

# XML 기반의 3 차원 그래픽 데이터 모델\*

이경미\* · 양정희 · 황수찬

\*한국항공대학교 컴퓨터공학과

e-mail : kkangmi@mail.hankong.ac.kr

jung@mail.hankong.ac.kr

schwang@mail.hankong.ac.kr

## An XML-Based 3 Dimensional Graphics Data Model

Kyong-Mi Lee\*, Jung-Hee Yang, Soo-Chan Hwang

\*Dept. of Computer Engineering, Hankuk Aviation University

### 요 약

최근 인터넷에는 기존의 문서나 이미지뿐만 아니라 3 차원 그래픽 데이터도 급속히 증가하고 있다. 이에 따라 3 차원 그래픽 데이터의 검색을 지원하는 웹 데이터베이스 시스템이 요구되고 있다. 이러한 필요성에도 불구하고 아직까지 3 차원 그래픽 데이터의 내용기반 검색에 대한 연구는 거의 이루어지지 않고 있는 실정이다. 본 논문에서는 3 차원 그래픽 데이터의 내용기반 질의를 위한 XML 기반의 데이터 모델인 3DGML(3-Dimensional Graphic Modeling Language)을 제시한다. 또한 본 논문에서는 3DGML에 적합한 시스템 구조를 제시한다.

### 1. 서론

최근 인터넷에는 기존의 문서나 이미지뿐만 아니라 가상공간, 시뮬레이션, 모델하우스 등과 같은 3 차원 그래픽 데이터 응용이 급속히 증가하고 있다. 이에 따라 3 차원 그래픽 데이터 검색이 가능한 웹 데이터베이스 시스템에 대한 요구가 증가되고 있지만 현재의 3 차원 그래픽 데이터들의 표현은 단순히 시각화를 위해 고안되었기 때문에 이미지의 색상, 질감 등과 같은 단순 질의 검색만이 가능하다. 하지만, 시뮬레이션이나 건축 인테리어 등을 포함한 여러 응용분야에서 3 차원 객체의 모양기반 검색이나 객체간의 공간관계에 기반한 검색을 포함한 다양한 내용기반 검색 기능이 요구되고 있다.

현재 XML은 인터넷에서 정보교환을 위한 표준으로 정착되었다. XML은 HTML과 같은 마크업 언어이지만, 다양한 특징 집합을 제공하지는 않는다. 대신, DTD(Document Type Definition)나 스키마(Schema)를 통해 사용자 정의 태그를 기술함으로써 구조화된 문서를 정의할 수 있도록 한다[1]. 본 논문에서는 이런

XML을 이용하여 다양한 방법의 3 차원 그래픽 데이터 내용기반 검색이 가능한 3 차원 그래픽 데이터 모델인 3DGML을 정의한다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구를 기술하고 3 장에서는 3DGML의 모델링 기능에 대해 기술한다. 그리고 4 장에서는 3DGML을 위한 시스템 설계도, 5 장에서는 결론 및 향후 연구에 대해서 기술한다.

### 2. 관련 연구

#### 2.1 문서 정의어

XML은 사용자가 DTD나 스키마를 사용하여 사용자 정의 태그를 기술함으로써 데이터 타입이 정의되지 않은 태그를 정의하여 사용할 수 있다. DTD나 스키마란 XML 문서 안의 데이터들에 대한 구조를 표현하는 규칙들의 집합이다[7].

DTD는 XML 문서 구조를 정의하기 위해서 XML 문법과 다른 EBNF(Extended Backus-Naur Form) 문법을 사용한다. 한 문서에는 하나의 DTD만을 적용할 수 있으며 여기에 정의된 문서 규칙에 따라서 XML 문서

\* 본 논문은 과학기술부 한국과학재단 지정 경기도 지역협력연구센터(RRC)인 한국항공대학교 인터넷정보검색연구센터의 지원에 의한 것임.

를 생성할 수 있다.

XML 스키마는 DTD의 많은 단점들을 보완하기 위해 제안된 메커니즘이다. XML 스키마는 XML 1.0 스펙의 문법을 사용한다. 따라서, XML 문서와 같은 방법으로 조작이 가능하며 DOM[9]을 지원한다. 뿐만 아니라 XML 스키마는 DTD의 제한된 데이터 타입을 지원과는 다르게 현재 프로그래밍 언어에서 이용하고 있는 많은 데이터 타입을 제공한다. 더 자세한 DTD, Schema 문법은 [8],[11]을 참고하길 바란다.

