



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0042292

(43) 공개일자 2015년04월20일

(51) 국제특허분류(Int. Cl.)

G06F 17/30 (2006.01)

(52) CPC특허분류

G06F 17/30017 (2013.01)

G06F 17/30572 (2013.01)

(21) 출원번호 10-2015-7007990(분할)

(22) 출원일자(국제) 2009년12월02일

심사청구일자 없음

(62) 원출원 특허 10-2011-7014656

원출원일자(국제) 2009년12월02일

심사청구일자 2014년12월02일

(85) 번역문제출일자 2015년03월27일

(86) 국제출원번호 PCT/US2009/066390

(87) 국제공개번호 WO 2010/065623

국제공개일자 2010년06월10일

(30) 우선권주장

61/119,201 2008년12월02일 미국(US)

(71) 출원인

아브 이니티오 테크놀로지 엘엘시

미국 02421 매사추세츠주 렉싱턴 스프링 스트리트 201

(72) 발명자

바토르 에릭

미국 02494 매사추세츠주 니드햄 센트럴 애비뉴 314

구드 조엘

미국 02474 매사추세츠주 알링턴 리 테라스 27

라디보예빅 두산

미국 01845 매사추세츠주 노쓰 앤도버 웨이랜드 씨클 140

(74) 대리인

유미특허법인

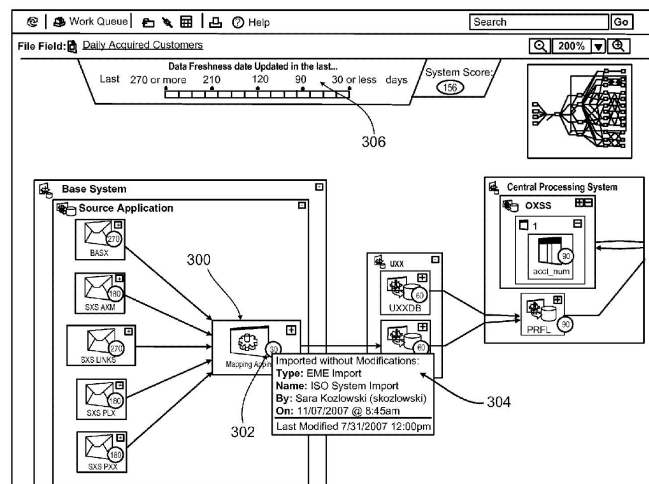
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 데이터 요소 간의 관계의 시각화 및 데이터 요소 속성의 그래픽 표현

(57) 요약

일반적으로, 메타데이터는 데이터 기억 시스템(100)에 기억된다. 데이터 기억 시스템에 기억된 다수의 메타데이터 객체의 각각의 하나 이상의 특징을 식별하는 요약 데이터를 산출하고, 소정의 메타데이터 객체를 특징짓는 요약 데이터는 소정의 메타데이터 객체와 연관시켜 기억된다. 메타데이터 객체를 각각 나타내는 노드(300)와 노드 간의 관계를 포함하는 다이어그램의 시각적 표현(200A)이 생성된다. 시각적 표현은 소정의 메타데이터 객체를 특징짓는 요약 데이터에 의해 식별된 특징의 표현(302)을, 소정의 메타데이터 객체를 나타내는 노드 가까이 수퍼임포스시켜서 생성한다.

대표도



## 명세서

### 청구범위

#### 청구항 1

발명의 상세한 설명에 기재된, 또는 도면에 도시된 바와 같은 장치.

### 발명의 설명

#### 기술 분야

[0001] 본 발명은 데이터 요소(data element) 간의 관계의 시각화 및 데이터 요소 속성의 그래픽 표현에 관한 것이다.

[0002] 관련출원

[0003] 본 출원은 2008년 12월 2일에 출원된 미국출원 61/119,201호에 대하여 우선권을 주장하며, 그 전체 내용을 본 명세서에서 참조에 의해 원용한다.

#### 배경 기술

[0004] 기업은 데이터 웨어하우징, 고객 관계 관리, 및 데이터 마이닝 등의 복잡한 데이터 처리 시스템을 사용해서 데이터를 관리한다. 많은 데이터 처리 시스템에서는, 데이터베이스 파일, 운영 체제, 플랫폼 파일, 인터넷 등과 같은 많은 여러 데이터 소스로부터 중앙의 리포지터리(repository)로 데이터를 가져온다. 데이터는 변환시킨 후에 데이터 시스템에 로딩하는 경우가 많다. 이러한 변환에는 정제(cleansing), 통합, 및 추출이 포함될 수 있다. 데이터, 데이터의 소스, 및 데이터 시스템에 저장된 데이터에 대해 이루어진 변환을 기록하기 위해, 메타데이터(metadata)가 사용될 수 있다. 메타데이터[이를 "데이터에 관한 데이터"(data about data)라고도 함]는 다른 데이터의 속성, 포맷, 기원, 이력, 연관성 등을 설명하는 데이터이다. 메타데이터의 관리는 복잡한 데이터 처리 시스템에서 핵심적인 역할을 할 수 있다.

[0005] 데이터베이스 사용자는 소정의 데이터를 여러 데이터 소스로부터 어떻게 얻는지를 알고 싶어하는 경우가 있다. 예를 들어, 데이터베이스 사용자는 데이터세트(dataset) 또는 데이터 객체(data object)가 어떻게 생성되었는지 또는 데이터세트 또는 데이터 객체를 어떤 소스로부터 가져왔는지를 알고 싶어할 수 있다. 데이터세트를 이러한 데이터세트를 얻는 소스로부터 역으로 추적하는 것을 데이터 계통 추적(data lineage tracing)[또는 "업스트림 데이터 계통 추적"]이라고 한다. 데이터베이스 사용자는 데이터세트가 어떻게 사용되었는지, 예를 들어 어떤 애플리케이션이 해당 데이터세트를 판독하였는지를 알기를 원할 수 있다[이것을 "다운스트림 데이터 계통 추적" 또는 "영향 분석"(impact analysis)이라 함]. 데이터베이스 사용자는 데이터세트가 다른 데이터세트와 어떻게 관련되어 있는지를 아는 것에 관심을 가질 수도 있다. 예를 들어, 사용자는 데이터세트가 수정될 것인지, 어떤 데이터가 영향을 받을 것인지 알고 싶어할 수 있다.

#### 발명의 내용

[0006] 일반적인 관점으로서, 본 발명의 방법은 메타데이터(metadata)를 데이터 기억 시스템(data storage system)에 기억시키는 단계를 포함한다. 데이터 기억 시스템에 기억된 다수의 메타데이터 객체(metadata object)의 각각의 하나 이상의 특징을 식별하는 요약 데이터(summary data)가 산출(compute)되고, 소정의 메타데이터 객체를 특징짓는 요약 데이터를 소정의 메타데이터 객체와 연관시켜 기억한다. 메타데이터 객체를 각각 나타내는 노드(node)와 노드 간의 관계(relationship)를 포함하는 다이어그램(diagram)의 시각적 표현(visual representation)이 생성된다. 시각적 표현은 소정의 메타데이터 객체를 특징짓는 요약 데이터에 의해 식별된 특징의 표현을, 소정의 메타데이터 객체를 나타내는 노드 가까이에 수퍼임포즈(superimpose)시켜서 생성한다.

[0007] 관점은 다음과 같은 특징들 중 하나 이상을 포함할 수 있다. 표현은 메타데이터 객체의 품질(quality)을 나타낸다. 표현은 메타데이터 객체가 최근에 갱신되었는지 여부를 나타낸다. 표현은 소스(source)를 나타내며, 이 소스로부터 제공된 메타데이터 객체가 가장 최근에 갱신된 메타데이터 객체이다. 표현은 이러한 표현을 분류하는 범례(legend)와 연관된다. 시각적 표현 위에 커서(cursor)를 갖다대는 것에 의해, 표현과 관련된 정보를 포함하는 창(window)이 생성된다. 표현은 사용자에 의해 선택될 수 있는 특징을 나타낸다.

