

(19)대한민국특허청(KR)

(12) 공개특허공보(A)

(51) 。 Int. Cl.

G06F 17/40 (2006.01)

G06F 17/30 (2006.01)

(11) 공개번호

10-2006-0049337

(43) 공개일자

2006년05월18일

(21) 출원번호 10-2005-0100701

(22) 출원일자 2005년10월25일

(30) 우선권주장 JP-P-2004-00309439 2004년10월25일 일본(JP)

(71) 출원인 휴렛-팩커드 디벨롭먼트 컴퍼니, 엘 피
미국 텍사스주 77070 휴스턴 스테이트 하이웨이 249 20555(72) 발명자 야마모토 아키오
일본 도쿄도 스기나미쿠 다카이도히가시 3-29-21 휴렛-팩커드재팬 가
부시키가이샤 내
시미즈 히로유키
일본 도쿄도 스기나미쿠 다카이도히가시 3-29-21 휴렛-팩커드재팬 가
부시키가이샤 내
우카이 후미토시
일본 도쿄도 스기나미쿠 다카이도히가시 3-29-21 휴렛-팩커드재팬 가
부시키가이샤 내(74) 대리인 김창세
김원준

심사청구 : 없음

(54) 데이터베이스 시스템, 데이터 관리 방법 및 컴퓨터 판독가능 매체

요약

하나 이상의 관련 노드와 하나 이상의 토픽 노드를 각각 관련시켜서 디렉토리 트리 형태의 데이터베이스를 작성하는 데이터베이스 시스템이 개시되며, 하나 이상의 토픽 노드 각각은 자신에게 속하는 데이터를 구비하고, 관련된 관련 노드와 토픽 노드 사이의 각각의 관련성을 나타내도록 관련 속성이 정의되며, 관련 노드 각각의 관련 노드 엔트리와 관련 속성 각각의 관련 속성 엔트리를 작성하고, 각각의 관련 노드와 각각의 관련 속성 사이의 관련성에 따라서 엔트리를 상관시킴으로써 디렉토리 트리를 작성하는 디렉토리 트리 작성 수단과, 토픽 노드와 각각의 관련 속성 사이의 관련성에 따라서, 토픽 노드 중 하나에 속하는 데이터와 작성된 관련 속성 엔트리를 상관시키는 데이터 상관 수단을 포함한다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 본 발명의 일실시예에 따른 관련성 네트워크형 데이터를 변환할 때의 데이터 구조 또는 모델을 개략적으로 도시한 도면이다.

도 2(a)는 일련의 기억되는 관련성 데이터를 개략적으로 도시한 도면이고, 도 2(b)는 도 1에 도시된 데이터 모델에 따라 상기 일련의 기억되는 일련의 관련성을 갖는 n 개의 데이터를 변환한 데이터구조를 개략적으로 도시한 도면이다.

도 3(a)-3(b)는 본 발명의 일실시예에 따른 관련 노드 사이의 관련성을 표현하는 방법의 일례를 각각 개략적으로 도시한 도면이다.

도 4는 본 발명의 일실시예에 따른 관련성 네트워크형 데이터를 표현하는 일례를 개략적으로 도시한 도면이다.

도 5는 본 발명의 바람직한 실시예에 관한 블록도이다.

도 6은 본 발명의 일실시예에 따른 제 1 데이터베이스 시스템(DB 시스템) 구성을 도시한 도면이다.

도 7은 도 6에 도시된 DB 서버 및 PC의 하드웨어 구성을 도시한 도면이다.

도 8은 재배열된 형태의 도 3(a)에 도시한 데이터 구조를 개략적으로 도시한 도면이다.

도 9는 일반적인 형태의 도 8에 도시된 데이터구조를 개략적으로 도시하는 도면이다.

도 10는 일반적인 형태의 도 3(a) 및 도 4의 데이터의 관계를 개략적으로 도시한 도면이다.

도 11은 도 9에 도시된 데이터 구조를 기억하기 위해서 이용되는 관련성 역할(AR(Association Role)) 테이블을 도시한 도면이다.

도 12는 도 9에 도시된 구조의 데이터를 기억하기 위해서 이용되는 T 노드용 ID(Identifier)) 테이블을 도시한 도면이다.

도 13은 도 9에 도시된 구조의 데이터를 기억하기 위해서 이용되는 A 노드용 ID 테이블을 도시한 도면이다.

도 14는 본 발명의 일실시예에, 따른 예를 들어, 도 6 및 도 7에 도시된 DB 장치에 있어서의 데이터 검색방법을 개략적으로 도시한 도면이다.

도 15는 도 6 및 도 7에 도시된 DB 서버에서 수행되는 검색 처리를 나타내는 흐름도이다.

도 16은 도 15에 도시된 검색필터에 근거하는 관련 노드의 선택처리를 나타내는 흐름도이다.

도 17은 도 15 및 도 16에 도시된 노드 ID 및 노드 명칭을 얻는 처리를 나타내는 흐름도이다.

도 18은 본 발명의 일실시예에 따른, 예를 들어, 도 6 및 도 7에 도시된 DB 서버 상에서 실행되는 DB 프로그램(2)의 구조를 개략적으로 도시한 도면이다.

도 19는 도 6 및 도 7에 나타난 DB 서버(DB 프로그램; 도 18)에 대하여 입력되는 데이터와, 이것에 포함되는 데이터의 검색에 포함되는 검색조건을 예시하는 도이다.

도 20은 본 발명의 일실시예에 따른 AR 엔트리 작성부(도 18) 및 ARDB 관리부에 의해 작성되어, ARDB에 기억되는 AR 테이블을 도시한 도면이다.

도 21은 본 발명의 일실시예에 따른 ID 엔트리 작성부(도 18) 및 IDDB 관리부에 의해 작성되어, IDDB에 기억되는 T 노드용의 ID 테이블을 도시한 도면이다.

- 도 22는 본 발명의 일실시예에 따른 ID 엔트리 작성부(도 18) 및 IDDB 관리부에 의해 작성되는 A 노드에 대한 ID 테이블을 예시하는 도이다.
- 도 23은 본 발명의 일실시예에 따른 제 2 DB 시스템의 구성을 도시한 도면이다.
- 도 24는 본 발명의 일실시예에 따른 제 3 DB 시스템의 구성을 도시한 도면이다.
- 도 25는 도 24에 도시된 DB 시스템의 그래픽적 표현 정보를 개략적으로 도시한 도면이다.
- 도 26은 도 25에 도시된 정보의 디렉토리 정보 트리 표현을 개략적으로 도시한 도면이다.
- 도 27은 본 발명의 일실시예에 따른 디렉토리 구조에 표현된 관련성 데이터를 분할하는 방법을 개략적으로 도시한 도면이다.
- 도 28은 도 27에 도시된 DB 서버(A)와 DB 서버(N)의 엔트리(dn_N) 사이의 관계를 참조하는 단계를 개략적으로 도시한 도면이다.
- 도 29는 본 발명의 일실시예에 따른 도 27에 도시된 DB 서버(A 내지 N)에 분할되고 기억되는 관련성 데이터를 관리하는 데 이용되는 디렉토리 트리 테이블의 단면을 도시한 도면이다.
- 도 30은 본 발명의 일실시예에 따른 DB 서버(A 내지 N)에 기억되는 디렉토리 트리(또는 서브 트리)의 상위 엔트리(dn_A 내지 dn_N) 및 서브 트리에 기억되는 관련성 데이터를 분류하는 데 이용되는 데이터 테이블의 도면이다.
- 도 31은 본 발명의 일실시예에 따른 도 24에 도시된 DB 관리 서버 상에서 실행되는 제 1 DB 관리 프로그램의 구조를 개략적으로 도시한 도면이다.
- 도 32는 본 발명의 일실시예에 따른 데이터 입력용으로 컴퓨터(PC)의 입력/출력 장치(도 6 및 7) 상의 도 31에 도시된 DB 관리 프로그램을 이용하여 디스플레이되는 GUI 스크린의 도면이다.
- 도 33은 본 발명의 일실시예에 따른 도 31에 도시된 DB 관리 프로그램의 등록 처리를 도시한 흐름도이다.
- 도 34는 본 발명의 일실시예에 따른 도 31에 도시된 DB 관리 프로그램을 이용하여 데이터 테이블(도 30)을 수정하는 처리를 도시한 흐름도이다.
- 도 35는 본 발명의 일실시예에 따른 도 31에 도시된 DB 관리 프로그램의 전체적 실행을 개략적으로 도시한 시퀀스 도면이다.
- 도 36은 본 발명의 일실시예에 따른 도 31에 도시된 DB 관리 프로그램의 전체적 실행을 개략적으로 도시한 시퀀스 도면이다.
- 도 37은 본 발명의 일실시예에 따른 도 24에 도시된 DB 서버 각각 상에서 실행되는 제 2 DB 프로그램을 개략적으로 도시한 도면이다.
- 도 38은 본 발명의 일실시예에 따른 도 24에 도시된 검색 장치 상에서 실행되는 검색 프로그램을 개략적으로 도시한 도면이다.
- 도 39는 도 38에 도시된 검색 프로그램을 이용하는 검색 처리를 도시한 흐름도이다.
- 도 40은 본 발명의 일실시예에 따른 도 31에 도시된 DB 관리 프로그램의 전체적 처리 및 도 38에 도시된 검색 프로그램을 도시한 시퀀스 도면이다.
- 도 41은 본 발명의 일실시예에 따른 도 24에 도시된 Db 시스템의 DB 서버에 등록된 평면 디렉토리 구조의 관련성 데이터를 개략적으로 도시한 제 1 도면이다.

도 42는 도 41에 도시된 디렉토리 구조의 관련성 데이터를 개략적으로 도시한 제 2 도면이다.

도 43은 도 41에 도시된 디렉토리 구조의 관련성 데이터를 개략적으로 도시한 제 3 도면이다.

도 44는 본 발명의 일실시예에 따른 도 24에 도시된 DB 시스템의 Db 서버에 등록되는 디렉토리 구조의 관련성 데이터를 개략적으로 도시한 제 4 도면으로, 단지 특정 지역의 세일즈 정보만을 도시하고 있다.

도 45는 본 발명의 일실시예에 따른 도 24에 도시된 DB 시스템의 Db 서버에 등록된 디렉토리 구조의 관련성 데이터를 개략적으로 도시한 도면으로, 복수의 지역의 세일즈 정보를 도시하고 있다.

도 46은 도 44 및 도 45에 도시된 세일즈 정보(관련성 데이터)의 상위 엔트리를 포함하는 디렉토리 트리 테이블의 도면이다.

도 47은 본 발명의 일실시예에 따른, 도 44 및 45에 도시된 세일즈 정보(관련성 데이터)가 서브 트리로 분할될 때 자신의 엔트리, 분류 종류 및 분류 값을 포함하는 데이터 테이블의 도면이다.

도 48은 본 발명의 일실시예에 따른 도 44 및 45에 도시된 세일즈 정보를 검색하는 데 이용되는 GUI 스크린의 도면이다.

도 49는 본 발명의 일실시예에 따른, 도 46 및 47에 도시된 디렉토리 트리 및 데이터 테이블로부터 검색되는 DB 서버를 검출하기 위해 도 24에 도시된 검색 장치에 의해 실행되는 처리를 개략적으로 도시한 도면이다.

도 50은 도 48에 도시된 검색 조건으로부터 얻어지며 도 44에 도시된 칸토 지방만의 세일즈 정보를 검색하는 데 이용되는 LDAP 동작을 발생시키는 검색 요청 메시지의 도면이다.

도 51(a) 내지 51(b)는 본 발명의 일실시예에 따른, 새로운 서브 트리의 추가 전후 각각에, 도 24에 도시된 DB 시스템의 DB 서버의 새롭게 추가되는 서브 트리를 개략적으로 각각 도시한 도면이다.

도 52는 도 51(a) 및 51(b)에 도시된 새로운 서브 트리의 추가에 이용되는 검색 조건 관리 테이블의 도면이다.

도 53(a) 내지 53(b)는 본 발명의 일실시예에 따른, 교환 전후 각각에, 도 24에 도시된 DB 시스템의 DB 서버 사이의 서브 트리 교환/이동을 개략적으로 각각 도시한 도면이다.

도 54(a) 내지 54(b)는 본 발명의 일실시예에 따른, 업데이트 전후 각각에, 도 54(a) 및 54(b)에 도시된 서브 트리의 교환에서의 DB 서버 분할용의 디렉토리 트리 테이블(도 46)의 업데이트를 도시한 도면이다.

도 55는 본 발명의 일실시예에 따른 제 3 DB 프로그램을 개략적으로 도시한 도면으로, 이는 도 24에 도시된 새로운 서브 트리의 추가 및 DB 시스템에서 수행되는 그 이동/교환 시에 DB 서버 상에서 실행된다.

도 56은 본 발명의 일실시예에 따른 제 2 DB 관리 프로그램을 개략적으로 도시한 도면으로, 이는 도 24에 도시된 새로운 서브 트리의 추가 및 DB 시스템에서 수행되는 그 이동/교환 시에 DB 관리 서버 상에서 실행된다.

도 57은 도 56에 도시된 DB 관리 프로그램의 처리를 도시한 흐름도이다.

도 58은 본 발명의 일실시예에 따른, 복수의 DB 서버를 지나는 동일한 레벨에 관한 엔트리를 개략적으로 도시한 도면이다.

도 59는 본 발명의 일실시예에 따른 제 2 검색 프로그램을 개략적으로 도시한 도면으로, 이는 도 24에 도시된 새로운 서브 트리의 추가 및 DB 시스템에서 수행되는 그 이동/교환 시에 검색 장치 상에서 실행된다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 출원은 2004년 10월 25일자 일본 출원 제 2004-309439 호에 기초하여 그 우선권을 주장하며, 본 명세서에서 그 전체를 참조한다.

본 발명의 실시예는 데이터 관리 및/또는 변환용 프로그램을 저장하기 위한 데이터 구조, 데이터베이스 시스템, 방법 및 컴퓨터 판독 가능한 매체에 관한 것이다.

관련되는 데이터를 기억하여, 기억된 이러한 데이터를 검색하기 위해서, 관계성 데이터베이스(relational data base)가 이용되고 있다.

이들의 예로는, "The Associative Model of Data White Paper"(Lazy Software, 2000년 9월), 일본 공개공보 2001-209647 및 WO 00/29980이 있으며, 본 명세서에서 이들 전체를 참조한다.

그러나, 전술한 참조 문헌에 개시된 방법 일부에 있어서, 이미 완성된 데이터베이스의 구조(schema)를 변경하는 것은 용이하지 않다.

전술한 문헌에 개시되어 있는 다른 방법에서는, 데이터의 기술(descriptive) 내용이 복잡하게 되고, 또한, 데이터의 표기·저장방법이 한결 같지 않으므로, 필요한 경우에 본래의 데이터를 정확하게 재발생시킬 수 없을 수 있다.

또한, 전술한 문헌에 개시된 방법에서는, 복수의 서버 사이에 데이터베이스가 분배되는 경우에 이 서버 사이에 부하를 분배하는 것이 쉽지 않다.

발명이 이루고자 하는 기술적 과제

본 발명의 일 실시예에 따르면, 하나 이상의 토픽 노드와의 하나 이상의 관련 노드 각각을 관련시킴으로써 디렉토리 트리 형태의 데이터베이스를 발생시키는 데이터베이스 시스템이 제공되는데, 하나 이상의 토픽 노드 각각은 자신에 속하는 데이터를 가지며, 관련 속성은 관련되는 관련 노드와 토픽 노드 사이의 관련 각각을 표현하도록 정의된다. 이 데이터베이스 시스템은, 관련 노드 각각에 대한 관련 노드 엔트리 및 관련 속성 각각에 대한 관련 속성 노드를 발생시키며 각 관련 노드와 각 관련 속성 사이의 관련에 따른 엔트리를 상호 연관시킴으로써 디렉토리 트리를 발생시키는 디렉토리 트리 발생 구성요소와, 토픽 노드와 각 관련 속성 사이의 관련에 따른 발생되는 관련 속성 엔트리와 토픽 노드 중 하나에 속하는 데이터를 상호 연관시키는 데이터 상호 연관 구성요소를 포함한다.

본 발명의 일 실시예에 따르면, 하나 이상의 관련 노드 각각을 하나 이상의 토픽 노드와 관련시킴으로써 디렉토리 트리 형태의 데이터베이스를 발생시키는 데이터베이스 관리 방법에 제공되는데, 하나 이상의 토픽 노드 각각은 자신에 속하는 데이터를 가지며, 관련 속성은 관련되는 관련 노드와 토픽 노드 사이의 관련 각각을 표현하도록 정의된다. 이 방법은, 관련 노드 각각에 대한 관련 노드 엔트리 및 관련 속성 각각에 대한 관련 속성 엔트리를 발생시키는 단계와, 각 관련 노드와 각 관련 속성 사이의 관련에 따라 엔트리를 상호 연관시킴으로써 디렉토리 트리를 발생시키는 단계와, 토픽 노드와 각 관련 속성 사이의 관련에 따라 발생된 관련 속성을 토픽 노드 중 하나에 속하는 데이터와 상호 연관시키는 단계를 포함한다.

본 발명의 일 실시예에 따르면, 프로그램이 저장된 컴퓨터 판독 가능한 매체가 제공된다. 이 프로그램은 하나 이상의 관련 노드 각각을 하나 이상의 토픽 노드와 관련시킴으로써 디렉토리 트리 형태의 데이터베이스를 발생시키는 컴퓨터를 포함하는 데이터베이스 시스템에서 사용되는데, 하나 이상의 토픽 노드 각각은 자신에 속하는 데이터를 가지며, 관련 속성은 관련되는 관련 속성 노드와 토픽 노드 사이의 관련 각각을 표현하도록 정의된다. 이 프로그램이 데이터베이스 시스템에서 실행되면, 컴퓨터로 하여금, 관련 노드 각각에 대한 관련 노드 엔트리 및 관련 속성 각각에 대한 관련 속성을 발생시키는 단계와, 각 관련 노드와 각 관련 속성 사이의 관련에 따라 엔트리를 상호 연관시킴으로써 디렉토리 트리를 발생시키는 단계와, 토픽 노드와 각 관련 속성 사이의 관련에 따라 발생된 관련 속성을 토픽 노드 중 하나에 속하는 데이터와 상호 연관시키는 단계를 수행하도록 한다.

본 발명의 일 실시예에 따르면, 데이터 구조가 저장된 컴퓨터 판독 가능한 메모리가 제공된다. 이 데이터 구조는 복수의 객체에 속하는 객체 데이터가 저장되는 토픽 노드 필드와, 이 객체들 사이의 관련을 기술하는 관련 데이터가 저장되는 관련 노드 필드와, 관련되는 관련에 대한 객체에 의해 수행되는 역할의 속성을 나타내는 역할 데이터가 저장되는 관련 속성 필드를 포함한다.

본 발명의 일실시예에 따르면, 복수의 링크에 의해 접속되는 복수의 노드를 갖는 제 1 데이터 구조를 복수의 관련 토픽 및 관련 노드를 갖는 제 2 데이터 구조로 변환하는 방법이 제공된다. 이 방법은, 제 1 데이터 구조의 노드 각각을 제 2 데이터 구조의 토픽 노드 중 하나로 변환하는 단계와, 제 1 데이터 구조의 링크 각각을 제 2 데이터 구조의 관련 노드 중 하나로 변환하는 단계와, 이 제 2 데이터 구조에서 관련 노드 각각을, 제 1 데이터 구조에서 관련 노드에 대응하는 링크에 의해 접속되는 노드에 대응하는 2개의 토픽 노드와 관련시키는 단계와, 제 2 데이터 구조에서 관련 속성을 관련되는 관련과 토픽 노드 사이의 각 관련에 할당하여 제 1 데이터 구조의 관련 링크에 대한 노드에 의해 수행되는 역할의 속성을 나타내는 단계를 포함한다.

본 발명의 실시예의 목적, 특징 및 장점은 첨부된 도면을 참조하는 다음의 상세한 설명으로부터 보다 명백해 질 것이다.

발명의 구성 및 작용

본 발명의 실시예는 도시를 위한 것이지 제한하기 위한 것이 아니며, 첨부된 도면 전체에 걸쳐 동일한 참조 번호를 갖는 구성요소는 동일한 구성요소를 가리킨다.

본 발명의 실시예를 자세히 설명하기 전에, 본 발명은 후속하는 상세한 설명에 설명되어 있는 또는 도면에 예시되어 있는 본 발명의 애플리케이션에 제한되지 않는다는 것을 이해해야 한다. 본 발명은 다른 실시예에도 적용될 수 있고 실행될 수 있으며 또는 다양한 방식으로 수행될 수 있다. 또한, 본 명세서에서 사용되는 어구 또는 용어는 단지 설명을 위한 것일 뿐 이들에 제한되어서는 안된다. 방법 또는 프로세스의 단계를 식별하는 문자의 사용은 단지 식별을 위한 것이며 이 단계들이 특정 순서로 실행되어야 한다는 것을 나타내는 것은 아니다.

여러 가지 요소를 포함하는 데이터를 구문분석(parse)하고, 구문분석된 데이터는 저장된다.

필요한 경우 원시 데이터가 이 저장된 데이터로부터 정확하게 재생되는 것이 바람직하다.

일반적으로, 집합($A \times B$)의 카테시안 곱의 부분 집합($R \subseteq A \times B$)에서, 순서 쌍 $(a, b) \in R$ 에 대한 aRb 의 표기는 "a는 b와 관계(R)를 갖는다"라는 것을 의미한다. 저장되고 이후에 재생될 필요가 있는 원시 데이터의 간단한 예로서, "작가 셰익스피어는 희곡 햄릿을 썼다"를 예로 든다.

원시 데이터는 2항 관계를 갖는다. 즉, 원시 데이터 "작가 셰익스피어는 희곡 햄릿을 썼다"는 " aRb " 형식으로 표현될 수 있는데, 여기서 a는 "작가 셰익스피어"이고, R은 "작자-작품"이며, b는 "희곡 햄릿"이다.

따라서, 원시 데이터는 데이터베이스에 "a", "R" 및 "b"로 저장되고, 필요시 재생될 수 있다.

그러나, 데이터 구성요소의 수가 증가하는 경우, 즉 원시 데이터가 2항 관계를 가지고 있는 것이 아니라 n-항 관계를 가지고 있는 경우, 이들 관련성을 갖는 일련의 데이터는 하이퍼그래프(hypergraph)로 표현되고, 데이터의 처리는 간단하지 않다.

따라서, 임의의 n-항 관계를 다수의 2항 관계로 분해하여, 데이터를 2항 관계의 조합으로 구문분석하고, 구문분석된 데이터를 데이터베이스에 저장하는 방법이 이용된다.

제 1 예로서, "작가 셰익스피어는 희곡 햄릿을 영국에서 1600년 대에 썼다"를 예로 들어 설명할 것이다.

이 예는 4항 관계를 도시하는데, 이 4항 관계는 표 1에 도시되어 있는 바와 같이 6개의 2항 관계의 조합으로 분해되는데, 그 이유는 $4C2=6$ 이기 때문이다.

[표 1]

a	R	b
작가 셰익스피어	작자-작품	희곡 햄릿
작가 셰익스피어	작자-창작연대	1600년경
작가 셰익스피어	작자-창작국	영국
희곡 햄릿	작품-창작연대	1600년경
희곡 햄릿	작품-창작국	영국
1600년경	창작연대-창작국	영국

달리 말해, 임의의 주어진 n에 있어서, nC2 2항 관계의 조합이 n-항 관계를 표현하는데 필요하다.

다음으로, 제 2 예로서, "작가 셰익스피어는 희곡 Twelfth Night을 1600년대에 썼다"를 예로 들것이다. 이 예는 3항 관계를 도시하는데, 이 관계는 3개의 2항 관계의 조합으로 분해된다. 즉, $n=3$, $3C2=3$ 이다. 결과는 표 2에 도시되어 있다.

[표 2]

a	R	b
작가 셰익스피어	작자-작품	희곡 십이야
작가 셰익스피어	작자-창작연대	1600년경
희곡 십이야	작품-창작연대	1600년경

이 경우, 제 1 예의 정보 및 제 2 예의 정보를 동일한 데이터베이스에 저장하게 되면 "작가 셰익스피어" - "작자-창작 연대" - "1600년대"으로 구성된 동일한 2항 관계가 저장된다는 문제가 발생한다.

또 다른 문제는 제 1 및 제 2 예의 데이터를 표현하는데 사용된 2항 관계의 조합 중 어느 것이 사용되어 원시 정보를 재생해야 하는지에 관해 판단을 할 수 없다는 것이다. 달리 말해, 데이터를, 예를 들어 제 1 및 제 2 예는 유일하게 구문분석 및 저장되지 않는다.

이들 문제점들은 각각의 2항 관계에 ID를 부가함으로써 해결될 수 있다. 그러나, 데이터 구조 및 프로세싱이 보다 복잡해진다는 단점이 있다.

2항 또는 3항 또는 임의의 n-항 데이터를 관계 데이터베이스에 저장하는 것이 알려져 있다.

관계 데이터베이스는 데이터 항목이 저장되는 하나 이상의 필드를 포함한다. 관계 데이터베이스의 간단한 예는 필드 또는 데이터 요소가 열로 배열되고 데이터 항목은 행으로 배열되는 테이블이다. 본 명세서에서 사용되는 "필드" 또는 "데이터 요소"는 데이터의 논리적 정의로서 이해되는 반면, "데이터 항목"은 필드에 저장되는 데이터의 실제 단위로서 이해될 것이다.

이제 제 1 예 "작가 셰익스피어는 희곡 햄릿을 영구에서 1600대에 썼다"로 되돌아간다.

(1) "누가", (2) "무엇을", (3) "언제", (4) "어디서", (5) "왜" 및 (6) "어떻게"가 데이터 요소 또는 필드로 지정될 수 있다.

이들 필드에 대응하여, (1) "작가 셰익스피어", (2) "희곡 햄릿", (3) "1600년대", (4) "영국", (5) "널(null)" 및 (6) "썼다"의 데이터 항목이 저장될 수 있다.

그러나, 관계 데이터베이스를 사용하여 데이터를 저장하는 방법에는 다음과 같은 고유한 문제점이 있다.

(1) 새로운 데이터가 발생하는 경우 이후에 데이터 요소를 추가하는 것이 쉽지 않다. 이러한 부가는 정형화된 구성요소로 구성된 데이터의 경우에는 문제가 되지 않는다. 그러나, 다양한 구성요소를 포함하는 데이터를 입력하는 경우, 새로운 데이터 구성요소를 갖는 새로운 데이터가 도달할 때마다 데이터 요소를 추가하는 데이터베이스의 스키마(schema)를 변경할 필요가 있다. 일반적으로 스키마 변경은 데이터가 온 라인으로 입력되는 경우에 특히 어렵다.

(2) 새로운 필드 또는 데이터 요소를 이후에 추가하는 것은 쉽지 않기 때문에, 초기 데이터베이스 구축 동안, 스키마는 최대 수의 필드 또는 데이터 요소로 구성될 수 있다. 그러나, 다수의 필드는 임의의 데이터 항목을 포함할 수 없다. 이것은 불가피하게 메모리 사용의 효율을 감소시킨다.

이러한 문제점을 해결하기 위해, "Lazy Software" 사(잉글랜드)가 개발한 "데이터 관련 모델(Associative Model of Data)" 기초한 데이터 저장 방법은 종래의 관계 데이터베이스 모델의 또 다른 모델로서 제안되었다.

이 관련 데이터 모델에 따르면, 정보는 객체 간의 관련성의 관점에서 처리되고, 이 관련성은 "Source-Verb-Target" 구문으로 표현될 수 있다.

이 모델은 관계 데이터베이스에 기초한 저장 방법에 고유한 문제점 중 일부를 해결한다.

그러나, 관련 데이터 모델은 다음과 같은 문제점이 있다. 복잡한 데이터 관계를 처리하는 경우에, 데이터 간의 관계를 표현하는 모델의 방법이 복잡하고 직관적이지 않다. 위에서 설명한 바와 같이, n-항 관계가 2항 트리로 표현되는 경우, 데이터는 유일하게 구문분석 및 저장되지 않는다. 또한, 오퍼레이터는 데이터가 데이터베이스에 저장되는 경우 데이터 구조를 선택에 따라 변경할 수 있다. 따라서, 원시 정보는 정확하게 재생되지 않을 수 있다.

위에서 설명한 바와 같이, 관계 데이터베이스는 통상적으로 서로에 관련된 다수의 데이터를 저장하는 방법으로서 알려져 있다.

이 방법에 따르면, 필드 또는 데이터 요소는 사전설정되고, 데이터 항목은 이러한 필드에 저장된다.

이 방법은, 데이터 관계가 쉽게 이해되고 및/또는 표현된다는 점에서 유리하다. 그러나, 새로운 필드 또는 데이터 요소를 이미 구축된 데이터베이스에 추가하는 것, 즉 이미 구축된 데이터베이스의 구조(스키마)의 변경은 쉽지 않다.

새로운 필드 또는 데이터 항목이 이후에 추가되는 경우, NULL 엔트리는 저장된 데이터에 대하여 생성되어, 메모리 사용 효율의 감소에 대한 문제를 야기할 것이다.

한편, Lazy Software 사가 제안한 "데이터의 관련 모델"의 경우에는, 새로운 데이터 항목을 추가함에 있어서의 어려움 및 메모리 사용 효율의 감소에 대한 문제점은 해결된다. 그러나, 데이터 기술 콘텐츠는 복잡해지고, 데이터 구조는 직관적이지 않으며, 따라서 구문분석된/저장된 데이터는 한결 같지 않다는 또 다른 문제점이 발생한다.

도 1은 토픽 노드(111,121) 및 이 노드(111,121) 간의 관련성을 나타내는 링크(131)로 구성된 관련 네트워크 데이터(101), 및 본 발명의 실시예에 따른 데이터 모델(102)을 도시한다. 관련성(링크)(131)은 데이터 모델(102)에서 관련성 노드(132)(이하에서 "A 노드"로 지칭됨)로 재정의된다. 토픽 노드(111,121)는 데이터 모델(101)의 토픽 노드(112,122)(이하에서 "T 노드"로 지칭된다)에 대응한다. 데이터 모델(102)은 제각각 T 노드(112,122)를 A 노드(132)에 접속시키고 관련 네트워크 데이터(101)의 관련성 또는 링크(131)에 대하여, 토픽 노드(111,121)가 담당하는 역할(이하에서 "관련성 역할"로 지칭됨)을 반영하는 속성을 갖는 링크(142,152)를 더 포함한다.

따라서, 관련성 네트워크 데이터(101)는 데이터 모델(102)로 변환된다. 본 발명의 실시예에서, 데이터 모델(102)은 새로이 구성된다. 간단한 예에서, 데이터 모델(102)은 세 개의 필드, 즉 하나의 "A 노드" 필드, 하나의 "T 노드" 필드 및 하나의 "관련성 역할" 필드를 포함하는 관련성 역할 테이블의 형태로 표현되고, 관련성 네트워크 데이터(101)로부터 추출된 정보는 데이터 항목으로서 표 3에 도시된 관련성 역할 테이블(이하에서 "AR 테이블"로 지칭됨)의 행(레코드)에 기입된다. 일 실시예에서, 관련 데이터베이스의 일부분으로서 정의되는 AR 테이블은 관계 데이터베이스 관리 시스템을 사용하여 관리된다.

[표 3]

A 노드	T 노드	관련성 역할
A1	T1	관련성 역할 1
A1	T2	관련성 역할 2

따라서, 소정의 데이터(토픽 노드)와 관련된 새로운 속성 정보는 또 다른 관련성 데이터로서 정의되고, 대응 A 및 T 노드와 노드(즉, 본 발명의 실시예에 따른 데이터 모델의 기본 구성요소) 간의 링크의 조합을 사용함으로써 AR 테이블의 행과 관련된 데이터로서 표현되어, 새로운 속성 정보는 기존의 테이블 구조(데이터베이스 스키마)를 변경하지 않고도 추가될 수 있다.

또한, 노드를 유일하게 식별하는 ID가 A 및 T 노드에 할당된다. 이 ID에 대해, ID 테이블(이하에서는 "ID 테이블"로 지칭됨)이 정의되고 노드 속성 유형을 나타내는 노드 유형 및 속성 값, 즉 노드의 특정 콘텐츠를 나타내는 노드 명칭을 포함한다(표 5).

ID 테이블은 AR 테이블의 경우에서와 같이 관계 데이터베이스 관리 시스템에 의해 관리된다.

본 발명의 실시예에 있어서, T 노드는 소정의 A 노드로 표현되는 관련성의 특정 의미를 기술하는 데이터로서 새롭게 추가되고, 이들 두 개의 노드, 즉 기존의 A 노드 및 새로운 T 노드는 "구상화(reification)"로서 사전정의된 관련성 역할에 의해서로 관련지어진다.

또 다른 A 노드에 대해 유사하게 추가된 새로운 T 노드와 또 다른 새로운 T 노드 간의 관련성을 더 정의/기술함으로써, 두 개의 원시 A 노드에 의해 표현되는 관련성 간의 관계를 표현할 수 있다.

새로운 T 노드 및 특히 A 노드의 의미를 설명하기 위해 도입된 관련성 역할 "구상화"는 AR 테이블에 의해 저장/관리될 수 있다.

ID 테이블 및 AR 테이블을 사용하여 노드를 관리함으로써, A와 T 노드 간의 관련성 뿐만 아니라 A 노드 간의 관련성도 표현할 수 있는 데이터 표현 방법이 실현된다.

종래의 방법에 있어서, nC2 2항 관계는 n-항 데이터의 일부를 표현하기 위해 필요하다. 그러나, 본 발명의 실시예에 따르면, 동일한 n-항 데이터를 표현하기 위해 n 개의 관계만이 요구될 것이다.

달리 말해, 하나의 공통 관련성(도 2(a))을 갖는 n 개의 데이터 구성요소로 구성된 데이터가 본 발명의 실시예에 따라 저장되는 경우, 데이터 집합에 공통인 새로운 노드(A 노드)가 추가되고, 관련성 역할은 각 데이터 구성요소마다, 즉 T 노드마다 정의된다(도 2(b)).

결과적인 AR 테이블은 표 4에 주어진다.

[표 4]

A 노드	T 노드	관련성 역할
A1	T1	관련성 역할 1
A1	T2	관련성 역할 2
A1	T3	관련성 역할 3
A1	T4	관련성 역할 4
:	:	:
A1	T(n-1)	관련성 역할 (n-1)
A1	Tn	관련성 역할 n

도 2(b) 및 표 4에 사용된 "A1"은 제각기의 A 노드에 할당된 ID이며, 데이터 구성요소(다른 ID("T1" 내지 "Tn")가 할당된 T 노드)가 소정의 공통 관련성을 갖고 있다는 것을 나타낸다.

다수의 상이한 ID를 갖는 다수의 A 노드는 데이터 구성요소가 다수의 상이한 공통 관련성을 갖는 경우 추가된다.

또한, ID가 할당된 A 및 T 노드에 대해, 데이터 속성으로서 "노드 유형" 필드 및 "노드 명칭" 필드를 갖는 ID 테이블은 표 5에 도시되어 있는 바와 같이 생성된다.

[표 5]

노드 ID	노드 타입	노드명
A1	A 노드 타입 1	A 노드명 1
T1	T 노드 타입 1	T 노드명 1
T2	T 노드 타입 2	T 노드명 2
:	:	:
Tn	T 노드 타입 n	T 노드명 n

도 3(a)-3(b)은 T 노드(T1 및 T2)가 A 노드(A1)에 의해 서로 관련되고 T 노드(T1 및 T3)는 A 노드(A2)에 의해 서로 관련되며(도 3(a)), T 노드(ID(T11))가 새롭게 추가되어 A 노드(A1)에 의해 표시되는 관련성의 특정 의미를 기술하며, "구상화"로서 사전정의된 관련성 역할에 기초하여 A 노드(A1)와 관련되는 예를 도시한다.

유사하게, A 노드(A2)는 관련성 역할 "구상화"에 기초하여 새로운 T 노드(T12)와 관련되고, 두 개의 T 노드(T11 및 T12) 간의 관련성은 A 노드(A11)를 사용하여 정의된다(도 3(b)).

따라서, 두 개의 원시 A 노드(A1 및 A2) 간의 관계는 표 6에 도시되어 있는 것과 유사한 AR 테이블을 사용하여 표현될 수 있다.

