터를 테스트하기 위해 분류를 수정한다. 다른 지향성 및 비지향성 모델 분류 접근법은 예를 들어 네이브 베이스, 베이시안 네트워크, 판정 트리, 신경망, 퍼지 이론 모델을 포함하며, 독립적인 상이한 패턴들을 제공하는 확률 분류 모델이 사용될 수 있다. 본 명세서에서 사용되는 분류는 또한 우선 순위의 모델을 개발하는 데 사용되는 통계 회귀를 포함한다.
- [0071] 또한, 지능 컴퓨포넌트(508)는 사용자 프로파일 및/또는 이력 데이터를 저장하기 위해 데이터 스토어(510)를 사용할 수 있다. 데이터 스토어(510)는 예를 들어 휘발성 메모리 또는 불휘발성 메모리거나, 휘발성 및 불휘발성 메모리를 포함할 수 있다. 예를 들어, 불휘발성 메모리는 ROM, PROM, EPROM, EEPROM 또는 플래시 메모리를 포함할 수 있지만 이에 한하지 않는다. 휘발성 메모리는 외부 캐시 메모리로서 동작하는 RAM을 포함할 수 있다. 예를 들어, RAM은 SRAM, DRAM, SDRAM, DDR SDRAM, ESDRAM, SDRAM, RDRAM, DRDRAM 및 RDRAM과 같은 많은 형태로 이용 가능하지만 이에 한하지 않는다. 본 시스템 및 방법의 데이터 스토어(510)는 이들 및 임의의 다른 적절한 타입의 메모리를 포함하는 것으로 의도되지만 이에 한하지 않는다. 또한, 데이터 스토어(510)는 서버 및/또는 데이터베이스일 수 있다는 것을 이해해야 한다.
- [0072] 도 6은 하이 레벨 스토어 구조를 나타낸다. 다양한 타입 인스턴스가 그들의 타입에 따라 테이블에 저장될 수 있는 스키마가 제공될 수 있다. 타입은 항목, 항목 확장, 항목 프래그먼트 및 링크일 수 있지만 이에 한하지 않는다. 각각의 타입은 객체 인스턴스를 갖는 열을 포함할 수 있는 대응 테이블을 가질 수 있다. 예를 들어, 항목 테이블 내의 하나의 열은 스토어 내의 모든 항목 인스턴스를 포함할 수 있다. 각각의 행에 대해, 열은 항목 타입 인스턴스를 표현하는 CLR 클래스 인스턴스의 직렬화된 표현을 포함할 수 있다. 항목 확장에 대해, 항목 프래그먼트 및 링크 타입 인스턴스가 유사한 구조로 표현될 수 있다는 것을 이해해야 한다. 인라인 타입 인스턴스는 개별 테이블 및/또는 열에 저장되는 것이 아니라 부모 객체 인스턴스 내부에 저장될 수 있다. 각각의 항목, 항목 확장, 항목 프래그먼트, 및 링크에 대해, 그 타입의 모든 인스턴스를 노출시키는 뷰가 생성될 수 있다. 각각의 타입은 타입 계층 구조의 일부이다. 예를 들어, 항목 뷰는 스토어 내의 모든 항목을 투영한다.
- [0073] 도 7을 참조하면, 항목(702)이 도시되어 있다. 항목(702)의 테이블은 스토어(도시되지 않음) 내의 항목들의 모든 인스턴스를 포함할 수 있다. 테이블(704)은 객체 인스턴스(706)(문서) 및 객체 인스턴스(708)(콘택)를 포함할 수 있다. 스키마 내의 각각의 타입은 스토리지 내의 CLR 클래스로 맵핑된다는 것을 이해해야 한다. 객체 인스턴스(706)는 제목, 초록, 인쇄소, 저자, 및 객체 인스턴스(706)와 연관된 다양한 다른 메타데이터를 포함할 수 있다. 유사하게, 객체 인스턴스(708)는 이름, 어드레스 이메일 등과 같은 다양한 메타데이터를 포함할 수 있다.
- [0074] 도 8은 타입 계층 구조(800) 및 대응하는 뷰 투영(820)을 나타낸다. 타입 계층 구조(800)는 항목(802), 콘택(804), 문서(806), 메시지(808), 사람(810), 조직(812), 이메일(814), 팩스(816) 및 음성(818)을 포함할 수 있다. 타입 계층 구조(800)는 일레이며, 임의의 적절한 계층 구조 및/또는 타입들이 본 발명에 따라 이용될 수 있다는 것을 이해해야 한다. 도시된 바와 같이, 항목(802)은 그 안에 모든 타입의 부모를 포함한 것으로 간주된다. 따라서, 콘택(804)은 사람(810) 및 조직(812)의 부모이고, 메시지(808)는 이메일(814), 팩스(816) 및 음성(818)의 부모이다. 또한, 대응하는 뷰 투영(820)은 타입 계층 구조를 반영할 수 있다. 주어진 타입과 연관된 뷰는 그의 베이스 타입과 연관된 뷰의 항목들의 서브 세트를 투영할 수 있다. 예를 들어, 항목 뷰는 스토

어 내의 모든 항목을 투영한다. 콘택 뷔는 콘택 타입을 가진 항목들만을 투영한다. 사람 뷔는 최다 도출 타입을 사람으로 갖는 콘택들만을 투영한다.

[0075] 도 6을 다시 참조하면, 타입 맵핑은 타입을 스토어 내의 대응 스토리지 구조를 기술하는 데 사용되는 타입으로 맵핑하기 위한 알고리즘을 이용하여 제공될 수 있다. 맵핑은 임의의 파일 스토리지 시스템(예를 들어, 정보 단위를 기술 및/또는 표현하기 위하여 복합 타입 인스턴스를 이용하는 시스템에 기초한 데이터 모델)과 연관될 수 있다는 것을 이해해야 한다. 스키마에서 선언되는 각각의 타입은 SQL UDT 콘택을 지원하는 CLR 클래스로 맵핑된다. SQL이 아래의 예에서 이용되지만 임의의 적당한 데이터베이스 관리 시스템도 이용될 수 있다는 것을 이해해야 한다. 타입들은 ".Store" 접미사가 첨부된 스키마의 명칭 공간에 대응하는 명칭을 가진 명칭 공간에 속 한다. 주어진 명칭 공간 내의 타입들은 스키마의 전개의 단위로서 이용되는 단일 어셈블리로 컴파일된다. CLR 타입의 명칭은 스키마에서 선언된 명칭과 동일하다. 선언된 각각의 타입 특성에 대해, 다음이 대응 CLR 타입에 추가된다: 1) "m_"이 접두사로 첨부된 스키마에서 지정된 명칭과 동일한 명칭을 가진 사적 필드. 이 필드는 UDT 특정 속성: System.Data.SqlTypes.SqlUdtField를 갖는 것으로 간주된다; 2) 스키마 및 대응하는 get/set 문장에서 지정되는 명칭과 동일한 명칭을 가진 공개 특성. 이 특성은 UDT 특정 속성 System.Data.SqlTypes.SqlUdtProperty를 갖는 것으로 간주된다; 및 3) 필드 및 특성의 타입은 스키마에서 선언되는 타입에 대응하는 CLR 타입이다. 타입이 스칼라 타입들 중 하나인 경우, 이 타입은 스칼라 SQL 타입들 중 하나로 맵핑된다(후술함).

[0076] 다음 테이블은 파일 스토리지 시스템 스칼라 타입들의 대응 SQL 관리 타입들로의 맵핑을 기술한다.