[0008] 일반적인 관점으로서, 본 발명의 시스템은, 메타데이터(metadata)를 데이터 기억 시스템(data storage system)에 기억시키기 위한 수단과, 데이터 기억 시스템에 기억된 다수의 메타데이터 객체(metadata object)의 각각의 하나 이상의 특징을 식별하는 요약 데이터(summary data)를 산출(compute)하기 위한 수단을 포함한다. 본 발명의 시스템은 또한 소정의 메타데이터 객체를 특징짓는 요약 데이터를 소정의 메타데이터 객체와 연관시켜 기억하기 위한 수단과, 메타데이터 객체를 각각 나타내는 노드(node)와 노드 간의 관계(relationship)를 포함하는 다이어그램(diagram)의 시각적 표현(visual representation)을 생성하는 수단을 포함한다. 시각적 표현은 소정의 메타데이터 객체를 특징짓는 요약 데이터에 의해 식별된 특징의 표현을, 소정의 메타데이터 객체를 나타내는 노드 가까이 수퍼임포즈(superimpose)시켜서 생성한다.

[0009] 일반적인 관점으로서, 컴퓨터 시스템은 메타데이터를 데이터 기억 시스템에 기억시키고, 데이터 기억 시스템에 기억된 다수의 메타데이터 객체의 각각의 하나 이상의 특징을 식별하는 요약 데이터를 산출하도록 구성된다. 소정의 메타데이터 객체를 특징짓는 요약 데이터는 소정의 메타데이터 객체와 연관시켜 기억되고, 메타데이터 객체를 각각 나타내는 노드와 노드 간의 관계를 포함하는 다이어그램의 시각적 표현이 생성된다. 시각적 표현은 소정의 메타데이터 객체를 특징짓는 요약 데이터에 의해 식별된 특징의 표현을, 상기 소정의 메타데이터 객체를 나타내는 노드 가까이 수퍼임포즈시켜서 생성한다.

[0010] 일반적인 관점으로서, 컴퓨터로 판독가능한 매체는 컴퓨터 프로그램을 기억하며, 컴퓨터 프로그램은, 컴퓨터로 하여금, 메타데이터를 데이터 기억 시스템에 기억시키기 위한 명령어를 포함한다. 데이터 기억 시스템에 기억된 다수의 메타데이터 객체의 각각의 하나 이상의 특징을 식별하는 요약 데이터가 산출되고, 소정의 메타데이터 객체를 특징짓는 요약 데이터는 상기 소정의 메타데이터 객체와 연관시켜 기억된다. 메타데이터 객체를 각각 나타내는 노드와 노드 간의 관계를 포함하는 다이어그램의 시각적 표현이 생성된다. 시각적 표현은 소정의 메타데이터 객체를 특징짓는 요약 데이터에 의해 식별된 특징의 표현을, 소정의 메타데이터 객체를 나타내는 노드 가까이 수퍼임포즈시켜서 생성한다.

[0011] 관점들은 다음과 같은 장점들 중의 하나 이상을 가질 수 있다.

[0012] 본 발명의 시스템에 의하면, 사용자는 객체 간의 관계를 시각화할 수 있으며, 컨텍스트에 따른 설정(contextual setting)으로 객체의 소정의 속성을 뷰(view) 할 수 있다. 메타데이터를 사용하면, 사용자는 어떤 동작을 취하기 전에 객체의 기원(origin)을 이해할 수 있게 된다. 사용자는 임의의 특징의 객체의 조작에 의해 어느 객체가 영향을 받는지 알 수 있다. 사용자는 또한 객체들 간의 관계가 명확하게 표시되는 환경에서 소정의 객체의 속성을 뷰 할 수 있다.

### 도면의 간단한 설명

[0013] 도 1은 계산 시스템의 블록도이다.

도 2a-2e는 데이터의 노드 사이의 관계를 나타내는 도면이다.

도 3-도 5는 데이터의 노드에 수퍼임포즈된 그래픽 오버레이를 나타내는 도면이다.

### 발명을 실시하기 위한 구체적인 내용

[0014] 사용자는 시스템을 사용하여 다양한 데이터 기억 시스템(data storage system)에 기억된, 객체(object) 간의 관계(relationship)를 시각화할 수 있다. 객체 간의 관계는 데이터 객체가 사용되는 애플리케이션에 적합한 다양한 종속성(dependency) 및/또는 연관성(association)을 표현할 수 있다. 이들 기술이 사용될 수 있는 이러한 유형의 시스템 중의 하나로서, 객체가 그래프 기반의 계산 환경의 요소를 표현하는 시스템에 대하여 설명한다.

[0015] 도 1은 그래프 기반의 계산(graph-based computation)을 개발, 실행 및 관리하기 위한 계산 시스템(computing system)(100)의 부분들 간의 상호관계를 나타내는 블록도이다. 그래프 기반의 계산은 방향성 그래프(directed graph)에 의해 표현되는 "데이터 플로우 그래프"(data flow graph)를 사용하여 실현된다. 방향성 그래프는 그래프 내에서 성분(component)을 표현하는 정점과, 그래프 내에서 성분들 사이의 데이터 흐름을 나타내는 방향성 링크 또는 "에지"를 포함한다. 그래픽 개발 환경(GDE: graphic development environment)(102)은 실행가능한 그래프를 특정하고 그래프 성분을 위한 파라미터를 정의하는 사용자 인터페이스(user interface)를 제공한다. 그래픽 개발 환경(GDE)은 아브 이니티오(AB Initio)로부터 입수할 수 있는 COOPERATING SYSTEM® GDE 등이 될 수 있다. GDE(102)는 리포지터리(repository)(104) 및 병렬 운영 환경(parallel operating environment)(106)과 통신을 행한다. 또한, 리포지터리(104)와 병렬 운영 환경(106)에는 사용자 인터페이스 모듈(108)과 관리 요소

(executive)(110)가 연결된다.

