

# 형태, 기능, 행위를 고려한 가상현실 객체 재사용

김덕남 LG전자 안양연구소 dnkim01@lge.com  
김정현 포항공대 컴퓨터공학과 가상현실 연구실 gkim@postech.ac.kr

## VR Object Reuse based on Form, Function and Behavior

Deoknam Kim, LG Electronics, Inc.

Gerard Jounghyun Kim, VR Lab. Dept. of Computer Science and Engineering POSTECH

### 요 약

Code and object reuse is big concern for fast and efficient Virtual Reality (VR) system development. Many VR packages offer reasonably nice abstractions for various functionalities. That is, software reuse at the functional level is well practiced. Many geometry models are rarely used for main "characters" but mostly for incidental and decorative objects. This is because the main "character" objects usually exhibit certain behavior incompatible with the way the geometric model is organized. Kim has made a clear distinction between form, function and behavior[1]. This naturally lends to a reuse method at the level form, function and behavior.

#### 1. 서론

코드나 객체의 재사용은 빠르고 효율적인 가상현실 시스템을 개발하고 유지보수 하는 데 큰 관심거리이다. 여러 하위 단계의 소프트웨어 패키지들이 개발되어 개발자가 재사용하기 쉬운 API[3,4,5,6,8]를 제공하여 하위단계의 상세한 내용을 숨기고 추상화 하여 개발자의 부담을 덜어주고 있다[7,9]. 이러한 패키지들은 장면 그래프 구성, 시점 변화, 여러 장치를 가상현실 애플리케이션에 통합, LOD(Level Of Detail), 다양한 모델 파일 읽기 등 일반적인 가상현실 시스템에 필요한 다양한 기능들을 제공한다. 또한 가상현실 애플리케이션을 개발할 때 "형태"는 많은 개발자들에 의해 재사용되고 있다. 모델링 전문가에 의해 개발된 다양한 모델 파일들은 현재 WWW(World Wide Web)의 갤러리 사이트나 모델 파일을 판매하는 회사 등에서 제공되고 있으며, 많은 가상 현실 개발자에 의해 사용되고 있다. 그러나 이 모델 파일들은 가상현실 애플리케이션을 위한 주요 가상현실 객체로 사용되지 못하고 주로 부수적이거나 장면을 장식하기 위해 사용되고 있다. 이것은 가상현실 객체가 수행할 행위가 형태의 조직과는 적합하지 않기 때문이다. 만약 기능적으로 움직이는 사람을 생각하면 형태적으로 따로 분리된 팔다리가 필요하다.

소프트웨어 공학에서는 특정 도메인(domain)에 대해 도메인 엔지니어링을 통하여 얻은 산출물을 그 도메인 내에서 애플리케이션을 개발할 때 재사용하는 연구가 진행되어 왔다[2].

본 논문에서는 특정 도메인 내에서 가상현실 객체를 재사용하는 방법론을 제안한다. 제안하는 방법론에서는 가상현실 객체가 형태(form), 기능(function), 행위(behavior)로 이루어져 있다고 하는데 초점을 맞추고 있다. 형태는 객체의 외부 형태와 그들의 구조 및 장면 구조를 말한다. 기능은 외부에서 발생한 사건들에 반응하거나 객체의 본질적인 행위를 완수하기 위한 기본적인 행위

(action)이다. 행위는 가상현실 객체의 상태가 어떻게 동적으로 바뀌고 그들의 기능을 언제 수행하는 지에 대한 것이다. 이것은 형태, 기능, 행위를 가진 하위 객체(sub-object) 즉, 컴포넌트(component) 단계에서의 재사용을 가능하게 한다.

본 논문에서 제안하는 방법론은 특정 도메인 내의 가상현실 객체에 대해서 도메인 엔지니어링을 통해 재사용 가능한 단위의 "컴포넌트"를 개발한다. 이 도메인 내의 가상현실 객체를 개발할 때 이들 라이브러리와 된 컴포넌트를 선택하여 형태, 기능 및 행위를 가진 가상현실 객체를 빠르고 쉽게 개발할 수 있을 것이며, 이를 위하여 재사용 관련 지원 도구인 CCTV(Component Combination Tool for VR)를 개발하였다.

