

**** 실내공간구성을 위한 시각 프로그래밍 언어 기반 3차원 가상현실 저작도구 개발에 관한 연구

Developing a Visual Programming Language-based Three-dimensional Virtual Reality Authoring Tool to Compose Virtual Interior Space

박현수* / Park, Hyeon-Soo 김지인*** / Kim, Jee-In
박성준** / Park, Sungjun 박재완**** / Park, Jae Wan

Abstract

This paper presents an attempt to develop a visual programming language-based 3D virtual reality authoring tool intended to compose virtual interior space. The rapid development of digital technology and the wide spread of the Internet have expanded the different uses of virtual reality in a number of applications ranging from interior design to building maintenance. In particular, the construction of cyber spaces based on existing interior spaces is becoming increasingly important. Current research, however, remains at the level of converting 3D models into virtual reality models, despite practitioners' needs for structural space models. Moreover, commercial tools to build virtual reality space have the disadvantage of targeting people who have professional knowledge of computer programs and computer graphics. Accordingly, the 3D virtual reality authoring tool developed in this research - called the VESL system - enables virtual and structural space to be easily composed using intuitive and interactive visual interfaces, which are based on visual programming techniques. The VESL system also provides an XML based semantic description of interior space, to be used to describe interior space information. We anticipate that the virtual reality spaces composed by this system will be of considerable use in the fields of architecture and interior design. Further research issues identified at the end of the research include developing a converter/filter for transforming Internet virtual reality standard language, or VRML, and evaluating the application of the system for practical use.

키워드 : 가상현실 저작도구, 시각 프로그래밍, XML, VESL, VESG

Keywords : Virtual Reality Authoring Tool, Visual Programming, XML, VESL, VESG

1. 서론

1.1. 연구의 배경 및 목적

디지털 기술의 발전과 인터넷의 급속한 확산에 따라 3차원 디지털 모델은 일방적인 프레젠테이션 단계를 넘어 사용자가 능동적으로 3차원 공간을 네비게이션(navigation)하고, 모델을 분석·검토할 수 있는 기술인 가상현실(virtual reality) 기술이 개발되었다. 또한 이는 인터넷상에서의 정보의 검색 및 공유를 가능하게 함으로써 설계 단계에서뿐만 아니라 시공관리, 사후

시설물관리 등 건축 및 실내디자인 분야에서 다양한 연구가 진행되어 왔다. 특히, 최근 사이버 공간 속에 존재하는 실내공간을 중심으로 하는 사이버 공간 기반 가상현실 디지털 주택에 관한 연구는 중요한 이슈(issue)가 되고 있다¹⁾. 이와 같은 연구들은 주로 상용 CAD 프로그램에 의해 완성된 3차원 모델을 가상현실 모델로 변환하여 이를 뷰어(viewer)를 통해 관찰하는 수준으로 연구가 진행되고 있고, 특히 상용 가상현실 저작 도구는 현실감 있는 표현을 강조하고 있어 모델링의 비구조성, 비표준화 파일 포맷 등의 문제점을 발생시키고 있다. 이에 건축물 정보를 기술(description)하고 시각화(visualization)하는 연구는 지금까지 많이 수행되어 왔고,²⁾ 또한 다양한 기능을 가지

* 정회원, 건국대학교 건축공학과 조교수, 건축학박사
** 건국대학교 인터넷, 미디어공학부 강의교수, 공학박사
*** 건국대학교 인터넷, 미디어공학부 교수, 컴퓨터과학박사
**** 정회원, (주)정림건축 기술연구소, 팀장
***** 본 연구는 인터넷, 미디어공학부와 학제 간 연구에 의해 수행됨.

1)이현수, 가상현실을 이용한 디지털 주택의 상호작용 사용자 인터페이스 환경에 관한 연구, 대한건축학회, 20권10호, 2004, pp.11-22.

2)김성아, VRML를 이용한 건축물 정보의 시각화에 관한 기초적 연구,

는 응용 프로그램들이 개발되었다³⁾. 하지만 이러한 연구들은 건축물 정보에 대한 시각화에 중점을 두고 있어 실내공간정보를 기술하는데 부족한 실정이다. 또한 컴퓨터 프로그래밍이나 컴퓨터 그래픽스에 관한 전문 지식을 가지고 있는 사람을 대상으로 한 도구라는 단점을 지니고 있다.

가상현실 기술이 초기 디자인 단계에서부터 최종의 프레젠테이션 단계 그리고 건축물 유지관리 단계에서 적극적인 활용이 가능한, 또한 실물기반 디지털 주택의 공간을 효과적으로 구축할 수 있는 가상의 실내공간구성을 위한 전문적인 저작 도구(authoring tool)가 필요한 실정이다. 이러한 저작도구는 컴퓨터 프로그래밍이나 컴퓨터 그래픽스에 관한 전문 지식이 없는 디자이너가 가상현실 환경을 직관적으로 설계, 제작 및 분석을 할 수 있고 구축된 실내공간은 위계구조에 의한 논리적 분석이 가능해야 한다.

이에 본 연구에서는 실내디자인의 다양한 분야에 응용될 수 있는 실내공간의 직관적 시각 설계에 의한 논리적 분석이 가능한 3차원 가상현실 저작 도구 개발하는 것을 목적으로 하며, 최종 연구 결과물로 프로토타입(prototype) 시스템인 VESL 시스템을 개발하였다. VESL 시스템은 VESL(Virtual Environment Specification Language)⁴⁾이라는 시각 프로그래밍(visual programming)⁵⁾ 언어를 기반으로 구축된 가상의 실내공간제작을 위한 가상현실 저작 도구(virtual reality authoring tool)로써 사용자에게 친숙한 아이콘을 사용하여 다이어그램(diagram)을 그려서 개개의 물체들에 대한 특성을 부여함으로써 쉽고, 빠르기 구조적 가상실내공간을 제작할⁶⁾ 수 있을 뿐 아니라, 시각적으로 표현된 가상실내공간정보를 논리적 구조로 쉽게 정의하고 분석할 수 있도록 해줌으로써 XML(eXtensible Markup Language)로 기술하는 것을 가능하게 한다.