[표 6]

A 노드	T 노드	관련성 역할
A1	T11	구상화
A2	T12	구상화
A11	T11	관련성 역할 1
A11	T12	관련성 역할 2

데이터 저장 방법/데이터 구조

이하에서, 본 발명의 실시예에 따른 데이터 저장 방법 및 데이터 구조가 설명될 것이다.

구체적인 예로서, 제 1 데이터로서 "작가 셰익스피어는 희곡 햄릿을 영국에서 1600년대에 썼다"를 예로 들것이다.

제 1 데이터는 (1) "작가 셰익스피어", (2) "희곡 햄릿", (3) "1600년대", (4) "영국"의 데이터 구성요소를 갖는 4항 관계를 나타낸다. 제 1 데이터가 종래의 방법을 통해 2항 관계로 구문분석되는 경우, $4C2=6$ 이기 때문에 6개의 2항 관계의 조합이 표 7에 도시되어 있는 바와 같이 요구될 것이다.

[표 7]

토픽 노드 1	링크(관련성)	토픽 노드 2
작가 셰익스피어	작자-작품	희곡 햄릿
작가 셰익스피어	작자-창작연대	1600년경
작가 셰익스피어	작자-창작국	영국
희곡 햄릿	작품-창작연대	1600년경
희곡 햄릿	작품-창작국	영국
1600년경	창작연대-창작국	영국

구문분석된 데이터는 본 발명의 실시예에 따라 변환된다. 표 1의 제 1 레코드(행)는 아래에 설명되어 있는 바와 같이 변환된다.

토픽 노드(1)의 데이터 구성요소 "작가 셰익스피어"는 이 정보에서 "작가"를 나타낸다. 따라서, 이 데이터 구성요소는 "작가"와 "셰익스피어"로 구문분석되고, "셰익스피어"는 T 노드로 설정되고, "작자"는 관련성 역할로 설정된다. 이후에 설명되는 바와 같이, "작가"는 T 노드로 설정된다.

관련성을 나타내는 링크 "작자-작품"에 대해, A 노드는 일련의 정보가 동일한 그룹에 속해 있다는 것을 나타내도록 "햄릿 저작자"에 추가된다.

"햄릿 저작"은 일련의 정보가 동일한 그룹에 속해 있다는 것을 나타내는데 사용되기 때문에, 이 정보가 다른 그룹의 정보와 구별될 수 있는 한 다른 표현도 허용된다.

따라서, 표 7의 제 1 레코드(행)는 도 8에 도시되어 있는 바와 같이 변환된다.

[표 8]

A 노드	T 노드	관련성 역할
햄릿 저작에 대하여	셰익스피어	작자

유사하게, 토픽 노드(2)의 "희곡 햄릿"은 표 9에 도시되어 있는 바와 같이 변환되는데, 그 이유는 그것은 이 정보에서 "작품"이기 때문이다. 또한, "희곡"은 T 노드로 설정된다.

[표 9]

A 노드	T 노드	관련성 역할
햄릿 저작에 대하여	햄릿	작품

다음으로, 표 7의 제 2 레코드(행)의 유사한 변환이 표 10에 도시되어 있다.

[표 10]

A 노드	T 노드	관련성 역할
햄릿 저작에 대하여	셰익스피어	작자
햄릿 저작에 대하여	1600년경	창작연대

여기서, 데이터 항목 "햄릿 저작자", "셰익스피어", 및 "작자"는 그들이 중복되기 때문에 생략될 수 있다. 이하에서는, 표 7의 모든 레코드(행)는 유사하게 변환되고, 중복 데이터는 생략되며, 결과는 표 11에 도시되어 있다.

[표 11]

A 노드	T 노드	관련성 역할
햄릿 저작에 대하여	셰익스피어	작자
햄릿 저작에 대하여	햄릿	작품
햄릿 저작에 대하여	1600년경	창작연대
햄릿 저작에 대하여	영국	창작국

따라서, 하나의 공통 관련성을 갖는 네 개의 데이터 구성요소로 구성된 데이터 집합이 종래의 방법을 사용하여 2항 관계로 표현되는 경우, $4C2=6$ 이기 때문에 6개의 레코드가 필요하다. 그러나, 본 발명의 실시예에 따르면 4개의 레코드만이 필요하다는 것을 알 수 있다.

일 실시예에서, 네 개의 데이터 구성요소를 갖는 데이터 집합이 구문분석되고 데이터베이스에 저장되는 경우, 이 데이터 집합에 공통인 새로운 노드(A 노드)가 부가되고, 그런 다음 관련성 역할이 각 데이터 구성요소에 대해 정의되며, "A 노드", "T 노드" 및 "관련성 역할" 필드를 갖는 데이터베이스가 정의된다. 따라서, 본 발명에 따른 데이터 구조는 새롭게 구성될 수 있다.

여기서, ID "A1"은 A 노드 "햄릿 저작자"에 할당된다. 공통 ID "A1"을 갖는 데이터는 동일한 그룹에 속한다는 것을 알 수 있다. 또한, ID "T11" 내지 "T14"는 네 개의 T 노드 "셰익스피어", "햄릿", "1600년대" 및 "영국"에 각각 할당된다.

따라서, 제 1 데이터에 대한 AR 표가 표 12에 도시한 바와 같이 작성된다.

[표 12]

A 노드	T 노드	관련성 역할
A1	T11	작자
A1	T12	작품
A1	T13	창작연대
A1	T14	창작국

"햄릿의 저작"을 나타내는 A 노드(ID A1)는 "저작 관련 정보"로서 설정되며, ID T11 내지 T14가 할당된 T 노드의 노드 타입은 "작가", "희곡", "년대", "국가"로서 설정된다. 따라서, 표 13에 나타낸 바와 같은 ID 표가 작성된다.

[표 13]

노드 ID	노드 타입	노드 명칭
A1	저작 관련 정보	(NULL)
T11	작가	셰익스피어
T12	희곡	햄릿
T13	연대	1600년경
T14	나라	영국

제 2 데이터로서, "희곡 햄릿을 원작으로 하는 일본어 번역판이 2003년 2월에 OO 출판사에서 출판되었다"는 진술이 다른 예로서 고려될 것이다.

제 2 데이터는 본 발명의 실시예에 따라서 다음과 같이 분석된다. 여기서, 모든 데이터 성분에 공통인 A 노드는 "햄릿의 일본어 번역판"이고, ID "A2"가 그에 할당된다. 이 결과는 표 14에 도시된다.

[표 14]

A 노드	T 노드	관련성 역할
A2	햄릿	원작
A2	햄릿	번역판
A2	2003년 2월	출판일
A2	OO출판사	출판

ID "T12"는 원작인 햄릿에 할당된다. 따라서, ID "T22"가 번역판 햄릿에 할당되고, ID "T23" 및 "T24"가 출판일 및 출판사에게 할당되면, 제 2 데이터에 대한 AR 표는 표 15와 같이 도시된다.

[표 15]

A 노드	T 노드	관련성 역할
A2	T12	원작
A2	T22	번역판
A2	T23	출판일
A2	T24	출판

또한, ID 표는 표 16에 나타난 바와 같이 작성된다.

[표 16]

노드 ID	노드 타입	노드명
A2	저작 관련 정보	(NULL)
T22	희곡	햄릿
T23	날짜	2003년 2월
T24	출판사	OO출판사

제 1 및 제 2 데이터와, 그 밖의 그러한 데이터는 동일한 데이터베이스에 기억된다. 따라서, 최종적으로는 표 17 및 표 18과 유사한 AR 표 및 ID 표를 얻는다.

[표 17]

A 노드	T 노드	관련성 역할
A1	T11	작자
A1	T12	작품
A1	T13	창작연대
A1	T14	창작국
A2	T12	원작
A2	T22	번역판
A2	T23	출판일
A2	T24	출판
:	:	:

[표 18]

노드 ID	노드 타입	노드명
A1	저작 관련 정보	(NULL)
T11	작가	셰익스피어
T12	희곡	햄릿
T13	연대	1600년경
T14	국가	영국
A2	제작 관련 정보	(NULL)
T22	희곡	햄릿
T23	날짜	2003년 2월
T24	출판사	○○출판사
:	:	:

또한, 도 4에 도시한 바와 같이, 부가적인 T 노드가 제공되어, ID A1 및 A2를 갖는 A 노드에 의해 표시된 관련성의 구체적인 의미를 설명한다. ID "T31" 및 "T32"는 부가/신규의 T 노드에 할당되며, 부가/신규의 T 노드는 관련성 역할 "구상화(reification)"을 통해서 노드 A1 및 A2와 관련된다.

이들 2개의 신규한 T 노드의 노드 타입은 "저작 정보"로서 설정되며, 노드 명칭은 각각 "햄릿의 저작권" 및 "햄릿의 일본어 번역판"으로서 설정된다.

노드 T31과 T32 사이의 관련성을 나타내는 A 노드는 ID A3으로서 새로이 추가되며, 노드 타입은 "원작 번역 정보"로서 설정된다.

관련성에 있어서 노드 T31 및 T32의 역할은 "원작 정보" 및 "번역 정보"이다. 전술한 처리를 통해서, AR 표 및 ID 표가 각각 표 19 및 표 20으로 나타낸 바와 같이 추가된다. 본 발명의 실시예에서는, 표 19 및 표 20이 표 17 및 표 18에 각각 추가된다.

[표 19]

A 노드	T 노드	관련성 역할
A1	T31	구상화
A2	T32	구상화
A3	T31	원작 정보
A3	T32	번역 정보

[표 20]

노드 ID	노드 타입	노드명
T31	저작 정보	햄릿 저작에 대하여
T32	저작 정보	햄릿 일본어 번역에 대하여
A3	원작-번역 정보	(NULL)

제 1 및 제 2 데이터를 동일한 상관 데이터베이스에 직접 기억하기 위해서, 항목(데이터 속성), 예를 들어 번역판으로서의 "햄릿" 또는 "OO 출판사"에 대응하는 항목은 새로운 필드로서 새로이 추가되어야 한다. 따라서 데이터베이스의 표 구조가 변화해야 하기 때문에, 종래 방법에 의해서는 이러한 추가가 용이하지 않다.

그러나, 본 발명의 실시예에 따르면, 새로운 행(즉, 새로운 필드가 아닌 새로운 레코드)을 상기에 도시한 기존의 AR 표에 추가함으로써, 상이한 데이터 속성을 갖는 데이터를 상이한 그룹으로서 기억하여, 제 1 데이터를 제 2 데이터와 구별하는 것이 가능하다.

노드들간의 관련성을 나타내는 A 노드, T 노드 및 링크를 사용하는 제 1 및 제 2 데이터는 도 4에 도시한 바와 같이 표현된다.

본 발명의 실시예는 복잡한 구조를 갖는 데이터를 용이하게 표현하는 방법 및 상관 데이터베이스를 사용하는 기억/관리 방법을 제공한다.

다음, 본 발명의 실시예의 데이터베이스에서 소망하는 데이터를 검색하는 경우가 설명된다.

예를 들어, 사용자는 작가 셰익스피어가 쓴 희곡 햄릿의 일본어 번역판의 출판사를 알기 원한다.

도 5는 본 발명의 실시예에 따라 수행되는 검색 과정을 나타낸 순서도이다. 도 5의 순서도는 이하에서 설명된다.

단계 110: 사용자는 "작자"(또는 "저자")인 "셰익스피어"가 쓴 "희곡"(또는 "작품")과 같은 검색 조건을 입력한다.

단계 120: 검색 조건을 충족하는 하나 이상의 데이터 그룹이 수집된다.

단계 130: 검색된 하나 이상의 데이터 그룹 중에서 "희곡"에 대응하는 데이터가 디스플레이된다.

단계 140: 사용자는 디스플레이된 데이터로부터 소망하는 희곡명, 즉 "햄릿"을 선택한다.

단계 150: 데이터베이스는 새로운 검색 조건으로서 "햄릿" 및 "번역판"으로 다시 검색된다.

단계 160: 새로운 검색 조건을 충족하는 다른 하나 이상의 데이터 그룹이 수집된다.

단계 170: 검색된 하나 이상의 데이터 그룹 중에서 "출판사" 및 "출판일"에 관한 데이터가 디스플레이된다.

단계 180: 사용자는 디스플레이된 데이터 중에서 소망하는 출판사를 선택한다.

이하, 상세히 설명된다.

사용자는 검색 조건으로서 "세익스피어" 및 "저자"를 입력하여, 데이터를 검색한다.

이 특정 예에서는, 데이터베이스에 T 노드의 노드 명칭 "세익스피어" 및 관련성 역할 "저자"가 존재한다. 관련성 역할이 "저자"인 T 노드(예컨대, T11)의 ID가 데이터베이스 내의 AR 표(예컨대, 표 17)를 참조하여 검색된다. 이어서, 노드 명칭이 "세익스피어"인 T 노드(예컨대, T11)에 대응하는 A 노드의 하나 이상의 ID 그룹이 ID 표(예컨대, 표 18)에 기억된 노드 명칭 속성을 참조하여 검색된다.

검색된 하나 이상의 A 노드 ID 그룹을 포함하는 검색 결과 내에서, 다른 검색 조건, 즉 관련성 역할이 "작품"인 T 노드 ID가 AR 표(예컨대, 표 17)에서 선택된다.

선택된 T 노드, 즉 관련성 역할 "작품"에 대응하는 희곡(들)에 관한 데이터 즉, 노드 명칭이 선택된 ID를 기초로 하는 ID 표(예컨대, 표 18)로부터 디스플레이된다.

예를 들어, 희곡 또는 노드 명칭 "햄릿", "말괄량이 길들이기", "베니스의 상인", "한 여름밤의 꿈", "리어왕" 등이 디스플레이된다. 사용자는 소망하는 희곡, 또는 노드 명칭, 즉 "햄릿"을 선택한다.

이어서, 선택된 노드 명칭 "햄릿"의 ID "T12"를 키로서 사용하여, 본 발명의 실시예에 따른 시스템은 ID "T12"를 T 노드 ID로서 포함하는 동시에 AR 표(예컨대, 표 17)로부터 관련성 역할 "번역판"을 갖는 A 노드 ID를 검색한다.

따라서, 검색 조건을 충족하는 A 노드 ID와 관련된 하나 이상의 데이터 그룹이 검색된다.

관련성 역할로서 "출판사" 및 "출판일"을 갖는 T 노드의 노드 명칭은 ID 표(예컨대, 표 18)를 참조하여 검색된 하나 이상의 데이터 그룹으로부터 디스플레이된다.

이제, 사용자는 디스플레이된 데이터로부터 소망하는 출판사, 즉 최근 "2003년 2월"에 희곡 햄릿의 번역판을 출판한 "OO출판사"를 선택할 수 있다.

도 5의 과정은 하나의 검색 예로서 설명되었다. 즉, 검색 조건의 수가 증가하면, 데이터베이스 내에서의 검색 횟수는 도 5에 도시한 바와 같이 2회로 제한되지 않으며, 검색 조건에 따라서 임의의 횟수가 될 수 있다.

설명한 특정 예에서, 단일 속성은 관련 역할로서 정의된다. 그러나, 각각의 관련 역할이 단지 하나의 속성으로 제한되는 것은 아니다.

다시 말해, 관련 역할은 다수의 속성을 가질 수 있다. 상기의 예에 있어서, 더 많은 상세한 속성이 "비극", "희곡", "낭만극", "사극" 등과 같은 장르를 관련성 역할 "희곡"에 추가 및 지정함으로써 정의될 수 있다.

데이터 기억 방법/데이터 구조의 특징(Features of Data Storage Method/Data Structure)

전술한 바와 같이, 본 발명의 실시예의 데이터 기억 방법 및 데이터 구조에 따르면, 널리 사용되는 상관 데이터베이스 형식을 사용함으로써, 3항 이상의 관계, 일반적으로 n항의 상호 관계를 갖는 일련의 데이터 관계를 나타내는 하이퍼그래프 구조를 유지하면서 데이터가 기억 및 관리될 수 있다.

본 발명의 실시예에 따라서 관련성 네트워크 데이터를 상관 데이터베이스의 표(들)에 직접 맵핑하는 방법에 따르면, 3항 이상의 관계, 일반적으로 n항의 간계를 갖는 데이터를 효율적으로 기억/관리할 수 없다는 종래의 문제점을 해결할 수 있다.

또한, 이미 데이터베이스에 기억되어 있는 데이터에 속성 데이터를 추가하는 것과 같은 변경은 상관 데이터베이스 표의 구조의 변화를 필요로 하여, 유연성의 손실을 초래하고 더 많은 노동력을 필요로 한다는 종래의 문제를 해결할 수 있다.

또한, 동일한 데이터베이스 방식의 구성에서 통상 데이터에 할당되는 추가의 ID 및 추가의 관련성 역할을 사용함으로써 일련의 관련성에 특정한 의미를 부여할 수 있다.

제 1 데이터베이스 시스템(First Database System)

이하, 본 발명의 실시예의 제 1 실시예가 설명된다.

도 6은 본 발명의 실시예에 따라 구현된 제 1 데이터베이스 시스템(DB 시스템)(1)의 구성을 도시한다.

도 6을 참조하면, 본 발명의 실시예의 제 1 DB 시스템(1)은 데이터베이스 서버(DB 서버)(12)를 LAN, WAN 또는 인터넷과 같은 네트워크(100)를 통해서 데이터를 입력하고 검색하는 데 사용된 컴퓨터(PC)(102)에 접속시킴으로써 구성된다.

이하의 설명에서, 방법은 도 4 및 도 5와 퍼 7 내지 표 20을 참조하여 데이터 기억 방법 및 데이터 구조를 설명하기 위해 상기에 사용된 방법과는 부분적으로 또는 약간 상이하다. 그러나, 이 설명의 대응하는 용어의 의미는 실질적으로 동일하다.

일치하지 않는 경우, DB 시스템(1)에 대한 하기의 설명에서 사용된 용어의 의미가 우선시된다.

이하에서 참조된 도면 전체에서, 유사한 구성 요소는 유사한 참조 부호로 표기된다.

하드웨어 구성(Hardware Configuration)

도 7은 도 6에 도시한 DB 서버(12) 및 PC(102)의 하드웨어 구성을 도시한다.

도 7을 참조하면, DB 서버(12) 및 PC(102)는 CPU(122), 메모리(124) 및 주변 회로를 포함하는 본체(120)와, 디스플레이 유닛 및 키보드를 포함한 입력/출력 디바이스(126)와, CD 또는 HDD와 같은 기록 디바이스(128)를 포함한다. 또한, DB 서버(12) 및 PC(102)(이하, 통신을 수행하는 구성 요소는 일반적으로 통신 노드라고 함)가 네트워크(100)에 접속된 경우, 네트워크(100)를 통해서 다른 통신 노드와의 통신을 수행하는 통신 디바이스(132)가 추가된다.

다시 말해, DB 서버(12) 및 PC(102)는 다른 통신 노드와의 통신을 수행하는 기능을 갖춘 컴퓨터로서의 구성 요소를 포함한다.

데이터 구조(Data Structure)

DB 서버(12)는 도 4 및 도 5와 퍼 7 내지 표 20을 참조하여 전술한 본 발명의 실시예의 데이터 기억 방법 및 데이터 구조에 따라서 데이터를 기억하고 기억된 데이터를 검색하도록 구성된다.

이제, DB 서버(12)에서의 데이터 구조 및 데이터 검색 메커니즘이 설명된다.

도 8은 도 3(a)의 데이터 관련성을 재배열된 형태로 나타낸다.

도 8을 참조하면, DB 서버(12)에서, 토픽 노드(이하, T 노드)는 하나 이상의 관련 노드(이하, A 노드)와 관련되며, 관련된 T 노드와 A 노드 사이에는 관련 속성 R이 정의된다.

관련 속성 R은 T 노드와 A 노드 사이의 관련성을 정의하는 임의의 속성일 수도 있다. 그러나, 이하에 제시될 구체적이고 명확한 설명을 위해서, 관련 속성 R이 관련성 역할 R인 특정 예(본 발명의 실시예의 데이터 기억 방법 및 데이터 구조에 대한 전술한 설명에서 상세히 나타냄)가 고려될 것이다.

도 8에 도시한 바와 같이 도 3(a)의 데이터 관련성이 재구성될 수 있다.

도 8에 도시한 A 노드 A1 내지 An(n은 1 이상의 정수이지만 모든 경우에 동일한 숫자를 나타내지는 않음) 중에서, A 노드 A1과 A 노드 A1과 관련된 T 노드 T1-1 내지 T1-3이 링크에 의해서 상호 접속된다.

마찬가지로, A 노드 A2와 관련된 T 노드 T2-1, T2-2 및 Tn-1과 A 노드 A2가 링크에 의해서 상호 접속된다.

A 노드 A_n 에 대해서도 동일하게 적용된다. A 노드 A_n 과 관련된 T 노드 T_{n-1} 내지 T_{n-4} 와 A 노드 A_n 이 링크에 의해서 상호 접속된다.

다시 말해, 도 8은 T 노드 T_{2-1} 이 A 노드 A_1 및 A_2 모두와 관련성을 가지며, T 노드 T_{n-1} 이 A 노드 A_2 및 A_n 모두와 관련성을 갖는다는 것을 나타내고 있다.

도 9는 도 8의 데이터 구조를 일반적인 형태로 나타내고 있다.

도 8에서 T 노드 T_{1-1} 로부터, A 노드 A_1 , T 노드 T_{2-1} , A 노드 A_2 및 T 노드 T_{n-1} 을 통해서 A 노드 A_n 에 이르는 링크(851, 852, 853, 854, 855)는 도 9에서 상측에서 바닥쪽으로의 방향으로 확장된 경로(961)를 따라서 표현된다.

특정 예에서, 도 9는 이하의 사항을 추가로 나타내고 있다.

(1) T 노드 T_{1-1} 및 T_{2-1} 은 A 노드 A_1 과 관련되며, 관련성 역할 R_{1-1} 내지 R_{1-m-1} 은 T 노드 T_{1-1} 내지 T_{1-m1} 과 A 노드 A_1 사이에서 관련성(링크)이 정의된다.

(2) T 노드 T_{2-1} 내지 T_{2-m2} 및 도 9에서 생략된 T 노드는 A 노드 A_2 와 관련되며, 관련성 역할 R_{2-1} 내지 R_{2-m2} 는 T 노드 T_{2-1} 내지 T_{2-m2} 와 A 노드 A_2 사이에서 관련성이 정의된다.

(3) 이하, 마찬가지로, 도 9에서 생략된 T 노드는 A 노드와 관련되며 그들 사이에서 관련성 역할 R_i 가 정의된다.

(4) T 노드 T_{n-1} 내지 T_{n-mn} 및 도 9에서 생략된 T 노드는 A 노드 A_n 과 관련되며, 관련성 역할 R_{n-1} 내지 R_{n-mn} ($m1$ 내지 mn 및 n 은 정수)은 T 노드 T_{n-1} 내지 T_{n-mn} 과 A 노드 A_n 사이에서 관련성이 정의된다.

다시 말해, DB 서버(12)에서, 각각의 T 노드는 하나 이상의 노드와 관련되고, 각각의 A 노드는 하나 이상의 T 노드와 관련되며, 이에 의해서 다수의 T 노드는 A 노드를 통해서 서로 관련될 수 있고 다수의 A 노드는 T 노드를 통해서 서로 관련될 수 있다.

DB 서버(12)에는, 도 9에 도시한 바와 같이 관련된 A 노드와 T 노드의 다수의 조합이 기억될 수 있다.

도 10은 도 3b 및 도 4의 데이터 관련성을 일반적인 형태로 도시한 다이어그램이다.

도 10을 참조하면, A 노드 A_1 과 관련된 T 노드 T_{1-1} 내지 T_{1-3} (및 T_{3-1})은 링크에 의해서 A 노드 A_1 에 접속되고, A 노드 A_2 와 관련된 T 노드 T_{201} 내지 T_{2-3} (및 T_{3-2})은 링크에 의해서 A 노드 A_2 에 접속되며, A 노드 A_n 과 T 노드 T_{n-1} 내지 T_{n-3} 및 T_{2-3} (및 T_{3-n})은 링크에 의해서 서로 접속된다.

T 노드 T_{2-3} 은 링크에 의해서 A 노드 A_2 와 A_n 모두에 접속되는데, 이는 T 노드 T_{2-3} 이 A 노드 A_2 및 A_n 모두에 관련된다는 것을 의미한다.

이러한 경우에, A 노드 A_1 , A_2 , A_n 에 의해 관련지어지는 일련의 정보가, 공통의 관련성을 갖는 때에는, 새로운 관련 노드 A_3 가 정의될 수 있다.

예컨대, A 노드 A_1 이, 햄릿의 원작에 관한 정보이며, A 노드 A_2 가 햄릿의 번역에 관한 정보이며, A 노드 A_n 이, 햄릿의 공연에 관한 정보이라고 하면, 이들의 A 노드 A_1 , A_2 , A_n 에 의해 표시되는 관련 정보는, 햄릿에 관한 정보로서 공통성을 갖는다.

그래서, A 노드 A_1 , A_2 , A_n 에 의해 관련지어지는 정보가, 공통성을 갖는 것을 나타내기 위해서, 새로운 관련 노드 A_3 가 정의되어, 데이터베이스에 저장된다.

또한, 도 10 중 파선으로 둘러싸여 도시하는 바와 같이, A 노드 A_1 및 T 노드 T_{1-1} 내지 T_{1-3} 에 의해 표시되는 일련의 정보를 구체적으로 기술하기 위해서, 새로운 T 노드 T_{3-1} 이 정의되어, 데이터베이스에 저장된다.

또한, 마찬가지로, A 노드 A2 및 An의 관련성을 구체적으로 기술하는 새로운 T 노드 T3-2, T3-n이 정의되어, 데이터베이스에 저장된다.

예컨대, T 노드 T3-1에는, 표제의 내용으로서 「햄릿 저자에 대하여」, T 노드 T3-2에는, 표제의 내용으로서 「햄릿 일본어 번역에 대하여」, T 노드 T3-n에는, 표제 내용으로서 「햄릿 공연에 대하여」 등의 데이터가 정의된다. 이들 데이터는 데이터베이스에 저장된다.

또한, 새로운 A 노드 A3와, T 노드 T3-1 내지 T3-n 각과의 사이에, 관련성 역할 R이 정의되어, 데이터베이스에 저장된다.

예컨대, 새로운 A 노드 A3와 T 노드 T3-1 사이에, 관련성 역할 R로서 「원작 정보」가 정의되고, 새로운 A 노드 A3와 T 노드 T3-2 사이에, 관련성 역할 R로서 「번역 정보」가 정의되고, 새로운 A 노드 A3와 T 노드 T3-n 사이에, 관련성 역할 R로서 「공연 정보」가 정의된다. 이들 데이터는 데이터베이스에 저장된다.

마찬가지로, 예컨대, A 노드 A1, A2 및 An과 T 노드 T3-1, T3-2, T3-n 사이에는, 「구상화(reification)」로서 미리 시스템에 의해 정의된 관련성 역할 R이 정의된다.

도 11은, 도 9에 도시한 구조를 채용하는 데이터를 저장하기 위해서 이용되는 관련성 역할(AR(association role)) 테이블을 도시하는 도면이다.

도 12는, 도 9에 도시한 구조를 채용하는 데이터를 저장하기 위해서 이용되는 T 노드용 ID(identifier) 테이블을 도시하는 도면이다.

도 13은, 도 9에 도시한 구조를 채용하는 데이터를 저장하기 위해서 이용되는 A 노드용 ID 테이블을 도시하는 도면이다.

DB 서버(12)에서, 도 9에 도시한 구조에 의해 관련된 A 노드 및 T 노드의 데이터와, T 노드의 데이터는, 도 11의 AR 테이블 및 도 12의 ID 테이블을 이용하여 저장된다.

도 11에 도시하는 AR 테이블의 엔트리 각각은 어떤 하나의 A 노드와, 이 A 노드에 관련된 하나의 T 노드와, 이들 관련된 A 노드와 T 노드 사이에 정의되는 관련성 역할(R)을 나타내고, 어떤 A 노드의 식별자(ID)와, T 노드의 ID와, 관련성 역할(R)을 포함한다.

즉, AR 테이블의 엔트리 각각은, 도 9에 도시한 링크 중 하나의 링크의 한쪽 단부에 있는 A 노드의 ID와, 이 링크의 다른 쪽 단부에 있는 T 노드의 ID와, 이 링크의 속성을 정의하는 관련성 역할을 포함한다.

이러한 엔트리를, 도 9에 도시한 모든 링크(T 노드 T1-1과 A 노드 A1 사이의 링크 및 T 노드 Tn-mn과 A 노드 An 사이의 예지)에 대하여 생성하여, AR 테이블에 저장함으로써, 도 9에 도시한 A 노드와 T 노드와의 관련성은, 도 11의 AR 테이블(제 1 데이터베이스)에 저장된다.

또한, T 노드 각각은 그 내용(T 노드의 명칭, T 노드 자체의 데이터 및 T 노드에 의해 참조되는 데이터 등)을 가진다. 또한, T 노드 각각은, AR 테이블의 각 엔트리에 저장되는 식별자(ID)에 더하여, 이 T 노드의 속성(노드 유형(NT); 표제 속성)이 정의된다. 이하, T 노드 각각이, 그 내용으로서, 그 명칭(노드 명칭(N))만을 갖는 경우를 구체적으로 한다.

도 12에 도시하는 T 노드용 ID 테이블의 엔트리 각각은, 도 9에 도시하는 T 노드 중 하나의 식별자(ID)와, 이 T 노드에 대하여 정의된 속성(노드 유형(NT))과, 이 T 노드의 명칭(노드 명칭(N))을 포함한다.

이러한 엔트리를, 도 9에 도시하는 모든 T 노드 T1-1 내지 Tn-mn에 대하여 생성하고, T 노드용 ID 테이블에 저장한다. 이에 따라, 도 9에 도시하는 모든 T 노드의 데이터가 저장된다.

도 13에 도시하는 A 노드용 ID 테이블의 엔트리 각각은, 도 9에 도시하는 A 노드 중 하나의 식별자(ID)와, 이 A 노드에 대하여 정의된 속성(노드 유형(NT'))과, 이 A 노드의 명칭(노드 명칭(N'))을 포함한다.

이러한 엔트리를, 도 9에 도시한 모든 A 노드 A1 내지 An에 대하여 생성하고, A 노드용 ID 테이블에 저장한다. 이에 따라, 도 9에 도시한 모든 A 노드에 데이터가 저장된다.

또, 도 9에 도시한 A 노드와 T 노드와의 관련성과, A 노드의 데이터 및 T 노드의 데이터를 저장하기 위해서는, 테이블 형태 이외로, 마찬가지로 형태가 채용될 수 있다. 그러나, 이하의 설명에서는, AR 테이블 및 ID 테이블을 이용하는 경우를 구체예로 고려할 것이다.

또, DB 서버(12)의 용도, 구성 또는 처리 내용에 따라, 도 11에 도시하는 바와 같이 AR 테이블에서, 각 엔트리는, T 노드의 식별자(ID)의 대신에, T 노드의 내용(노드 명칭(N))을 포함할 수 있다.

또한 마찬가지로, AR 테이블에서, 각 엔트리는, T 노드의 내용을 더 포함할 수 있다.

데이터 검색

도 14는, 도 6 및 도 7에 도시한 DB 서버(12)에서 데이터 검색 방법을 도시한다.

여기서, 도 14에 도시하는 바와 같이, A 노드에, T 노드 T1 내지 Tn 및 검색 결과라고 되는 T 노드 Tret(T return)가 관련되고, A 노드와, T 노드 T1 내지 Tn 및 Tret 사이에, 관련성 역할 R1 내지 Rn 및 Rret이 정의되고, T 노드 T1 내지 Tn 및 Tret이, 노드 명칭 N1 내지 Nn 및 Nret을 갖는 경우를 구체예로 한다.

DB 서버(12)에서, 검색에 사용되는 Tret에 대해 정의된 관련성 역할 Rret와, 검색을 위해 이용되는 A 노드의 속성(노드 유형 NT; 도 14에서 ANT1, ANT2)와, 검색을 위해 이용되는 T 노드의 관련성 역할 R과 노드 명칭 N의 하나 이상의 조합을 검색 조건으로서 사용한다.

이 검색 조건은, 도 14에 도시하는 바와 같이 예컨대, (Rret, (ANT1, ANT2...), Filter), 여기서 Filter=((R1, N1),(R2, N2),..., (Rn, Nn))과 표기된다.

또, 이 검색 조건은, 후술하는 바와 같이, 또한 T 노드의 속성 NT(제 3 조건데이터)를 더 포함할 수 있다.

상기 검색 조건 중에서, Filter에 포함되는 T 노드의 관련성 역할 R과 노드 명칭 N의 하나 이상의 조합 (R1, N1),(R2, N2), ..., (Rn, Nn) 각각은, 검색을 위한 필터로서 이용되기 때문에, 이하, 검색 필터라고도 한다.

또한, 상기 검색 조건 중에서, A 노드의 속성(ANT1, ANT2 ...)은 생략될 수 있다.

도 15는 도 6 및 도 7에 도시한 DB 서버(12)에서의 검색의 전체적인 처리(S20)를 나타내는 제 1 순서도이다.

도 15에 도시하는 바와 같이, 단계 200(S200)에서, DB 서버(12)는 예컨대 PC(102)(도 6) 또는 DB 서버(12)의 입출력 장치(126)를 이용하여 검색자의 조작에 따라서, 도 14에 도시한 검색 조건을 수신한다.

단계 22(S22)에서, 도 16을 참조하여 후술하는 검색 필터에 근거하여 관련 노드를 선택한다.

단계 24(S24)에서, 도 17을 참조하여 후술하는 노드 ID 및 노드 명칭을 획득한다.

단계 202(S202)에서, DB 서버(12)는, S24의 처리에 의해 검색 결과로서 얻어진 T 노드 Tret의 식별자(노드 ID) 및 노드 명칭(Nret)을 기초로 하여 검색자의 질의에 대한 응답을 생성한다.

이 응답으로서, 노드 명칭 Nret만, 노드 Tret에 의해 참조되는 여러 가지 데이터 또는 노드 Tret를 나타내는 데이터를 사용할 수 있다.

단계 204(S204)에서, DB 서버(12)는 검색자에 의한 질의가 종료했는지 여부를 판단한다.

DB 서버(12)는 검색자에 의한 질의가 종료한 경우에 처리를 종료하고, 이외의 경우에는 S200의 처리로 되돌아간다.

도 16은, 도 15에 도시한 검색 필터를 기초로 한 관련 노드의 선택 처리(S22)를 나타내는 순서도이다.

도 16을 참조하면, 도 15에 도시한 S200의 처리에서, DB 서버(12)는, 검색 조건(Rret, (ANT1, ANT2,...), Filter), 여기서, Filter=((R1, N 1),(R2, N 2),..., (Rn, Nn))을 수신시에, 단계 220(S220)에서, 처리를 위해 사용되는 관련 노드 리스트를 초기화한다.

이 관련 노드 리스트에는, AR 테이블(도 11)부터 얻어진 A 노드 중, 검색 조건 중 A 노드의 속성(노드 유형; ANT1, ANT2,...)중 어느 것을, 그 속성(노드 유형 NT)으로서 포함하는 A 노드의 ID가 저장된다.

또, 검색 조건에서, A 노드의 속성이 생략((ANT1, ANT2,...) = null)되면, S 220의 처리에서, AR 테이블(도 11)로부터 얻어진 A 노드의 모든 ID가 저장된다.

단계 222(S222)에서, DB 서버(12)는 모든 검색 필터(Ri, Ni)에 대하여 실행되었는지 여부를 판단한다.

DB 서버(12)가 모두에 대하여 처리를 한 경우에는 S24(도 15, 도 17)의 처리로 진행한다. 이외의 경우에는, 아직 처리가 대상으로 되지 않은 어느 한 항에 검색 필터(Ri, Ni) 중 어느 하나를 다음 처리 대상으로 하여 S224의 처리로 진행한다.

단계 224(S224)에서, DB 서버(12)는 T 노드용 ID 테이블(도 12)을 검색하여, 검색 필터(Ri, Ni)의 노드 명칭 Ni를 포함하는 엔트리의 모두를 수집하고, 수집된 엔트리에 포함되는 T 노드의 ID의 집합(노드 ID 집합 T)을 생성하는데, 여기서 T = {Ti | ID 테이블에서 노드 명칭 = Ni}이다.

또, 검색 조건이 T 노드의 속성(노드 유형; NT)을 포함하고, 검색 필터가(Ri, Ni, NTi)로 나타나는 경우에, DB 서버(12)는 S224의 처리에서, ID 테이블로부터, 검색 필터(Ri, Ni, NTi)의 노드 명칭 Ni 및 노드 유형 NTi를 포함하는 엔트리를 수집하고, 수집된 엔트리에 포함되는 T 노드의 ID의 집합을 노드 ID 집합 T로서 생성하는 것만을 필요로 한다는 것을 유념하라.

