



(19)대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) 。 Int. Cl.	(45) 공고일자	2007년08월03일
G06F 17/30 (2006.01)	(11) 등록번호	10-0745442
G06F 17/00 (2006.01)	(24) 등록일자	2007년07월27일

(21) 출원번호	10-2004-7012103	(65) 공개번호	10-2004-0086346
(22) 출원일자	2004년08월05일	(43) 공개일자	2004년10월08일
심사청구일자	2005년09월14일		
번역문 제출일자	2004년08월05일		
(86) 국제출원번호	PCT/GB2003/000679	(87) 국제공개번호	WO 2003/075176
국제출원일자	2003년02월14일	국제공개일자	2003년09월11일

(30) 우선권주장      10/091,919      2002년03월05일      미국(US)

(73) 특허권자      인터내셔널 비지네스 머신즈 코포레이션  
미국 10504 뉴욕주 아몬크 뉴오차드 로드

(72) 발명자      리차드트랜달제임스  
미국 95120 캘리포니아주 산 호세 마운틴 쿠아일 서클 1141

초이데이비드문-히엔  
미국 94024 캘리포니아주 로스 알토스 브라이트 오크스 코트 1300

후타웨이  
미국 95120 캘리포니아주 산 호세 실버 캐논 드라이브 1153

리양틸리  
미국 95138 캘리포니아주 산 호세 몬아키 서클 893

넬슨케네스칼린  
미국 95023 캘리포니아주 홀리스터 던빌 웨이 6450

(74) 대리인      김원준  
김창세  
장성구

(56) 선행기술조사문헌  
WO 02/01384

심사관 : 손영태

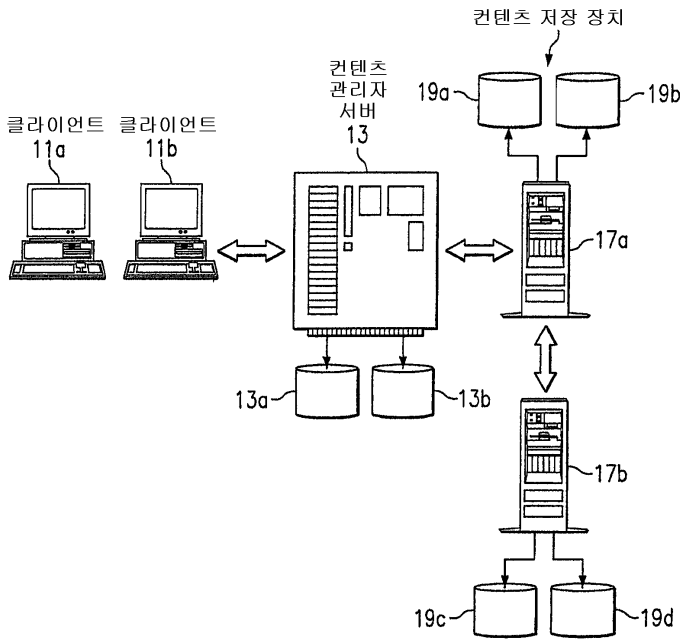
전체 청구항 수 : 총 10 항

(54) 멀티미디어 콘텐츠 관리 객체 표현 방법, 멀티미디어 콘텐츠 관리 시스템의 관리 방법, 컴퓨터 판독 가능 기록 매체, 멀티미디어 데이터 객체 아이템의 관리 및 전송 시스템 및 멀티미디어 콘텐츠 관리 시스템의 파플레이트 방법

## (57) 요약

본 발명은 멀티미디어 콘텐츠를 관리하는 방법, 시스템 및 프로그램 제품에 관한 것이다. 이들은 특정한 멀티미디어 콘텐츠 타입을 위한 하이 레벨 콘텐츠 모델(high level content model)과 데이터 엔진에 대한 매핑을 위한 로우 레벨 물리적 모델(low level physical model)의 조합으로 구성된다. 이러한 방법, 시스템 및 프로그램 제품은 멀티미디어 콘텐츠 관리 객체를 표현하고, 관계형 데이터베이스(relational database) 내에서 객체를 관리하는 것을 포함한다. 관계형 데이터베이스는 복수의 관계형 데이터베이스 테이블 내에 데이터를 표현하기에 적합하고, 각각의 관계형 데이터베이스 테이블은 복수의 열을 갖는 적어도 하나의 행을 구비한다. 콘텐츠는, 객체를 아이템으로 하여 제 1 관계형 데이터베이스 테이블 내의 하나의 행에 연관시키고, 객체의 추가적인 구성 요소가 존재한다면 이 추가적인 구성 요소를 추가적인 관계형 데이터베이스 테이블에 연관시키는 것에 의해 관리된다.

## 대표도



## 특허청구의 범위

### 청구항 1.

각각 복수의 열(column)을 갖는 적어도 하나의 행(row)을 구비하는 하나 이상의 관계형 데이터베이스 테이블 내에 멀티미디어 콘텐츠 관리 데이터를 표현하기에 적합한 관계형 데이터베이스에서 멀티미디어 콘텐츠 관리 객체를 아이템(item)으로서 표현하는 방법으로서,

상기 콘텐츠 관리 객체의 루트 구성 요소(root component)를 제 1 관계형 데이터베이스 테이블 내의 행과 연관시키는 단계와,

상기 루트 구성 요소의 속성을 상기 제 1 관계형 데이터베이스 테이블의 대응하는 열과 연관시키는 단계와,

상기 콘텐츠 관리 객체의 추가적인 구성 요소가 존재한다면, 그 콘텐츠 관리 객체의 추가적인 구성 요소를 추가적인 관계형 데이터베이스 테이블 내의 행과 연관시키는 단계와,

상기 관계형 데이터베이스 테이블에 의해 정의된 상기 아이템을 이용하여 상이한 애플리케이션에 각각 대응하는 복수의 하이 레벨 콘텐츠 관리 데이터 모델을 구성하는 단계

를 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

## 청구항 2.

제 1 항에 있어서,

상기 추가적인 구성 요소의 각각은, 루트 구성 요소의 자녀 구성 요소(child component) 또는 다른 구성 요소의 자녀 구성 요소를 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

## 청구항 3.

청구항 3은(는) 설정등록료 납부시 포기되었습니다.

제 2 항에 있어서,

자녀 구성 요소에 있어서의 외부 키(foreign key)를 이용하여 그 부모 구성 요소를 참조하는 단계를 더 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

## 청구항 4.

청구항 4은(는) 설정등록료 납부시 포기되었습니다.

제 1 항에 있어서,

상기 속성은 별개의 리포지터리(repository) 내에 저장된 데이터 리소스에 대한 포인터(pointer)를 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

## 청구항 5.

청구항 5은(는) 설정등록료 납부시 포기되었습니다.

제 1 항에 있어서,

상기 속성은 다른 콘텐츠 관리 객체에 대한 포인터를 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

## 청구항 6.

청구항 6은(는) 설정등록료 납부시 포기되었습니다.