본 논문에서는 문서의 규칙을 정하기 위한 문서 정의어로 다양한 데이터 타입 지원이 가능한 XML 스키마를 사용할 것이다.

### 2.2.3 3차원 객체 모델

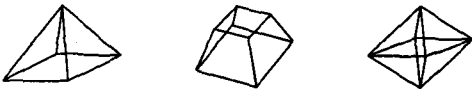
본 논문의 XML 스키마에서 정의할 3차원 객체는 기본 객체, 사용자 정의 객체 그리고 복합 객체로 구분한다. 기본 객체는 3차원 그래픽을 표현하는데 기본이 되는 객체로서 그림 2.1과 같이 직육면체, 원뿔, 원기둥, 구로 구성되어 있으며, 기본 객체로 표현할 수 없는 객체에 대해서는 다각형(polygon)으로 구성된 사용자 정의 객체로 정의한다.

기본 객체는 3차원 객체를 모델링 할 때 가장 많이 사용되는 객체로서 본 논문에서는 기본 객체를 BObject(Basic Object)로 명칭한다.



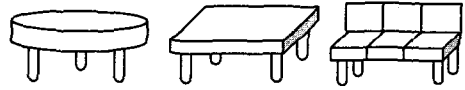
(a) 직육면체 (b) 원뿔 (c) 원기둥 (d) 구  
그림 2.1 기본 객체

그림 2.2와 같은 사용자 정의 객체는 3차원 객체를 모델링 할 때 BObject를 사용하여 표현할 수 없을 때 사용자가 다각형(polygon)을 사용하여 객체를 정의하는 것으로 사용자 정의 타입과 같다. 사용자 정의 객체의 각 면은 BObject와 동일하게 특징(feature) 값으로 자기의 고유 색상과 텍스처 값을 갖고 이 값을 이용하여 내용 기반 질의를 할 수 있다. 본 논문에서는 사용자 정의 객체를 UObject (User Define Object)로 명칭한다.



(a) 오면체 (b) 육면체 (c) 팔면체  
그림 2.2 사용자 정의 객체

그림 2.3과 같은 AObject는 책상, 의자, 식탁, 침대 등과 같은 의미를 갖는 의미 단위(semantic unit)이고 BObject와 UObject를 이용하여 모델링된 복합 객체이다.



(a) 원형탁자 (b) 사각탁자 (c) 3인용의자  
그림 2.3 복합 객체

AObject는 자신만의 의미와 특성값을 담고 있다. 그래서 3차원 그래픽 데이터의 단순한 의미 검색 뿐만 아니라 내용기반 검색도 가능하게 한다 [9].

## 3. 3DGML

VRML, X3D와 같은 3차원 그래픽을 표현하는 언어들이 존재하였지만, 이런 표현들은 오직 시각화를 위한 것들이었다. 이에 반하여 본 논문에서 제시한 3DGML은 시각화를 위한 3차원 그래픽 데이터의 외각정보와 내용기반 검색을 위한 특징정보, 공간관계 검색을 위한 3차원 객체 관계 정보 등을 포함한다. 3DGML의 선언부에서는 객체의 주요 골격정보를 정의하며, 디스플레이부에서는 선언부에 정의된 객체를 참조하여 시각화 정보를 첨가한다. 선언부와 디스플레이부간의 관계는 일반 프로그래밍 시스템의 타입/인스턴스 관계와 유사하다.

### 3.1 스키마 구조

스키마는 문서 구조에 사용되는 규칙의 집합을 정의한다. 본 논문에서 사용할 스키마의 개념적 구조를 보면 다음의 그림과 같다.

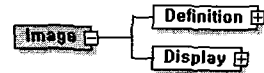


그림 3.1 3DGML의 개념도

위의 그림 3.1과 같이 하나의 3차원 Scene은 크게 선언부와 디스플레이부로 나뉘어진다. 선언부는 공간 관계 검색을 위한 특징 정보, 단순 질의 검색을 위한 Scene의 설명 정보, 3차원 그래픽 데이터의 내용기반 검색을 위한 특징 정보 등을 포함한다. 디스플레이부에서는 화면에 보여지기 위해서 필요한 객체 정보를 포함한다. 여기서 시각화 하기 위한 3차원 그래픽 데이터는 선언부에서 정의한 의미 있는 객체를 참조하여 나타낼 수도 있다.