- [0016] 예를 들어, 리포지터리(104)는 기본 데이터 기억 요소(base data store)(105A)와 인터페이스 데이터 기억 요소(interface data store)(105B)를 모두 포함한다. 기본 데이터 기억 요소는 테크니컬 메타데이터(technical metadata)를 기억하며, 그래프 및 변환식(transform) 등의 애플리케이션 관련 메타데이터와 함께 애플리케이션을 포함할 수 있다. 테크니컬 메타데이터를 기억하는 것 외에, 기본 데이터 기억 요소는 종속성 분석(dependency analysis)[예를 들어, 이하에 더 상세하게 설명하는, 데이터 계통의 계산]을 포함하는 다양한 종류의 분석을 수행하거나, 이러한 분석의 결과를 수신 및 기억할 수 있다. 일례로, 기본 데이터 기억 요소(105A) 및 인터페이스 데이터 기억 요소(105B)를 합해서 단일의 데이터 기억 요소로 구현해도 된다.
- [0017] 테크니컬 메타데이터는 다양한 기능이 있어서 개발자에게 유용하지만, 많은 경우, 더 높은 레벨의 메타데이터를 분석하고 조작할 필요가 있다. "엔터프라이즈" 메타데이터 또는 "비즈니스" 메타데이터라고도 부르는 이러한 높은 레벨의 메타데이터는 데이터 분석에 자주 사용된다. 비즈니스 메타데이터(business metadata)의 예로서 데이터 스튜어드십(data stewardship)과 데이터 사전(data dictionary)이 있는데, 데이터 스튜어드십은 어느 종업원이 데이터에 책임을 지고 있는지를 나타내는 것이며, 데이터 사전은 파일 및 파일 내의 필드에 대한 비즈니스 정의(business definitions)이다. 비즈니스 메타데이터는 데이터의 기술적 설명의 범위를 넘어서며, 인터페이스 데이터 기억 요소(105B)와 같이, 기본 데이터 기억 요소(105A)와는 별개의 플랫폼(platform)에 기억될 수 있다.
- [0018] 인터페이스 데이터 기억 요소(105B)는 주요 기능으로서 비즈니스 메타데이터를 기억하는 관계형 데이터베이스(relational database)가 될 수 있다. 인터페이스 데이터 기억 요소는 기본 데이터 기억 요소와 통신을 행하여 메타데이터를 추출할 수 있으며, 그래프, 스프레드시트, 논리 모델, 데이터베이스 테이블, 또는 추가의 제3자 데이터 소스 등과 같은 다양한 다른 소스로부터 정보를 인출할 수 있다.
- [0019] 일례로, 기본 데이터 기억 요소(105A)는 그래프 기반의 애플리케이션의 개발 및 실행과, 그래프 기반의 애플리케이션과 다른 시스템(예를 들어, 다른 운영 체제) 사이에서 메타데이터의 교환을 지원하도록 설계된 스케일러블 객체 지향 데이터베이스 시스템(scalable object-oriented database system)이다. 리포지터리(104)는 문서, 레코드 포맷(예를 들어, 테이블 내의 레코드의 필드 및 데이터 타입), 변환 함수, 그래프, 업무 및 모니터링 정보 등의 모든 종류의 메타데이터를 위한 기억 시스템이다. 리포지터리(104)는 또한 외부 데이터 기억 요소(112)에 기억된 데이터를 비롯하여, 계산 시스템(100)에 의해 처리해야 할 실제 데이터를 나타내는 메타데이터 객체도 기억한다. 다양한 소스로부터 메타데이터를 임포트(import)하고 관리하기 위한 요소를 포함하는 리포지터리의 예는, "DATA MAINTENANCE SYSTEM"이란 명칭으로 2008년 12월 2일자로 출원된 공동 계류중인 미국 가 특허출원 61/119,148호에 개시되어 있으며, 상기 특허문헌의 내용을 본 명세서에 참조에 의해 포함한다. 유사한 요소가 리포지터리(104)에 포함될 수 있다.
- [0020] 병렬 운영 환경(106)은 GDE(102)에서 생성된 데이터 플로 그래프의 명세(specification)를 채택하고 그래프에 의해 정의된 프로세스 논리 및 리소스에 대응하는 컴퓨터 명령(computer instruction)을 생성한다. 병렬 운영 환경(106)은 전형적으로 이들 명령을 다수의 프로세서(동일 종류일 필요는 없음)에서 실행시킨다. 적절한 병렬 운영 환경의 예는 COOPERATING SYSTEM® 이다.
- [0021] 사용자 인터페이스 모듈(108)은 리포지터리(104)의 콘텐츠의 웹 브라우저 기반의 뷰(web-browser-based view)를 제공한다. 사용자(103)는, 사용자 인터페이스 모듈(108)을 사용하여, 객체의 브라우즈, 신규 객체의 작성, 기존 객체의 변경, 애플리케이션 파라미터의 변경, 업무(job)의 스케줄링 등을 할 수 있다. 사용자 인터페이스 모듈(108)은 사용자가 리포지터리(104)에 기억된 객체 및 객체 관련 정보를 검색하고 뷰(view) 할 수 있도록 하는 형식 기반의 브라우저 스크린(form-based browser screen)을 생성한다.
- [0022] 리포지터리(104)는 계산 그래프(computation graph)를 구축하기 위한, 그래프 성분을 포함하는 그래프 기반의 애플리케이션을 위한 메타데이터 객체 및 다른 기능적 객체를 포함하는 메타데이터를 기억한다. 앞서 설명한 바와 같이, 리포지터리(104)의 기본 데이터 기억 요소(105A)에 기억된 메타데이터는, 예를 들어 "테크니컬"(technical) 메타데이터(예를 들어, 애플리케이션 관련 비즈니스 규칙, 레코드 포맷, 및 실행 통계)를 포함하며, 인터페이스 데이터 기억 요소(105B)는 업무 기능, 규칙 및 책무의 사용자 정의 문서(user-defined documentation) 등의 비즈니스 메타데이터를 포함할 수 있다.
- [0023] 리포지터리(104)에 메타데이터 객체의 형태로 기억된 정보에 의해, 애플리케이션 및 이러한 애플리케이션에 의해 처리되는 데이터에 관련된 다양한 종류의 분석이 가능하게 된다. 이러한 정보의 서브세트(subset)는 인터페이스 데이터 기억 요소(105B)에 기억될 수 있다. 예를 들어, 나중에 설명하는 바와 같이, 사용자는 데이터 계통



에 관한 문의(소정의 값이 어디서부터 왔는가? 출력 값을 어떻게 계산하였는가? 어느 애플리케이션이 이러한 데이터를 생성하고 의존하는가?)에 대한 답변을 얻을 수 있다. 개발자는 제안된 수정(예를 들어, 이 부분이 변경되면 다른 어떤 것이 영향을 받을 것인가? 이 소스 포맷이 변경되면 어느 애플리케이션이 영향을 받을 것인가?)의 결과를 이해할 수 있다. 사용자 또는 개발자는 테크니컬 메타데이터와 비즈니스 메타데이터를 포함하는 답변에 대한 문의(예를 들어, 어느 그룹이 이러한 데이터를 생성 및 사용하는 것에 대한 책임을 지는가? 누가 이 애플리케이션을 마지막으로 변경하였는가? 그들은 무엇을 변경하였는가?)를 얻을 수 있다.

[0024]

리포지터리(104)는 기억된 메타데이터 객체의 상태를 추적(track)할 수 있다. 리포지터리(104)에 기억된 객체를 버전화(version)함으로써, 지난 주, 지난 달, 또는 작년의 그 상태를 조사할 수 있게 되며, 그것을 현재의 상태와 비교할 수 있게 된다. 리포지터리(104)는 업무 추적(job-tracking)을 수집하거나, 경향 분석(trend analysis)(예를 들어, 우리의 데이터가 얼마나 빨리 성장하는가?)과 용량 산정(capacity planning)(예를 들어, 애플리케이션을 실행하는 데에 얼마나 걸리나? 얼마나 많은 데이터를 처리하였는가와 그 비율은? 애플리케이션이 어떤 리소스를 사용하였는가? 언제 다른 서버를 추가할 필요가 있는가?)을 가능하게 하는 실행 정보를 수집한다.