단계 226(S226)에서, DB 서버(12)는 S224의 처리에 의해 얻어진 노드 ID 집합 T이 공집합인지 여부를 판단한다.

DB 서버(12)는 노드 ID 집합 T가 공집합이면, 검색 처리를 종료하기 위한 처리("매칭되는 것이 없음" 메시지 등을 검색자에게 표시)를 하고, 검색 처리를 종료한다. 이외의 경우에는 DB 서버(12)는 S228의 처리로 진행한다.

단계 228(S228)에서, DB 서버(12)는 AR 테이블(도 11)을 검색하여, 관련 노드 리스트 A를 갱신한다.

즉, DB 서버(12)는 AR 테이블로부터, 검색 필터(Ri, Ni)의 관련성 역할 Ri와, S224의 처리에 의해 얻어진 노드 ID 집합 T에 포함되는 T 노드의 ID를 포함하는 엔트리의 모두를 수집하고, 수집된 엔트리에 포함되는 A 노드의 ID를, 관련 노드 리스트 A(A = A_j | 역할 Ri, T 노드의 ID Ti(모든 i), A 노드 ID ∈ AR 테이블 내의 A)에 저장한다.

단계 230(S230)에서, DB 서버(12)는 S228의 처리에 의해 얻어진 관련 노드 리스트 A가 공집합인지 여부를 판단한다.

DB 서버(12)는 관련 노드 리스트 A가 공집합이면, 검색 처리를 종료하기 위한 처리를 하여, 검색 처리를 종료한다. 이외의 경우에는 S232의 처리로 진행한다.

단계 232(S232)에서, DB 서버(12)는 검색 조건에 포함되는 처리되어 있지 않은 검색 필터를 판독하고, S222의 처리로 되돌아간다.

도 17은 도 15 및 도 16에 도시한 노드 ID 및 노드 명칭 획득 처리(S24)를 나타내는 순서도이다.

도 17을 참조하면, 검색 필터에 근거하여 관련 노드의 선택 처리(S22)가 종료하면, 단계 240(S240)에서, DB 서버(12)는 AR 테이블(도 11)을 검색하여, T 노드 ID 집합 T을 생성한다.

즉, DB 서버(12)는 AR 테이블로부터, 검색 조건에 포함되는 관련성 역할 Rret과, S22(S228)의 처리에 의해 얻어진 관련 노드 리스트 A에 포함되는 A 노드의 ID를 포함하는 엔트리의 모두를 수집하고, 수집된 엔트리에 포함되는 T 노드의 ID의 집합(T 노드 ID 집합 T)을 생성한다(T={Tm | 역할= Rret, A 노드의 ID ∈ AR 테이블 내의 A).

단계 242(S242)에서, DB 서버(12)는 S240의 처리에서 얻어진 T 노드 ID 집합 T가 공집합인지 여부를 판단한다.

DB 서버(12)는 T 노드 ID 집합 T가 공집합이면 종료 처리를 하여, 검색처리를 종료하고, 이외의 경우에는 S244의 처리로 진행한다.

단계 244(S244)에서, DB 서버(12)는 T 노드용 ID 테이블(도 12)을 검색하여, 노드 ID와 노드 명칭의 집합 P를 생성한다.

즉, DB 서버(12)는, ID 테이블로부터, S240의 처리에서 생성된 T 노드 ID 집합 T에 포함되는 T 노드의 ID Tm 중 어느 것을 포함하는 엔트리 모두를 수집하고, 이 엔트리에 포함되는 노드 명칭 Nm과, T 노드의 ID Tm의 집합 $P(P=(Tm, Nm) | T \text{ 노드의 ID} = \text{ID 테이블 내의 Tm(모든 m)})$ 를 생성한다.

단계 246(S246)에서, DB 서버(12)는 S244의 처리에서 얻어진 T 노드 ID와 T 노드 명칭의 집합 P가 공집합인지 여부를 판단한다.

DB 서버(12)는 T 노드 ID와 T 노드 명칭의 집합 P가 공집합이면, 종료 처리를 하여 검색 처리를 종료한다. 이외의 경우에는 S202의 처리로 진행한다.

이 집합 P는 도 15에 도시한 S202의 처리에서, 검색자에 대한 응답을 생성하기 위해서 이용된다.

DB 프로그램2

도 18은 도 6 및 도 7에 도시한 DB 서버(12)에서 실행되는 DB 프로그램2의 구성을 도시하는 도면이다.

도 18에서, 도시의 명확성을 위해, 데이터의 흐름을 나타내는 선은, 적절히 생략되어 있다.

도 18을 참조하면, DB 프로그램 2는 DB 관리부(20), DB부(24) 및 DB 검색부(26)로 구성된다.

DB 관리 장치(20)는, 관리 조작 수신기(200), AR 엔트리 생성 장치(202), ID 엔트리 생성 장치(204), AR 데이터베이스 관리 장치(ARDB)(206) 및 ID 데이터베이스 관리부(IDDB 관리 장치)(208)로 구성된다.

DB 장치(24)는 AR 데이터베이스(ARDB)(240), T 노드용 ID 데이터베이스(IDDB)(242) 및 A 노드용 IDDB(244)로 구성된다.

DB 검색 장치(26)는 검색 조작 수신기(260), 검색 조건 생성 장치(262), 검색 제어 장치(264), AR 데이터베이스 검색 장치(ARDB)(266) 및 ID 데이터베이스 검색 장치(IDDB)(268)로 구성된다.

DB 프로그램 2는 예컨대, 기록 매체(130)(도 7)로부터 DB 서버(12)에 공급되어, 메모리(124)에 로딩되어 특히 DB 서버의 하드웨어(12의 하드웨어를 이용(이하 각각의 프로그램과 마찬가지로)를 이용하여 DB 서버(12)에서 실행된다.

DB 프로그램 2는, 이들의 구성 부분에 의해, 도 9 내지 도 13을 참조하여 설명한 AR 데이터베이스(도 11) 및 ID 데이터베이스(도 12 및 도 13)를 생성하고 데이터베이스를 이용한 데이터의 검색(도 14 내지 도 17)을 한다.

DB 장치(24)에서, ARDB(240)는 도 11에 도시한 AR 테이블을 저장한다.

IDDB(242)는 도 12에 도시한 T 노드용 ID 테이블을 저장한다.

IDDB(244)는, 도 13에 도시한 A 노드용 ID 테이블을 저장한다.

또한, 도 18에는 도 12 및 도 13에 도시한 T 노드용 ID 테이블 및 A 노드용 ID 테이블이, 각각 IDDB(242, 244)에 저장되는 경우를 구체예로 도시한다. 그러나 T 노드용 ID 테이블과, A 노드용 ID 테이블은 데이터베이스에 저장될 수 있다.

T 노드용 ID 테이블과, A 노드용 ID 테이블은, 반드시 별도로 생성되지 않을 수 있고, 하나의 데이터베이스 내에 일체적으로 생성될 수 있다.

DB 관리 장치(20)에서, 관리 조작 수신기(200)는 AR 테이블 및 ID 테이블에 저장된 데이터를 관리하거나 변경하기 위한 조작을 입출력 장치(126)(도 7)로부터, 또는, 네트워크(100)를 통해서 PC(102)(도 6)로부터 수신하여, ARDB 관리 장치(206) 및 IDDB 관리 장치(208)에 출력한다.

관리 조작 수신부(200)는 A 노드와 T 노드를 지정하는 사용자 조작, A 노드와 T 노드의 조합, A 노드와 T 노드 사이에 규정된 관련성 역할(R), A 노드와 T 노드에 할당된 식별자(ID), T 노드에 할당된 노드 명칭(N), 및 T 노드에 있어서 규정된 속성(도 9)을 수신하여, 그 조작을 AR 엔트리 작성부(202)와 ID 엔트리 작성부(204)에 출력한다.

예를 들면, 관리 조작 수신부(200)는 A 노드와 T 노드를 나타내는 사용자 인터페이스(UI) 이미지와, 입력/출력 장치(126) 내의 도 14에 나타난 이들 노드간의 조합을 표시하고, UI 이미지에 대한 사용자의 조작을 수신하고, 지정을 수용한다.

AR 엔트리 작성부(202)는 관리 조작 수신부(200)로부터 입력된 사용자 지정에 따라서 도 11에 나타난 AR 테이블의 엔트리를 작성하고, 그 엔트리를 ARDB 관리부(206)로 출력한다.

ARDB 관리부(206)는 AR 엔트리 작성부(202)로부터 입력된 AR 테이블의 엔트리를 ARDB(240)에 저장된 AR 테이블에 추가한다.

ARDB 관리부(206)는 관리 조작 수신부(200)로부터 입력된 사용자 조작에 따라서 ARDB(240)에 저장된 AR 테이블의 내용을 수정한다.

ARDB 관리부(206)는 ARDB 검색부에 의한 검색 요청에 따라서 ARDB(240)에 저장된 AR 테이블의 엔트리를 검색하여, 그 엔트리를 ARDB 검색부(266)로 출력한다.

ID 엔트리 작성부(204)는 관리 조작 수신부(200)로부터 입력된 사용자 지정에 따라서 도 12 및 도 13에 나타난 T 노드와 A 노드 ID 테이블에 대한 엔트리를 작성하여, 그 엔트리를 IDDB 관리부(208)로 출력한다.

IDDB 관리부(208)는 ID 엔트리 작성부(204)로부터 입력된 T 노드 ID 테이블의 엔트리를 IDDB(242)에 저장된 T 노드 ID 테이블에 추가한다.

IDDB 관리부(208)는 ID 엔트리 작성부(204)로부터 입력된 A 노드 ID 테이블의 엔트리를 IDDB(244)에 저장된 A 노드 ID 테이블에 추가한다.

IDDB 관리부(208)는 관리 조작 수신부(200)로부터 입력된 사용자 조작에 따라서 IDDB(242, 244)에 저장된 ID 테이블의 내용을 수정한다.

IDDB 관리부(208)는 IDDB 검색부(268)에 의한 검색 요청에 따라서 IDDB(242, 244)에 저장된 ID 테이블의 엔트리를 검색하여, 그 엔트리를 IDDB 검색부(268)로 출력한다.

DB 검색부(26)에서, 검색 조작 수신기(260)는 네트워크(100)를 통해 입력/출력 장치(126)(도 7) 또는 PC(102)(도 6)로부터, 도 14 내지 도 17에 나타난 검색 처리 동안에 사용되는 검색 조건(도 14, 옵션의 T 노드의 속성(노드 타입(NT))이 추가로 포함될 수 있음)을 지정하는 검색자 조작을 수취한다.

검색 조작 수신기(260)는 수취된 조작을 검색 조건 작성부(262)로 출력한다.

예를 들면, 검색 조작 수신기(260)가 원어의 질문 형태로 검색 조건을 수용할 때, 검색 조건 작성부(262)는 질의 문장을 분석하여 단어를 추출한다.

다음에, 검색 조건 작성부(262)는 ARDB 검색부(266), IDDB 검색부(268), ARDB 관리부(206) 및 IDDB 관리부(208)를 통해 ARDB(240)와 IDDB(242)에 저장된 AR 테이블과 ID 테이블을 검색하여, 검색 조건으로서 사용된 단어를 추출한다.

추가로, 검색 조건 작성부(262)는 질의문의 문장의 구조에 따라서 그 추출된 단어를 조합하고, 도 14에 나타난 (Rret, (ANT1, ANT2, ...), ((R1, N1), (R2, N2), ..., (Rn, Nn)))의 형태로 검색 조건을 검색하고, 검색 조건을 검색 제어부(264)로 출력한다.

검색자가 도 14에 나타낸 (Rret, (ANT1, ANT2, ...), ((R1, N1), (R2, N2),..., (Rn, Nn)))의 형태로 검색 조건을 직접 지정하는 경우에, 검색 조건 작성부(262)는 생략될 수 있음을 알아야 한다.

검색 조건 작성부(262)는 검색자에 의한 검색 조건 (Rret, (ANT1, ANT2, ...), ((R1, N1), (R2, N2),..., (Rn, Nn)))의 검색을 보조하는 툴일 수 있다.

도 15 내지 도 17에 나타낸 바와 같이, 검색 제어부(264)는 검색 조건 작성부(262)(검색 조작 수신기(260))로부터 입력된 검색 조건(Rret, (ANT1, ANT2, ...), ((R1, N1), (R2, N2),..., (Rn, Nn)))에 따라서 ARDB 검색부(266)와 IDDB 검색부(268)를 제어하여, ARDB 관리부(206)와 IDDB 검색부(208)를 통해 ARDB(240)(AR 테이블, 도 11)와 IDDB(242, 244)(ID 테이블, 도 12 및 도 13)에서의 검색을 수행한다.

검색 결과(세트 P, 도 17)가 검색 조건에 기초한 검색에 의해 획득될 때, 검색 제어부(264)는 검색 결과에 기초하여 응답을 작성하고, 그 응답을 입력/출력 장치(126)(도 7)에 표시하거나, 그 응답을 검색자로서의 네트워크(100)(도 6)를 통해 PC(102)의 입력/출력 장치(126)에 표시한다.

ARDB 검색부(266)는 검색 제어부(264)의 제어하에 ARDB 관리부(206)를 통해 ARDB(240)(AR 테이블, 도 11)를 검색하여, 검색 결과를 검색 제어부(264)로 반환한다.

IDDB 검색부(268)는 검색 제어부(264)의 제어하에 IDDB 관리부(208)를 통해 IDDB(242, 244)(ID 테이블, 도 12 및 도 13)를 검색하여, 검색 결과를 검색 제어부(264)로 반환한다.

전체 조작

이하에서는, 도 6 및 도 7에 나타낸 DB 서버(12)(DB 프로그램(2), 도 18)의 전체 조작을 특정 실시예와 관련하여 설명될 것이다.

AR 테이블과 ID 테이블의 작성

우선, DB 프로그램(2)의 DB 관리부(20)에 의한 AR 테이블과 ID 테이블의 작성 처리가 설명될 것이다.

도 19는 도 6 및 도 7에 나타낸 DB 서버(12)(DB 프로그램(2), 도 18)에 입력된 데이터와, 포함된 데이터를 검색하는 검색 조건을 도시한다.

예를 들면, 도 19에 나타낸 관련 데이터가 DB 프로그램(2)의 관리 조작 수신부(200)에 입력된다.

도 19에 나타낸 데이터는 이하에 설명되는 관련 A 노드와 T 노드를 포함하며, 노드 명칭은 T 노드에 할당된다(그러나, A 노드의 속성(노드 타입(NT) 및 노드 명칭과, T 노드의 속성(노드 타입(NT)))은 이하의 단락 (1) 내지 (8) 및 도 19에서 생략된다).

(1) "A 노드 A1" 및 "T 노드 T11"은 서로 관련되어 있으며, "작자"의 관련성 역할(R)이 서로간에 규정되며, 노드 명칭 "세익스피어(동명 이인)"가 "T 노드 T11"에 할당된다.

(2) "A 노드 A9" 및 "T 노드 T92 및 T41"은 서로 관련되어 있으며, "작자"와 "작품"의 관련성 역할(R)이 그들 사이에 규정되며, 노드 명칭 "베니스의 상인"과 "세익스피어"가 "T 노드 T92 및 T42"에 할당된다.

(3) "A 노드 A4" 및 "T 노드 T41 및 T42"은 서로 관련되어 있으며, "작품"과 "작자"의 관련성 역할(R)이 그들 사이에 규정되며, 노드 명칭 "햄릿"이 "T 노드 T42"에 할당된다.

(4) "A 노드 A13" 및 "T 노드 T42"은 서로 관련되어 있으며, "각본"의 관련성 역할(R)이 그들 사이에 규정된다.

(5) "A 노드 A19" 및 "T 노드 T42"은 서로 관련되어 있으며, "원작"의 관련성 역할(R)이 그들 사이에 규정된다.

(6) "A 노드 A10" 및 "T 노드 T42"은 서로 관련되어 있으며, "원작"의 관련성 역할(R)이 그들 사이에 규정된다.

(7) "A 노드 A10" 및 "T 노드 T103 및 T101"은 서로 관련되어 있으며, "출판"과 "번역"의 관련성 역할(R)이 서로간에 규정되며, 노드 명칭 "○○ 출판사"가 "T 노드 T103"에 할당된다.

(8) "A 노드 A19" 및 "T 노드 T191"은 서로 관련되어 있으며, "번역"의 관련성 역할(R)이 그들 사이에 규정된다.

관리 조작 수신부(200)는 입력 데이터를 수신하여, 그 데이터를 AR 엔트리 작성부(202)와 ID 엔트리 작성부(204)로 출력한다.

AR 엔트리 작성부(202)는 도 19에 나타난 데이터로부터 AR 테이블의 엔트리 각각을 작성하여, 엔트리를 ARDB 관리부(206)로 출력한다.

도 20은 AR 엔트리 작성부(202)(도 18)와 ARDB 관리부(206)에 의해 작성되어 ARDB(240)에 저장된 AR 테이블을 도시한다.

이하의 도면에서, NULL은 속성(노드 타입)/명칭(노드 명칭)이 없음을 표시한다는 것을 알아야 한다.

ARDB 관리부(206)는 AR 엔트리 작성부(202)로부터 입력된 AR 테이블의 엔트리를 ARDB(240)에 저장된 AR 테이블에 순차적으로 추가한다.

AR 엔트리 작성부(202)와 ARDB 관리부(206)에 의한 처리의 결과로, 도 20에 나타난 AR 테이블은 도 19에 나타난 데이터로부터 작성되어 ARDB(240)에 저장된다.

도 21은 ID 엔트리 작성부(204)(도 18)와 IDDB 관리부(208)에 의해 작성되어 IDDB(242)에 저장된 T 노드 ID 테이블을 도시한다.

ID 엔트리 작성부(204)는 도 19에 나타난 데이터로부터 T 노드의 ID 테이블의 각 엔트리를 작성하여, IDDB 관리부(208)에 출력한다.

IDDB 관리부(208)는 ID 엔트리 작성부(204)로부터 입력된 T 노드의 ID 테이블의 엔트리를 IDDB(242)에 저장된 ID 테이블에 순차적으로 추가한다.

ID 엔트리 작성부(204) 및 IDDB 관리부(208)에 의한 처리의 결과로, 도 19에 나타난 데이터로부터, 도 21에 나타난 바와 유사한 T 노드의 ID 테이블이 작성되어, IDDB(242)에 저장된다.

도 22는 ID 엔트리 작성부(204)(도 18) 및 IDDB 관리부(208)에 의해 작성되는 A 노드에 대한 ID 테이블을 예시하는 도면이다.

ID 엔트리 작성부(204)는 도 19에 나타난 데이터로부터 A 노드의 ID 테이블의 각 엔트리를 작성하여, IDDB 관리부(208)에 출력한다.

IDDB 관리부(208)는 ID 엔트리 작성부(204)로부터 입력된 ID 테이블의 엔트리를 IDDB(244)에 저장된 A 노드의 ID 테이블에 순차적으로 추가한다.

ID 엔트리 작성부(204) 및 IDDB 관리부(208)에 의한 처리의 결과로, 도 19에 나타난 데이터로부터, 도 22에 나타난 바와 유사한 A 노드의 ID 테이블이 작성되어, IDDB(244)에 저장된다.

데이터의 검색

예를 들면, 검색자가 DB 서버(12)의 입출력 장치(126)(도 7)에 대하여, "작가 셰익스피어를 작자로 하는 작품 중 희곡 햄릿을 원작으로 하는 번역판을 출판하고 있는 출판사명은 무엇인가?"이라고, 검색 조건을 질문 문장의 형식으로 입력하면, 검색 조건 작성부(262)는 이 질문 문장을 분석하여, 도 19에 나타내는 바와 같이 이 질문 문장을 제 1 절반과 제 2 절반의 2개로 분할한다.

질문 문장의 제 2 절반 "작가 세익스피어를 작자로 하는 작품"에서, "작품"에 관한 데이터를 수집하기 위해, 검색 조건 작성부(262)는 검색 결과로 되는 T 노드 Tret의 관련성 역할(Rret)을 "작품"으로서 설정한다.

검색 조건 작성부(262)는, 제 1 절반의 "세익스피어를 작자로 한다"의 부분으로부터 검색 필터(R1="작자", N1="세익스피어")를 작성한다.

또한, 검색 조건 작성부(262)는, T 노드 Tret의 관련성 역할(Rret="작품")과, 검색 필터(R1="작자", N1="세익스피어")로부터, 질문 문장의 제 2 절반에 대응하는 검색 조건(Rret, (ANT1),((R1, N1)))=(작품, (공백),((작자, 세익스피어)))을 작성한다.

질문 문장의 제 1 절반 "희곡 햄릿을 원작으로 하는 번역판을 출판하고 있는 출판사명은 무엇인가?"에서, "출판하고 있는 출판사"에 관한 정보를 검색하기 위해서, 검색 조건 작성부(262)는 검색 결과로 되는 T 노드 Tret의 관련성 역할(Rret)을 "출판"으로서 설정한다.

검색 조건 작성부(262)는, 제 1 절반에 포함되는 희곡 "햄릿"의 "원작"의 조건으로부터 제 1 검색 필터(R1="원작", N1="햄릿")를 작성하고, 또한, 제 1 절반에 포함되는 "번역판"으로부터 제 2 검색 필터(R2="번역판", N2="공백(미지정)")를 작성한다.

또한, 검색 조건 작성부(262)는, 관련성 역할(Rret="출판")과, 제 1 검색 필터(R1="원작", N1="햄릿") 및 제 2 검색 필터(R2="번역판", N2="공백")로부터, 질문 문장의 제 1 절반에 대응하는 검색 조건(Rret, (ANT1),((R1, N1), (R2, N2)))=(출판, (공백), ((원작, 햄릿), (번역판, 공백)))를 작성한다.

검색 조건 작성부(262)에 의해 작성된 검색 조건에 근거하여, 검색 제어부(264)는, 이하에 나타내는 바와 같이, ARDB 검색부(266) 및 IDDB 검색부(268)를 거쳐서 ARDB(240)(AR 테이블; 도 20) 및 IDDB(242)(T 노드의 ID 테이블; 도 21)를 검색하여, 검색 결과를 얻는다.

우선, 질문 문장의 제 2 절반으로부터 얻어진 검색 조건으로부터, 검색 제어부(264)는 다음을 수행한다.

(1) 검색 제어부(264)는 IDDB(242)(도 18)에 저장된 T 노드의 ID 테이블을 참조하여, 노드 명칭이 "세익스피어"인 T 노드의 모든 식별자(ID)를 검색하고, 이 처리의 결과로서 T11, T31, T41, T51 및 T81(도 21)을 얻는다.

(2) 검색 제어부(264)는 ARDB(240)에 저장된 AR 테이블(도 20)을 참조하여, 역할이 "작자"이며, T 노드 ID가 (1)의 처리에서 얻어진 T 노드의 ID와 일치하는 모든 A 노드 ID를 검색하여, 이 처리의 결과로서, A1, A4 및 A9를 얻는다.

(3) 검색 제어부(264)는 AR 테이블을 참조하여, (2)의 처리에서 얻어진 A 노드 ID 중에서 관련성 역할(R)이 "작품"인 A 노드에 대응하는 T 노드의 식별자(ID; 일반적으로는 복수개)를 검색하여, 이 처리의 결과로서, T42 및 T92를 얻는다.

(4) 검색 제어부(264)는 ID 테이블을 참조하여, (3)의 처리에 의해 얻어진 T 노드 ID에 대응하는 ID의 노드 명칭을 그 T 노드의 식별자(ID)와 함께 검색 결과로서 설정한다.

즉, 검색 제어부(264)는, (1)~(4)의 처리에 의해 질문 문장의 후반부분으로부터 얻어진 검색 조건(작품, (공백), ((작자, 세익스피어)))에 근거하는 검색을 행하여, 검색 결과(T42, 햄릿), (T92, 베니스의 상인)를 얻는다.

다음에, 질문 문장의 제 1 절반으로부터 얻어진 검색 조건과 제 2 절반에 대응하는 검색 결과로부터, 검색 제어부(264)는 다음을 수행한다.

(5) 질문 문장의 제 2 절반에 대응하는 검색 결과(T42, 햄릿), (T92, 베니스의 상인)로부터, 검색 제어부(264)는, 노드 명칭이 제 1 검색 필터(원작, 햄릿)에 대응하는 (T42, 햄릿)을 선택하여, 그 노드 식별자(ID)(T42)를 얻는다.

(6) 검색 제어부(264)는 AR 테이블을 참조하여, 관련성 역할이 "원작"이며, T 노드 ID가 (5)의 처리에 의해 얻어진 노드 식별자(ID)와 일치하는 모든 A 노드의 ID를 검색하여, 이 결과로서 A10 및 A19를 얻는다.

(7) 검색 제어부(264)는 AR 테이블을 참조하여, (6)의 처리에 의해 얻어진 A 노드 ID 중에서 역할이 "번역관"인 모든 ID를 검색하여, 이 결과로, A10 및 A19를 얻는다.

(8) 검색 제어부(264)는 AR 테이블을 참조하여, (7)의 처리에 의해 얻어진 A 노드 ID 중에서 역할이 "출판"인 A 노드에 대응하는 T 노드 ID를 검색하여, 이 결과로 T103을 얻는다(번역관이 복수의 출판사에서 나온 경우에는, 복수의 T 노드를 얻게 된다).

(9) 검색 제어부(264)는 ID 테이블을 참조하여, (8)의 처리에 의해 얻어진 T 노드 ID에 대응하는 노드 ID의 노드 명칭을 그 노드 ID와 함께 검색 결과(T103, ○○ 출판사)로서 설정한다.

즉, 검색 제어부(264)는, (5)~(9)의 처리에 의해, 질문 문장의 제 1 절반으로부터 얻어진 검색 조건(출판, (공백), ((원작, 랩트), (번역관, 공백)))에 근거하는 검색을 행하여, 검색 결과(T013, ○○ 출판사)를 얻는다.

(10) 검색 제어부(264)는 검색 결과를 입출력 장치(126)(도 7) 등에 표시하여, 검색자에게 나타낸다.

제 2 데이터베이스 시스템(3)

도 23은 본 발명의 실시예에 따른 제 2 DB 시스템(3)의 구성을 도시한 도면이다.

도 23을 참조하면, DB 서버(12)와, DB 서버(12)와 유사한 하드웨어(도 7) 상에서 DB 프로그램(2)의 DB 관리 유닛(20)과 DB 유닛(24)을 동작시키는 DB 서버(30), 및 DB 서버와 유사한 하드웨어 상에서 DB 프로그램(2)의 DB 검색 유닛(26)을 네트워크(100)를 통해 동작시키는 수집 장치(32)를 상호 접속시킴에 의해 제 2 DB 시스템(3)이 구성된다.

따라서, DB 프로그램(2)은 하나의 컴퓨터상에서 항상 실행될 필요는 없으며, 실행될 네트워크를 통해 상호 접속된 다수의 컴퓨터에 분산될 수 있다.

제 3 DB 시스템(4)

이하에서는, 디렉토리 구조에 기반하여 관련 정보를 저장 및 검색하도록 구성된 본 발명의 실시예에 따른 제 3 DB 시스템(4)이 설명될 것이다.

도 24에는 제 3 DB 시스템(4)의 구성이 도시된다.

도 24를 참조하면, DB 시스템(4)은 PC(102), 수집 장치(40), DB 관리 서버(5) 및 n개의 DB 서버(6-1 내지 6-n)를 네트워크(100)를 통해 상호 접속시킴으로서 구성된다.

DB 서버(6-1 내지 6-n)들과 같은 다수의 부품들 중 임의의 한 부품이 지정되지 않을 경우에는, 단순히 DB 서버(6)를 지칭함을 알아야 한다.

수집 장치(40)와, DB 관리 서버(5) 및 DB 서버(6)의 하드웨어는 도 7에 도시된 구성을 채용한다.

DB 서버(6)가 DB 서버(12)(도 6) 및 도 6에 도시된 제 1 DB 시스템(1)의 DB 프로그램(2)(도 18)의 경우에서와 마찬가지로의 검색 기능을 포함하면, 수집 장치(40)는 불필요하게 된다.

또한, DB 시스템(4)에 있어서, 복제 서버(미러 서버; DB 서버(6'))는 각 DB 서버(6)에 대응하여 제공된다.

이하에서는 DB 서버(6-1 내지 6-n)를 DB 서버(A 내지 N)라 할 것이다.

다음, 도 24의 DB 시스템(4)에서의 데이터 표시가 설명될 것이다.

도 25는 도 24의 DB 시스템(4)에 저장된 데이터의 그래픽 표시를 도시한 도면이다.

도 26은 도 25에 도시된 데이터의 디렉토리 정보 트리 표시를 도시한 도면이다.

도 8을 참조하여 상술한 바와 같이, 노드 명칭(A노드의 노드 명칭은 "관련 노드에 대응하는 데이터"이고, T 노드의 노드 명칭은 "토픽 노드에 대응하는 데이터"임)등이 A 및 T에 대해 정의되고, A 노드와 T 노드간에 관련 역할이 정의되면, 이들 노드에 의해 관련된 관련 데이터는 도 25에 도시된 그래픽 형태로 표시될 수 있다.

도 25는 T 및 A 노드의 노드 유형(클래스)을 정의하는 노드(T1, T3-1 및 T3-2)에 대한 노드 명칭으로서 "위치 정보", "저장" 및 "영역"이 정의되고, T 노드(T2-1 및 T2-2)에 대한 노드 명칭으로써 "저장 1" 및 "영역 3"이 정의되고, A 노드(A1)에 대한 노드 명칭으로써 "위치"가 정의되고, T 및 A노드간의 관련 역할로서 "빌딩" 및 "장소"가 정의되는 것을 도시한 도면이다.

A 노드(A2-1 및 A2-2)는 노드 유형(클래스)을 정의하는 노드와, 그의 순시치로서의 노드(엔티티에 대응하는 노드)를 나타낸다. 따라서, 노드(A2-1 및 노드 A2-2)는, 그들이 소정 목적에 따라 사용자에게 의해 독립적으로 정의된 역할에 기반한 것이 아니라, 시스템에 의해 사전 정의된 "클래스" 및 "순시치"의 두 역할에 기반하여 T 노드의 관련성을 인에이블한다는 점에서 전용 노드이다.

도 25에 도시된 그래픽 표시의 관련 데이터는 도 26에 도시된 바와 같이 LDAP(Lightweight Directory Access Protocol)의 데이터 모델인 DIT(Directory Information Tree)의 형태로 표시될 수 있다.

즉, 도 25에 도시된 관련 데이터는 A 노드에 대응하는 엔트리와, 관련 역할에 대응하고 도 26에서 실선에 의해 표시된 바와 같이 속성들로서 T 노드 데이터를 가진 엔트리에 의해 표시될 수 있으며, 그 속성들은 DIT 표시에 따라 도 26에서 점선으로 표시된 바와 같은 제 1 관련, 제 2 관련 등(관련 #1, #2, ...)을 나타내는 엔트리들 및 각 DB 서버(6)의 상위 엔트리로서 가상적으로 제공되는 가상 루트 엔트리 아래에 배치된다.

다시 말해, 도 26에 도시된 경우에 있어서, "빌딩" 및 "장소"의 두 역할에 대응하는 엔트리는 노드 유형(assocType)이 "위치 정보"이고, 관련 데이터를 나타내는 엔트리 "위치" 바로 아래에 제공되고, 그 역할을 수행하는 부재 "저장 1" 및 "영역"은 대응하는 엔트리의 속성이다.

도 26에 도시된 DIT 형태에 있어서, "디렉토리"는 데이터의 위치 선정을 용이하게 하고 그 데이터를 수집 및 갱신하는 수단을 제공하기 위해, 실세계와 관련된 이해하기 쉬운 형태로 실세계의 여러 객체에 관하여 데이터를 구성하도록 정의된다.

"엔트리"는 객체 클래스에 따라 실세계의 객체에 관하여 데이터를 배열/분류하고, 그 데이터를 디렉토리 정보로서 나타내거나, 그 디렉토리에 저장된 데이터로서 정의된다.

"디렉토리 정보 트리"는 엔트리(디렉토리 정보)가 계층적 방식으로 관리된 경우에, 트리 구조의 계층적 관계를 나타내도록 정의된다.

다시 말해, 도 26의 사각형들은 엔트리에 대응하고, 그것은 트리 구조 내의 그의 계층적 관계를 나타내는 디렉토리 정보 트리(Directory Information Tree : DIT)이다.

상술한 바와 같이, 엔트리는 객체에 관한 데이터 세트이며, 이 객체에 관한 데이터를 "속성"이라 한다.

그 속성은 "속성 유형"과 하나 이상의 값인 "속성값"을 포함한다. 도 26을 참조하면, 각 엔트리 구조는 아래의 테이블 21 내지 23에 도시된 바와 같이 "속성 유형 : 속성값" 형태의 LDAP의 엔트리 표시로 도시된다.

"ou = association #1" 바로 아래의 엔트리에 있어서, 예를 들어, 속성 유형 객체 클래스의 속성값은 관련 노드이고, 속성 유형 cn(공통 명칭)의 속성값은 위치이고 등의 방식으로 엔트리의 각 속성이 정의된다.

[표 21]

dn : associd = 관련 노드 ID
 objectclass : 관련 노드
 associd : 관련 노드 ID
 assocType : 위치 정보
 cn : 위치
 role : 빌딩
 role : 장소

[표 22]

dn : 역할 = 빌딩, assoid : 관련 노드 ID
 objectclass : 관련 역할
 cn : 빌딩
 role : 빌딩
 member : 상점 1

[표 23]

dn : 역할 = 장소, associd = 관련 노드 ID
 objectclass : 관련 역할
 cn : 장소
 role : 장소
 member : 영역 3

도 27에는 디렉토리 구조 내에 표시된 관련 데이터를 분할하는 방법이 도시된다.

도 28에는 도 27에 도시된 DB 서버(6-1)(A)의 엔트리 dn_A2와 DB 서버 6-n(N)의 엔트리 dn_N간의 참조 관련성이 도시된다.

도 27에 도시된 바와 같이, DIT 형태로 표시된 관련 데이터는 디렉토리 구조에 따라 분할될 수 있다.

도 27에 도시된 경우에 있어서, 관련 데이터는 엔트리 dn_A와 하위 엔트리 dn_A1 및 dn_A2를 포함하는 디렉토리 구조에 의해 표시되며, 엔트리 dn_A1 또는 하위의 디렉토리에 대응하는 관련 데이터는 엔트리 dn_B 내지 dn_(N-1)에 대응하여 분할되고, 엔트리 dn_A2 또는 그 하위의 디렉토리에 대응하는 관련 데이터는 엔트리 dn_N 또는 그 하위의 디렉토리 구조에 대응한다(도 27에 도시된 예시에서, 상위 엔트리로서 엔트리 dn_B 내지 dn_N을 가진 디렉토리 구조는 서브 트리이고, 그러한 서브 트리와, 상위 엔트리로서 엔트리 dn_A를 포함하는 디렉토리 구조가 디렉토리 트리임을 알아야 한다).

예를 들어, 도 28에 도시된 바와 같이, 도 27에 도시된 DB 서버 6-n(N)의 상위 엔트리 dn_N은 DB 서버 6-1(A)의 엔트리 dn_A2 아래에 있으며, dn_N은 dn_A2를 참조한다.

따라서, 디렉토리 트리에 따라 분할된 관련 데이터는 분할되어 도 27에 도시된 DB 서버 6-1 내지 6-n에 저장될 수 있다.

도 29에는, 분할되어 도 27에 도시된 바와같이 DB 서버 6-1 내지 6-n(DB 서버 A 내지 N)에 저장된 관련 데이터를 관리하는데 이용되는 디렉토리 트리 테이블이 도시된다.

분할되어 도 27에 도시된 바와 같이 DB 서버 6-1 내지 6-n(A 내지 N)에 저장된 관련 데이터를 관리하기 위해, 도 29에 도시된 것과 유사한 디렉토리 트리 테이블이 이용된다.

도 29를 참조하면, 디렉토리 트리 테이블은 DB 서버 6-1 내지 6-n의 서버 명칭(A 내지 N)과, DB 서버(6)의 상위 엔트리(도 26의 가상 엔트리와 유사한 dn_A 내지 dn_N)와, DB 서버(6)의 상위 엔트리에 의해 참조되는 엔트리(참조 엔트리) 및 DB 서버(6) 내에 대응하는 방식으로 복제 서버(미리 서버(제 2 데이터베이스 서버))가 있을 경우의 장치 명칭(A', B'등)을 포함한다.

