

멀티미디어 프리젠테이션을 위한 데이터베이스 질의어 설계

(Design of a Database Query Language for Multimedia
Presentation)

이 중 화 †
(Junghwa Lee)

요 약 멀티미디어 데이터베이스 시스템에서는 멀티미디어 데이터를 질의하고 질의 결과를 어떻게 프리젠테이션 할 것인가에 대한 연구는 매우 중요하다. 또한 질의의 결과를 다양한 응용에서 사용하기 위해서는 보다 일반화된 형태로 질의결과를 제공해야할 필요가 있다.

본 논문에서는 멀티미디어 데이터베이스 시스템에서 질의결과를 프리젠테이션 하기 위한 프리젠테이션 모델과 멀티미디어 프리젠테이션을 지원하는 MMSQL을 제안하고 MMSQL 질의를 처리하기 위한 질의처리기를 설계하고 구현하였다.

또한 질의 결과 프리젠테이션을 보다 일반화된 형태인 SMIL문서로 제공하도록 함으로써 질의 결과를 다양한 응용에서 쉽게 사용할 수 있도록 하였다.

키워드 : 프리젠테이션, 멀티미디어, 데이터베이스

Abstract In multimedia database system, it is very important issue how to query multimedia data and present the result of query. Also, in order to reuse the query result in other applications, multimedia database system must be considered to provide the query result which is more generalized form.

In this paper, I propose a presentation model for presenting query result in multimedia database system and MMSQL based on the developed model for processing multimedia database query. This paper includes the design and implementation of MMSQL query processor. MMSQL query processor provides the presentation of query result which is SMIL (Synchronized Multimedia Integration Language) document. Therefore, MMSQL query result in this paper can be use to variant multimedia applications.

Key words : presentation, multimedia, database

1. 서 론

멀티미디어 데이터는 물리적인 구조면에서 대용량이라는 특성을 가지고 있기 때문에 기존의 데이터를 관리하는 관리기법으로는 처리하기에 어려움이 따른다. 이러한 요구 사항을 만족하기 위해 데이터베이스에서 대용량의 데이터를 다루기 위한 연구들이 진행되었다. 그 결과로 최근에 발표되는 데이터베이스 관리 시스템(Database Management System, 이하 DBMS)들은 대부분 대용량 데이터를 저장하고 관리하기 위한 BLOB(Binary Large

Object)을 지원함으로써 멀티미디어 데이터를 저장하는 문제는 어느 정도 해결된 것으로 생각된다.

멀티미디어 데이터를 질의할 수 있는 질의어 설계의 측면에서 살펴보면, 멀티미디어 데이터의 구조 정보나 내용 정보를 질의해 낼 수 있도록 하는 연구[1, 2, 3]뿐만 아니라 질의의 결과를 어떻게 사용자가 원하는 형태로 제공할 것인가에 대한 연구도 중요하다. 이를 멀티미디어 프리젠테이션이라고 한다.

멀티미디어 데이터베이스 질의어에서 프리젠테이션을 지원하는 방법으로는 기존의 구문에 프리젠테이션을 위한 새로운 구문을 추가하는 형태가 일반적인데, 이는 프리젠테이션을 위해 새로운 질의어를 디자인하는 방법에 비해 사용자가 새로운 질의어를 배울 필요가 없으며, 기

† 정 회 원 : 동의대학교 컴퓨터소프트웨어공학부 교수

junghwa@dongeui.ac.kr

논문접수 : 2002년 8월 6일

심사완료 : 2003년 1월 13일

존의 질의어의 장점을 그대로 살릴 수 있다[4, 5, 6, 7].

그러나 새로운 구문을 추가하는 방법에 있어서도 기존의 여러 연구들을 살펴보면 사용자 요구 사항의 분석에 따른 시,공간 관계의 정의에서부터 서로 다른 견해를 가지고 있으며 제공되는 기능들도 차이점이 많기 때문에 기존의 SQL과 같이 보편적으로 사용되기에는 문제점이 있다.

따라서 본 논문에서는 멀티미디어 데이터를 데이터베이스에서 처리하기 위하여 멀티미디어 데이터와 멀티미디어 프리젠테이션을 모델링하고 이를 기반으로 SQL을 확장하여 멀티미디어 데이터에 대한 사용자가 질의를 원활히 할 수 있도록 멀티미디어 데이터베이스 질의어 MMSQL (MultiMedia Structured Query Language)을 제안하고 MMSQL을 위한 MMSQL 질의 처리기를 설계하고 구현한다. MMSQL의 질의 결과는 W3C(World Wide Web Consortium)의 멀티미디어 프리젠테이션 표준인 SMIL (Synchronized Multimedia Integration Language) 문서로 제공함으로써 다양한 멀티미디어 응용에서 MMSQL 질의 결과를 사용할 수 있는 방안을 제공한다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서는 멀티미디어 데이터 처리를 위한 멀티미디어 프리젠테이션과 질의어 개발에 관련된 기존의 연구들에 대해 살펴보고, 3장에서 멀티미디어 데이터베이스에서 질의 결과로 추출된 데이터를 사용자가 원하는 형태로 볼 수 있도록 하는 멀티미디어 프리젠테이션 모델에 대해 살펴본다. 또한 프리젠테이션 모델을 기반으로 멀티미디어 프리젠테이션을 위한 시간 배치 기법과 공간 배치 기법을 제안한다.

4장에서는 본 논문에서는 제안하는 객체 모델과 프리젠테이션 모델을 효과적으로 지원할 수 있는 멀티미디어 데이터베이스 질의어 MMSQL을 설계하고, 5장에서 MMSQL 질의 처리 시스템에 대해 살펴본다. 마지막으로 6장에서 결론과 향후 연구 과제에 대해 살펴본다.

2. 관련 연구

2.1 멀티미디어 프리젠테이션에 관한 연구

멀티미디어 데이터베이스에서의 프리젠테이션에 관한 기존의 연구들로는 [6, 8, 9, 10] 등이 있다.

MPGS는 멀티미디어 프리젠테이션 생성기에 대한 연구로서 다양한 멀티미디어 데이터에 대한 프리젠테이션을 정의하고 이를 생성해 내는 시스템을 개발하였다[8]. MPGS에서는 멀티미디어 데이터가 가질 수 있는 시간 제약 조건과 공간 제약 조건들을 정의하고 각 조건을 표현하는 다양한 시,공간 연산자를 정의하고 있으며, 사용자는 정의된 시,공간 연산자를 통해 멀티미디어 프리

젠테이션을 정의할 수 있다. 사용자가 정의한 멀티미디어 프리젠테이션은 프리젠테이션 생성기를 통해 생성하여 MPGS 프리젠테이션 시스템을 통해 사용자에게 보여진다.

[9]는 DOOR 데이터베이스 시스템에서의 멀티미디어 프리젠테이션에 대한 연구이다. 이 연구에서는 멀티미디어 데이터가 프리젠테이션에서 수행하는 역할을 규정함으로써 멀티미디어 프리젠테이션을 정의할 수 있도록 하고 있다. 또한 멀티미디어 데이터의 역할을 동적으로 정의할 수 있도록 함으로써 멀티미디어 프리젠테이션을 동적으로 변경 가능하도록 하고 있다.

그러나 위의 연구들은 고유의 멀티미디어 프리젠테이션 처리기를 통해서 화면에 보여지도록 설계되어 일반화하는데 어려움이 따르며, 멀티미디어 데이터베이스 질의를 통해 표현하는 방법을 정의하지 않고 있다.

SQL+D는 기존의 SQL 질의어에 프리젠테이션 절을 추가하여 결과 출력을 위한 그래픽 인터페이스를 디자인할 수 있도록 하였다[6]. 확장된 프리젠테이션 절은 정형화된 GUI 요소들인 텍스트 박스, 체크 박스, 리스트, 버튼 등을 제공하며, 사용자가 결과 화면을 디자인할 수 있도록 한다.