제 1 항에 있어서,

상기 테이블 내의 행은 소스 아이템과 타겟 아이템 사이의 링크를 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

## 청구항 7.

멀티미디어 콘텐츠 데이터의 하이 레벨 콘텐츠 모델 및 로우 레벨 물리적 모델을 포함하고, 상기 로우 레벨 물리적 모델은 데이터 엔진에 대한 매핑을 제공하는 데이터베이스에서 멀티미디어 콘텐츠 관리 객체를 표현하는 방법으로서,

상기 로우 레벨 물리적 모델 내에 멀티미디어 콘텐츠 데이터 메타데이터(metadata) 및 스키마(schema)를 입력하는 단계와,

상기 메타데이터 및 스키마를 상기 데이터 엔진에 대해 매핑하는 단계를 포함하며,  
상기 로우 레벨 물리적 모델은 복수의 하이 레벨 콘텐츠 모델을 지원하고,  
상기 하이 레벨 콘텐츠 모델은 상이한 애플리케이션에 각각 대응하는  
멀티미디어 콘텐츠 관리 객체의 표현 방법.

#### 청구항 8.

청구항 8은(는) 설정등록료 납부시 포기되었습니다.

제 7 항에 있어서,

상기 하이 레벨 콘텐츠 모델은 하나 이상의 데이터 구조 및 제약에 관한 표현을 구체화하는 애플리케이션 프로그램 인터페이스를 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

#### 청구항 9.

청구항 9은(는) 설정등록료 납부시 포기되었습니다.

제 7 항에 있어서,

상기 하이 레벨 콘텐츠 모델은 복수의 콘텐츠 애플리케이션 요건을 지원하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

#### 청구항 10.

청구항 10은(는) 설정등록료 납부시 포기되었습니다.

제 7 항에 있어서,

상기 로우 레벨 물리적 모델은 확장 가능한 멀티미디어 콘텐츠 관리 객체의 표현 방법.

#### 청구항 11.

제 7 항에 있어서,

추가적인 하이 레벨 콘텐츠 모델을 추가하는 단계를 더 포함하는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

#### 청구항 12.

제 7 항에 있어서,

상기 데이터 엔진은 관계형 데이터베이스 관리 시스템, 객체 지향형 데이터베이스 관리 시스템, 객체 관계형 데이터베이스 관리 시스템 및 XML 데이터 리포지터리(repositories)로 이루어지는 그룹으로부터 선택되는 멀티미디어 콘텐츠 관리 객체의 표현 방법.

#### 청구항 13.

멀티미디어 객체 구성 요소 및 멀티미디어 객체 속성을 포함하는 멀티미디어 콘텐츠 관리 객체를 구비하는 멀티미디어 콘텐츠 관리 시스템을 관리하는 방법-상기 멀티미디어 콘텐츠 관리 객체는, 각각 복수의 열(column)을 갖는 적어도 하나의 행(row)을 구비하는 하나 이상의 관계형 데이터베이스 테이블 내에 멀티미디어 콘텐츠 관리 데이터를 표현하기에 적합한 관계형 데이터베이스에서 아이템(item)으로서 표현됨-으로서,

상기 방법이 청구 범위 제 1 항 내지 제 6 항 중 어느 한 항에 기재된 방법의 각각의 단계를 수행하는

멀티미디어 콘텐츠 관리 시스템의 관리 방법.

#### 청구항 14.

하나 이상의 멀티미디어 데이터 객체 아이템을 관리하고, 멀티미디어 데이터 객체 콘텐츠 서버를 통해서 멀티미디어 데이터 객체 콘텐츠 리포지터리로부터 클라이언트로 멀티미디어 콘텐츠 관리 객체의 상기 멀티미디어 데이터 객체 아이템을 전송하는 시스템-상기 멀티미디어 데이터 객체 아이템은 각각 멀티미디어 데이터 객체 속성 및 구성 요소를 포함함-으로서,

상기 멀티미디어 데이터 객체 콘텐츠 서버는,

청구 범위 제 1 항 내지 제 6 항 중 어느 한 항에 기재된 방법의 각각의 단계를 수행하도록 제어되고 구성되는

멀티미디어 데이터 객체 아이템의 관리 및 전송 시스템.

#### 청구항 15.

데이터베이스에서 멀티미디어 콘텐츠 관리 객체를 표현함으로써, 멀티미디어 콘텐츠 관리 시스템을 관리하기 위한 하나 이상의 컴퓨터를 구비하는 컴퓨터 시스템을 제어하고 구성할 수 있는 컴퓨터 프로그램을 구비한 컴퓨터 판독 가능 기록 매체로서,

상기 컴퓨터 시스템 상에서 상기 컴퓨터 프로그램이 실행될 때, 상기 시스템이 청구 범위 제 7 항 내지 제 12 항 중 어느 한 항에 기재된 방법의 각각의 단계를 수행하게 하는 프로그램 코드

를 포함하는 컴퓨터 판독 가능 기록 매체.

#### 청구항 16.

멀티미디어 객체 구성 요소 및 멀티미디어 객체 속성을 갖는 멀티미디어 콘텐츠 관리 객체를 포함하는 멀티미디어 콘텐츠 관리 시스템을 관리하기 위한 하나 이상의 컴퓨터를 구비하는 컴퓨터 시스템을 제어하고 구성할 수 있는 컴퓨터 프로그램을 구비한 컴퓨터 판독 가능 기록 매체로서,

상기 컴퓨터 시스템 상에서 상기 컴퓨터 프로그램이 실행될 때, 상기 시스템이 청구 범위 제 1 항 내지 제 4 항 중 어느 한 항에 기재된 방법의 각각의 단계를 수행하게 하는 프로그램 코드

를 포함하는 컴퓨터 판독 가능 기록 매체.

#### 청구항 17.

청구항 17은(는) 설정등록료 납부시 포기되었습니다.

제 16 항에 있어서,

컨텐츠 스키마 및 메타데이터를 갖는 멀티미디어 컨텐츠 관리 시스템을 파플레이트(populate)하는 프로그램 코드를 더 구비하며, 컨텐츠 아이템에 관해 질의를 사용자에게 제공하고, 최종 사용자의 응답에 근거하여, 상기 컨텐츠 아이템에 관해 후속의 질의를 제공하며, 최종 사용자의 추가적인 응답에 근거하여, 하위 구성 요소 및 아이템의 속성을 결정하는 컴퓨터 판독 가능 기록 매체.