[0025]

사용자는 기억된 메타데이터에 포함된 및/또는 연관된 정보를, 사용자 인터페이스 모듈(108)을 통해 뷰[임의 선택적으로, 편집(edit)]할 수 있다. 메타데이터 뷰잉 환경(metadata viewing environment)은 사용자 인터페이스 모듈(108)에 의해 제공되는 아이콘 및 아이콘 그룹을 포함하는 다양한 그래픽 표현을 사용하여 다양한 타입의 메타데이터 객체를 디스플레이 상에 표현할 수 있다. 메타데이터 객체는 상이한 타입의 데이터 요소(data element)(예를 들어, 실행가능한 프로그램의 입력 또는 출력으로서 사용되는 데이터) 및/또는 변환식(transformation)[예를 들어, 데이터를 처리 또는 생성하는, 데이터 플로 그래프 등의 데이터 처리 엔티티(data processing entity)와 연관된 임의 타입의 데이터 조작]을 표현할 수 있다. 이러한 메타데이터 뷰잉 환경은 이하에 더 상세하게 설명하는 바와 같이, 메타데이터 객체 또는 메타데이터 객체의 그룹을 나타내는 그래픽 노드(graphical node)를 연결하는 라인으로, 관계를 나타낼 수 있다. 어떤 경우에, 인터페이스 데이터 기억 요소(105B)는 기본 데이터 기억 요소(105A)로부터 또는 다른 데이터 소스로부터 관계[예를 들어, 계통 정보(lineage information)]를 추출할 수 있다. 인터페이스 데이터 기억 요소(105B)는 데이터 계통의 높은 레벨의 요약(summary)을 유지(hold)할 수 있다. 계통 정보(또는 다른 데이터 종속성 분석)는 시스템(100) 내에서 자동으로 계산되거나, 외부 시스템으로부터 또는 수동의 입력으로부터 수신될 수 있다. 예를 들어, 시스템(100)은 코드를 분석하는 사람에 의해 수집되고 준비된 계통 정보를 수신할 수 있다. 계통 정보는 파일로부터 미리 정해진 다양한 포맷 중의 임의의 포맷(예를 들어, 스프레드시트)으로 리포지터리(104)에 제공될 수 있다.

[0026]

도 2a는 메타데이터 뷰잉 환경의 예를 나타낸다. 일례로, 메타데이터 뷰잉 환경은 브라우저의 상단에서 실행되는 인터페이스이다. 도 2a의 예에서, 메타데이터 뷰잉 환경은 데이터 계통도(data lineage diagram)(200A)와 관련된 정보를 표시한다. 메타데이터 뷰잉 환경의 한가지 예는 사용자가 메타데이터를 시각화하고 편집할 수 있는 웹 기반의 애플리케이션이다. 사용자는, 메타데이터 뷰잉 환경에 의해, 기업 내의 어느 곳에서든 표준 웹 브라우저(standard Web browser)를 사용해서 메타데이터를 검색, 분석 및 관리할 수 있다. 메타데이터 객체의 각각의 유형은 하나 이상의 뷰 또는 시각적 표현(visual representation)을 가지고 있다. 도 2a의 메타데이터 뷰잉 환경은 타겟 요소(target element)(206A)에 대한 계통도를 나타낸다.

[0027]

예를 들어, 계통도 또는 계통 다이어그램은 리포지터리(104)에 기억된 메타데이터 객체, 즉 소정의 시작 객체(starting object)가 의존하는 객체(소스) 및 소정의 시작 객체가 영향을 미치는 객체(타겟)를 나타내는 데이터 및/또는 처리 노드에 대한 엔드투엔드(end-to-end) 계통을 표시한다. 본 예에서는, 메타데이터 객체의 2가지 예에 해당하는, 데이터 요소(202A)와 변환식(204A) 사이에 접속이 도시되어 있다. 메타데이터 객체는 계통도에서 노드에 의해 표현된다. 데이터 요소(202A)는, 예를 들어 데이터세트, 데이터세트 내의 테이블, 테이블 내의 컬럼(column), 및 파일, 메시지 및 리포트 내의 필드를 나타낼 수 있다. 변환식(204A)의 예는 데이터 요소의 단일의 출력이 어떻게 생성되는지를 설명하는 실행가능 파일(executable)의 요소이다. 노드 간의 접속은 메타데이터 객체들 간의 관계에 기초한다.

[0028]

도 2b는 도 2a에 나타난 동일한 타겟 요소(206A)에 대한 대응한 계통도(200B)를 나타내지만, 각각의 요소(202B)가 그룹화되어 있고 컨텍스트(context)에 기초한 그룹으로 도시된 점이 도 2a와 다르다. 예를 들어, 데이터 요소(202B)는 데이터세트(208B)(예를 들어, 테이블, 파일, 메시지 및 리포트), 애플리케이션(210B)(그래프, 플랜 및 프로그램 등의 실행가능 파일과, 이들이 동작하는 데이터세트를 포함), 및 시스템(212B)으로 그룹화된다. 시스템(212B)은 데이터 및 데이터를 처리하는 애플리케이션의 기능 그룹으로서, 시스템은 애플리케이션 및 데이터 그룹(예를 들어, 데이터베이스, 파일 그룹, 메시징 시스템, 및 데이터세트 그룹)으로 이루어져 있다. 변환식

(204B)은 실행가능 파일(executables)(214B), 애플리케이션(210B), 및 시스템(212B)으로 그룹화된다. 그래프, 플랜(plan), 또는 프로그램 등의 실행가능 파일(executables)은 데이터세트를 관독 및 기입한다. 어떤 그룹이 확장(expand)되는지와 어떤 그룹이 디폴트(default)에 의해 축소(collapse)되는지에 대한 파라미터를 설정할 수 있다. 이에 의하면, 사용자는 불필요한 레벨의 상세(details)를 제거함으로써 사용자에게 중요한 그룹에 대한 상세만을 볼 수 있다.

[0029] 메타데이터 뷰잉 환경을 사용해서 데이터 계통 계산을 수행하는 것은 여러 가지 이유에 의해 유용하다. 예를 들어, 데이터 요소와 변환식 간의 관계를 계산하고 나타내는 것은, 보고된 값이 소정의 필드 리포트(field report)에 대해 어떻게 산출되었는지를 사용자가 판정하는 데에 도움을 줄 수 있다. 사용자는 어떤 데이터세트가 특정 타입의 데이터를 기억하는지와 어떤 실행가능 파일이 해당 데이터세트에 대해 관독 및 기입하는지를 부할 수 있다. 비즈니스 용어(business terms)의 경우에, 데이터 계통도는 어느 데이터 요소(예를 들어, 컬럼 및 필드)가 소정의 비즈니스 용어(기업 내의 정의)와 연관된 것인지를 나타낼 수 있다.

[0030] 메타데이터 뷰잉 환경에 나타난 데이터 계통도는 사용자에게 영향 분석(impact analysis)으로 도움을 줄 수 있다. 구체적으로, 사용자는 컬럼이나 필드가 데이터세트에 추가된 경우에 어떤 다운스트림 실행가능 파일(downstream executables)이 영향을 받는지와, 누구에게 통지할 필요가 있는지를 알고 싶어할 수 있다. 영향 분석은 소정의 데이터 요소가 사용되는 위치를 판정할 수 있으며, 해당 데이터 요소를 변경한 분지(ramification)를 판정할 수 있다. 마찬가지로, 사용자는 어떤 데이터세트가 실행가능 파일 내에서의 변경에 의해 영향을 받는지, 또는 그것이 생성(production)으로부터 소정의 데이터베이스 테이블을 제거하기에 안전한지 여부를 부할 수 있다.

[0031] 메타데이터 뷰잉 환경을 사용하여 데이터 계통 계산을 수행함으로써 데이터 계통도를 생성하는 것은 비즈니스 용어 관리에 유용하다. 예를 들어, 기업 내의 고용인이 기업, 이들 용어 간의 관계, 및 이러한 용어가 가리키는 데이터 전반에 걸친 비즈니스 용어의 의미에 대해 동의하는 것이 바람직한 경우가 많다. 비즈니스 용어의 일관된 사용은 기업체 데이터의 투명성을 강화하고 비즈니스 요건의 소통을 용이하게 한다. 따라서, 비즈니스 용어의 기반을 이루는 물리적 데이터(physical data)를 발견할 수 있는 위치와, 어떤 비즈니스 로직(business logic)이 계산에 사용되는지를 아는 것이 중요하다.