2.2 멀티미디어 데이터베이스 질의어에 관한 연구

위의 연구와 더불어 멀티미디어 데이터베이스에서 멀티미디어 데이터에 대해 질의할 수 있고 또한 질의 결과의 프리젠테이션을 정의할 수 있는 멀티미디어 데이터베이스 질의어(query language)의 개발이 필요하다. 멀티미디어 데이터베이스 질의어 개발은 현재 데이터베이스에서 표준 질의어로 사용되고 있는 SQL을 확장하여 멀티미디어 데이터를 지원하는 형태와, SQL과는 다른 새로운 구조를 가지는 질의어 개발의 형태로 연구가 진행되었다.

국제 표준화 기구인 ISO(International Organization for Standardization)에서는 SQL을 확장하여 객체 지향 데이터베이스 표준 질의어인 SQL3을 개발하고 있다[2]. SQL3에는 부울형(boolean type)과 열거형(enumeration type) 등의 새로운 데이터 형과 멀티미디어 데이터를 처리하기 위해 필수적인 추상 데이터 형(abstract data type), 상속(inheritance), 사용자 정의형 생성(user defined type) 등도 지원한다.

ODMG(Object Data Management Group)에서는 객체 지향 데이터베이스 질의어인 OQL을 정의하였다. OQL은 기존의 SQL 형식을 그대로 유지하면서 객체 지향 데이터베이스에 필요한 추상 데이터 형과 상속, 객체 식별, 사용자 정의 형 등의 다양한 기능을 제공한다[3].

OQL과 SQL3는 객체 지향 데이터베이스 언어로 개발되었기는 하지만 멀티미디어 프리젠테이션과 같이 멀티미디어 데이터를 위한 기능을 제공하지 않기 때문에 멀티미디어 데이터베이스 질의어라고 할 수 없다. 그러나 멀티미디어 데이터 역시 객체 지향 개념으로 표현하는 것이 바람직하기 때문에 위의 두 연구는 멀티미디어 데이터베이스 질의어 개발의 기반이 되는 연구이다.

EVA는 집합 이론을 기반으로 하는 함수 언어(functional language) 형태의 객체 지향 언어이다. EVA에서는 멀티미디어 데이터의 시,공간 관계를 표현하기 위한 시,공간 연산자를 정의하고 이를 통해 질의 결과의 프리젠테이션을 제공한다[11]. 그러나 EVA에서는 정의된 시,공간 연산자가 부족하여 비디오 데이터에 대한 공간 관계 등을 정의하지 못한다는 단점이 있으며 [12]에서도 이와 같은 문제를 찾아볼 수 있다.

SEQL(Spatial Evolutionary Query Language)은 의료 이미지 정보 시스템에서 사용할 목적으로 SQL을 확장한 언어인데, SEQL에서는 기존의 SQL 구문에 when이라는 새로운 구문을 추가하여 미디어간의 공간과 시간 관계를 표현한다[13].

SQL과 다른 구조의 질의어를 제안한 연구로는 [10, 14]이 있다. [14]에서는 멀티미디어 데이터에 대해 시간 명세(temporal specification), 공간 명세(spatial specification), 내용 명세(media-content specification), 사용자에 의한 외부 명세(external specification) 등을 질의어에서 표현할 수 있는 방법을 제공한다.

PICQUERY+는 의료 이미지 데이터베이스에서 사용할 목적으로 개발된 데이터베이스 질의어로서, 데이터베이스에 관한 질의와 질의 결과의 프리젠테이션을 분리하여 정해진 템플릿 형식에 따라 질의를 하도록 설계되었다[10]. 질의의 결과는 프리젠테이션 템플릿의 입력이 되며, 프리젠테이션 템플릿 내의 정보에 따라 질의결과를 화면에 보여준다. 그러나 이 연구에서 제안하고 있는 질의어는 기존에 널리 사용되고 있는 SQL과 질의어 구조가 다르기 때문에 일반 사용자가 사용하기에 어려울 뿐만 아니라, 이미지 데이터베이스를 위한 내용만을 정의하고 있기 때문에 이를 광범위한 멀티미디어 질의어로 이용하기에는 적합하지 않다.

이 외에도 멀티미디어 데이터베이스 질의어에 대한 연구들은 [15, 16, 17] 등이 있으며, 다양한 미디어들에 대해 프리젠테이션을 정의할 수 있는 SQL/MR[18] 등이 있다.

2.3 SMIL(Synchronized Multimedia Integration Language)

SMIL(Synchronized Multimedia Integration Language)은 WWW(World Wide Web)에 관련된 표준을 만들어내는 단체인 W3C에서 작성한 표준 권고안으로, 웹 기반에서 멀티미디어에 대한 프리젠테이션을 기술하는 방법이다.

SMIL은 XML을 바탕으로 하며 텍스트, 비디오, 그래픽, 오디오, 벡터 애니메이션 등 여러 종류의 미디어 이루어진 멀티미디어 프리젠테이션을 기술할 수 있다[19].

SMIL은 W3C, Bell Labs, RealNetworks, Apple, Netscape, DEC 등의 많은 기관과 업체들이 참여하여 개발하였으며, 웹과 함께 이미 널리 배포되어 있는 RealNetworks사의 RealPlayer, Oratrix Development VB사의 Grins 등 많은 멀티미디어 플레이어들이 SMIL을 지원하고 있다.

3. 멀티미디어 프리젠테이션 모델링

3.1 멀티미디어 객체 클래스 정의

멀티미디어 데이터를 처리하기 위해서는 먼저 멀티미디어 데이터를 구성하는 객체들이 어떤 것들이 있는지를 정의해야 한다.

멀티미디어 객체 정의 단계에서는 먼저 응용 영역에서의 사용자 요구 사항을 분석하여 연관성 있는 객체 클래스들을 식별하고 임시적인 객체 클래스를 추출한 다음, 필요 없는 것과 필요한 것을 분리해 낸다.

본 논문에서는 정의된 멀티미디어 객체 클래스의 이름은 기존의 멀티미디어 연구들에서 사용하는 보편적인 용어들인 FullText, Image, Graphic, Audio, Video 등으로 사용하도록 한다.

본 논문에서 정의한 멀티미디어 객체 클래스들은 데이터베이스에서 객체 형으로 확장되어 사용자가 멀티미디어 데이터베이스를 구축할 때 사용된다.

미디어 클래스는 상속을 통해 하위 클래스와 연관되는데 아래 그림 1의 객체도는 객체 모델링의 결과로 만들어진 각 미디어 클래스의 상속관계를 보여준다[20].

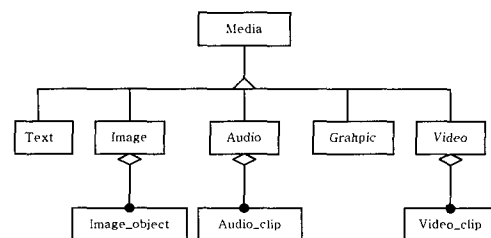


그림 1 미디어 클래스간의 상속 관계를 나타내는 객체도

3.2 미디어 객체간의 시간 관계

미디어간의 시간 관계는 Allen이 정의한 13가지의 시간 관계를 통해 표현할 수 있다. 예를 들어 그림 2와 같이 시간축 상에 여러 가지 미디어가 위치 할 경우 시간축 상에서 한 객체를 기준으로 다른 객체들이 어떠한 시간 관계를 가지느냐를 정의한다.

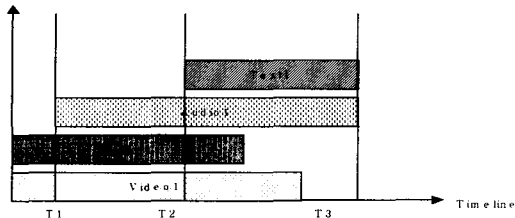


그림 2 시간축 상의 시나리오 표현

그림 2의 시나리오(scenario)에서 살펴보면, 먼저 Video1과 Image1은 시작과 동시에 화면에 나타나며, T1의 시간이 후 Audio1이 재생되고, 시작 시점부터 T2시간이 지난 후 화면에 Text2가 화면에 나타나며 Audio1이 끝남과 동시에 화면에서 없어지도록 하는 것이다.