## 청구항 18.

컨텐츠 스키마 및 메타데이터를 갖는 멀티미디어 컨텐츠 관리 시스템을 파플레이트하는 방법으로서,

컨텐츠 아이템에 관해 질의를 사용자에게 제공하고, 최종 사용자의 응답에 근거하여, 상기 컨텐츠 아이템에 관해 후속의 질의를 제공하는 단계와,

최종 사용자의 추가적인 응답에 근거하여, 하위 구성 요소 및 아이템의 속성을 결정하는 단계와,

청구 범위 제 1 항 내지 제 6 항 중 어느 한 항에 기재된 방법의 각각의 단계

를 포함하는 멀티미디어 컨텐츠 관리 시스템의 파플레이트 방법

## 명세서

### 기술분야

본 발명은 멀티미디어 데이터 구조 모델을 포함하는 데이터 구조 모델에 관련된 것으로, 컨텐츠 관리 애플리케이션에 대한 액세스, 복원 및 데이터 변환을 용이하게 한다. 관계형, 네트워크형, 계층형(hierarchical) 및 개체-관계형(entity-relationship) 모델을 포함하는 데이터 구조 모델은, 데이터 및 파일, 그 중에서도 그래픽, 텍스트 및 웹 페이지 파일을 체계화하고, 상호 연관시키며, 액세스하기 위한 용도로 사용된다.

### 배경기술

멀티미디어 데이터 관리 및 전송은 복잡한, 컴퓨터 및 메모리 집약형 작업이다. 본 명세서에서 사용된 바와 같은 "멀티미디어" 데이터는 음향, 그래픽, 동화상(animation), 텍스트 및 비디오 데이터가 복합된 것을 지칭하는 것으로, 저장 및 복원과 연관된 시스템에 연결될 수 있고, 대화형(interactive)일 수 있으며, 다른 매체에 연결될 수도 있다.

멀티미디어 컨텐츠 관리 시스템은 구조화 데이터뿐만 아니라 비구조화 데이터를 포함하여 대용량의 이종 데이터(disparate data)를 관리해야 한다. 구조화 데이터는 때때로 비구조화 데이터를 나타내는 "메타데이터(metadata)"를 포함하므로, 구조화 데이터는 여러 방식, 즉 탐색, 복원 및 관리의 방식으로 체계화되고, 그룹화되며, 상관(correlated)될 수 있다. 컨텐츠 관리 시스템 내에서는, 때때로 관계형 데이터베이스 관리 시스템(relational database management system: RDBMS)을 사용하여 이들 구조화 데이터를 관리함으로써 시판 중인 RDBMS 데이터 관리 기법을 적용할 수 있다. 그러나, 비구조화 데이터, 즉, 멀티미디어 컨텐츠의 타입은 애플리케이션마다 상이하다.

데이터의 타입(스키마(schema), 메타데이터, 적절한 확장자 등을 포함함)은 애플리케이션마다 상이하고, 컨텐츠 데이터 모델(스트리밍(streaming), 그래픽, 영상, 비디오, 오디오, 텍스트, 숫자 등)마다 상이하기 때문에, 각각의 애플리케이션과 데이터 모델의 메타데이터 및 스키마 데이터베이스를 지원하기 위해 서로 다른 데이터베이스가 필요하다. 지금까지, 하나의 범용(general-purpose) 컨텐츠 관리 시스템을 가지고 서로 다른 컨텐츠 데이터 모델을 사용하여 서로 다른 컨텐츠 관리 애플리케이션을 관리하는 것은 불가능했다. 이러한 시스템은 사전에 설계된 즉, "일률적인 방식(one size fits all)"의 데이터베이스 관리 시스템을 구비할 수 없다. 그 대신에, 컨텐츠 관리 시스템은 애플리케이션 개발자가 요구되는 구조화 데이터를 기술하게 하여 해당 애플리케이션을 위한 적합한 설계를 갖는 데이터베이스를 생성할 수 있게 하는 능력을 구비해야 한다. 이 기술(description)은 일반적으로 API(application programming interface) 또는 GUI(graphical user interface)를 이용하여 컨텐츠 관리 시스템에 의해 지원되는 하이 레벨 데이터 모델(high-level data model)의 추상화

(abstraction)로 통상적으로 표현된다. 이러한 하이 레벨 데이터 모델은 하나 이상의 애플리케이션을 지원할 수 있게 하는 유연성을 제공하지만, 해당되는 특정한 하이 레벨 모델로 표현될 수 있는 애플리케이션으로 적용 범위가 한정되어 버리기도 한다.

데이터 모델은 시스템의 한계뿐만 아니라 시스템의 잠재적 능력을 결정하기 때문에 콘텐츠 관리 시스템의 필수적인 특징이다. 그러나, 때때로 데이터 모델은 어느 정도 확장될 수 있지만, 이러한 확장은 일반적으로 매우 제한되어 있고, 상이한 패러다임, 특히 양립할 수 없거나 충돌되는 개념을 갖는 패러다임을 포착할 수 없다.

그러므로, 광범위한 이중 콘텐츠 데이터 모델을 용이하게 지원하고, 시스템 또는 프로그램 제품 내에서 용이하게 구현될 수 있는 확장 가능 및 크기 조절 가능 데이터 모델에 대한 필요성이 분명하게 존재한다.

확장 가능 및 크기 조절 가능 콘텐츠 관리형 RDBMS에 대하여 이중 콘텐츠 스키마 및 메타데이터의 관계형 데이터베이스를 생성하고 파플레이트(populate)하기 위한 툴(tool), 툴 키트(tool kit), 툴 세트(tool set) 또는 위저드(wizard)에 대한 추가적인 필요성이 존재한다.

### 발명의 상세한 설명

따라서, 본 발명은 데이터베이스 내에 멀티미디어 콘텐츠 관리 객체를 표현하는 방법을 제공한다. 본 발명의 방법에 따르면, 멀티미디어 콘텐츠 데이터로 이루어진 하나 이상의 하이 레벨 콘텐츠 모델 및 로우 레벨 물리적 모델을 포함하는 데이터베이스를 이용한다. 로우 레벨 물리적 모델은 데이터 엔진(data engine)에 대한 매핑(mapping)을 제공한다. 이 방법은 로우 레벨 물리적 표현 내에 멀티미디어 데이터 메타데이터 및 스키마를 입력하는 단계와, 데이터 엔진에 대해 메타데이터 및 스키마를 매핑하는 단계를 포함한다. 데이터 엔진은 관계형 데이터베이스 관리 시스템, 객체 지향형 데이터베이스 관리 시스템 또는 XML 데이터 리포지터리(repository)인 것이 바람직하다.

이 방법은 데이터 엔진에 대한 매핑을 제공하는 것이 바람직하고, 하나 이상의 관계형 데이터베이스 테이블 내에 멀티미디어 콘텐츠 관리 데이터를 표현하기에 적합한 관계형 데이터베이스 내의 아이템으로서 멀티미디어 콘텐츠 관리 객체를 표현하는 방법을 더 포함한다. 각각의 관계형 데이터베이스 테이블은 적어도 하나의 행과 복수의 열을 갖는다. 이 방법은 객체의 루트 구성 요소(root component)를 아이템으로서 제 1 관계형 데이터베이스 테이블 내의 행에 대해 연관시키는 단계와, 루트 구성 요소의 속성을 제 1 관계형 데이터베이스 테이블의 대응되는 열에 연관시키는 단계와, 객체의 추가적인 구성 요소들(존재하는 경우)을 추가적인 관계형 데이터베이스 테이블 내의 행들에 연관시키는 단계와, 이러한 아이템을 복수의 하이 레벨 콘텐츠 관리 데이터 모델을 구성하기 위한 구축 블록으로서 사용하는 단계를 포함한다.

