

웹3D 물리 단위 스키마 정의와 구현

김이현 박창섭 이명원

수원대 컴퓨터학과, 동덕여대 컴퓨터학과, 수원대 인터넷정보공학과

kokorian85@naver.com, cspark@dongduk.ac.kr, mwlee@suwon.ac.kr

Schema Definition and Implementation for Web3D Physical Units

Lee Hyun Kim, Chang-Sup Park, Myeong Won Lee

Dept. of Computer, The U. of Suwon, Dept. of Computer Science, Dongduk Women's U.

Dept. of Internet Information Engineering, The U. of Suwon

요약

본 논문에서는 가상환경에서 실세계 물체들의 사실적 시뮬레이션과 표현에 필요한 물체의 물리적 속성 정의를 위해서 물리 단위를 도입하는 방법을 설명한다. 표준화된 물리 단위 규격의 정의를 위하여 물리 단위는 SI 단위계(International System of Units)를 기본으로 하고 가상세계 데이터의 표준화된 규격을 위하여 웹3D 표준인 X3D(Extensible 3D)를 기반으로 하여 물리 단위 명세를 정의한다. 물리 단위를 X3D 규격 안에서 정의하는 과정에서 물리 단위의 적용을 받은 물체가 X3D 장면 안에서 사실적인 변화로 표현되기 위해서는 X3D 전체 스키마에 합당하게 정의되어야 한다. 실세계 물리 단위를 X3D 스키마에 확장하여 정의하고, 확장된 X3D 규격으로 구현된 브라우저에서 물리 단위를 포함하는 물체가 물리 단위 개념이 없었던 기존 X3D 물체가 어떻게 다른가를 물리 단위 예제를 이용하여 비교 분석한다.

Abstract

This paper describes how to define and implement the schema for 3D virtual objects with physical units so that the objects can be compared in virtual environments based on physical properties, such as length, according to the specified units. We define physical units for virtual objects using the International System of Units and based on the X3D (Extensible 3D) specification. The schema must be defined with validation so that it does not violate the original X3D data structure. In this paper, we have extended the original X3D schema with a physical unit specification, and demonstrate the difference between units-specified and non-units-specified 3D scenes using an X3D browser that we developed.

키워드: 가상환경, 가상 물체, 물리적 속성, 물리 단위 명세, SI 단위, 가상현실, 가상환경 스키마

Keywords: Virtual environments, Virtual objects, Physical properties, SI units, Virtual reality, Virtual environment schema

1. 서론

3차원 그래픽스 데이터의 전송이나 상호 교환을 위해서는 Web3D Consortium과 ISO/IEC JTC1 SC24가 공동으로 추진되고 있는 국제표준인 X3D(Extensible 3D)를 사용하게 된다. X3D는 기존의 VRML(Virtual Reality Modeling Language)의 후속 버전으로서 종래의 HTML에서 사용했던 VRML기능을 XML 기반으로 확장시킨 것으로 3D 정보의 교환에 필요한 파일 형식과 런타임 구조를 정의하는 표준이다[1][2][3].

X3D는 컴퓨터그래픽스 기술로 생성된 3D 물체와 가상공간의 표현과 전송에 필요한 모든 정보를 포함하고는 있으나, 현재 대부분의 그래픽스 물체 표현 기술이 가지고 있는 한계라고 할 수 있는, 길이 단위와 같은 물리적 속성에 관한 규격과 구현 방법의 부재로 인해서 실세계 시각 정보나 시뮬레이션 구현에서 실세계 물체의 물리적 속성을 비교할 수 없는 문제가 발생한다. 또한, 물리 속성 중 하나인 시간을 제어하는 컴포넌트가 있으나 동영상의 재생 시작과 종료를 나타내는 기능이고, 사운드를 제어하는 컴포넌트도 있으나 가상공간에서 3D 사운드를 표현하기 위한 음원의 위치를 표현하는 등의 간단한 가시적 표현 기능의 역할만 주어졌다. 이와 같은 3D 장면 내의 실제적인 물리적 속성 제어 기능의 부재로 인한 문제점 해결을 위하여, 본 연구팀에서는 그동안 가상환경에서의 물리 단위를 정의하고 구현하는 방법을 연구해왔다[4][5][6]. 3D 시각 정보에 단위 명시와 관련된 연구는 그다지 이루어지고 있지는 않으나 3D 어플리케이션이 늘면서 최근 이에 대한 필요성이 대두되고 있는 것이 현실이다. X3D에서도 길이에 대한 기본 단위는 명시하고는 있으나 미터(meter)만 기술하고 있고 다른 단위는 사용할 수가 없다. DICOM과 같은 의료 영상이나 영상 스캐너 파일도 픽셀당 기본 단위를 정의하고 있으나 3D 데이터와는 아직 관계가 없다[7].