표 1

파일 스토리지 시스템 타입	관리 SQL 타입	설명
String	SqlString	2^31 문자의 최대 길이를 가진 가변 길이 유니코드 테이터. 길이는 1~4000 문자로 고정되거나 "max" 키워드를 이용하여 제한되지 않을 수 있다.
Binary	SqlBinary	2^32 바이트의 최대 길이를 가진 가변 길이 바이너리 데이터. 길이는 1~8000 바이트로 고정되거나, "max" 키워드를 이용하여 제한되지 않을 수 있다.
Boolean	SqlBoolean	널 가능 부울 값
Byte	SqlByte	단일 비서명 바이트
Int16	SqlInt16	-2^15(-32,768) 내지 2^15-1(32,767)의 정수 네이터
Int32	SqlInt32	-2^31(-2,147,483,648) 내지 2^31-1(2,147,483,647)의 정수(whole number) 네이터
Int64	SqlInt64	-2^63(-9223372036854775808) 내지 2^63-1(9223372036854775808)의 정수(whole number) 네이터
Single	SqlSingle	-3.40E+38 내지 3.40E+38의 부동 정수 테이터
Double	SqlDouble	-1.79E+308 내지 1.79E+308의 부동 정수 네이터
Decimal	SqlDecimal	고정 정밀도 및 스케일을 가진 수치 테이터. 정밀도 및 스케일 속성은 타입 Decimal의 특성을 지원한다. 정밀도는 10진 값의 10진 숫자들의 최대수를 정의한다. 이것은 소수점 좌우의 숫자를 포함한다. 정밀도 속성의 값은 1 내지 28의 정수이다. 스케일은 소수점 우측의 10진 숫자들의 최대수를 정의한다. 스케일 속성의 값은 0과 정밀도 속성의 값 사이에 있어야 한다. 마일스톤 B에 대한 주: 이들 속성은 이 마일스톤에서 지원되지 않는다. 정밀도는 28로 고정되며, 스케일은 14로 고정된다.
DateTime	SqlDateTime	1/300 초 또는 3.33 밀리초의 정밀도를 가진 1753년 1월 1일부터 9999년 12월 31일 까지의 날짜 및 시간 테이터.
Guid	SqlGuid	글로벌 고유 식별자(GUID)
Xml	SqlString	XML은 현재 스트링 필드로서 저장된다. 네이티브 Xml 타입 지원이 포스트 베타 1에 대해 고려되고 있다.
Stream	TBD	포스트 베타 1에 대해 계획된 지원. 지원될 경우, string(max) 및 binary(max) 선언을 대체할 수 있다.

[0077]

[0078] 파일 스토리지 시스템 스토어에서 생성된 데이터베이스 객체는 파일 스토리지 시스템 스키마 명칭으로부터 도출

된 SQL 스키마 명칭에 저장될 수 있다. 접미사 ".Store"는 파일 스토리지 시스템 스키마 명칭에 첨부되어 SQL 스키마 명칭을 생성한다. 예를 들어, 파일 스토리지 시스템 스토리지 스키마는 "[System.Storage.Store].Item"과 같은 "[System.Storage.Store]" 내의 객체를 생성한다.

[0079] 파일 스토리지 시스템 내의 내용은 뷔를 통해 액세스될 수 있다. 아래에 도시된 뷔는 판독 전용이지만, 뷔는 기입 가능할 수 있으므로 본 발명은 그에 한하지 않는다. 각각의 항목 타입은 뷔 타입으로 맵핑될 수 있다. 각각의 항목 뷔 타입은 명명 규약 [<schema name>.Store].[<Item type name>]을 이용하여 파일 스토리지 시스템에서 식별될 수 있다. 타입 T에 대한 타입 뷔는 타입 T인 모든 항목 및 T로부터 도출되는 모든 타입을 리턴 할 수 있다. System.Storage.Item 타입에 대응하는 뷔는 [System.Storage.Store].[Item]이다. 이 뷔는 파일 스토리지 시스템 스토어 내의 모든 항목을 리턴할 수 있다. 다음 테이블은 항목 타입 뷔의 열을 기술한다.

표 2

열 명칭	타입	설명
ItemId	[System.Storage.Store].ItemId	항목의 ItemId
TypeId	[System.Storage.Store].TypeId	스키마 타입 id
NamespaceName	nvarchar(255)	항목의 고유 명칭
ContainerId	[System.Storage.Store].ItemId	컨테이너 항목의 id
Item	항목 타입	항목 타입의 인스턴스
EntityState	[System.Storage.Store].EntityState	항목의 상태 정보
ObjectSize	Bigint	항목 객체의 크기(지속 열)
ChangeInformation	[System.Storage.Store].ChangeInformation	변경 트랙킹 정보
ItemSyncMetadata	[System.Storage.Store].ItemSyncMetadata	동기 메타데이터
PathHandle	[System.Storage.Store].BinPathHandle	항목으로의 경로 핸들
PromotionStatus	Int	항목에 대한 프로모션의 상태

[0080]

[0081] 각각의 링크 타입은 타입 뷔로 맵핑된다. 각각의 타입 링크 뷔는 명명 규약 [<schema name>.Store].[<link type name>]을 이용하여 파일 스토리지 시스템 스토어에서 식별된다. System.Storage.Link 타입에 대응하는 뷔는 [System.Storage.Store].[Link]이다. 이 뷔는 파일 스토리지 시스템 스토어 내의 모든 링크를 포함할 수 있다. 다음 테이블은 링크 타입 뷔의 열을 설명한다.

표 3

열 명칭	타입	설명
SourceRef	[System.Storage.Store].ItemId	소스 항목의 ItemId
LinkId	[System.Storage.Store].LinkId	링크의 Id
TargetRef	[System.Storage.Store].ItemId	타겟 항목의 ItemId
TypeId	[System.Storage.Store].TypeId	링크 인스턴스의 최종 도출 타입의 TypeId
Link	링크 타입	링크 타입의 인스턴스
EntityState	[System.Storage.Store].EntityState	링크에 대한 상태 정보
ObjectSize	Bigint	링크 객체의 크기(지속 열)
ChangeInformation	[System.Storage.Store].ChangeInformation	변경 트랙킹 정보
LinkSyncMetadata	[System.Storage.Store].ItemSyncMetadata	동기 메타데이터
PathHandle	[System.Storage.Store].BinPathHandle	링크의 소스인 항목으로의 경로 핸들

[0082]

[0083] 모든 항목 프래그먼트는 단일 뷔 [System.Storage.Store].[ItemFragment]를 통해 액세스될 수 있다. 다음 테이블은 글로벌 항목 프래그먼트 뷔의 열을 기술한다.

표 4

열 명칭	타입	설명
ItemId	[System.Storage.Store].ItemId	소유 항목의 ItemId
SetId	[System.Storage.Store].SetId	항목 프레그먼트의 유통
FragmentId	[System.Storage.Store].FragmentId	프레그먼트 인스턴스의 Id
TypeId	[System.Storage.Store].TypeId	ItemFragment 인스턴스의 최다 도출 타입의 TypeId
ItemFragment	[System.Storage.Store].[ItemFragmen t]	ItemFragment 타입의 인스턴스
EntityState	[System.Storage.Store].EntityState	링크에 대한 상태 정보
ChangeInformation	[System.Storage.Store].ChangeInform ation	변경 트랙킹 정보
PathHandle	[System.Storage.Store].BinPathHandle	링크의 소스인 항목으로의 경로 헤들

[0084]

- [0085] 각각의 스토어는 [System.Storage.Store].[ItemExtension]이라는 명칭의 글로벌 확장 뷔를 제공한다. 모든 확장 타입의 인스턴스는 이 뷔를 통해 액세스 가능하다. 다음 테이블은 글로벌 확장 뷔의 열을 기술한다.