1.2. 연구의 방법 및 범위

본 연구에서는 실내공간구축을 위한 시각 프로그래밍 기반 가상현실 저작 도구⁷⁾ 개발을 위해 다음과 같이 진행되었다.

(1) 가상현실 기술의 실내공간의 적용 사례를 바탕으로 활용

방안과 그에 따른 필요기능을 분석에 관한 문헌조사를 행한다.

(2) 응용 애플리케이션 제작에 사용되는 기존의 시각 프로그래밍 도구의 장점 및 문제점을 분석하여 가상의 실내공간구축을 위한 시각 프로그래밍 기반 3차원 가상현실 저작도구의 개발 전략을 세운다.

(3) 본 시스템에 적용될 시각 프로그래밍 기법 및 알고리즘(algorithm)을 연구한다.

(4) 시각 프로그래밍 기법 및 알고리즘을 기반으로 본 연구의 최종목표인 VESL 시스템을 개발한다.

본 연구에서 구현한 프로토타입 시스템인 시각 프로그래밍 기반 가상현실 저작도구는 시각적인 설계를 바탕으로 사용자들이 편리하게 가상 실내공간을 구축할 수 있도록 하였으며, 정적인 3차원 가상 실내공간을 설계할 수 있을 뿐만 아니라 동적인 가상공간을 정의할 수 있도록 하였다. 즉, 물체간의 충돌검사(collision detection), 객체 간의 제한조건(constraints) 표현 및 물체의 움직임에 정의하는 등의 다양한 임무(task)를 처리할 수 있는 기능을 갖추고 있다. 또한 만들어진 가상의 3차원 실내공간내부를 향해할 수 있는 워크루(walk-through) 기능을 구현하였다.

본 연구는 시각언어를 기반으로 하는 프로토타입 시스템 개발로 연구의 범위를 제한한다. 실무적으로 사용되기 위해서는 사용자의 실제적인 사용을 통해 실효성 검토와 평가가 수반되어야 할 것이다. 실효성 검토와 평가는 향후 과제로 남겨둔다.

2. 문헌고찰

아이콘을 이용한 다이어그램을 그려서 응용 애플리케이션(application)을 시각적으로 작성하는 시각 프로그래밍에 관한 연구는 다양하게 수행되었다⁸⁾. 이러한 시각화 프로그래밍 기술을 사용하면, 객체, 기능, 특성 등을 시각적 아이콘으로 대처하여 프로그래밍 능력이 없는 사용자도 쉽게 가상현실 구축을 위한 프로그래밍이 가능하다. 각각의 시각적 컴포넌트들은 사용자가 직접 만들 수 있으며, 시각화 프로그래밍 제조업체에서 제공하기도 한다. 따라서 사용자는 이러한 컴포넌트를 사용하여 빠르게 응용 애플리케이션을 제작할 수 있다. Labview⁹⁾, AVS¹⁰⁾, Amulet¹¹⁾ 등은 이러한 대표적인 시각 프로그래밍 도구이다.

카네기 멜론 대학교(Carnegie Mellon University)에서 개발

8) Dyer D. S., A dataflow toolkit for visualization, IEEE Computer Graphics and Applications, 1990, pp.60-69.

9) Labview의 홈페이지, <http://www.ni.com>

10) AVS의 홈페이지, <http://www.avis.org>

11) Schurr, A., Winter, A. & Zndorf, A., Visual Programming with Graph Rewriting Systems, 11th International IEEE Symposium on Visual Languages, 1995, p.326.

대한건축학회논문집, 18권11호, 2002, pp.91-100.

3) 차주현 · 이순길 · 전희연, VR을 이용한 인터넷 기반 원격 제어 시스템, 한국 CAD/CAM학회논문집, 5권1호, 2000, pp.88-94.

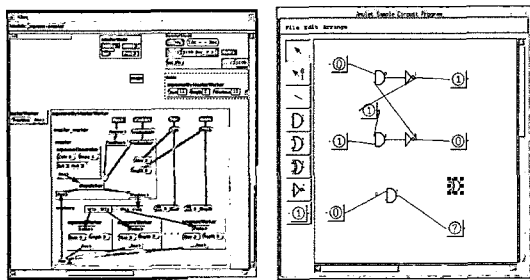
4) VESL은 시각 프로그래밍을 가능하게 해주는 내부적 언어인 시각 프로그래밍 언어이다. 이 시각 프로그래밍 언어는 1997년 건국대학교 인터넷, 미디어 공학부 HCI 연구실에서 개발된 언어이다.

5) 시각 프로그래밍이란 사용자가 아이콘 등을 이용하여 프로그래밍을 하는 것을 의미한다.

6) 예를 들어, 방문을 여닫는 방법, 창문을 여닫는 방법, 등을 정의하는 데에도 다이어그램을 사용하여 쉽게 할 수가 있다.

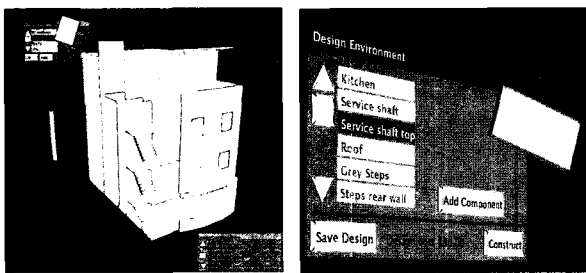
7) 가상현실 저작도구는 3차원 그래픽 라이브러리를 이용하여 가상현실 애플리케이션 및 가상현실공간을 구축할 수 있는 환경을 의미한다. 예를 들어, WorldToolKit, OpenGL, DirectX SDK, VRML 등이 이에 속한다.