AObject는 객체의 의미를 기술하는 Descriptor 태그를 가지고 있어서 '의자', '책상', '책장' 등의 텍스트 기반 검색이 가능하다. Feature 태그는 3차원 객체의 유사도 검색을 하기 위해 존재하는 엘리먼트로써, 자세한 유사도 검색의 방법론은 참고문헌 [5]에 자세히 기술되어 있다.

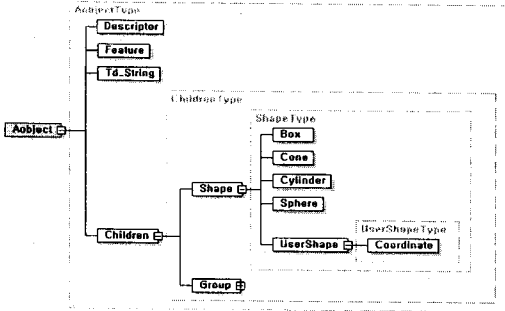


그림 3.2 선언부의 AObject 개념도

하위 Children 타입은 AObject 를 구성하고 있는 3 차원 그래픽 객체들의 정보를 갖는다. Shape 태그에는 기본 객체와 사용자 정의 객체를 포함하고 있으며, Group 태그는 Shape 객체들을 그룹핑 할 수 있도록 한다.

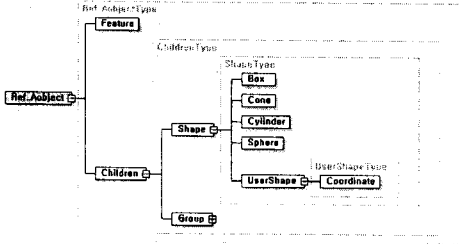


그림 3.3 디스플레이부의 AObject 를 참조하는 Ref\_AObject

디스플레이부에서 Ref\_AObject 라는 엘리먼트를 사용하여 선언부에서 미리 정의된 AObject 를 참조하여 시각화 할 수 있다. AObject 를 참조하되 부분적인 객체의 크기나 위치가 변하였을 경우 Children 이라는 태그를 통하여 변형된 객체의 정보만을 다시 포함시킴으로써, 한 가지 의미 있는 객체를 참조하지만 여러 가지 시각화 객체를 만들 수 있다.

### 3.2 3DXML 의 예

그림 3.4 와 그림 3.5 는 각각 모델링 된 객체와 디스플레이부에서 시각화시킨 객체를 나타낸다. 하나의 타입만을 선언부에 정의해둠으로써 디스플레이부에서는 다수의 참조 객체를 생성할 수 있다.

표 1 은 3DXML 의 선언부를 나타내고 있다. 'a0' oid 를 갖고 있는 AObject 가 선언되어 있으며 의미는 "chair" 이다. 유사도 검색에 필요한 특징값을 Feature 태그에서 갖고 있으며, Children 태그 밑에 'a0'를 구성하는 세 개의 원기둥 정보를 가지고 있다. 각각의 원기둥은 유일한 ID 값을 가지고 있으며, 이것은 시각화 부분에서 수정된 객체 정보만을 갖기 위한 식별자 역할을 한다.

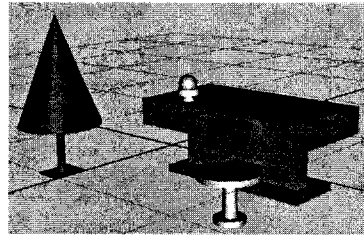


그림 3.4 3차원 모델링 객체

```

<!-- version: 1.0 encoding: UTF-8 -->
<DOCTYPE image SYSTEM "3dxml.dtd" -->
<image>
  <Definition>
    <AObject oid="a0">
      <Descriptor value="chair"/>
      <Feature value="0.7" radius="0.75"/>
      <Children>
        <Shape ShapeId="b0" translation="1.29e+002 -1.52100e+002" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.8941 0.8392 0.6">
          <Cylinder height="0.360" radius="1.149"/>
        <Shape>
          <Shape ShapeId="b1" translation="1.88e+002 -3.1e+002 5.54e+002" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.8941 0.8392 0.6">
            <Cylinder height="3" radius="0.5"/>
          <Shape>
            <Shape ShapeId="b2" translation="0.15 0.0 0.0" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.8941 0.8392 0.6">
              <Cylinder height="0.4297" radius="2.278"/>
            <Shape>
          </Children>
        </AObject>
      <AObject oid="a2">
        <Descriptor value="tee">
          <Feature value="0.8" value="0.67"/>
        ...
  </image>
  