도 30에는, 도 27 및 28에 도시된 DB 서버 6-1 내지 6-n에 저장된 디렉토리 트리(또는 서브 트리)의 상위 엔트리(dn_A 내지 dn_N)와, 디렉토리 트리(또는 서브 트리)에 저장된 관련 데이터를 분류하는데 이용되는 분류 유형/분류 값(분류 데이터)을 상호 상관시키는 데이터 테이블이 도시된다.

도 25 및 도 26에 도시된 바와 같이, 디렉토리 구조에 따라 분할되고 DB 서버 6-1 내지 6-n(A 내지 N)에 저장된, DIT 형태의 디렉토리 구조로 표시된 관련 데이터는 DB 서버 6-1 내지 6-n(A 내지 N)내의 공통 관련성을 각각 가진 데이터 세트로서 간주될 수 있다.

따라서, 저장된 관련 데이터를 분류하는데 이용되는 속성 등을 나타내는 분류 유형 및 분류값은 도 27 및 28에 도시된 DB 서버 6-1 내지 6-n(A 내지 N)에 저장된 디렉토리 트리(또는 서브 트리)에 대해 정의될 수 있다.

도 30에 도시된 바와 같이, DB 서버 6-1 내지 6-n(A 내지 N)에 저장된 관련 데이터를 관리하기 위해, 상위 엔트리 dn_A 내지 dn_N 아래의 서브 트리 a 내지 n에 대해 정의된 분류 유형 및 분류값과, 상위 엔트리 dn_A 내지 dn_N을 포함하는 데이터 테이블이 이용되며, 상위 엔트리의 서브 트리 및 DB 서버 6-1 내지 6-n(A 내지 N)의 하위는 도 27 및 도 28에 대응되는 방식으로 도시된다.

도 30에 도시된 "분류 유형"은 관련 유형(A 노드의 노드 유형(도 22):속성)에 대응한다.

"분류값"은 LDAP 데이터 모델에서의 경우와 대응한다.

도 27 내지 30에 도시된 상위 엔트리일 수 있는 특정 예시의 엔트리들은,

- (1) 임의 분류 유형의 특정 분류값(인스턴스)을 포함하는 서브 트리의 루트 엔트리(관련 유형: A 노드의 노드 유형)와;
- (2) 임의 특정 분류값(인스턴스)을 포함하는 관련 데이터를 포함하는 서브 트리의 루트 엔트리

를 포함하지만 이에 제한되는 것은 아니다.

본 명세서에서는, "분류 유형"이 관련 유형에 대응하고, "분류값"이 그의 인스턴스에 대응하는 경우가 설명된다. 그러나, 이것은 도 30이 도 27 및 도 28에 상관계는 단지 예시적인 것일 뿐이다. "분류 유형"은 관련 유형에 대응하고, "분류값"은 그의 인스턴스에 대응하지 않을 수도 있다.

예를 들어, 도 30에 도시된 서브 트리는 "분류 유형: 카테고리 및 분류값: 계열"의 관련 데이터를 포함한다. 이것은, 예를 들어 "분류 유형"이 관련 유형에 대응하지 않고, "분류값"이 그의 인스턴스에 대응하는 않는다.

예를 들어, 도 30에 도시된 데이터 테이블의 상위 엔트리 dn_B를 포함하는 엔트리 분류 유형으로서 도시된 "위치 정보"는, 예를 들어, 분류값들중 하나로서 분류값 "(배치된) 위치"를 포함하는 (1) "임의 분류 유형의 특정 분류값(인스턴스)을 포함하는 서브 트리의 루트 엔트리(관련 유형: A 노드의 노드 유형)"의 분류 유형이다.

이 엔트리는 분류값으로서 "(배치된) 위치"를 가진 모든 관련 데이터가 DB 서버 6-2(B) 내에 저장됨을 나타낸다.

분류 유형 "위치 정보"의 분류값의 다른 예시로서, 분류값 "(옆의) 인접성" 등이 예시될 수 있다.

대안적으로, 도 30에 도시된 상위 엔트리 dn_(N-1)을 포함하는 엔트리 분류 유형으로써 기록된 "빌딩"은, 예를 들어, 분류값들 중 하나로서 분류값 "저장 1"을 포함하는, (2) "임의의 특정 분류값(인스턴스)을 포함하는 관련 데이터를 포함하는 서브 트리의 루트 엔트리"의 분류 유형이다.

이러한 엔트리는 분류값으로서 "저장 1"을 가진 모든 관련 데이터가 DB 서버 6-(n-1)(N-1)에 저장됨을 나타낸다.

관련 데이터의 작성 및 검색

이하에서는, 도 25 내지 도 30을 참조하여 상술한 DIT 형태의 디렉토리 구조에 표시된 관련 데이터의 생성 및 생성된 관련 데이터의 검색이 설명될 것이다.

제 1 DB 관리 프로그램(50)

도 31은 도 24에 도시된 DB 관리 서버(5)상에서 구동되는 제 1 DB 관리 프로그램(50)의 구조를 도시한 도면이다.

도 31을 참조하면, DB 관리 프로그램(50)은 사용자 인터페이스 유닛(500), 데이터 등록 유닛(502), 데이터 전송 유닛(504), 관련 데이터 생성 및 관리 유닛(510)(디렉토리 트리 생성 수단 및 데이터 상관 수단), 트리 정보 제공 유닛(512), 디렉토리 트리 테이블 생성 유닛(522)(상위 엔트리 상관 수단), 디렉토리 트리 테이블 관리 유닛(524), 디렉토리 트리 테이블 DB(526), 데이터 테이블 생성 유닛(532)(분류 데이터 상관 수단), 데이터 테이블 관리 유닛(534) 및 데이터 테이블 DB(536)를 포함한다.

이러한 부품에 있어서, DB 관리 프로그램(50)은 도 26에 도시된 DIT 형태의 디렉토리 구조에 표시된 관련 데이터를 생성한다.

DB 관리 프로그램(50)은 생성된 관련 데이터를 분할하여 도 27 및 도 28에 도시된 DB 서버 6-1 내지 6-n에 저장하며, 도 29 및 도 30에 도시된 디렉토리 트리 테이블과 데이터 테이블을 이용하여 관련 데이터를 관리한다.

추가적으로, DB 관리 프로그램(50)은 수집 장치(40)(도 24)가 검색하는데 필요한, 디렉토리 트리 테이블 및 데이터 테이블 내에 포함된 데이터(트리 데이터)를 적절히 제공한다.

도 32에는, 도 31에 도시된 DB 관리 프로그램(50)이 DB 관리 서버(5)의 입력/출력 장치(126)(도 7)에 디스플레이하는 DUI 이미지가 도시된다.

DB 관리 프로그램(50)에 있어서, UI 유닛(500)은 DB 관리 서버(5)나 PC(102)의 사용자를 위한 GUI 환경을 제공하고, 입력/출력 장치(126)(도 7)에 도 32에 도시된 것과 유사한 DUI 이미지를 디스플레이한다.

UI 유닛(500)은 디스플레이된 GUI 이미지에 대한 관련 데이터를 등록하거나, 관리하거나, 정정하는 사용자 동작을 수용하고, 데이터 등록 유닛(502)과 관련 데이터 생성 및 관리 유닛(510)과 같은 DB 관리 프로그램(50)의 부품으로 그 동작을 출력하며, 그의 동작을 제어한다. 또한, UI 유닛(500)은 DB 관리 프로그램(50)의 각 부품에 의해 생성된 데이터를 입력/출력 장치(126)에 디스플레이/출력한다.

UI 유닛(500)은 DB 관리 서버(5) 또는 PC(102)에 설치되고, 그 기능은 PC(102)의 사용자에게 제공됨을 알아야 한다.

관련 데이터 생성 및 관리 유닛(510)은 디렉토리 트리 테이블 생성 유닛(522) 및 데이터 테이블 생성 유닛(532) 등의 동작을 제어하고, T 및 A 노드를 나타내는 데이터 및 UI 유닛(500)으로부터 입력된 관련 속성으로부터 도 26에 도시된 바와 같은 DIT 포맷의 디렉토리 구조에 표시된 관련 데이터를 생성하고, 도 29 및 도 30에 도시된 디렉토리 트리 테이블과 데이터 테이블을 이용하여 생성된 관련 데이터를 관리한다.

요청에 응답하여, 관련 데이터 생성 및 관리 유닛(510)은 디렉토리 트리 테이블과 데이터 테이블에 포함되고, 수집 장치(40)에 의한 수집을 위해 이용되는 데이터(트리 데이터)를 트리 데이터 제공 유닛(512)에 출력한다.

또한, 사용자 동작에 따라, 관련 데이터 생성 및 관리 유닛(510)은 도 27 및 도 28에 도시된 DIT 형태의 디렉토리 구조에 표시된 관련 데이터를 분할하여 데이터 등록 유닛(502)에 관련 데이터를 출력한다.

데이터 등록 유닛(502)은 관련 데이터 생성 및 관리 유닛(510)으로부터 입력된 관련 데이터를, 데이터 송신 유닛(504)을 통해, DB 서버(6-1 내지 6-n)로 송신하고, 관련 데이터의 등록을 요청한다.

트리 정보 준비 유닛(512)은 수집 장치(40)로부터의 요청에 응답하여 관련 데이터 생성 및 관리 유닛(510)으로부터 입력된 트리 데이터를 제공한다.

디렉토리 트리 테이블 생성 유닛(522)은 관련 데이터 생성 및 관리 유닛(510)의 제어하에, 도 29에 도시된 디렉토리 트리 테이블을 생성하여, 그 테이블을 디렉토리 트리 테이블 관리 유닛(524)으로 출력한다.

디렉토리 트리 테이블 관리 유닛(524)은 디렉토리 트리 테이블 생성 유닛(522)으로부터 입력된 디렉토리 트리 테이블을 디렉토리 트리 테이블 DB(526)에 저장한다.

요청에 응답하여, 디렉토리 트리 테이블 DB(526)는 디렉토리 트리 테이블 DB(526)에 저장된 디렉토리 트리 테이블을 관련 데이터 생성 및 관리 유닛(510)으로 출력한다.

DB 관리 유닛(20)은 도 18에 도시된 DB 프로그램(2)에서의 관련 데이터로부터 AR 테이블 및 ID 테이블을 생성하는 처리를 수행하지만, 도 31에 도시된 DB 관리 프로그램(50)에서, 관련 데이터 생성 및 관리 유닛(510)은 디렉토리 구조(도 21 및 22)의 관련 데이터를 생성하고, DB 서버(6)는 관련 데이터 생성 및 관리 유닛(510) 및 데이터 송신 유닛(504)을 통해 디렉토리 구조의 생성된 관련 데이터를 수신하여, 관련 데이터를 저장/관리함을 알아야 한다.

데이터 테이블 생성 유닛(532)은 관련 데이터 생성 및 관리 유닛(510)의 제어하에, 도 30에 도시된 데이터 테이블을 생성하고, 데이터 테이블을 데이터 테이블 관리 유닛(534)으로 출력한다.

데이터 테이블 관리 유닛(534)은 데이터 테이블 생성 유닛(532)으로부터 입력된 데이터 테이블을 데이터 테이블 DB(536)에 저장한다.

또한, 요청에 응답하여, 데이터 테이블 관리 유닛(534)은 데이터 테이블 DB(536)에 저장된 데이터 테이블을 관련 데이터 생성 및 관리 유닛(510)으로 출력한다.

이하, DB 관리 프로그램(50)의 전체 처리가 기술될 것이다.

도 33은 도 31에 도시된 DB 관리 프로그램(50)을 이용한 관련 데이터의 등록 처리 S30을 도시하는 흐름도이다.

도 33을 참조하면, 단계 S300에서, DB 관리 프로그램(50) 또는 PC(102)의 UI 유닛(500)은 도 32에 도시된 GUI 이미지를 입/출력 장치(126)에 디스플레이한다.

사용자는 디스플레이된 GUI 이미지에 대해 조작을 수행하여, 새롭게 추가된 관련 데이터에 관한 데이터를 각 필드에 입력한다.

단계 S302에서, 관련 데이터 생성 및 관리 유닛(510)은 단계 S300의 처리 동안 관련 데이터 입력에 근거하여 관련성(테마; 예를 들면, '셰익스피어의 작품')을 식별한다.

단계 S304에서, 관련 데이터 생성 및 관리 유닛(510)은 관련 데이터의 노드 유형(분류 유형; 예를 들면, '저작자 정보')을 식별하고, 이 유형을 엔트리 속성으로서 설정한다.

단계 S306에서, 관련 데이터 생성 및 관리 유닛(510)은 데이터 테이블 관리 유닛(534)을 통해 데이터 테이블을 참조하여, 생성된 엔트리가 단계 S302 및 S304의 처리에서 식별된 테마 및 노드 유형에 근거하여 분류되는 서브 트리를 결정한다.

단계 S310에서, 관련 데이터 생성 및 관리 유닛(510)은 디렉토리 트리 테이블 관리 유닛(524)을 통해 디렉토리 트리 테이블을 참조하여, 단계 S306의 처리에서 획득된 서브 트리가 저장되는 DB 서버(6)를 결정한다.

단계 S312에서, 관련 데이터 생성 및 관리 유닛(510)은 UI 유닛(500)을 통해 입력된 관련 데이터에 관한 데이터에 근거하여 대응 엔트리를 생성한다.

단계 S314에서, 관련 데이터 생성 및 관리 유닛(510)은 추가된 관련 데이터 엔트리를, 데이터 등록 유닛(502) 및 데이터 송신 유닛(504)을 통해 단계 S310의 처리에서 결정된 DB 서버(6)로 송신한다. DB 서버(6)는 수신된 관련 데이터 엔트리가 저장되었는지의 여부를 판정한다.

관련 데이터 생성 및 관리 유닛(510)은 DB 서버(6)에 대한 저장시의 전처리에서 DB 서버(6) 내의 데이터를 이용함을 알아야 한다.

각 엔트리 dn(distinguished name)이 이 데이터에 할당되고, DIT내의 모든 엔트리는 dn에 의해 고유하게 식별될 수 있다. 따라서, 동일한 엔트리의 존재가 판정될 수 있다.

DB 관리 프로그램(50)은 추가된 관련 데이터 엔트리가 DB 서버(6)로부터 저장된 결과의 수신시에 처리를 종료하거나, 또는 그렇지 않은 경우 단계 S316의 처리로 진행한다.

단계 S316에서, 관련 데이터 생성 및 관리 유닛(510)은 추가된 관련 데이터 엔트리를 결정된 DB 서버(6)에 등록한다.

도 34는 도 31에 도시된 DB 관리 프로그램(50)에 의한 데이터 테이블(도 30)의 갱신 처리 S34를 도시하는 흐름도이다.

단계 S340에서, 사용자는 DB 관리 서버(5)(도 24) 또는 PC(102)를 조작하여, 변경될 서브 트리를 지정하고, 지정된 서브 트리의 분류 유형/분류 값을 설정한다.

관련 데이터 생성 및 관리 유닛(510)은 UI 유닛(500)을 통해 조작을 수신한다.

단계 S342에서, 사용자는 DB 관리 서버(5)(도 24) 또는 PC(102)를 조작하여, 지정된 서브 트리의 상위 엔트리를 설정한다.

관련 데이터 생성 및 관리 유닛(510)은 UI 유닛(500)을 통해 조작을 수신한다.

단계 S344에서, 관련 데이터 생성 및 관리 유닛(510)은 S340 및 S342에서의 처리에서 수신된 조작에 의해 지시된 분류 유형, 분류 값 및 상위 엔트리가 지정된 서브 트리에 대해 모두 설정되었는지의 여부를 판정한다.

관련 데이터 생성 및 관리 유닛(510)은 데이터 테이블 관리 유닛(534)을 통해 데이터 테이블을 관독하고, 모든 데이터가 지정된 서브 트리에 대해 설정된 경우 처리를 종료하거나, 또는 그렇지 않은 경우 S346에서의 처리로 진행한다.

단계 S346에서, 관련 데이터 생성 및 관리 유닛(510)은 수신된 설정 값(지정된 서브 트리, 분류 유형, 분류 값 및 상위 엔트리)을 데이터 테이블 관리 유닛(534)으로 출력한다.

데이터 테이블 관리 유닛(534)은 관련 데이터 생성 및 관리 유닛(510)으로부터 입력된 설정 값을 데이터 테이블에 설정하고, 그들을 데이터 테이블 DB(536)에 저장한다.

도 35는 도 31에 도시된 DB 관리 프로그램(50)에 의한 디렉토리 트리 테이블(도 29)의 갱신 처리 S36을 도시하는 흐름도이다.

도 35를 참조하면, 단계 S360에서, UI 유닛(500)은 DB 관리 서버(5)로부터의 디렉토리 정보 트리를 분할하는 사용자의 조작을 수신한다.

단계 S362에서, 관련 데이터 생성 및 관리 유닛(510)은 디렉토리 트리 테이블 관리 유닛(524)으로부터 디렉토리 트리 테이블(도 29)을 획득하고, DB 서버(6)의 구성 및 디렉토리 구조를 나타내는 데이터를 수집하고, 데이터를 사용자에게 디스플레이한다.

단계 S364에서, UI 유닛(500)은 디스플레이된 디렉토리 구조에 대한 사용자의 조작에 따라 분할 이후에 서브 트리의 상위 엔트리 및 참조 엔트리의 설정을 수신한다.

단계 S366에서, 관련 데이터 생성 및 관리 유닛(510)은 분할된 서브 트리가 디렉토리 정보 트리를 분할하는 사용자의 조작에 근거하여 저장되는 DB 서버(6)에 대한 복제 서버를 설정할지 여부를 판정한다.

DB 관리 프로그램(50)은 복제 서버가 설정되는 경우 S368에서의 처리로 진행하거나, 또는 그렇지 않은 경우 S370에서의 처리로 진행한다.

단계 S368에서, 관련 데이터 생성 및 관리 유닛(510)은 복제 서버의 설정이 S366에서의 처리에서 구성되는 경우 복제 서버를 결정한다.

단계 S370에서, 분할 이후의 서브 트리의 상위 엔트리 및 참조 엔트리 및 복제 서버가 설정되는 경우, 관련 데이터 생성 및 관리 유닛(510)은 복제 서버의 명칭을 디렉토리 트리 테이블 관리 유닛(524)으로 출력한다.

디렉토리 트리 테이블 관리 유닛(524)은 디렉토리 트리 테이블 DB(526)의 지정된 부분에 기록시에 관련 데이터 생성 및 관리 유닛(510)으로부터 입력된 데이터를 반영한다.

단계 S372에서, 관련 데이터 생성 및 관리 유닛(510)은 디렉토리 트리 테이블의 갱신 처리가 모든 DB 서버(6)에 대해 완료되었는지의 여부를 판정한다.

DB 관리 프로그램(50)은 갱신 처리가 모든 DB 서버(6)에 대해 완료되지 않은 경우에 S364에서의 처리로 리턴된다.

다음, DB 관리 서버(5)의 전체 처리가 기술될 것이다.

도 36은 도 31에 도시된 DB 관리 프로그램(50)의 전체 조작 S40을 도시하는 순차도이다.

도 36을 참조하면, 단계 S400에서, 사용자는 관련 데이터를 DB 관리 프로그램(50)의 UI 유닛(500)에 입력한다.

단계 S402에서, UI 유닛(500)은 입력된 관련 데이터에 관한 데이터를 관련 데이터 생성 및 관리 유닛(510)으로 출력한다.

관련 데이터 생성 및 관리 유닛(510)은 데이터를 수신한다.

단계 S404에서, 관련 데이터 생성 및 관리 유닛(510)은 수신된 관련 데이터에 관한 데이터 중에서 테마 및 노드 유형과 같은 관련 데이터를 분류하는데 이용된 분류 데이터(분류 유형/분류 값)를 데이터 테이블 관리 유닛(534)으로 출력한다.

단계 S406에서, 데이터 테이블 관리 유닛(534)은 관련 데이터 생성 및 관리 유닛(510)으로부터의 관련 데이터를 분류하는데 이용된 데이터를 수신하고, 대응하는 서브 트리의 상위 엔트리를 관련 데이터 생성 및 관리 유닛(510)으로 리턴한다.

관련 데이터 생성 및 관리 유닛(510)은 서브 트리의 상위 엔트리를 수신한다.

대응하는 서브 트리가 없는 경우, DB 서버(6-1)의 상위 엔트리(즉, 루트 엔트리)가, 대응하는 서브 트리의 상위 엔트리 대신에, 관련 데이터 생성 및 관리 유닛(510)으로 리턴됨을 알아야 한다.

S406에서의 처리에서, 데이터 테이블(도 30)에 저장된 분류 유형 및 분류 값에 대응하는 데이터가 이용되고, 도 32에 도시된 GUI 이미지를 통해 입력된 관련 데이터에 관한 데이터가, 관련 데이터 생성 및 관리 유닛(510)에 의해 모두 이용될 수 있다.

즉, 관련 데이터 생성 및 관리 유닛(510)은 노드 유형 및 특정 경우의 조합을 데이터 테이블 관리 유닛(534)에 공급함으로써, 데이터 테이블 관리 유닛(534)은 관련 데이터를 분류하는데 이용된 데이터에 대응하는 서브 트리의 상위 엔트리를 획득할 수 있다. 또한, 예를 들면, 분류 유형으로서의 "카테고리" 및 분류 값으로서의 "세익스피어의 작품"이 데이터 테이블 관리 유닛(534)으로 공급될 수 있다.

단계 S408에서, 관련 데이터 생성 및 관리 유닛(510)은 S406에서의 처리에서 획득된 서브 트리의 상위 엔트리를 디렉토리 트리 테이블 관리 유닛(524)으로 출력한다.

디렉토리 트리 테이블 관리 유닛(524)은 서브 트리의 상위 엔트리를 수신한다.

단계 S410에서, 디렉토리 트리 테이블 관리 유닛(524)은 수신된 서브 트리의 상위 엔트리를 이용해서 디렉토리 트리 테이블 DB(526)를 검색하여, 서브 트리를 저장하기 위한 DB 서버(6)를 결정한다. 결정된 DB 서버(6)를 나타내는 데이터는 관련 데이터 생성 및 관리 유닛(510)으로 리턴된다.

단계 S412에서, 관련 데이터 생성 및 관리 유닛(510)은 DIT 형태의 관련 데이터 및 결정된 DB 서버(6)를 나타내는 데이터를 데이터 등록 유닛(502)으로 출력한다.

데이터 등록 유닛(502)은 입력된 관련 데이터를 결정된 DB 서버(6)로 전달하여, 데이터를 등록한다.

단계 S416 내지 S422에서, 등록 결과(정상 또는 비정상 종료)가, 데이터 등록 유닛(502)으로부터 UI 유닛(500)으로 통지되고, 사용자에게 디스플레이된다.

제 2 DB 프로그램(60)

도 37은 도 24에 도시된 각각의 서버(6)상에서 실행되는 제 2 DB 프로그램(60)을 도시한다.

도 37을 참조하면, DB 프로그램(60)은 데이터 관리 유닛(600), 정보 DB(602), 데이터 송신 유닛(604) 및 검색 실행 유닛(606)을 포함한다.

이들 구성 요소를 가지고, DB 프로그램(60)은 DIT 형태의 디렉토리 구조에서 표현되고 DB 관리 서버(5)(DB 관리 프로그램(50); 도 31)로부터 적절하게 분할된 관련 데이터의 등록을 수신하고, 데이터를 저장한다.

DB 프로그램(60)은 수집 장치(40)로부터의 검색 요청에 응답하여, 등록된 관련 데이터를 제공한다.

DB 프로그램(60)은 DB 관리 서버(5)의 제어하에, 등록된 관련 데이터를 수정한다.

DB 프로그램(60)에서, 데이터 송신 유닛(604)은 DB 관리 서버(5)로부터 관련 데이터, 그것의 등록 요청 및 그것의 수정 요청을 수신하여, 그들을 데이터 관리 유닛(600)으로 출력한다.

검색 실행 유닛(606)은 수집 장치(40)로부터의 검색 조건(이후에 기술되고도 38에 도시된 검색 프로그램(42)의 검색 조건 생성 유닛(420)에 의해 생성된 LDAP 조작 및 도 50에 도시된 LDAP 조작)에 따라 데이터 관리 유닛(600)으로 관련 데이터를 검색한다.

더욱이, 검색 실행 유닛(606)은 데이터 관리 유닛(600)으로부터 획득된 관련 데이터를, 검색 요청을 전송하였던 수집 장치(40)에게, 검색 결과로서 리턴한다.

데이터 관리 유닛(600)은 정보 DB(602)에서의 DB 관리 서버(5)로부터의 등록 요청 및 수정 요청에 응답하여, DB 관리 서버(5) 또는 다른 DB 서버(6)로부터 전송된 관련 데이터(도 26 내지 28)를 저장한다.

또한, 수집 장치(40)에 의한 수집에 응답하여, 데이터 관리 유닛(600)은 정보 DB(602)로부터의 관련 데이터를 판독하여, 데이터를 검색 실행 유닛(606)으로 출력한다.

제 1 검색 프로그램(42)

도 38은 도 24에 도시된 수집 장치(40)상에서 실행되는 검색 프로그램(42)을 도시한다.

도 38을 참조하면, 검색 프로그램(42)은 검색 조건 생성 유닛(420) 및 검색 결과 출력 유닛(422)이 도 18에 도시된 DB 검색 유닛(26)에 추가된 구조를 갖는다.

이들 구성 요소를 가지고, 검색 프로그램(42)은 DB 서버(6)에 대해 검색 처리를 수행하고, 검색의 결과로서 획득된 관련 데이터를 검색자에게 디스플레이한다.

검색 프로그램(42)에서, 검색 조작 수신기(260)는 검색자가 수집 장치(40)의 입/출력 장치(126)(도 7) 등으로 입력하는 검색 조작을 수신하고, 그 조작을 검색 조건 생성 유닛(420)으로 출력한다.

검색 조건 생성 유닛(420)(디렉토리 트리 검색 수단)은 검색 조작 수신기(260)로부터 입력된 검색 조작의 내용을 분석한 후, DB 관리 서버(5)로부터 검색 조작의 내용과 매칭되는 트리 데이터(도 29 및 30의 디렉토리 트리 테이블 및 데이터 테이블에서 검색을 위해 필요한 데이터)를 획득하고, 검색 조작 및 트리 데이터와 매칭되는 검색 조건을 생성하며, 검색 조건을 검색 제어 유닛(264)으로 출력한다.

특히, 검색 조작의 내용을 분석함으로써 획득된 관련 데이터의 분류 유형 및 분류 값에 따라, 검색 조건 생성 유닛(420)은 DB 관리 서버(5)로부터 데이터 테이블(도 30)을 수신하여, 내용을 검색하고, 검색 조작의 내용과 매칭되는 관련 데이터의 상위 엔트리를 획득한다(그러나, 그러한 상위 엔트리가 존재하지 않는 경우, 검색 조건 생성 유닛(420)은 DB 서버(6-1)의 상위 엔트리 dn_A를, 검색 조작의 내용과 매칭되는 관련 데이터의 상위 엔트리로서 이용함).

그 후, 검색 조건 생성 유닛(420)은 DB 관리 서버(5)로부터 수신된 디렉토리 트리 테이블(도 29) 내의 내용을 검색하고, 데이터 테이블을 검색함으로써 획득된 상위 엔트리를 저장하는 DB 서버(6)를 결정한다.

더욱이, 검색 조건 생성 유닛(420)은 검색 조작의 내용 및 데이터 테이블로부터 획득된 관련 데이터를 분류하는데 이용된 속성(분류 유형 및 값)을 이용함으로써, DB 서버(6)에 의해 실행될 LDAP 조작의 코멘드 및 파라미터를 생성하고, 코멘드 및 파라미터를 검색 조건으로서 검색 제어 유닛(264)으로 출력한다.

검색 제어 유닛(264)(데이터 검색 수단)은 검색 조건 생성 유닛(420)으로부터 입력된 검색 조건을 DB 서버(6)로 출력하고, 관련 데이터에 대한 검색을 제어한다.

검색 결과 출력 유닛(422)은 DB 서버(6)로부터 수집 장치(40)의 입/출력 장치(126)(도 7)로, 그리고 검색자에게로, 검색의 결과로서 리턴된 관련 데이터를 디스플레이/출력한다.

이하, 검색 프로그램(42)의 전체 처리가 기술될 것이다.

도 39는 도 38에 도시된 검색 프로그램(42)에 의한 검색 처리 S44를 도시하는 흐름도이다.

도 39를 참조하면, 단계 S440에서, 검색 조작 수신기(260)는 사용자로부터 검색 조건을 수신한다.

단계 S442에서, 검색 조작 수신기(260)는 검색 요청 메시지(이후에 도 50을 참조하여 기술됨)를 생성하고, 그 메시지를 검색 조건 생성 유닛(420)으로 출력한다.

단계 S444에서, 검색 조건 생성 유닛(420)은 분류 유형 및 분류 값이 검색 요청 메시지에서 지정되는지의 여부를 판정한다.

검색 조건 작성 유닛(420)은 분류 타입 및 분류 값이 지정될 때 단계(S446)의 처리로 진행되고, 지정되지 않으면 단계(454)의 처리로 진행된다.

단계(S446)에서, 검색 조건 작성 유닛(420)은 분류 타입 및 분류 값을 DB 관리 서버(5)로 전달하고, 대응하는 서브 트리의 상위 엔트리를 결정한다.

단계(S448)에서, 검색 조건 작성 유닛(420)은 단계(S446)에서의 처리에서 서브 트리가 결정되어야 하는지 여부를 판정한다.

검색 프로그램(42)은 서브 트리가 결정될 때 단계(450)의 처리로 진행되고, 서브 트리가 결정되지 않았다면 단계(S454)의 처리로 진행된다.

단계(S450)에서, 검색 조건 작성 유닛(420)은 결정된 상위 엔트리를 DB 관리 서버(5)로 전달하고, 상위 엔트리에 대응하는 서브 트리가 저장되어야 하는 DB 서버(6)를 결정한다.

단계(S452)에서, 검색 조건 작성 유닛(420)은 서브 트리의 결정된 상위 엔트리를 검색 시작 엔트리로서 설정한다(질문 기반).

단계(S454)에서, 검색 조건 작성 유닛(420)은 DB 관리 서버(5)로부터 디렉토리 트리 테이블을 수신하고, 수신된 디렉토리 트리 테이블을 참조하여 상위 엔트리(루트 엔트리)를 모든 DB 서버에 저장된 서브 트리를 통해 질문 기반으로 설정하며, 루트 엔트리를 저장하는 DB 서버(6)를 결정한다.

단계(S456)에서, 검색 제어 유닛(264)은 검색 조건 작성 유닛(420)으로부터 검색 요청, 질문 기반 등에 관련된 데이터를 획득하고, 결정된 DB 서버(6)에 대한 LDAP 조작 커맨드를 생성한다.

단계(S458)에서, 검색 제어 유닛(264)은 생성된 LDAP 조작 커맨드를 이용함으로써 DB 서버(6)에 대한 검색 처리를 실행하고, 그 결과를 수신 및 출력한다.

이후에, 검색 프로그램(42)의 전체 처리는 DB 관리 프로그램(50)의 처리와 관련하여 추가적으로 설명될 것이다.

도 40은 도 31에 도시된 DB 관리 프로그램(50) 및 도 38에 도시된 검색 프로그램(42)의 전체 처리 단계(S52)를 도시하는 순차 도면이다.

도 40을 참조하면, 단계(S520)에서, 검색 프로그램(42)의 검색 조작 수신기(260)는 사용자의 질문 조작을 수신한다.

단계(S522)에서, 검색 조작 수신기(260)는 검색 조건 작성 유닛(420)에 대해 검색 요청 메시지를 송신한다.

단계(S526)에서, 검색 조건 작성 유닛(420)은 검색 요청 메시지에 포함된 분류 타입 및 분류 값을 DB 관리 프로그램(50)의 관련 데이터 작성 및 관리 유닛(510)에 송신한다.

단계(S526)에서의 처리는 도 39에 도시된 단계(S446)에서의 검색 조건 작성 유닛(420)으로부터 DB 관리 서버(5)로 분류 타입/값을 송신하는 처리에 대응한다.

단계(S528)에서, 관련 데이터 작성 및 관리 유닛(510)은 검색 조건 작성 유닛(420)으로부터 수신된 분류 타입/값을 데이터 테이블 관리 유닛(534)으로 출력한다.

단계(S530)에서, 데이터 테이블 관리 유닛(534)은 관련 데이터 작성 및 관리 유닛(510)으로부터의 분류 타입/값 입력을 이용하여 데이터 테이블을 참조함으로써 입력 분류 타입/값에 대응하는 상위 엔트리를 획득한다.

데이터 테이블 관리 유닛(534)은 획득된 상위 엔트리를 관련 데이터 작성 및 관리 유닛(510)으로 되돌려보낸다.

단계(S532)에서, 관련 데이터 작성 및 관리 유닛(510)은 단계(S530)에서의 처리로 획득된 상위 엔트리를 디렉토리 트리 테이블 관리 유닛(524)으로 출력한다.

단계(S534) 및 단계(S536)에서, 디렉토리 트리 테이블 관리 유닛(524)은 입력 상위 엔트리에 대응하는 DB 서버(6)를 관련 데이터 작성 및 관리 유닛(510)을 통해 검색 조건 작성 유닛(420)으로 되돌려보낸다.

단계(S538)에서, 검색 조건 작성 유닛(420)은 검색 조건을 생성하고, 결정된 DB 서버(6)의 명칭 등과 함께 검색 조건을 검색 제어 유닛(264)으로 출력한다.

단계(S540)에서, 검색 제어 유닛(264)은 LDAP 커맨드를 생성하고, 결정된 DB 서버(6)에 액세스하여 검색을 실행한다.

단계(S542) 내지 단계(S546)에서, 검색 결과는 DB 서버(6)로부터 되돌아오고, 사용자에게 디스플레이된다.

DB 시스템(4) 내에서 관련 데이터 및 검색 예(1)

이하에서, 관련 데이터가 등록되고, 등록된 관련 데이터가 DB 시스템(4) 내에서 수집되는 방법을 특정한 예를 가지고 설명할 것이다.

도 41은 도 24에 도시된 DB 시스템(4)의 DB 서버(6) 내에 등록된 디렉토리 구조의 관련 데이터 예시에 대한 제 1 도면으로서, 평면 디렉토리 구조를 나타낸다.

도 41에 도시된 디렉토리 구조에서, 도 26에서 점선에 의해 표시된 엔트리(ou= 관련성 #1 내지 #n)는 존재하지 않지만, 모든 관련 데이터는 루트 엔트리 바로 아래에 위치된다.

이러한 디렉토리 구조는 "평면 디렉토리 구조"로 지칭된다.

사용자는 DB 관리 서버(5)(도 24) 또는 PC(102)의 입/출력 장치(126)(도 7)에 디스플레이된 GUI 이미지(도 32)와 관련하여 순차적으로 등록된 관련 데이터에 대한 조작을 실행한다.

이러한 조작으로, 다음의 데이터는 이하의 형태로 DB 관리 서버(5) 내에 입력된다.

(관련 명칭, 관련 타입, 명칭, 역할):

(햄릿의 작가, 작가 정보, 셰익스피어, 저자),

(햄릿의 작가, 작가 정보, 햄릿, 작품),

(베니스의 상인의 작가, 작가 정보, 셰익스피어, 저자),

(베니스의 상인의 작가, 작가 정보, 베니스의 상인, 작품),

(햄릿의 일본어 번역문, 번역문 정보, 햄릿, 원작),

(햄릿의 일본어 번역문, 번역문 정보, 햄릿의 일본어 번역문, 번역문),

(햄릿의 일본어 번역문, 번역문 정보, 출판, ○○ 출판사),

....

사용자에 의한 등록 조작의 결과로서, 관련 데이터는 도 41에 도시된 것과 유사한 DIT 형태의 디렉토리 구조로 생성된다.

도 41에 도시된 DIT 형태의 디렉토리 구조, 즉 디렉토리 정보 트리를 서브 트리로 분할하는 것은 고려 대상이 되지 않고, 분류 타입 및 분류 값은 전체 디렉토리 정보 트리에 대해 정의될 수 있지만, 평면 디렉토리 구조는 필요한 경우에 서브 트리 구조로 구획될 수 있다는 것을 유의하라.