한 Amulet은 다양한 GUI(graphical user interface) 객체 컴포넌트를 아이콘으로 이용하여 응용 애플리케이션을 작성하는 시각 프로그래밍 저작도구이다. 이 도구는 객체 컴포넌트에 제한 조건을 명시하여 서로 간에 관계를 일관성 있게 유지할 수 있게 해준다. Amulet의 가장 큰 특징은 객체지향 개념을 사용한 프로토타입 인스턴스 객체 시스템(Prototype-Instance Object System)이라는 점이다. 이는 실시간으로 객체들의 생성과 삭제 등이 동적으로 이루어지는 것을 가능하게 하고, 동적으로 생성된 객체들에 대해 자동적인 관리와 유지를 가능하게 한다. 또한, 객체들에 대해 계층을 두어 합쳐지거나 분리될 수 있는 그룹핑(grouping)이 가능하다. 합쳐진 객체들은 상속(inheritance)이 가능하므로 복잡한 객체들에 대해 쉽게 처리할 수 있는 장점이 있다. 그림1은 Amulet으로 구축된 응용 애플리케이션의 예이다.



<그림 1> Amulet으로 구축된 응용 애플리케이션

이러한 특성을 응용하여 가상현실 분야에 적용하려는 연구는 활발히 진행되고 있으며¹²⁾, 대표적인 예로 가상 건축 분야에 단순한 제한조건을 적용한 Salford 대학의 Future Home을 들 수 있다<그림 2>.



<그림 2> Salford 대학의 Future Home

기존의 가상환경을 구축하기 위한 시각 프로그래밍 기반 저작 도구들은 기본적으로 시각 프로그램 언어를 기반으로 하는 GUI를 제공하고 있으나 몇 가지 문제점들을 가지고 있다. 첫째 기존의 도구들은 객체들 간의 관계를 명시할 수 있는 기법이 시각적으로 나타나지 않는다는 점이다. 둘째, 시각적인 환경과 가상환경 내의 물체들 간의 제한조건을 일관성 있게 유지하

는 시스템이 제공되지 못한다는 점이다. 마지막으로 동적인 가상 환경을 만들기 위해 물리적인 작업을 직접적으로 C와 같은 컴퓨터 프로그래밍 언어를 사용하여 코딩해야 한다는 어려움이 있다. 따라서 이러한 문제점들을 사용자 측면에서 해결할 수 있는 시각 프로그래밍 기법의 연구와 이를 지원하는 시스템 개발이 필요하다.

3. 개발 전략

실내공간구조를 위한 가상현실 저작도구는 다음과 같은 전략을 바탕으로 하여 시스템을 구성한다.

(1) 실내공간구조를 위한 가상현실 도구는 디자인 초기 단계에서부터 최종의 프레젠테이션 단계까지 사용이 가능하도록 직관적인 인터페이스에 의한 다이어그램이나 스케치를 이용한 시각화 설계를 가능하도록 한다.

(2) 편리적인 논리적인 설계를 위해 실내공간과 다양한 물체 간의 상호관계에 대한 구조화를 통해 논리적 관계를 형성하도록 한다.

(3) 다양한 시뮬레이션을 위해 동적인 가상환경 구축이 용이해야 한다.

(4) 빠른 수정과 더 많은 피드백(feedback)이 가능하도록 실시간 방식의 오류체크가 가능하도록 한다.

(5) VESL 문법을 검증, 정보의 생산성, 재사용성, 효율적 정보관리 및 정보의 이식성을 위해 실내공간정보의 기술이 XML로 표현하고 저장이 가능하도록 한다.

(6) 각각의 모듈¹³⁾에서 물체간의 일관성 있는 제한조건을 갖도록 개발되어야 한다.

4. 시각 프로그래밍 언어의 정의

4.1. VESL(Virtual Environment Specification language)








본 연구에서는 가상실내공간구조를 위한 시각 프로그램 도구를 개발하기 위해 시각 프로그래밍 언어인 VESL(Virtual Environment Specification Language)를 개발하고 이를 수학적으로 적용하기 위해 VESG(Virtual Environment Specification Grammar)¹⁴⁾를 정의하였다. 따라서 VESL을 통해 사용자는 2차원 다이어그램 기법을 이용하여 대화형(integrative) 가상환경을 구축할 수 있다. <표 1>은 VESL로서 개발된 시각 컴포넌트들을 나타낸다.

12) Fa, M., Fernando, T., & Dew, Peter M., Interactive Constraint-based Solid Modeling using allowable Motion, Proceedings on the second ACM symposium on Solid modeling and applications, 1993, pp.243-252.

13)본 연구에서 개발된 시스템은 VESL Editor, Gallery Builder, Space Viewer 이렇게 3가지 모듈로 구성된다.

14)VESG는 VESL로 정의된 시각 컴포넌트들 사이에 관계를 설정하기 위한 문법이다. 이 문법은 기본적으로 6개의 공간 오퍼레이터들로 구성되어 있다.