표 5

열 명칭	타입	설명
ItemId	[System.Storage.Store].ItemId	확장의 ItemId
TypeId	[System.Storage.Store].TypeId	스키마 타입 id
ItemExtension	[System.Storage.Store].ItemExtension	확장 타입 인스턴스
EntityState	[System.Storage.Store].EntityState	확장의 상태 정보
ObjectSize	Bigint	확장 객체의 크기(지속 열)
ChangeInformation	[System.Storage.Store].ChangeInform ation	변경 트랙킹 정보
PathHandle	[System.Storage.Store].BinPathHandle	확장을 소유하는 항목으로의 경로 헤들

[0086]

- [0087] 각각의 확장 타입은 타입 뷔로 맵핑된다. 각각의 타입 확장 뷔는 명명 규약 [<schema name>.Store].[<extension type name>]를 이용하여 파일 스토리지 시스템에서 식별된다. 다음 테이블은 확장 타입 뷔의 열을 기술한다.

표 6

열 명칭	타입	설명
ItemId	[System.Storage.Store].Ite mId	확장의 ItemId
TypeId	[System.Storage.Store].Typ eId	스키마 타입 id
ItemExension	확장 타입	확장 타입 인스턴스
ObjectSize	Bigint	확장 객체의 크기(지속 열)
EntityState	[System.Storage.Store].Ent ityState	확장에 대한 상태 정보
ChangeInformation	[System.Storage.Store].Cha ngeInformation	변경 트랙킹 정보
PathHandle	[System.Storage.Store].Bin PathHandle	확장을 소유하는 항목으로의 경로 헤들

[0088]

- [0089] 파일 스토리지 시스템에서, 인라인 타입 인스턴스는 엔티티 타입 인스턴스 내에 저장될 수 있다. 이들은 적절한 검색 뷔에 질의함으로써 액세스 된다는 점을 이해해야 한다.
- [0090] 모든 항목은 [System.Storage.Store].[Table!Item]이라고 하는 단일 항목 테이블에 저장된다. 다음 테이블의 고유 키는 ItemId이다.

표 7

열 명칭	타입	설명
ItemId	[System.Storage.Store].ItemId	항목의 ItemId
TypeId	[System.Storage.Store].TypeId	스키마 타입 id
NamespaceName	Nvarchar(255)	항목의 고유 명칭
ContainerId	[System.Storage.Store].ItemId	컨테이너 항목의 id
Item	항목 테이블	항목
EntityState	[System.Storage.Store].EntityState	항목에 대한 메타 정보를 포함하는 EntityState udt
SDId	Int	내부 전용(보안 서브 시스템에 의해 사용됨)
SDLastUpdatedLocalTS	Bigint	SDId의 최종 변경의 타임 스텝 포. 변경 추적에 사용됨.
ChangeInformation	[System.Storage.Store].ChangeInformation	변경 트랙킹 정보
ItemSyncMetadata	[System.Storage.Store].ItemSyncMetadata	글로벌 id에 대한 동기 메타데이터
TombstoneStatus	Int	터미널 상태
LastUpdatedLocalTS	Bigint	내부 전용(인덱싱에 사용됨). 이 열은 실제로 ChangeInformation.LastUpdatedLocalTS로 맵핑된다. 그 위에 인덱스를 생성하기 위하여 그를 맵핑해야 한다.
MaxOrd	int	내부 전용(이 컨테이너 항목에 추가되는 새로운 항목들에 대한 PathHandle를 계산하는 데 사용됨)
TypePath	Hierarchical_Type_Id	내부 전용(지속 열)
ObjectSize	Bigint	항목 객체의 크기(지속 열)
PathHandle	[System.Storage.Store].BinPathHandle	이 항목에 대한 경로 핸들
PromotionStatus	Int	항목에 대한 프로모션 상태
LastAccessTime	DateTime	항목과 연관된 파일 스트림에 대한 최종 액세스 시간
StreamSize	Bigint	항목과 연관된 스트림의 크기
AllocationSize	Bigint	항목과 연관된 스트림에 대한 할당 크기

[0091]

[0092] 항목 테이블 상의 인덱스는 다음 테이블에 기술된다.

표 8

열	고유성	클리스터	포함 열
TypePath, ItemId	예	예	
ItemId	예	아니오	TombstoneStatus, TypePath, SDId
PathHandle	예	아니오	SDId, TombstoneStatus
ContainerId, NamespaceName	예	아니오	SDId, TombstoneStatus
LastUpdatedLocalTS	아니오	아니오	SDId, TombstoneStatus, PathHandle

[0093]

[0094] 모든 링크는 [System.Storage.Store].[Table!Link]라고 하는 링크 테이블에 저장된다. 다음 테이블의 고유 키는 ItemId, LinkId이다.

표 9

열 명칭	타입	설명
SourceRef	[System.Storage.Store].ItemId	소스 항목의 ItemId
LinkId	[System.Storage.Store].LinkId	항목의 Id
TargetRef	[System.Storage.Store].ItemId	티켓 항목의 ItemId
TypeId	[System.Storage.Store].Typeid	스키마 타입 id
TypePath	Hierarchical_Type_Id	내부 전용(지속 열)
Link	링크 타입	링크 객체
EntityState	[System.Storage.Store].EntityState	링크에 대한 상태 정보
SDId	Int	내부 전용
TombstoneStatus	Int	톰스톤 상태
Objectsize	Bigint	링크 객체의 크기(지속 열)
ChangeInformation	[System.Storage.Store].ChangeInformation	변경 트랙킹 정보
LinkSyncMetadata	[System.Storage.Store].LinkSyncMetadata	동기 메타데이터
LastUpdatedLocalTS	Bigint	내부 전용(인덱싱에 사용됨). 이 열은 실체로 ChangeInformation.LastUpdate dLocalTS로 맵핑된다. 그 위에 인덱스를 생성하기 위해 그를 꺾어야 한다.
PathHandle	[System.Storage.Store].BinPath Handle	링크의 소스인 항목으로의 경로 헤더

[0095]

[0096] 링크 테이블 상의 인덱스는 다음 테이블에서 기술된다.

표 10

열	고유성	클러스터	포함 열
SourceRef, LinkId	예	아니오	
TypePath	아니오	아니오	SDId, TombstoneStatus
SourceRef, TypePath	아니오	아니오	SDId, TombstoneStatus
PathHandle	아니오	아니오	SDId, TombstoneStatus
TargetRef	아니오	아니오	SDId, TombstoneStatus
LastUpdatedLocalTS	아니오	아니오	SDId, TombstoneStatus, PathHandle

[0097]

[0098] 모든 EntityExtension은 [System.Storage.Store].[Table!ItemExtension]이라고 하는 단일 테이블에 저장된다. 다음 테이블은 항목 확장 테이블을 기술한다.

표 11

열 명칭	타입	설명
ItemId	[System.Storage.Store].ItemId	확장의 ItemId
TypeId	[System.Storage.Store].Typeid	스키마 타입 id
TypePath	Hierarchical_Type_Id	내부 전용(지속 열)
ItemExtension	[System.Storage.Store].ItemExtension	ItemExtension 객체
EntityState	[System.Storage.Store].EntityState	EntityExtension에 대한 메타 정보를 포함하는 EntityState udt
SDId	Int	내부 전용
TombstoneStatus	Int	톰스톤 상태
ObjectSize	Bigint	EntityExtension 객체의 크기(지속 열)
ChangeInformation	[System.Storage.Store].ChangeInformation	변경 트랙킹 정보
LastUpdatedLocalTS	Bigint	내부 전용(인덱싱에 사용됨). 이 열은 실체로 ChangeInformation.LastUpdaedLocalTS로 맵핑된다. 그 위에 인덱스를 생성하기 위해 그를 꺾어야 한다.
PathHandle	[System.Storage.Store].BinPath Handle	EntityExtension을 소유하는 항목에 대한 경로 헤더

[0099]

[0100] ItemExtension에 대한 인덱스는 다음 테이블에 기술된다.

표 12

열명	고유성	클러스터	포함 열
TypePath, ItemId, TypeId	예	예	
ItemId, TypeId	아니오	아니오	SDId, TombstoneStatus, PathHandle
PathHandle	예	아니오	SDId, TombstoneStatus
LastUpdatedLocalTS	아니오	아니오	SDId, TombstoneStatus, PathHandle

[0101]

[0102] 모든 ItemFragment 타입 인스턴스는 [System.Storage.Store].[Table!ItemFragment]라고 하는 단일 테이블에 저장된다. 다음 테이블은 ItemFragment 타입을 나타낸다.