3.3 시간 배치(Temporal layout) 모델링

실제 미디어들이 프리젠테이션 될 때에는 시간 관계와 아울러 시간축 상에서 명확한 위치를 정할 수 있는 방법이 필요한데 이를 시간 배치(temporal layout)라 한다.

미디어가 시간축에 위치 할 때 가질 수 있는 값들은 언제 시작할 것인지에 대한 명시적인 시작 시점(Ts), 얼마동안 지속적으로 나타날 것인가에 대한 지속 값(duration), 그리고 언제 끝날 것인지에 대한 명시적인 끝 시점(Te) 등이 있다. 그림 3은 미디어가 시간축에 나타날 수 있는 값들을 보여 준다.

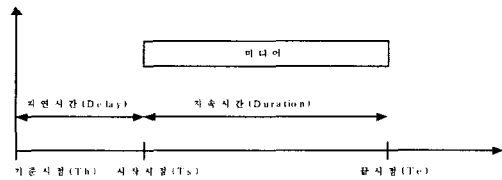


그림 3 시간축 상의 미디어가 가질 수 있는 값

그림 3의 시간 값들을 정할 수 있을 경우 Allen의 시간 관계는 병렬(parallel) 관계와 순차(sequential) 관계 그리고, 미디어가 가질 수 있는 시간 값들로 단순화시켜

표현할 수 있다[14]. 순차 관계는 시간축 상에서 두 미디어가 겹치는 부분 없이 위치하는 경우이며 병렬 관계는 반대로 미디어가 겹치는 경우를 말한다.

본 논문에서는 Allen의 시간 관계를 병렬 관계를 나타내는 par와 순차 관계를 나타내는 sync, 그리고 의 두 가지 시간 관계와 미디어들의 시간 값들로 나타낸다.

par와 sync를 위한 속성들로는 시간 관계를 이루는 미디어들의 식별자(ID)와 지속시간을 나타내는 duration 그리고 동기화 위치를 나타내기 위한 begin, end 속성 등이 있다.

시간축 상의 두 미디어간의 시간 관계 즉, 이진 시간 관계를 나타낼 때에는 연관되어 있는 미디어와의 상대적인 시간 관계를 표현할 수 있어야 한다. 미디어간의 상대적인 시작 시점은 begin 속성을 그림 4의 BNF구문과 같이 정의함으로써 표현할 수 있다.

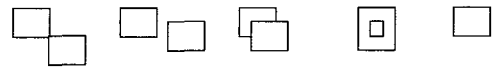
```

Element-exp ::= "id("Source")("Event")
Source ::= id-value
Event ::= "begin" | Clock-val | "end"
    
```

그림 4 begin 속성을 위한 BNF

3.4 미디어 객체간의 공간 관계

미디어 객체가 화면상에서 나타날 수 있는 관계는 먼저 위상 관계와 방향 관계로 정의할 수 있다. 두 미디어 객체간의 위상 관계는 그림 5와 같이 접합(touch), 떨어짐(disjoint), 겹침(overlap), 포함(contain), 동일(equal) 등의 관계로 나타날 수 있다.



접합(Touch) 떨어짐(Disjoint) 겹침(Overlap) 포함(Contain) 동일(Equal)

그림 5 미디어 객체간의 위상 관계

그림 5의 위상 관계만으로는 화면상의 위치를 정할 수 없는데, 여기에 필요한 것이 방향 관계이다. 방향 관계는 미디어 객체가 기준이 되는 미디어 객체로부터 어느 방향으로 위치하느냐를 정의하는 것인데, 여기에는 그림 6과 같이 왼쪽(left), 오른쪽(right), 위(above), 아래(below), 왼쪽 위(left_above), 왼쪽 아래(left_below), 오른쪽 위(right_above), 오른쪽 아래(right_below) 등의 8가지 방향 관계가 존재한다.

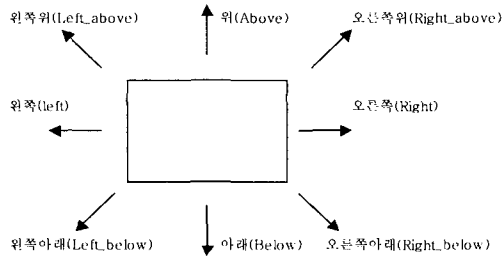


그림 6 미디어간의 방향 관계

위에서 살펴본 위상 관계와 방향 관계를 사용하면 두 객체간의 상대적인 공간 관계를 나타낼 수 있는데, 실제로 화면상에 미디어 객체가 보여질 때에는 좌표를 통해 미디어 객체들의 위치 영역을 정하게 된다. 이 경우 미디어 객체들의 영역 좌표간의 관계를 통해서 위상 관계와 방향 관계를 정할 수 있다.

3.5 공간 배치(spatial layout) 모델링

미디어 객체를 화면에 배치할 때에는 화면의 위치를 정하기 위한 속성들의 정의가 필요하다. 이미지나 비디오와 같이 정확화된 크기 속성을 가지는 미디어 객체들은 미디어 객체가 위치할 시작점의 좌표와 미디어 객체의 크기 속성을 가지고 화면에 배치할 수 있으나, 텍스트와 같이 크기 속성을 가지지 않는 미디어 객체들은 화면에 배치하는데 어려움이 있다. 또한 크기 속성을 가지는 미디어 객체들도 사용자가 화면에서의 크기를 조정할 수 있는 방법이 필요하다. 이를 위해서 본 논문에서는 일정한 영역을 가지는 미디어 창(media window)을 정의한 다음 미디어 객체들은 정의된 미디어 창 내에 위치하도록 한다. 미디어 창은 그림 7과 같이 top, left 속성과 width, height 속성을 통해 화면에 배치한다.

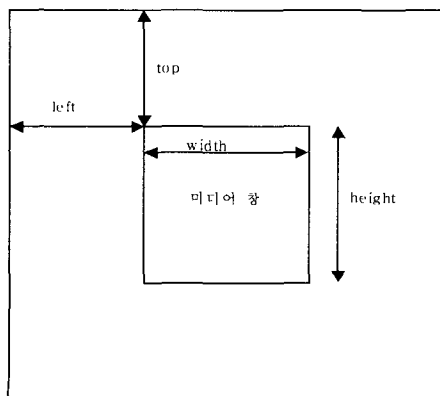


그림 7 미디어 창의 배치

미디어 객체가 정해진 창에 위치할 때, 이미지나 비디오와 같이 자체 크기 속성을 가지는 미디어 객체일 경우 창의 크기에 따라 미디어 객체의 크기를 어떠한 형태로 변환 시켜야 할 지를 정의할 필요가 있다. 본 논문에서는 미디어 객체의 크기와 미디어 창의 크기가 같지 않을 경우 미디어 객체의 크기 변환을 위해서 adjust 속성을 정의한다. adjust 속성에는 표 1과 같은 값들을 가질 수 있다.

표 1 adjust 속성이 가질 수 있는 속성 값

속성 값	설 명
hidden	크기의 변화 없이 창에 출력한다. - 창의 크기가 미디어 크기보다 클 경우 왼쪽 위부터 출력하고 남은 영역은 그대로 둔다. - 창의 크기가 미디어 크기보다 작을 경우 왼쪽 위부터 출력하고 나머지는 잘라낸다.
fill	높이와 너비를 독립적으로 배율을 조정하여 창 크기에 맞춤
meet	모든 내용이 표시될 때까지 높이와 너비의 배율을 유지하면서 잘리는 내용이 없도록 크기를 조정한다.
slice	높이와 너비의 배율을 유지하면서 창 크기에 맞춤
scroll	창에 그대로 출력하고 창의 크기를 벗어나는 경우 scroll bar를 표시

위의 속성들 중에서 meet와 slice는 높이와 너비의 배율을 그대로 유지하면서 크기를 조정한다는 측면에서는 동일하지만 그림 8과 같이 빈 공간을 처리하는 측면에서 차이점이 있다.