<표 1> VESL로서 개발된 시각 컴포넌트

Node 설명	아이콘	설 명
House Element		전체적인 집을 구성하는 최 상위 노드 (House)
Room Element		House Element를 구성하는 하위노드 (Room, Livingroom, Bathroom, Kitchen 등)
Furniture Element		Room Element를 구성하는 하위노드 (Table, Vase, Chair, ShowerBath, SinkTable 등)
Task		Furniture Element에 부여되는 속성 (Move, Rotation, Animation, Gravitation, On/Off, Open/Close 등)
Node Connection		House Element, Room Element, Furniture Element를 연결하는 노드 (House-Room, House-Livingroom 등)
Constraint Connection		각 노드들 간의 Constraint를 정의할 때 사용되는 연결 선
Constraint		Constraint를 정의 (Left_of, Over, Front_of, On, In, Links_to 등)

4.2. 공간 오퍼레이터

VESG에서는 객체들 사이의 관계에 따라서 공간 오퍼레이터들(spatial operators)을 6가지로 정의하였다. 공간 오퍼레이터들은 가상환경의 객체들 사이에 포함되어 있는 하나 이상의 관계 제약 조건(relation constraints)을 명시하게 되며, 이를 위해 3차원 상에서 객체들의 관계를 정확히 포함할 수 있다. 또한 공간 오퍼레이터 사이의 합성(composition)을 허용함으로써 3차원 상에서 일어날 수 있는 모든 상황을 표현하는 것이 가능하다. <표 2>는 VESG 문법으로써 정의된 공간 오퍼레이터들을 나타낸다.

<표 2> 공간 오퍼레이터

공간 오퍼레이터	설 명
Left_of (, <c>)	b는 c의 왼쪽에 있다. (+ X 방향)
Over (, <c>)	b는 c의 위에 있다. (+ Y 방향)
Front_of (, <c>)	b는 c의 앞에 있다. (+ Z 방향)
On (, <c>)	b는 c의 위에 붙었다.
In (, <c>)	b는 c의 안에 있다.
Links_to (, <c>)	b는 c에 붙어 연결되어 있다. (창, 문, 계단)

<그림 3>은 시각 언어로 프로그래밍을 작성할 경우 적용되는 문법의 예를 나타낸다.

```

Rule 1 :
Floor Wall Door Window Furniset -> Bedroom
<Bedroom cover> = ( Compose <Furniset cover> <Wall cover>
                    <Door cover> <Window cover> <Floor cover> )
on(<Wall cover><Floor cover>)
on(<Furniset cover><Floor cover>)
links_to(<Window cover><Wall cover>)
links_to(<Door cover><Wall cover>)

Rule 2 :
Furniset2 Furniset3 -> Furniset1
<Furniset1 cover> = (Compose<FurniSet2 cover><FurniSet3 cover>)
<FurniSet1 syntax> = Union(<FurniSet2 syntax><FurniSet3 syntax>)
    ∃ f1 ∈ <FurniSet2, syntax>
    ∃ f1 ∈ <FurniSet3, syntax>
such that
    left_of(f1, f2) | front_of(f1, f2) | (left_of ? front_of)(f1, f2)

Rule 3 :
Furniture -> Furniset1
<FurniSet1 cover> = <Furniture cover>
<FurniSet1, syntax> = Set_of(<Furniture, syntax>)
    
```

<그림 3> 공간 오퍼레이터를 이용한 문법적용의 예

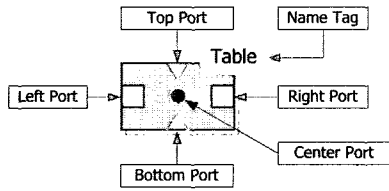
Rule 1, Rule 2, Rule 3은 방(창문, 문, 벽 포함)안에 침대, 책상, 의자를 포함하고 있는 가상공간의 문법을 나타낸다. 예를 들어 이탤릭체로 표시 되어있는 Furniture -> Furniset1 시각 컴포넌트를 의미하는 규칙이고, 시각 컴포넌트 규칙 밑에 나타나 있는 항목들은 시각컴포넌트를 만족하기 위한 조건이다. 따라서 Rule에서 나타나는 on(<Wall cover><Floor cover>)과 같은 경우 벽의 cover, 즉 경계 위치 값은 마루 cover 위에 있어야 한다는 것을 의미한다.

5. VESL 시스템의 시각 프로그래밍 기법

5.1. 시각 컴포넌트의 구성

VESL는 프로그램 지식이 없는 사용자가 가상공간에서 객체의 정의 및 제한사항을 시각적으로 설계할 수 있는 시각 컴포넌트(visual component)를 제공한다. 시각 컴포넌트로 구성된 다이어그램으로 설계된 공간정보와 제한사항은 자동적으로 VESG 문법을 생성한 후, 시스템의 3차원 뷰어를 통해 실시간 렌더링되어 사용자에게 보이게 된다. 동시에 설계된 공간정보는 내부적으로 파싱되어 XML 파일로 저장된다.

시각 프로그래밍 인터페이스에서 사용자가 표현하는 구성 요소는 시각 컴포넌트 노드(node), 포트(port), 연결선(communication link)으로, 노드는 아이콘 모양<그림 4>으로 표현되고 VESL에서 정의된 논리적 설계에 따라 해당하는 입출력 포트를 가진다. 또한 각 노드 객체에 대해 논리적이고, 동적인 속성을 정의하여 보다 현실감 있는 공간 구성을 할 수 있도록 제한조건 노드와 임무 노드로 구성된다.

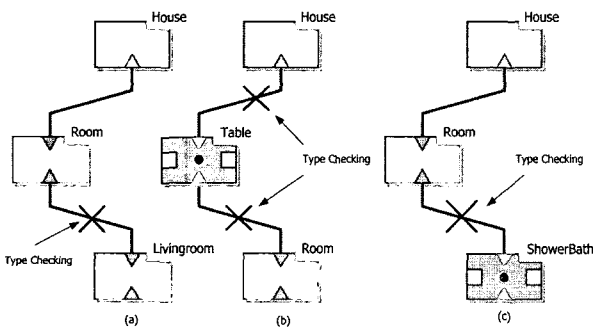


<그림 4> 가구 요소 노드 아이콘

포트는 가상공간의 논리적 설계에 따른 위상관계를 위해 두 개의 포트 위 포트(Top-Port)와 아래 포트(Bottom-Port)로 구성된다. 따라서 논리적 설계가 VESG에 적용되지 않으면 연결이 되지 않게 된다. 좌 포트(Left-Port)와 우 포트(Right-Port)는 공간 오퍼레이터에 의해 정의된 제한사항을 명시하기 위해 계획되었고, 중앙 포트(Center-Port)는 해당 노드가 가지고 있는 동적인 속성을 정의하기 위해 계획되었다.