표 13

열명	타입	설명
ItemId	[System.Storage.Store].ItemId	소유 항목의 ItemId
SetId	[System.Storage.Store].SetId	항목 프래그먼트의 용도
FragmentId	[System.Storage.Store].FragmentId	프래그먼트 인스턴스의 Id
ItemFragment	[System.Storage.Store].ItemFragment	ItemFragment 객체
TypeId	[System.Storage.Store].TypeId	스키마 타입 id
TypePath	Hierarchical_Type_Id	내부 전용(지속 열)
EntityState	[System.Storage.Store].EntityState	프래그먼트에 대한 메타 정보를 포함하는 EntityState udt
SDId	int	내부 전용
TombstoneStatus	int	톱스톤 상태
ObjectSize	bigint	ItemFragment 객체의 크기(지속 열)
ChangeInformation	[System.Storage.Store].ChangeInformation	변경 트랙킹 정보
LastUpdatedLocalTS	bigint	내부 전용(인덱싱에 사용됨). 이 열은 실제로 ChangeInformation.LastUpdatedLocalTS로 맵핑된다. 그 위에 인덱스를 생성하기 위해 그를 맵핑해야 한다.
PathHandle	[System.Storage.Store].BinPathHandle	소유 항목으로의 경로 핸들

[0103]

[0104] ItemFrament 테이블 상의 인덱스는 다음 테이블에 기술된다.

표 14

열명	고유성	클러스터	포함 열
TypePath, ItemId, SetId, FragmentId	예	예	
ItemId, SetId, FragmentId	예	아니오	SDId, TombstoneStatus, PathHandle
PathHandle	예	아니오	SDId, TombstoneStatus
LastUpdatedLocalTS	아니오	아니오	SDId, TombstoneStatus, PathHandle

[0105]

[0106] 도 9 및 10은 본 발명에 따른 방법을 나타낸다. 설명의 간략화를 위해, 이 방법의 일련의 동작으로 도시되고 설명된다. 본 발명은 도시된 동작들에 의해, 그리고/또는 동작들의 순서에 의해 한정되지 않으며, 예를 들어 동작들은 다양한 순서로, 그리고/또는 동시에 일어날 수 있고, 본 명세서에서 설명되지 않은 다른 동작들을 포함할 수 있다는 것을 이해해야 한다. 또한, 본 발명에 따른 방법을 실시하기 위해 모든 도시된 동작들이 필요하지 않을 수 있다. 또한, 이 방법은 상태도 또는 이벤트를 통해 일련의 상호 관련된 상태들로서 달리 표현될 수 있다는 것을 당업자는 이해할 것이다.

[0107]

도 9는 데이터 모델로 맵핑되고, 그리고/또는 질의를 제공하는 타입 인스턴스의 저장을 용이하게 하는 방법(900)을 나타낸다. 데이터 모델은 정보의 저장, 발견 및 연관을 가능하게 하는 파일 스토리지 데이터 모델일

수 있다. 정보 타입은 문서, 이미지, 비디오, 콘택, 메시지, 이메일, 오디오 클립 등일 수 있지만 이에 한하지 않는다. 이를 정보 타입(예를 들어, 정보 단위)은 상속을 지원하는 타입 시스템의 일부인 복합 타입들의 인스턴스들로서 표현될 수 있는데, 여기서 상속은 객체들이 다른 객체들로부터 속성 및/또는 거동을 상속하는 것을 허용한다. 상속은 계층 구조 및/또는 포맷으로 간주될 수 있는 것을 이해해야 한다.

- [0108] 참조 번호(902)에서, 데이터가 수신될 수 있는데, 이 데이터는 타입, 기준, 스키마, 질의 기준 등을 포함할 수 있다. 타입 인스턴스는 참조 번호(904)에서 데이터 모델(예를 들어, 파일 스토리지 데이터 모델)로 맵핑되도록 저장될 수 있다. 예를 들어, 모델은 스키마에 기술된 타입들을 사용자 정의 타입들 및 데이터베이스 객체들로 맵핑할 수 있다. 스토리지 맵핑은 스키마 정의에 기초하여 생성되는 데이터베이스 객체들 및 스키마에 기술된 타입 인스턴스가 저장 및/또는 액세스되는 방법을 기술할 수 있다. 일례에서, 데이터베이스 구조는 타입 인스턴스 및 타입 선언의 데이터베이스 객체로의 맵핑을 위한 규칙이 제공될 수 있도록 설계될 수 있다.
- [0109] 참조 번호(906)에서, 항목, 문서 및 콘택을 효율적으로 그리고 효과적으로 발견하기 위하여 적어도 하나의 기준을 만족하도록 질의가 호출될 수 있다. 예를 들어, 정보 타입에 관련된 타입 인스턴스의 저장은 1) 파일 스토리지 데이터 모델(102) 내의 적어도 하나의 항목의 발견, 2) 파일 스토리지 데이터 모델(102) 내의 적어도 하나의 문서의 발견, 및 3) 소정 기준을 만족시키는 적어도 하나의 콘택(예를 들어, 개인, 조직 및 그룹을 포함)의 발견 중 적어도 하나를 제공하도록 질의를 이용할 수 있다.
- [0110] 도 10은 타입 인스턴스의 저장 및/또는 질의 제공을 용이하게 하는 방법(1000)을 나타낸다. 참조 번호(1002)에서, 데이터가 취득 및/또는 수신된다. 데이터는 타입, 기준, 스키마, 질의 기준 등을 포함할 수 있다. 1004에서, 데이터베이스 엔진을 이용하여 타입 인스턴스의 저장 및/또는 질의를 위한 적어도 하나의 메카니즘을 제공한다. 예를 들어, 관계형 데이터베이스 엔진을 이용하여 관계 스토리지 및 관계 질의 능력을 제공할 수 있다. 관계형 데이터베이스 엔진은 형식적으로 기술된 한 세트의 테이블로서 체계화된 데이터 항목들의 집합을 이용할 수 있다. 테이블 내의 데이터는 데이터베이스 테이블의 재구성 없이 다양한 방식으로 액세스 및/또는 재결합될 수 있다. 또한, 관계형 데이터베이스는 기존 애플리케이션 및/또는 데이터에 대한 수정 없이 새로운 카테고리를 추가하는 등 쉽게 확장될 수 있다.
- [0111] 참조 번호(1006)에서, 데이터 모델로 맵핑되는 타입 인스턴스가 저장되는데, 데이터 모델은 파일 스토리지 시스템일 수 있다. 스토리지 맵핑은 스키마에 기초하여 생성되는 데이터베이스 객체들 및 스키마에 기술된 타입 인스턴스가 저장 및/또는 액세스되는 방법을 기술한다. 맵핑은 스키마에 기술된 타입일 수 있는데, 이러한 맵핑은 사용자 정의 타입 및 데이터베이스 객체에 대한 것이다. 참조 번호(1008)에서, 항목, 문서 및 콘택 중 적어도 하나를 발견하기 위한 질의가 제공될 수 있다. 질의를 이용하여 적어도 부분적으로 기준에 기초하여 검색을 행할 수 있다. 더욱이, 참조 번호(1010)에서, 뷰를 이용하여 타입의 인스턴스들을 노출시킬 수 있다. 예를 들어, 타입 시스템은 계층 구조를 가질 수 있는데, 여기서 특정 타입의 임의의 인스턴스를 노출시키기 위한 뷰를 생성할 수 있다. 즉, 계층 구조로 인해, 뷰는 그의 베이스 타입과 연관된 뷰의 특정 타입들의 서브 세트를 투영하는 주어진 타입과 연관된다.
- [0112] 본 발명의 다양한 양태를 구현하기 위하여 추가 컨텍스트를 제공하기 위하여, 도 11, 도 12, 및 다음의 설명은 본 발명의 다양한 양태가 구현될 수 있는 적절한 컴퓨팅 환경의 간단하고 일반적인 설명을 제공하고자 한다. 본 발명은 로컬 컴퓨터 및/또는 원격 컴퓨터에서 실행되는 컴퓨터 프로그램의 컴퓨터 실행 가능 명령들의 일반적인 컨텍스트에서 전술하였지만, 본 발명은 다른 프로그램 모듈과 함께 구현될 수도 있다는 것을 당업자는 이해할 것이다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하고 그리고/또는 특정 추상 데이터 타입을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조 등을 포함한다.
- [0113] 더욱이, 본 발명의 방법은 각자가 하나 이상의 관련 장치와 통신할 수 있는 단일 프로세서 또는 멀티 프로세서 컴퓨터 시스템, 미니 컴퓨터, 메인프레임 컴퓨터, 개인용 컴퓨터, 핸드헬드 컴퓨팅 장치, 마이크로프로세서 기반 및/또는 프로그래머블 소비자 전자 장치 등을 포함하는 다른 컴퓨터 시스템 구성으로 실시될 수 있다는 것을 당업자는 이해할 것이다. 본 발명의 도시된 양태들은 통신망을 통해 링크된 원격 처리 장치들에 의해 소정의 태스크가 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다. 그러나, 본 발명의 모든 양태는 아니지만 그 일부는 독립식 컴퓨터 상에서 실시될 수 있다. 분산 컴퓨팅 환경에서 프로그램 모듈들은 로컬 및/또는 원격 메모리 저장 장치 양자에 위치할 수 있다.
- [0114] 도 11은 본 발명이 상호작용할 수 있는 샘플 컴퓨팅 환경(1100)의 개략 블록도이다. 시스템(1100)은 하나 이상의 클라이언트(1110)를 포함한다. 클라이언트(1110)는 하드웨어 및/또는 소프트웨어(예를 들어, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 시스템(1100)은 또한 하나 이상의 서버(1120)를 포함한다. 서버(1120)는 하드