[0032] 데이터 노드들 간의 뷰잉 관계(viewing relationship)는 메타데이터를 관리하고 유지하는 데에도 도움이 될 수 있다. 예를 들어, 사용자는 누가 메타데이터를 변경하였는지, 어떤 소스(또는 "레코드의 소스")가 메타데이터에 대한 것인지, 또는 외부 소스로부터 메타데이터를 로딩 또는 리로딩(reloading)할 때 어떤 변경이 이루어졌는지를 알고 싶어할 수 있다. 메타데이터를 유지함에 있어서, 지정된 사용자가 메타데이터 객체를 생성하거나, 메타데이터 객체의 특성(예를 들어, 어느 객체의 다른 객체에 대한 설명 및 관계)을 편집하거나, 사용하지 못하는 메타데이터 객체를 삭제할 수 있도록 하는 것이 바람직하다.

[0033] 메타데이터 뷰잉 환경은 객체의 다수의 그래픽 뷰(graphical view)를 제공함으로써, 사용자가 메타데이터를 탐색하고 분석할 수 있도록 한다. 예를 들어, 사용자는 시스템 및 애플리케이션의 콘텐츠를 부할 수 있으며, 임의의 객체의 상세를 탐색할 수 있고, 데이터 계통 뷰(data lineage view)를 사용해서 객체들 간의 관계를 부할 수 있다. 이에 의해, 사용자는 앞서 설명한 데이터 계통 분석 및 영향 분석과 같은 여러 타입의 종속성 분석을 용이하게 수행할 수 있다. 객체의 계층(hierarchy)도 부할 수 있으며, 특정의 객체에 대한 계층을 검색할 수 있다. 객체를 찾았으면, 해당 객체에 대한 북마크(bookmark)를 생성해서, 사용자가 다시 찾는 것을 용이하게 한다.

[0034] 적절한 허가(permission)에 의해, 사용자는 메타데이터 뷰잉 환경에서 메타데이터를 편집(edit)할 수 있다. 예를 들어, 사용자는 객체의 설명(description)을 갱신하거나, 비즈니스 용어를 작성하거나, 객체 간의 관계를 정의하거나(예컨대, 리포트 내의 필드 또는 테이블 내의 컬럼에 비즈니스 용어를 링크시킴), 객체를 이동시키거나(예를 들어, 하나의 애플리케이션으로부터 다른 애플리케이션으로 데이터세트를 이동시킴), 또는 객체를 삭제할 수 있다.

[0035] 도 2c에는, 타겟 요소(206A)에 대한 대응하는 계통도(200C)가 도시되어 있지만, 분석 정도(resolution)의 레벨은 타겟 데이터 요소(206A)에 대한 계산에 참가하는 애플리케이션으로 설정된다. 구체적으로, 타겟 데이터 요소(206A)에 대한 계산에 직접 참가하는 애플리케이션으로서 애플리케이션(202C, 204C, 206C, 208C, 210C)이 도시되어 있다. 사용자가 다른 레벨의 분석 정도(예컨대, 계통도에서 더 구체적으로 또는 덜 구체적으로 표시하기 위해)로 계통도의 임의의 부분을 부하길 원하는 경우에, 사용자는 대응하는 확장/축소 버튼(212C)을 활성화시킬 수 있다.

- [0036] 도 2d는 대응하는 계통도(200D)를 여러 레벨의 분석 정도로 나타낸다. 본 예에서, 확장/축소 버튼(212C)은 사용자에게 의해 활성화되었으며, 메타데이터 뷰잉 환경은 현재는 동일한 계통도를 디스플레이하지만, 애플리케이션(202C)은 애플리케이션(202C) 내에 데이터세트(214D)와 실행가능 파일(216D)을 보여주도록 확장되었다.
- [0037] 도 2e는 대응하는 계통도(200E)를 여러 레벨의 분석 정도로 나타낸다. 본 예에서, 사용자는 커스텀 확장(custom expansion)에 의해 확장된 모든 것을 나타내도록 선택을 행하였다. 최종적인 데이터 소스(예컨대, 업스트림 시스템을 갖지 않음)에 해당하는 임의의 필드 또는 컬럼이 확장된다. 또한, 특정의 플래그 세트를 갖는 필드도 확장된다. 본 예에서, 특정의 플래그는 계통에서의 주요한 중간 포인트에서 데이터세트 및 필드에 설정되고, 하나의 컬럼은 계통이 표시된 컬럼이다. 사용자 인터페이스 모듈(108)은 어느 노드를 축소할 필요가 있는지와 어느 노드가 계통도로부터 전체적으로 배제될 필요가 있는지를 판정한다.
- [0038] 사용자는 자신들의 계통도를 구성할 수도 있다. 예를 들어, 계통도는 메타데이터에서의 주 키(primary key) 또는 외래 키(foreign key) 관계를 따르도록 구성될 수 있다. 계통도로부터 정보를 배제하기 위해 종속성 분석에 필터(filter)를 적용할 수도 있다. 예를 들어, 사용자가 계통도로부터 거부 파일(reject file)에 해당하는 데이터세트를 배제하길 원하는 경우, 사용자는 계통도에서 거부 파일의 디스플레이를 온 및 오프로 토글할 수 있다.
- [0039] 메타데이터 뷰잉 환경에서의 뷰잉 요소 및 관계는 이들을 나타내는 각각의 노드에 관련 정보를 추가함으로써 더 유용하게 될 수 있다. 관련 정보를 노드에 추가하는 한가지 방식의 예는 소정의 노드의 상단에 정보를 오버레이(overlay)시키는 것이다. 이러한 그래픽은 노드에 의해 표현되는 데이터의 몇 가지의 값 또는 특징을 나타낼 수 있으며, 메타데이터 데이터베이스에서의 임의의 특징이 될 수 있다. 이 방식은 정보의 둘 이상의 정상적으로 이질적인 부분(데이터의 노드와 노드에 의해 표현된 데이터의 특징 간의 관계)을 조합하는 장점을 가지며, 유용한 정보를 "컨텍스트"에 넣기 위한 시도를 한다. 예를 들어, 메타데이터 품질, 메타데이터 최신성(freshness), 또는 레코드 정보의 소스와 같은 특징이 데이터 노드 간의 관계의 시각적 표현과 관련해서 디스플레이될 수 있다. 이러한 정보 중 일부는 표 형식으로 액세스될 수 있지만, 사용자가 상이한 데이터 노드들 간의 관계와 함께 데이터의 특징을 뷰할 수 있도록 하는 데에 도움이 될 수 있다. 사용자는 데이터의 어떤 특징이 메타데이터 뷰잉 환경 내에서 데이터 요소 및/또는 변환식 노드의 상단에 나타날 것인지를 선택할 수 있다. 어떤 특징이 나타날 것인지는 디폴트 시스템 세팅에 따라 설정될 수 있다.
- [0040] 도 3의 예에서, 노드(300)는 노드에 의해 표현되는 메타데이터의 최신성에 관련된 정보를 포함하는 그래픽 오버레이(graphical overlay)(302)를 디스플레이한다. "메타데이터 최신성"이라는 말은 메타데이터가 외부 소스로부터 얼마나 최근에 갱신 또는 수정되었는지를 의미한다. 커서(cursor)를 그래픽 오버레이(302) 위에 갖다대면, 그래픽 오버레이(302)에 의해 현재 디스플레이된 특징에 관해 더 구체적인 상세를 포함하는 창(window)(304)이 열릴 수 있다. 그래픽 오버레이는 컬러 코드화될 수 있으며, 그래픽의 여러 컬러는 범례(legend)(306)를 통해 상이한 의미에 매핑된다.
- [0041] 도 4의 예에서, 메타데이터 품질의 레벨을 나타내는 그래픽 오버레이는 노드(400) 상의 오버레이(402)를 포함하는 데이터 요소 노드의 상단에 수퍼임포즈(superimpose)된다. 메타데이터 품질의 측정은, 예를 들어 데이터를 임포트(import) 또는 처리하기 전에 비즈니스 파트너로부터 전송된 주기적(예컨대, 월 단위)인 데이터 피드를 프로파일하기 위해, 비즈니스에 의해 사용될 수 있다. 이것은 비즈니스가 "불량한" 데이터(예를 들어, 임계값보다 높은 유효하지 않은 값의 비율을 가진 데이터)를 검출해서, 원래대로 되돌리기가 어려울 수 있는 액션에 의해 기존의 데이터 기억 요소가 오염되지 않도록 한다. 앞서 설명한 예와 유사하게, 커서를 그래픽 오버레이(402) 위에 갖다대면, 그래픽 오버레이(402)에 의해 현재 디스플레이된 특징에 관한 더 구체적인 상세를 포함하는 창(404)이 열릴 수 있다.
- [0042] 도 5의 예에서, 레코드의 소스의 타입을 나타내는 그래픽 오버레이는 데이터 요소 및 변환식 노드의 상단에 수퍼임포즈된다. 노드(500)는 레코드의 소스가 "액티브 임포트 소스"(Active Import Source)라는 것을 나타내는 오버레이(502)를 갖는다. 이것은 메타데이터가 스프레드시트 등의 소스로부터 자동으로 임포트된 것임을 의미한다. 커서를 그래픽 오버레이(502) 상에 갖다대면, 임포트에 사용되는 파일의 타입(본 예에서는, 엑셀 스프레드시트), 파일의 이름, 파일의 소유자, 및 임포트 날짜 등과 같은 상세를 포함하는 창(504)을 불러올 수 있다. 노드(506)는 레코드의 소스가 "수동으로 유지되는" 것을 나타내는 오버레이(508)를 갖는다. 이것은 메타데이터가 사용자에게 의해 수동으로[예를 들어, 사용자 인터페이스 모듈(108)을 사용해서] 수정되었음을 의미한다. 커서를 그래픽 오버레이(508) 상에 갖다대므로써, 메타데이터를 수정한 사용자의 이름과 수정한 날짜 등과 같은 상세를 포함하는 창(510)을 불러올 수 있다.
- [0043] 시스템(100)의 모듈과 시스템(100)에 의해 수행되는 과정을 포함하는, 상기 설명한 레코드 기억 및 인출