따라서, 본 발명은 또한 비휘발성 메모리 저장 장치를 구비하는 데이터 처리 시스템 상에서 실행될 때, 시스템이 상술된 방법을 수행하게 하는 인스트럭션을 포함하는 컴퓨터 프로그램 제품을 제공한다.

따라서, 본 발명은 또한 멀티미디어 데이터 객체 아이템을 관리하고, 멀티미디어 데이터 객체 콘텐츠 서버를 통해서 멀티미디어 데이터 객체 콘텐츠 리포지터리로부터 클라이언트로 멀티미디어 데이터 객체 아이템을 전송하는 시스템을 제공한다. 멀티미디어 데이터 객체는 멀티미디어 데이터 객체 아이템을 포함하고, 각각의 멀티미디어 데이터 객체 아이템은 멀티미디어 데이터 객체 속성과 구성 요소를 포함한다. 멀티미디어 데이터 객체 콘텐츠 서버는 객체의 루트 구성 요소를 제 1 관계형 데이터베이스 테이블 내의 하나의 행에 연관시키고, 루트 구성 요소의 속성을 제 1 관계형 데이터베이스 테이블의 대응되는 열에 연관시키며, 객체의 추가적인 구성 요소들(존재하는 경우)을 추가적인 관계형 데이터베이스 테이블 내의 대응되는 행들에 연관시키도록 제어되고 구성된다. 아이템은 복수의 하이 레벨 데이터 모델을 형성하기 위한 구축 블록으로서 사용된다.

본 발명의 바람직한 실시예는 이하의 도면을 참조하여 오직 예시의 방법으로만 세부적으로 설명될 것이다.

### 실시예

도 1에 도시된 바와 같이, 멀티미디어 시스템은 다수의 클라이언트(11a, 11b), 서버(13) 및 하나 이상의 콘텐츠 서버(17a, 17b)(상기 하나 이상의 콘텐츠 서버는 각각 하나 이상의 콘텐츠 저장 장치(19a, 19b, 19c, 19d)를 가짐)를 구비하는 복합 클라이언트-서버 시스템이다. 콘텐츠 저장 장치는, 콘텐츠에 대한 메타데이터 등과 같은 구조화 데이터 또는 텍스트 데이터, 이미지 데이터, 스트리밍 오디오 또는 스트리밍 비디오 등과 같은 비구조화 데이터를 포함할 수 있다는 점을 유의해야 한다. 이러한 데이터는 다양한 이중의 형태를 가질 수 있고, 전형적으로도 그러하다.

웹 서버를 포함할 수 있는 서버(13)는 또한 콘텐츠 관리자 서버를 포함한다. 콘텐츠 관리자 서버는 본 명세서에서 설명된 콘텐츠 관리 RDBMS의 리포지터리이다. 본 발명은 관계형 데이터베이스 관리 시스템과 관련하여 설명되었으나, 그 대신에 객체 지향형 데이터베이스 관리 시스템 등과 같은 데이터 리포지터리 또는 XML 데이터 리포지터리를 사용할 수도 있다.

하이 레벨 데이터 모델 내에서 하나 이상의 콘텐츠 데이터 모델을 수용하기 위해서는, 확장 가능 콘텐츠 관리 시스템을 사용하여 적절한 메타데이터 및 스키마를 도입할 수 있고, 새로운 API를 사용하여 새로운 하이 레벨 데이터 모델을 지원할 수 있다. 이 확장 가능성은 물리적 데이터 모델을 지원하고, 행 내에서 대응되는 행과 열(속성)을 가지고 콘텐츠 및 콘텐츠 메타데이터의 RDBMS 테이블의 계층적 구조를 통해서 아이템, 구성 요소, 하부 구성 요소 및 속성으로 구현된 로우 레벨 하부 구조 주위에 형성되는 아키텍처에 의해서 가능해진다. 아키텍처와, 그와 연관된 하부 구조 및 물리적 데이터 모델은 하이 레벨 데이터 모델을 지원하기 위한 구축 블록으로 이용된다. 이 물리적 데이터 모델은 애플리케이션 지정 시맨틱(application-specific semantics)과 무관하고, RDBMS 기능을 활용하도록 설계된 관계형 데이터베이스에 대해 직접적이고 효과적으로 매핑하며, 시간에 따라 강화되어 기능의 확장 및/또는 새로운 RDBMS 기능의 활용이 가능해진다.

이러한 물리적 데이터 모델을 위한 설계 포인트는 애플리케이션 개발자들에게 공개되어 있는 하이 레벨 데이터 모델과는 다르다. 하이 레벨 데이터 모델은 그 논리적 측면(애플리케이션의 클래스와 관련된 그의 유용성 및 표현성)에 크게 중점을 두는 반면에, 물리적 데이터 모델은 여러 콘텐츠 관리 애플리케이션에게 광범위하게 이용될 수 있는 일반적이고, 기본적인 기능을 제공하기 위해서 그의 물리적 측면(저장 액세스 효율 및 RDBMS 기능을 활용하는 능력)에 크게 중점을 둔다. 이러한 물리적 데이터 모델 또는 그의 임의의 부분은 하이 레벨 데이터 모델을 통해 사용자에게 노출될 수도 있고 그렇지 않을 수도 있다.

개념적으로, 도 2에 도시된 바와 같이, 본 발명에 따른 방법, 시스템 및 프로그램 제품은 다양한, 오픈 세트의 콘텐츠 애플리케이션 요건과 복수의 하이 레벨 데이터 모델을 지원할 수 있는 하이 레벨 콘텐츠 모델과, 새로운 기법, 새로운 표준 및 새로운 요건을 수용할 수 있도록 확장 가능하고, 데이터 엔진(관계형 데이터베이스 관리 시스템 또는 객체 지향형 데이터베이스 관리 시스템 등과 같은 데이터베이스 관리 시스템-이들은 예로서 제시되었으며, 이것으로 제한되지 않음-)에 대한 효과적인 매핑을 제공하는 데이터 콘텐츠의 로우 레벨 물리적 모델을 활용한다. 각각의 하이 레벨 데이터 모델은 데이터 구조 및 제약에 대한 표현을 포함하는 API(Application Program Interface)의 형태를 갖는다는 것을 유의하라. 또한, 로우 레벨 데이터 모델 또는 하부 구조(25)는 다수의 하이 레벨 모델(21, 22, 23)을 지원할 수 있다는 점을 유의해야 한다. 본 발명에 따른 확장 가능 및 크기 조절 가능 콘텐츠 관리 시스템은 일반적인 로우 레벨 하부 구조(25), 즉 물리적 데이터 모델-이 물리적 데이터 모델은 관계형 데이터베이스 관리 시스템(RDBMS)(27)에 의해서 지원됨-에 의해 지원되는 이종의 하이 레벨 데이터 모델(21, 22, 23)을 지원하는 하나 이상의 API를 포함한다.