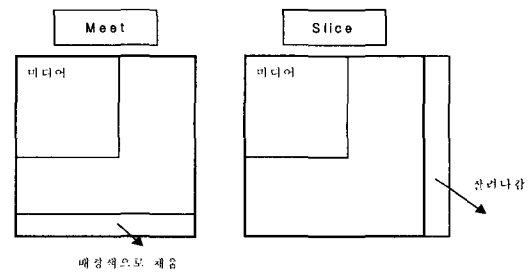


그림 8 meet와 slice 속성 값의 사용 예

앞의 그림 5에서 살펴본 위상 관계들 중에서 겹침(overlap)과 포함(contain) 동일(equal) 등의 위상 관계는 화면상에 두 개의 미디어가 겹치는 부분이 있는 경우이다. 이 때는 겹치는 부분에 어느 미디어의 내용을 표시할 것인지를 정의할 수 있어야 한다. 본 논문에서는 미디어가 놓일 창의 화면 표시 순서를 기술할 수 있는

order 속성을 추가한다. 화면에 창을 출력할 때에는 order 값이 작은 순서대로 출력함으로써 출력될 창에 미디어가 중첩(overlap) 될 때 어느 미디어를 사용자에게 보여줄 지를 정할 수 있다.

4. 멀티미디어 데이터를 지원하기 위한 SQL의 확장

4.1 멀티미디어 프리젠테이션을 위한 구문 정의

질의어에서 프리젠테이션을 정의하기 위해서 본 논문에서는 새로운 구문을 정의하고 기존의 질의어에 추가한다. 이 방법은 질의어를 확장하는 데 있어서 기존의 구문을 변경 없이 그대로 사용할 수 있어 확장이 용이하다는 장점이 있다.

본 논문에서는 질의 결과의 프리젠테이션 정보를 나타내기 위해 새로운 절인 PRESENT절을 기존의 구문에 추가한다. 추가할 프리젠테이션 절은 그림 9와 같이 미디어의 공간 배치를 정의하는 SLAYOUT절과, 시간 배치를 정의하는 TLAYOUT 절을 포함한다.

```
<present> ::= [<slayout>] <tlayout>
```

그림 9 PRESENT 절의 정의

4.2 공간 배치를 위한 질의어 확장

앞의 3.5 절에서 공간 배치를 표현할 수 있는 여러 가지 속성들을 정의하였다.

본 논문에서는 위의 속성들을 포함하여 미디어 객체의 공간 배치를 기술할 수 있는 구문으로 SLAYOUT 절을 정의한다.

SLAYOUT 절에서는 먼저 화면의 크기를 정할 수 있는 ROOTREGION과 미디어가 놓일 창인 REGION을 정의한다. ROOTREGION에서는 화면의 크기를 나타내는 width, height속성과 바탕색을 지정할 수 있는 bgcolor 속성이 정의되며, REGION에서는 창에 놓을 미디어를 지정하는 RID와 창의 위치를 지정하기 위한 left, top, width, height 속성 그리고 바탕색 지정을 위한 bgcolor 속성이 정의된다.

REGION이 가질 수 있는 키워드 중에서 ORDER는 창의 활성화 순서를 결정하는 키워드이다. 앞에서도 살펴본 바와 같이 미디어가 놓일 창이 겹쳐서 나타날 경우 겹치는 부분에 어느 창의 내용이 표시될 지를 결정해야 하는데, 이 때는 ORDER의 값이 작은 순서대로 활성화되기 때문에 사용자에게는 가장 큰 ORDER값을 가지는 창에 위치한 미디어의 내용이 보여지게 된다.

ADJUST는 미디어 크기 조절을 위한 것으로 표 1의 속성 값을 가질 수 있다.

그림 10은 화면의 배치를 결정하는 SLAYOUT 절의 문법을 BNF 표기법으로 나타낸 것이다.

```
<slayout> ::= SLAYOUT <slayout_spec>
<slayout_spec>
  ::= [<root_region>],<region_spec>{,
    <region_spec>}
<root_region>
  ::= ROOTREGION(<width>,<height>
    [,<bgcolor>] )
<region_spec>
  ::= REGION( <rid>,<left>,<top>,
    <width>,<height> [,<bgcolor>] )
    [ORDER=<priority>] [ADJUST=<fit>]

<fit> ::= fill | hidden | meet | scroll | slice
<bgcolor> ::= BGCOLOR "=" <color-value>
<color-value> ::= <RGB-value> | <color-name>
```

그림 10 공간 배치를 위한 SLAYOUT 절의 문법

4.3 시간 배치를 위한 질의어 확장

멀티미디어 데이터의 시간 배치를 위해 TLAYOUT 절을 정의하고 이를 질의어에 추가함으로써 사용자가 결과에 대한 시간 배치를 정의할 수 있도록 한다.

TLAYOUT 절은 그림 11과 같이, 미디어 자체의 시간 속성과 출력될 창의 위치를 정하는 SHOW 절과 미디어간의 동기화 관계를 정의하는 SYNC 절로 나눈다.

```
<tlayout> ::= SHOW <show_spec> [SYNC <sync_spec>]
```

그림 11 TLAYOUT 절의 정의

SHOW 절에서는 추출된 객체들이 화면상에 얼마동안 나타날 것인지를 결정한다. 먼저 추출된 객체들이 놓일 미디어 창을 지정하기 위해 ON 키워드를 정의하고 얼마동안 화면상에 나타날 것인지를 결정하는 DURATION, 그리고 지연 시간을 지정하는 DELAY 키워드를 정의한다. 이 때 추출 객체의 ID는 명시적 식별자이며 select 절에서 정의한다.

SYNC 절은 SHOW 절에서 정의된 객체들이 어떠한 시간 관계를 가지는지를 기술한다. 본 논문의 시간 배치 모델링에서는 시간 관계를 병렬 관계, 순차 관계의 두 가지 형태로 정의하고 있는데, 병렬 관계와 순차 관계를 질의어에서 표현하기 위해 PAR와 SEQ를 정의한다.

동기화를 위한 사용자의 편의를 위해서 특별 동기화 키

워드도 START와 END 키워드를 따로 정의한다. 예를 들어 "PAR(M1 M2 0) START" 로 나타낼 경우 앞에서 정의한 M2의 시간 지연시간은 무시하고 m1과 m2를 동시에 시작한다. "PAR(M1 M2 0) END" 나타낼 경우는 M2가 창에 나타나 있는 동안에는 M1을 화면에서 유지한다. 이때 오디오와 비디오 같이 시간 속성을 가지는 미디어일 경우에는 반복되거나 중간에 중지하는 것을 의미한다. 아래 그림 12는 시간 배치를 위해 정의된 TLAYOUT 절의 문법을 BNF로 표현한 것이다.

```

<tlayout> ::= SHOW<show_spec>
           [SYNC <sync_spec>]
<show_spec> ::= <media_spec> | {,<media_spec>}
<media_spec> ::= <media_id>
               "ON" <region_id> [<delay>][
               <duration>]
<delay> ::= DELAY "=" <time>
<duration> ::= DURATION "=" <time>

<sync_spec> ::= {<seq_spec> | <par_spec>}
<seq_spec> ::= SEQ(<media_id>, <media_id>, <time>)
<par_spec> ::= PAR(<media_id>, <media_id>, <time>)
              [<special_keyword>]
<special_keyword> ::= START | END
    
```

그림 12 시간 배치를 위한 TLAYOUT 절의 문법

이상에서 멀티미디어 데이터를 질의하기 위해 멀티미디어 객체 형을 정의하고 질의어 상에서 결과로 추출될 객체의 프리젠테이션을 정의할 수 있도록 SQL을 확장하였다. 본 논문에서 제안하는 확장된 SQL을 MMSQL (MultiMedia Structured Query Language) 이라 부르도록 한다. MMSQL은 그림 13과 같이 기존의 SQL 구문 형식인 select, from where 구문 형식을 그대로 유지하면서 여기에 멀티미디어 프리젠테이션을 위한 새로운 구문인 present절이 추가된 형식을 가진다.

```

select attribute-list
from table-list
where condition
slayout spatial_layout
tlayout temporal_layout
    
```

그림 13 MMSQL의 구문 형식

MMSQL의 프리젠테이션 구문에서는 프리젠테이션 될 미디어를 지칭하기 위해 명시적 식별자를 사용한다.

