



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년03월31일
(11) 등록번호 10-1608101
(24) 등록일자 2016년03월25일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G06F 17/10 (2006.01)
(21) 출원번호 10-2010-7027380
(22) 출원일자(국제) 2009년05월15일
심사청구일자 2014년04월18일
(85) 번역문제출일자 2010년12월06일
(65) 공개번호 10-2011-0030444
(43) 공개일자 2011년03월23일
(86) 국제출원번호 PCT/US2009/044258
(87) 국제공개번호 WO 2009/151873
국제공개일자 2009년12월17일
(30) 우선권주장
12/136,616 2008년06월10일 미국(US)
(56) 선행기술조사문헌
US20040215626 A1
US20020078094 A1
US20060224556 A1
JP2007234013 A

(73) 특허권자
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
(72) 발명자
허그레스, 그레고리, 엘.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이
커러, 클레먼스
미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이
올렌닉, 브래드 엠.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이
(74) 대리인
제일특허법인

전체 청구항 수 : 총 20 항

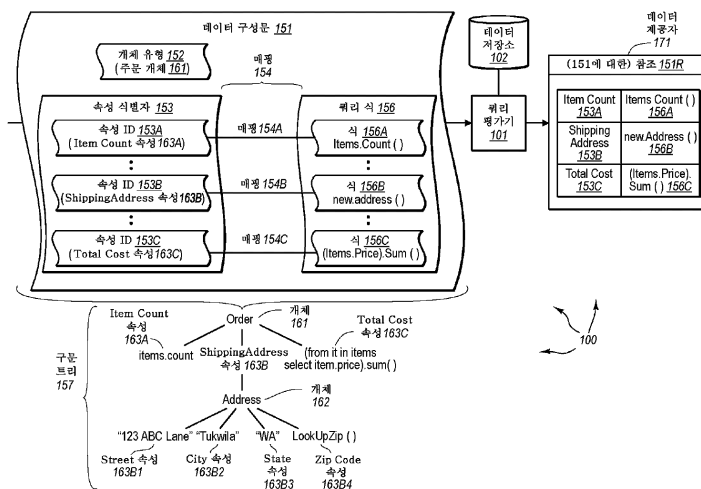
심사관 : 박상현

(54) 발명의 명칭 쿼리 내의 객체의 가상화 기법

(57) 요약

본 발명은 쿼리 내의 객체를 가상화하는 방법, 시스템 및 컴퓨터 프로그램 제품에 관한 것이다. 본 발명의 실시예들은, 쿼리에서 사용하기 위한 데이터 액세스를 가상화한다. 가상화는 구문 트리의 임의의 부분 내에서 구현될 수 있다. 예를 들어, 그 자체가 다른 객체인 객체의 속성에 대한 데이터가 가상화될 수 있다. 데이터 가상화는 쿼리 식의 느린 평가를 용이하게 해준다. 즉, 쿼리가 실제 속성 값을 구체적으로 요청할 때까지, 데이터 구성문 내의 속성들에 대한 실제 속성 값이 가상화된다. 게다가, 데이터 가상화는 또한 자원들을 절감하고 그 결과 쿼리 평가가 보다 효율적으로 된다.

대표도



명세서

청구범위

청구항 1

하나 이상의 프로세서와 메모리를 포함하는 컴퓨터 시스템 상에서 구현되는 방법으로서,

상기 메모리는 명령어를 포함하고,

상기 명령어는 상기 하나 이상의 프로세서에 의해 실행되어, 객체 전체를 구성하지 않고도 상기 객체의 복수의 속성의 세브세트가 액세스될 수 있도록 상기 객체에 대한 데이터 액세스를 가상화하는 데이터 제공자(data provider)를 생성하는 방법을 구현하며,

상기 방법은

상기 하나 이상의 프로세서를 포함하는 컴퓨터 시스템이 복수의 속성을 가진 객체의 구문 트리(syntax tree)의 적어도 일부를 기술하는 데이터 구성문(data construction statement)을 수신하는 단계와,

상기 데이터 구성문을 수신하는 것에 응답하여, 상기 컴퓨터 시스템이 데이터 제공자를 생성하는 단계를 포함하되,

상기 복수의 속성 중 적어도 하나의 속성의 값은 쿼리 식(query expression)의 실행에 의해 결정가능하고,

상기 데이터 구성문은 상기 데이터 구성문이 하나 이상의 속성 식별자(property identifier)와 하나 이상의 매핑의 구성을 나타내는 객체 유형의 표지(indication)를 포함하며,

상기 하나 이상의 속성 각각은 상기 객체의 속성을 식별해주고,

상기 하나 이상의 매핑 각각은 하나의 속성 식별자를 대응하는 하나의 쿼리 식에 매핑해주며,

상기 대응하는 쿼리 식은 상기 대응하는 속성 식별자에 의해 식별되는 상기 객체의 상기 속성의 값을 어떻게 계산해야 하는지 나타내고,

상기 데이터 제공자는 상기 객체에 대한 데이터 액세스를 가상화하고, 상기 데이터 구성문에 기초하여 상기 객체의 부분 구성(partial construction)을 생성함으로써 상기 객체에 대한 쿼리의 지연 평가(lazy evaluation)를 용이하게 하며,

상기 객체의 상기 부분 구성은 상기 객체 전체를 완전히 구성하거나 상기 데이터 구성문 전체를 평가할 필요 없이 상기 객체의 상기 하나 이상의 속성에 대응하는 하나 이상의 쿼리 식을 실행함으로써 상기 객체의 하나 이상의 속성의 값에 대한 액세스를 제공하고,

상기 데이터 제공자는 상기 데이터 구성문에 대한 참조를 포함하며,

상기 데이터 구성문에 대한 참조는 상기 데이터 구성문의 상기 대응하는 매핑, 상기 유형, 상기 속성 식별자에 대한 참조를 임시로(transitively) 포함하는

데이터 제공자 생성 방법.

청구항 2

제1항에 있어서,

상기 데이터 구성문을 수신하는 단계는 상기 데이터 구성문에 의해 기술되는 가상화된 객체의 속성들에 대한 임의의 쿼리를 제출하기 전에 평가를 위하여 상기 데이터 구성문을 제출하는 사용자 입력을 수신하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 3

제1항에 있어서,

상기 데이터 구성문을 수신하는 단계는 기존의 데이터 제공자로부터 상기 데이터 구성문을 수신하는 단계를 포함하고,

상기 데이터 구성문은 상기 기존의 데이터 제공자의 속성 매핑으로부터의 쿼리 식 내에 포함되는 데이터 제공자 생성 방법.

청구항 4

제1항에 있어서,

상기 데이터 제공자 내의 속성의 값에 대한 데이터 소비자(data consumer)로부터의 요청을 수신하는 단계와,

상기 대응하는 하나 이상의 매핑을 참조하여 식별된 상기 속성에 대응하는 쿼리 식을 찾아내는 단계와,

상기 쿼리 식을 평가하여 상기 속성에 대한 속성 값을 획득하는 단계와,

상기 속성 값을 상기 데이터 소비자에게 반환하는 단계

를 더 포함하는 데이터 제공자 생성 방법.

청구항 5

제4항에 있어서,

상기 쿼리 식을 평가하여 상기 속성에 대한 속성 값을 획득하는 단계는 상기 쿼리 식을 평가하여 상기 속성에 대한 실제 속성 값을 획득하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 6

제5항에 있어서,

상기 쿼리 식을 평가하여 상기 속성에 대한 실제 속성 값을 획득하는 단계는 데이터 저장소(data repository)를 참조하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 7

제6항에 있어서,

상기 데이터 저장소를 참조하는 단계는 상기 데이터 구성문에 의해 기술되는 임의의 다른 속성들의 값을 계산하기 위해 데이터에 액세스하지 않고, 상기 실제 속성 값을 계산할 목적으로 데이터에 액세스하기 위해 상기 데이터 저장소를 참조하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 8

제4항에 있어서,

상기 쿼리 식을 평가하여 상기 속성에 대한 속성 값을 획득하는 단계는 상기 쿼리 식을 평가하여 추가적인 데이터 제공자를 획득하는 단계를 포함하고,

상기 추가적인 데이터 제공자는 상기 데이터 제공자로부터의 상기 속성 값을 가상화하는

데이터 제공자 생성 방법.

청구항 9

제8항에 있어서,

상기 쿼리 식을 평가하여 추가적인 데이터 제공자를 획득하는 단계는 상기 데이터 제공자에 포함된 추가적인 데이터 구성문을 평가하여 상기 추가적인 데이터 제공자를 생성하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 10

제1항에 있어서,

상기 데이터 구성문으로부터의 속성 식별자를 포함하는 속성 값 액세스 쿼리(property value access query)를 수신하는 단계와,

상기 속성 식별자에 기초하여 상기 속성 값 액세스 쿼리에 응답하기 위해 초기 데이터 제공자를 식별하는 단계와,

쿼리 평가기와 연관된 기지의 장소에 상기 초기 데이터 제공자를 위치시키는 단계와,

상기 속성 값 액세스 쿼리에 대한 데이터 소비자를 생성하는 단계와,

상기 데이터 소비자가 상기 기지의 장소로부터 상기 초기 데이터 제공자에 액세스하는 단계와,

상기 데이터 소비자가 상기 초기 데이터 제공자로부터 상기 식별된 속성의 속성 값을 요청하는 단계와,

상기 초기 데이터 제공자가 이전에 수신된 상기 데이터 구성문으로부터 상기 식별된 속성에 대응하는 쿼리 식을 검색하는 단계와,

상기 쿼리 식을 평가하여 상기 속성에 대한 속성값을 획득하는 단계와,

상기 초기 데이터 제공자가 상기 획득된 속성 값을 상기 데이터 소비자에게 반환하는 단계와,

상기 데이터 소비자가 상기 획득된 속성 값을 수신하는 단계와,

상기 데이터 소비자가 추가적인 처리를 위해 수신된 상기 속성 값을 상기 기지의 장소에 위치시키는 단계를 더 포함하는 데이터 제공자 생성 방법.