5.2. 타입검사

자동적인 타입검사 작업은 다이어그램에서 실시간적으로 수행되며, VR 디자이너로 하여금 보다 정확하게 가상공간을 설계할 수 있도록 도와준다. 그림 5는 다이어그램에서 시각 프로그래밍을 했을 경우, 타입검사(type checking) 오류에 대한 예를 보이고 있다. 그림 5(a)는 침실(room)과 거실(livingroom)이 같은 레벨이기 때문에 연결할 때 오류를 발생한다. <그림 5(b)>의 경우는 가구(furniture) 요소인 테이블(table)은 직접적으로 주택(house) 밑에 추가할 수 없기 때문에 오류가 발생한다. <그림 5(c)>의 경우, 샤워실(shower bath)은 욕실(bathroom)에 포함되어야 하기 때문에 잘못된 연결을 나타낸다. 이러한 개체들에 대한 타입검사는 시각적 설계뿐만 아니라, 내부적으로 공간상의 구조화된 논리설계를 지원한다.

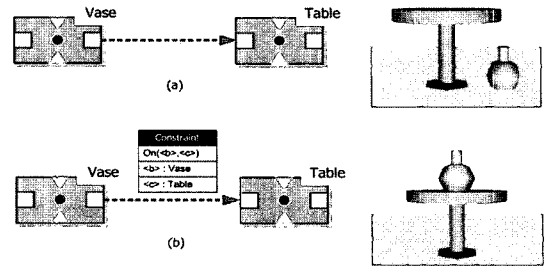


<그림 5> 연결선의 타입검사

5.3. 제한사항

제한사항(constraint)은 크게 두 가지 부분으로 나누어질 수 있다. 첫 번째는 객체들 사이의 관계(relationship)를 설정하는 것이고, 두 번째는 각 객체들이 가지는 동적인 속성을 명시하는 제한사항이다. 객체간의 관계설정은 공간 오퍼레이터에 정의된 공간적 제한사항으로 VR 디자이너는 시각 다이어그램

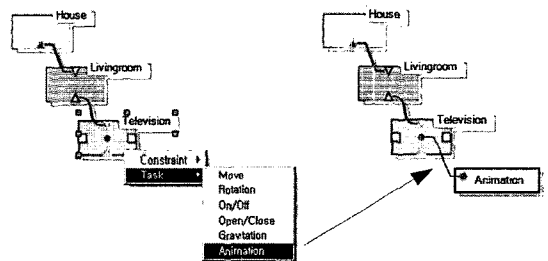
(visual diagram)을 통해 객체간의 관계를 정의할 수 있으며, 한번 정의된 관계는 제한요소 관리자에 의해 계속 유지된다.



<그림 6> 테이블과 꽃병 사이의 제약사항 명시 예

<그림 6>은 본 시스템에서 제공하는 제한조건을 설명한다. VESL 시스템에서 테이블 노드와 꽃병(vase) 노드를 추가하였을 경우 자동적으로 공간 다이어그램에서는 해당하는 객체를 3차원으로 보여준다. 일반적인 가상환경 저작 시스템들은 이러한 경우 마우스로 끌어다 테이블 위에 놓는 방법을 택하게 된다<그림 6(a)>. 이러한 방법은 정확하지도 못하며 시간과 노력을 많이 필요로 하게 된다. 반면 본 시스템에서는 테이블 위에 꽃병이 "On" 되어있다는 논리적인 관계를 명시 해줌으로서 사용자 하여금 보다 편리하게 작업할 수 있도록 계획했다. <그림 6(b)>. 결과적으로, 화병이 테이블 위에 놓여져 있다는 관계를 설정함으로써 테이블의 움직임에 따라 화병도 같이 따라 움직여야 한다는 제한사항을 가지게 된다.

본 시스템에서는 보다 동적인 가상환경을 제공해 주기 위하여 각 객체에 대해 이동, 회전, 중력(gravitation)¹⁵⁾, 애니메이션(animation), On/Off, Open/Close 등의 임무를 부여하고 있다. 모듈화되어 있는 이러한 임무들은 다이어그램에서 생성된 노드들에 대해 독립적으로 적용할 수 있으며, 실시간으로 공간 뷰어에서 확인 가능하도록 계획했다.



<그림 7> 임무 설정 과정

15) "테이블 위에 꽃병이 놓여져 있다"라는 관계(Relationship)를 설정한 후 타자를 이동(move)할 경우 테이블 위의 꽃병도 따라 이동하게 된다. 반면, 관계를 설정하지 않는다면, 테이블을 이동할 경우 꽃병은 허공에 있게 된다. 이러한 현상은 일반적인 가상환경 저작 도구에서 자주 볼 수 있는 일이다. 본 논문에서는 'Gravitation Function'을 적용하여, 테이블을 이동할 경우에 자연스럽게 중력 기능을 적용하여 꽃병이 떨어지도록 하였다.