명시적 식별자는 질의 결과로 추출된 미디어 각각에 대해 부여하는데, 이를 위해서는 그림 14와 같이 기존의 select절을 확장하여 사용자가 멀티미디어 데이터에 대해 명시적 식별자를 부여할 수 있도록 한다.

```

<select_stmt>:=SELECT <att_list> {,<att_list>}
<att_list> = <att_name> [":"<pid>]
<pid> = "("<media_type> <id>)"
<media_type>= "TEXT"|"IMAGE"|"
              "GRAPHIC"|"AUDIO"|"VIDEO"
    
```

그림 14 select 절의 확장

5. MMSQL 질의 처리 시스템의 설계 및 구현

앞에서 정의한 MMSQL은 기존의 DBMS에서 그대로 사용할 수 없기 때문에 MMSQL을 처리할 수 있는 별도의 질의 처리 시스템이 필요하다. 따라서 본 논문에서는 MMSQL로 작성된 질의어를 처리하기 위한 MMSQL 질의 처리 시스템을 설계하고 구현한다.

5.1 시스템 구조

그림 15는 MMSQL 질의 처리 시스템이 운용되는 멀티미디어 데이터베이스 시스템의 전체적인 구조를 보여준다.

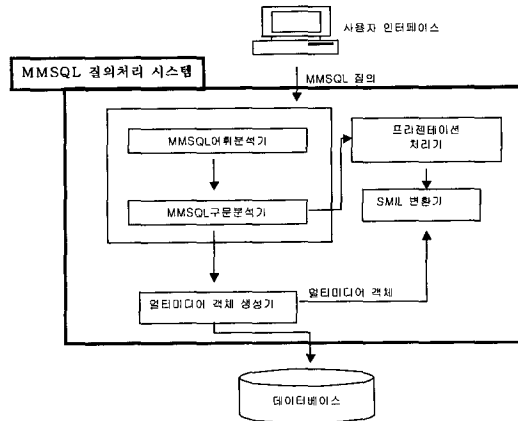


그림 15 MMSQL 질의 처리 시스템의 구조

5.2 구문 분석기의 설계

MMSQL 구문 분석기는 사용자로부터 입력받은 MMSQL의 구문을 분석한다. MMSQL 구문 분석기는 그림 16과 같이 구문 오류 검사기과 프리젠테이션 질의 추출기로 구성된다.

구문 오류 검사기는 전형적인 구문 오류 분석 과정을

그대로 따르는데, 어휘분석기에서 토큰(token)으로 분리된 토큰 리스트에 대해 MMSQL 문법을 사용하여 구문 오류가 있는지를 검사한다. 구문 오류가 없을 경우 프리젠테이션 질의 추출기를 통하여 MMSQL 사용자 질의에서 객체 추출 구문과 프리젠테이션을 기술한 구문을 분리하여 멀티미디어 객체 생성기와 프리젠테이션 처리기로 각각 전달한다.

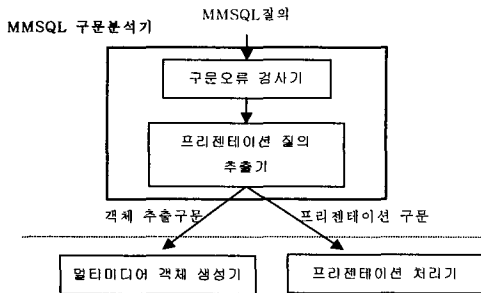


그림 16 구문 분석기의 동작과정

5.3 멀티미디어 객체 생성기의 설계

멀티미디어 객체 생성기에서는 사용자가 요구한 질의에서 프리젠테이션에 참여할 객체들을 생성하는데, 위의 그림 16에서 살펴볼 수 있는 바와 같이 구문 분석기에서 분리된 객체 추출 구문을 전달받아 필요한 객체들에 대한 정보를 저장하는 MEDIA_OBJECT_LIST 클래스를 생성한다(그림 17).

MEDIA_OBJECT_LIST 클래스		
속 성	속성 타입	설 명
Table_name	String	미디어가 속한 테이블 명
Satt_name	String	미디어가 정의된 속성의 이름
Satt_type	String	미디어 형
PID	String	프리젠테이션 시 사용될 미디어의 ID

그림 17 MEDIA_OBJECT_LIST 클래스의 구조

위에서 정의된 MEDIA_OBJECT_LIST 클래스는 미디어 객체 생성뿐만 아니라, 아래에서 살펴볼 SMIL 생성기에서 SMIL 문서를 구성할 때 미디어 요소를 정의하거나, 프리젠테이션 객체 생성기에서 프리젠테이션이 정해지지 않은 미디어를 찾아내어 해당 미디어에 대한 기본 프리젠테이션(default presentation)을 작성할 때에도 사용한다.

MEDIA_OBJECT_LIST 클래스에 저장된 정보는 MMSQL 구문 분석기를 통해 분리된 객체 추출 구문과

함께 미디어 데이터가 저장된 데이터베이스로 전달되어 필요한 미디어 객체를 추출하게 된다.

5.4 프리젠테이션 처리기의 설계

프리젠테이션 처리기는 사용자가 MMSQL에서 정의한 프리젠테이션 관련 구문들을 분석하여, 추출된 결과가 사용자가 원하는 형태로 보여질 수 있도록 프리젠테이션을 구성하는 부분이다.

프리젠테이션 처리기는 먼저, 전달된 MMSQL 문장에서 미디어가 위치할 미디어 창의 공간 배치에 대한 정보를 기술한 SLAYOUT 절을 분석하여 RESION 절에서 기술한 속성들을 저장하는 REGION_OBJECT 클래스를 생성한다(그림 18). 만약 사용자가 REGION을 하나도 정의하지 않았을 경우에는 시스템이 자동으로 ROOTREGION을 생성하여 REGION_OBJECT 클래스에 등록한다.

REGION_OBJECT 클래스		
속성	속성 타입	설 명
RID	String	미디어 창의 고유 식별자
Left	int	ROOTREGION으로부터의 X축 거리
Top	int	ROOTREGION으로부터의 Y축 거리
Width	int	미디어 창의 가로크기
Height	int	미디어 창의 세로크기
BGcolor	String	배경 색의 RGB값
Adjust	int	미디어 크기 조절 속성 정의
Order	int	미디어 창의 활성화 순서

그림 18 REGION_OBJECT 클래스의 구조

MMSQL에서는 SHOW 절을 통해 먼저 미디어 자체의 시간 속성을 나타낸 다음, SYNC 절을 사용하여 미디어간의 시간 동기화를 표현한다.

프리젠테이션 처리기는 SHOW 절에서 기술된 미디어가 놓일 창의 번호와 그 외 SHOW절의 시간 속성들을 가지는 그림 19의 SHOW_OBJECT 클래스를 생성한다.