청구항 11

제10항에 있어서,

상기 쿼리 식을 평가하여 상기 속성에 대한 속성값을 획득하는 단계는 데이터베이스로부터의 데이터에 액세스를 하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 12

제10항에 있어서,

상기 쿼리 식을 평가하여 상기 속성에 대한 속성값을 획득하는 단계는 추가 데이터 제공자를 구성하는 단계를 포함하는

데이터 제공자 생성 방법.

청구항 13

프로세서와 메모리를 포함하는 컴퓨터 시스템 상에서 구현되는 방법으로서,

상기 메모리는 명령어를 포함하고,

상기 명령어는 상기 프로세서에 의해 실행되어, 데이터 제공자를 이용하는 쿼리 평가기(query evaluator)를 이용하여 쿼리에 응답하여, 객체 전체를 구성하지 않고도 상기 객체의 복수의 속성의 세브셋트가 쿼리 평가자(query evaluator)에게 액세스가능하도록 상기 객체에 대한 데이터 액세스를 가상화하는 데이터 제공자(data provider)를 생성하는 방법을 구현하되,

상기 방법은

상기 프로세서를 포함하는 컴퓨터 시스템이 속성 값 액세스 쿼리를 수신하는 단계 - 상기 속성 값은 복수의 속성을 갖는 객체의 구문 트리의 적어도 일부를 기술하는 이전에 수신된 데이터 구성문에서 식별되는 속성의 속성 값에 대해 질의하고, 상기 복수의 속성 중 적어도 하나의 속성의 값은 쿼리 식의 실행에 의해 결정가능함 - 와,

상기 컴퓨터 시스템이 상기 속성 값 액세스 쿼리 내에 포함된 데이터에 기초하여 상기 이전에 수신된 데이터 구성문으로부터 생성된 데이터 제공자를 식별하는 단계 - 상기 데이터 제공자는 상기 객체에 대한 데이터 액세스를 가상화하고, 상기 데이터 구성문에 기초하여 상기 객체의 부분 구성을 생성함으로써 상기 객체에 대한 쿼리의 지연 평가를 용이하게 하며, 상기 객체의 부분 구성은 상기 객체 전체를 완전히 구성하거나 상기 데이터 구성문 전체를 평가할 필요 없이 상기 하나 이상의 속성에 대응하는 하나 이상의 쿼리 식을 실행함으로써 상기 객체의 하나 이상의 속성의 값에 대한 액세스를 제공함 - 와,

상기 컴퓨터 시스템이 상기 식별된 데이터 제공자를 상기 쿼리 평가와 연관된 기지의 장소에 위치시키는 단계와,

상기 컴퓨터 시스템이 상기 속성 값 액세스 쿼리에 대한 데이터 소비자를 생성하는 단계와,

상기 데이터 소비자가 상기 기지의 장소로부터의 상기 식별된 데이터 제공자에 액세스하는 단계와,

상기 데이터 소비자가 상기 데이터 제공자에게 상기 식별된 속성의 속성 값을 요청하는 단계와,

상기 데이터 소비자가 상기 데이터 제공자로부터 상기 식별된 속성의 속성 값을 수신하는 단계와,

상기 데이터 소비자가 추가적인 처리를 위해 수신된 상기 속성 값을 상기 기지의 장소에 위치시키는 단계를 포함하는 쿼리 응답 방법.

청구항 14

제13항에 있어서,

상기 데이터 소비자가 상기 데이터 제공자로부터 상기 식별된 속성의 속성 값을 수신하는 단계는 상기 데이터 소비자가 상기 데이터 제공자로부터 상기 속성의 실제 속성 값을 수신하는 단계를 포함하는

쿼리 응답 방법.

청구항 15

제13항에 있어서,

상기 데이터 제공자로부터 상기 속성의 실제 속성 값을 수신하는 단계는 상기 이전에 수신된 데이터 구성문에 의해 기술되는 다른 속성들에 대한 쿼리 식을 평가할 필요없이 상기 속성의 실제 속성 값을 수신하는 단계를 포함하는

쿼리 응답 방법.

청구항 16

제13항에 있어서,

상기 데이터 소비자가 상기 데이터 제공자로부터 상기 식별된 속성의 속성 값을 수신하는 단계는 상기 데이터 소비자가 상기 데이터 제공자로부터 추가적인 데이터 제공자를 수신하는 단계를 포함하되,

상기 추가적인 데이터 제공자는 상기 데이터 제공자로부터의 상기 속성 값을 가상화하는

쿼리 응답 방법.

청구항 17

제13항에 있어서,

상기 데이터 소비자가 추가적인 처리를 위해 수신된 상기 속성 값을 기지의 장소에 위치시키는 단계는 상기 데이터 소비자가 상기 속성의 실제 속성 값을 상기 기지의 장소에 위치시키는 단계를 포함하는

쿼리 응답 방법.

청구항 18

제13항에 있어서,

상기 데이터 소비자가 추가적인 처리를 위해 획득된 상기 속성 값을 기지의 장소에 위치시키는 단계는 추가적인 데이터 제공자를 상기 기지의 장소에 위치시키는 단계를 포함하되,

상기 추가적인 데이터 제공자는 상기 데이터 제공자로부터의 상기 속성 값을 가상화하는

쿼리 응답 방법.

청구항 19

제18항에 있어서,

상기 속성 값 액세스 쿼리에 대한 추가적인 데이터 소비자를 생성하는 단계와,

상기 추가적인 데이터 소비자가 상기 기지의 장소로부터의 상기 추가적인 데이터 제공자에 액세스하는 단계와,

상기 추가적인 데이터 소비자가 상기 추가적인 데이터 제공자에게 상기 식별된 속성의 속성 값을 요청하는 단계와,

상기 데이터 소비자가 상기 추가적인 데이터 제공자로부터 상기 식별된 속성의 속성 값을 수신하는 단계와,

상기 데이터 소비자가 추가적인 처리를 위해 수신된 상기 속성 값을 상기 기지의 장소에 위치시키는 단계

를 더 포함하는 쿼리 응답 방법.

청구항 20

제19항에 있어서,

상기 속성의 속성 값을 수신하는 단계는 상기 이전에 수신된 데이터 구성문에 의해 기술되는 속성들에 대한 쿼리 식을 평가하기 위해 데이터 저장소를 참조할 필요없이 그리고 상기 추가적인 데이터 구성문에 의해 기술되는 다른 속성들에 대한 쿼리 식을 평가할 필요없이 상기 속성의 실제 속성 값을 수신하는 단계를 포함하는

쿼리 응답 방법.

발명의 설명

배경 기술

[0001]

컴퓨터 시스템 및 관련 기술이 사회의 많은 측면들에 영향을 미치고 있다. 실제로, 컴퓨터 시스템의 정보 처리 능력은 우리가 살고 일하는 방식을 바꾸어 놓았다. 컴퓨터 시스템은 이제 통상 컴퓨터 시스템의 출현 이전에 수작업으로 수행되었던 다수의 작업들(예를 들어, 워드 프로세싱, 스케줄링, 회계, 기타)을 수행한다. 보다 최근에, 컴퓨터 시스템들은 유선 및 무선 컴퓨터 네트워크 둘다를 형성하기 위해 서로 또 다른 전자 장치들과 결합되고 있으며 이 네트워크를 통해 컴퓨터 시스템들 및 다른 전자 장치들은 전자 데이터를 전송할 수 있다. 따라서, 많은 컴퓨팅 작업들의 수행이 다수의 서로 다른 컴퓨터 시스템들 및/또는 다수의 서로 다른 컴퓨팅 환경들에 걸쳐 분산된다.

[0002]

전부는 아닐지라도 대부분의 컴퓨팅 작업들이 통상적으로, 예를 들어, 원하는 결과를 얻기 위해 데이터를 생성, 수정, 삭제, 기타 등등을 하는 것에 관한 동작들을 수행하는 소프트웨어(예를 들어, 응용 프로그램 또는 운영 체제)를 포함한다. 그에 따라, 소프트웨어가 원하는 결과를 얻기 위해, 소프트웨어는 적절한 데이터에 액세스해야만 한다. 예를 들어, 워드 프로세싱 문서에 대한 편집을 하기 위해, 사용자는 (예를 들어, 파일 시스템 내에서) 워드 프로세싱 문서를 찾아내고 워드 프로세서에게 워드 프로세싱 문서를 열라고 지시하는 명령을 워드 프로세서에 입력해야 한다.

[0003]

사용자가 문서 위치를 알지 못하는 경우, 사용자는 문서를 찾아내기 위해 다양한 검색 도구를 사용할 수 있다. 사용자는 검색 모듈(예를 들어, 파일 파인더(file finder))에 다양한 검색어를 입력할 수 있다. 검색 모듈은 검색어에서 식별된 특성들을 포함하는 파일을 파일 시스템에서 검색하여 임의의 식별된 파일들의 위치를 사용자

에게 반환할 수 있다. 따라서, 사용자가 쿼리(검색어)를 검색 모듈에 제출하는 한, 검색 모듈은 쿼리를 처리하고(모든 일치하는 것을 식별하고), 검색 모듈은 결과(모든 식별된 일치하는 것)를 반환한다.