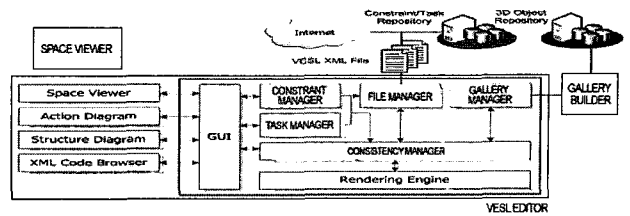
이러한 임무들은 OOP(Object-Oriented Programming)의 개념을 도입하여 다형성과 상속 개념을 사용하고 있다. 예를 들어 Open(Open-U, Open-R)은 문에 대한 임무인데 여닫이문과 미닫이문에 다같이 적용될 수 있다. Open은 적용되는 문에 따라 행동양식이 달라진다. 즉, 회전하여 열리거나, 오른쪽으로 미끄러지면서 열리게 된다. 또한, 상위계층의 속성은 하위계층의 크기와 색상 등의 객체속성에도 적용되어진다. 이러한 상속성은 건축설계의 공통된 특징을 계층화함으로써 적용될 수 있다.

6. 가상환경 저작 도구: VESL 시스템

6.1. 시스템 구성도

본 연구에서 개발한 가상현실 시각프로그램 저작도구인 VESL 시스템¹⁶⁾은 크게 3가지 모듈인 VESL 편집기(VESL Editor), 갤러리 빌더(Gallery Builder), 공간 뷰어(Space Viewer)로 구성된다. VESL 편집기는 그림 8과 같이 5개의 컴포넌트(component)¹⁷⁾로 구성되며, 서로 유기적으로 연결되어 긴밀하게 작동된다.

파일 관리자(File Manager)는 사용자에 의해 만들어진 3차원 공간상의 객체들에 대한 정보를 담고 있는 XML 파일 관리자이다. 이러한 XML은 정보의 생산성, 재사용성, 효율적 정보 관리 및 정보의 이식성이 우수한 구조적 언어로서 인터넷을 통해 여러 사용자들에게 효과적으로 공유될 수 있다¹⁸⁾. 제한요소 관리자(Constraint Manager)는 가상공간 안에서 객체들 간의 제한사항을 정의하고 관리한다. 객체간의 관계에 의한 제한사항들은 객체들의 공간정보와 함께 XML 파일로 만들어져 제한요소 저장소(Constraint Repository)에 저장된다. 갤러리 관리자(Gallery Manager)는 3D 응용 애플리케이션을 통하여 만들어진 객체를 3차원 공간 뷰어에서 효율적으로 나타내기 위해 사용되는 컴포넌트이다. 이렇게 만들어진 3차원 객체들은 3차원 객체 저장소(3D Object Repository)로부터 필요한 정보를 가지고 와서 사용자에게 전달해 주는 역할을 담당한다.

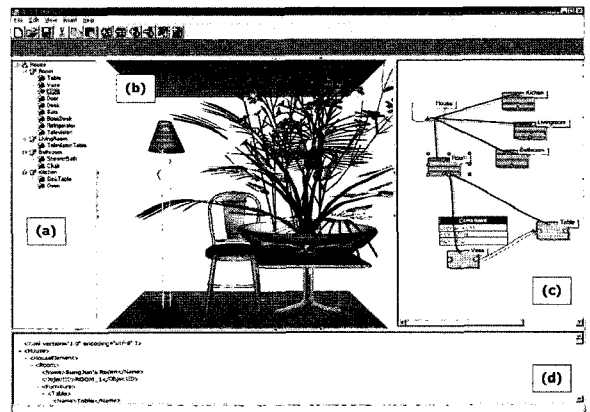


<그림 8> VESL 시스템 구성도

작업 관리자(Task Manager)는 VR 디자이너가 구축한 가상 공간에 동적인 속성을 부여할 수 있도록 모듈화된 라이브러리를 생성해주는 시스템이다. 임무(task)의 기능에는 이동(move), 회전(rotation) 등이 있으며, 객체에 대한 속성을 정의하게 된다. 일관성 유지 관리자(Consistency Manager)는 사용자에 의해 설계되어지는 시각 프로그래밍과 3차원 뷰어를 동기화 해주며, 내부적으로 잘못된 설계를 검사하고 유지하는 시스템이다. 또한 렌더링 엔진(rendering engine)은 사용자에 의해 설계된 가상실내공간을 실시간 시각화하는 역할을 수행한다.

6.2. VESL 편집기

VESL 편집기의 GUI는 직관적인(intuitive) 작업이 가능하도록 4개의 부분으로 구성되어 있다. 물체들 간의 계층적 구조를 트리(tree) 형식으로 보여주는 구조 다이어그램(Structure Diagram)<그림 9(a)>과, 물체들 간의 논리적 구조를 3차원으로 렌더링하여 사용자로 하여금 관찰하고 물체의 이동 등의 조작인 가능한 공간 빌더(Space Builder)<그림 9(b)>, 실제적으로 가상실내공간을 설계하기 위한 시각 프로그램 인터페이스인 액션 다이어그램(Action Diagram)<그림 9(c)>, 그리고 공간정보를 XML 파일 형태로 보여주는 XML 코드 뷰어(XML Code Viewer)<그림 9(d)> 부분으로 구성된다.



<그림 9> VESL 편집기 실행화면

구조 다이어그램은 VESL를 이용하여 저작된 가상환경의 논리적 설계를 트리 형식으로 시각화하여 표현한 것이다. 사용자는 동적으로 트리를 구성할 수 있으나, VESG에 정의된 규칙에

16)본 연구에서 개발된 VESL 시스템은 건국대학교 인터넷, 미디어공학부 HCI 연구실과 학제 간 연구에 의해서 개발된 시스템으로써 OpenGL 라이브러리를 기반으로 객체지향언어인 C++로 초기버전이 개발되었으며, 이후 C#으로 시각프로그램 인터페이스를 지원하는 액션다이어그램이 추가로 개발되었다.

17)5개의 컴포넌트는 각각 File Manager, Constraint Manager, Gallery Manager, Consistency Manager, Task Manager를 의미한다.