SHOW_OBJECT 클래스		
속성	속성 타입	설 명
PID	String	미디어 창에 위치할 미디어 식별자
RID	String	미디어 창 식별자
Delay	Time	지연시간
Duration	Time	지속시간

그림 19 SHOW_OBJECT 클래스의 구조

다음으로, 시간 배치의 동기화 정보를 기술한 SYNC 절을 분석하여 동기화에 필요한 정보를 추출한다. 그림

20은 SYNC 절에 포함되어 있는 병렬관계에 필요한 속성을 저장하는 PAR_OBJECT 클래스의 구조이다. PAR 절에서는 미디어에 대한 시간 지연 값을 정할 속성과 시작 시점 또는 끝 시점을 동기화 시키는 특별 동기화 키워드를 저장할 속성이 추가적으로 필요하다.

PAR_OBJECT 클래스		
속성	속성 타입	설명
SID	String	기준이 되는 미디어의 식별자
DID	String	동기화 대상 미디어의 식별자
Delay	Time	시간 지연 값
Special	int	특별 동기화 연산자 0:START, 1:END

그림 20 PAR_OBJECT 클래스의 구조

순차관계 정보를 저장하는 SEQ_OBJECT 클래스도 PAR_OBJECT 클래스와 구조는 같지만 SEQ에서는 특별 동기화 연산자가 필요 없기 때문에 Special 속성이 없는 형태이다.

이상에서 MMSQL 질의 처리 시스템의 프리젠테이션 처리기에 대해 설계하였다. 프리젠테이션 처리기에서 생성된 각 클래스들은 SMIL 변환기로 보내어 저서 사용자가 질의어에서 정의한 프리젠테이션을 SMIL 문서로 표현할 때 사용된다.

5.5 SMIL 변환기의 설계

SMIL 변환기는 미디어 정보 생성기와 프리젠테이션 처리기를 통해 얻어진 정보를 이용하여 MMSQL 질의 결과를 SMIL 문서로 구성한다. 이 때는 SMIL 문서에 대한 구조 정보가 필요한데, 본 논문에서의 결과로 얻어질 수 있는 프리젠테이션의 형태에 맞추어 SMIL 문서 템플릿을 작성하고 템플릿으로부터 SMIL 태그(tag)에 필요한 정보들을 추출하여 사용한다.

SMIL 변환기는 먼저 그림 18의 REGION_OBJECT 클래스에 저장된 공간 배치 정보를 읽어 SMIL에서 공간 배치를 표현하는 LAYOUT 절을 작성한다. LAYOUT 절을 작성할 때에는 그림 21의 LAYOUT 절의 템플릿을 읽어

```
<LAYOUT>
  <ROOT_LAYOUT width="%Width" height="%Height" background_color="%BGcolor">
    <REGION "%RID" left="%Left" top="%Top" width="%Width" height="%Height" background_color="%BGcolor" fit="%Adjust" z-index = "%Order" />
  </LAYOUT>
```

그림 21 LAYOUT 절을 위한 템플릿

서 필요한 부분에 REGION_OBJECT 값을 위치시킨다.

다음은 미디어의 시간 배치를 기술하는데, 화면 출력될 각 미디어 객체의 시간 정보가 저장되어 있는 SHOW_OBJECT 클래스에서 인스턴스를 추출하여 그림 22와 같은 MEDIA 템플릿에 값을 채움으로써 미디어 절을 생성한다.

이때 media_type에 대한 정보는 MEDIA_OBJECT_LIST 클래스에서 얻는다.

```
<%media_type SRC="%PID" REGION="%RID" BEGIN="%Delay" DUR="%Duration" />
```

그림 22 미디어 절을 위한 템플릿

다음으로 SEQ_OBJECT 클래스와 PAR_OBJECT 클래스에서 동기화 정보를 읽어 SMIL 문서의 <SEQ> 절과 <PAR> 절을 구성한다.

SHOW_OBJECT 클래스에서는 미디어 객체의 지연 시간이 기술되어 있는데, 미디어 객체의 지연시간은 MEDIA1과 MEDIA2가 순차 관계일 때는 기준 시점으로부터의 지연시간이며, MEDIA1과 MEDIA2의 관계가 병렬 관계일 때에는 MEDIA2의 지연시간은 MEDIA1에 대한 상대적인 지연시간이 된다. 따라서 병렬 관계일 때는 MEDIA2의 미디어 절을 그림 23과 같이 MEDIA1과에 대한 상대적인 지연시간을 뜻하도록 BEGIN 속성 값을 변환한다. 이때 %Pid1은 MEDIA1의 식별자이며 %Delay 는 MEDIA2의 시간지연 값과 PAR_OBJECT 클래스에서 정의한 Delay 속성 값의 합이 된다.

```
<...BEGIN="%Delay" ... /> =>
  <...BEGIN="id(%Pid1)(%Delay)".../>
```

그림 23 병렬 관계에서의 BEGIN 속성의 변환

MMSQL에서는 특별한 동기화 연산자로 START와 END를 지정할 수 있다. 이 경우 그림 24의 알고리즘을 통해 MEDIA2의 begin과 duration 속성을 조정하여 시작 동기화와 끝 동기화를 한다.

```
알고리즘 : Special_sync_operator
get instance from PAR_OBJECT class
if special_operator = "START" then
  get media2 template
  set media2.delay = 0
  set par_object.delay=0
endif
```

```

if special_operator = "END then
  get media2 template
  set media1.playtime = media1.delay+media2.duration
  set media2.all_delay = media2.delay + par_object.delay
  set media2.duration = media1.playtime - media2.all_delay
endif

```

그림 24 특별 동기화 연산자 처리 알고리즘

최종적으로 SMIL_META 클래스에서 문서의 종결을 위한 템플릿을 추출하여 MMSQL 질의어에서의 프리젠테이션을 위한 SMIL 문서를 완성한다.

5.6 MMSQL 질의 처리 시스템 구현

이상에서 본 논문에서 제안하는 MMSQL을 위한 질의 처리 시스템을 설계하였다. MMSQL 질의 처리 시스템은 사용자가 MMSQL의 통해 필요한 미디어들을 추출하고 결과 미디어의 배치 즉, 프리젠테이션을 작성할 수 있도록 한다.

5.6.1 구현 환경

MMSQL 질의 처리기는 오라클의 PROC와 PL/SQL을 사용하여 DBMS와 MMSQL 질의 처리기의 인터페이스를 구현하였으며 MMSQL 질의 처리 시스템은 C언어 사용하여 구현하였다. 아래 표 2는 MMSQL 질의 처리기를 구현한 구현 환경을 보여준다.

표 2 MMSQL 질의 처리기의 구현 환경

항목	종류
서버시스템	SUN ultra sparc II
운영체제	solaris 8
하부 DBMS	ORACLE 8i
사용언어	C, PROC, PL/SQL

MMSQL 질의 처리 시스템은 DBMS와 독립적으로 동작하도록 설계되었기 때문에 MMSQL 질의 처리기와 DBMS간의 인터페이스만 재 정의하면 다른 종류의 DBMS에서도 운용할 수 있도록 이식성을 고려하였다.

5.6.2 DBMS와의 인터페이스 구현

본 논문에서는 하부 구조로 오라클 사의 Oracle 8i DBMS를 사용하였다. Oracle 8i는 객체-관계형 DBMS로서 사용자 정의형을 지원할 뿐만 아니라 BLOB(Binary Large Object) 데이터 저장 기능을 지원하는데, 본 논문의 미디어 데이터의 저장구조로 사용한다. 그러나 BLOB 데이터는 SQL을 통해 직접 접근할 수 없기 때문에 PROC나, PL/SQL 등으로 작성된 별도의 BLOB 처리 루틴을 필요로 한다.