[0004] 많은 환경들에서, 보다 큰 데이터 부분의 데이터 서브셋(data subset)을 생성하고 액세스하기 위해 보다 복잡한 데이터 동작들이 사용된다. 예를 들어, 데이터베이스 시스템 내에서, 데이터베이스 내의 데이터에 대한 뷰를 생성하기 위해 데이터 생성문(data creation statement)이 사용될 수 있다. 원하는 데이터 서브셋을 검색하기 위해 데이터베이스(또는 심지어 전체 데이터베이스)에 관한 뷰(또는 기타 객체)에 대해 쿼리가 발행될 수 있다. 데이터베이스에서의 데이터의 배열 및 양에 기초하여, 데이터 생성문 및 쿼리가 비교적 복잡할 수 있다. 게다가, 보다 세분하여 데이터에 액세스하기 위해 일련의 데이터 생성문 및 쿼리가 서로 함께 사용될 수 있으며, 그 결과 더욱 복잡하게 된다.

[0005] 그에 부가하여, 원하는 컴퓨팅 동작을 수행하기 위해 그룹화된 데이터의 세트(또는 서브셋)의 전부가 아닌 일부가 요청되는 경우가 종종 있다. 예를 들어, 데이터베이스는 다수의 고객들에 대한 사업체 주소를 포함하는 주소 객체를 포함할 수 있다. 각각의 주소 객체는 거리, 도시, 주, 및 우편번호 필드를 포함할 수 있다. 그렇지만, 원하는 동작(예를 들어, 우편번호별 고객들의 수를 구하는 것)을 수행하기 위해, 우편번호 필드 내의 값들만이 관련성 있다. 따라서, 각각의 주소 객체의 우편번호 필드로부터 값을 획득하기 위해 쿼리가 발행될 수 있다.

[0006] 불행히도, 전부는 아닐지라도 대부분의 데이터베이스 시스템에서, 쿼리가 객체의 일부분(예를 들어, 객체 내의 필드)을 참조할 때, 전체 객체가 인스턴스화되며, 이는 전체 객체를 열심히 평가할 것을 필요로 한다. 즉, 객체의 일부분만을 참조하는 쿼리를 처리하기 전에 전체 객체가 평가된다. 예를 들어, 우편번호 쿼리는 우편번호 쿼리를 처리하기 전에 전체 주소 객체의 인스턴스화를 필요로 할 것이다. 쿼리에 관련성 없는 데이터를 인스턴스화하고 평가하는 것은 불필요하게 컴퓨터 시스템 자원을 소비하고 쿼리 처리 비효율을 야기한다.

[0007] 게다가, 객체 크기가 증가함에 따라, 컴퓨터 시스템 자원의 불필요한 소비 및 쿼리 처리 비효율도 역시 증가한다. 예를 들어, 주문을 나타내는 객체의 생성을 설명하는 다음과 같은 객체 쿼리를 생각해보자.

```

Order order =
    new Order()
    {
        ItemCount = items.Count(),

        ShippingAddress =
            new Address()
            {
                Street = "123 Abc Lane",
                City = "Tukwila",
                State = "WA",
                ZipCode = LookupZipCode(...)
            },
        TotalCost =
            (from it in items
             select item.Price).Sum()
    }
    
```

[0008] 객체에 대한 쿼리:

[0009] Address address = order.Address

[0010] 이 쿼리의 통상적인 평가에서는 Order 객체가 먼저 인스턴스화되어야 하고, 그 결과 그 때의 ItemCount, ShippingAddress, 및 TotalCost 속성들에 대한 표현식들을 열심히 평가해야 한다. 이어서, ShippingAddress 속성의 값이 Order 객체 인스턴스로부터 검색될 것이다. Order 객체의 생성과 ItemCount 및 TotalCost 속성에 대한 표현식의 평가가 두번째 쿼리에서는 사용되지 않지만 컴퓨터 시스템 자원 및 시간이 어쨌든 이들의 생성 및 평가에 사용되었다.

발명의 내용

- [0013] <발명의 요약>
- [0014] 본 발명은 쿼리 내의 객체를 가상화하는 방법, 시스템 및 컴퓨터 프로그램 제품에 관한 것이다. 몇몇 실시예들에서, 전체 객체를 구성하지 않고 객체의 일부분들이 액세스될 수 있도록 객체가 생성된다. 쿼리 평가기(query evaluator)는 데이터 구성문(data construction statement)(예를 들어, 객체 쿼리 단편)을 수신한다. 데이터 구성문은, 데이터 구성문이 그 구성을 표현하고 있는 객체의 유형의 표시와 하나 이상의 속성 식별자(property identifier)를 포함한다. 각각의 속성 식별자는 객체의 속성을 식별해준다. 데이터 구성문은 또한 하나 이상의 대응하는 매핑도 포함하고 있다. 각각의 대응하는 매핑은 속성 식별자를 쿼리 식(query expression)에 매핑한다. 쿼리 식은 속성 식별자에 의해 식별되는 객체의 속성에 대한 값을 어떻게 계산해야 하는지를 나타낸다.
- [0015] 쿼리 평가기는 데이터 구성문을 수신한 것에 응답하여 데이터 제공자(data provider)를 생성한다. 데이터 제공자는 데이터 구성문에 의해 기술되는 객체의 속성들이, 전체 데이터 구성문을 평가할 필요없이 또는 전체 객체를 완전히 구성할 필요없이, 액세스될 수 있도록 데이터 액세스를 가상화한다. 몇몇 실시예들에서, 데이터 제공자는 데이터 구성문에 대한 참조를 포함한다. 이 참조는 과도적으로 데이터 구성문의 유형, 속성 식별자 및 대응하는 매핑에 대한 역참조(reference back)를 포함한다.
- [0016] 다른 실시예들에서, 쿼리 평가기는 쿼리에 응답한다. 쿼리 평가기는 속성 값 액세스 쿼리를 수신한다. 속성 값 액세스 쿼리는 이전에 수신된 데이터 구성문에서 식별된 속성의 속성 값에 대해 질의를 한다. 쿼리 평가기는 속성 값 액세스 쿼리에 포함된 데이터에 기초하여 이전에 수신된 데이터 구성문으로부터 생성된 데이터 제공자를 식별한다. 데이터 제공자는 데이터 구성문에 의해 기술되는 객체의 속성들이, 전체 데이터 구성문을 평가할 필요없이 또는 전체 객체를 완전히 구성할 필요없이, 액세스될 수 있도록 데이터 액세스를 가상화한다.
- [0017] 쿼리 평가기는 식별된 데이터 제공자를 쿼리 평가기와 연관된 기지의 장소(예를 들어, 쿼리 평가기에 의해 사용되는 스택)에 위치시킨다. 쿼리 평가기는 속성 값 액세스 쿼리에 대한 데이터 소비자(data consumer)를 생성한다. 데이터 소비자는 기지의 장소로부터의 식별된 데이터 제공자에 액세스한다.
- [0018] 데이터 소비자는 데이터 제공자에게 식별된 속성의 속성 값을 요청한다. 데이터 소비자는 데이터 제공자로부터 식별된 속성의 속성 값을 수신한다. 데이터 소비자는 추가적인 처리를 위해 수신된 속성 값을 기지의 장소에 위치시킨다.
- [0019] 이 요약은 이하에서 상세한 설명에 더 기술되는 일련의 개념들을 간략화된 형태로 소개하기 위해 제공된 것이다. 이 요약은 청구된 발명 대상의 주요 특징들 또는 필수적인 특징들을 확인하기 위한 것이 아니며, 청구된 발명 대상의 범위를 정하는 데 보조 수단으로 사용되기 위한 것도 아니다.
- [0020] 본 발명의 부가의 특징들 및 이점들이 이하의 설명에서 기술될 것이며, 부분적으로 이 설명으로부터 명백하게 되거나 본 발명의 실시예에 의해 알게 될 수 있다. 본 발명의 특징들 및 이점들이 첨부된 청구항들에 특정되어 있는 수단들 및 조합들에 의해 실현되고 획득될 수 있다. 본 발명의 이들 및 기타 특징들이 이하의 설명 및 첨부된 특허청구범위로부터 더욱 명백하게 되거나, 이후에 설명되는 바와 같이 본 발명의 실시예에 의해 알게 될 수 있다.

도면의 간단한 설명

- [0021] 본 발명의 상기한 이점들 및 특징들과 기타 이점들 및 특징들이 달성될 수 있는 방식을 설명하기 위해, 이상에서 간략히 설명한 본 발명의 보다 상세한 설명이 첨부 도면에 도시되어 있는 본 발명의 구체적인 실시예들을 참조하여 행해질 것이다. 이들 도면이 본 발명의 통상적인 실시예들만을 도시하고 있고 따라서 본 발명의 범위를 제한하는 것으로 간주되어서는 안된다는 것을 염두에 두고서, 첨부 도면을 사용하여 보다 구체적이고 상세하게 본 발명에 대해 기술하고 설명할 것이다.

도 1a는 데이터 액세스를 가상화하는 객체를 생성하는 것을 용이하게 해주는 예시적인 컴퓨터 아키텍처를 나타낸 도면.

도 1b는 쿼리에 응답하는 것을 용이하게 해주는 추가적인 구성요소를 포함하는 도 1a의 예시적인 컴퓨터 아키텍처를 나타낸 도면.

도 1c는 쿼리에 응답하는 것을 용이하게 해주는 추가적인 구성요소를 포함하는 도 1b의 예시적인 컴퓨터 아키텍처를 나타낸 도면.

도 2는 데이터 액세스를 가상화하는 객체를 생성하는 예시적인 방법의 플로우차트.

도 3은 쿼리에 응답하는 예시적인 방법의 플로우차트.