도 3, 도 4, 도 5, 도 6은 RDBMS 환경 내에서 물리적 모델과 그에 대한 표현 사이의 관계를 나타낸다. 관계형 데이터베이스에서, 데이터는 테이블로 표현된다. 관계형 데이터베이스 테이블의 구축 블록은 행과 열(속성)로 이루어지는 테이블이다. 특히, 테이블은 데이터 값으로 이루어진 0개 이상의 행과 함께 열 제목(column headings)을 나타내는 하나의 행으로 구성된다. 주어진 테이블에 있어서, (1) 열 제목은 하나 이상의 열을 지정하고, (2) 각각의 데이터 행은 열 제목 행 내에 지정된 각 열마다에 대해 정확하게 하나의 값(또는 널 값(null value))을 포함한다.

본 발명에 따른 콘텐츠 관리 시스템의 물리적 계층에 의해 관리되는 콘텐츠의 주요한 독립(standalone) 단위는 "아이템"이다. 도 3은 아이템 행(32a, 32b)을 구비하는 루트 구성 요소(31)를 포함하는 아이템 데이터베이스(30)를 도시한다. 아이템은 구성 요소(35)의 계층적 구조 및 속성(33a, 33b, 33c, 33d, 34a, 34b, 34c)을 갖는 하나 이상의 하부 구성 요소 계층(38a, 38b)으로 구성되고, 고유의 식별자인 "아이템 ID"를 갖는다.

각각의 구성 요소(35)는 부모 구성 요소(parent component)에 대한 참조(36a)를 포함한다. 참조는 외부 키(foreign key)를 통해 이루어진다.

하부 구성 요소 테이블(38a, 38b)은 다른 하부 구성 요소 또는 속성을 참조할 수 있다. 하부 구성 요소 테이블(38b)은 콘텐츠 데이터 리포지터리(39a, 39b)에 대한 참조 포인트로서 도시되었다.

도 4는 도 3에 도시된 모델에 대한 보다 일반적이고 보다 높은 계층의 도면을 나타낸다. 하이 레벨 데이터 모델을 통해서, 또는 소정의 애플리케이션 내에서, "아이템", 다시 말해, 멀티미디어 콘텐츠 관리 객체 아이템을 사용하여 실제 대상을 문서, 폴더, 미디어 에셋(media asset), 또는 그 외의 리얼-월드 콘텐츠 개체(real-world content entity)로서 나타낼 수 있다. 도 4는 우측에 있는 계층적 물리적 아이템 및 그의 속성(41b)을 좌측에 있는 계층적 RDBMS 표현, 다시 말해, RDBMS



아이템 타입(41a)으로 매핑하는 것을 나타낸다. "아이템 타입"은 사용자 정의형 스키마 및 RDBMS 테이블로 네임이 지정된 계층적 구조이고, 각각은 "아이템 타입"의 "구성 요소 타입"(35)으로 지칭된다. 각각의 "구성 요소 타입"은 소정의 시스템 속성뿐만 아니라 사용자 지정형 속성(들)을 포함한다.

사용자 지정형 속성은 RDBMS에 의해 지원되는 임의의 데이터 모델이거나 콘텐츠 관리 시스템에 의해 지원되는 특수한 데이터 모델일 수 있다. "아이템" 내에 포함된 자녀 행(child row)의 부모 행(parent row)도 "아이템" 내에 포함되어야 한다는 조건 하에서, "아이템"은 논리적으로 제각기의 "아이템 타입"의 루트 테이블 내에 있는 하나의 행과, "아이템 타입"의 각 자녀 테이블 내에 있는 임의의 개수의 행으로 구성된다. "아이템"은 테이블의 계층 내에 저장되는 행의 계층이다. 이러한 행은 각각 "아이템"의 "구성 요소"로 지칭되고, 제각기의 "구성 요소 타입" 내에서 "구성 요소"를 고유하게 식별하는 "구성 요소 ID"를 갖는다.

카디널리티(cardinality) 제약은 각각의 자녀 "구성 요소 타입"마다 정의되어, 각각의 해당되는 부모 행 아래에 해당 타입을 갖는 최소수 및/또는 최대수의 행이 존재하게 할 수 있다. 디폴트 최소값은 0이고, 디폴트 최대값은 무한대이다. 또한, "아이템 타입"은 순차적이거나 비순차적일 수 있다. 순차적인 "아이템 타입"의 "아이템"에 있어서, 각 부모 행 아래의 자녀 행은 순차적이고, 그 순차화는 시스템에 의해서 유지된다.

"아이템 타입"은 버전(version)을 가질 수 있다. 그렇게 되면, "아이템 타입" 내의 모든 "아이템"마다 하나의 "아이템"에 대한 다수의 버전이 유지된다. 이들 버전은 동일한 "아이템 ID"를 공유하지만, 각각 서로 다른 버전 번호를 갖는다. "아이템"에 대해 유지되는 최신 버전의 번호는 각각의 버전을 갖는 각각의 "아이템 타입"에 지정될 수 있다. 디폴트는 무한대이고, 다시 말해, 구 버전의 자동적인 퍼징(purging)은 시스템에 의해 실행되지 않을 것이다.

바람직한 일 실시예에서, "아이템 타입" 내에서 2개의 "구성 요소 타입" 간의 부모-자녀 관계는 부모 "구성 요소 타입"을 참조하는 자녀 "구성 요소 타입"(또는 하부 구성 요소 타입) 내에서 생성된 RDBMS 외부 키(Foreign Key)에 의해 유지된다. 그러므로, 자신의 부모 "구성 요소"에 대한 각 자녀 "구성 요소"(하부 구성 요소)의 참조 무결성(referential integrity)은 RDBMS에 의해 보장된다. 본 실시예에서, "아이템" 내의 각 "구성 요소"는 적어도 "아이템 ID", "구성 요소 ID" 및 부모 ID 등과 같은 시스템 속성을 포함하고, 선택적으로 "아이템" 버전 번호 및 ACL Code(Access Control List code)를 포함할 수 있다. "아이템 ID", "아이템" 버전 번호 및 ACL 코드는 오직 루트 "구성 요소"에 포함되지만 하부 되지만, 액세스를 용이하게 하기 위해서 모든 "구성 요소" 내에 포함될 수도 있다. 그 외의 시스템 속성은 또한 액세스를 용이하게 하거나, 예를 들면 "구성 요소 타입", 임의의 "구성 요소"에 직속되어 있는 자녀 구성 요소의 개수 등의 유지를 용이하게 하기 위해서 포함될 수 있다. 또한, 임의의 "아이템 타입"에 대해 정의되는 각각의 "구성 요소 타입"마다 별개의 RDBMS 테이블을 생성하고, 즉, 테이블은 2개의 "아이템 타입" 사이에서 공유되지 않는다. 이 설계는 논리 구조를 단순화하고, 또한 데이터베이스 크기 조절성(scalability)을 강화한다. 다른 실시예에서, 테이블은 수 개의 "아이템 타입"들 사이에서 공유되어 이러한 수 개의 "아이템 타입"에 대해 유사하게 정의된 구성 요소 타입의 구성 요소(행)를 저장할 수 있다. 이 경우에, 이러한 공유형 테이블 내의 시스템 속성으로서 "아이템 타입"을 포함하는 것에 의해 액세스가 용이해질 수 있다. 이러한 방식으로 하나의 테이블을 공유하는 것은, 검색 대상인 공통 속성(들)을 포함하고 있는 다수의 "아이템 타입"이 검색 범위에 포함될 때, 더 빠른 검색이 가능하게 한다.