방식은, 컴퓨터에서 실행하기 위한 소프트웨어를 사용해서 구현해도 된다. 예를 들어, 소프트웨어는, 하나 이상의 프로세서, 하나 이상의 데이터 기억 시스템(휘발성 메모리, 비휘발성 메모리 및/또는 기억 요소를 포함), 하나 이상의 입력 장치 또는 포트, 및 하나 이상의 출력 장치 또는 포트를 각각 포함하는 하나 이상의 프로그램된 또는 프로그램가능한 컴퓨터 시스템(분산형, 클라이언트/서버, 또는 그리드 등의 다양한 구조가 될 수 있음)에서 실행하는 하나 이상의 컴퓨터 프로그램에서의 과정을 형성한다. 소프트웨어는, 예를 들어 계산 그래프의 설계 및 구성에 관련된 다른 서비스를 제공하는 더 큰 프로그램의 하나 이상의 모듈을 형성할 수 있다. 그래프의 노드 및 요소는 컴퓨터로 판독가능한 매체에 기억된 데이터 구조 또는 데이터 리포지터리에 기억된 데이터 모델에 부합하는 그와 다르게 구성된 데이터로서 구현될 수 있다.

[0044]

소프트웨어는 범용 또는 전용의 프로그램가능한 컴퓨터에 의해 판독가능한 CD-ROM 등의 기억 매체에 제공되거나, 소프트웨어가 실행되는 컴퓨터에 네트워크의 통신 매체를 통해 전달(전파 신호로 부호화됨)될 수 있다. 모든 기능은 전용의 컴퓨터 또는 코프로세서와 같은 전용의 하드웨어를 사용해서 구현될 수 있다. 소프트웨어는 소프트웨어에 의해 특정되는 계산의 여러 부분이 여러 컴퓨터에 의해 수행되는 분산형 방식으로 구현해도 된다. 이러한 각각의 컴퓨터 프로그램은 본 명세서에서 설명한 과정을 수행하기 위해 기억 매체 또는 장치가 컴퓨터 시스템에 의해 판독될 때에, 컴퓨터를 구성하고 동작시키기 위한, 범용 또는 전용의 프로그램가능한 컴퓨터에 의해 판독가능한 기억 매체 또는 장치(예를 들어, 고체 메모리 또는 매체, 자기 매체 또는 광학 매체)에 기억 또는 다운로드되는 것이 바람직하다. 본 발명의 시스템은 컴퓨터 프로그램으로 구성된, 컴퓨터로 판독가능한 기억 매체로서 구현되는 것으로 간주할 수 있다. 이에 의하면, 기억 매체에 의해, 컴퓨터 시스템은 본 명세서에 개시된 기능을 수행하기 위해 특정되고 미리 정해진 방식으로 동작할 수 있다.

[0045]

본 발명의 실시예에 대하여 많은 예를 들어 설명하였지만, 본 발명의 범위를 벗어남이 없이 다양한 변형이 가능하다는 것을 알 수 있을 것이다. 예를 들어, 상기 설명한 일부 단계는 순서에 의존하지 않을 수 있으며, 따라서 설명한 것과 다른 순서로 수행될 수 있다.

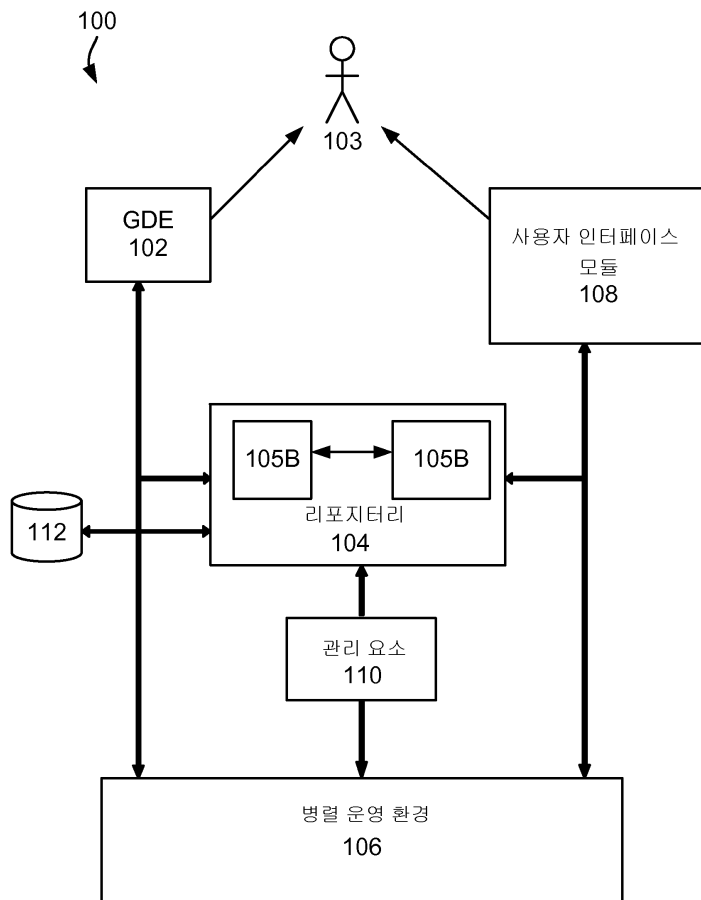
[0046]

이상의 설명은 예시를 위한 것이며 청구범위에 의해 정해지는 본 발명의 범위를 제한하기 위한 것이 아니라는 것을 이해하여야 한다. 예를 들어, 상기 설명된 다수의 기능적 단계는 전체적인 처리 과정에 실질적인 영향을 주지 않으면서 다른 순서로 수행될 수 있다. 다른 실시예도 본 청구범위의 범위에 포함된다.