```

표 1. 표 1. 3DXML 선언부 예

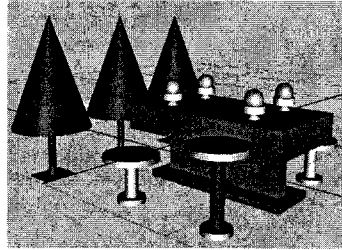


그림 3.5 디스플레이부에서 시각화한 객체

```

<!-- version: 1.0 encoding: UTF-8 -->
<DOCTYPE image SYSTEM "3dxml.dtd" -->
<image>
  <Definition>
    <AObject oid="a0" refId="a0" translation="0 0 0" rotation="0 0 0" scale="1 1 1">
      <Children>
        <Shape RefShape="b1" translation="2.56e+002 -2.56e+002 5.54e+002" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.7">
          <Cylinder height="3" radius="0.5"/>
        <Shape>
          <Shape ShapeId="b2" translation="0.15 0.0 0.0" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.8941 0.8392 0.6">
            <Cylinder height="0.4297" radius="2.278"/>
          <Shape>
        </Children>
      </AObject>
    <Ref_AObject refId="a1" refId="a0" translation="0 0 0" rotation="0 0 0" scale="1 1 1"></Ref_AObject>
    <Shape shapeId="b3" translation="0.115e+003 -2.95900e+002" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.4541176 0.5862353 0.737545">
      <Box size="1.236 3.64 11.05"/>
    <Shape>
      <Shape shapeId="b4" translation="0.9297e+002 -2.37e+002 -2.95900e+002" rotation="0.0 0.0 0.0" scale="1.1 1.1" color="0.2528804 0.9156067">
        <Box size="4.336 0.5494 12.847"/>
      <Shape>
    ...
  </image>
  
```

표 2. 3DXML 의 디스플레이부 예

표 2 는 그림 3.5 를 나타낸 XML 문서를 보여준다. 이 XML 문서에서 보여질 수 있는 3 차원 그래픽 데이터는 크게 두 가지 이다. 하나는 표 1 에서 선언한 'a0' 객체를 참조하여 보이는 'r0', 'r1' 이고, 다른 하나는 의미는 없으나 Scene 에서 필요한 'b3', 'b4' 객체이다. 'r0' 는 미리 선언된 'a0'의 구성 객체들 중에서 'b1'의 ID 를 갖는 기본 객체의 크기, 위치, 색상 정보

를 변경하였으며, 'b2'의 ID를 갖는 기본 객체의 위치 정보를 변경하였다. 그 변경 정보는 Children 이라는 태그에 포함된다. 만약 변경된 정보에 의하여 객체의 특징 값이 크게 변한다면 Ref\_Aobject 하위에 Feature 태그를 명시하여 새로 계산된 특징 값을 갖는다.

4. 시스템 구조

본 논문에서는 3 차원 그래픽 데이터의 다양한 검색을 가능하게 하기 위해 XML 기반의 3DGML 라는 데이터 모델을 제시하였다. 이번 절에서는 3DGML 을 테스트할 시스템 구조에 대해서 설명한다.

3 차원 그래픽을 모델링 하는 도구로는 3DMax 나 MAYA 등의 상용 그래픽 프로그램을 사용한다. 모델링 하는 사람은 상용 프로그램을 사용하여 쉽게 시각화 시킬 수 있다. 단, 모델링 프로그램에서는 3 차원 그래픽 데이터를 VRML 문서로 생성할 수 있어야 한다. 생성된 VRML 문서는 'Sementic Editor' 을 통해서 객체의 특징 정보를 추출하고, 의미를 부여한 후에 3DGML 문서로 생성된다. 이렇게 생성된 문서는 RDBMS 에 저장된다.

사용자는 인터넷 브라우저를 통해서 단순한 텍스트 질의를 할 수 있을 뿐만 아니라, QBE(Query by Example)를 통해서 유사한 3 차원 그래픽 데이터를 검색할 수도 있다.