이러한 디렉토리 구조에서, 엔트리는 이하에서 5개의 A 노드("햄릿의 작가", "베니스의 상인의 작가", "햄릿의 일본어 번역문", "햄릿의 중국어 번역문" 및 "햄릿의 공연"으로서, A4, A9, A10, A19 및 A13(도 19 등 참조))에 대응하는 가상 루트 엔트리로 구성된다. 가상 루트 엔트리에 있어서, 분류 타입 및 분류 값이 정의되고, 정의된 분류 타입 및 값은 DB 관리 서버(5) 내에서 도 30에 도시된 데이터 테이블에 기초하여 관리된다.

추가적으로, 디렉토리 구조 내에서, A 노드의 관련성 역할에 대응하는 엔트리는 A 노드에 대응하는 엔트리로 이하에 배치되고, 각각의 역할을 수행하는 T 노드(T41, T42, T92, T101, T103 및 T191; 도 19 참조)에 관련된 데이터는 엔트리 속성으로서 설정된다.

도 41에 도시된 관련 데이터는 모두 하나의 DB 서버(6-1)(도 24) 내에 저장된다. 이와 다르게, 관련 데이터는 각각의 상위 엔트리에 기초하여 구획되고, 5개의 DB 서버(6-1 내지 6-n) 내에 저장되며, 검색 프로그램(42)에 의한 검색을 위한 타겟을 형성한다.

도 42 및 도 43은 도 41에 도시된 디렉토리 구조의 관련 데이터를 도시하는 제 2 도면 및 제 3 도면이다.

도 42는 셰익스피어의 작품인 소정의 관련 (테마)에 대해 주의를 기울이는 것에 의해 루트 엔트리 아래에 대응 엔트리 "셰익스피어의 작품"이 새로 배치될 수 있고, 관련 데이터가 그 아래에 배치되어 해당 테마에 기초한 계층화를 실현하는 경우를 도시한다. 엔트리 "셰익스피어의 작품"은 도 26의 점선에 의해 표시된 엔트리(ou = 관련성 #1 내지 #n) 중 하나에 대응한다.

도 43은 "작가", "번역문" 및 "공연"(관련성 타입; A 노드의 노드 타입)의 분류 타입을 나타내는 엔트리가 정의되고, 엔트리 "셰익스피어의 작품" 아래에 새롭게 배치되는 경우를 나타낸다.

앞서 설명된 바와 같이, 관련성 타입(노드 타입)에 따라서 엔트리 아래에 관련 데이터를 등록함으로써, 계층적 구조가 관련성 타입(노드 타입)에 기초하여 구현된다.

도 42에 도시된 바와 같이, 도 41의 관련 데이터는 더욱 계층화될 수 있다.

다시 말해서, "셰익스피어의 작품"은 도 41에 도시된 관련 데이터 내에 포함된 노드에 대한 공통 테마로서 정의될 수 있다. 따라서, 도 42에 도시된 바와 같이, 사용자는 DB 관리 서버(5)를 조작하여 엔트리 내부 및 아래에 저장된 데이터가 5개의 A 노드에 대응하는 엔트리 상의 셰익스피어의 작품에 관련된다는 것을 나타내는 새로운 엔트리를 생성하고, 관련 데이터를 계층화할 수 있다.

도 42에 도시된 경우에, 관련 데이터는 엔트리 "셰익스피어의 작품" 아래에 엔트리 "햄릿의 작가(A4)" 내지 "햄릿의 공연(A13)"을 포함하는 디렉토리 구조로 표현된다.

예를 들면, 엔트리 "셰익스피어의 작품" 및 엔트리 "햄릿의 공연(A4)" 내지 "햄릿의 중국어 번역문(A19)" 및 그 아래의 엔트리는 DB 서버(6-1) 내의 서브 트리로서 저장되고, 엔트리 "햄릿의 공연(A13)" 또는 그 아래의 엔트리는 DB 서버(6-2) 내의 서브 트리로서 저장되며, 이것에 의해서 관련 데이터는 디렉토리 구조에 따라서 서브 트리로 구획될 수 있다. 그에 따라 획득된 서브 트리는 구획될 수 있고, DB 서버(6-1 내지 6-n) 내에 저장될 수 있다.

이러한 경우에, "셰익스피어의 작품" 및 "햄릿의 공연" 디렉토리는 상위 엔트리로서 설정되고, 분류 타입 및 값은 이러한 상위 엔트리에 대해 정의되면, 정의된 분류 타입 및 값은 DB 관리 서버(5)에 의해 도 30에 도시된 데이터 테이블 내에서 관리되며, 그의 참조 관계는 도 29에 도시된 디렉토리 트리 테이블 내에서 관리된다.

다른 예로서, 도 42에 도시된 관련 데이터의 계층적 디렉토리 구조에서, "셰익스피어의 작품" 및 "햄릿의 작가"에 대응하는 관련 데이터 및 그 아래의 "베니스의 상인의 작가"는 DB 서버(6-1) 내에 저장되고, "햄릿의 일본어 번역문" 및 "햄릿의 중국어 번역문"에 대응하는 관련 데이터는 DB 서버(6-2) 내에 저장되며, "햄릿의 공연"에 대응하는 관련 데이터는 DB 서버(6-3) 내에 저장된다.

상위 엔트리는 서브 트리 내에서 루트(상위) 엔트리이고, 그에 따라 오로지 하나의 상위 엔트리만이 DB 서버(6) 내에 저장된 각각의 서브 트리에 대해 정의된다는 것을 유의하라. 분류 타입/값은 이러한 상위 엔트리에 대해 정의된다.

따라서, 도 42에 도시된 예에서, DB 서버(6-2) 내에 "햄릿의 일본어 번역문" 및 "햄릿의 중국어 번역문"에 대응하는 관련 데이터를 저장하기 위해서, 상위 엔트리(예를 들면, 도 43에 도시된 "번역문 정보"에 대응하는 엔트리)는 2개의 엔트리를 통합하기 위해 제공되고, 이것이 상위 엔트리로서 설정되는 서브 트리가 저장된다.

그에 따라 구획된 서브 트리의 참조 관계는 도 29에 도시된 디렉토리 테이블을 이용하여 관리되고, 각각의 서브 트리에 대해 정의된 분류 타입 및 분류 값은 도 30에 도시된 데이터 테이블을 이용하여 관리된다.

또한, "작가 정보"가 도 41에 도시된 5개의 A 노드 중에서 A4 노드("햄릿의 작가") 및 A9 노드("베니스의 상인의 작가")에 대한 관련성 타입(노드 타입)으로서 정의될 때, "번역문 정보"는 A10 노드("햄릿의 일본어 번역문") 및 A19 노드("베니스의 상인의 중국어 번역문")에 대한 관련성 타입(노드 타입)으로서 정의되며, "공연 정보"는 A13 노드("햄릿의 공연")에 대한 관련성 타입(노드 타입)으로서 정의되며, 정의된 관련성 타입(노드 타입)은 도 43에 도시된 바와 같이 계층화될 수 있다.

예를 들면, 도 43에 도시된 관련 데이터의 계층적 구조에서, "셰익스피어의 작품"에 대응하는 관련 데이터 및 그 아래의 "작가 정보"는 DB 서버(6-1) 내에 저장되고, "번역문 정보"에 대응하는 관련 데이터는 DB 서버(6-2) 내에 저장되며, "공연 정보"에 대응하는 관련 데이터는 DB 서버(6-3) 내에 저장된다.

이러한 경우에, "셰익스피어의 작품", "번역문 정보" 및 "공연 정보"의 엔트리는 상위 엔트리로서 설정되고, 분류 타입 및 값은 상위 엔트리에 대해 정의되며, 정의된 분류 타입 및 값은 DB 관리 서버(5) 내에서 도 30에 도시된 데이터 테이블 내에서 관리되며, 그의 참조 관계는 도 29에 도시된 디렉토리 트리 테이블 내에서 관리된다.

검색자(사용자)가 검색 프로그램(42)(도 24)을 실행하여 도 41 내지 도 43에 도시된 바와 같이 DB 서버(6) 내에 저장된 관련 데이터를 검색할 때, 검색 프로그램(42)은 검색 조건의 콘텐츠를 분석하고, 트리 정보(디렉토리 트리 테이블) 및 DB 관리 서버(5)로부터의 검색 조건의 콘텐츠와 일치하는 데이터 테이블(도 29 및 도 30)의 필요한 정보를 획득한다.

또한, 검색 프로그램(42)은 검색 조건 및 트리 정보와 일치하는 검색 조건(LDAP 조장을 위한 커맨드 및 파라미터)을 생성하고, 검색 조건에 일치하는 디렉토리 정보를 저장하는 DB 서버(6) 내에 검색을 실행하여 검색 결과를 획득한다.

그에 따라 획득된 검색 결과는 검색 프로그램(42)의 사용자에게 디스플레이된다.

DB 시스템(4) 내에서의 관련 데이터 및 검색 예(2)

이하에서, 관련 데이터를 등록하고, 등록된 데이터가 DB 시스템(4) 내에서 수집되는 방법을, 상품의 판매를 촉진하기 위해 상품, 상점 및 프로그램을 표시하는 판매 정보의 특정한 예를 가지고 보다 상세하게 설명될 것이다.

도 44 및 도 45는 도 24에 도시된 DB 시스템(4)의 DB 서버(6) 내에 등록된 디렉토리 구조의 관련 데이터를 도시하는 제 4 및 제 5 도면이다. 도 44는 예를 들면, 간토 지방 등과 같은 특정한 지방의 판매 정보만을 나타내고, 도 45는 예를 들면, 간토 및 간사이 지방 등의 여러 영역의 판매 정보를 도시한다.

사용자가 도 24에 도시된 DB 관리 서버(5) 또는 PC(102)를 조작함으로써, 간토 지방의 판매 정보는 도 44에 도시된 디렉토리 구조 내에 표시된 관련 데이터로서 등록된다.

다시 말해, 사용자에게 의한 관련 데이터 입력의 등록된 데이터에 기초하여, "판매 정보"와 관련된 디렉토리 정보를 통합하기 위한 엔트리는 가상 루트 엔트리 바로 아래에 배치된다. 이러한 엔트리 아래에는, "간토 지방"의 판매 정보에 관한 엔트리가 배치된다.

"간토 지방"의 판매 정보와 관련하여, 관련성 타입(노드 타입)으로서 "위치 정보", "신청 정보" 및 "참여 정보"를 갖는 엔트리가 배치된다. 이러한 엔트리 아래에는, "상점", "영역", "신청" 및 "주최자(Organizer)" 등과 같은 관련성 역할에 대응하는 엔트리가 배치된다. 이러한 엔트리의 속성(속성은 DB 서버(6) 내에 저장되지만, 독립 관리 테이블은 속성을 관리하기 위해 정의되거나 생성되지 않음)과 같이, "상점 A"의 역할을 하는 멤버, "신주쿠" 등과 관련된 데이터(T 노드 ID 테이블(도 21)의 노드 ID 및 노드 명칭에 대응함)가 등록된다.

도 44 등에 도시된 "역할"은 도 20에 도시된 AR 테이블의 관련성 역할에 대응하고, "멤버"는 T 노드에 대응한다는 것을 유의하라.

도 20에 도시된 예에서, 관련성 노드(A 노드)(A4)에 대해, 멤버 T41은 "저자"의 역할을 수행하고, 멤버 T42는 "작품"의 역할을 한다.

도 45에 도시된 바와 같이, 판매 정보가 간토 지방의 판매 정보에 추가하여 정보(간사이 지방의 판매 정보 등)를 더 포함할 때, "판매 정보"의 엔트리는 가상 루트 엔트리의 바로 아래에 배치된다. 이러한 엔트리 아래에, "간토 지방" 및 "간사이 지방"의 엔트리가 배치된다. "간사이 지방"의 엔트리 아래에는, "위치 정보", "신청 정보" 및 "참여 정보"의 관련성 타입(노드 타입)을 나타내는 엔트리가 배치된다.

이러한 엔트리 아래에는, "상점", "신청", "영역", "주최자" 등의 역할에 대응하는 엔트리가 배치된다. 엔트리의 속성과 마찬가지로, "상점 B", "상점 C", "오사카", "캠페인 A" 등의 역할을 수행하는 멤버에 관한 데이터가 등록된다.

예를 들면, 도 44 및 도 45에 도시된 판매 정보는 서브 트리로 구획된다. 예를 들면, 가상 루트 엔트리 및 그 아래의 "판매 정보"를 제외한 디렉토리 정보는 DB 서버(6-1)(A; 도 24) 내에 저장되고, "간토 지방" 및 그 아래의 "판매 정보"의 디렉토리 정보는 DB 서버(6-2)(B)에 저장되며, "간토 지방" 및 그 아래의 디렉토리 정보는 DB 서버(6-3)(C)에 저장되고, 이러한 지역 이외의 지역의 디렉토리(도시하지 않음) 정보는 DB 서버(6-4)(D)에 저장된다.

도 46은 도 44 및 도 45에 도시된 판매 정보(관련 데이터)의 상위 엔트리 및 참조 엔트리를 나타내는 디렉토리 트리 테이블을 도시한다.

도 47은 도 44 및 도 45에 도시된 판매 정보(관련 데이터)의 상위 엔트리의 분류 타입 및 값을 나타내는 데이터 테이블을 도시한다.

DB 서버(6-1 내지 6-3)에 저장된 판매 정보의 상위 엔트리 및 참조 엔트리, 구획된 디렉토리 트리(또는 서브 트리)의 상위 엔트리 및 서브 트리 내에 저장된 디렉토리 정보의 분류 타입/값은, 도 46 및 도 47에 도시된 디렉토리 트리 테이블 및 데이터 테이블에 기초하여 DB 관리 서버(5)에 의해 관리된다.

도 48은 도 44 및 도 45에 도시된 판매 정보에 사용된 GUI 이미지를 도시한 것이다.

도 48에 도시된 GUI 이미지는 DB 관리 서버(5)의 입력/출력 장치(126) 또는 PC(102)에 디스플레이된다. GUI 이미지에 대한 사용자의 조작에 따라서, DB 서버 6-1 내지 6-3에 등록된 판매 정보가 검색된다.

간토 지역만의 판매 정보에 대한 검색

우선, 도 44에 도시된 간토 지방만의 판매 정보에 대한 검색을 설명한다.

그러나, 도 44는 간토 지방만의 판매 정보를 도시하고 있지만, 실제 디렉토리 정보로서 다른 지역의 정보 및 판매 정보 외의 정보가 모두 등록/저장된다.

검색자(사용자)는 검색 장치(40)의 입력/출력 장치(126)에 디스플레이된 GUI 이미지(도 48 참조) 상에서, 예를 들어 "간토 지방 내의 어떠한 판매 프로그램에 참가하는 것을 통해 상점 B와 제휴한 상점"의 검색 조작을 수행한다.

도 49는 도 24에 도시된 수집 장치(40)가 도 46 및 도 47에 도시된 디렉토리 트리 테이블 및 데이터 테이블로부터 타겟 DB 서버(6)를 검색하는 프로세스를 도시한 것이다.

예를 들면, 도 49에 도시된 바와 같이, 수집 장치(40)는 검색자의 검색 요구의 내용을 분석하고, DB 관리 서버(5)로부터 획득한 트리 정보에 포함된 데이터 테이블(도 47 참조)의 내용을 참조하여 "지역" 및 "간토 지방"이 분류 타입 및 분류 값으로서 상관되는 서브 트리("간토 지방"의 디렉토리)의 상위 엔트리(dn_B)를 획득한다.

또한, 수집 장치(40)는 트리 정보에 포함된 디렉토리 트리 테이블(도 46 참조)의 내용을 검색하고, 상위 엔트리(dn_B)가 저장되어 있는 DB 서버 6-2(B)를 검색한다.

도 50은 도 48에 도시된 검색 조건으로부터 획득되어 도 44에 도시된 간토 지방만의 판매 정보에 대한 검색에 사용된 LDAP를 생성하기 위한 검색 요구 메시지를 도시한 것이다.

도 50은 XML 메시지로서 사용자에게 의한 검색 요구 입력을 나타내며, 실제 검색에 사용된 LDAP 조작 커맨드는 이 검색 요구로부터 생성되지만, XML 외의 다른 포맷이 본 발명의 범위로부터 제외되지는 않는다는 점에 유의하라.

"간토 지방 내의 어떠한 판매 프로그램에 참가하는 것을 통해 상점 B와 제휴한 상점"에 대한 검색을 위한 LDAP 조작(LDAP 커맨드 및 파라미터)은 조작자에 의한 검색 조건 입력, 트리 정보로부터 획득된 서브 트리의 엔트리와, DB 서버 6-2(B) 내의 저장을 나타내는 정보로부터 발생되며, DB 서버 6-2(B)에서 검색이 수행된다.

도 50은 수집 장치(40)가 간토 지방의 디렉토리 및 그 디렉토리 아래의 엔트리를 저장하는 DB 서버 6-2(B)(도 24 참조)에서 다음의 검색 조작(1) 및 (2)를 수행하는 것을 나타내는 LDAP 조작을 생성하는 검색 요구 메시지를 나타낸다.

(1) 하위 엔트리 "역할(Role)"이 "참가자(Participant)"이고, 하위 엔트리 "멤버(Member)"가 "상점 B(Store B)"인 엔트리는 "관련 타입(Association type)"(도 22에서 도시된 A 노드의 노드 타입; 관련 타입)이 "참가 정보(Participation Information)"인 모든 엔트리로부터 획득되고, 멤버 속성 값(멤버 명)이 "역할"이 "조직자(Organizer)"인 하위 엔트리의 속성 값으로부터 획득된다.

이 프로세싱의 결과로서, 상점 B가 참가하는 판매 프로그램으로서 "프로그램 A"가 획득된다.

(2) 하위 엔트리 "역할"이 "조직자"이고 하위 엔트리 "멤버"가 "프로그램 A"인 엔트리는 "관련 타입"이 "참가 정보"인 모든 엔트리로부터 획득되고, 멤버 속성 값(멤버 명)은 "역할"이 "참가자"인 하위 엔트리의 속성 값으로부터 획득된다.

이 프로세싱의 결과로서, "상점 A" 및 "상점 B"는 판매 프로그램 "프로그램 A"에 참가하는 상점으로서 획득된다. 수집 장치(40)는 최종 검색 결과로서 검색 조작의 내용(도 48 참조)에 포함된 "상점 B"와 다른 "상점 A"를 선택한다.

특정 지방에서의 판매 정보에 대한 검색

그 다음에, 간토 및 도 45에 도시된 다른 지역의 판매 정보에 대한 검색을 설명한다.

검색자(사용자)는 검색 장치(40)의 입력/출력 장치(126)에 디스플레이된 GUI 이미지(도 48 참조) 상에서, 예를 들어 "상점 A가 지역에 관계없이(모든 지역에서) 상품을 제공함으로써 참가하는 캠페인에 참가하는 상점"의 검색 조작을 수행한다.

검색 장치(40)는 검색자의 검색 조작의 내용을 분석하고, DB 관리 서버(5)로부터 획득한 트리 정보에 포함된 데이터 테이블(도 47 참조)을 검색한다. 그러나, 검색 조작의 내용이 지역을 한정하지 않기 때문에, 수집 장치(40)는 "간토 지방", "간사이 지방" 등의 지역 엔트리에 의해 참조된 "판매 정보"(예를 들어 도 46의 dn_A 참조)의 엔트리를 수집한다.

또한, 수집 장치(40)는 디렉토리 트리 테이블(도 46 참조)을 검색하고, 상위 엔트리(dn_A)를 저장하고 있는 DB 서버 6-1(A)를 검색한다.

"지역과 관계없이(모든 지역에서) 상품을 제공함으로써 상점 A가 참가하는 캠페인에 참가하는 상점"에 대한 검색을 위한 LDAP 조작(LDAP 커맨드 및 파라미터)은 조작자에 의한 검색 조건 입력, 트리 정보로부터 획득된 상위 엔트리 및 DB 서버 6-1(A) 내의 저장을 나타내는 정보로부터 발생되며, DB 서버(6)에서 검색이 수행된다.

LDAP 조작은 다음의 검색 조작을 포함한다.

(1) 하위 엔트리 "멤버(Member)"가 "상점 A(Store A)"이고, 하위 엔트리 "역할(Role)"이 "피제공자(Offeree)"인 엔트리는 "관련 타입(Association type)"(도 22에서 도시된 A 노드의 노드 타입; 관련 타입)이 "제공 정보(Offer Information)"인 모든 엔트리로부터 획득되고, 멤버 속성 값(멤버 명)이 "역할"이 "제공(Offering)"인 하위 엔트리의 속성 값으로부터 획득된다.

이 프로세싱의 결과로서, 상점 A가 참가하는 상품으로서 "상품 A"가 획득된다.

(2) 하위 엔트리 "멤버"가 "상품 A"이고, 하위 엔트리 "역할"이 "참가자"인 엔트리는 "관련 타입"이 "참가 정보"인 모든 엔트리로부터 획득되고, 멤버 속성 값(멤버 명)은 "역할"이 "조직자"인 하위 엔트리의 속성 값으로부터 획득된다.

이 프로세싱의 결과로서, 상품 A가 제공되는 캠페인 프로그램으로서 "캠페인 A"가 획득된다.

(3) 하위 엔트리 "멤버"가 "캠페인 A"이고 "역할"이 조직자"인 엔트리는 "관련 타입"이 "참가 정보"인 모든 엔트리로부터 획득되고, 멤버 속성 값(멤버 명)은 "역할"이 "참가자"인 하위 엔트리의 속성 값으로부터 획득된다.

이 프로세싱의 결과로서, 수집 장치(40)는 캠페인 A에 참가하는 상점에 의해 제공된 상품으로서 "상품 A" 및 "상품 C"를 수집한다.

(4) 하위 엔트리 "멤버"가 상품 A"이고 하위 엔트리 "역할"이 "제공"이거나 또는 하위 엔트리 "멤버"가 "상품 C"이고 하위 엔트리 "역할"이 "제공"인 엔트리는 "관련 타입"이 "제공 정보"인 모든 엔트리로부터 획득되고, 멤버 속성 값(멤버 명)은 "역할"이 "피제공자"인 하위 엔트리의 속성 값으로부터 획득된다.

이 프로세싱의 결과로서, "상점 A" 및 "상점 C"가 상품 A 및 C를 캠페인 A에 제공하는 상점으로서 획득되며, 수집 장치(40)는 "상점 A" 및 "상점 C"로부터, 최종 검색 결과로서 검색 조작(도 48 참조)의 내용에 포함된 "상점 A"와 다른 "상점 C"를 선택한다.

DB 서버들 사이의 디렉토리 구조 및 부하 분산의 동적 수정

전술한 바와 같이, DB 시스템(4)(도 24 참조)에서, 관련 데이터는 디렉토리 구조 내에 표시된다. 따라서, DB 서버들(6) 사이의 서브 트리에 기초한 분할 및 복제가 이루어지기 쉽다.

따라서, DB 시스템(4)에서, 예를 들면, DB 서버(6)에 저장된 임의의 엔트리가 이런 저런 이유로 과도해지면, 그 엔트리를 저장하기 위한 복제 서버(미러 서버)를 추가하고 엔트리 및 그 하위 엔트리(이하에서는 엔트리 세트 및 그 하위 엔트리를 서브 트리라고 하기도 한다)를 함께 다른 DB 서버(6)로 전송하는 것과 같은 부하 분산이 쉽게 이루어진다.

이하에서는, DB 시스템(4)의 특성을 이용하여 DB 서버(6) 사이의 부하 분산 및 DB 시스템(4)의 시스템 성능 저하 방지에 적합한 디렉토리 구조 및 서브 트리 교환/이동의 동적 수정을 설명한다.

디렉토리 구조의 동적 수정 및 DB 시스템(4) 내의 DB 서버(6) 사이의 서브 트리 교환/이동을 위해 다음 방법을 이용한다.

(1) 동일한 DB 서버(6) 내에서의 새로운 서브 트리의 생성(새로운 서브 트리의 제공)

(2) 높은 부하 DB 서버(6)로부터 낮은 부하 DB 서버(6)로의 전송 또는 높은 부하 DB 서버(6)의 빈번하게 액세스된 서브 트리와 낮은 부하 DB 서버(6)의 드물게 액세스된 서브 트리 사이의 교환(서브 트리 교환/이동).

새로운 서브 트리의 제공

우선, DB 시스템(4) 내에 새로운 서브 트리를 제공하는 것을 설명한다.

도 51은 도 24에 도시된 DB 시스템(4)의 DB 서버(6) 내에 새로운 서브 트리를 제공하는 것을 도시한 것이므로, (A)는 새로운 서브 트리를 제공하기 전의 디렉토리 구조를 나타내고, (B)는 새로운 서브 트리 제공 후의 디렉토리 구조를 나타낸다.

도 51(a)에 도시된 바와 같이, 수집 장치(40)가 질문에 기초하여 queryBaseDN을 설정하고 필터(queryFilter)를 사용하여 검색을 수행하는 경우, 검색 조건과 일치하는 엔트리가 DB 서버(6)에 저장된 디렉토리 트리(또는 서브 트리)의 일부에 집중되는 경우를 고려한다.

필터(queryFilter)는 LDAP의 검색 조건에 대응하고, "역할"="제공"이고, "멤버"="상품 C" 등의 조건이 지금까지 설명한 검색 조작 내의 필터(queryFilter)를 구성한다는 점에 주의하라.

필터(queryFilter)에서, 복잡한 조건이 일부 조건의 논리 곱으로 표현될 수 있다.

DB 서버(6)는 각 엔트리의 검색 조건과의 매칭 횟수를 계수하고, 그 계수의 결과는 DB 관리 서버(5)에서 합계되며, 이것에 의해 빈번하게 액세스된 엔트리 세트 및 그것의 상위 엔트리가 특정 검색 조건 하에서 획득될 수 있다.

예를 들면, 새로운 서브 트리의 제공 전의 디렉토리 구조(도 51(a) 참조)에서, 이러한 측정을 수행함으로써, 도면에서 빗금친 부분의 엔트리가 검색 조건(질문 베이스: queryBaseDN, 필터: queryFilter)과 일치하고 액세스가 어떠한 검색 조건 하에서 이들 엔트리에 집중되는 것을 검출할 수 있다.

이러한 액세스 집중이 검출되면, 상위 엔트리(groupDN)는 액세스 집중 엔트리 세트에 제공되고, 이들 엔트리는 새로운 서브 트리로서 구축된다. 따라서, 엔트리에 대한 액세스가 국부화될 수 있으며, DB 시스템의 성능의 저하가 방지될 수 있다.

또한, 하나의 DB 서버(6)와 다른 DB 서버 사이에 액세스 집중 서브 트리를 이동하기 위한 준비가 이루어질 수 있다.

도 52는 도 51(a) 및 51(b)에 도시된 새로운 서브 트리의 제공을 위해 사용된 검색 조건 관리 테이블을 도시한 것이다.

상위 엔트리(groupDN)를 제공함으로써 서브 트리가 새롭게 생성되는 경우, DB 관리 서버(5)(도 24 참조)는 검색 개시 엔트리(질문 베이스; queryBaseDN), 검색에 사용된 필터(queryFilter), 검색 개시 엔트리 아래의 엔트리(새로운 질문 베이스; groupDN) 및 상기 액세스 집중 엔트리 위의 셋업이 상관되는 도 52에 도시된 검색 조건 관리 테이블을 생성하여, 이 테이블을 수집 장치(40)로 공급한다.

검색 조건 관리 테이블을 수신하여 검색 조건을 생성하는 경우에, 질문 필터 내의 논리 상품으로서 전술한 특정 필터(queryFilter)를 포함하는 검색이 변경된 엔트리(질문 베이스: queryBaseDN) 아래의 서브 트리에서 발생할 때, 수집 장치(40)는 원래의 검색 개시 엔트리(질문 베이스; queryBaseDN) 아래의 서브 트리에 대해서가 아니라, 새로 생성된 검색 개시 엔트리(새로운 질문 베이스; groupDN) 아래의 서브 트리에 대해서 특정 필터(queryFilter)를 제외한 필터를 사용함으로써 검색을 수행한다. 따라서, 검색에 필요한 시간을 단축시키는 것이 가능하다.

필터가 특정 필터(queryFilter)를 포함하지 않는 경우, 원래의 엔트리(queryBaseDN)는 검색 개시 엔트리로서 설정되고, 모든 하위 서브 트리의 범위에서 검색이 이루어진다.

새로운 서브 트리의 제공 후에, 검색 조건 관리 테이블을 사용하지 않는 경우에, 검색을 위한 상위 엔트리는 데이터 테이블(도 47 참조) 또는 검색 조건 관리 테이블로부터 획득될 수 없을 때, 수집 장치(40)는 DB 서버 6-1 내지 6-n(A 내지 N) 내의 상위 디렉토리로부터 검색을 수행한다.

서브 트리의 이동/교환

도 53(a) 및 53(b)는 도 24에 도시된 DB 시스템(4)의 DB 서버 6-1 및 6-2(A 및 B) 사이의 서브 트리 교환/이동을 예시하는데, 도 53(b)는 서브 트리 교환 전의 디렉토리 구조를 나타내고, 도 53(b)는 서브 트리 교환 후의 디렉토리 구조를 나타낸다.

도 53(a)에 도시된 바와 같이, 액세스가 DB 서버 6-2(B)에 집중되는 서브 트리(dn_3 또는 그보다 낮은 엔트리의 서브 디렉토리)가 존재하고, DB 서버 6-1(A)에 거의 아무런 액세스도 없는 서브 트리(dn_2 또는 그보다 낮은 엔트리의 서브 트리)가 존재하는 경우를 고려한다.

각 엔트리에 대한 액세스 횟수는 DB 서버(6)에 의해 계수되고, 그 계수의 결과는 DB 관리 서버(5)에 의해 합계되며, 이에 따라 빈번하게 액세스된 서브 트리가 획득될 수 있다. 동시에, DB 서버(6)의 로딩 상태가 획득될 수 있다. 또한, 도 52에 도시된 검색 조건 관리 테이블을 사용함으로써 액세스가 집중되는 서브 트리가 획득될 수 있다.

이러한 액세스 집중(비균일성)이 검출되면, 도 53(B)에 도시된 바와 같이, 고 부하 DB 서버 6-2(B)의 dn_3 또는 이보다 낮은 엔트리의 서브 트리가 DB 서버 6-1(A)의 dn_A-1 또는 이보다 낮은 엔트리의 서브 트리로서 구성되는 반면에, DB 서버 6-1(A)의 엔트리 dn_2의 서브 트리는 DB 서버 6-2(B)의 dn_2가 상위 엔트리인 서브 트리로서 구성되며, 이들 서브 트리는 DB 서버 6-1과 6-2(A 및 B) 사이에서 교환된다.

그러나, 매크로 레벨에서 액세스 집중/비균일성을 고려하면(예를 들어, 검색되는 횟수가 적어서 액세스 및 부하가 낮은 경우), 서브 트리 이동/교환에 대해 검색 조건 관리 테이블을 사용하는 것이 항상 필요치는 않다.

새로운 서브 트리(검색 조건 관리 테이블)의 제공은 소정의 시점까지 일어난 특정 검색 조건 하에서의 액세스 집중 정도에 따라서 그 후에 고속의(고 효율의) 검색 처리를 달성하도록 설계된다.

반면에, 서브 트리 이동/교환은 검색 조건에 상관없이 또는 특정 검색 조건에 한정되지 않고 각 서버 또는 서브 트리에 대한 절대 액세스 횟수(빈도)의 증가/감소로 인해 상당히 불균일한 부하의 분산이 발생하는 문제를 해결한다.

도 54는 도 53(a) 및 53(b)에 도시된 서브 트리 교환으로 인한 DB 서버 6-1 및 6-2(A 및 B)에 대한 디렉토리 트리 테이블(도 46 참조)의 업데이트를 도시한 것으로, (A)는 업데이트 전의 디렉토리를 나타내고, (B)는 업데이트 후의 디렉토리를 나타낸다.

주목할 것은 상기 서브 트리 교환(subtree exchange)이 도 51(a) 및 도 51(b)에 도시된 바와 같이 신규로 제공된 서브 트리를 및 기존의 서브 트리들에 대해서 수행된다는 것이다.

업데이트가 수행될 때, DB 관리 서버(5)는 도 54(a) 및 도 54(b)에 도시된 바와 같은 DB 서버(6-1) 및 (6-2)(A 및 B)에 대한 디렉토리 트리 테이블의 내용을 업데이트한다.(그러나, 이 경우, DB 서버(6-1(A))에 대한 디렉토리 트리 테이블의 내용에는 변화가 없다.)

가령, 전체의 DB 서버(6-2(B))에 대한 액세스는 빈번하지 않고 액세스가 도 53(a)에 도시된 DB 서버(6-1(A))의 디렉토리 dn_1에 집중될 때, 저저항 상태를 설정함으로써 부하는 단지 DB 서버(6-1(A))로부터 DB 서버(6-2(B))로 디렉토리 dn_1 이하의 서브 트리를 이동시킴으로써 DB 서버(6) 중에 분포된다.

전술한 서브 트리 이동/교환은 아래의 과정에 따라 수행된다.

- (1) 액세스 주파수 측정: DB 서버(6)는 가상 루트 엔트리 바로 아래의 엔트리로부터 연관 데이터의 기본 컴포넌트(도 26)의 제 1 레벨의 하나의 계층만큼 상위 위치의 엔트리로 엔트리에 대한 액세스의 회수(주파수)를 측정하며, DB 관리 서버(5)는 측정된 값들을 합계한다.
- (2) 이동/교환 소스의 서브 트리의 선택: 동일한 레벨에 속하는 다수의 엔트리들 아래의 서브 트리에 대한 상당히 불균일한 액세스가 각각의 DB 서버(6)에서 검출될 때, DB 관리 서버(5)는 이동/교환 소스의 서브 트리로서 엔트리 아래의 서브 트리를 선택한다.
- (3) 이동/교환 목적지의 DB 서버의 선택: 이동/교환 소스의 서브 트리의 선택 시, DB 관리 서버(5)는, 이동/교환 소스의 서브 트리가 하위 엔트리로서 저장되지 않고 그 액세스 주파수(부하)가 대체로 이동/교환 소스의 서브 트리를 저장한 DB 서버(6)의 액세스 주파수보다는 낮은 다른 DB 서버를 선택한다.

제 3 DB 프로그램(62)

도 55는 도 24에 도시된 DB 시스템(4)에서 서브 트리가 신규로 생성되거나 이동/교환될 때 제 3 DB 프로그램(62)이 DB 서버(6) 상에서 실행되는 것을 도시한다.

도 55를 참조하면, DB 프로그램(62)은 데이터 송신 유닛(620) 및 액세스 모니터링 유닛(624)이 도 37에 도시된 제 2 DB 프로그램(60)에 추가되는 구성을 사용하고 있다.

이러한 컴포넌트에 대해, DB 프로그램(62)은 제 2 DB 프로그램(60)의 것과 유사한 기능에 추가하여 새로운 서브 트리 및 이동/교환의 제공에 필요한 기능을 구현한다.

DB 프로그램(62)에서, 데이터 송신 유닛(620)은 DB 관리 서버(5)로부터의 제어에 따라 도 53(a) 및 도 53(b)에 도시된 서브 트리 이동/교환에 필요한 연관 데이터를 송신한다.

액세스 모니터링 유닛(624)은 DB 서버(6)에 저장된 디렉토리 정보의 각각의 엔트리에 대한 액세스의 회수(주파수)를 측정하여 측정된 값을 DB 관리 서버(5)에 송신한다.

제 2 DB 관리 프로그램(56)

도 56은 도 24에 도시된 DB 시스템(4)에서 서브 트리가 신규로 구성되고 이동/교환될 때 제 2 DB 관리 프로그램(56)이 DB 관리 서버(5) 상에서 실행되는 것을 도시한다.

도 56을 참조하면, 제 2 DB 프로그램(56)은 재구성 처리 유닛(560), 데이터 전송 제어 유닛(562)(데이터 전송 수단), DB 모니터링 유닛(570), 모니터링 DB(572), 검색 조건 관리 유닛(580), 검색 조건 DB(582) 및 검색 조건 제공 유닛(584)이 도 31에 도시된 제 1 DB 관리 프로그램(50)에 추가되는 아키텍처를 사용한다.

이러한 컴포넌트에 대해, DB 관리 프로그램(56)은 제 1 DB 관리 서버(50)의 것과 유사한 기능에 부가적으로 새로운 서브 트리 및 이동/교환의 제공을 위해 필요한 기능들을 구현한다.