본 연구팀의 기존의 연구에서는 국제표준인 SI 단위계(International System of Units)를 기반으로 하여 X3D 가상세계에서의 실세계 길이 단위 규격을 구체적인 내용으로 하여 물리적 단위의 정의와 구현을 진행하였다[8]. 본 연구팀의 기존 연구는 X3D에서 길이 단위 정의에 초점을 둔 연구였으나, 본 연구에서는 가상환경 단위 표현을 위하여 국제단위 표준인 SI 단위계에서 길이 단위 외의 다른 기본 단위들을 추가로 추출하고 이를 X3D에 정의할 수 있도록 X3D 스키마를 새로 정의하고 유효성을 검사하였다. 유효성 검사를 하는 이유는 X3D에서 단위 명세를 추가적으로 구현하기 위해서는 기존의 X3D 스키마 구조를 기반으로 하므로 추가하는 단위 규격이 기존의 X3D 스키마 구조에 합당하게 정의되었는지를 확인해야 하기 때문이다. 본 논문에서는 길이 단위를 포함하여 기본 물리 단위 규격을 SI 단위계를 이용하여 새로운 스키마를 정의하였다. 또한, 물리 단위 규격을 이용한 3D 물체 적용 사례와 함께 기존 X3D 시스템과의

차이점에 대해서 논의한다.

2. 3D 물체의 물리 단위 정의

3D Max와 같은 일반 그래픽스 도구에서 각기 별도로 모델링한 3D 물체들을 한 장면으로 들여와서 디스플레이를 하게 되면 X3D에 물리 단위 규격이 명시되어 있지 않아서 범용 브라우저에 의해 잘못 해석되는 경우가 발생한다. 이것은 기존의 물리 단위 개념이 없는 환경에서는 각각 개별적으로 객체들을 표현하고 저장하는 데는 문제가 없었으나 물리 단위의 부재로 실제 상대적인 크기 개념이 없기 때문에 두 개 이상의 물체를 동시에 한 장면에 표현할 때는 의도하지 않은 결과를 가져올 수 있다. 이와는 달리 각 물체에 길이단위를 적용하면 물체의 상대적인 크기 비교가 가능하게 되어 각기 별도로 모델링된 물체들을 한 장면으로 들여올 때 브라우저에 의해 크기 변환될 수 있어서 실세계에서와 같이 크기가 비교되는 사실적인 결과를 얻을 수 있게 된다. 예를 들어 한 디자이너가 건물들을 배치하고 여러 소품들을 가상공간에 배치한 것을 필요에 의해 다른 디자이너가 가져와서 사용하게 되면 물체 크기에 대한 개념이 이미 모호해진다. 이것에 대하여 길이단위를 적용하여 하나의 단위로 통일하게 되면, 이후 사용 시 편의에 따라 크기 단위 기준을 바꿀 수도 있고 기존의 모델에 맞추어 모델링할 수도 있다.

본 연구는 객체 모델에 대한 단위 개념이 없는 X3D 명세에서 모델을 표현하는데 있어서 실세계에서 사용되는 다양한 물리 단위 속성을 정의하고 이를 X3D 규격에서 기술하여 웹3D 브라우저에서 인식될 수 있도록 X3D 스키마를 확장하는 것이다. 본 연구에서의 물리적 단위는 ISO/IEC에서 승인된 SI 단위계의 7개의 물리 단위인 길이(Length), 질량(Mass), 시간(Time), 전류(Electric Current), 열역학온도(Temperature), 물질량(Amount of Substance), 광도(Luminous Intensity)와 함께 특수 단위로써 음압(Sound Pressure)과 각도(Angle)로 정의하였다[8].