웨어 및/또는 소프트웨어(예를 들어, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 서버(1120)는 예를 들어 본 발명을 이용함으로써 변환을 수행하도록 스레드를 하우징할 수 있다.

[0115] 클라언트(1110)와 서버(1120) 간의 하나의 가능한 통신은 둘 이상의 컴퓨터 프로세스 사이에 전송되기에 적합한 데이터 패킷의 형태일 수 있다. 시스템(1100)은 클라이언트(1110)와 서버(1120) 사이의 통신을 용이하게 하는 데 사용될 수 있는 통신 프레임워크(1140)를 포함한다. 클라이언트(1110)는 클라이언트(1110)에 로컬한 정보를 저장하는 데 사용될 수 있는 하나 이상의 클라이언트 데이터 스토어(1150)에 동작적으로 접속된다. 마찬가지로, 서버(1120)는 서버(1120)에 로컬한 정보를 저장하는 데 사용될 수 있는 하나 이상의 데이터 스토어(1130)에 동작적으로 접속된다.

[0116] 도 12를 참조하면, 본 발명의 다양한 양태를 구현하기 위한 예시적인 환경(1200)은 컴퓨터(1212)를 포함한다. 컴퓨터(1212)는 처리 유닛(1214), 시스템 메모리(1216), 및 시스템 버스(1218)를 포함한다. 시스템 버스(1218)는 시스템 메모리(1216)를 포함하지만 이에 한하지 않는 시스템 캠퍼넌트를 처리 유닛(1214)에 결합시킨다. 처리 유닛(1214)은 이용가능한 다양한 마이크로프로세서 중 어느 하나일 수 있다. 이중 마이크로프로세서 및 다른 마이크로프로세서 아키텍쳐도 처리 유닛(1214)으로 사용될 수 있다.

[0117] 시스템 버스(1218)는 메모리 버스 또는 메모리 제어기, 주변 버스 또는 외부 버스, 및/또는 ISA, MSA, EISA, IDE, VESA 로컬 버스(VLB), PCI, 카드 버스, USB, AGP, PCMCIA, 방화벽(IEEE 1394) 및 SCSI를 포함하지만 이에 한하지 않는 임의의 다양한 이용가능한 버스 아키텍처를 이용하는 로컬 버스를 포함하는 여러 타입의 버스 구조 중 어느 하나일 수 있다.

[0118] 시스템 메모리(1216)는 휘발성 메모리(1220) 및 불휘발성 메모리(1222)를 포함한다. 예를 들어 시동시에 컴퓨터(1212) 내의 요소들 사이에 정보를 전달하는 기본 루틴을 포함하는 기본 입출력 시스템(BIOS)은 불휘발성 메모리(1222)에 저장된다. 예를 들어, 불휘발성 메모리(1222)는 ROM, PROM, EPROM, EEPROM, 또는 플래시 메모리를 포함할 수 있지만 이에 한하지 않는다. 휘발성 메모리(1220)는 외부 캐시 메모리로서 동작하는 RAM을 포함한다. 예를 들어, RAM은 SRAM, DRAM, SDRAM, DDR SDRAM, ESDRAM, SDRAM, 램버스 다이렉트 RAM(RDRAM), DRDRAM, 및 램버스 다이나믹 RAM(RDRAM)과 같은 다양한 형태로 이용가능하지만 이에 한하지 않는다.

[0119] 컴퓨터(1212)는 분리식/비분리식, 휘발성/불휘발성 컴퓨터 저장 매체도 포함한다. 도 12는 예를 들어 디스크 저장 장치(1224)를 도시한다. 디스크 저장장치(1224)는 자기 디스크 드라이브, 플로피 디스크 드라이브, 테이프 드라이브, 재즈 드라이브, 짚 드라이브, LS-100 드라이브, 플래시 메모리 카드 또는 메모리 스틱을 포함하지만 이에 한하지 않는다. 또한, 디스크 저장 장치(1224)는 CD-ROM, CD-R 드라이브, CD-RW 드라이브 또는 DVD-ROM과 같은 광 디스크 드라이브를 포함하지만 이에 한하지 않는 다른 저장 매체와 별개로 또는 조합하여 저장 매체를 포함할 수 있다. 디스크 저장 장치(1224)와 시스템 버스(1218)의 접속을 용이하게 하기 위하여, 인터페이스(1226)와 같은 분리식 또는 비분리식 인터페이스가 일반적으로 사용된다.

[0120] 도 12는 적절한 동작 환경(1210)에서 설명되는 사용자와 기본 컴퓨터 자원 간의 매개물로서 동작하는 소프트웨어를 나타낸다는 것을 이해해야 한다. 이 소프트웨어는 운영 체제(1228)를 포함한다. 디스크 저장 장치(1224)에 저장될 수 있는 운영 체제(1228)는 컴퓨터 시스템(1212)의 자원을 제어하고 할당한다. 시스템 애플리케이션(1230)은 시스템 메모리(1216) 또는 디스크 저장 장치(1224)에 저장되는 프로그램 모듈(1232) 및 프로그램 데이터(1234)를 통해 운영 체제(1228)에 의해 자원의 관리를 이용한다. 본 발명은 다양한 운영 체제 또는 운영 체제들의 조합에 의해 구현될 수 있다는 것을 이해해야 한다.