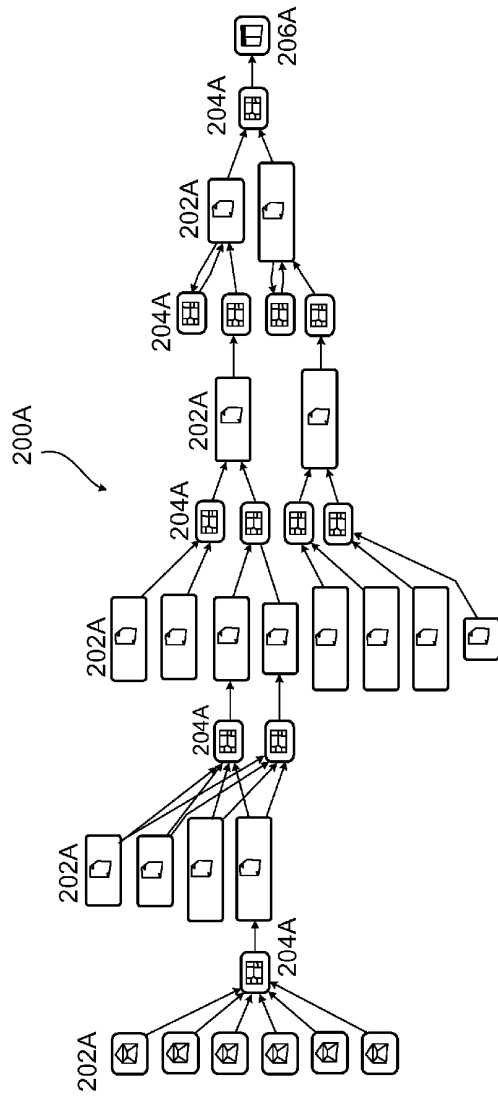


도면

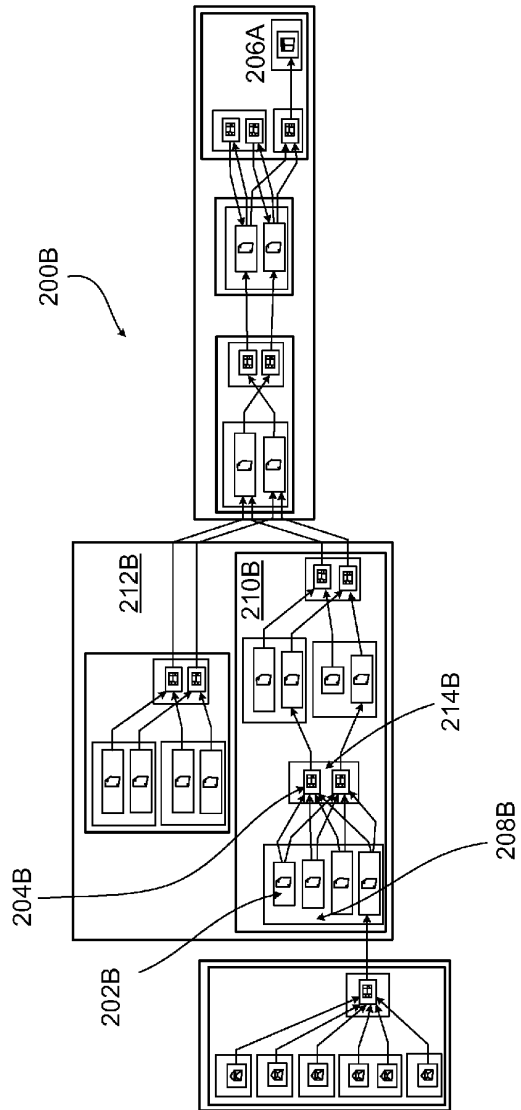
도면1



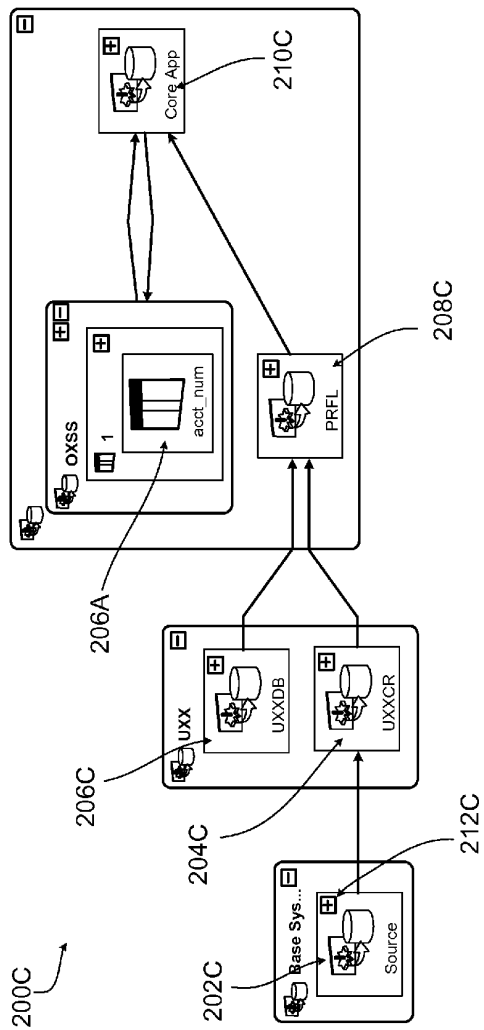
도면2a



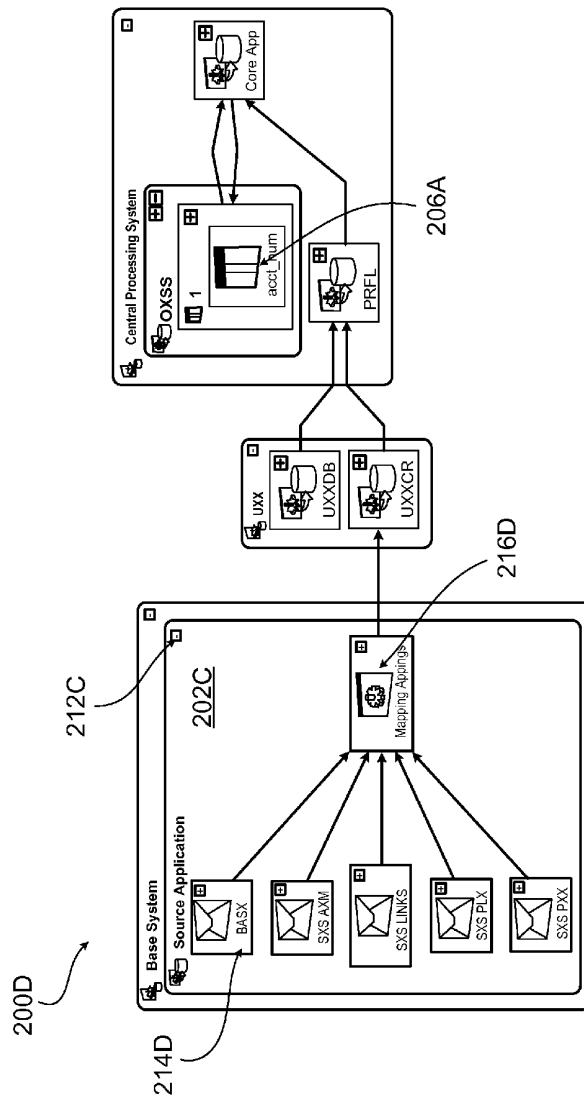