발명을 실시하기 위한 구체적인 내용

[0022]

본 발명은 쿼리 내의 객체를 가상화하는 방법, 시스템 및 컴퓨터 프로그램 제품에 관한 것이다. 몇몇 실시예들에서, 전체 객체를 구성하지 않고 객체의 일부분들이 액세스될 수 있도록 객체가 생성된다. 쿼리 평가기(query evaluator)는 데이터 구성문(data construction statement)(예를 들어, 객체 쿼리 단편)을 수신한다. 데이터 구성문은, 데이터 구성문이 그 구성을 표현하고 있는 객체의 유형의 표시와 하나 이상의 속성 식별자(property identifier)를 포함한다. 각각의 속성 식별자는 객체의 속성을 식별해준다. 데이터 구성문은 또한 하나 이상의 대응하는 매핑도 포함하고 있다. 각각의 대응하는 매핑은 속성 식별자를 쿼리 식(query expression)에 매핑한다. 쿼리 식은 속성 식별자에 의해 식별되는 객체의 속성에 대한 값을 어떻게 계산해야 하는지를 나타낸다.

[0023]

쿼리 평가기는 데이터 구성문을 수신한 것에 응답하여 데이터 제공자(data provider)를 생성한다. 데이터 제공자는 데이터 구성문에 의해 기술되는 객체의 속성들이, 전체 데이터 구성문을 평가할 필요없이 또는 전체 객체를 완전히 구성할 필요없이, 액세스될 수 있도록 데이터 액세스를 가상화한다. 몇몇 실시예들에서, 데이터 제공자는 데이터 구성문에 대한 참조를 포함한다. 이 참조는 과도적으로 데이터 구성문의 유형, 속성 식별자 및 대응하는 매핑에 대한 역참조(reference back)를 포함한다.

[0024]

다른 실시예들에서, 쿼리 평가기는 쿼리에 응답한다. 쿼리 평가기는 속성 값 액세스 쿼리를 수신한다. 속성 값 액세스 쿼리는 이전에 수신된 데이터 구성문에서 식별된 속성의 속성 값에 대해 질의를 한다. 쿼리 평가기는 속성 값 액세스 쿼리에 포함된 데이터에 기초하여 이전에 수신된 데이터 구성문으로부터 생성된 데이터 제공자를 식별한다. 데이터 제공자는 데이터 구성문에 의해 기술되는 객체의 속성들이, 전체 데이터 구성문을 평가할 필요없이 또는 전체 객체를 완전히 구성할 필요없이, 액세스될 수 있도록 데이터 액세스를 가상화한다.

[0025]

쿼리 평가기는 식별된 데이터 제공자를 쿼리 평가기와 연관된 기지의 장소(예를 들어, 쿼리 평가기에 의해 사용되는 스택)에 위치시킨다. 쿼리 평가기는 속성 값 액세스 쿼리에 대한 데이터 소비자(data consumer)를 생성한다. 데이터 소비자는 기지의 장소로부터의 식별된 데이터 제공자에 액세스한다.

[0026]

데이터 소비자는 데이터 제공자에게 식별된 속성의 속성 값을 요청한다. 데이터 소비자는 데이터 제공자로부터 식별된 속성의 속성 값을 수신한다. 데이터 소비자는 추가적인 처리를 위해 수신된 속성 값을 기지의 장소에 위치시킨다.

[0027]

본 발명의 실시예들은, 이하에서 더 상세히 설명하는 바와 같이, 컴퓨터 하드웨어를 포함하는 전용 또는 범용 컴퓨터를 포함하거나 이용할 수 있다. 본 발명의 범위 내의 실시예들은 또한 컴퓨터 실행가능 명령어 및/또는 데이터 구조를 전달 또는 저장하는 물리 및 기타 컴퓨터 판독가능 매체를 포함한다. 이러한 컴퓨터 판독가능 매체는 범용 또는 전용 컴퓨터 시스템에 의해 액세스될 수 있는 매체라면 어느 것이라도 될 수 있다. 컴퓨터 실행가능 명령어를 저장하는 컴퓨터 판독가능 매체는 물리 저장 매체이다. 컴퓨터 실행가능 명령어를 전달하는 컴퓨터 판독가능 매체는 전송 매체이다. 따라서, 제한이 아닌 예로서, 본 발명의 실시예들은 적어도 2개의 서로 다른 종류의 컴퓨터 판독가능 매체, 즉 물리 저장 매체 및 전송 매체를 포함할 수 있다.

[0028]

물리 저장 매체는 컴퓨터 실행가능 명령어 또는 데이터 구조 형태의 원하는 프로그램 코드 수단을 저장하는 데 사용될 수 있고 또 범용 또는 전용 컴퓨터에 의해 액세스될 수 있는 RAM, ROM, EEPROM, CD-ROM 또는 기타 광 디스크 저장 장치, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 임의의 다른 매체를 포함한다.

[0029]

"네트워크"는 컴퓨터 시스템들 및/또는 모듈들 및/또는 기타 전자 장치들 간의 전자 데이터의 전송을 가능하게 해주는 하나 이상의 데이터 링크로서 정의된다. 정보가 네트워크 또는 다른 통신 연결(유선, 무선 또는 유선과 무선의 조합)을 통해 컴퓨터로 전송 또는 제공될 때, 컴퓨터는 적절하게도 이 연결을 전송 매체로 본다. 전송 매체는 컴퓨터 실행가능 명령어 또는 데이터 구조의 형태의 원하는 프로그램 코드 수단을 전달하는 데 사용될 수 있는 또 범용 또는 전용 컴퓨터에 의해 접속될 수 있는 네트워크 및/또는 데이터 링크를 포함할 수 있다. 상기한 것들의 조합도 역시 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

- [0030] 게다가, 다양한 컴퓨터 시스템 구성요소들에 도달할 때, 컴퓨터 실행가능 명령어 또는 데이터 구조 형태의 프로그램 코드 수단이 전송 매체로부터 물리 저장 매체로(또는 그 반대로) 자동으로 전달될 수 있다. 예를 들어, 네트워크 또는 데이터 링크를 통해 수신되는 컴퓨터 실행가능 명령어 또는 데이터 구조가 네트워크 인터페이스 모듈(예를 들어, "NIC") 내의 RAM에 버퍼링될 수 있고, 그 다음에 궁극적으로 컴퓨터 시스템 RAM 및/또는 컴퓨터 시스템에 있는 저휘발성(less volatile) 물리 저장 매체로 전달될 수 있다. 따라서, 물리 저장 매체가 전송 매체도 역시(또는 심지어 전송 매체를 주로) 이용하는 컴퓨터 시스템 구성요소에 포함될 수 있다는 것을 잘 알 것이다.
- [0031] 컴퓨터 실행가능 명령어는, 예를 들어, 범용 컴퓨터, 전용 컴퓨터 또는 전용 처리 장치로 하여금 어떤 기능 또는 어떤 일군의 기능들을 수행하게 하는 명령어 및 데이터를 포함한다. 컴퓨터 실행가능 명령어는, 예를 들어, 바이너리, 어셈블리어 등의 중간 포맷 명령어(intermediate format instruction), 또는 심지어 소스 코드일 수 있다. 발명 대상이 구조적 특징들 및/또는 방법적 동작들과 관련하여 기술되어 있지만, 첨부된 특허청구범위에 정의된 발명 대상이 상기한 기술된 특징들 또는 동작들로 반드시 제한되는 것은 아니라는 것을 잘 알 것이다. 오히려, 기술된 특징들 및 동작들은 청구항들을 구현하는 예시적인 형태로서 개시되어 있다.
- [0032] 당업자라면 본 발명이 개인용 컴퓨터, 데스크톱 컴퓨터, 랩톱 컴퓨터, 메시지 프로세서, 핸드헬드 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 이동 전화, PDA, 페이지, 라우터, 스위치, 기타 등등을 비롯한 많은 유형의 컴퓨터 시스템 구성을 갖는 네트워크 컴퓨팅 환경에서 실시될 수 있다는 것을 잘 알 것이다. 본 발명은 또한 네트워크를 통해 (유선 데이터 링크, 무선 데이터 링크, 또는 유선 및 무선 데이터 링크의 조합에 의해) 연결되어 있는 로컬 및 원격 컴퓨터 시스템 둘다가 작업을 수행하는 분산 시스템 환경에서 실시될 수 있다. 분산 시스템 환경에서는, 프로그램 모듈들이 로컬 및 원격 메모리 저장 장치 둘다에 위치할 수 있다.
- [0033] 도 1a는 데이터 액세스를 가상화하는 객체를 생성하는 것을 용이하게 해주는 예시적인 컴퓨터 아키텍처(100)를 나타낸 것이다. 도 1b는 쿼리에 응답하는 것을 용이하게 해주는 추가적인 구성요소를 포함하는 컴퓨터 아키텍처(100)를 나타낸 것이다. 도 1c는 쿼리에 응답하는 것을 용이하게 해주는 추가적인 구성요소를 포함하는 컴퓨터 아키텍처(100)를 나타낸 것이다. 따라서, 컴퓨터 아키텍처(100)는 쿼리 평가기(101), 데이터 저장소(102) 및 캐시(103)를 포함한다. 객체 생성 및/또는 쿼리 평가 동안 서로 다른 때에, 컴퓨터 아키텍처(100)는 또한 하나 이상의 데이터 소비자[예를 들어, 데이터 소비자(106(도 1b), 107(도 1c) 및 108(도 1c)) 등] 및 하나 이상의 데이터 제공자[예를 들어, 데이터 제공자(171)(도 1a 및 도 1b) 및 데이터 제공자(173B)(도 1c) 등]를 포함할 수 있다.
- [0034] 도시된 구성요소들 각각은, 예를 들어, 근거리 통신망(LAN), 원거리 통신망(WAN), 또는 심지어 인터넷 등의 네트워크(또는 그의 일부)를 통해 서로 연결될 수 있다. 따라서, 도시된 구성요소들 각각은 물론 임의의 다른 연결된 구성요소들은 메시지 관련 데이터를 생성하고 메시지 관련 데이터[예를 들어, 인터넷 프로토콜(IP) 데이터그램 및 IP 데이터그램을 이용하는 기타 상위 계층 프로토콜(TCP(Transmission Control Protocol), HTTP(Hypertext Transfer Protocol), SMTP(Simple Mail Transfer Protocol), 기타 등등)]를 네트워크를 통해 교환할 수 있다.
- [0035] 본 발명의 실시예들은 또한 일반적으로 요청/응답 메시지의 쌍을 포함하는 프로토콜을 구현한다. 요청이 데이터 소비자로부터 데이터 제공자로 보내지고, 이어서 응답이 데이터 제공자로부터 데이터 소비자로 반환된다. 요청은 액세스될 속성에 대한 식별자와, 소비자 및 제공자 사이의 보조 프로토콜의 유지를 위한 가능한 기타 정보를 포함한다. 응답은 액세스된(데이터 객체/프로토콜 제공자의 다른 가상화된 표현일 수 있는) 속성의 값과, 제공자와 소비자 사이의 보조 프로토콜의 유지를 위한 기타 정보를 포함한다.
- [0036] 도 1을 참조하면, 쿼리 평가기(101)는 일반적으로 표현식으로부터 컴파일된 일련의 명령어들을 실행하도록 구성된다. 쿼리 평가기는 각각의 명령어가 실행될 때 그 명령어가 읽고 업데이트할 수 있는 일련의 내부 상태를 유지할 수 있다. 일반적으로, 명령어는 쿼리 평가기(101)의 현재 상태에서부터 이전의 명령어들의 결과를 읽고, 결과를 추가적인 명령어들이 이용할 수 있도록, 결과를 다시 쿼리 평가기(101)의 현재 상태에 기입한다.
- [0037] 쿼리 평가기(101)가 객체 생성 명령어(예를 들어, new Order() { ... })를 만날 때마다, 쿼리 평가기는 데이터 제공자를 생성하고 데이터 제공자가 가상화하고 있는 객체의 유형에 관한 정보(예를 들어, 속성 식별자들의 목록)로 데이터 제공자를 채운다. 데이터 제공자는 이어서 평가기의 현재 상태에 위치되며, 이에 따라 데이터 제공자가 추가적인 명령어들에 의해 이용될 수 있게 된다. 쿼리 평가기(101)가 속성 액세스 명령어(예를 들어, order.ItemCount)를 만날 때마다, 쿼리 평가기는 데이터 소비자를 생성하고, 쿼리 평가기(101)의 현재 상태로부