[0121] 사용자는 입력 장치(1236)를 통해 컴퓨터(1212)에 명령 또는 정보를 입력한다. 입력 장치(1236)는 마우스, 트랙볼, 스타일러스, 터치 패드와 같은 포인팅 장치, 키보드, 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너, TV 투너 카드, 디지털 카메라, 디지털 비디오 카메라, 웹 카메라 등을 포함하지만 이에 한하지 않는다. 이들 및 다른 입력 장치는 인터페이스 포트(1238)를 경유하여 시스템 버스(1218)를 통해 처리 유닛(1214)에 접속된다. 인터페이스 포트(1238)는 직렬 포트, 병렬 포트, 게임 포트 및 USB를 포함하지만 이에 한하지 않는다. 출력 장치(1240)는 입력 장치(1236)와 동일한 타입의 포트들 중 일부를 이용한다. 따라서, 예를 들어, USB 포트를 사용하여 컴퓨터(1212)에 입력을 제공하고 컴퓨터(1212)에서 출력 장치(1240)로 정보를 출력할 수 있다. 출력 어댑터(1242)는 특수 어댑터를 필요로 하는 다른 출력 장치들 중에서 디스플레이(예를 들어, 평판 패널 및 CRT), 스피커 및 프린터와 같은 소정의 출력 장치(1240)가 있다는 것을 나타내기 위하여 제공된다. 출력 어댑터(1242)는 예를 들어 출력 장치(1240)와 시스템 버스(1218) 사이의 접속 수단을 제공하는 비디오 및 사운드 카드를 포함하지만 이에 한하지 않는다. 다른 장치 및/또는 장치들의 시스템도 원격 컴퓨터(1244)와 같은 입력 출력 능력 양자를 제공한다는 점에 유의해야 한다.

[0122] 컴퓨터(1212)는 원격 컴퓨터(1244)와 같은 하나 이상의 원격 컴퓨터에 대한 논리적 접속을 이용하여 네트워크 환경에서 동작할 수 있다. 원격 컴퓨터(1244)는 개인용 컴퓨터, 서버, 라우터, 네트워크 PC, 워크스테이션, 마이크로프로세서 기반 기기, 피어 장치 또는 다른 공통 네트워크 노드 등일 수 있으며, 일반적으로 컴퓨터(1212)와 관련하여 설명된 요소들의 다수 또는 전부를 포함한다. 간략화를 위해, 메모리 저장 장치(1246)만이 원격 컴퓨터(1244)와 함께 도시되어 있다. 원격 컴퓨터(1244)는 네트워크 인터페이스(1248)를 통해 컴퓨터(1212)에 논리적으로, 그리고 통신 접속(1250)을 통해 물리적으로 접속된다. 네트워크 인터페이스(1248)는 LAN 및 WAN과 같은 통신 네트워크를 포함한다. LAN 기술은 FDDI, CDDI, 이더넷, 토큰 링 등을 포함한다. WAN 기술은 점대점 링크, ISDN 및 그 변형과 같은 회로 교환망, 패킷 교환망, 및 DSL을 포함하지만 이에 한하지 않는다.

[0123] 통신 접속(1250)은 네트워크 인터페이스(1248)를 버스(1218)에 접속시키는 데 사용되는 하드웨어/소프트웨어를 나타낸다. 통신 접속은 간략화를 위해 컴퓨터(1212) 내부에 도시되어 있지만, 컴퓨터(1212) 외부에 있을 수도 있다. 네트워크 인터페이스(1248)의 접속에 필요한 하드웨어/소프트웨어는 예를 들어 정규 전화급 모뎀, 케이블 모뎀, 및 DSL 모뎀과 같은 모뎀들, ISDN 어댑터 및 이더넷 카드와 같은 내장 및 외장 기술을 포함한다.

[0124] 전술한 내용은 본 발명의 예를 포함한다. 물론, 본 발명을 설명할 목적으로 컴퓨터(1212) 또는 방법의 모든 구상 가능한 조합을 설명하는 것을 불가능하지만, 본 발명의 많은 추가적인 조합 및 교환이 가능하다는 것을 당업자는 이해할 수 있을 것이다. 따라서, 본 발명은 첨부된 청구범위의 사상 및 범위 내에 있는 모든 교체, 수정 및 변경을 포함하는 것으로 의도된다.

[0125] 구체적으로, 그리고 전술한 컴퓨터(1212), 장치, 회로, 시스템 등에 의해 수행되는 다양한 기능과 관련하여, 이러한 컴퓨터(1212)들을 설명하는 데 사용되는 용어들("수단"에 대한 참조 포함)은 달리 지시하지 않는 한, 본 명세서에서 설명되는 본 발명의 예시적인 양태에서 기능을 수행하는 개시된 구조와 구조적으로 동일하지 않더라도 설명된 컴퓨터(1212)의 지정 기능을 수행하는 임의의 컴퓨터(1212)에 대응하는 것으로 의도된다. 이와 관련하여, 본 발명은 시스템은 물론, 본 발명의 다양한 방법의 동작 및/또는 이벤트를 수행하기 위한 컴퓨터(1212) 실행 가능 명령을 구비한 컴퓨터(1212) 판독 가능 매체도 포함한다는 것을 이해할 것이다.

[0126] 또한, 본 발명의 특징이 여러 구현 중 하나에만 관련하여 설명되었을 수 있지만, 이러한 특징은 임의의 주어진 또는 특정 애플리케이션에 바람직하고 이로울 수 있는 바와 같이 다른 구현들의 하나 이상의 다른 특징과 조합될 수 있다. 또한, "포함", "구비"라는 용어 또는 그 변형은 상세한 설명 또는 청구범위에서 사용되는 범위 정도로, 이들 용어는 용어 "포함"과 유사한 방식으로 포괄적인 것으로 의도된다.

발명의 효과

[0127] 본 발명은 데이터 모델의 데이터베이스 객체로의 맵핑을 용이하게 하는 시스템 및/또는 방법을 제공한다.

도면의 간단한 설명

[0001] 도 1은 데이터 모델과 연관된 타입 인스턴스의 저장을 용이하게 하는 예시적인 시스템의 블록도.

[0002] 도 2는 타입 인스턴스의 저장 및/또는 항목, 문서 및 콘택 중 적어도 하나를 발견하기 위한 질의를 용이하게 하는 예시적인 시스템의 블록도.

[0003] 도 3은 관계 스토리지 및/또는 관계 질의 능력의 이용을 용이하게 하는 예시적인 시스템의 블록도.

[0004] 도 4는 타입 스토리지 시스템과 관련한 맵핑 및/또는 뷰잉을 용이하게 하는 예시적인 시스템의 블록도.

[0005] 도 5는 데이터 모델과 연관된 타입 인스턴스의 저장을 용이하게 하는 예시적인 시스템의 블록도.

[0006] 도 6은 파일 스토리지 데이터 모델 내의 스토어의 하이 레벨 구조의 블록도.

[0007] 도 7은 관련 테이블을 가진 타입 인스턴스의 블록도.

[0008] 도 8은 타입 계층 구조 및 대응 뷰 투영의 블록도.

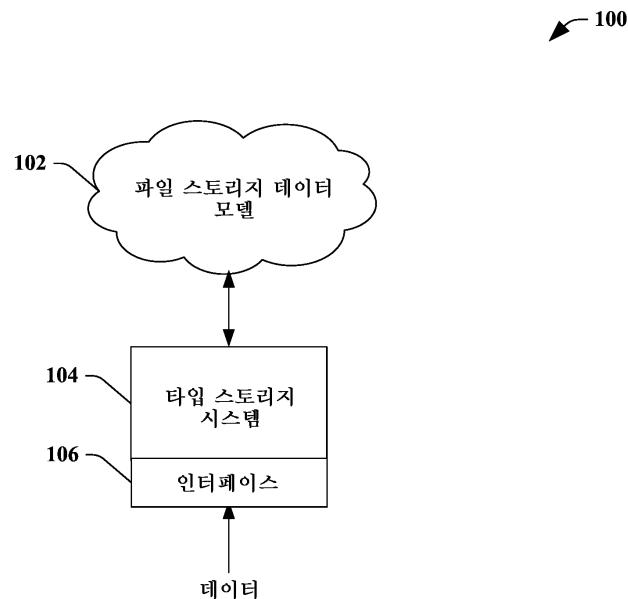
[0009] 도 9는 데이터 모델로 맵핑되는 타입 인스턴스의 저장 및/또는 질의 제공을 위한 예시적인 방법을 나타내는 도면.

