

# 객체지향 데이터베이스 기반의 가상현실 시스템 \*

원석진\* 차상균  
서울대학교 전기공학부

## Virtual Reality Systems based on Object-Oriented Databases

Won Seok-jin Sang Kyun Cha  
School of Electrical Engineering,  
Seoul National University.

### 요약

기존의 가상현실(Virtual Reality) 시스템들은 가상세계(Virtual World)에 관한 데이터를 대부분 파일의 형태로 관리하고 있다. 가상현실에 대한 연구가 진전되어 공유가상세계(Shared Virtual World)를 기반으로 하는 대규모 응용분야가 나타나면서, 데이터의 효율적 관리와 공유를 위한 상위 레벨의 데이터 관리층이 필요하게 되었다. 본 논문에서는 가상세계 데이터의 관리를 위한 객체지향 DBMS(Database Management System)[1]의 사용을 제안한다. 그리고 데이터베이스에 저장된 지속객체(Persistent Object)[2]를 공유하는 가상현실 응용시스템의 구조를 연구한다.

## 1. 서론

가상현실은 컴퓨터가 생성하는 감각적 경험을 통해 인간이 현실감을 느끼도록 하는 기술이다. 현재 가상현실 기술은 군사용 시뮬레이터, CAD(Computer-Aided Design), 원격작업(Telepresence), WWW(World Wide Web) 등 여러 분야에서 사용되고 있다. 가상현실 기술이 발전하면서 여러 참가자가 동시에 가상세계에서 상호 작용을 하는 공유가상세계 기반의 대규모 시스템에 대한 연구가 진행되고 있다[3][4]. 이러한 시스템의 구현을 위해서 가상세계 데이터의 공유와 효율적 관리에 대한 필요성이 증가하게 되었다. 그러나 기존의 시스템은 파일의 형식으로 데이터를 관리하므로 그 한계를 가진다. 본 논문에서는 객체지향 DBMS를 사용하여 가상세계 데이터를 지속객체의 형태로 관리하고 공유하는 가상현실 시스템을 연구한다.

본 논문의 구성은 다음과 같다. 2절에서는 기존 가상현실 시스템의 데이터 관리 방법과 그 문제점, 그리고 객체지향 DBMS를 이용한 데이터 관리의 장점을 서술한다. 3절에서는 지속객체의 구현 방법과 실제로 구현된 지속객체의 구조를 설명한다. 4절에서는 지속객체를 공유하는 가상현실 응용시스템의 구조를 기술한다. 5절에서는 본 논문의 내용을 요약하고 앞으로의 연구 과제를 제시한다.

## 2. 가상현실 시스템의 데이터 관리

### 2.1 기존 가상현실 시스템의 경우

기존 시스템들은 가상세계의 데이터를 대부분 파일의 형태로 다룬다. 예를 들어 미국의 Naval Postgraduate School에서 연구 개발한 군사용 분산 시뮬레이션인 NPSNET는 2500km<sup>2</sup> 면적의 지형 데이터를 2500개의 데이터 파일로 나누어서 관리한다 [5][6]. 스웨덴에서 연구 개발한 DIVE의 경우 파일을 이용하여 지속객체를 관리한다.[7]

데이터를 파일 형태로 관리하면 다음과 같은 문제점이 발생할 수 있다. 첫째, 데이터의 규모가 커지면 데이터 파일의 수가 증가하여 그 관리가 어렵다. 둘째, 가상현실 시스템이 특정한 파일 형식에 종속된다. 셋째, 응용프로그램의 요구에 따라 데이터 모델을 변경하거나 재구성하는 것이 어렵다. 넷째, 여러 응용프로그램의 데이터 공유가 어렵다.

\* 본 연구는 자동제어특화연구센터와 국방과학연구소의 지원에 의해 수행되었음.

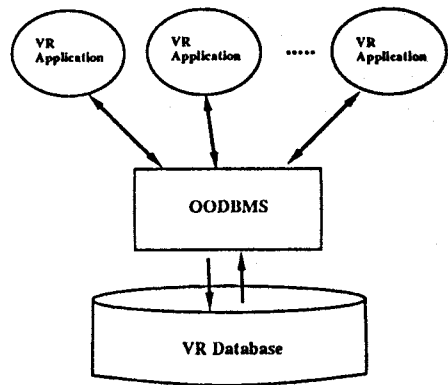


그림 1: 객체지향 DBMS를 이용한 가상현실 시스템

대규모의 가상현실 시스템을 구현할 경우에는 이러한 문제점을 해결하기 위한 상위 레벨의 데이터 관리층이 요구된다.

### 2.2 객체지향 DBMS

본 논문에서는 대규모 가상현실 시스템의 데이터 관리에 객체지향 DBMS를 사용할 것을 제안한다. 그림 1은 객체지향 DBMS를 기반으로 한 가상현실 시스템의 구조를 나타내고 있다. 가상세계 데이터는 객체지향 DBMS가 관리하고, 여러 응용프로그램들은 이를 통하여 데이터에 접근하도록 되어 있다.

#### 2.2.1 객체지향 DBMS의 특징

객체지향 DBMS는 객체지향 프로그래밍 패러다임용 데이터베이스 시스템에 적용한 것이다. 데이터는 객체(Object)와 관계성(Relationship)으로 모델링되어 지속객체의 형태로 데이터베이스에 저장된다. 기존의 관계형 DBMS에 비하여 실제계를 모델링한 프로그램이나 객체지향 프로그램에 적합한 구조를 가진다[1]. 현재 ODMG(Object Data Management Group)라는 표준 프로토콜이 제안되어 있다[8].