이러한 물리 정보들은 X3D에서 정의된 메타 태그(Meta tag)를 이용하여 기술할 수도 있으나, 메타 태그는 X3D의 헤더(Header)에서 선언되는 하나의 구조 전체에 대한 정보이며, 이러한 정보는 X3D 장면 내 물체들에 직접적인 영향을 미치지 못하는 못한다. 즉, 모든 브라우저에 의해 메타 태그의 정보가 장면 내 표현될 것이라는 보장은 없다. 본 논문에서의 물리적 단위들은 헤더가 아닌 X3D 장면(Scene) 내에서 직접 영향을 주는 노드로 정의되고 물리 단위가 X3D 구조 내에서 정확하게 적용될 수 있도록 물리 노드의 스키마를 기존 X3D 스키마에 확장한다.

3. X3D 스키마와 물리 노드

X3D 스키마는 X3D DTD의 기능을 가지며 X3D 노드와 속

성 값들의 타입을 철저히 체크하는 기능을 가진다. X3D 노드는 X3D 노드 타입을 구현하며 XML 요소(element)로 표현된다. X3D 노드가 아닌 필드들은 필드 타입을 이용하여 XML 속성들로 표현된다. 현재 X3D 표준에는 많은 컴포넌트들이 노드로서 정의되어 있고 XML에 기반한 태그(<>)의 문법구조를 이루고 있다. 기본적으로 문서의 모든 내용은 버전규약을 포함하는 <X3D>노드로 시작되고 </X3D>로 끝난다. <HEAD>노드는 본 문서에 대한 머리글을 의미하며 meta 태그는 이곳에 사용하여 문서의 정보를 기술한다. <SCENE>노드는 가상공간에서 표현되는 물체들에 대한 정보를 가지고 그 하위에는 <SHAPE>노드를 두어 3D 물체의 형상과 외형적 특징을 표현한다.

X3D 스키마는 X3D 장면을 생성하고 관리하는 전체 구조를 설명하며 수많은 노드와 필드들로 구성된다. 노드는 XML 요소로서 기술되며 그 종류는 수십 개 존재하고, 물리 노드가 X3D 장면에서 원활하게 작동하도록 하기 위해선 기존의 X3D 스키마 구조에 합당하게 물리노드가 정의되어야 한다. X3D 스키마 확장은 물리 노드 정의에 있어서의 아래 주요 사항이 실현되도록 구성한다:

- X3D 장면에서의 물리 단위는 SI 기본 단위 (SI Basic Units)를 기반으로 한다.
- X3D 물리 단위는 하나의 X3D 파일에서 한 번 정의한다.
- 여러 개 X3D 파일이 하나의 장면으로 합병되어 들어올 때에 통합된 장면에서의 물리 단위는 첫 번째 X3D 파일에서 정의된 물리 단위를 따른다.
- 하나의 X3D 파일에서 정의된 길이 단위는 장면 상에서 물체 외관에 영향을 미치지 않는다.
- 둘 이상의 X3D 파일에서 정의된 길이 단위가 한 장면에 합병되어 들어올 때는 첫 번째 X3D 파일에 길이 단위에 따라 후속 파일들의 단위가 크기변환되어 들어온다.
- SI 기본 단위 중 길이 단위 외 다른 물리 단위명제는 X3D 파일간 혹은 응용간의 표준화된 물리 속성 전달의 의미를 갖는다.

4. 물리단위 스키마 정의

X3D 파일 내에서 물리 단위 스키마의 정의는 그림 1과 같이 네임스페이스 선언으로 정의된다. 즉, URL 에 존재하는 스키마 파일이 기존 X3D 스키마를 확장하는 내용이다.

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.web3d.kr/mwlee/x3d-3.1-ext"
elementFormDefault="qualified">
```

그림 1: 스키마 파일 확장

XML 의 <redefine>선언으로 복수의 XML 스키마를 합치면 외부 스키마에 선언된 복합 타입, 단순 타입, 모델 그룹,

속성(attribute) 그룹을 수정할 수 있다(그림 2). 이를 이용하여 물리 단위 노드를 삽입하여 X3D 스키마를 확장한다. 삽입된 물리단위의 구조는 SceneGraphStructureNodeType이라는 이름의 전역 타입으로 생성되는데 혼합모드가 아니므로 abstract와 mixed는 default값인 false로 정의한다. 하위에는 0에서 1번 선언될 수 있는 Physical 엘리먼트(element)를 정의한다.