[0010] 도 10은 데이터 모델로 맵핑되는 타입 인스턴스의 저장 및/또는 질의 제공을 위한 예시적인 방법을 나타내는 도면.

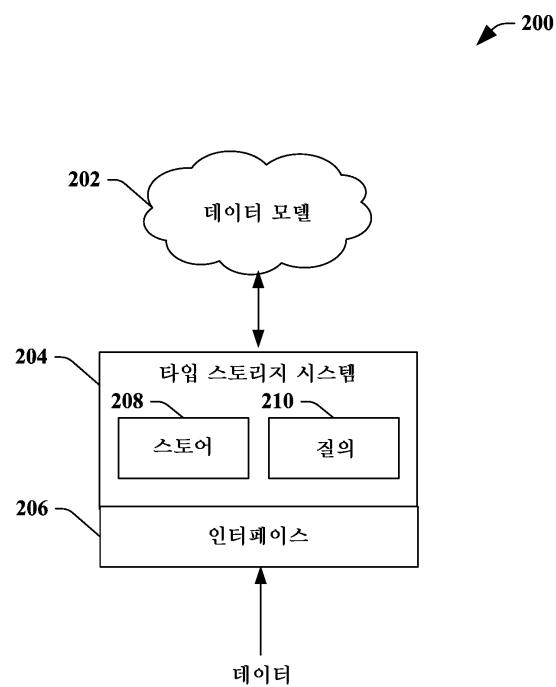
- [0011] 도 11은 본 발명의 신규 양태들이 이용될 수 있는 예시적인 네트워킹 환경을 나타내는 도면.
- [0012] 도 12는 본 발명에 따라 이용될 수 있는 예시적인 동작 환경을 나타내는 도면.
- [0013] <도면의 주요 부분에 대한 부호 설명>
- [0014] 102: 파일 스토리지 데이터 모델
- [0015] 104: 타입 스토리지 시스템
- [0016] 106: 인터페이스
- [0017] 202: 데이터 모델
- [0018] 204: 타입 스토리지 시스템
- [0019] 206: 인터페이스
- [0020] 208: 스토어
- [0021] 210: 질의
- [0022] 302: 데이터 모델
- [0023] 304: 타입 스토리지 시스템
- [0024] 306: 인터페이스
- [0025] 308: 스토어
- [0026] 310: 질의
- [0027] 312: 관계
- [0028] 402: 데이터 모델
- [0029] 404: 타입 스토리지 시스템
- [0030] 406: 인터페이스
- [0031] 408: 맵 컴포넌트
- [0032] 410: 뷰 컴포넌트
- [0033] 502: 데이터 모델
- [0034] 504: 타입 스토리지 시스템
- [0035] 506: 인터페이스
- [0036] 508: 지능 컴포넌트
- [0037] 510: 데이터 스토어