#### 2. 관련 연구

컴퓨터 그래픽스나 애니메이션 프로그램들이 이용하는 하위 단계의 시뮬레이션 프로그래밍 패키지나 라이브러리들은 가상 환경의 기능과 행위를 구현할 수 있도록 제공하고 객체 중심적인 프로그래밍, 가상현실 장치들과의 통신을 제공한다[3,4,5,6,7,8]. 그러나 이들은 모두 전체 시스템에 관련된 기능적인 객체만을 재사용하도록 개발되었다. Alice[10]는 가상현실 객체를 모델 파일로부터 형태를 개발하고, 스크립트(script) 형식으로 객체에게 행위를 부여할 수 있는 시스템이다. 하지만 객체 형태의 구조나 객체 각 부분별 형태의 구조에 대한 명시가 없어 각 부분을 조합해서 새로운 객체를 구성할 수 없다. 그래서 다양한 형태, 행위를 가진 객체 구성을 통한 참신한 콘텐츠(content) 개발이 힘들다. 이는 이 시스템이 구조적으로 형태 정보와 기능을 꼭 결합해 놓았기 때문이다. 형태와 기능이 결합된 상태에서의 재사용성은 높으나 재사용을 통한 새로운 객체 개발에는 유연성이 낮다.

3. 재사용 방법론

3.1 도메인 엔지니어링

도메인 엔지니어링을 통하여 주어진 시스템의 군(family)에 대해서 애플리케이션을 개발할 때 재사용 가능한 컴포넌트의 집합인 라이브러리를 얻게 된다. 라이브러리, L,은 다음과 같이 정의된다.

$L = (\text{set of SCs, set of FCs, set of CCs})$

SC: spatial component

FC: functional component

CC: composite component

도메인 엔지니어링을 통하여 개발된 컴포넌트는 공간적 컴포넌트(spatial component), 기능적 컴포넌트(functional component), 합성된 컴포넌트(composite component) 등 3가지 종류가 있다. 각각의 컴포넌트에 대해서 살펴보자.

3.1.1 공간적 컴포넌트

공간적 컴포넌트는 선택 가능한 여러 형태와 이 컴포넌트 내의 형태에 영향을 미치는 기능과 행위를 가지고 있다. 공간적 컴포넌트, SC,는 다음과 같이 정의된다.

$SC = (cid, G, M, C)$

① cid는 공간적 컴포넌트(SC)의 유일한 이름이다.

②  $G = (\text{set of MFs, SG})$

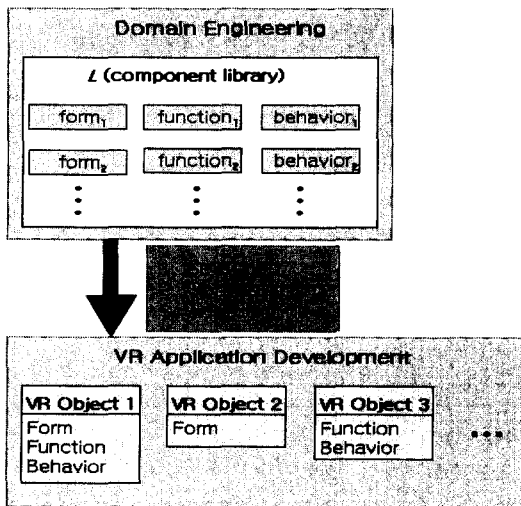
MF: model file

SG: scene graph

공간적 컴포넌트는 다양한 형태(MF)를 가지고 있다. 이들 형태는 이후에 가상현실 객체를 개발할 때 선택됨으로써 다양한 형태의 가상현실 객체를 개발할 수 있게 한다. 그런데 이들 형태(MF)는 컴포넌트 내의 기능과 행위(M)에 적합해야(compatible) 한다. 즉 형태(MF)는 컴포넌트 내의 기능과 행위가 정의될 수 있도록 특정한 장면 그래프(SG)를 가져야 한다. 장면 그래프는 트리로 표현되는 데, 트리가 나타내는 것은 움직임 계층(Motion Hierarchy)을 의미한다. 장면 그래프, SG,는 다음과 같이 표현한다.

$SG = (\text{ChildrenOf}(node_1) = node_2, node_3, \dots, node_n$

$\text{ChildrenOf}(node_2) = node_a, node_b, \dots, node_m$   
 ... )



[그림1] Overall Process of Reuse Method

③  $M = (\text{set of Fs, set of Bs})$

F: function, B: behavior

$F = (f, fc_1, fc_2, \dots, fc_n), B = (b, bc_1, bc_2, \dots, bc_m)$

F와 B는 공간적 컴포넌트 내의 형태(G)에 영향을 미치는 기능과 행위이다. f와 b는 각각 기능과 행위의 이름이며  $fc_1, fc_2, \dots, fc_n$ 는 f의 인자이며  $bc_1, bc_2, \dots, bc_m$ 는 b의 인자이다.