터 검색된 제공자와 함께 속성의 식별자를 데이터 소비자에게 전달한다. 데이터 소비자의 평가의 결과는 데이터 평가기(101)의 현재 상태에 위치되고, 여기서 이 결과가 추가적인 명령어들에 의해 사용될 수 있다.

[0038] 그에 따라, 쿼리 평가기(101)는 일반적으로 데이터 구성문을 수신하고 데이터 구성문으로부터 데이터 제공자를 생성하도록 구성되어 있다.

[0039] 몇몇 실시예들에서, 데이터 구성문은 구문 트리(syntax tree)에 따라 정의된 객체를 표현한다. 예를 들어, 데이터 구성문(151)은 구문 트리(157)에 따라 정의된 주문 객체(161)를 표현한다. 일반적으로, 구문 트리[때때로 AST(abstract syntax tree)라고도 함]는 각각의 내부 노드가 구성체(construct)를 나타내고 그 노드의 자식들이 구성체의 의미있는 구성요소를 나타내는 유향 트리이다. 예를 들어, 구문 트리(157)를 참조하면, ShippingAddress 속성(163B)의 자식은 Address 구성체이다. Street 속성(163B1), City 속성(163B2), State 속성(163B3), 및 ZipCode 속성(163B4)은 ShippingAddress 속성(163B)의 Address 구성체 자식의 의미있는 구성요소를 나타낸다.

[0040] 구문 트리(157)은 다음과 같은 것과 유사한 쿼리 단편을 나타낼 수 있다.

```
Order order =
    new Order()
    {
        ItemCount = items.Count(),
        ShippingAddress =
            new Address()
            {
                Street = "123 Abc Lane",

                City = "Tukwila",
                State = "WA",
                ZipCode = LookupZipCode(...)
            },
        TotalCost =
            (from it in items
             select item.Price).Sum()
    }
```

[0041]

[0042]

[0043] 따라서, 쿼리 평가기(101)는 데이터 구성문(151)을 수신하고 데이터 구성문(151)으로부터 데이터 제공자(171)를 생성할 수 있다. 데이터 제공자(171)는 객체(161)에 대한 데이터 액세스를 가상화한다. 데이터 액세스를 가상화하는 것은, 데이터 구성문(151)을 전체적으로 평가할 필요없이[즉, 구문 트리(157) 내의 각각의 속성에 대한 실제 데이터 값을 획득할 필요없이], 객체(161)의 속성들에 대한 액세스를 제공하는 것을 용이하게 해준다. 데이터 액세스를 가상화하는 것은 또한, 객체(161)를 완전히 구성할 필요없이, 객체(161)의 속성들에 대한 액세스를 제공하는 것을 포함한다.

[0044] 데이터 저장소(102)는 데이터를 저장하는 데이터베이스 또는 기타 데이터 저장 장소를 나타낸다. 데이터 구성문은 데이터 저장소(101)에 데이터를 추가하거나, 데이터 저장소(101)로부터의 데이터에 대한 뷰를 제공하거나, 또는 데이터 저장소(101)에 저장된 데이터를 다른 방식으로 수정하도록 구성될 수 있다.

[0045] 일반적으로, 예를 들어, Order order = new Order() { ItemCount = items.Count(), ... }와 같은, 데이터의 구성을 기술하는 쿼리 식이 평가될 때마다, 데이터 제공자가 [예를 들어, 쿼리 평가기(101)에 의해] 생성된다. 각각의 데이터 제공자는 그가 표현하는 객체의 유형에 관한 정보를 포함한다. 이 정보의 일부는 객체에 관한 포함된 속성들(예를 들어, ItemCount)의 모든 식별자들의 목록이다. 각각의 데이터 제공자는 또한 목록 내의 각각의 속성 식별자와 그 가상화된 인스턴스에 대한 속성의 값을 계산하는 데 사용될 쿼리 식[예를 들어, items.Count()] 간의 연관관계 맵을 포함한다. 데이터 제공자는 또한 (각각의 속성에 대한 결과들의 캐시와 같은) 속성 식별자들에 대한 추가적인 연관관계 맵을 유지할 수 있다.

[0046] 요청 메시지가 데이터 제공자에 수신될 때, 데이터 제공자는 메시지에서부터 속성 식별자를 추출하고 그의 연관관계 맵에서 대응하는 쿼리 식을 탐색한다. 데이터 제공자는 또한 속성에 대한 그의 다른 연관관계 맵으로부터

어떤 정보(예를 들어, 속성에 대한 이전에 캐싱된 값)라도 검색할 수 있다. 데이터 제공자는 또한 요청 메시지에 포함된 어떤 부가적인 정보[예를 들어, 캐싱된 값에 대한 TTL(time-to-live) 사양]라도 사용할 수 있다.