도면2b



도면2c

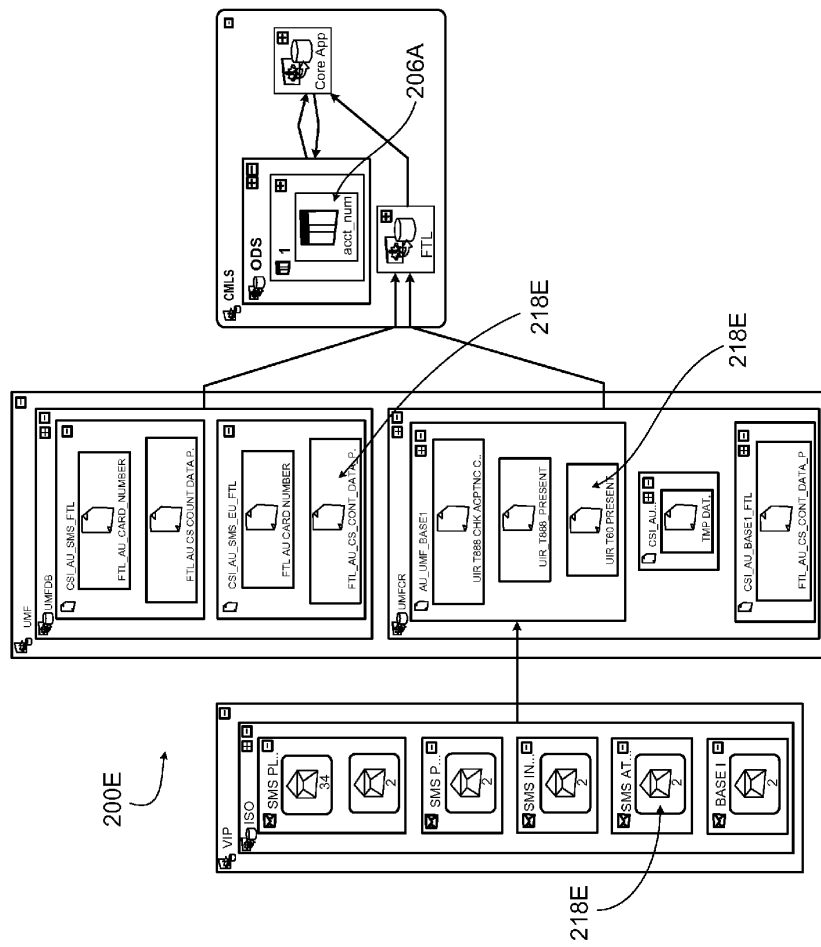


도면2d

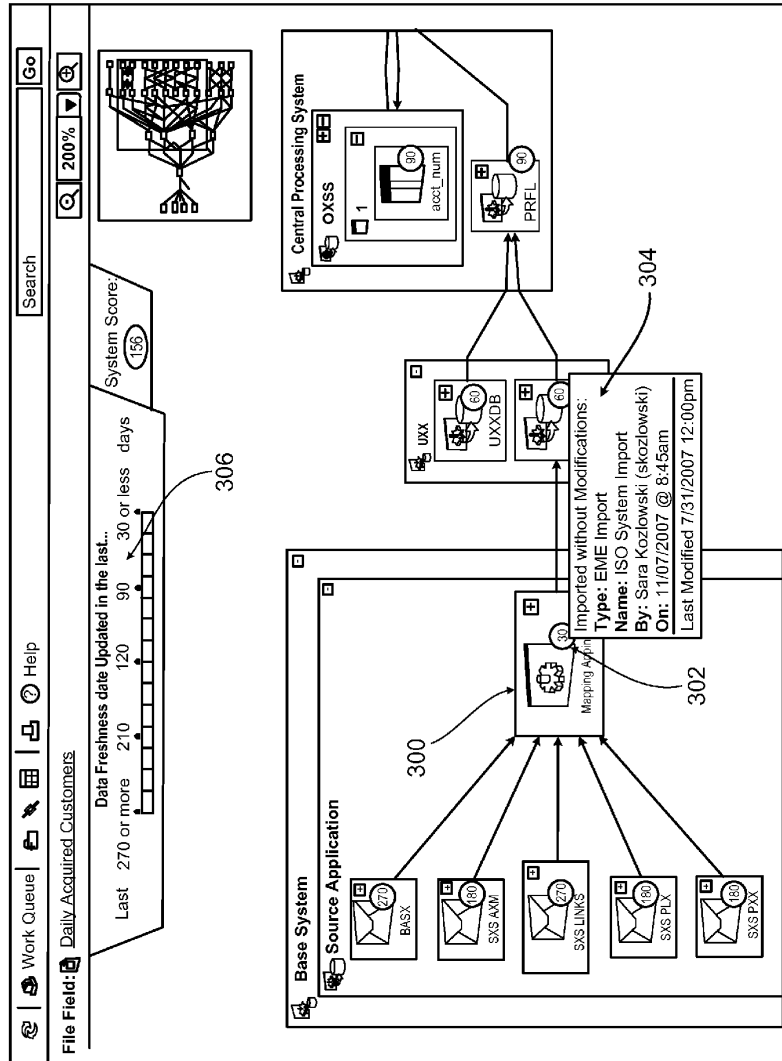




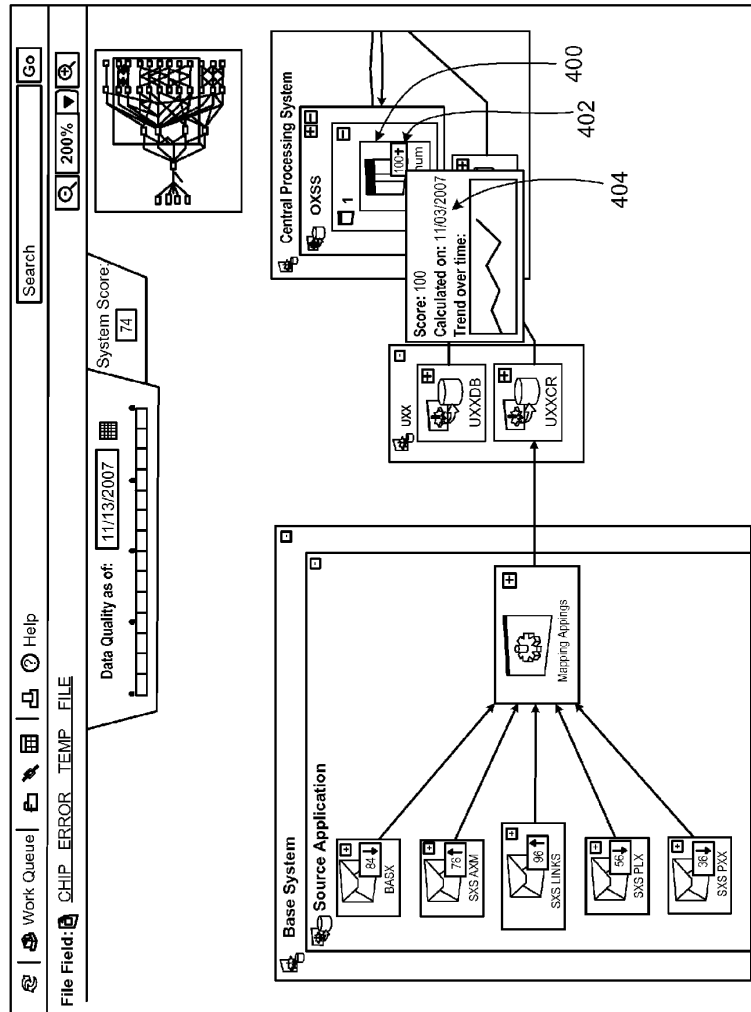
도면2e



도면3



도면4



도면5