본 연구에서는 상용 객체지향 DBMS인 Objectivity/DB를 사용하였다. Objectivity/DB는 지속객체를 지원하며, 객체간의 관계성을 Association이라는 용어로 정의한다. 응용프로그램은 Objectivity/DB를 이용하여 지속객체에 직접 접근할 수 있다[2].

### 2.2.2 객체지향 DBMS 사용의 장점

가상현실 시스템에서 객체지향 DBMS를 사용할 경우, 가상세계 데이터베이스에 통합하여 관리할 수 있다. 또한 데이터를 여러 계층으로 나누고 필요에 따라 계층 구조를 변경시키는 것이 용이하다. 다른 중요한 장점 중 하나는 여러 응용프로그램이 데이터 모델과 Instance라는 두 가지 측면에서 데이터를 공유할 수 있다는 것이다.

먼저 데이터 모델의 공유 측면에서 보면 다음과 같은 장점이 기대된다. 첫째, 객체지향 데이터 모델(지속객체)을 직접 데이터로 다루기 때문에 특정 데이터 파일 형식에 종속되지 않는다. 둘째, 객체간의 관계성에 대한 저장, 관리물 객체지향 DBMS가 처리하므로 응용프로그램 작성자의 부담을 줄일 수 있다. 셋째, 데이터 모델의 변경, 개선이 용이하다. 예를 들면 클래스 Versioning 기술을 응용하여 지속객체의 Evolution을 구현할 수 있다[2].

Instance의 공유 측면에서는 다음과 같은 장점이 기대된다. 첫째, 여러 응용프로그램이 하나의 데이터베이스를 공유함으로써 저장 공간을 절약할 수 있다. 둘째, 여러 응용프로그램이 가상세계의 상태를 공유할 경우, 공유 가상세계의 구현에 도움을 줄 수 있다. 이때 트랜잭션을 사용한 공유 가상세계의 일관성 유지가 필요하다[9].

## 3. 가상세계 지속객체

가상세계의 객체들을 객체지향 데이터베이스에 저장하기 위해서는 지속객체로 구현해야 한다. 이 절에서는 Objectivity/DB를 사용하여 지속객체를 구현하는 방법과 실제로 구현된 지속객체의 구조를 설명한다.

### 3.1 구현 방법

Objectivity/DB의 경우, 일반적인 객체를 지속객체로 구현하려면 다음 과정이 필요하다[2]. 우선 Objectivity/DB가 제공하는 ooObj 클래스로부터 상속을 받아야 한다. 또한 다른 지속객체에 대한 관계성을 나타내기 위해서는 Objectivity/DB에서 제공하는 Association의 기능을 이용해야 한다. 지속객체는 DDL(Data Definition Language) 스키마 파

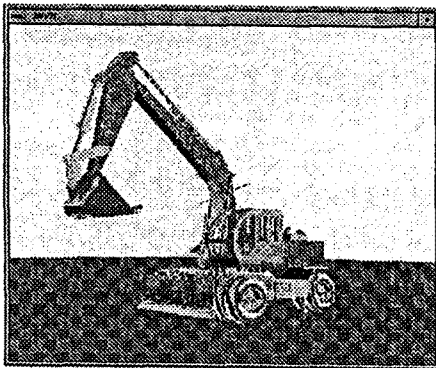


그림 2: HIVR의 실행 예

일에 정의되고 DDL 실행시기가 이것을 처리하여 일반적인 C++ 코드로 변환시킨다. 응용프로그램이 지속객체에 접근하려면 데이터베이스안에서의 지속객체 위치를 나타내는 OID(Object Identifier)를 사용한다. ooHandle과 ooRef라는 특별한 클래스를 통해 OID를 사용할 수 있다. 아래 코드는 일반적인 객체와 지속객체를 비교한 것이다.

```
// 일반 객체
class HE { // 중장비(Heavy Equipment)
...
// 다른 객체에 대한 포인터
BODY *body;
CAB *cab;
WHEEL *wheel;
...
};

// 지속객체
class dbHe : public ooObj {
...
// 다른 지속객체에 대한 Association
ooHandle(dbBody) body; // 1 : 1
ooHandle(dbCab) cab;
ooHandle(dbWheel) wheel[]; // 1 : N
...
};
```

## 3.2 지속객체의 구현

### 3.2.1 HIVR

본 연구에서는 삼성 종합 기술원에서 만든 중장비 실제 검증용 위한 HIVR(Heavy Industry Virtual Reality)이라는 프로토타입 가상현실 시스템을 이용하여 가상세계의 지속객체를 구현하였다. HIVR 시스템은 SGI사의 GL(Graphic Library)을 사용하였으며[10], 포크래인용 모델링한 HE(Heavy Equipment) 클래스를 가상세계 객체로 가지고 있다. HE 클래스는 dxl파일을 사용하여 구현되었으며, 약 10000개의 폴리곤으로 이루어져 있다. 그림 2는 HIVR을 실행한 모습이다.

### 3.2.2 dbHe 클래스

일반적인 객체인 HE 클래스를 사용하여 지속객체인 dbHe 클래스를 구현하였다. 그림 3은 dbHe 클래스의 계층 구조

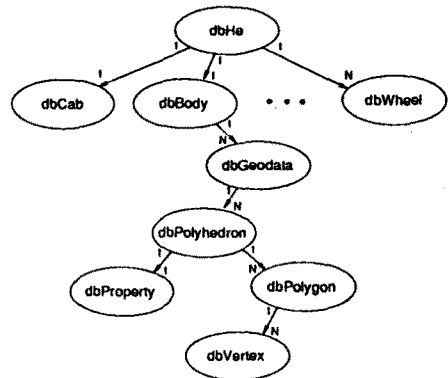


그림 3: dbHe 클래스의 계층 구조