DB 관리 프로그램(56)에서, DB 모니터링 유닛(570)은 DB 서버(6)로부터 전송된 엔트리에 대한 액세스의 회수(주파수)와 상기 액세스를 위해 사용된 LDAP의 검색 조작에 관한 데이터(질문 베이스, 및 필터)를 합산하여 모니터링 DB(572) 내에 상기 합산 값을 저장한다.

DB 모니터링 유닛(570)은 액세스 주파수의 저장된 합산 값과 검색 조작에 관한 데이터를 상기 재구성 처리 유닛(560)으로 출력한다.

상기 재구성 처리 유닛(560)은 DB 모니터링 유닛(570)으로부터 입력되는 엔트리에 대한 액세스의 회수의 합산 값에 따라 데이터 전송 제어 유닛(562)을 제어하며, 도 51(a) 및 도 51(b)에 도시된 새로운 서브 트리의 제공을 위한 처리를 수행한다.

재구성 처리 유닛(560)은 엔트리에 대한 액세스의 회수의 합산값에 따라 데이터 전송 제어 유닛(562)을 제어하며 도 53(a) 및 도 53(b)에 도시된 서브 트리의 이동/교환에 필요한 처리를 수행한다.

또한, 재구성 처리 유닛(560)은 데이터 테이블 관리 유닛(534)을 제어하여, 도 54(a) 및 도 54(b)에 도시된 신규 서브 트리 및 이동/교환을 제공함으로써 야기되는 디렉토리 트리 테이블의 변형(도 46)에 필요한 처리를 수행한다.

검색 조건 관리 유닛(580)은 서브 트리가 신규로 작성되거나 이동/변경될 때 도 52에 도시된 검색 조건 관리 테이블을 작성하여, 검색 조건 DB(582)내에 그 테이블을 저장한다.

또한, 검색 조건 관리 유닛(580)은 상기 저장된 검색 조건 관리 테이블을 검색 조건 제공 유닛(584)에 출력한다.

검색 조건 제공 유닛(584)은 검색 조건 관리 유닛(580)으로부터 입력되는 검색 조건 관리 테이블을 수집 장치(40)(검색 프로그램(44)(도 59)에 제공한다.

이후, 액세스 회수의 카운팅은 보다 상세히 기술될 것이다.

도 57은 도 56에 도시된 DB 관리 프로그램(56)의 처리 S56을 도시한 플로우차트이다.

도 57을 참조하면, 단계 S560에서, DB 관리 프로그램(50)은 초기화를 수행한다.

단계 S562에서, DB 모니터링 유닛(570)은 DB 서버(6)의 각각의 엔트리에 대한 액세스의 회수의 측정을 시작한다.

단계 S564에서, DB 서버(6)는 검색 제어 유닛(264)으로부터 LDAP 검색 조작 커맨드를 수신한다.

단계 S566에서, DB 모니터링 유닛(570)은 DB 서버(6)에 입력되는 검색 조작 커맨드에 포함되는 검색 조건을 수집한다.

단계 S568에서, DB 모니터링 유닛(570)은 DB 서버(6)로부터 검색 조건과 매칭하는 엔트리의 데이터를 수집한다.

주목할 것은 단계 S564 내지 S568의 처리가 DB 서버(6)에 의해 수행되는 검색 조작에 관한 데이터를 합산하도록 수행되며, 합산된 값이 DB 서버(6)로부터 DB 모니터링 유닛(570)으로 전송된다는 것이다.

단계 S570에서, DB 모니터링 유닛(570)은 검색 조작에 관한 데이터를 합산하기 위한 사전 결정된 측정 시간이 경과했는지의 여부를 판정한다.

DB 관리 프로그램(56)은 측정 시간이 경과했다면 단계 S572의 처리로 진행하거나 그렇지 않으면 단계 S564의 처리로 진행한다.

단계 S572에서, DB 모니터링 유닛(570)은 액세스 회수의 측정을 종료한다.

단계 S574에서, DB 모니터링 유닛(570)은 모든 엔트리에 대한 액세스 회수와 측정의 시작에서 종료까지의 측정 시간 동안 모든 DB 서버(6)의 로딩 상태를 계산한다.

단계 S576에서, 재구성 처리 유닛(560)은 DB 모니터링 유닛(570)에 의해 계산되는 측정 결과에 기반하여 모든 엔트리에 대한 액세스 집중의 존재를 검출한다.

단계 S578에서, DB 모니터링 유닛(570)은 처리를 지속할 것인지의 여부를 판정한다.

DB 관리 프로그램(56)은 처리가 지속되면 단계 S560의 처리를 진행하고 그렇지 않으면 처리를 종료한다.

도 58은 다수의 DB 서버에 걸쳐 저장된 동일 레벨에서의 엔트리를 나타낸다.

가령, 하나의 엔트리 혹은 하나의 서브 트리에 대한 액세스 집중은 다음의 방법 (1) 또는 (2)에 의해 결정된다.

(1) 액세스 집중 인덱스(액세스 집중 인덱스=검색 조건을 만족하는 엔트리의 수/특정 레벨에 존재하는 엔트리의 수(도 51(a) 참조))는 사전 결정된 측정 시간 내의 모든 검색 조작에 대해 계산된다.

각각의 검색 조작의 실행 회수가 사전 설정된 회수보다 크고 계산된 액세스 집중 인덱스가 임계값이하일 때, 검색 조건에 관한 액세스 집중의 빈도가 결정된다.

(2) 서브 트리들을 저장하는 DB 서버는 측정 타겟으로 설정되며, 각각의 서브 트리의 최상 엔트리(top entry)와 그 최상 엔트리와 동일한 레벨에 속하는 모든 엔트리가 검색 요구에 따라 질문 베이스로서 사용되는 처리의 회수(주파수)는 서브 트리에 대한 액세스의 주파수로서 설정되며, 액세스 집중은 방법 (1)의 것과 유사한 방법으로 결정된다.

제 2 검색 프로그램(44)

도 59는 도 24에 도시된 DB 시스템(4)에서 서브 트리들이 신규로 제공되고 이동/교환될 때 수집 장치(40) 상에서 실행되는 제 2 검색 프로그램을 나타낸다.

도 59를 참조하면, 제 2 검색 프로그램(44)은 검색 조건 변환 유닛(440)이 도 38에 도시된 제 1 수집 장치(40)에 추가되는 구성을 사용한다.

검색 프로그램(44)은 제 1 수집 장치(40)의 기능과 유사한 기능에 추가하여 검색 조건 관리 테이블을 사용함으로써 검색 조건을 변환한다.

검색 프로그램(44)에서, 검색 조건 변환 유닛(440)은 DB 관리 유닛(5)으로부터 검색 조건 관리 테이블(도 52)을 수신하고, 검색 조건 작성 유닛(420)에 의해 작성된 검색 조건에 포함된 질문을 시작하는 필터 및 검색(질문 베이스)이 검색 조건 관리 테이블에 도시된 조건과 매칭하는지의 여부를 판정한다.

검색 조건에 포함된 필터 및 질문 베이스가 검색 조건 관리 테이블에 도시된 조건과 매칭할 때, 검색 조건 변환 유닛(440)은 질문을 시작하는 원본 검색(질문 베이스; queryBaseDN)을 질문을 시작하는 새로운 검색(새로운 질문 베이스; groupDN)으로 변경하며, 상기 검색 조건 작성 유닛(420)을 제어하여 상기 필터를 검색 조건 관리 테이블에 도시된 필터(queryFilter)가 검색 조건의 필터로부터 제거되는 새로운 필터로 변환한다.

서브 트리가 새롭게 작성될 때의 DB 시스템(4)의 조작

이후, 서브 트리가 새롭게 작성될 때의 DB 시스템(4)의 전체 조작이 기술될 것이다.

도 51(a)를 참조하면 위에서 기술된 바와 같이, 엔트리가 각각의 검색 조작을 위한 검색 조건과 매칭하는 회수는 DB 서버(6)에 의해 카운팅되며, 그러한 카운팅 결과는 DB 관리 서버(5)에 의해 합산되며, 그에 의해 빈번하게 액세스되는 엔트리의 집합과 특정 조건하의 최상 엔트리가 획득될 수 있다.

액세스가 집중되는 엔트리의 집합 내에 최상 엔트리(groupDN)를 제공하고 그의 하위 엔트리를 새로운 서브 트리로서 구성함으로써 액세스 집중이 결정될 때, 엔트리의 집합에 대한 액세스를 국소화하고 DB 시스템의 성능의 저하를 방지할 수 있다.

DB 서버(6)는 DB 관리 서버(5)에 대해 각각의 엔트리에 대한 액세스의 회수와 액세스를 위해 사용되는 필터를 통지한다.

DB 관리 서버(5)는 엔트리에 대한 액세스의 회수(주파수) 및 DB 서버(6)로부터의 액세스를 위해 사용되는 필터의 수를 카운팅한다.

소정의 DB 서버(6)의 서브 트리에 대한 액세스가 도 51(a) 및 도 51(b)에 도시된 바와 같이 빈번할 때, DB 관리 서버(5)는 DB 서버(6)를 제어하여 새로운 서브 트리를 작성한다.

DB 관리 서버(5)는 새롭게 작성된 서브 트리에 매칭하도록 검색 조건 관리 테이블(도 52)의 내용을 변경한다.

DB 관리 서버(5)는 변경된 검색 조건 관리 테이블을 수집 장치(40)에 제공한다.

검색기의 검색 요구에 응답하여 검색 조작에 기반한 검색 조건을 작성할 때, 수집 장치(40)는 DB 관리 서버(5)로부터 제공된 검색 조건 관리 테이블을 사용하여 변환을 수행하고 DB 서버(6)에서 검색을 수행한다.

서브 트리가 이동/교환될 때의 DB 시스템(4)의 조작

다음, 서브 트리가 이동/교환될 때의 DB 시스템(4)의 전체 조작이 기술될 것이다.

도 58을 참조하여 위에서 기술된 바와 같이, 각각의 엔트리가 다수의 DB 서버에 걸쳐 동일한 레벨에서 엔트리에 대한 질문 베이스로서 설정되는 회수를 합산함으로써, 서브 트리에 대한 액세스 집중이 결정될 수 있다.

DB 서버(6)는 각각의 엔트리가 질문 베이스로서 사용되는 회수를 측정하고 그 결과를 DB 관리 서버(5)에 통지한다.

DB 관리 서버(5)는 각각의 엔트리가 각각의 DB 서버(6)의 최상 엔트리와 그 최상 엔트리와 동일한 레벨에서 다른 DB 서버들의 모든 엔트리에 대한 질문 베이스로서 설정되는 회수를 카운팅하며, 각각의 서브 트리의 액세스 주파수를 계산한다.

소정의 DB 서버(6)의 서브 트리에 대한 액세스가 도 53(a) 및 도 53(b)에 도시된 바와 같이 빈번할 때, DB 관리 서버(5)는 DB 서버(6)를 제어하여 서브 트리를 이동/교환한다.

또한, DB 관리 서버(5)는 디렉토리 트리 테이블(도 46) 및 데이터 테이블(도 47) 등의 내용을 변경하여 이동/교환 서브 트리에 매칭시킨다.

DB 관리 서버(5)는 변경된 디렉토리 트리 및 데이터 테이블을 수집 장치(40)에 제공한다.

수집 장치(40)는 제공된 디렉토리 트리 및 데이터 테이블을 검색기의 검색 조작에 따라 사용함으로써 DB 서버에서 검색을 수행한다.

주목해야 할 것은 본 발명의 개시된 일부의 실시예가 컴퓨터 관독가능한 매체에 포함되고 컴퓨터에 의해 실행가능한 소프트웨어 인스트럭션의 형태로 구현되지만 본 발명은 이러한 장치에 국한되는 것이 아니다. 대안의 실시예에서, 소프트웨어 인스트럭션 대신에 혹은 그와 조합하여 하드 와이어형 회로(hard-wired circuitry)가 사용될 수도 있다. 따라서, 본 발명은 하드웨어 회로 및 소프트웨어의 특정 조합에 국한되지 않는다.

발명의 효과

또한 본 발명의 개시된 실시예는 다음의 효과를 제공한다.

- (1) 개시된 실시예는 데이터베이스의 체계를 변경할 필요없이 다양한 종류의 정보, 특히 데이터 요소 혹은 필드를 용이하게 추가할 수 있는 데이터 모델 및 데이터베이스 시스템을 제공한다.
- (2) 개시된 실시예는 또한 데이터가 재생을 위해 독특하게 분석되고 저장되고 검색되는 데이터 모델 및 데이터베이스 시스템을 제공한다.
- (3) 개시된 실시예는 또한 부하가 다수의 처리 장치들에 걸쳐 용이하게 분포될 수 있는 데이터베이스 시스템 및 관리 방법 및 소프트웨어를 제공한다.
- (4) 개시된 실시예는 또한 사용자가 등록된 정보를 검색할 수 있는 데이터 모델 및 데이터베이스 시스템 및 관리 방법 및 소프트웨어를 제공한다.
- (5) 개시된 실시예는 기존의 데이터베이스를 데이터베이스의 체계를 변경할 필요없이 다양한 종류의 정보, 특히 데이터 요소 혹은 필드를 용이하게 추가할 수 있는 데이터 모델 및 데이터베이스 시스템으로 변환하는 방법을 제공한다.

본 발명의 특정 실시예가 기술 및 도시되었지만, 분명한 것은 특정적으로 예시 및 기술된 실시예의 세부사항에 관해 첨부된 청구범위에서 규정되는 바와 같은 본 발명의 진정한 사상 및 영역의 범위 내에서 변형이 가해질 수 있다는 것이다.

(57) 청구의 범위

청구항 1.

하나 이상의 관련 노드와 하나 이상의 토픽 노드를 각각 관련시켜서 디렉토리 트리 형태의 데이터베이스를 작성하는 데이터베이스 시스템에 있어서,

상기 하나 이상의 상기 토픽 노드 각각은 자신에게 속하는 데이터를 구비하고,

상기 관련된 관련 노드와 상기 토픽 노드 사이의 각각의 관련성을 나타내도록 관련 속성이 정의되며,

상기 관련 노드 각각의 관련 노드 엔트리와 상기 관련 속성 각각의 관련 속성 엔트리를 작성하고, 상기 각각의 관련 노드와 상기 각각의 관련 속성 사이의 관련성에 따라서 상기 엔트리를 상관시킴으로써 디렉토리 트리를 작성하는 디렉토리 트리 작성 수단과,

상기 토픽 노드와 상기 각각의 관련 속성 사이의 관련성에 따라서, 상기 토픽 노드 중 하나에 속하는 데이터와 상기 작성된 관련 속성 엔트리를 상관시키는 데이터 상관 수단을 포함하는

데이터베이스 시스템.

청구항 2.

제 1 항에 있어서,

상기 관련 속성은 상기 관련된 관련 노드와 토픽 노드 사이에 정의된 역할을 나타내는

데이터베이스 시스템.

청구항 3.

제 1 항에 있어서,

상기 디렉토리 트리는 하나 이상의 서브 트리를 포함하고,

상기 디렉토리 트리의 상위 엔트리는 상기 디렉토리 트리 및 상기 서브 트리에 대해서 정의되며,

상기 서브 트리 각각의 적어도 하나의 상위 엔트리는 상기 디렉토리 트리 및/또는 상기 다른 서브 트리의 상기 엔트리 중 적어도 하나를 나타내고,

상기 관련 노드와 토픽 노드는 상기 디렉토리 트리와 상기 서브 트리에 계층 방식으로 상관되는

데이터베이스 시스템.

청구항 4.

제 3 항에 있어서,

상기 서브 트리 각각의 적어도 하나의 상위 엔트리와 상기 디렉토리 트리의 적어도 하나의 엔트리 및/또는 상기 적어도 하나의 상위 엔트리에 의해 나타내어지는 다른 서브 트리를 상관시키는 상위 엔트리 상관 수단을 더 포함하는

데이터베이스 시스템.

청구항 5.

제 3 항에 있어서,

상기 디렉토리 트리과 상기 서브 트리 중 하나에 포함된 상기 토픽 노드의 적어도 하나에 속하는 데이터 - 상기 데이터는 상기 적어도 하나의 토픽 노드와 관련된 상기 관련 노드에 속함 - 및 상기 디렉토리 트리과 상기 서브 트리 중 하나에 따라서 상기 관련 토픽과 관련 노드 사이에 정의된 역할을 저장하는 하나 이상의 제 1 데이터베이스 서버를 더 포함하는

데이터베이스 시스템.

청구항 6.

제 5 항에 있어서,

상기 제 1 데이터베이스 서버 또는 다른 제 1 데이터베이스 서버에 저장된 상기 데이터를 전송하고 저장하는 데이터 전송 수단을 더 포함하는

데이터베이스 시스템.

청구항 7.

제 6 항에 있어서,

상기 데이터 전송 수단은 데이터가 전송될 상기 제 1 데이터베이스 서버의 동작 상태에 기초해서 전송될 데이터를 결정하는

데이터베이스 시스템.

청구항 8.

제 6 항에 있어서,

상기 데이터 전송 수단은 상기 디렉토리 트리 또는 상기 서브 트리로의 액세스 상태에 기초해서 전송될 데이터를 결정하는

데이터베이스 시스템.

청구항 9.

제 5 항에 있어서,

하나 이상의 상기 제 1 데이터베이스 서버에 저장된 상기 데이터의 복제를 저장하는 제 2 데이터베이스 서버와,

적어도 하나의 상위 엔트리가 나타내는 상기 디렉토리 트리 및/또는 다른 서브 트리의 적어도 하나의 엔트리와 상기 서브 트리 각각의 적어도 하나의 상위 엔트리를 상관시키고, 상기 제 1 데이터베이스 서버의 상위 엔트리와 상기 제 2 데이터베이스 서버를 더 상관시키는 상위 엔트리 상관 수단

을 더 포함하는 데이터베이스 시스템.

청구항 10.

제 9 항에 있어서,

상기 디렉토리 트리 및 상기 서브 트리 각각은 하나 이상의 서브 트리를 더 포함하고,

상기 데이터베이스 시스템은 상기 디렉토리 트리와 상기 서브 트리 각각으로부터 적어도 하나의 상기 추가 서브 트리를 작성하는 서브 트리 작성 수단을 더 포함하는

데이터베이스 시스템.

청구항 11.

제 9 항에 있어서,

상기 관련 노드와 상기 토픽 노드를 분류하는 데 사용되는 분류 정보가 상기 디렉토리 트리와 상기 서브 트리 각각에 대해서 정의되고,

상기 데이터베이스 시스템은 상기 디렉토리 트리 및 상기 서브 트리 각각을 상기 디렉토리 트리 또는 서브 트리에 대해 정의된 상기 분류 정보와 상관시키는 분류 정보 상관 수단을 더 포함하는

데이터베이스 시스템.

청구항 12.

제 11 항에 있어서,

상기 토픽 노드 중 적어도 하나에 속하는 상기 저장 데이터를 수집하는 수집 장치를 더 포함하는

데이터베이스 시스템.

청구항 13.

제 12 항에 있어서,

상기 수집 장치는

검색 조건을 수용하는 검색 조건 수용 수단과,

상기 수용된 검색 조건에 대응하는 상기 분류 정보에 상관된 상기 디렉토리 트리 또는 상기 서브 트리를 검색하는 디렉토리 트리 검색 수단과,

상기 디렉토리 트리 검색 수단이 찾은 상기 디렉토리 트리 또는 상기 서브 트리에 포함된 상기 토픽 노드 중 적어도 하나에 속하는 상기 저장 데이터를 검색하는 데이터 검색 수단

을 포함하는

데이터베이스 시스템.

청구항 14.

하나 이상의 관련 노드와 하나 이상의 토픽 노드를 각각 관련시켜서 디렉토리 트리 형태의 데이터베이스를 작성하는 데이터 관리 방법에 있어서,

상기 하나 이상의 상기 토픽 노드 각각은 자신에게 속하는 데이터를 구비하고,

상기 관련된 관련 노드와 상기 토픽 노드 사이의 각각의 관련성을 나타내도록 관련 속성이 정의되며,

상기 관련 노드 각각의 관련 노드 엔트리와 상기 관련 속성 각각의 관련 속성 엔트리를 작성하는 단계와,

상기 각각의 관련 노드와 상기 각각의 관련 속성 사이의 관련성에 따라서 상기 엔트리를 상관시킴으로써 디렉토리 트리를 작성하는 단계와,

상기 토픽 노드와 상기 각각의 관련 속성 사이의 관련성에 따라서, 상기 토픽 노드 중 하나에 속하는 데이터와 상기 작성된 관련 속성 엔트리를 상관시키는 단계를 포함하는

데이터 관리 방법.

청구항 15.

제 14 항에 있어서,

상기 디렉토리 트리는 하나 이상의 서브 트리를 포함하고,

상기 디렉토리 트리의 상위 엔트리는 상기 디렉토리 트리 및 상기 서브 트리에 대해서 정의되며,

상기 서브 트리 각각의 적어도 하나의 상위 엔트리는 상기 디렉토리 트리 및/또는 상기 다른 서브 트리의 상기 엔트리 중 적어도 하나를 나타내고,

상기 디렉토리 트리과 상기 서브 트리 각각은 하나 이상의 추가 서브 트리를 더 포함하며,

상기 방법은 상기 디렉토리 트리과 상기 서브 트리 각각으로부터 적어도 하나의 상기 추가 서브 트리를 작성하는 단계를 더 포함하는

데이터 관리 방법.

청구항 16.

제 14 항에 있어서,

상기 디렉토리 트리는 하나 이상의 서브 트리를 포함하고,

상기 디렉토리 트리의 상위 엔트리는 상기 디렉토리 트리 및 상기 서브 트리에 대해서 정의되며,

상기 서브 트리 각각의 적어도 하나의 상위 엔트리는 상기 디렉토리 트리 및/또는 상기 다른 서브 트리의 상기 엔트리 중 적어도 하나를 나타내고,

상기 방법은

상기 디렉토리 트리와 상기 서브 트리 중 하나에 포함된 상기 토픽 노드의 적어도 하나에 속하는 데이터 - 상기 데이터는 상기 적어도 하나의 토픽 노드와 관련된 상기 관련 노드에 속함 - 및 상기 디렉토리 트리와 상기 서브 트리 중 하나에 따라서 상기 관련 토픽과 관련 노드 사이에 정의된 역할을 저장하는 단계와,

상기 토픽 노드에 속하는 상기 저장 데이터를 검색하는 단계

를 포함하는 데이터 관리 방법.

청구항 17.

제 16 항에 있어서,

검색 조건을 수용하는 단계와,

상기 수용된 검색 조건에 대응하는 분류 정보에 상관된 상기 디렉토리 트리 또는 상기 서브 트리를 검색하는 단계와,

상기 검색 결과로서 획득된 상기 디렉토리 트리 또는 상기 서브 트리에 포함된 상기 토픽 노드에 속하는 상기 저장 데이터를 검색하는 단계

를 더 포함하는 데이터 관리 방법.

청구항 18.

제 14 항에 있어서,

상기 디렉토리 트리와 상기 서브 트리 중 하나에 포함된 상기 토픽 노드의 적어도 하나에 속하는 데이터 - 상기 데이터는 상기 적어도 하나의 토픽 노드와 관련된 상기 관련 노드에 속함 - 및 상기 디렉토리 트리와 상기 서브 트리 중 하나에 따라서 복수의 제 1 데이터베이스 서버 내의 상기 관련 토픽과 관련 노드 사이에 정의된 역할을 저장하는 단계와,

상기 제 1 데이터베이스 서버 또는 다른 제 1 데이터베이스 서버에 저장된 상기 데이터를 전송하고 저장하는 단계

를 더 포함하는 데이터 관리 방법.

청구항 19.

하나 이상의 관련 노드와 하나 이상의 토픽 노드를 각각 관련시켜서 디렉토리 트리 형태의 데이터베이스를 작성하는 컴퓨터를 포함하는 데이터베이스 시스템에 사용되는 프로그램이 저장된 컴퓨터 판독 가능 저장 매체에 있어서,

상기 하나 이상의 상기 토픽 노드 각각은 자신에게 속하는 데이터를 구비하고,

상기 관련된 관련 노드와 상기 토픽 노드 사이의 각각의 관련성을 나타내도록 관련 속성이 정의되며,

상기 프로그램은 상기 데이터베이스 시스템에서 수행될 때 상기 컴퓨터로 하여금

상기 관련 노드 각각의 관련 노드 엔트리와 상기 관련 속성 각각의 관련 속성 엔트리를 작성하는 단계와,

상기 각각의 관련 노드와 상기 각각의 관련 속성 사이의 관련성에 따라서 상기 엔트리를 상관시킴으로써 디렉토리 트리를 작성하는 단계와,

상기 토픽 노드와 상기 각각의 관련 속성 사이의 관련성에 따라서, 상기 토픽 노드 중 하나에 속하는 데이터와 상기 작성된 관련 속성 엔트리를 상관시키는 단계를

더 수행하게 하는

컴퓨터 판독 가능 매체.

청구항 20.

제 19 항에 있어서,

상기 디렉토리 트리는 하나 이상의 서브 트리를 포함하고,

상기 디렉토리 트리의 상위 엔트리는 상기 디렉토리 트리 및 상기 서브 트리에 대해서 정의되며,

상기 서브 트리 각각의 적어도 하나의 상위 엔트리는 상기 디렉토리 트리 및/또는 상기 다른 서브 트리의 상기 엔트리 중 적어도 하나를 나타내고,

상기 디렉토리 트리과 상기 서브 트리 각각은 하나 이상의 서브 트리를 더 포함하며,

상기 프로그램은 상기 데이터베이스 시스템에서 수행될 때 상기 컴퓨터로 하여금 상기 디렉토리 트리와 상기 서브 트리 각각으로부터 적어도 하나의 추가 서브 트리를 작성하는 단계를 더 수행하게 하는

컴퓨터 판독 가능 매체.

청구항 21.

제 19 항에 있어서,

상기 디렉토리 트리는 하나 이상의 서브 트리를 포함하고,

상기 디렉토리 트리의 상위 엔트리는 상기 디렉토리 트리 및 상기 서브 트리에 대해서 정의되며,

상기 서브 트리 각각의 적어도 하나의 상위 엔트리는 상기 디렉토리 트리 및/또는 상기 다른 서브 트리의 상기 엔트리 중 적어도 하나를 나타내고,

상기 프로그램은 상기 데이터베이스 시스템에서 수행될 때 상기 컴퓨터로 하여금

상기 디렉토리 트리와 상기 서브 트리 중 하나에 포함된 상기 토픽 노드의 적어도 하나에 속하는 데이터 - 상기 데이터는 상기 적어도 하나의 토픽 노드와 관련된 상기 관련 노드에 속함 - 및 상기 디렉토리 트리와 상기 서브 트리 중 하나에 따라서 상기 관련 토픽과 관련 노드 사이에 정의된 역할을 저장하는 단계와,

상기 토픽 노드에 속하는 상기 저장 데이터를 검색하는 단계

를 더 수행하게 하는

컴퓨터 판독 가능 매체.

청구항 22.

제 21 항에 있어서,

상기 프로그램은 상기 데이터베이스 시스템에서 수행될 때 상기 컴퓨터로 하여금

검색 조건을 수용하는 단계와,

상기 수용된 검색 조건에 대응하는 분류 정보에 상관된 상기 디렉토리 트리 또는 상기 서브 트리를 검색하는 단계와,

상기 검색 결과로서 획득된 상기 디렉토리 트리 또는 상기 서브 트리에 포함된 상기 토픽 노드에 속하는 상기 저장 데이터를 검색하는 단계

를 더 수행하게 하는

컴퓨터 판독 가능 매체.

청구항 23.

제 19 항에 있어서,

상기 프로그램은 상기 데이터베이스 시스템에서 수행될 때 상기 컴퓨터로 하여금

상기 디렉토리 트리와 상기 서브 트리 중 하나에 포함된 상기 토픽 노드의 적어도 하나에 속하는 데이터 - 상기 데이터는 상기 적어도 하나의 토픽 노드와 관련된 상기 관련 노드에 속함 - 및 상기 디렉토리 트리와 상기 서브 트리 중 하나에 따라서 복수의 제 1 데이터베이스 서버 내의 상기 관련 토픽과 관련 노드 사이에 정의된 역할을 저장하는 단계와,

상기 제 1 데이터베이스 서버 또는 다른 제 1 데이터베이스 서버에 저장된 상기 데이터를 전송하고 저장하는 단계

를 더 수행하게 하는

컴퓨터 판독 가능 매체.

청구항 24.

데이터 구조가 저장된 컴퓨터 판독 가능 메모리에 있어서,

상기 데이터 구조는

복수의 객체에 속하는 객체 데이터가 저장된 토픽 노드 필드와,

상기 객체들 사이의 관련성을 기술하는 관련 데이터가 저장된 관련 노드 필드와,

상기 관련된 관련성에 대한 상기 객체에 의해 수행되는 역할의 속성을 나타내는 역할 데이터가 저장된 관련 속성 필드를 포함하는

컴퓨터 판독 가능 메모리.

청구항 25.

복수의 링크에 의해 접속되는 복수의 노드를 구비한 제 1 데이터 구조를 복수의 관련 토픽 노드 및 관련 노드를 구비한 제 2 데이터 구조로 변환하는 방법에 있어서,

상기 제 1 데이터 구조의 노드 각각을 상기 제 2 데이터 구조의 상기 토픽 노드 중 하나로 변환하는 단계와,

상기 제 1 데이터 구조의 링크 각각을 상기 제 2 데이터 구조의 상기 관련 노드 중 하나로 변환하는 단계와,

상기 제 2 데이터 구조에서, 관련 노드 각각을, 상기 제 1 데이터 구조에서 상기 관련 노드에 대응하는 링크에 의해 접속되는 노드에 대응하는 2개의 상기 토픽 노드와 관련시키는 단계와,

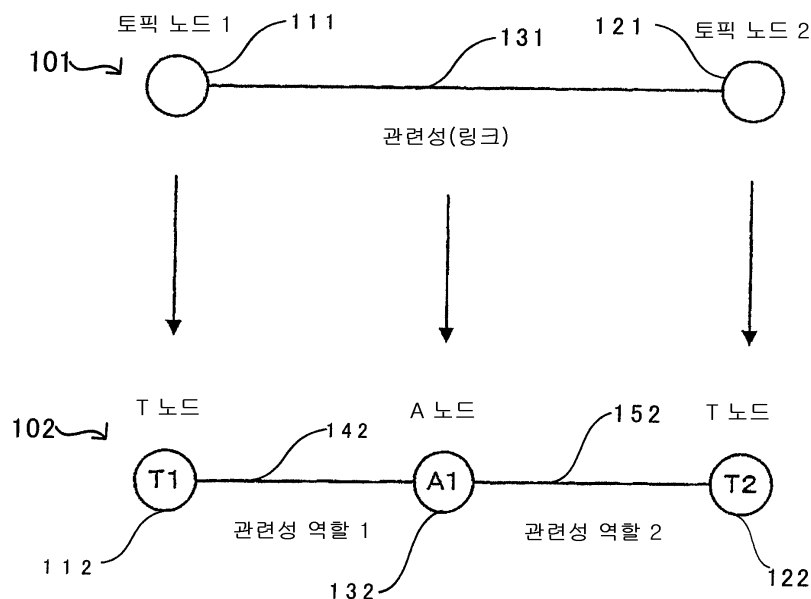
상기 제 2 데이터 구조에서, 관련된 상기 관련 노드와 상기 토픽 노드 사이의 각각의 관련성에 관련 속성을 할당하여 상기 제 1 데이터 구조의 상기 관련 링크에 대한 상기 노드에 의해 수행되는 역할의 속성을 나타내는 단계

를 포함하는

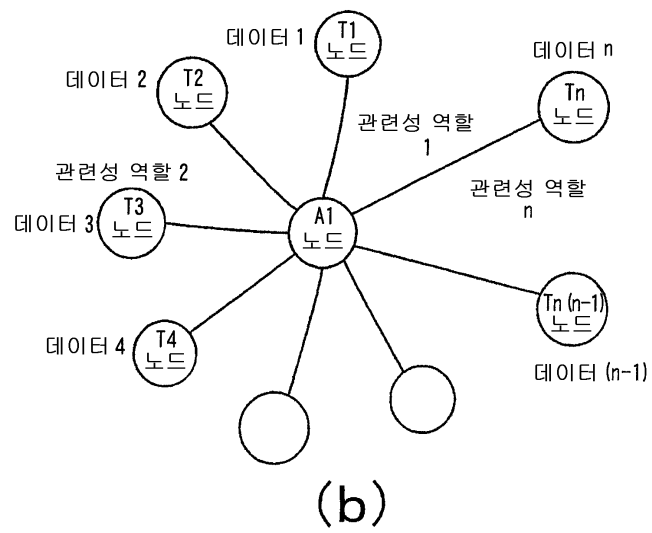
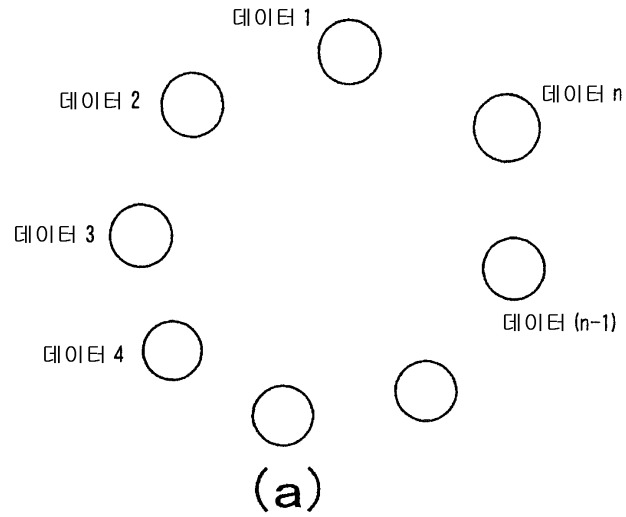
방법.