④  $C = (ME(f_1, f_2, \dots, f_i, b_1, b_2, \dots, b_m), \dots)$

C는 제약조건(constraint)을 표현한다. ME는 공간적 컴포넌트 내에서 함께 실행될 수 없는 기능이나 행위의 상호 배제(mutual exclusion) 관계를 표현한다.  $f_1, f_2, \dots, f_i, b_1, b_2, \dots, b_m$  ( $f_x$ : 기능 이름,  $b_x$ : 행위 이름)은 함께 실행될 수 없다.

3.1.2 기능적 컴포넌트

기능적 컴포넌트는 형태는 없는 기능과 행위로 이루어진 컴포넌트로서 기능이나 행위는 이 컴포넌트를 구성 요소로 하는 상위 객체 전체의 형태에 영향을 미친다. 그리고 기능적 컴포넌트의 기능이나 행위는 형태의 장면 그래프 구조에 상관없이 정의될 수 있다. 기능적 컴포넌트, FC,는 다음과 같이 정의된다.

$FC = (cid, M, C)$

① cid는 기능적 컴포넌트(FC)의 유일한 이름이다.

②  $M = (\text{set of Fs, set of Bs})$

F: function, B: behavior

$F = (f, fc_1, fc_2, \dots, fc_n), B = (b, bc_1, bc_2, \dots, bc_m)$

F와 B는 상위 객체 전체에 영향을 미치는 기능과 행위이다.

③ C는 공간적 컴포넌트의 것과 동일하다.

3.1.3 합성된 컴포넌트

한 컴포넌트 내에서는 정의가 불가능하고 2개 이상의 컴포넌트 내의 기능이나 행위의 조합으로 가능한 행위가 존재할 수 있다. 합성된 컴포넌트, CC,는 다음과 같이 정의된다.

$CC = (cid, M, C)$

① cid는 합성된 컴포넌트(CC)의 유일한 이름이다.

②  $M = (\text{set of Bs})$

$B = (b, bc_1, bc_2, \dots, bc_m)$

b는 두개 이상의 컴포넌트의 기능이나 행위로 가능한 합성된 행위의 이름이며  $bc_1, bc_2, \dots, bc_m$ 은 b의 인자이다.

③  $C = (MD(C_1, C_2, \dots, C_n))$

$C_x$ 는 공간적 컴포넌트 혹은 기능적 컴포넌트의 이름(cid)이다. 합성된 컴포넌트 내의 행위(M)는 "어느 컴포넌트"의 기능이나 행위로 구성되어 있는지를 알 수 있어야 한다. 즉 C는 상호의존 관계(mutual dependency relationship)를 표현한다.

3.2 가상현실 객체(VR Object) 개발

주어진 컴포넌트 라이브러리부터 [그림1]에서처럼 선택, 조합, 적용을 통하여 새로운 객체를 개발할 수 있다. 개발하려는 가상현실 객체의 기능과 행위를 고려하여 컴포넌트 중에서 이들 기능과 행위를 만족하는 컴포넌트를 선택한다. 즉 필요한 컴포넌트를 구성(composition)하게 된다. 컴포넌트를 선택한 후, 선택된 공간적 컴포넌트에 대

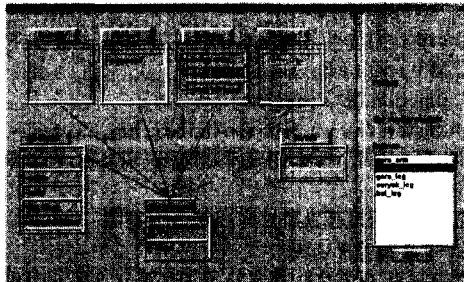
해서 형태를 선택한다. 이렇게 선택된 컴포넌트를 조합하여 새로운 가상현실 객체를 생성하게 된다. 조합은 다음과 같이 정의된다.