18)이주영·김인환, XML기반의 건설 분야 설계정보의 표현 및 활용에 관한 연구: STEP 데이터 모델의 활용을 중심으로, 대한건축학회, 20권 10호, 2003, p.3.

기반을 두어 트리를 생성될 수 있다. 논리적 설계를 이루고 있는 트리 노드들은 계층적으로 구성되어 있으며, 계층적으로 구성된 각각의 물체들에 대해서 물리적인 속성을 정의하는 임무를 할당하여 물체의 애니메이션이나 시뮬레이션을 위한 행동양식을 지정할 수 있다. 또한 구조 다이어그램은 컨시스턴시 관리자 컴포넌트에 의해 공간 뷰어, 액션 다이어그램, XML 코드 뷰어와 내부적으로 동기화 된다.

공간 빌더는 구조 다이어그램과 액션 다이어그램에서 설계된 가상의 실내공간을 3차원으로 보여주고 생성된 물체의 이동 등의 작업이 가능하다. 갤러리 빌더를 통해 만들어진 물체들은 공간 빌더를 통하여 실시간으로 업데이트 된다. 공간 빌더는 앵커(anchor)의 개념을 적용하였다. 예를 들어 액션 다이어그램에서 침실(room) 노드를 클릭하면 해당되는 침실로 이동할 수 있으며, 침실에 속하는 물체에 대한 노드를 클릭하면 확대되어 사용자에게 보여준다. 이러한 기법은 가상의 실내공간에서 다량의 객체를 구별할 때 효과적으로 이용할 수 있다.

액션 다이어그램은 사용자가 실제적으로 가상실내공간을 설계하기 위한 시각 프로그래밍 인터페이스이다. 사용자는 가상의 실내공간을 설계하기 위해, 이미 만들어져 있는 각각의 아이콘을 사용하여 직관적인 방법으로 쉽게 만들 수 있다. 기존의 시각적 설계뿐만 아니라, 논리적 설계를 정의할 수 있으며, 동적인 가상 환경을 제공하기 위해 제한사항과 다양한 임무를 정의하게 된다.

VESL 문법을 검증하기 위해서는 입력된 가상공간 객체들에 대해 파싱(parsing)이 이루어 져야 한다. VESL 편집기에서는 잘못된 입력에 대해 시간과 공간상의 영향을 받지 않으면서 오류를 검출할 수 있는 기능을 제공하고 있다. 워크루에 필요한 공간상의 객체들은 잘 정의된 XML 파일 형태로 저장되어 파싱을 하게 된다. 예를 들어 화장실(bathroom) 영역에 있어야 할 Chamber_Pot이 침실(room) 영역에 있을 경우 이것은 잘못된 시각 프로그램이다. 옳지 않은 VESL은 객체들 간의 위치 관계나 포함 관계 및 실제와 같지 않은 건축 프로그램을 파악해줄 수 있다. 이러한 유기적인 관계가 사용자에게 제공됨으로 VESL를 기반으로 하는 본 VESL 편집기는 더욱 가상실내공간을 현실감 있고 편하게 제작할 수 있게 하는 특징을 갖는다.

<그림 10>은 공간상의 객체들에 대한 정의를 XML 파일 형태로 만든 것이다. 이러한 XML 파일은 건축공간에 속한 각각의 객체의 형상의 특징을 구조적으로 표현해 준다. 1-4행까지는 House와 Room에 대한 정의이며, 5-21행까지가 Furniture에 대한 정의이다. 여기에는 VESL 문법에 맞는 객체들을 포함할 수 있으며, 또한 객체들에 대한 제한조건과 기능을 포함하게 된다. 잘 정의된 공간적 정보를 담고 있는 XML 파일은 저장소에 저장되어진다. XML은 표준화된 언어로, 객체지향 개념을 지향하며 텍스트 기반이라는 장점을 지니고 있어, 다른 건축공

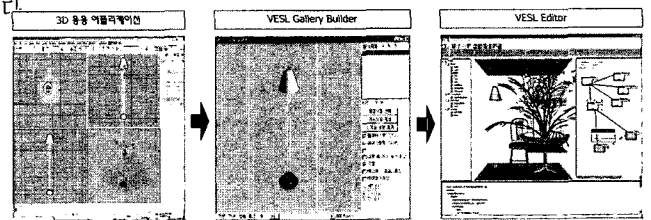
행	코드
1	<HouseElement>
2	<House id="H1" name="House">
3	<RoomElement>
4	<Room id="R1" name="Room">
5	<FurnitureElement>
6	<Vase id="F1" name="RoomVase">Vase </Vase>
7	<Table id="F2" name="RoomTable">Table </Table>
8	<ConstraintList>
9	<Constraint id="C1" name="VaseTableConstraint">
10	<Property> On </Property>
11	<FromObject> F1 </FromObject>
12	<ToObject> F2 </ToObject>
13	</Constraint>
14	</ConstraintList>
15	<TaskList>
16	<Task id="T1" name="VaseTask">
17	<Property> Rotation </Property>
18	<PropertyObject> F1 </PropertyObject>
19	</Task>
20	</TaskList>
21	</FurnitureElement>
22	</Room>
23	</RoomElement>
25	</House>
26	</HouseElement>

<그림 10> 공간정보를 담고 있는 XML 코드

간에 임베드(embed)되어 재사용될 수 있는 다양한 확장성을 지니게 된다¹⁹⁾.

6.3. Gallery Builder

본 연구에서는 갤러리 빌더를 통한 3차원 디자인 도구에 의해 생성된 객체를 불러들여(loading) 작업을 수행하게 된다. 따라서 3차원 객체는 3차원 응용 프로그램을 통하여 만든 후, ASE 파일²⁰⁾ 포맷 형태로 저장해야한다. 객체는 갤러리 빌더의 ASE 로더(loader)를 통하여 VESL 편집기의 공간 빌더에 추가된다. <그림 11>은 3D 응용 프로그램을 이용하여 가상공간 내에 객체를 생성하기까지의 변환과정을 실제로 수행한 그림이다.