- [0047] 데이터 제공자는 이어서 속성에 대한 쿼리 식을 처리하기 위해 쿼리 평가기(101)를 호출할 수 있으며, 어쩌면 그의 내부 맵 및/또는 요청 메시지에서부터의 임의의 보조 정보를 다음으로 전달할 수 있다. 쿼리 평가의 결과(임의의 보조 정보를 포함함)가 데이터 제공자 내의 연관관계 맵을 업데이트(예를 들어, 쿼리의 결과를 캐싱)하는 데 사용된다. 쿼리의 결과(및 임의의 보조 정보)가 응답 메시지 내에 패키징된다. 응답 메시지는 이어서 데이터 제공자에 대해 요청을 시작했던 클라이언트로 보내진다.
- [0048] 이제 도 2를 참조하면, 도 2는 데이터 액세스를 가상화하는 객체를 생성하는 예시적인 방법(200)의 플로우차트를 나타낸 것이다. 방법(200)은 도 1a에 도시된 컴퓨터 아키텍처(100)의 구성요소 및 데이터와 관련하여 기술될 것이다.
- [0049] 방법(200)은 데이터 구성문을 수신하는 동작(동작 201)을 포함한다. 예를 들어, 쿼리 평가기(101)는 데이터 구성문(151)을 수신할 수 있다. 데이터 구성문(151)이 컴퓨터 아키텍처(100) 내의 어떤 다른 네트워크화된 컴퓨터 시스템(도시 생략)으로부터 수신될 수 있다. 다른 대안으로서, 데이터 구성문(151)은 쿼리 평가기(101)가 쿼리 식을 평가한 결과로서 수신될 수 있다.
- [0050] 데이터 구성문은, 데이터 구성문이 그 구성을 표현하고 있는 객체의 유형의 표시를 포함한다. 예를 들어, 데이터 구성문(151)은 데이터 구성문(151)이 주문 객체(161)의 구성을 표현한다는 것을 나타내는 객체 유형(152)을 포함한다. 데이터 구성문(151)은 데이터 저장소(102) 내의 데이터에 대한 뷰를 정의할 수 있다.
- [0051] 데이터 구성문은 또한 하나 이상의 속성 식별자도 포함하고 있다. 예를 들어, 데이터 구성문(151)은 속성 식별자(153)를 포함하고 있다. 각각의 속성 식별자는 객체의 속성을 식별해준다. 예를 들어, 속성 ID(153A)는 ItemCount 속성(163A)을 식별해주고, 속성 ID(153B)는 ShippingAddress 속성(163B)을 식별해주며 속성 ID(153C)는 TotalCost 속성(163C)을 식별해준다.
- [0052] 데이터 구성문은 또한 하나 이상의 대응하는 매핑도 포함하고 있다. 예를 들어, 데이터 구성문(151)은 매핑(154)을 포함하고 있다. 각각의 대응하는 매핑은 속성 식별자를 쿼리 식(query expression)에 매핑한다. 예를 들어, 매핑(154A)은 속성 ID(153A)를 쿼리 식(156A)에 매핑하고, 매핑(154B)은 속성 ID(153B)를 쿼리 식(156B)에 매핑하며, 매핑(154C)은 속성 ID(153C)를 쿼리 식(156C)에 매핑한다. 각각의 쿼리 식은 속성 식별자에 의해 식별되는 객체의 속성에 대한 값을 어떻게 계산해야 하는지를 나타낸다. 예를 들어, 쿼리 식(156A)은 ItemCount 속성(163A)에 대한 값을 어떻게 계산하는지를 나타내고, 쿼리 식(156B)은 ShippingAddress 속성(163B)에 대한 값을 어떻게 계산하는지를 나타내며, 쿼리 식(156C)은 TotalCost 속성(163C)에 대한 값을 어떻게 계산하는지를 나타낸다.
- [0053] 방법(200)은 데이터 구성문을 수신한 것에 응답하여 데이터 제공자를 생성하는 동작(동작 202)을 포함한다. 예를 들어, 쿼리 평가기(101)는 데이터 구성문(151)을 수신한 것에 응답하여 데이터 제공자(171)를 생성할 수 있다. 데이터 제공자(171)는 [데이터 구성문(151)에 의해 기술되는] 주문 객체(161)의 속성들이, 데이터 구성문(151)을 전체적으로 평가할 필요없이 또는 주문 객체(161)를 완전히 구성할 필요없이, 액세스될 수 있도록 데이터 액세스를 가상화한다.
- [0054] 예를 들어, 쿼리 평가기(101)가 ShippingAddress 속성(163B)의 모든 속성들을 평가하는 대신에, 데이터 제공자(171)가 그 대신에 그 속성들(예를 들어, 163B1, 163B2, 163B3, 및 163B4)에 대한 값들이 차후에 요청되는 경우 그 값들이 어떻게 평가될 수 있는지를 나타낸다(예를 들어, 차후의 속성 값 액세스 쿼리에 응답하여 추가적인 데이터 구성문 "new Address()"을 평가함). 게다가, 데이터 제공자(171)는 주문 객체(161)의 부분 구성(partial construction)을 나타낸다. 따라서, 데이터 저장소(102)에 대한 불필요할 수 있는 액세스를 최소화하고 객체(161)의 부분들을 저장하기 위해 불필요할 수 있는 메모리 할당을 최소화함으로써 자원들이 절감된다. 데이터 액세스 및 메모리 할당을 최소화하는 것은 또한 더 효율적인 쿼리 처리를 용이하게 해준다.
- [0055] 몇몇 실시예들에서, 데이터 제공자(예를 들어, 최상위 레벨 객체에 대해 최초로 생성된 데이터 제공자)는 데이터 구성문에 대한 참조를 포함한다. 예를 들어, 데이터 제공자(171)는 데이터 구성문(151)에 대한 참조(151R)를 포함한다. 이 참조는 과도적으로 데이터 구성문의 유형, 속성 식별자 및 대응하는 매핑에 대한 참조(reference)를 포함한다. 예를 들어, 참조(151R)는 과도적으로 객체 유형(152), 속성 식별자(153) 및 매핑(154)에 대한 참조를 포함한다.
- [0056] 이제 도 1b를 참조하면, 도 1b는 쿼리에 응답하는 것을 용이하게 해주는 추가의 구성요소를 포함하는 컴퓨터 아

키텍처(100)를 나타낸 것이다. 따라서, 일반적으로, 쿼리 평가기(101)는 또한, 예를 들어, 속성 값 액세스 쿼리와 같은 쿼리에 응답하도록 구성되어 있다. 속성 값 액세스 쿼리에 응답하여, 쿼리 평가기(101)는 속성 값 액세스 쿼리에 응답할 적절한 데이터 제공자를 식별할 수 있다. 쿼리 평가기(101)는 식별된 데이터 제공자를 스택(104)에 위치시킬 수 있다.

[0057] 쿼리 평가기(101)는 식별된 데이터 제공자와 상호작용할 데이터 소비자를 생성할 수 있다. 데이터 소비자는 스택(104)으로부터의 식별된 데이터 제공자에 액세스하여 속성 식별자를 식별된 데이터 제공자로 보낼 수 있다. 데이터 제공자는 속성 식별자에 대응하는 쿼리 식을 평가하고 결과를 데이터 소비자에게 반환할 수 있다. 데이터 소비자는 결과를 스택(104)에 위치시킬 수 있다. 쿼리 식의 평가는 실제 데이터 값에 액세스하기 위해 데이터베이스(102)를 참조하는 것을 포함할 수 있다.

[0058] 몇몇 실시예들에서, 이전에 생성된 데이터 제공자가 캐시(103)로부터 액세스된다.

[0059] 도 3은 쿼리에 응답하는 예시적인 방법(300)의 플로우차트를 나타낸 것이다. 방법(300)은 도 1b에 도시된 컴퓨터 아키텍처(100)의 구성요소 및 데이터와 관련하여 기술될 것이다.

[0060] 방법(300)은 이전에 수신된 데이터 구성문에서 식별된 속성의 속성 값에 대해 질의하는 속성 값 액세스 쿼리를 수신하는 동작(동작 301)을 포함한다. 예를 들어, 쿼리 평가기(101)는 속성 값 액세스 쿼리(121)를 수신할 수 있다. 속성 값 액세스 쿼리(121)는 데이터 구성문(151)으로부터의 ItemCount 속성(163A)의 속성 값에 대한 쿼리이다. 쿼리 평가기(101)는, 속성 값 액세스 쿼리(121)의 "order" 부분이 주문 객체(161)를 식별해주고 속성 값 액세스 쿼리(121)의 "ItemCount" 부분이 ItemCount 속성(163A)을 식별해주는지를 판정하기 위해 속성 값 액세스 쿼리(121)를 파싱할 수 있다. 따라서, 이 정보로부터, 쿼리 평가기(101)는 또한 속성 ID(153A)가 질의되는 속성에 대응하는지도 판정할 수 있다.

[0061] 방법(300)은 속성 값 액세스 쿼리에 포함된 데이터에 기초하여 이전에 수신된 데이터 구성문으로부터 생성된 데이터 제공자를 식별하는 동작(동작 302)을 포함한다. 예를 들어, 쿼리 평가기(101)는 속성 값 액세스 쿼리(121)에 포함된 용어 "order"에 기초하여 [데이터 구성문(151)으로부터 생성된] 데이터 제공자(171)를 식별할 수 있다.

[0062] 액세스된 데이터 제공자는 데이터 구성문에 의해 기술되는 객체의 속성들이, 전체 데이터 구성문을 평가할 필요 없이 또는 전체 객체를 완전히 구성할 필요없이, 액세스될 수 있도록 데이터 액세스를 가상화한다. 예를 들어, 이전에 기술한 바와 같이, 데이터 제공자(171)는 주문 객체(161)의 속성들에 대한 데이터 액세스를 가상화한다.

[0063] 방법(300)은 쿼리 평가기와 연관된 기지의 장소에 식별된 데이터 제공자를 위치시키는 동작(동작 303)을 포함한다. 예를 들어, 쿼리 평가기(101)는 데이터 제공자(171)를 스택(104)에 위치시킬 수 있다.

[0064] 일반적으로, 속성 값의 액세스를 기술하는 쿼리 식(예를 들어, order.ItemCount 등)이 평가될 때마다, 프로토콜 소비자(protocol consumer)가 [예를 들어, 쿼리 평가기(101)에 의해] 평가된다. 프로토콜 소비자는 데이터 제공자로부터 액세스할 속성의 식별자 및 통신할 데이터 제공자의 인스턴스를 [예를 들어, 쿼리 평가기(101)에 의해] 제공받는다.

[0065] 데이터 소비자는 속성 식별자와 어찌된 쿼리 평가기로부터 입수가능한 부가적인 대역의 정보(예를 들어, 캐싱된 값의 TTL)도 포함하는 요청 메시지를 구성한다. 소비자는 이어서 요청을 제공자에게 보내고 응답을 기다린다. 응답이 데이터 제공자로부터 수신되고 소비자가 속성의 결과 값을 뽑아내어 이를 쿼리 평가기(101)에게 반환할 때, 쿼리 평가기(101)는 추가적인 쿼리에서 소비하기 위해 결과 값을 스택(104)에 위치시킬 수 있다. 데이터 소비자는 또한 응답 메시지 내에 포함된 부가적인 정보에 기초하여 쿼리 평가기(101)의 임의의 부가적인 상태/대역의 정보를 업데이트시킨다.

[0066] 따라서, 방법(300)은 속성 값 액세스 쿼리에 대해 데이터 소비자를 생성하는 동작(동작 304)을 포함한다. 예를 들어, 쿼리 평가기(101)는 데이터 소비자(106)를 생성할 수 있다. 생성된 데이터 소비자는 속성 값이 질의되고 있는 속성에 대응하는 속성 식별자를 전달받을 수 있다. 예를 들어, 쿼리 평가기(101)는 속성 ID(153A)를 데이터 소비자(106)에게 전달할 수 있다[또는 속성 ID(153A)를 포함하도록 데이터 소비자(106)를 구성할 수 있다].