$$O = C_1 + C_2 + \dots + C_n$$

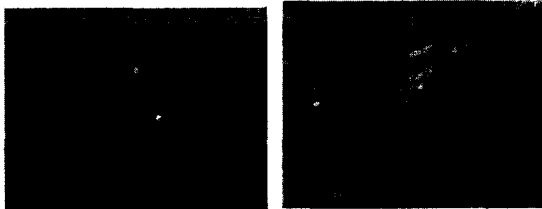
O는 새롭게 생성되는 가상현실 객체이며  $C_n$ 는 컴포넌트이다. 생성된 객체, O,를 적용(A)시켜 완성된 가상현실 객체(O')를 얻는다.

$$O' = A(O)$$

단순히 조합된 컴포넌트들은 새로운 애플리케이션에서 적용하기에는 충분하지 않다. 그래서 형태, 기능, 행위가 적절하게 커스텀되어야 한다. CCTV는 컴포넌트를 선택하고 조합하여 적용시켜 새로운 객체를 만들어 가시화하고 실행시켜 검증할 수 있게 하고 완성된 가상현실 객체에 대한 코드를 자동으로 생성하여 가상현실 시스템 개발자가 쉽고 빠르게 가상현실 객체를 개발하도록 하였다. [그림 2]는 CCTV에서 서로 다른 형태, 기능, 행위를 조합하여 새로운 가상현실 객체를 생성하는 과정을 보여주고 있다. [그림 3]은 CCTV의 시뮬레이션 환경에서 합성된 캐릭터를 검증하는 모습을 보여 주고 있으며, [그림 4]는 이 논문의 방법론에 의해 개발된 가상현실 객체를 간단한 3D 게임에 적용한 모습이다.



[그림2] Snapshot of CCTV: Composing different reusable forms, functions and behaviors to create an animation character



[그림3]

[그림4]

[그림 3] A newly composed character tested in CCTV's Simulation Environment

[그림 4] The new character used in a Simple 3D game

#### 4. 결론과 향후 연구 방향

이 논문에서는 형태, 기능, 행위를 섞어서 골라 맞춤(mixing and matching)으로써 새로운 가상현실 객체를 개발하는 재사용 방법론을 소개하였다. 제안된 재사용 방법론은 가상객체의 형태, 기능, 행위의 동시 개발(concurrent development), 계층적, 점진적 모델링에 기반을 두고 가상현실 시스템을 설계하는 구조적(structured) 접근법이다. 향후 연구 방향은 제안

된 구조적 개발과 재사용 방법론을 계속 연구하여 멀티모달 인터랙션(multimodal interaction), 몰입감(presence), 성능 튜닝(performance tuning) 등과 같은 가상현실 시스템의 여러 분야에 확장하는 것이다.

#### 5. 참고문헌

- [1] G. Jounghyun Kim, K. C. Kang, H. Kim and J. Lee, "Software engineering of virtual world" in *Proceedings of the ACM symposium on Virtual Reality Software and Technology*, pp. 131-139, 1998
- [2] Kyo C. Kang, S. Kim, J. Lee, K. Kim, "Form: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures", *Annals of Software Engineering*, 5, pp. 143-168, 1998
- [3] C. Shaw, J. Liang, M. Green, and Y. Sun, "The Decoupled Simulation Model for Virtual Reality Systems" *Proceedings of the ACM Computer Graphics Conference (SIGGRAPH '92)*, pp. 321-328, 1992
- [4] Vega User's Manual, Paradigm Simulation, Inc., 1996
- [5] IRIS Performer Programming Guide, Silicon Graphics, Inc., 1994
- [6] Then Inventor Mentor, Josie Vernecke, Open Inventor Architecture Group, 1994
- [7] Java3D API, Sun Microsystems, Inc., (Available at <http://java.sun.com/products/java-media/3D>).
- [8] WorldToolKit Reference Manual, Sense8 Corporation, 1996
- [9] P. Strauss and R.Cargey, An Object-Oriented 3D Graphics Toolkit, *Proceedings of SIGGRAPH 92*, pp. 341-349, 1992
- [10] R. Pausch, Alice: Easy to Learn Interactive 3D Graphics, (Available at <http://alice.cs.cmu.edu>).
- [11] J. Sametinger, Software Engineering with Reusable Components, Springer, 1997
- [12] Clemens Szyperski, Component Software Beyond Object-Oriented Programming, Addison-Wesley, 1999
- [13] Deok-Nam Kim, "VR Object Reuse based on mixing and matching form, function and behavior", *Unpublished Master Thesis, Pohang Univ. of Science and Technology*, January 2001.