```
<xs:redefine schemaLocation =
"http://www.web3d.org/specifications/x3d-3.1.xsd">
<xs:annotation>
<xs:documentation> Insert Physical element into
SceneGraphStructureNodeType
</xs:documentation>
</xs:annotation>
<xs:complexType name="SceneGraphStructureNodeType"
abstract="false" mixed="false">
<xs:complexContent>
<xs:extension base="SceneGraphStructureNodeType">
<xs:sequence>
<xs:element name="Physical"
type="PhysicalUnitType"
minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:redefine>
```

그림 2: 스키마 재정의

이 노드는 세부적으로 9개의 엘리먼트 중 하나를 선택할 수 있게 정의하였다. 그리고 각각의 노드는 0에서 1번의 횟수로 X3D 스키마 상에 선언될 수 있다(그림 3).

```
<xs:complexType name="PhysicalUnitType">
<xs:choice>
<xs:element name="Length" type="LengthType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Mass" type="MassType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Time" type="TimeType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Current" type="CurrentType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Temperature" type="TemperatureType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Substance" type="SubstanceType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Luminous" type="LuminousType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="SoundPressure"
type="SoundPressureType"
minOccurs="0" maxOccurs="1"/>
<xs:element name="Angle" type="AngleType"
minOccurs="0" maxOccurs="1"/>
</xs:choice>
</xs:complexType>
```

그림 3: 물리 단위 엘리먼트 정의

다음 각 절에서는 각 물리 단위에 대한 스키마 확장을 기술한다.

4.1 LengthType

LengthType은 길이 단위를 정의하는데 실제 사용되는 SI 단위계의 실제 단위를 포함한다(그림 4). 그리고, 사용자의 편의에 맞게 사용할 수 있도록 하는 USER 단위를 추가하여 사용자가 직접 단위를 정의하여 쓸 수 있도록 하였다(그림 5).

또한 선택적으로 부모 요소에 의해 기술된 요소들 안에 세 가지 세부 속성을 정의하였다. 먼저 unit 속성은 default로 UNI(meter)를 가지며 그 밖에 실세계에서 사용되는 단위를 정의하고, basis 속성은 사용자 단위를 지정할 때 미터당 단위설정 및 속성이 적용된 단위에 대한 비율(축척)을 정의한다. numeral 속성은 단위 값의 표기 유형으로 건축, 과학, 수학, 분수, 십진 표기 유형들로 전문 분야에서 많이 사용되는 표기법을 정의하였다.

```
<xs:complexType name="LengthType">
  <xs:attribute name="unit" type="LengthUnitType"
    use="optional" default="UNI"/>
  <xs:attribute name="basis" type="xs:double"
    use="optional" default="1"/>
  <xs:attribute name="numeral" type="LengthNumeralType"
    use="optional" default="DEC"/>
</xs:complexType>
<xs:simpleType name="LengthUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of length.
      default is "UNI" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="YOTTA|ZETTA|EXA|PETA|TERA|
      GIGA|MEGA|KILO|HECTO|DECA|UNI|DECI|
      CENTI|MILLI|MICRO|NANO|PICO|FEMTO|
      ATTO|ZEPTO|YOCTO|INCH|LINK|FT|YD|RD|
      CHAIN|FL|MILE|LG|MIL|AU|LY|PC|KPC|
      NMILE|ANG|USER"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="LengthNumeralType">
  <xs:annotation>
    <xs:documentation>values for numeral attribute
      of length. default is "DEC"
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="SC|EN|DEC|ENG|IN|ARCH|FRAC"/>
  </xs:restriction>
</xs:simpleType>
```

그림 4: 길이 단위 정의

Label	Symbol	Concept Definition	Code
USER	user	10 ⁰	37

그림 5: 사용자 정의 단위

4.2 MassType

MassType은 질량 단위를 정의한다(그림 6). 선택적으로 사용 가능한 unit 속성은 기본값으로 KG(Kilogram)을 가진다. 그 밖에 실제 사용되는 질량 단위들을 추가적으로 정의하고, value 속성은 기본값으로 1을 가지며 unit 속성에 따른 모델의 실제 질량값을 포함한다.