[0067] 방법(300)은 데이터 소비자가 기지의 장소로부터의 식별된 데이터 제공자에 액세스하는 동작(동작 305)을 포함한다. 예를 들어, 데이터 소비자(106)는 스택(104)으로부터의 데이터 제공자(171)에 액세스할 수 있다. 방법(300)은 데이터 소비자가 데이터 제공자에게 식별된 속성의 속성 값을 요청하는 동작(동작 306)을 포함한다.

예를 들어, 데이터 소비자(106)는 ItemCount 속성(163A)의 속성 값을 요청하기 위해 데이터 제공자(171)에게 속성 ID(153A)를 제출할 수 있다.

- [0068] 데이터 제공자(171)는 속성 ID(153A)를 쿼리 식(156A)과 매칭시킬 수 있다. 쿼리 평가기(101)는 이어서 ItemCount 속성(163A)에 대한 값을 생성하기 위해 쿼리 식(156A)을 평가할 수 있다. 예를 들어, 쿼리 평가기(101)는, 값(108)(예를 들어, 주문에 있는 물품의 총수)을 계산하기 위해, 데이터 저장소(102)를 참조하고 데이터(197)에 액세스할 수 있다. 데이터 제공자(171)는 값(108)을 데이터 소비자(106)에게 반환할 수 있다.
- [0069] 방법(300)은 데이터 소비자가 데이터 제공자로부터 식별된 속성의 속성 값을 수신하는 동작(동작 307)을 포함한다. 예를 들어, 데이터 소비자(106)는 데이터 제공자(171)로부터 값(108)을 수신할 수 있다. 방법(300)은 데이터 소비자가 수신된 속성 값을 추가적인 처리를 위해 기지의 장소에 위치시키는 동작(동작 308)을 포함한다. 예를 들어, 데이터 소비자(106)는 값(108)을 스택(104)에 위치시킬 수 있다[예를 들어, 데이터 제공자(171)를 대체시킴]. 따라서, ItemCount 속성(163A)에 대한 속성 값이, 객체(161)를 완전히 생성하지 않고 또 속성들(163C, 162B1, 162B2, 162B3, 및 162B4)에 대한 실제 값을 평가할 필요없이 획득될 수 있다.
- [0070] 도 1c는 쿼리에 응답하는 것을 용이하게 해주는 추가의 구성요소를 포함하는 컴퓨터 아키텍처(300)를 나타낸 것이다. 일반적으로, 쿼리 평가기(101)는 또한 수신된 속성 값 액세스 쿼리의 처리 동안에 평가되는 추가적인 데이터 구성문으로부터 추가적인 데이터 제공자를 생성하도록 구성되어 있다. 이제 도 1c를 참조하면, 몇몇 실시예들에서, 속성 ID가 추가적인 데이터 구성문인 쿼리 식에 매핑될 수 있다. 예를 들어, 데이터 제공자(171)에서, 속성 ID(153B)가 주소 객체를 생성하는 데이터 구성문 "new Address()"인 쿼리 식(156B)에 매핑된다. 추가적인 데이터 구성문의 평가에 의해, 추가적인 데이터 제공자가 생성될 수 있다. 예를 들어, 쿼리 식(156B)의 평가에 의해, 데이터 제공자(173B)가 생성될 수 있다.
- [0071] 따라서, 자원들을 절감하고 효율적인 쿼리 처리를 용이하게 해주기 위해 그 자체가 객체인 객체의 속성도 역사가상화될 수 있다. 예를 들어, 계속하여 도 1c를 참조하면, 쿼리 평가기는 속성 값 액세스 쿼리(121)를 수신할 수 있다. 속성 값 액세스 쿼리(121)는 데이터 구성문(151)으로부터의 Street 속성(163B1)의 속성 값에 대한 쿼리이다. 쿼리 평가기(101)는, 속성 값 액세스 쿼리(121)의 "order" 부분이 주문 객체(161)를 식별해주고 속성 값 액세스 쿼리(121)의 "ShippingAddress" 부분이 ShippingAddress 속성(163B)을 식별해주며 속성 값 액세스 쿼리(121)의 "Street" 부분이 Street 속성(163B1)을 식별해주는지를 판정하기 위해 속성 값 액세스 쿼리(121)를 파싱할 수 있다. 따라서, 이 정보로부터, 쿼리 평가기(101)는 또한 속성 ID(153B1)가 질의되는 속성에 대응하는지도 판정할 수 있다.
- [0072] 쿼리 평가기(101)는 속성 값 액세스 쿼리(121)에 포함된 항 "order.ShippingAddress.Street"에 기초하여 [데이터 구성문(151)으로부터 생성된] 데이터 제공자(171)를 식별할 수 있다. 쿼리 평가기(101)는 데이터 제공자(171)를 스택(104)에 위치시킬 수 있다.
- [0073] 쿼리 평가기(101)는 데이터 소비자(107)를 생성하고 데이터 소비자에게 속성 ID(153B)를 제공할 수 있다. 속성(153B)으로부터, 데이터 소비자(107)는 ShippingAddress 속성(163B)에 대한 값을 요청할 수 있다. 이것을 하기 위해, 데이터 소비자(107)는 스택(104)으로부터의 데이터 제공자(171)에 액세스할 수 있다. 데이터 소비자(107)는 이어서 ShippingAddress 속성(163B)의 속성 값을 요청하기 위해 데이터 제공자(171)에게 속성 ID(153B)를 제출할 수 있다.
- [0074] 데이터 제공자(171)는 속성 ID(153B)를 쿼리 식(156B)과 매칭시킬 수 있다. 쿼리 평가기(101)는 이어서 쿼리 식(156B)을 평가할 수 있다. 쿼리 식(156B)(추가적인 데이터 구성문)의 평가에 의해 데이터 제공자(173B)가 생성된다. 데이터 제공자(171)가 주문 객체(161)에 대한 데이터 액세스를 가상화하는 방법과 유사하게, 데이터 제공자(173B)는 ShippingAddress 속성(163B)의 값인 Address 객체(162)를 나타내는 객체에 대한 데이터 액세스를 가상화한다. 데이터 제공자(171)는 데이터 제공자(173B)를 데이터 소비자(107)에게 반환할 수 있다. 데이터 소비자(107)는 데이터 제공자(173B)를 수신하고 데이터 제공자(173B)를 스택(104)에 위치시킬 수 있다[예를 들어, 데이터 제공자(171)를 대체시킴].
- [0075] 쿼리 평가기(101)는 데이터 소비자(108)를 생성하고 속성 ID(153B1)를 전달할 수 있다. 속성(153B1)으로부터, 데이터 소비자(108)는 Street 속성(163B1)에 대한 값을 요청할 수 있다. 데이터 소비자(108)는 이어서 스택(104)으로부터의 데이터 제공자(173B)에 액세스할 수 있다. 데이터 소비자(108)는 Street 속성(163B1)의 속성 값을 요청하기 위해 데이터 제공자(173B)에게 속성 ID(153B1)를 제출할 수 있다.
- [0076] 데이터 제공자(173B)는 속성 ID(153B1)를 쿼리 식(156B1)과 매칭시킬 수 있다. 쿼리 평가기(101)는 이어서 쿼리

리 식(156B1)을 평가할 수 있다. 쿼리 식(156B1)은 리터럴 데이터 값 "123 ABC Lane"으로 평가된다. 데이터 제공자(173B)는 리터럴 데이터 값 "123 ABC Lane"를 데이터 소비자(108)에게 반환한다. 데이터 소비자(108)는 리터럴 데이터 값 "123 ABC Lane"을 수신하고 리터럴 데이터 값 "123 ABC Lane"을 스택(104)에 위치시킬 수 있다[예를 들어, 데이터 제공자(173B)를 대체시킴].

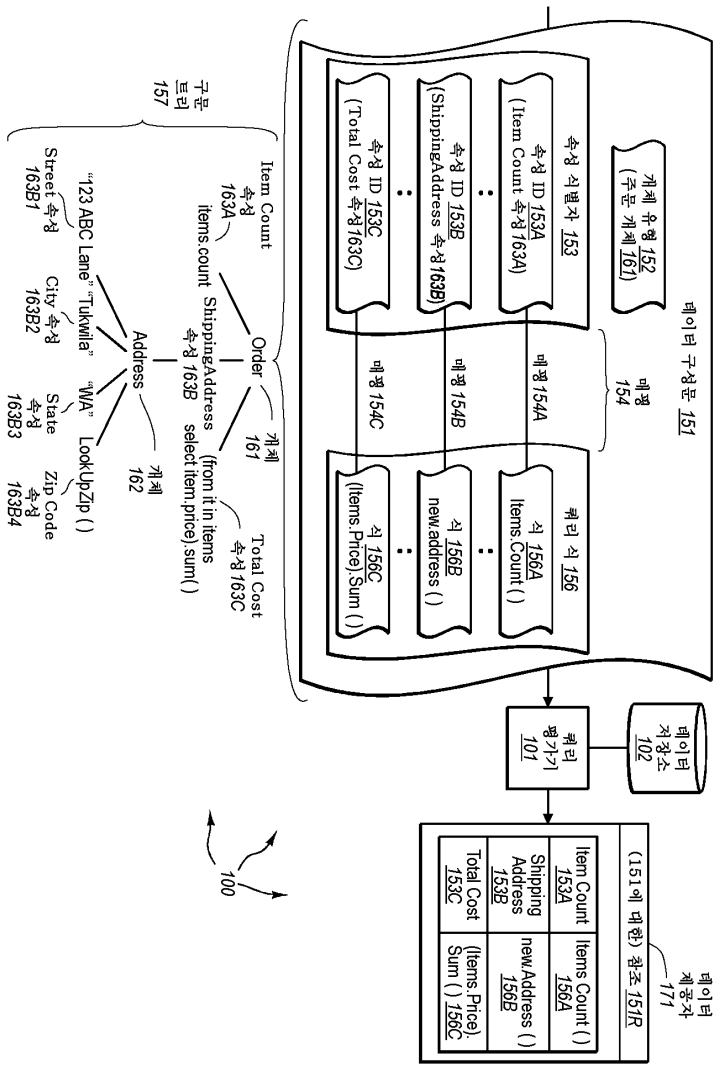
[0077] 몇몇 실시예들에서, 대역외 데이터가 합성된 쿼리를 통해 전송된다. 대역외 데이터는 쿼리들 간의 내부 프로토콜을 설정 및 유지하는 데 사용될 수 있다. 예를 들어, TTL(time-to-live) 값은 얼마동안 데이터 제공자가 캐싱되어야 하는지를 결정하는 데 사용될 수 있다. 예를 들어, 데이터 소비자는, 얼마동안 캐싱된 채로 있어야 하는지를 데이터 제공자에게 알려주기 위해, 요청에 TTL 값(예를 들어, 250 ms)을 포함시킬 수 있다.