도면

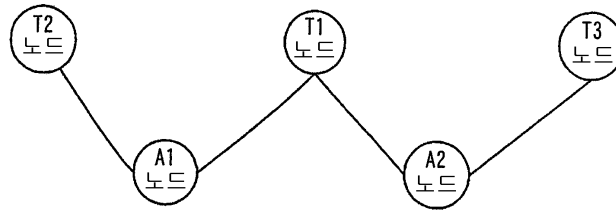
도면1



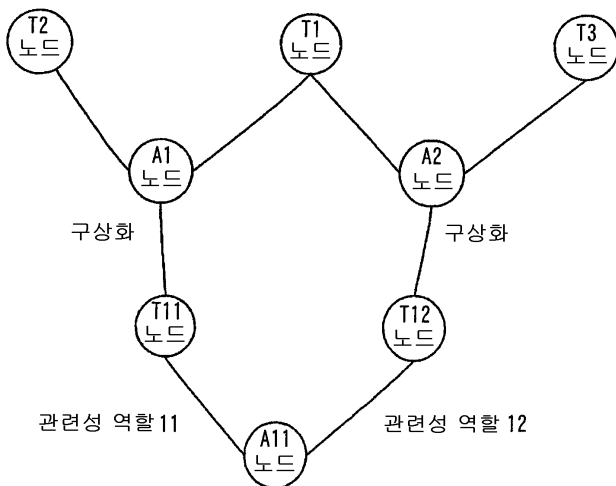
도면2



도면3

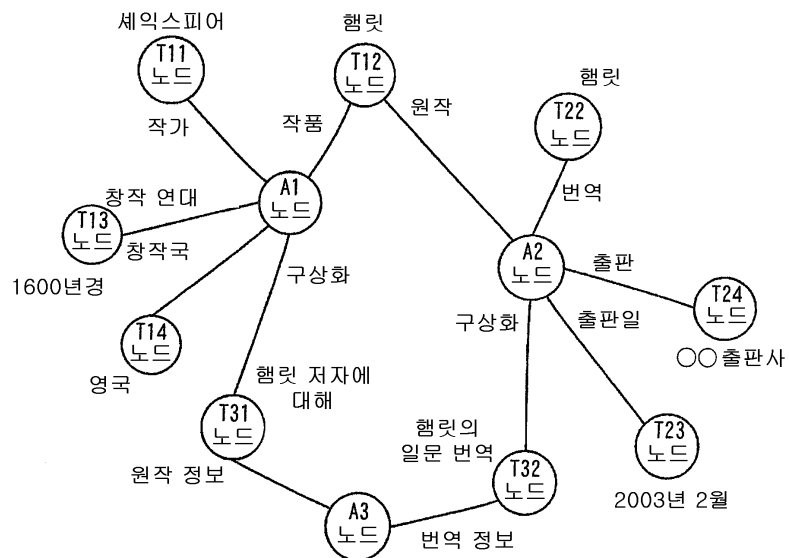


(a)

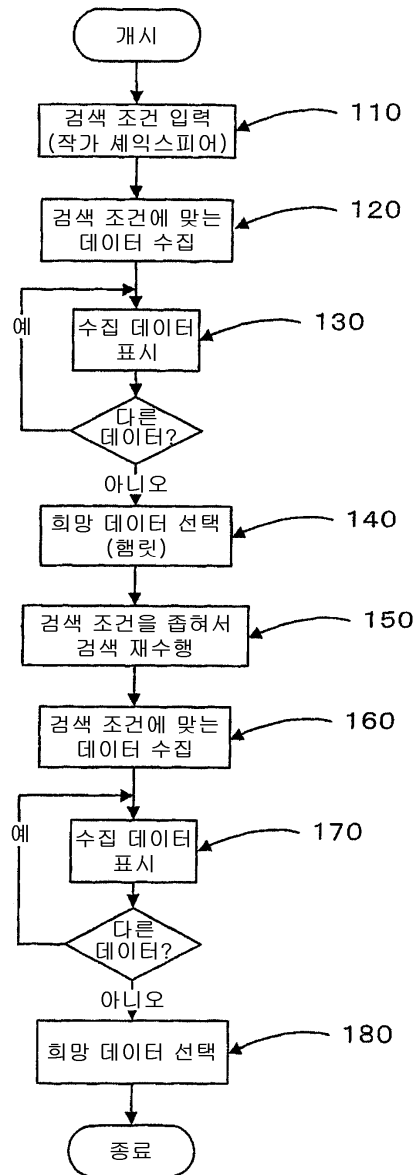


(b)

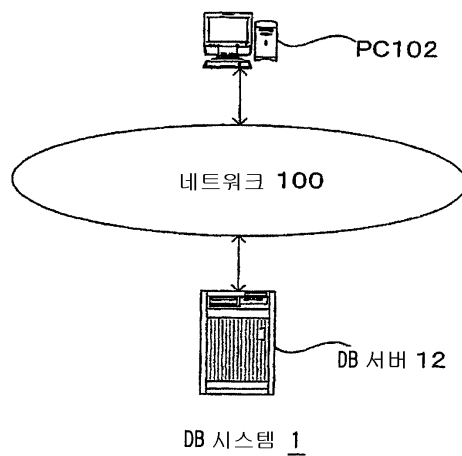
도면4



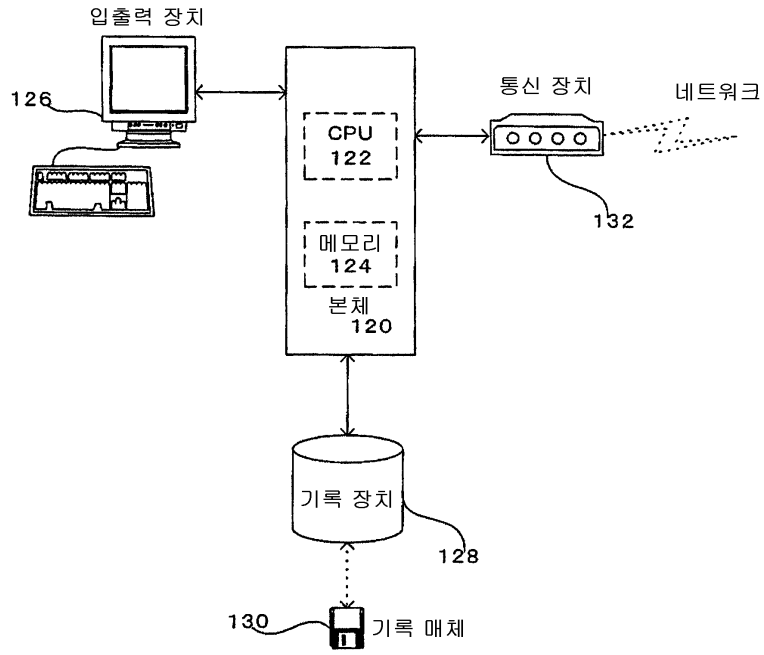
도면5



도면6

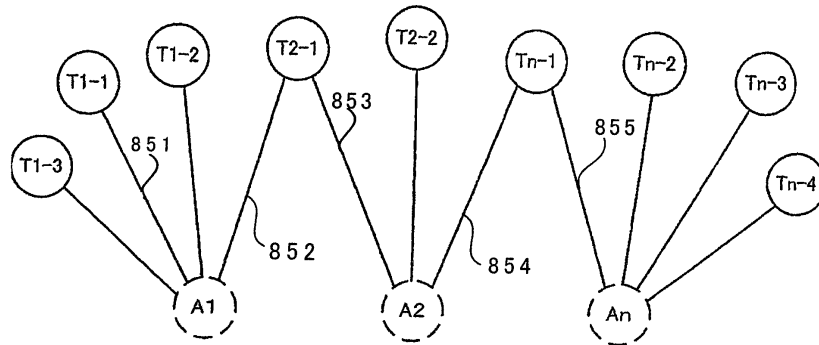


도면7

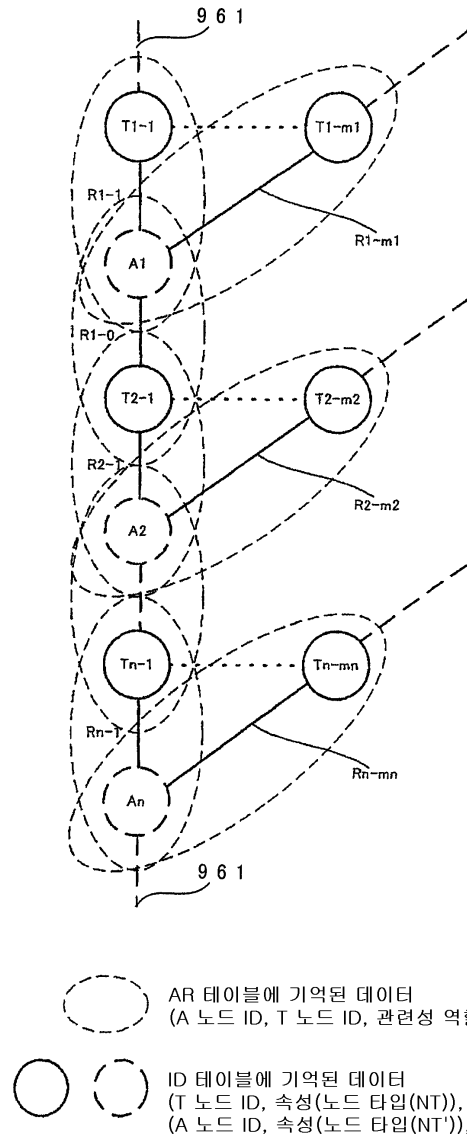


12. 102(30, 32, 40, 5, 6)

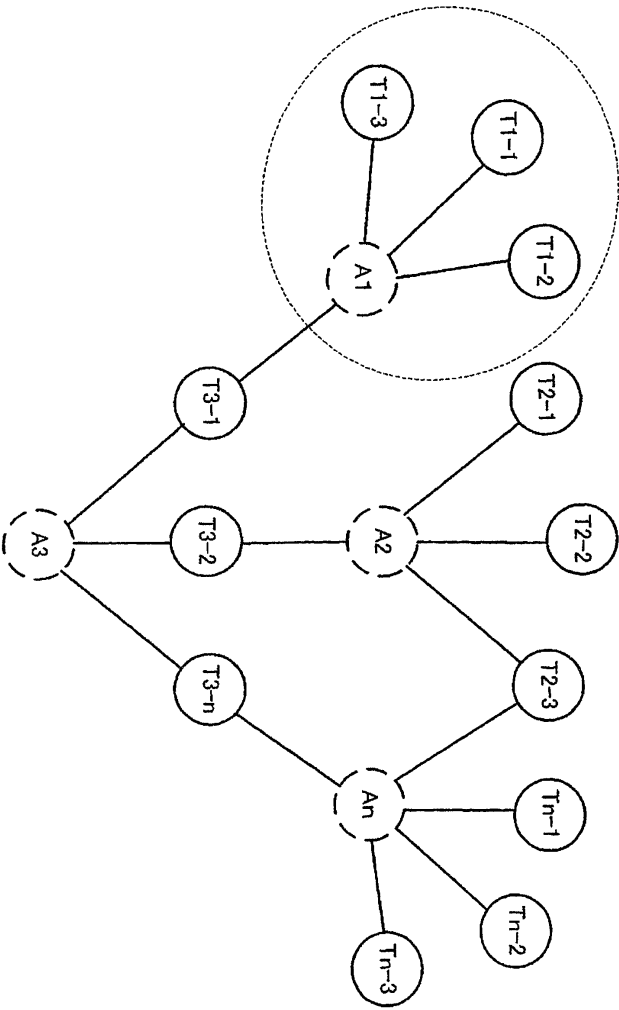
도면8



도면9



도면10



도면11

AR 테이블		
A 노드 ID	T테이블ID (노드 명칭M)	관련성 역할 R
A1	T1-1	R1-1
A1	T1-m1	R1-m1
A1	T2-1	R1-0
A2	T2-1	R2-1
A2	T2-m2	R2-m2
An	Tn-1	Rn-1
An	Tn-mn	Rn-mn

도면12

ID 테이블(T 노드용)

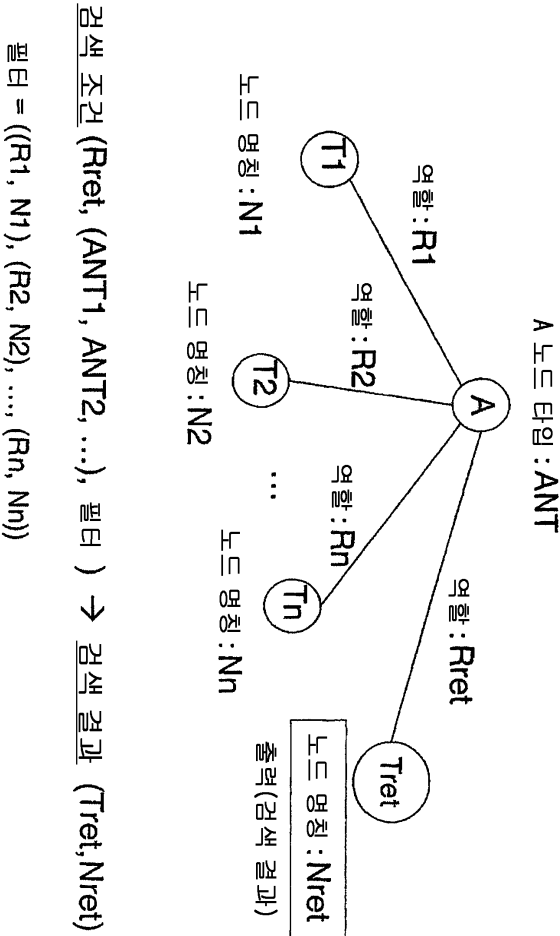
T 노드 ID	노드 타입 (NT)	노드 명칭 (N)
T1-1	NT1-1	N1-1
T1-m1	NT1-m1	N1-m1
T2-1	NT2-1	N2-1
T2-m2	NT2-m2	N2-m2
Tn-1	NTn-1	Nn-1
Tn-mn	NTn-mn	Nn-mn

도면13

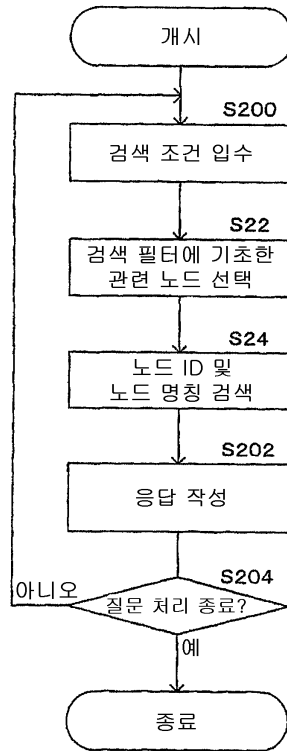
ID 테이블(A 노드용)

A 노드 ID	노드 타입 (NT)	노드 명칭 (N)
A1	NT1	N1
A2	NT2	N2
An	NTn	Nn

도면14

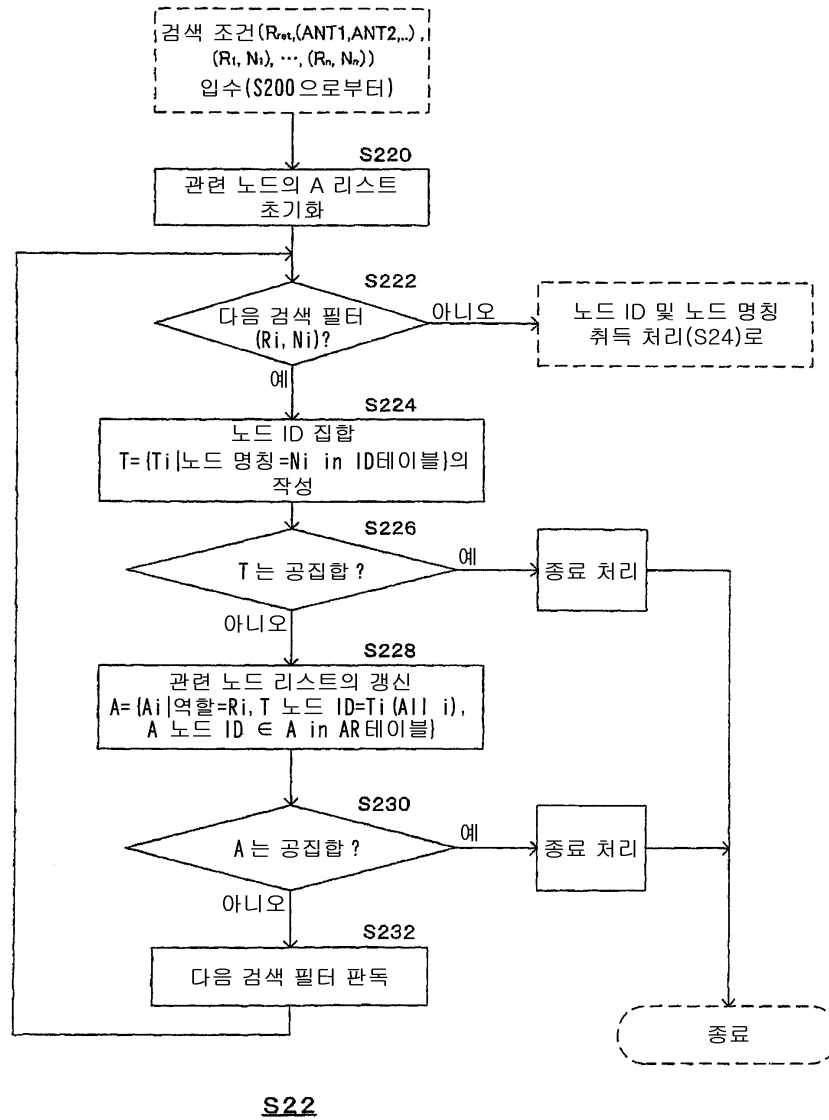


도면15

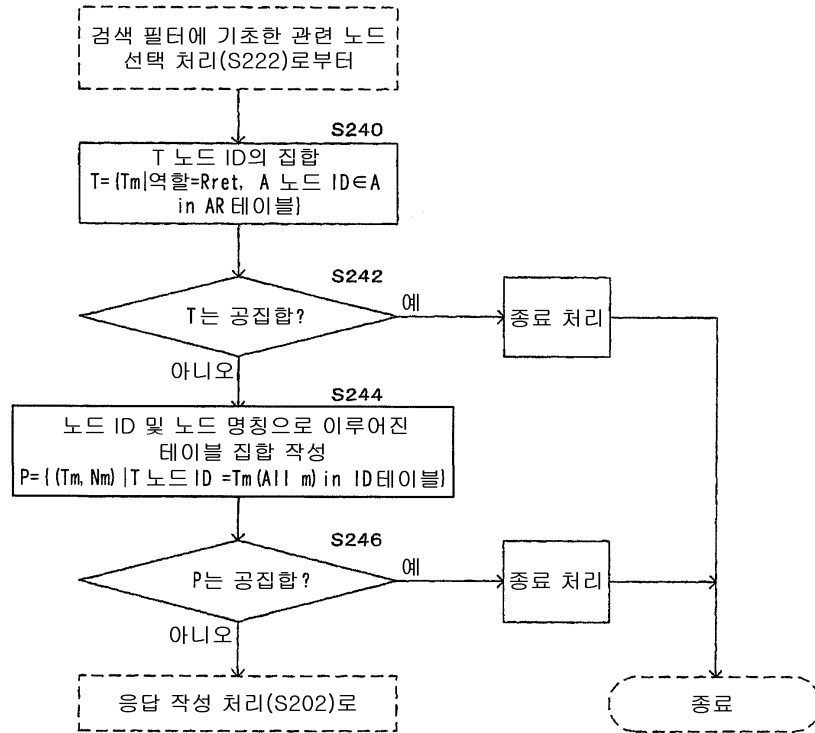


S20

도면16

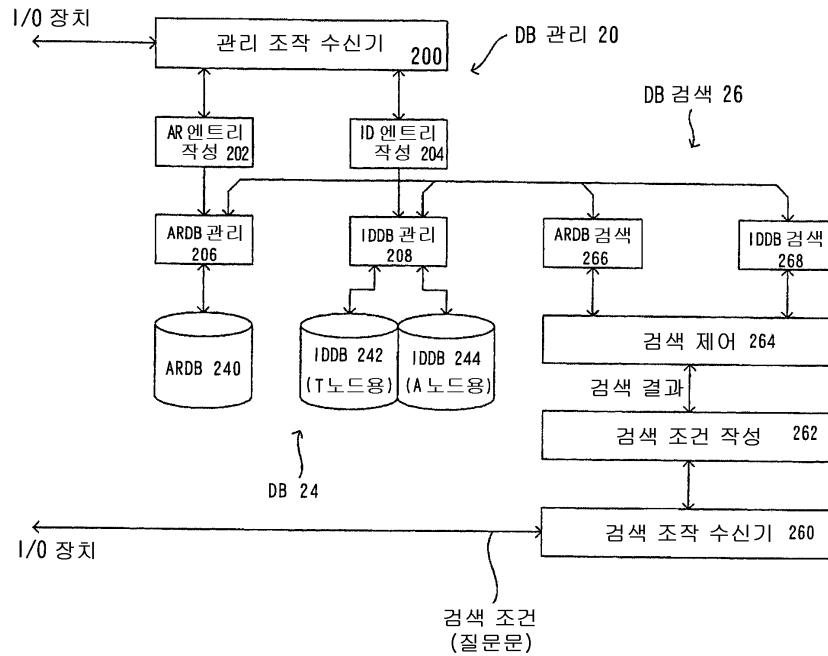


도면17



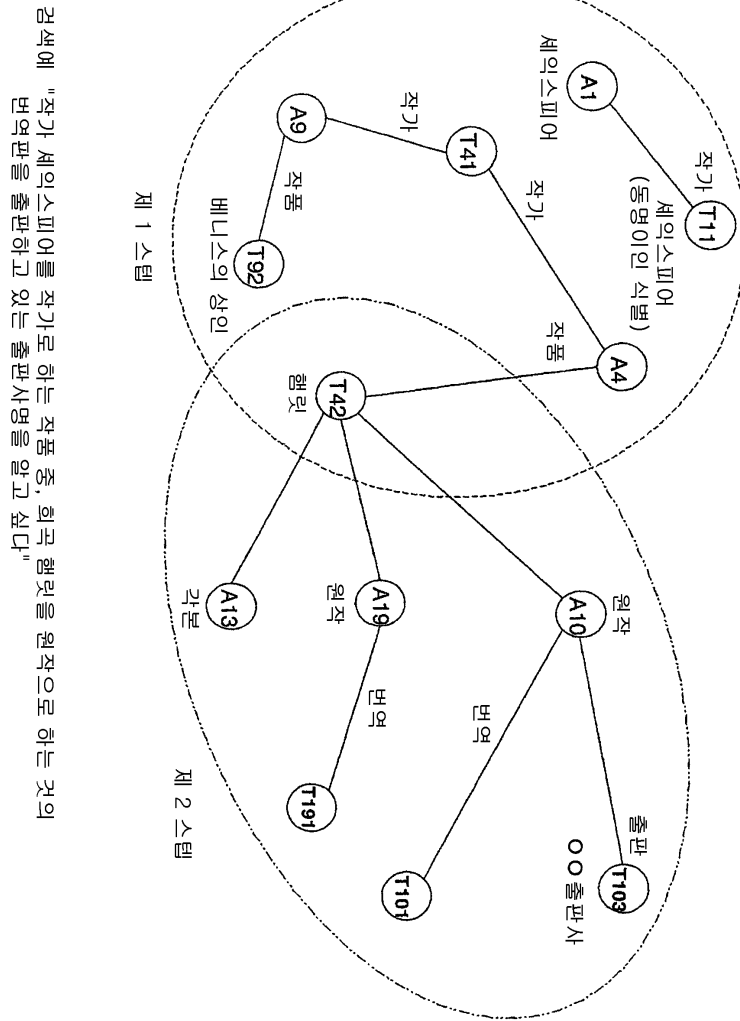
S24

도면18



DB 프로그램 2

도면19



도면20

AR 테이블

A 노드	T 노드	관련성 역할
A1	T11	작가
A4	T41	작가
A4	T42	작품
A9	T41	작가
A9	T92	작품
A10	T42	원작
A10	T101	번역
A10	T103	출판
A13	T42	각본
A19	T42	원작
A19	T191	번역

도면21

ID 테이블 (T 노드용)

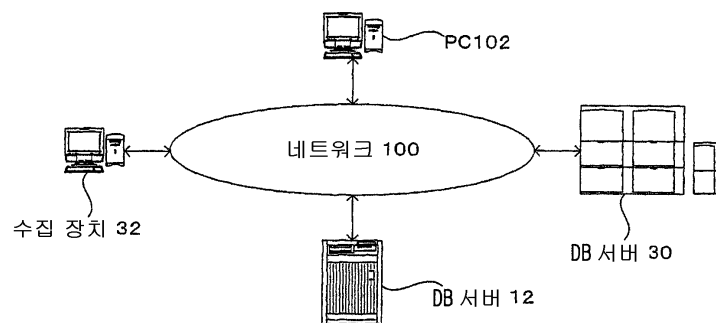
노드 ID	노드 타입	노드 명칭
T11	극작가	셰익스피어
T31	미국인	셰익스피어
T41	극작가	셰익스피어
T42	희곡	햄릿
T51	상점 주인	셰익스피어
T81	장소	셰익스피어
T92	희곡	베니스의 상인
:	:	:
T101	희곡	햄릿의 일본어역
T103	출판사	OO 출판사
T191	희곡	햄릿의 중국어역

도면22

ID 테이블 (A 노드용)

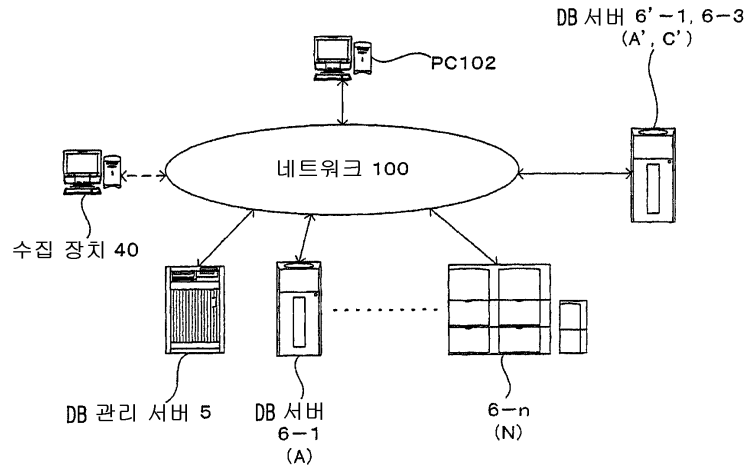
노드 ID	노드 타입	노드 명칭
A1	출생 정보	(null)
A4	저작 정보	(null)
A9	저작 정보	(null)
A10	번역 정보	(null)
A13	공연 정보	(null)
A19	번역 정보	(null)
:	:	:

도면23



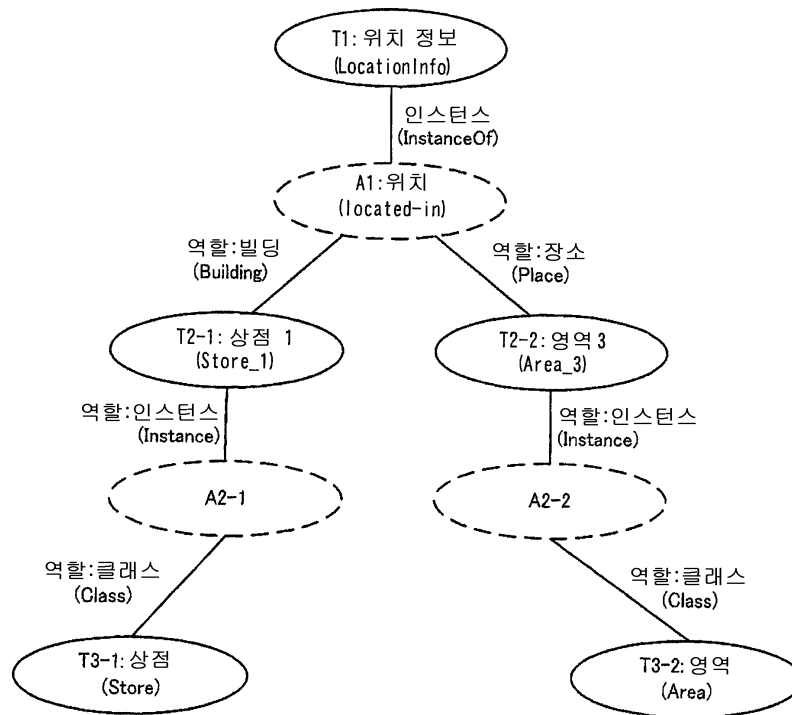
DB 시스템 30

도면24



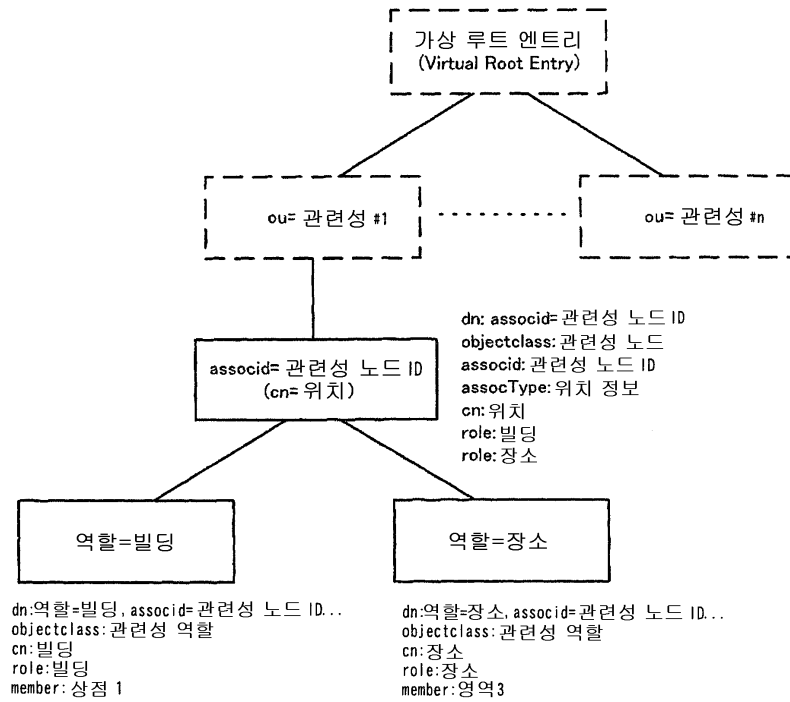
DB시스템 4

도면25



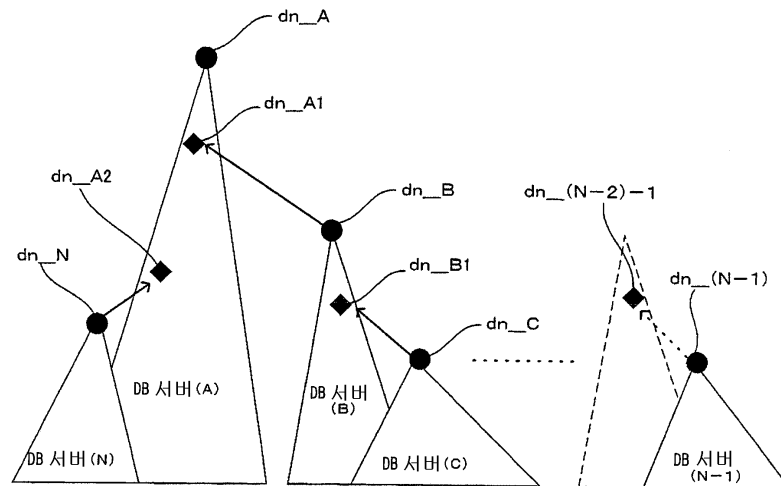
관련 데이터의 그래프 표현

도면26

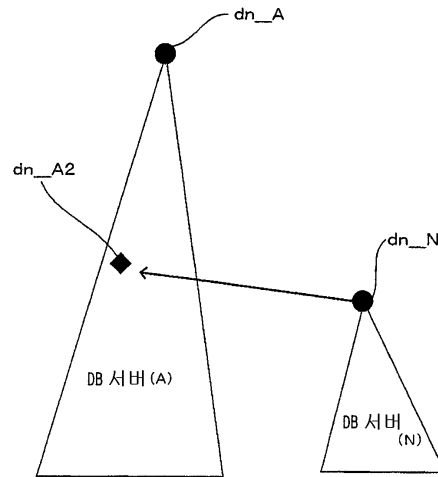


관련성 데이터의 DIT 표시

도면27



도면28



도면29

DB 서버 명칭	상위 엔트리	리퍼링 엔트리	복제 서버
A	dn_A	N/A	A'
B	dn_B	dn_A-1	N/A
C	dn_C	dn_B-1	C'
D	dn_D	dn_A-2	N/A
N	dn_N	dn_(N-1)-1	N/A

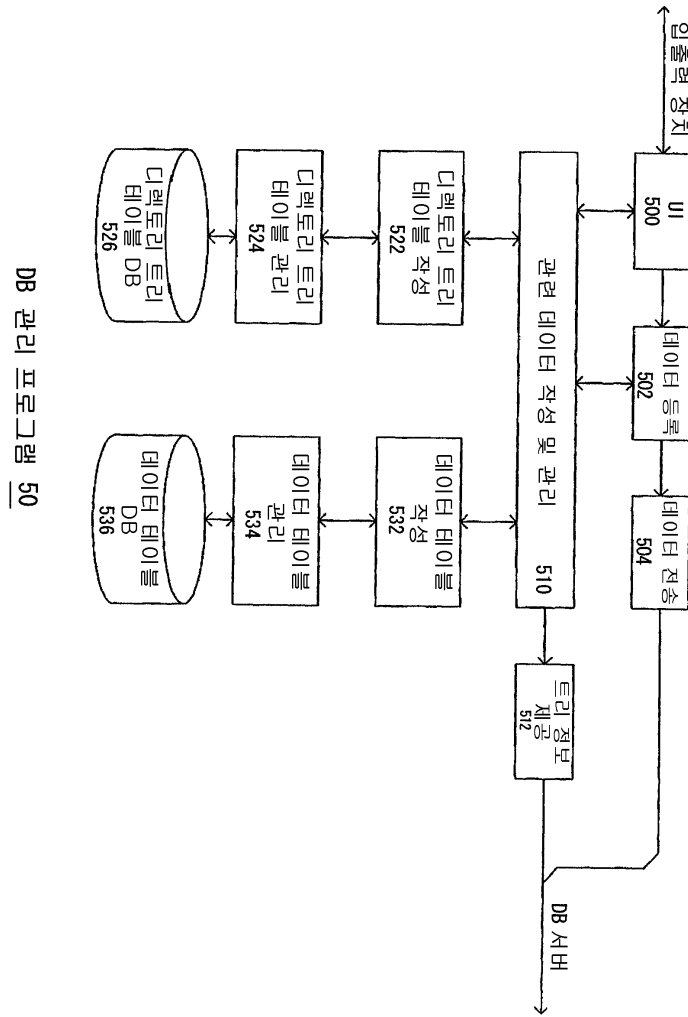
디렉토리 트리 테이블

도면30

서브 트리	분류 타입	분류 값	상위 엔트리
a	카테고리 (Category)	협력사 정보 (Affiliate)	dn_A
b	위치 정보 (LocationInfo)	위치 (located-in)	dn_B
n-1	빌딩 (Building)	상점 1 (Store_1)	dn_(N-1)
n			

데이터 테이블

도면31



도면32

셰익스피어의 작품

~관한 데이터 입력

관련 명칭:

햄릿에 대해

관련 타입:

집필 정보 ▼

기타:

명칭:

셰익스피어

역할:

작가 ▼

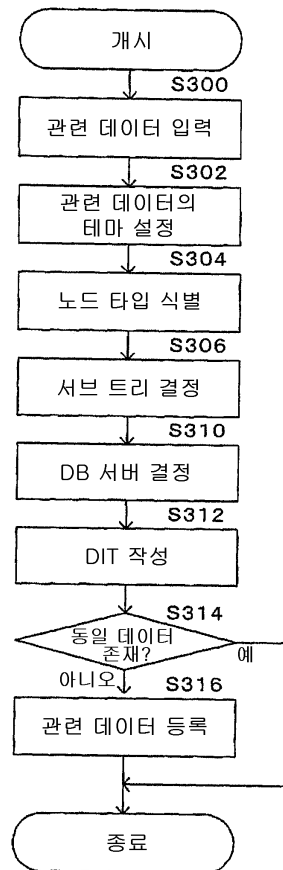
기타:

추가

삭제

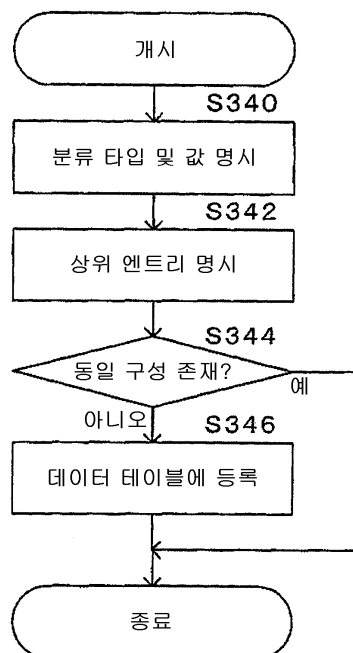
나가기

도면33



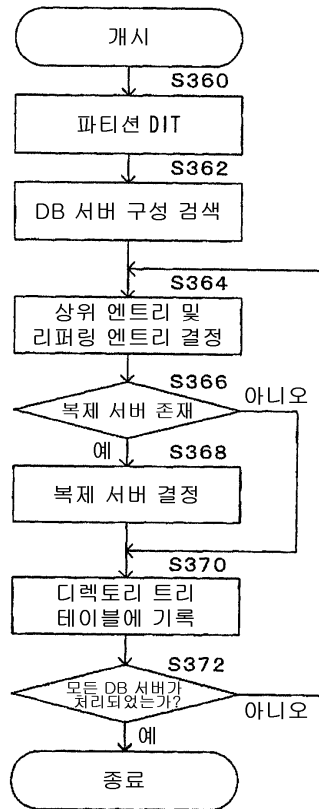
S30

도면34



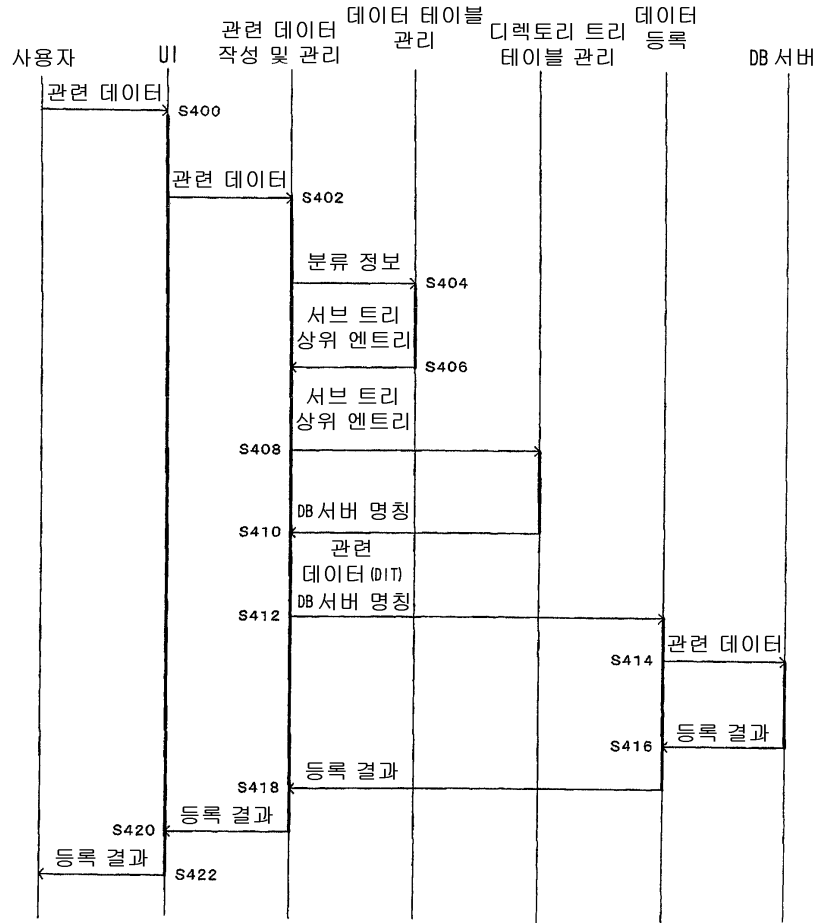
S34

도면35



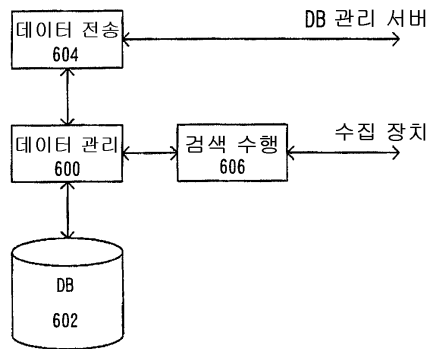
S36

도면36



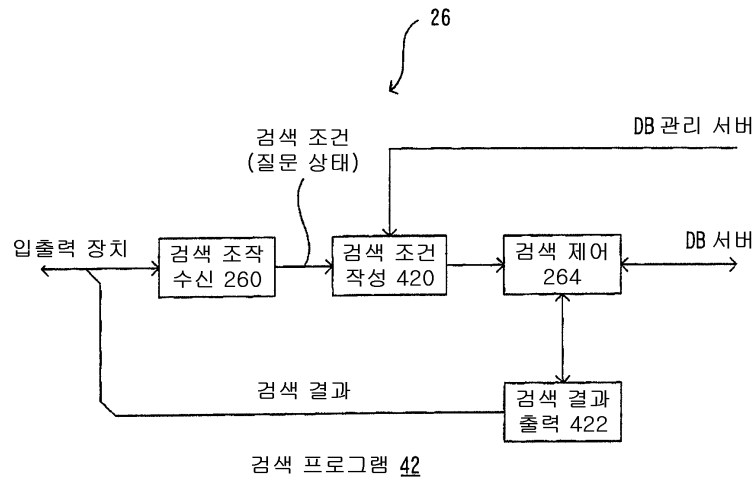
S40

도면37

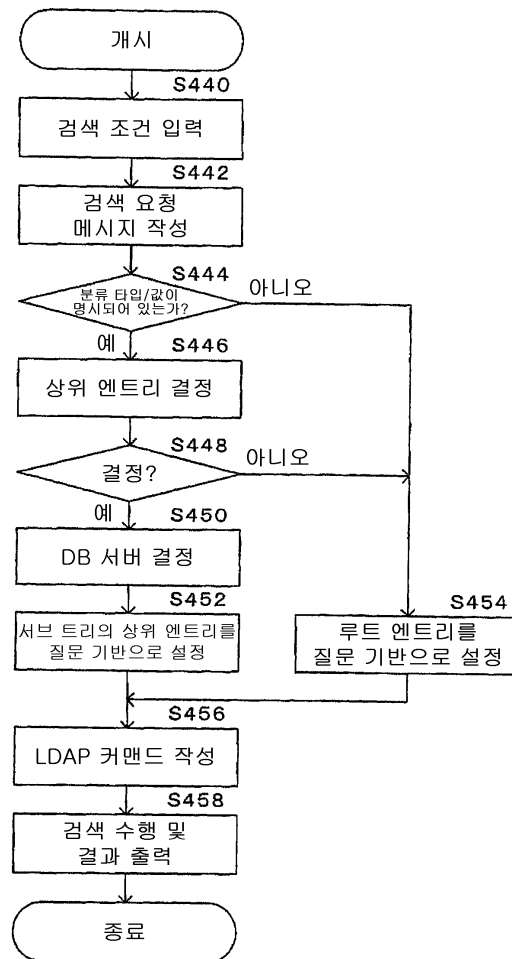


DB 프로그램 60

도면38

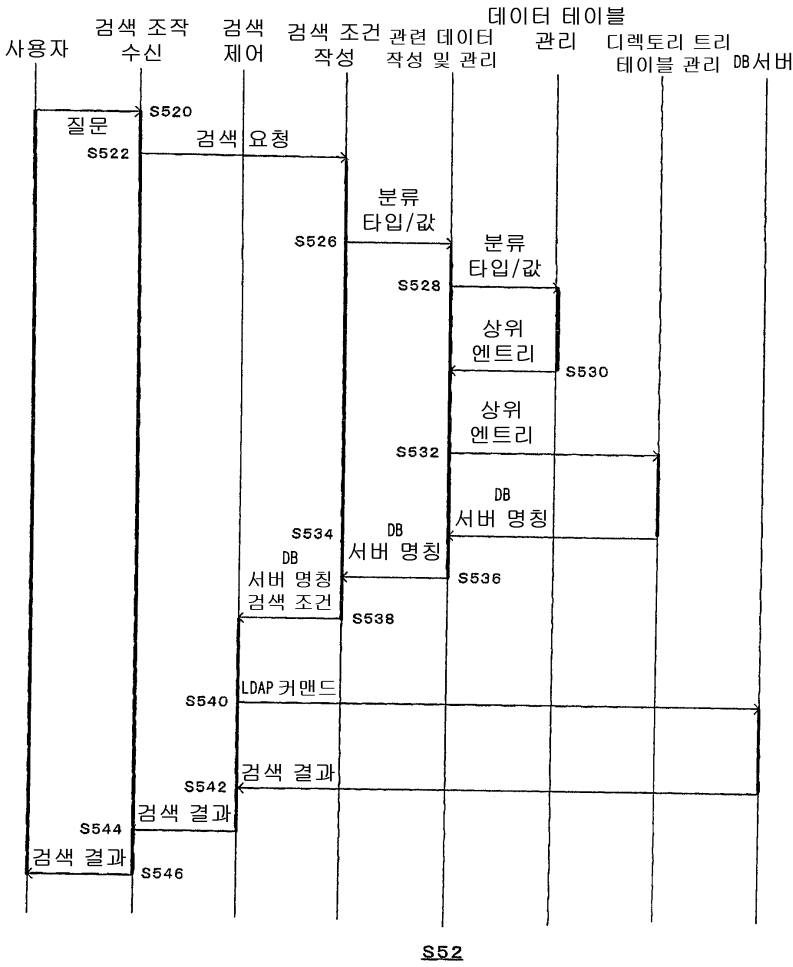


도면39

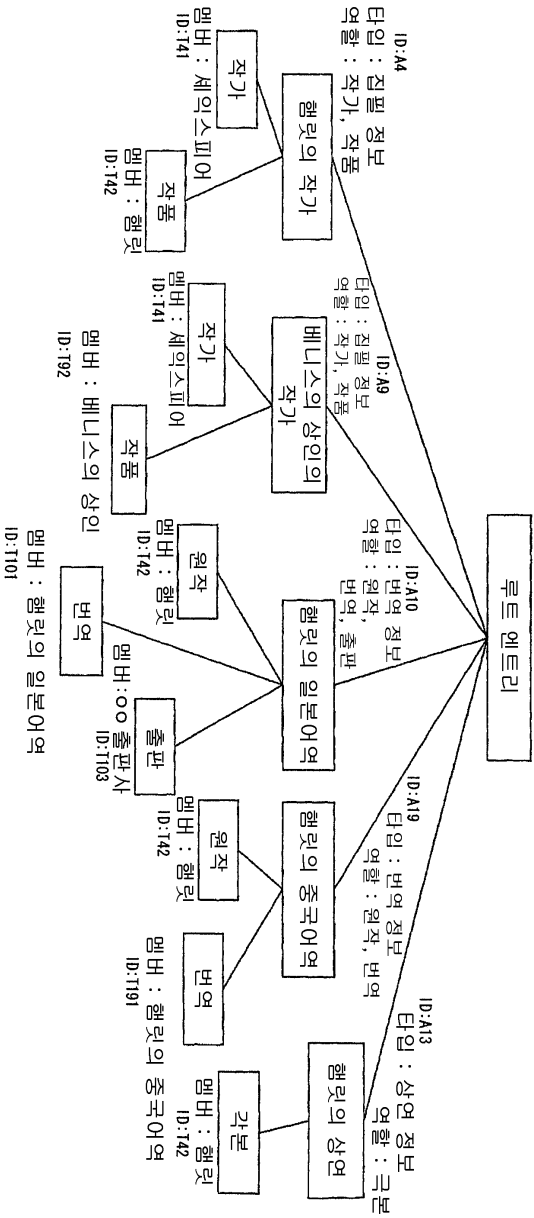


S44

도면40

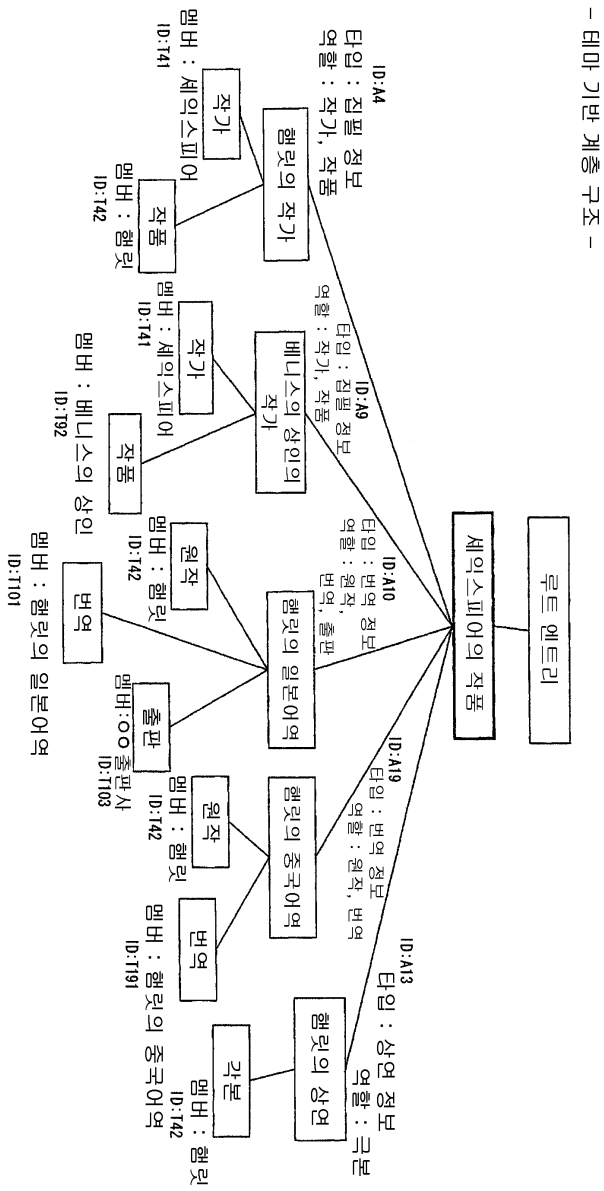


도면41

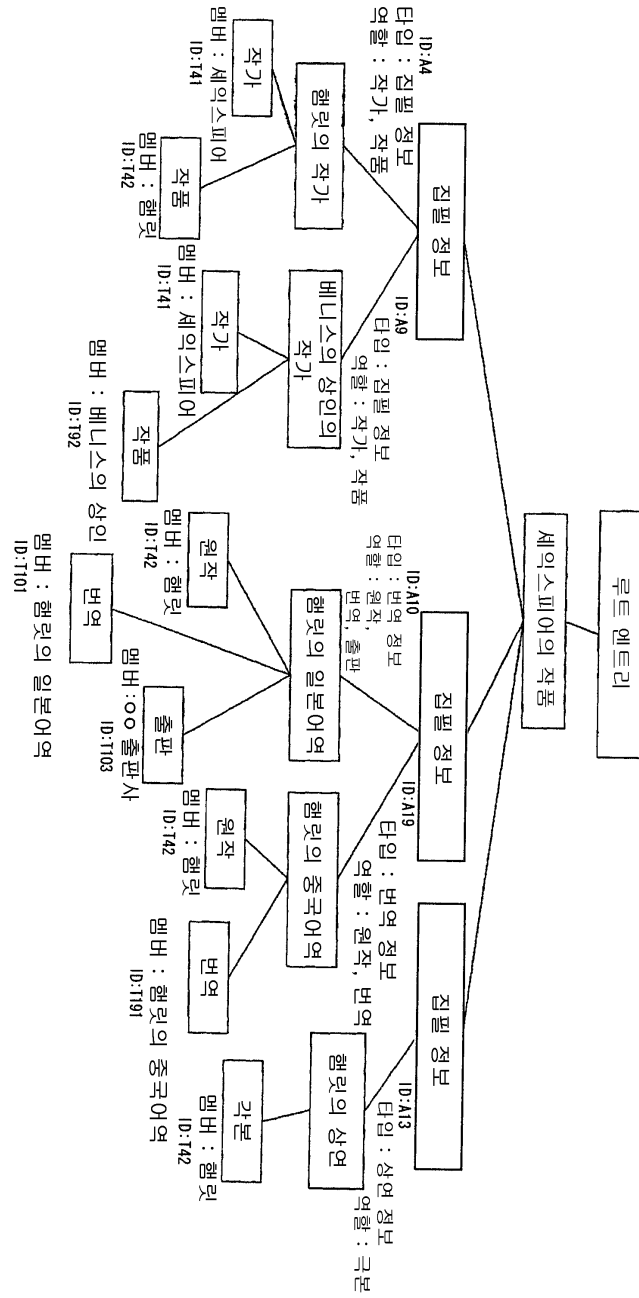


- 플랫폼 디렉토리 구조 -

도면42



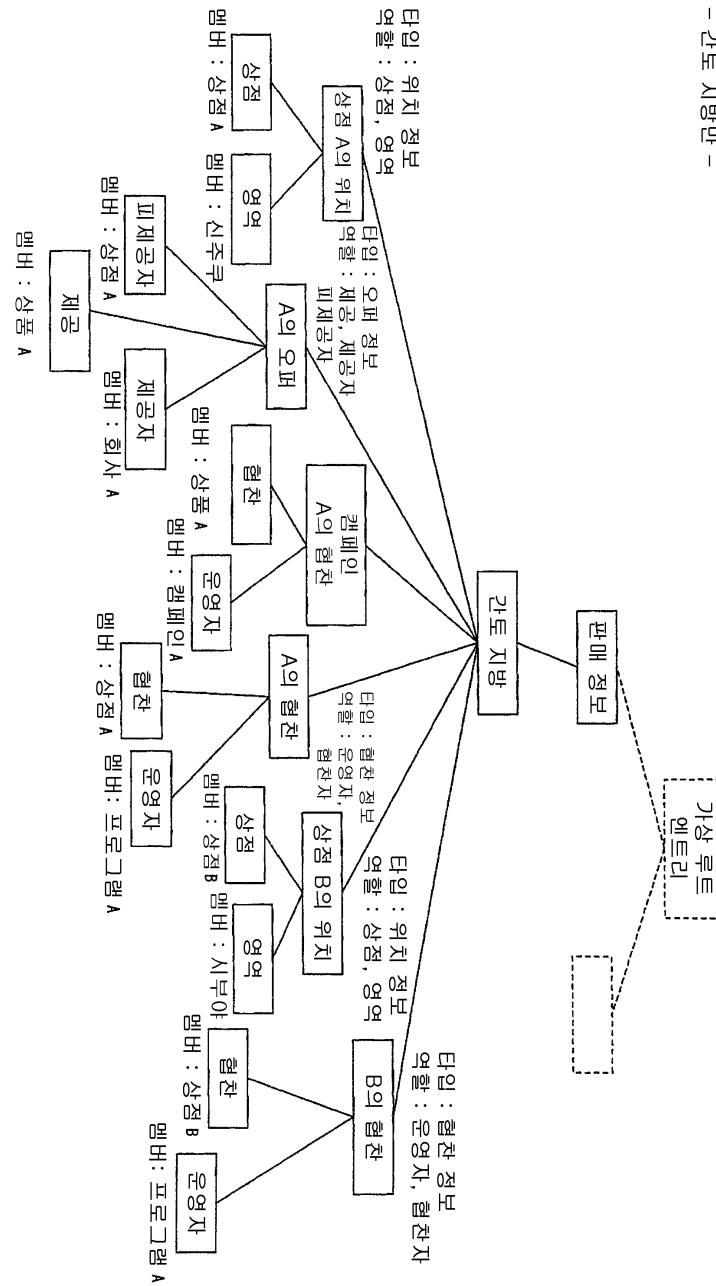
도면43



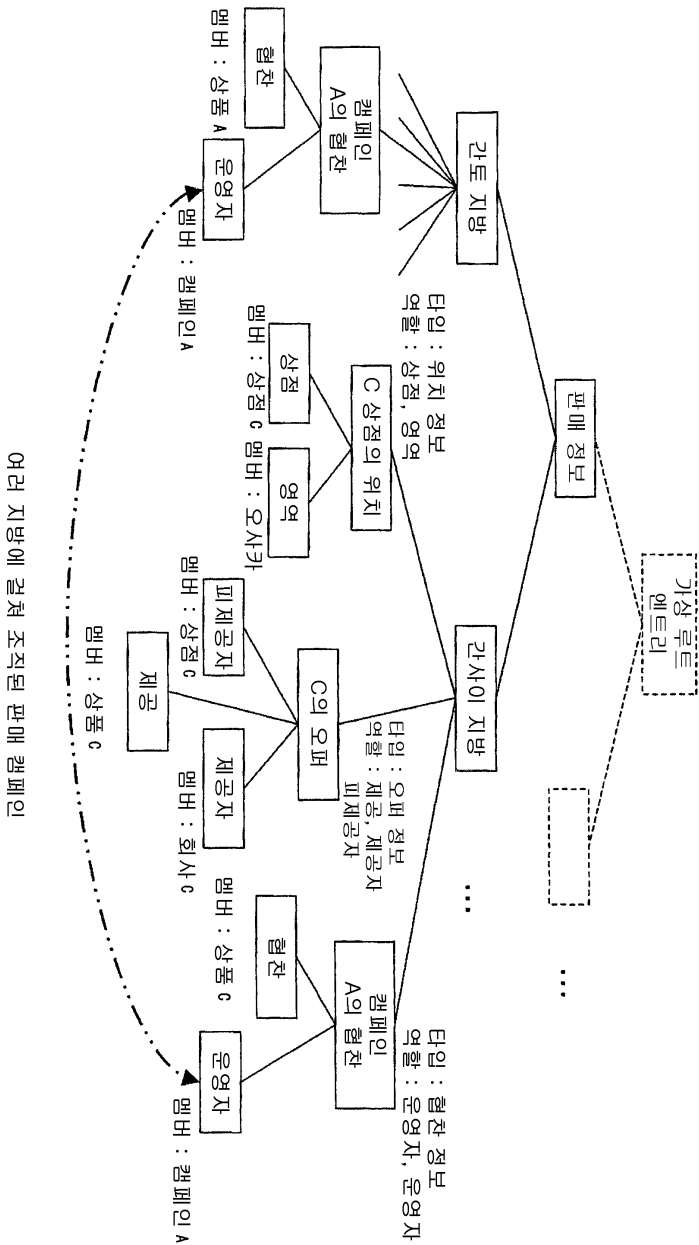
- 관련성 타입 기반 계층 구조 -

도면44

- 간토 지방만 -



도면45



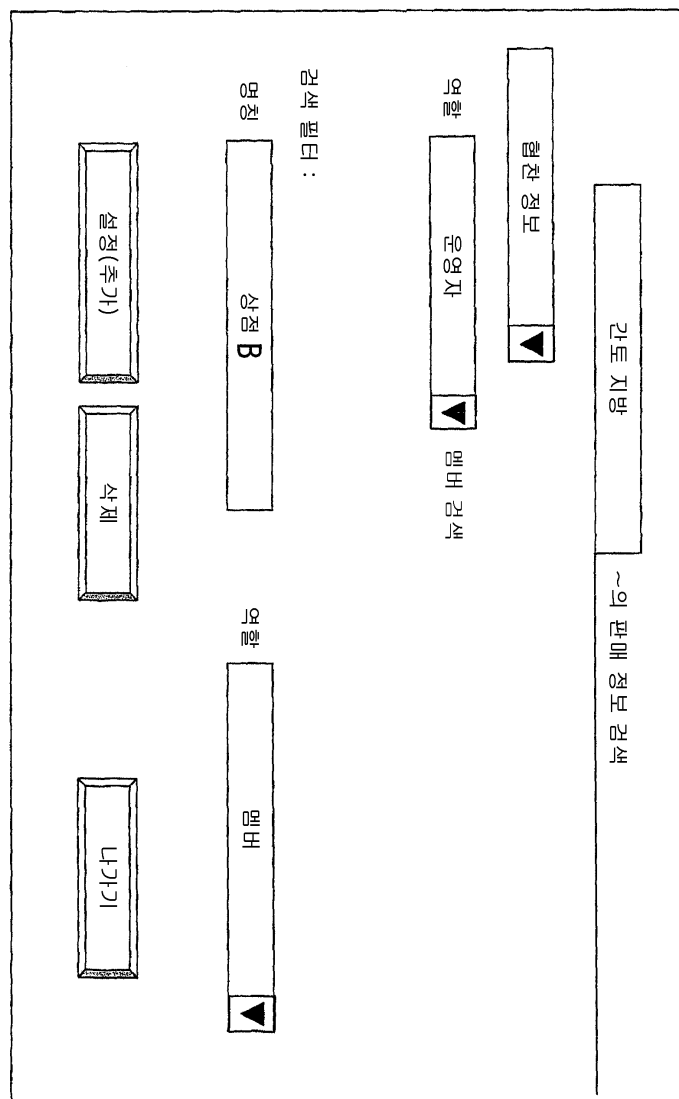
도면46

서버 명칭	상위 엔트리	리퍼링 엔트리	복제 서버
서버 A	dn_A	N/A	서버 A'
서버 B	dn_B	dn_A ₁	N/A
서버 C	dn_C	dn_B ₁	서버 C'
:	:	:	:

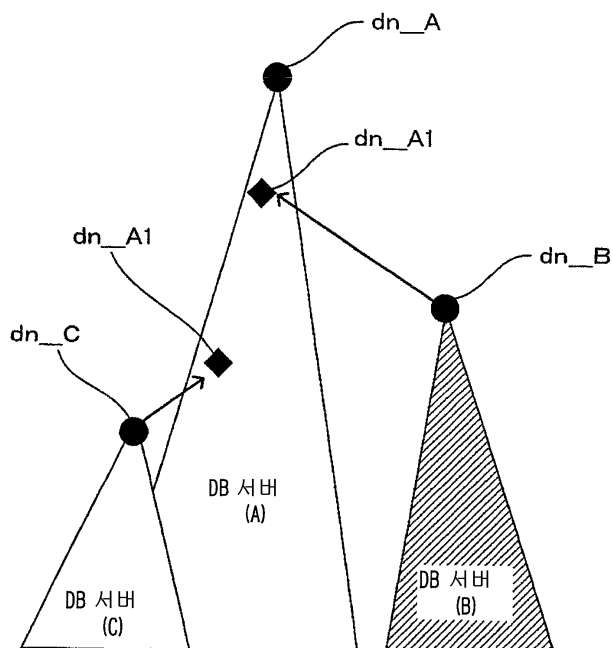
도면47

서브 트리	분류 타입	분류 값	상위 엔트리
a	카테고리	협력사 정보	dn_A
b	지방	간도 지방	dn_B
:	:	:	:

도면48



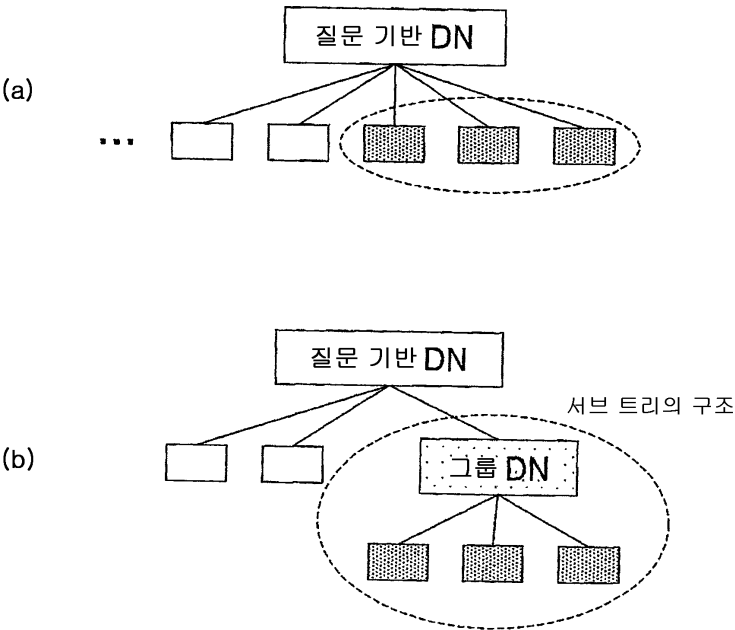
도면49



도면50

```
<searchRequest>
  <condition region=" 간토 지방 " category=" 판매 정보 " />
  <query result="res1">
    <associationType> 협찬 정보 </associationType>
    <targetRole> 운영자 </target Role>
    <memberSpec>
      <memberID> 상점 B </member ID>
      <role> 운영자 </role>
    </memberSpec>
  </query>
  <query result="res2">
    <associationType> 협찬 정보 </associationType>
    <targetRole> 운영자 </targetRole>
    <memberSpec>
      <memberID> $res1;</memberID>
      <role> 운영자 </role>
    </memberSpec>
  </query>
</searchRequest>
```

도면51

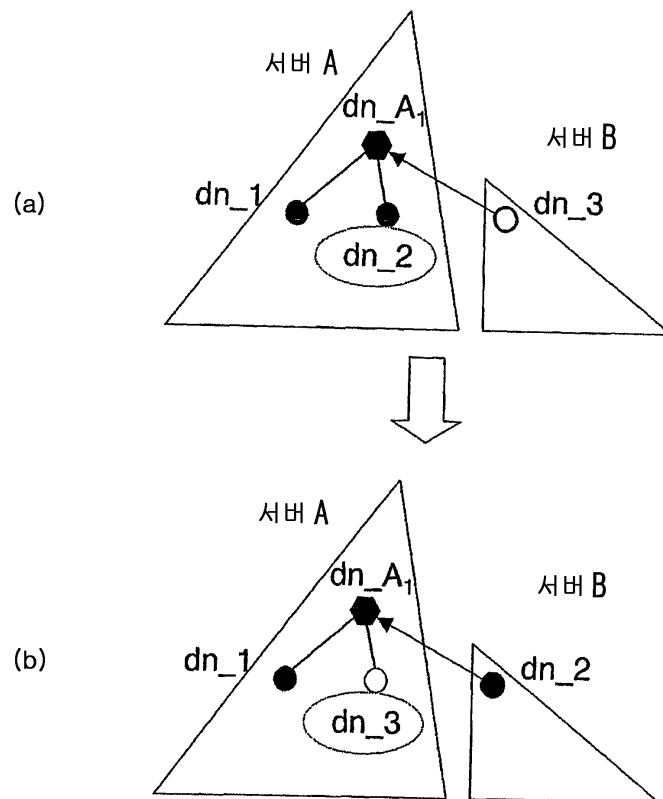


도면52

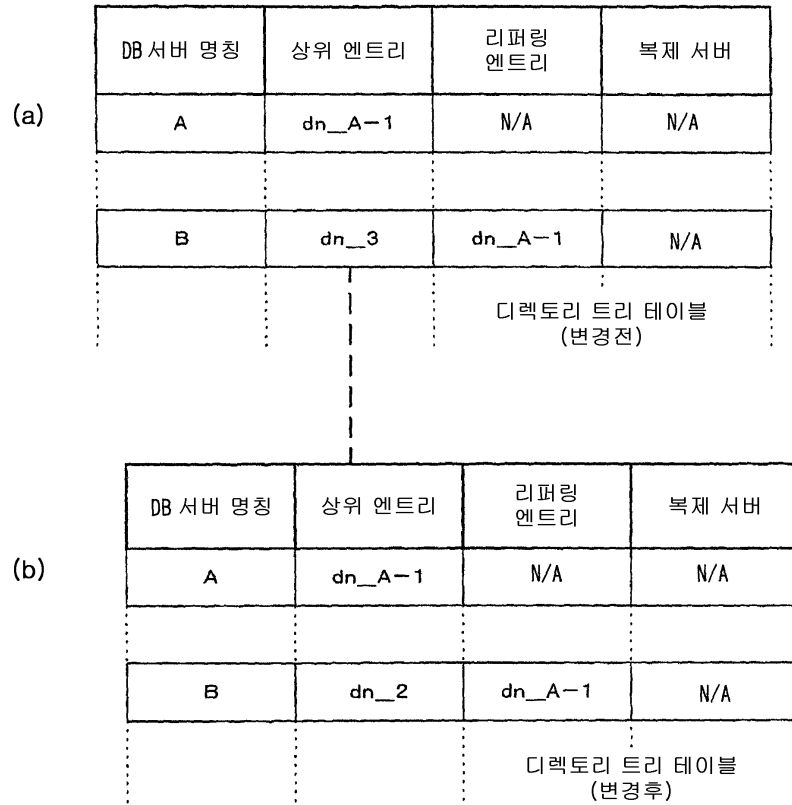
검색 조건 관리 테이블

질문 기반	필터	새로운 질문 기반
질문 기반 DN	질문 필터	그룹 DN
▪ ▪ ▪	▪ ▪ ▪	▪ ▪ ▪

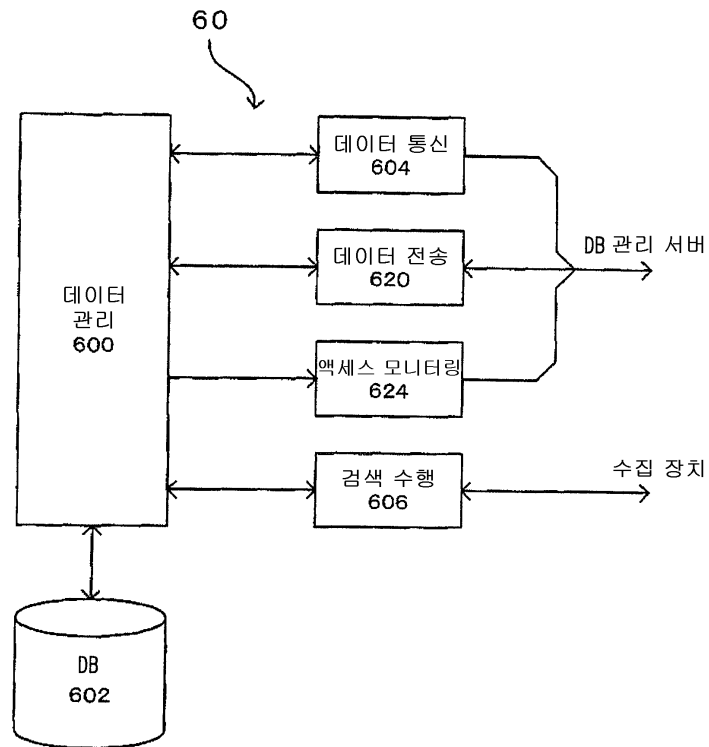
도면53



도면54

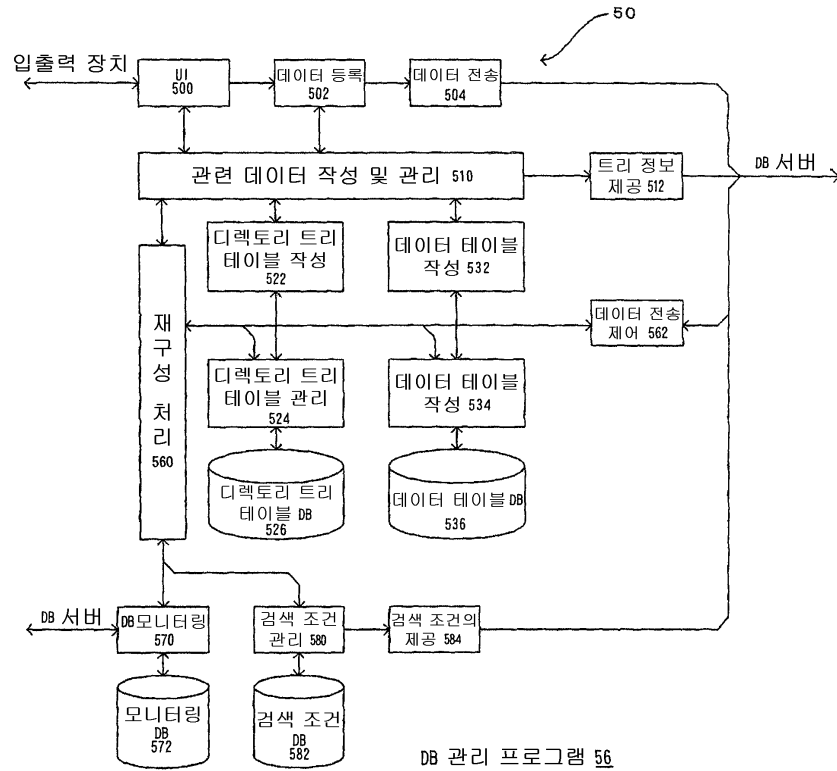


도면55

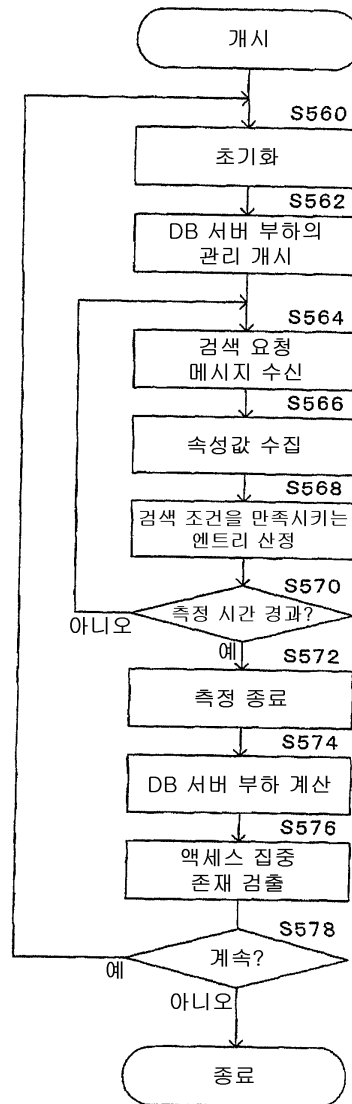


DB 프로그램 62

도면56

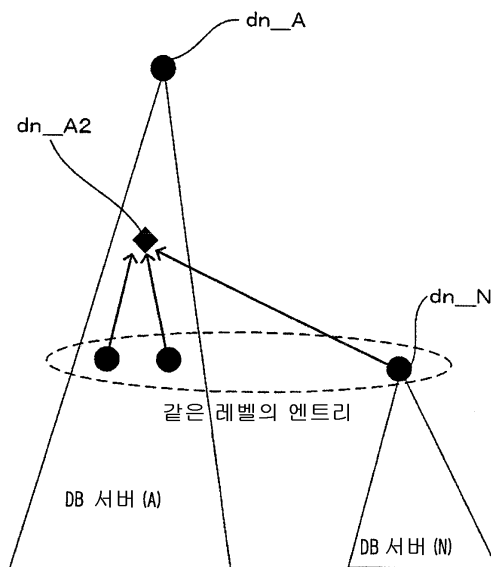


도면57



S56

도면58



도면59