본 논문에서는 앞에서 언급한 바와 같이 MEDIA_OBJ

ECT_LIST클래스에 저장되어 있는 미디어 정보와 MMSQL 구분 분석기에서 분리된 객체 추출 구문을 사용하여 그림 25와 같은 미디어 추출 루틴을 통해 필요한 미디어 데이터를 추출한다.

```

:
EXEC SQL EXECUTE
  declare
  lobd BLOB;
  stmt VARCHAR2(3000);
  BEGIN
      stmt := :sqlstmt;
      EXECUTE IMMEDIATE
      stmt INTO lobd;
      blob_it.my_blob := lobd;
      :total:= DBMS_LOB.GETLENGTH
      (blob_it.my_blob);
  END;
END-EXEC;
:

```

그림 25 미디어 추출 루틴

미디어 추출 루틴은 MMSQL 질의처리 시스템과 별도로 동작하며, 어떤 DBMS를 하부구조로 사용할 것인가에 따라 달라진다. 따라서 다른 종류의 DBMS를 사용할 경우 이 부분만을 재 작성하면 MMSQL 질의 처리 시스템을 수정하지 않고 그대로 사용할 수 있다.

5.6.3 MMSQL의 질의 예

사용자의 요구사항이 아래와 같을 때, 본 논문에서 정의한 프리젠테이션 문법을 사용해서 멀티미디어 데이터베이스에 질의하기 위해 작성한 질의는 아래와 같다.

- 크기가 500x400인 기본 창 내 제목이 "쉬리"인 영화의 내용과 영화를 설명 전문 그리고 영화제목 이미지를 출력하라. 단, 영화 제목 이미지가 나올 창은 시작점이 (20,20) 크기가 450X100이며, 설명이 위치할 창은 (20,160)에서 시작하여 크기가 200X240 이다. 또 영화가 나올 창은 시작점이 (250, 200) 크기가 200×170이다.

영화 내용과 영화 제목 이미지는 크기 배율을 조정하여 창의 크기에 맞도록 나타내며, 설명은 스크롤 바를 표시하라. 그리고, 영화의 제목과 내용은 기본 창이 화면에 활성화된 후 2초 후에 화면에 표시하고 10초 동안 화면에 유지하라. 내용은 영화의 제목이 시작된 후 3초 후에 화면에 출력하여 영화가 마칠 때까지 유지하라.

```

select t.Title.Content : (IMAGE image1),
       t.Story.Content : (FULLTEXT text1),

```

```

t.Mvideo.Content : (VIDEO video1),
from Movie t
where t.Mtitle = "쉬리"
layout ROOTREGION(500, 400, bgcolor=black),
      REGION(region1, 20, 20, 450, 100)
ADJUST=fill,
      REGION(region2, 20, 160, 200, 240)
ADJUST=scroll,
      REGION(region3, 250, 200, 200, 170)
ADJUST=fill
show imagel on region1 delay=2 duration=10,
      text1 on region2,
      videol on region3
sync PAR(imagel, text1, 0) END
      PAR(imagel, videol, 3) END
    
```

위의 질의를 수행한 결과는 그림 26과 같으며 그림 27은 최종적으로 얻어진 SMIL문서이다.

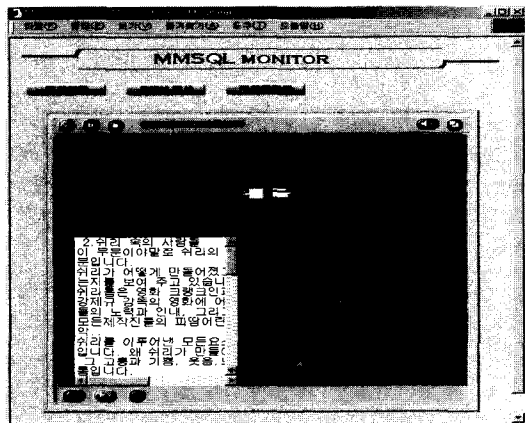


그림 26 질의결과 수행화면

5.7 과거 연구와의 비교

본 논문에서는 멀티미디어 프리젠테이션을 위해, 멀티미디어 데이터가 프리젠테이션 될 때 발생할 수 있는 공간 관계와 시간 관계의 모델링하고 필요한 속성들을 정의하였다. 이를 기반으로 MMSQL에서 사용할 시간 배치 기법과 공간 배치 기법을 제안함으로써 MMSQL에서 멀티미디어 프리젠테이션을 정의할 수 있도록 하였다.

멀티미디어 프리젠테이션을 질의어에서 제공하고자 하는 기존의 연구들에서는 SQL과 같은 표준화된 질의

```

<smil>
  <head>
    <meta name="SMIL document"
          content="MMSQL
          presentation" />
    <layout>
      <root-layout width="500" height="400"
                  background-color="black" />
      <region id="REGION1" left="20" top="20"
              width="450" height="100" fit="fill" />
      <region id="REGION2" left="20" top="160"
              width="200" height="240"
              background-color="white" fit="scroll" />
      <region id="REGION3" left="250" top="200"
              width="200" height="170" fit="fill" />
    </layout>
  </head>
  <body>
    <par>
      
      <text id="text1" src="text1.txt"
            region="REGION2" begin="2s" dur="10s" />
      <video id="videol" src="videol.rm"
             region="REGION3" begin="5s" dur="7s" />
    </par>
  </body>
</smil>
    
```

그림 27 질의결과 수행 시 얻어진 SMIL문서

어 형식을 따르지 않거나, SQL의 질의어 형식을 따르더라도 SQL의 기본 형식인 select, from, where 형식 내에서 프리젠테이션을 기술할 수 있도록 하기 때문에 프리젠테이션을 지원하도록 SQL을 수정하는데 많은 어려움이 따른다.

본 논문에서 제안하는 멀티미디어 데이터베이스 질의어인 MMSQL은 SQL 형식을 그대로 따르기 때문에 사용자가 쉽게 사용할 수 있으며, 객체 모델링과 프리젠테이션 모델링 과정에서 얻은 객체 모델을 기반으로 개발되었기 때문에 특정 미디어에 의존적이지 않으며 확장된 구문을 통해 멀티미디어 프리젠테이션을 효과적으로 기술할 수 있다. 또한 기존의 질의 처리기와 독립적으로 동작하도록 MMSQL 질의 처리기를 설계하였기 때문에 다른 응용에서 사용하기에 용이하다.

앞에서도 살펴본 바와 같이 멀티미디어 데이터를 정의하거나 질의어를 설계하는 대부분의 연구들은 비디오 데이터베이스 시스템, 오디오 데이터베이스 시스템과 같은 특정 미디어에 국한되거나, VOD 시스템이나 이미지 검색 시스템과 같이 특정 응용에 사용하기 위한 연구들

인데, 이러한 응용에서 질의를 통해 얻은 미디어들은 그 응용 내에서만 사용할 수밖에 없다. 또한 결과로 화면에 보여주기 위한 프리젠테이션 역시 응용에서 제공되는 전용환경을 통해서만 제공받을 수 있다.

질의 결과를 재 사용하기 위해서는 질의 결과를 표준화된 형태로 제공할 수 있는 방안이 필요한데, 여기에 관한 기존의 연구는 없는 실정이다. 본 논문에서는 MMSQL을 통해 얻어진 질의 결과와 프리젠테이션을 SMIL이라고 하는 표준화된 문서로 제공하기 때문에 SMIL 표준을 따르는 모든 응용에서 수정 없이 사용할 수 있는 방법을 제공한다.

아래 표 3은 본 연구와 과거연구와의 차이점을 요약하여 표로 나타낸 것이다.