검색된 결과는 3DGML 문서로 돌려주며, 사용자가 검색 결과를 3 차원 그래픽 데이터로 보고자 할 때는, VRML 로 변환하여 VRML 브라우저를 통해서 디스플레이 된다. VRML 브라우저로는 플러그인 프로그램인 코스모 플레이어 같은 것이 있다. 그림 4 에서는 3DGML 을 테스트할 시스템의 전체 구성을 보여준다.

5. 결론 및 향후 연구

최근 인터넷에서 3 차원 그래픽 데이터가 급속히 증가하고 있다. 이에 따라 웹 데이터베이스 시스템에서도 3 차원 그래픽 데이터의 검색이 요구되고 있다. 3 차원 그래픽 데이터의 단순 질의 검색뿐만 아니라 공간관계 검색, 유사도 검색 등 다양한 방법의 질의 검색

을 가능하게 하기 위해서는 오직 시각화를 위한 마크업 언어가 아닌 새로운 데이터 언어의 모델을 필요로 한다.

본 논문에서는 3DGML 을 사용할 3 차원 그래픽 데이터베이스 시스템의 설계를 제안하였다. 본 논문에서 제시한 시스템은 모델링된 3 차원 그래픽 데이터에서 특징 값을 계산하고 3DGML 로 문서를 추출하였으며 XML 을 지원하는 다양한 기능이 있는 데이터베이스에 추출된 문서를 저장한다. 사용자는 인터넷 브라우저를 통하여 저장된 3DGML 문서를 다양한 방법으로 검색이 가능하고 검색 결과 문서는 VRML 변환기를 통하여 VRML 브라우저로 시각화 될 수 있다.

참고문헌

- [1] Bertino, E, Catania. B. "Integrating XML and databases," Internet Computing, IEEE Volume: 5, Issue: 4, Jul/Aug 2001 P 84-88.
- [2] Kevin Williams 외 9인 저, "XML Databases," 2001
- [3] 노재호, "XML 기반 3 차원 그래픽 데이터베이스 시스템의 설계 및 구현," 한국항공대학교대학원 컴퓨터공학과 석사논문, 2000.
- [4] H. M. Deitel 외 4인 저, "XML How To Program," 2000
- [5] W3 Consortium, "DOM (Document Object Model)," <http://www.w3.org/DOM/>, 2003
- [6] Armed K. Elmagarmid, "Indexing Techniques for Advanced Database System," Kluwer Academic Publishers, 1997.
- [7] David A. White, Ramesh Jain, "Similarity Indexing with the SS-tree," Visual Computing Laboratory, University of California, San Diego, 1996.
- [8] Julio Barros, James French, Worthy Martin, "Indexing multi-spectral images for content-based retrieval," Computer Science Dept. University of Virginia, 1994.
- [9] N. S Chang and K. S. Fu, "Query-by-pictorial-example," IEEE Transactions on Software Engineering, Vol. SE-6, NO. 6, November 1980.
- [10] W3 Consortium, "X3D (Extensible 3D)," <http://www.web3d.org/TaskGroups/x3d>, 2003.
- [11] W3 Consortium, "XML Schema," <http://www.w3.org/XML/Schema>, 2003

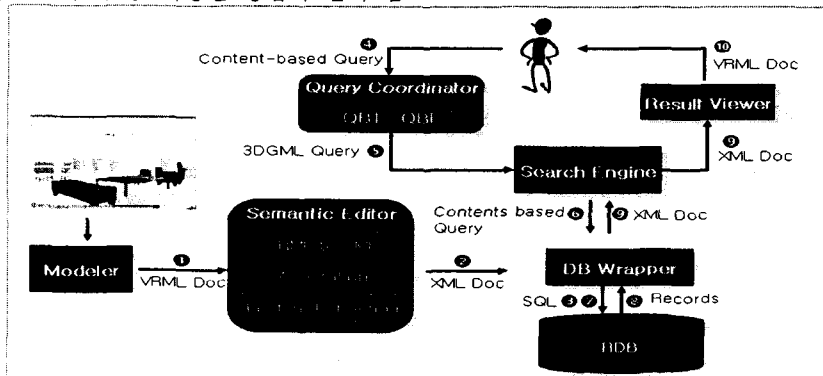


그림 4. 3DGML 구현 시스템 구조