도 5에 도시된 바와 같이, 아이템(자녀 구성 요소, 순차 구성 요소 및 속성을 포함함)은 예를 들면 링크를 이용하는 시스템 테이블을 통해서 서로 관련되거나 연관될 수 있다. 링크는 소스 "아이템"(51)을 링크 타입(55)과 함께 타겟 "아이템"(53) 및 링크 "아이템"(57)에 결합시키는 "외부적(external)" 관계이다. 이 관계는 소스 또는 타겟 "아이템"의 어느 쪽에도 포함되지 않기 때문에 "외부적"이다. 링크 타입은 애플리케이션이 특정한 타입을 갖는 링크를 선택적으로 통과하거나 검색할 수 있게 하는 카테고리 라벨이다. 링크 "아이템"은 필요한 경우 이러한 관계를 기술하는 데 사용될 수 있는 제 3의 "아이템"이다. 링크를 사용하여 집합 관계(aggregation relationship) 및 포함 관계(containment relationship)를 포함하여 일-대-다(one-to-many) 및 다-대-다(many-to-many) 관계를 지원할 수 있다.

각 관계에 대한 시맨틱(semantics)은 하이 레벨 데이터 모델에 의해 정의된다. 일례로는 "폴더링(folding)"이 있는데, 이 경우에 폴더를 나타내는 "아이템"은 이 폴더에 "포함"된 아이템(문서 또는 다른 폴더를 나타냄)에 연결될 수 있다. 예를 들면, 소스 아이템(51)은 타겟 아이템(53)에 대한 컨테이너(container)로서 기능한다. 링크를 소스 또는 타겟 "아이템"에 결합시키는 것은 버전 특유(version-specific) 형태이거나 버전 독립(version-independent) 형태일 수 있다. 버전 특유형은 특정한 버전의 아이템을 결합시키는 반면에, 버전 독립형은 임의의/모든 버전의 "아이템"을 결합시킨다. 일 실시예에서, 링크는 적어도 다음의 속성, 즉 소스 "아이템 ID", 소스 "아이템" 버전 번호, 타겟 "아이템 ID", 타겟 "아이템" 버전 번호, 링크 타입, 링크 "아이템 ID" 및 링크 "아이템" 버전 번호를 포함하는 하나 이상의 전용 RDBMS 테이블 내에서 행으로서 유지된다.

도 6에 도시된 바와 같이, 아이템은 또한 "아이템" 내에 포함된 다른 속성과 마찬가지로 유지되는 "아이템 참조" 속성을 이용하여 서로 관계될 수 있다. "아이템 참조" 속성은 "아이템 타입"(32) 내에 있는 "구성 요소 타입"(36)의 일부분으로서 생성된 사용자 정의형 속성이다. 이 속성은 콘텐츠 관리 시스템에 의해 유지되는 특수한 데이터 모델을 갖는다. 이 속성은 타겟 "아이템"(32')을 가리키도록 지정된 참조(58)를 포함한다. 도 5의 링크와 유사하게, 참조(58)를 타겟 "아이템"(32')에 결합시키는 것은 버전 특유형이거나 버전 독립형일 수 있다. "아이템 참조"(58) 속성이 지정되면, 참조된 아이템에 대한 참조의 범위 및 삭제 시맨틱(Delete Semantics)이 정의된다. 범위는 참조된 아이템이 어떤 "아이템 타입"(들)에 속할 수 있는지를 지정하는 반면에, 삭제 시맨틱은 RDBMS에 의해 유지되는 삭제 법칙과 유사해서, 참조된 "아이템"이 삭제되는 경우에 참조 무결성을 보장한다. "아이템 참조" 속성(58)은 또한 범위가 지정되지 않을 수 있는데, 이 경우에 참조된 "아이템"은 임의의 "아이템 타입"에 속할 수 있다. 바람직한 실시예에서, 균일하게 범위가 지정된 "아이템 참조" 속성(즉, 범위가 단일 "아이템 타입"일 때)은 RDBMS 외부 키에 의해서 지원된다. 균일하지 않게 범위가 지정되고(즉, 범위가 다수의 "아이템" 타입을 포함할 때), 범위가 지정되지 않은 "아이템 참조" 속성은 RDBMS 트리거(triggers)를 이용하는 콘텐츠 관리 시스템에 의해 유지된다.

이러한 물리적 데이터 모델에 의해 지원되는 다른 특수한 데이터 모델은 도 6에 도시되어 있는 리소스 참조 속성(59)이다. 도 6에 도시된 바와 같이, 리소스 참조(59)는 콘텐츠 관리 시스템의 리소스 관리자에 의해 관리되는 리소스(61)에 대한 참조를 포함한다. 리소스 관리자는 특정한 타입의 리소스를 관리하는 전용 서버이다. 그에 대한 예시로는 대용량의 이진 객체(binary objects)를 관리하는 객체 서버와, 스트리밍 객체(예를 들면, 오디오 및 비디오 콘텐츠)를 관리하고 전송하는 스트리밍 서버가 있다. 확장 가능 콘텐츠 관리 시스템은 새로운 리소스 관리자를 수용할 수 있는데, 각각의 새로운 리소스 관리자는 본 명세서에 설명된 물리적 데이터 모델에 의해서 설계되어야 할 필요가 없는 특수한 표현 및 방법을 가지고 특수한 타입의 리소스를 관리하도록 특별히 설계된다. 콘텐츠 관리 시스템에 의해 관리되는 비구조화 데이터는 하나 이상의 이러한 리소스 관리자에 의해서 대부분 내부적으로 관리된다. 리소스 참조의 무결성은 콘텐츠 관리 시스템에 의해 유지된다.