<그림 11> 3D 응용 프로그램부터의 객체 생성과정

본 논문에서는 물체의 재질정보와 위치정보를 갖고 있는 ASE 파일을 직접 파싱한 후, 갤러리 빌더를 통해 효과적으로 축소한 후 사용하였다. 이러한 이유는 3차원 응용프로그램을 통해 만들어진 물체의 데이터양의 크기가 상당히 크기 때문에 실시간 렌더링을 위해 적절하게 재구성할 필요가 있기 때문이다. 이렇게 갤러리 빌더를 통해 만들어진 물체는 공간 빌더와

19)이연희 외 3인, 가상건축물 기술 언어를 이용한 자동화된 건축물 외관 모델링, 한국정보과학회 인간과 컴퓨터 상호작용 연구회 학술대회, 2002.

20)ASE 파일 포맷은 ASCII로 구성된 3D 모델링 파일 포맷으로 다양한 3차원 정보를 가독성 있는 텍스트 형식으로 제공해 준다.

공간 뷰어를 통해 자연스럽게 보인다.

7. 결론 및 향후 과제

본 연구에서는 기존의 가상현실 저작 도구들이 대부분 인터페이스에 의한 시각적 설계에만 중점을 두었다는 점과 가상환경을 저작하기 위하여 컴퓨터 프로그래밍 기술을 필요로 한다는 문제점을 해결하기 위해 새롭게 만든 시각언어(VESL: Visual Environment Specification Language)를 가지고 프로그래밍 기술이 없는 사용자가 쉽고 편리하게 현실적인 논리적 가상공간을 구축할 수 있는 3차원 가상현실 저작도구를 개발하였다.

VESL 편집기는 객체들 간의 논리적 특성과 관계를 시각화하고 그들 간의 관계를 지속적으로 유지할 수 있도록 제한조건 정의와 유지에 기반을 두고 있다. VESL 편집기는 물체들의 움직임과 같이 다양한 물리적 속성을 정의하고 있는 임무를 제공하며, 이러한 행동양식을 이용하여 보다 현실적이고 동적인 가상 건축실내 환경을 만들어 낼 수 있다.

향후 과제로는 이러한 행동양식을 스크립트(script)화 하여 시각화하는 것이다. 스크립트화 한 행동 양식들은 하나의 시나리오로 만들어질 수 있으며, 다수의 시나리오를 조합하여 보다 복잡하고 동적인 가상의 건축공간을 만들어낼 수 있을 것이다. 또한, VESL 편집기에서 제작한 가상공간을 인터넷 가상현실 표준 언어인 VRML로 전환할 수 있는 번역기(converter)를 만들 예정이다. 마지막으로 본 연구의 시스템을 실용적인 적용을 위한 사용성 평가 또한 향후에 이루어질 예정이다.

참고문헌

1. Burnett, M. M., Golberg, A., & Lewis, T., Visual Object Oriented Programming, Manning Publications Co, 1995.
2. Marriott, K., & Meyers, B., Visual Language Theory, Springer Verlag, 1998.
3. Dennis, K., & Smeulders, A.W.M., A visual programming interface for an image processing environment, In Pattern Recognition Letters, 1994.
4. 김성아, VRML를 이용한 건축물 정보의 시각화에 관한 기초적 연구, 대한건축학회논문집, 18권11호, 2002.
5. 김성아, XML과 VRML의 보완적 적용에 의한 건축물 정보표현방안, 한국실내디자인학회논문집 35호, 2002.
3. 서정범·박성준·김지인, 가상 공간설계를 위한 대화형 시각언어에 관한 연구, 한국정보과학회, 1997.
7. 이주영·김인환, XML기반의 건설 분야 설계정보의 표현 및 활용에 관한 연구: STEP 데이터 모델의 활용을 중심으로, 대한건축학회, 20권 10호, 2003.
3. 이연희·황지은·이윤길·최진원, 가상건축물 기술 언어를 이용한 자동화된 건축물 외관 모델링, 한국정보과학회 인간과 컴퓨터 상호작용 연구회 학술대회, 2002.
3. 이윤길·이연희·최진원, 지능형 가상공간의 구축 방법론에 대한 연구, 한국정보과학회 인간과 컴퓨터 상호작용 연구회 학술대회, 2002.
10. 이현수, 가상현실을 이용한 디지털 주택의 상호작용 사용자 인터페이스 환경에 관한 연구, 대한건축학회, 20권10호, 2004.
11. 차주현·이순걸·전희연, VR을 이용한 인터넷 기반 원격 제어 시스템,

한국 CAD/CAM학회논문집, 5권1호, 2000.

12. C. Williams, J., Raruse, & Hansen, C., The state of the art of visual languages for visualization, Proceedings of Visualization 1992.
13. Dyer D. S., A dataflow toolkit for visualization, IEEE Computer Graphics and Applications, 1990.
14. Fa, M., Fernando, T., & Dew, Peter M., Interactive Constraint-based Solid Modeling using allowable Motion, Proceedings on the second ACM symposium on Solid modeling and applications, 1993.
15. Fernando, T., Murray, N., Tan, K., & Wimalaratne, P., Software Architecture for a Constraint-based Virtual Environment, ACM, 1999.
16. Garber, M., & Lin, M., Constraint-based Motion Planning for Virtual Prototyping, ACM, 2002.
17. Schurr, A. Winter A., & Zndorf, A., Visual Programming with Graph Rewriting Systems, 11th International IEEE Symposium on Visual Languages, 1995.
18. AVS Homepage, <http://www.avs.org>
19. Labview Homepage, <http://www.ni.com>

<접수 : 2005. 8. 31>