```
<xs:complexType name="MassType">
  <xs:attribute name="unit" type="MassUnitType"
    use="optional" default="KG"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="MassUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of mass.
      default is "KG" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="MG|G|KG|LB|TON|GRAIN|OZ"/>
  </xs:restriction>
</xs:simpleType>
```

그림 6: 질량 단위 정의

4.3 TimeType

TimeType은 시간 단위를 정의한다(그림 7). 선택적으로 사용 가능한 unit 속성은 기본값으로 초(Second)를 가진다. 그 밖에 실제 사용되는 시간 단위들을 추가적으로 정의하고, value 속성은 기본값으로 1을 가지며 unit 속성에 따른 모델의 실제값을 포함한다.

```
<xs:complexType name="TimeType">
  <xs:attribute name="unit" type="TimeUnitType"
    use="optional" default="S"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="TimeUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of time.
      default is "S" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value=""/>
  </xs:restriction>
</xs:simpleType>
```

그림 7: 시간 단위 정의

4.4 CurrentType

CurrentType은 전류량의 단위를 정의한다(그림 8). 선택적으로 사용 가능한 unit 속성은 기본값으로 A(Ampere)을 가진다. 그 밖에 실제 사용되는 전류량의 단위들을 추가적으로 정의하고, value 속성은 기본값으로 1을 가지며 unit 속성에 따른 모델의 실제값을 포함한다.

```
<xs:complexType name="CurrentType">
  <xs:attribute name="unit" type="CurrentUnitType"
    use="optional" default="A"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="CurrentUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of current.
      default is "A" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value=""/>
  </xs:restriction>
</xs:simpleType>
```

그림 8: 전류량 단위 정의

4.5 TemperatureType

TemperatureType은 온도 단위를 정의한다(그림 9). 선택적으로 사용 가능한 unit 속성은 기본값으로 K(Kelvin)을 가진다. 그 밖에 실제 사용되는 온도 단위들을 추가적으로 정의하고, value 속성은 기본값으로 1을 가지며 unit 속성에 따른 모델의 실제값을 포함한다.

```
<xs:complexType name="TemperatureType">
  <xs:attribute name="unit" type="TemperatureUnitType"
    use="optional" default="K"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="TemperatureUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of temperature.
      default is "K" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="OC|OF|K|OE"/>
  </xs:restriction>
</xs:simpleType>
```

그림 9: 온도 단위 정의

4.6 SubstanceType

SubstanceType은 물질량의 단위를 정의한다(그림 10). 선택

적으로 사용 가능한 unit 속성은 기본값으로 Mol(Mole)을 가진다. 그 밖에 실제 사용되는 물질량의 단위들을 추가적으로 정의하고, value속성은 기본값으로 1을 가지며 unit 속성에 따른 모델의 실제값을 포함한다.

```
<xs:complexType name="SubstanceType">
  <xs:attribute name="unit" type="SubstanceUnitType"
    use="optional" default="MOL"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="SubstanceUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of substance.
      default is "MOL" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value=""/>
  </xs:restriction>
</xs:simpleType>
```

그림 10: 물질량 단위 정의

4.7 LuminousType

LuminousType은 광도 단위를 정의한다(그림 11). 선택적으로 사용 가능한 unit 속성은 기본값으로 CD(Candela)를 가진다. 그 밖에 실제 사용되는 광도 단위들을 추가적으로 정의하고, value 속성은 기본값으로 1을 가지며 unit 속성에 따른 모델의 실제값을 포함한다.

```
<xs:complexType name="LuminousType">
  <xs:attribute name="unit" type="LuminousUnitType"
    use="optional" default="CD"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="LuminousUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of luminous.
      default is "CD" </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value=""/>
  </xs:restriction>
</xs:simpleType>
```

그림 11: 광도 단위 정의

4.8 SoundPressureType

SoundPressureType은 음압의 단위를 정의한다(그림 12). 선택적으로 사용 가능한 unit 속성은 기본값으로 dB (Decibel)을 가진다. 그 밖에 실제 사용되는 음압 단위들을 추가적으로 정의하고, value 속성은 기본값으로 1을 가지며

unit 속성에 따른 모델의 실제 값을 포함한다.