본 명세서에서 설명된 물리적 데이터 모델은 넓은 범위의 하이 레벨 데이터 모델을 지원할 수 있어서 여러 산업 표준(예를 들면, Dublin Core, XML)을 지원할 뿐만 아니라 애플리케이션 도메인의 다양한 스펙트럼을 포함할 수 있다. 또한, 이와 동등한 정도로 중요하게, 이러한 물리적 데이터 모델은 또한 데이터베이스 최적화를 가능하게 하기에 충분한 유연성을 제공하여 하이 레벨 데이터 모델을 효과적으로 지원할 수 있게 한다. 일실시예에서, 하이 레벨 데이터 모델은 하나의 "아이템 타입"을 갖는 아이템을 사용하여 독립형의 공유 가능 이미지를 나타내는데, 그 각각은 자신의 속성, 버전 및 ACL을 갖는다. 실제 이미지는 리소스로서 저장된다. 다음에 다른 "아이템 타입"의 아이템을 사용하여 문서를 나타내고, 링크를 사용하여 각 문서를 해당 문서가 포함하는 이미지와 결합시킨다. 다른 실시예에서는, 고성능 제품 애플리케이션을 지원하기 위해서, 단일 "아이템"을 사용하여 문서 내에 포함된 이미지와 함께 해당 문서를 나타낸다(실제 이미지는 또한 리소스로서 저장됨). 이 경우에, 성능이 중요하기는 하지만 이미지의 공유 조건은 존재하지 않는다. "아이템 뷰(Item View)"는 "아이템 타입"에 대해 정의될 수 있다. "아이템 뷰"는 소정의 "구성 요소" 테이블 및 속성에 대한 "아이템 타입"으로의 액세스를 제한하기 위해 네임이 지정된 템플릿이다. "아이템 뷰"는 기본적으로 "아이템 타입" 내에서 대응되는 "구성 요소" 테이블에 대한 RDBMS 뷰의 세트이다.

본 발명의 하나의 애플리케이션은, 예를 들면, 자기 카드 보안 시스템, 신용 카드 또는 입출금 카드(debit card) 또는 ATM(automatic teller machine) 카드 시스템이나 항공 상용 고객 시스템(airline frequent traveler system)에서 사용되는 것과 같은 분산형 카드 판독기 액세스(distributed card reader access)를 구비하는 멀티미디어 분산형 데이터베이스 관리 시스템이다. 이러한 시스템의 하나의 버전을 도 7에 도시하였다. 시스템(70)은 식별 표시, 예를 들면 카드 판독기(70a)에 의해 자기 또는 광학 카드(70b)에서 판독되는 암호화된 데이터에 의해서 작동된다.

카드(70b)를 판독한 후에, 바이오메트릭 판독(biometric readings)을 수행한다. 예시적인 바이오메트릭 데이터는 특히 지문 판독기(70c)에서의 지문과, 음성 지문(voice prints) 및 음성 샘플과, 디지털 카메라(70d)에서의 안면 이미지 및 안구 이미지를 포함한다. 바이오그래픽, 오디오 및 이미지 데이터는 전형적으로 별개의 서버(72a, 73a, 74a, 75a)에 의해서 액세스되고 처리되며, 별개의 데이터베이스(72b, 73b, 74b, 75b)에 저장되는데, 예를 들면, 지문 데이터, 음성 데이터 및 안면 및 안구 이미지 데이터는, 리포지터리 어드레스(예를 들면 URL 어드레스 또는 IP 어드레스 등)뿐만 아니라 데이터 모델(이미지, 이미지 데이터의 수학적 표현, 오디오 데이터 및 오디오 데이터의 수학적 표현) 등과 같은 속성을 가지고 별개의 지문 데이터, 음성 데이터 및 안면 및 안구 이미지 데이터베이스에 멀티미디어 데이터 객체로서 저장되는데, 여기에서 데이터는 아이템이고, 데이터 모델의 어드레스 및 정의는 속성이며, 선별자(screener), 보안 요원, 출납원, 탑승 관리자(boarding agent) 또는 수사 전문 요원(enforcement professional)에게 제공되는 이미지는 리소스이다.

본 발명의 방법 및 시스템은 프로그램 제품 내에서 물리적으로 구현된다. 프로그램 제품은 하나의 컴퓨터 또는 수 개의 컴퓨터, 분산형 서버, 하나의 디스크 또는 여러 디스크 또는 테이프에 상주할 수 있다. 프로그램 그 자체는 예를 들어, 분산형 매체 또는 분산형 서버의 경우에서와 같이 설치되기 이전에는 암호화 및/또는 압축되어 있거나, 하나 이상의 컴퓨터에 설치될 수 있다.

본 발명의 다른 애플리케이션은 콘텐츠 스키마 및 메타데이터를 가지고 하이 레벨 콘텐츠 모델 및 로우 레벨 물리적 모델을 포함하는 멀티미디어 콘텐츠 관리 시스템을 파퓰레이트하는 위저드(wizard)(즉, 질문을 물어보고 답변 또는 선택을 사용하여 결과를 생성하는 구조화된 일련의 대화(dialogs))의 속성을 갖는 방법 및 프로그램 제품이다. 이는 사용자에게 콘텐츠 아이템에 대한 질의를 제시하고, 최종 사용자의 응답에 기반하여 콘텐츠 아이템에 대해 하나 이상의 후속 질의를 제시하며, 해당되는 응답에 기반하여 해당 아이템의 하부 구성 요소 및 속성을 결정하는 것에 의해서 이루어진다.

## 도면의 간단한 설명

도 1은 다수의 클라이언트, 콘텐츠 관리 서버 및 다수의 콘텐츠 서버를 구비하는 멀티미디어 데이터베이스 시스템을 도시하는 도면,

도 2는 API의 구조를 하이 레벨 데이터 모델로서 구비하고, 범용의 로우 레벨 하부 구조를 물리적 데이터 모델로서 구비하며, 그 기반이 되는 관계형 데이터베이스 관리 시스템을 구비하는 본 발명에 따른 방법, 시스템, 및 프로그램 제품의 적응형 구조를 도시하는 도면,

도 3은 루트 구성 요소를 위한 행을 가지고 구성 요소 및 속성 셀(열)을 포함하는 아이템 테이블과, 하부 구성 요소 및 속성 셀(열)을 갖는 구성 요소 테이블과, 적절한 포인터를 갖는 하부 구성 요소 및 속성 모두를 위한 손자 테이블(grandchild tables)을 포함하는 계층적 RDBMS 테이블을 도시하는 도 2의 관계형 데이터베이스 관리 시스템의 단순화된 개략도,

도 4는 도 3에 도시된 RDBMS 테이블 사이의 관계를 도시하고, 아이템 타입의 예로서, 루트 구성 요소, 자녀 구성 요소 및 손자 구성 요소를 포함하는 아이템에 대해 매핑된, 루트 테이블, 자녀 테이블 및 손자 테이블을 포함하는 아이템 타입의 병렬식 매핑(side-by-side mapping)을 도시하는 도면(각각의 구성 요소는 대응되는 테이블 내에서 하나의 행임),