표 3 기존의 연구와의 비교

구분	기존 연구	본 논문의 연구
모델링의 대상	· 오디오, 비디오와 같이 특정 미디어를 대상으로 함[4, 7, 21]	· 모든 멀티미디어 데이터를 대상으로 함
질의 언어	· SQL 형태를 따르지 않음[10, 14] · 응용에 제한 적임[4, 5, 13, 17, 21]	· 표준 질의어인 SQL 형태를 따름 · 기존의 질의 처리기와 독립적으로 동작하도록 설계하여, 다른 응용에서 쉽게 사용할 수 있음
프리젠테이션	· 질의 결과의 모든 프리젠테이션을 형태를 정의하지 못함[8, 9, 11, 12] · 기존의 SQL 구문의 수정이 필요[10, 14]	· 질의어 상에서 결과의 모든 프리젠테이션 형태를 정의할 수 있음 · 기존의 SQL 구문을 그대로 사용할 수 있는 형태로 개발
질의 결과의 재사용성	· 질의 결과의 재사용 방법을 제공하지 않음[4, 7, 10-14, 21, 22]	· 질의 결과를 SMIL 형태로 제공하기 때문에 결과를 다른 응용에서 사용하기 용이함

6. 결론

본 논문에서는 멀티미디어 데이터 프리젠테이션을 위하여 멀티미디어 객체를 화면에 보여줄 때 미디어 객체들간에 존재하는 시간 관계와 공간 관계를 모델링하고, 제안하는 프리젠테이션 모델을 효과적으로 지원할 수 있도록 SQL을 확장한 MMSQL을 설계하고 질의 처리기를 구현하였다. MSQL은 사용자가 보다 쉽게 사용할 수 있도록 SQL을 확장하는 형태로 설계하였으며, SQL 구문의 수정을 최소화하여 기존의 질의 처리기를 그대로 사용할 수 있게 함으로써 개발이 용이하도록 하였다.

MMSQL의 질의 결과는 W3C의 멀티미디어 프리젠테이션

표준인 SMIL 문서로 제공되므로, 널리 배포되어 있는 기존의 SMIL 플레이어들을 통해서 결과를 프리젠테이션할 수 있으며 다양한 멀티미디어 응용에서 결과를 사용할 수 있다.

본 논문의 결과인 멀티미디어 프리젠테이션 모델, 그리고 이를 지원하는 MMSQL은 응용 시스템의 하부구조인 데이터베이스 시스템의 측면에서 범용 모델을 제공하고자 하는 목적으로 개발되었기 때문에, 다양한 멀티미디어 응용 시스템에서 그대로 적용할 수 있다.

현재 개발된 MMSQL은 기존의 DBMS의 상위 구조로 동작하도록 설계되었기 때문에 하부 구조로 사용할 DBMS가 BLOB 데이터를 처리할 수 있을 경우에 동작 가능하다.

본 논문에서 제안하는 MMSQL은 멀티미디어 데이터 추출하고 결과의 시,공간 관계를 질의어에서 표현하는 방법을 중심으로 연구하였다. 이 이외에 효과적인 미디어 추출과 질의 수행 속도 향상을 위해서는 멀티미디어 데이터를 위한 인덱스 구조와 질의 최적화 알고리즘 등이 필요하며 이러한 점을 중심으로 향후 MMSQL의 질의 처리 성능을 높이기 위한 별도의 연구가 필요하다. 또한 본 연구에서는 저장된 멀티미디어 데이터에서 필요한 정보를 추출하는 데이터 조작어(Data Manipulation Language)를 중심으로 연구하였는데, 이와 아울러 멀티미디어 데이터에 대해 보다 편리하게 적용할 수 있는 멀티미디어 데이터 정의어(Data Definition Language)의 개발이 필요할 것으로 본다.

참 고 문 헌

[1] G. Blakowski, et. al., "A Media Synchronization Survey : Reference Model, Specification and Case Studies," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, Jan. 1996.

[2] International Organization for Standardization, SQL3, ISO/IEC JTC1/SC21/WG3 DBL, 1994.

[3] A. M. Alashqur, Stanley Y. W. Su, Herman Lam: OQL: A Query Language for Manipulating Object-oriented Databases. VLDB 1989 pp.433-442.

[4] Hacid, M., Declair, C., Kouloumdjian, J., "A database approach for modeling and querying video data," IEEE Transactions on Knowledge and Data Engineering, Vol. 12 No. 5, pp. 729 -750, Sept.-Oct. 2000.

[5] Nepal, S., Ramakrishna, M.V., "Query processing issues in image (multimedia) databases," Proceedings., 15th International Conference on Data Engineering, pp. 22 -29, 1999.

- [6] Graciela Gonzalez, Chitta Baral, Amarendra Nandigam. "SQL+D: Extended Display Capabilities for Multimedia Database Queries," University of Texas at El Paso Dept. CS, ACM Multimedia 98-Electronic Proceedings. 1998.
- [7] S.Adali, K.S.Candan, V.S.Subrahmanian, "Advanced video information system: Data Structures and query processing," ACM Multimedia Journal, 1995.
- [8] Bertino, E., Ferrari, E., Stolf, M., "MPGS: an interactive tool for the specification and generation of multimedia presentations," IEEE Transactions on Knowledge and Data Engineering, Vol. 12 No. 1, pp.102-125, Jan.-Feb. 2000.
- [9] Wong, R.K., "Heterogeneous data integration and presentation in multimedia database management systems," IEEE International Conference on Multimedia Computing and Systems, Vol.2, pp. 666-671, 1999.
- [10] A. F. Cardenas, I. T. Jeong, R. K. Taira, R. Barker, and C. M. Breant, "The knowledge-based object-oriented PICQUERY+ language," IEEE Transactions on Knowledge and Data Engineering, Aug. 1993.
- [11] N. Dimitrova and F.Golshani. "EVA: A query language for multimedia information systems," Proc. of Multimedia Information Systems, Tempe, Arizona, Feb. 1992.
- [12] Day, Y.F., Wasfi Al-Khatib, Paul, R., Ghafoor, A., "Specification of a query language for multimedia database systems," Proceedings. International Workshop on Multimedia Software Engineering, pp.111-118, 1998.
- [13] W.W.Chu, I.T.Jeong, R.K.Taira, "A semantic modeling approach for image retrieval by content," The VLDB Journal, 3, pp. 445-477, 1994.
- [14] Nael B. Hirzalla, Ahmed Karmouch, "A Multimedia Query Specification Language," Multimedia Database System, Kluwer Academic Pub, 1996.
- [15] T.C.T.Kuo, A.L.P. Chen, "A content-based query language for video databases," Proc. of IEEE International Conference on Multimedia Computing and Systems, pp. 456-461, Hiroshima, Japan, Jun. 1996.
- [16] H. Arisawa, T. Tomii, and L.Salev. "Design of multimedia database and a query language for video image data," Proc. of IEEE International Conference on Multimedia Computing and Systems, pp. 462-467, Hiroshima, Japan, Jun 1996.
- [17] R. Hjelsvold, R. Midstraum, "Modeling and Querying Video Data," Proc. of the 20th VLDB Conference, No. 4. pp. 629-643, Santiago, Chile, 1993.
- [18] 나연목, "멀티미디어 데이터베이스의 프리젠테이션을 위한 멀티미디어 검색 질의어", 한국정보처리학회 논문지, 제4권 제5호, 1997.
- [19] Hoschka, P., "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," PR-smil-19980409, W3C, 1998.
- [20] Junghwa Lee, Yoohyun Park, Kyongsok Kim, "SQL Extension for supporting Multimedia data," International Conference on Multimedia Modeling '99, pp.147-165, Canada, Sep 1999.
- [21] V. Oria, M. T. Ozsu, X. Li, L. Liu, J. Li, Y. Niu, and P. J. Iglinski, "Modeling images for content-based queries: The DISIMA approach," Proc. of 2nd DEFINING VIEWS IN AN IMAGE DATABASE SYSTEM 19 International Conference of Visual Information Systems, pp. 339-346, San Diego, California, Dec. 1997.
- [22] Jia-Ling Koh, A.L.P. Chen, "A Query Language and Interface for Integrated media and Alphanumeric Database System," 1996.



이 중 화

1992년 부산대학교 전자계산학과 학사
 1995년 부산대학교 전자계산학과 석사
 2001년 부산대학교 전자계산학과 박사
 2002년 3월~현재 동의대학교 컴퓨터소프트웨어공학부 전임강사. 관심분야는 멀티미디어 데이터베이스, 객체지향 데이터베이스, 가상강의시스템, 한글정보처리