도면

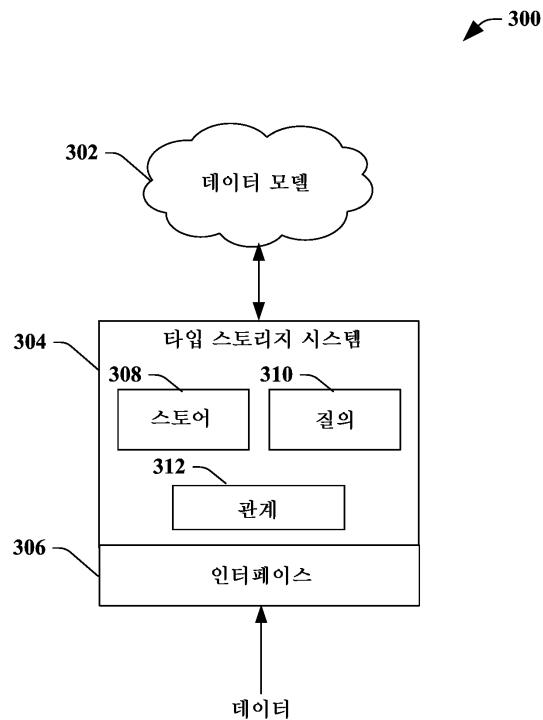
도면1



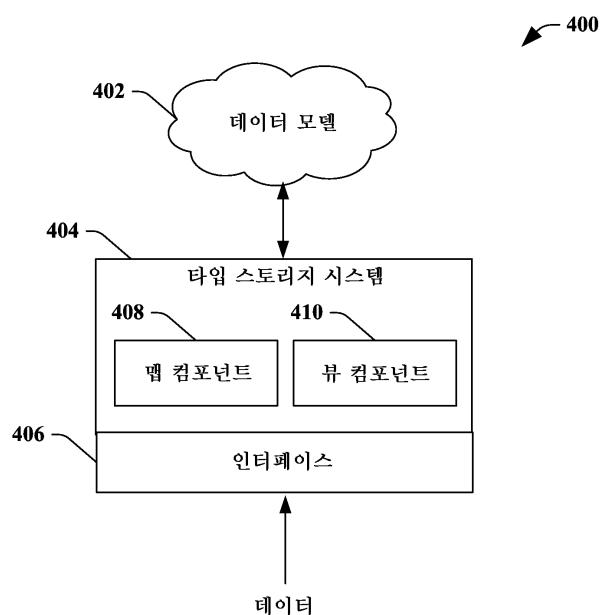
도면2



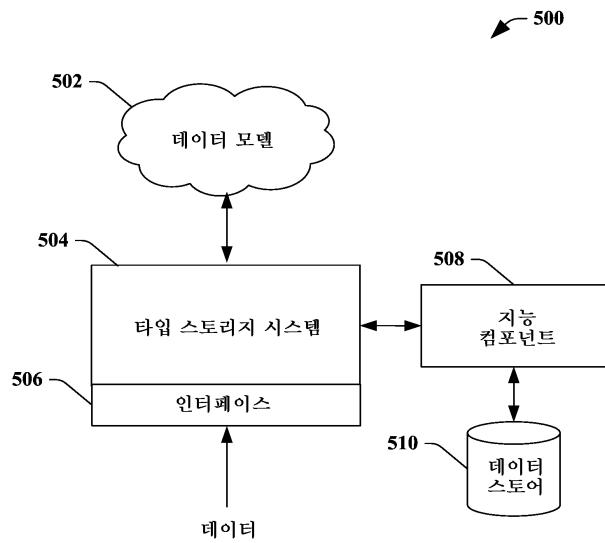
도면3



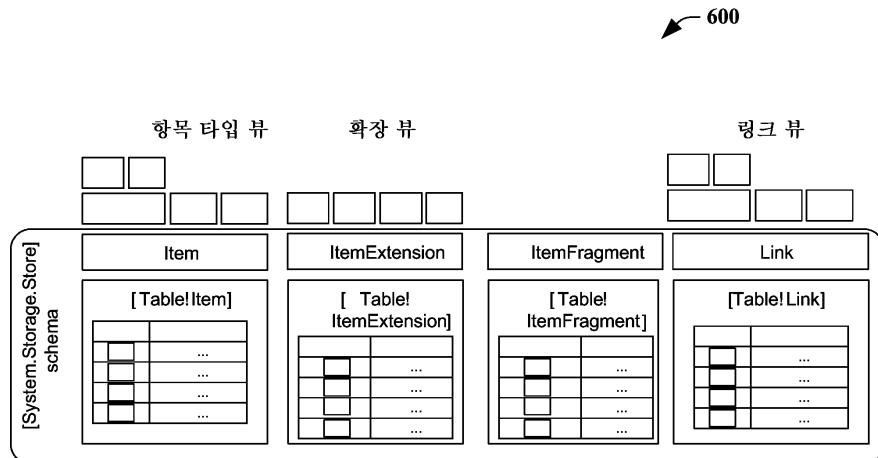
도면4



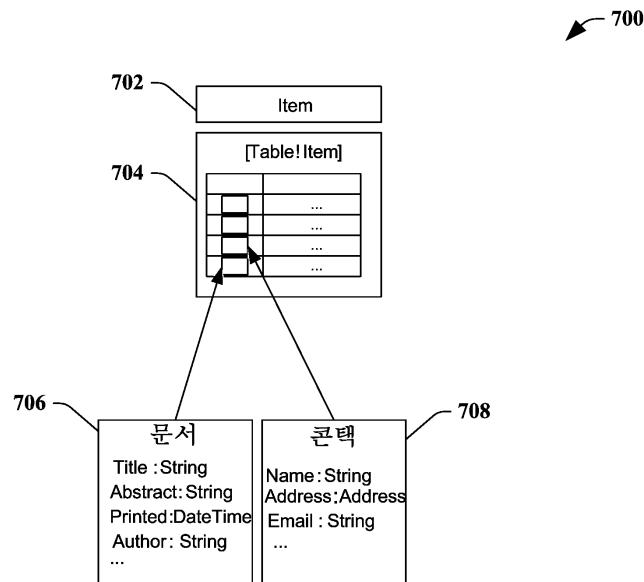
도면5



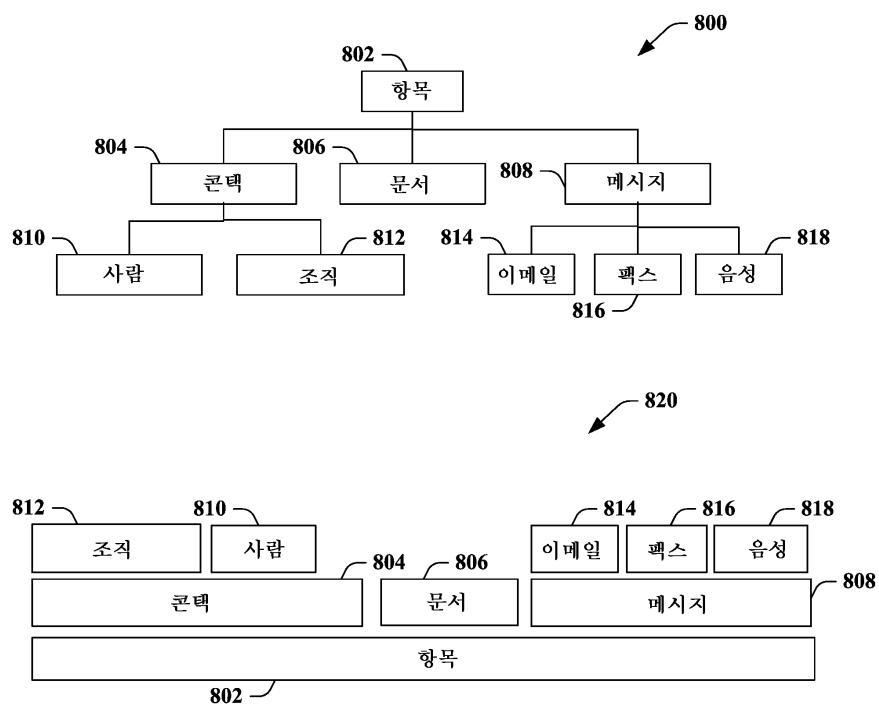
도면6



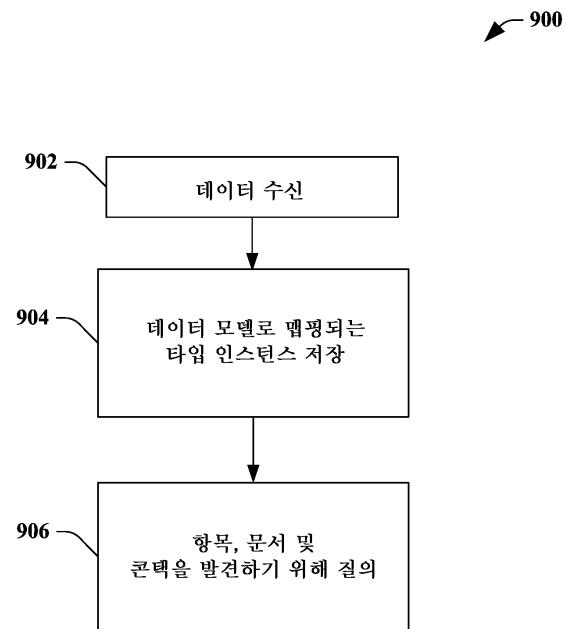
도면7



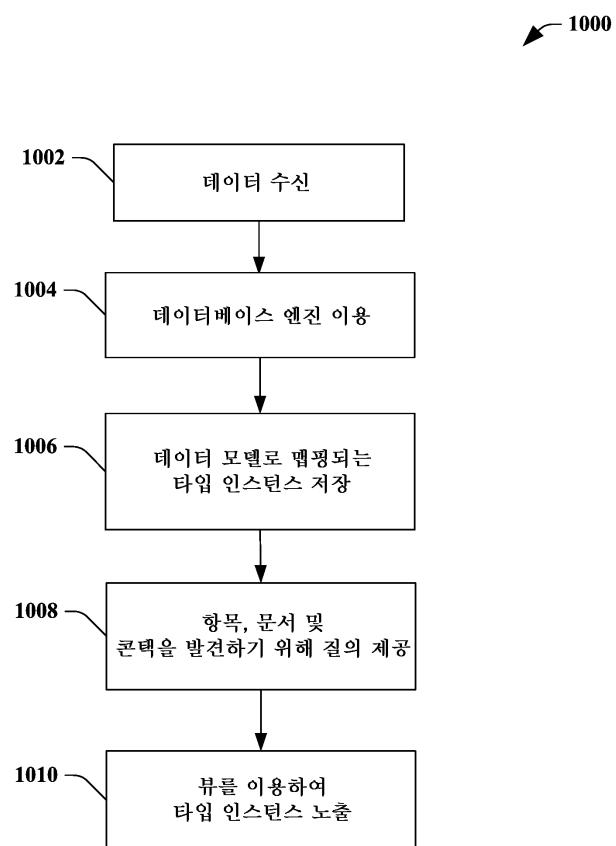
도면8



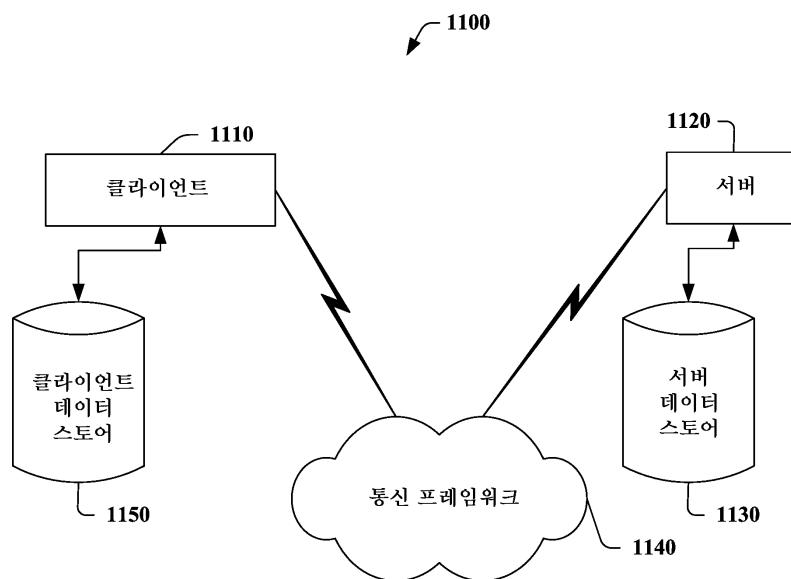
도면9



도면10



도면11



도면12