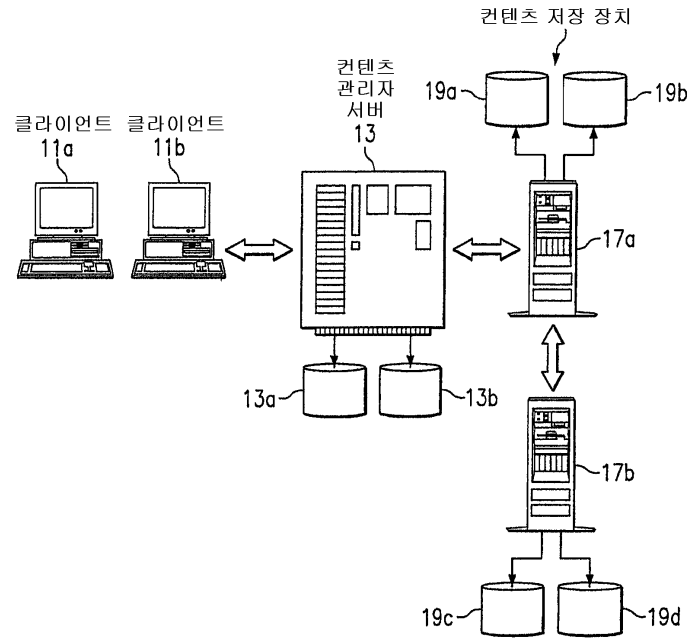
도 5는 도 2에 도시된 RDBMS의 테이블 내의 하나의 행에서 소스 아이템, 타겟 아이템 및 링크 아이템이 연관되어 있는 것을 도시하는 도면,

도 6은 타겟 아이템에 대한 리소스 참조의 애플리케이션을 도시하는 도면,

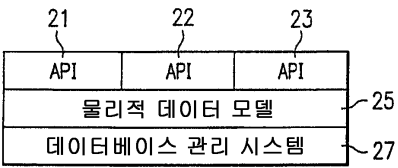
도 7은 예를 들면 보안 시스템 내에서 단말기에서 사용되는 여러 서버에 저장된 멀티미디어 콘텐츠를 갖는 개인 식별 애플리케이션(personal identification application)을 도시하는 도면.

## 도면

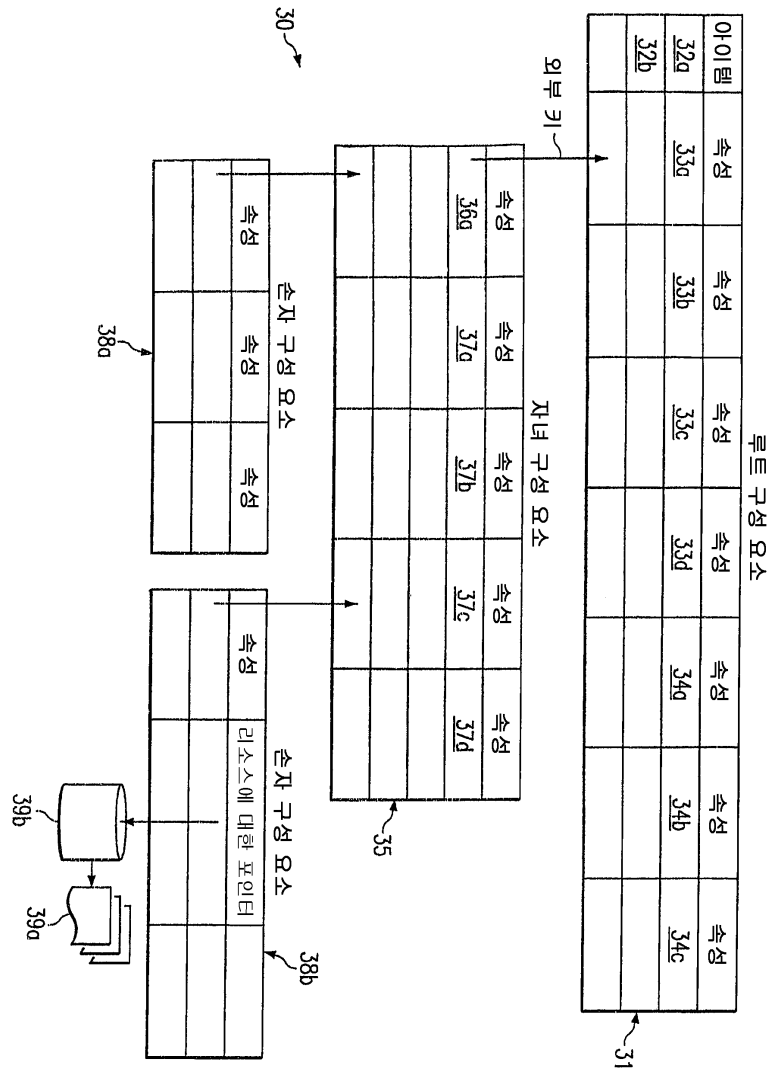
도면1



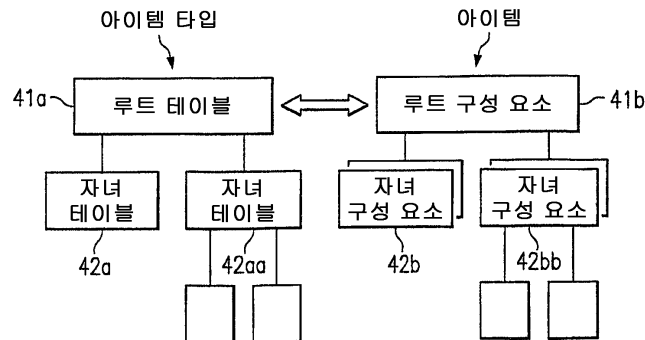
도면2



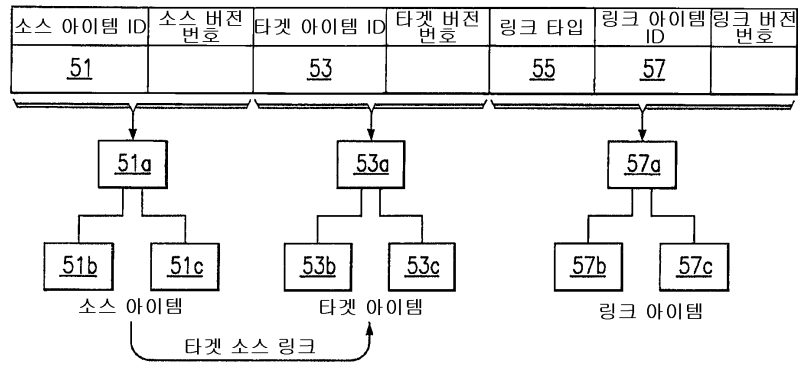
도면3



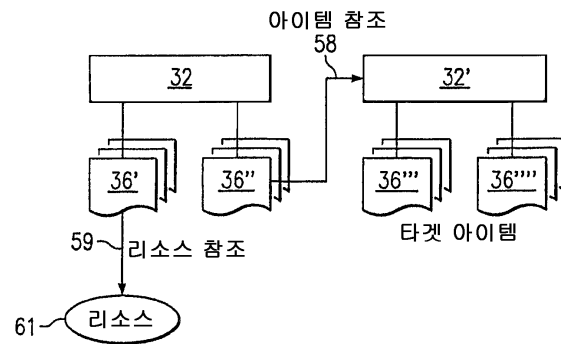
도면4



도면5



도면6



도면7