[0078] 따라서, 본 발명의 실시예들은, 쿼리에서 사용하기 위한 데이터 액세스를 가상화한다. 가상화는 구문 트리의 임의의 부분 내에서 구현될 수 있다. 예를 들어, 그 자체가 다른 객체인 객체의 속성에 대한 데이터가 가상화될 수 있다. 데이터 가상화는 쿼리 식의 느린 평가를 용이하게 해준다. 즉, 쿼리가 실제 속성 값을 구체적으로 요청할 때까지, 데이터 구성문 내의 속성들에 대한 실제 속성 값이 가상화된다. 게다가, 앞서 기술한 바와 같이, 데이터 가상화는 또한 자원들을 절감하고 그 결과 쿼리 평가가 보다 효율적으로 된다.

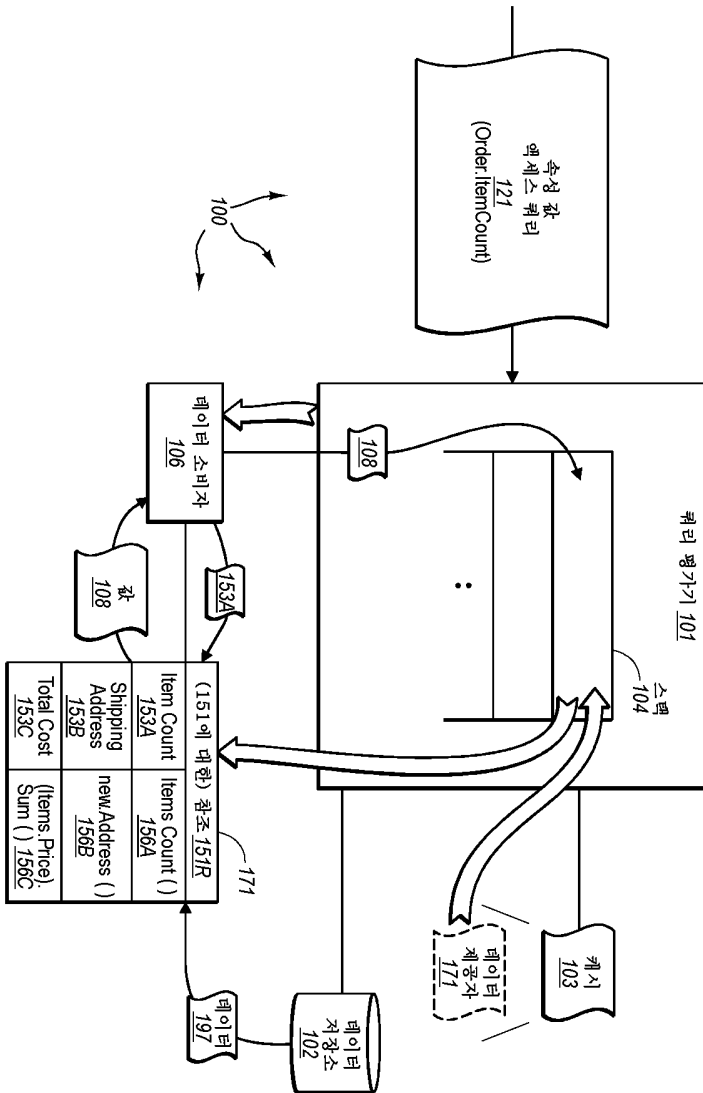
[0079] 본 발명은 그의 정신 또는 본질적인 특성들을 벗어나지 않고 다른 특정의 형태로 구현될 수 있다. 기술된 실시예들이 모든 점에서 제한적이 아니라 단지 예시적인 것으로 간주되어야 한다. 따라서, 본 발명의 범위는 이상의 설명이 아니라 첨부된 특허청구범위에 의해 나타내어진다. 특허청구범위의 등가성의 의미 및 범위 내에 속하는 모든 변경들이 발명의 범위 내에 포함된다.

도면

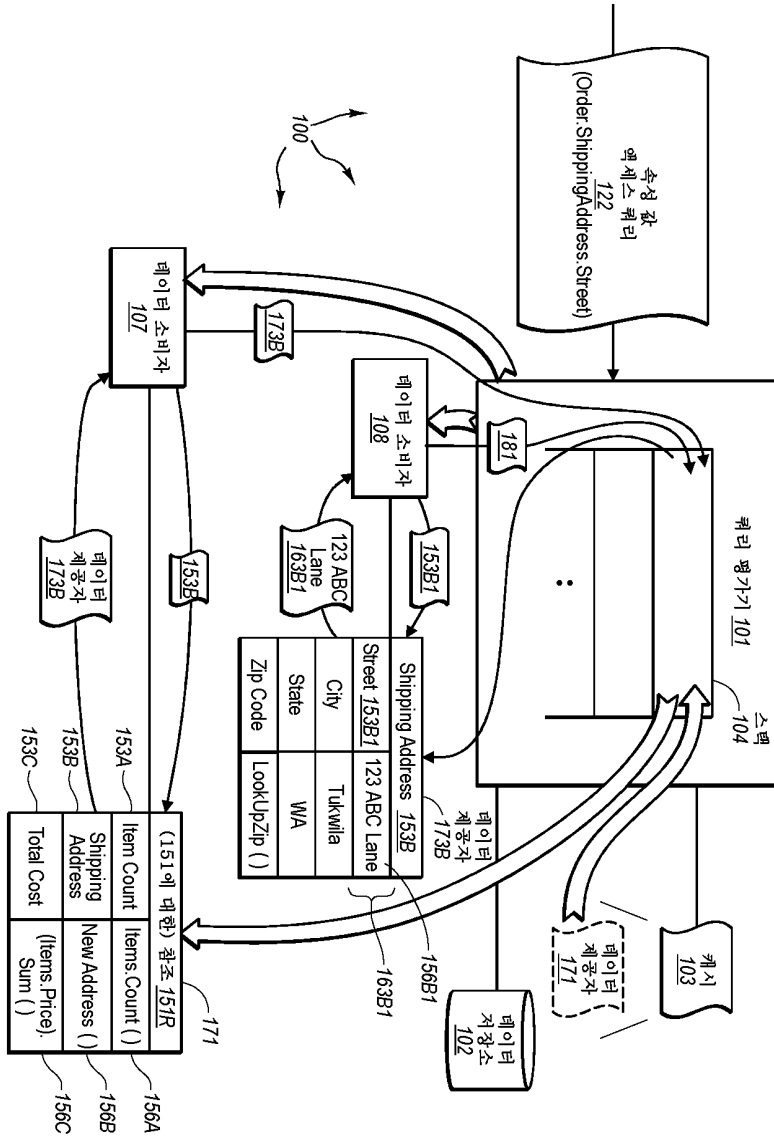
도면1a



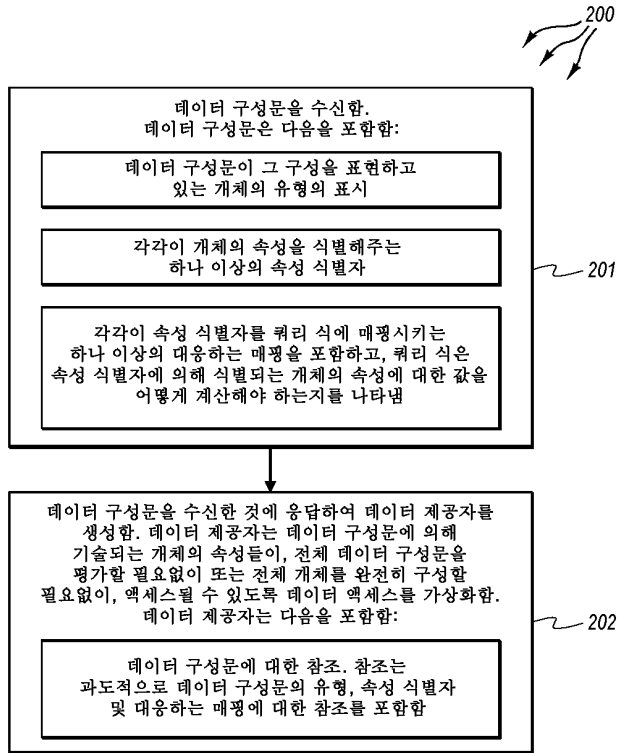
도면1b



도면1c



도면2



도면3