```
<xs:complexType name="SoundPressureType">
  <xs:attribute name="unit" type="SoundPressureUnitType"
    use="optional" default="DB"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="SoundPressureUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of sound pressure.
      default is "DB"</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value=""/>
  </xs:restriction>
</xs:simpleType>
```

그림 12: 음압 단위 정의

4.9 AngleType

AngleType은 각도 단위를 정의한다(그림 13). 선택적으로 사용 가능한 unit 속성은 기본적으로 RAD(Radian)을 가진다. 그 밖에 실제 사용되는 각도 단위들을 추가적으로 정의하고, value 속성은 기본적으로 1을 가지며 unit 속성에 따른 모델의 실제값을 포함한다.

```
<xs:complexType name="AngleType">
  <xs:attribute name="unit" type="AngleUnitType"
    use="optional" default="RAD"/>
  <xs:attribute name="value" type="xs:double"
    use="optional" default="1"/>
</xs:complexType>
<xs:simpleType name="AngleUnitType">
  <xs:annotation>
    <xs:documentation>values for unit attribute of Angle.
      default is "RAD"</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value=""/>
  </xs:restriction>
</xs:simpleType>
```

그림 13: 각도 단위 정의

5. 물리 단위 스키마 유효성 및 뷰어 구현

앞 절에서 정의된 물리 단위 스키마의 유효성은 Eclipse 와 X3D-Edit 를 사용하여 검증하였다. 본 연구에서 구현된 뷰어는 이 스키마를 기본으로 해서 물리 단위를 확장한 X3D 파일을 기존의 뷰어에서와는 달리 실제 단위를 적용하여 디스플레이한다. 단위의 적용은 하나의 X3D 파일이 디스플레이될 때는 기존의 뷰어와 동일하나 둘 이상의 X3D 파일이 한 장면에 통합되어 들어올 때는 상대적 단위가 적용된 결과를 보여준다. 본 절에서는 서로 다른 3D 제작 환경에서

생성된 물체들을 한 장면에 들어왔을 때 어떠한 문제를 일으킬 수 있는지를 분석하기 위해서, 3D 장면에서 물체를 표현할 때 물리 단위를 적용한 경우와 적용하지 않은 경우를 비교하여 설명한다.

서로 다른 제작 환경에서 모델링을 할 경우 디자이너의 의도에 따라 모델의 크기 비율이 커다란 차이를 가질 수 있다. 예를 들어 3D MAX와 같은 모델링 도구로 물체를 제작한 뒤에 모델링 도구와는 다른 X3D 뷰어 환경으로 읽어들이는 경우에 처음에 디자이너가 생각했던 물체의 길이 단위가 적용되지 못한 채로 뷰어에 표현되게 된다. 이 때 물체들을 개별적으로 들여올 경우에는 아무런 문제가 없어 보이지만 이들을 함께 읽어들이 한 화면에 표현하고자 하면 문제가 발생한다. 그림 14는 두 미생물인 클라미도모나스와 박테리아를 표현한 것으로, 클라미도모나스의 실제 크기는 0.01-0.03 mm이고, 박테리아는 0.4-1µm 이다.

보통 물체를 모델링하는 디자이너가 각 미생물 제작사에 적용하는 스케일은 X3D 파일로 저장될 때 Transform 노드로 표현된다. 이것은 각 모델별로 서로 다른 사용자가 설정한 Transform 값을 가지고 오기 때문에 두 물체가 같이 들어올 경우 두 물체간 서로 비교할 수 있는 스케일이 아니므로 상대적 크기 표현이 어렵다. 그림 15는 이 경우의 두 물체를 보여주며 결국 실제와는 달리 박테리아가 클라미도모나스보다 훨씬 큰 크기의 미생물로 잘못 표현된 결과를 나타낸다.

이와는 달리 길이 단위를 적용하게 되면 두 물체간 상대적인 크기 변환이 정확히 이루어져서 그림 16과 같이 박테리아와 클라미도모나스의 실제 크기 비교가 가능하게 된다. 이 때 크기 비교는 최초로 들어온 모델의 단위를 기준으로 하며 이후 렌더링되는 모델은 상대적 단위가 적용된다.



그림 14: 클라미도모나스(좌)와 박테리아(우)

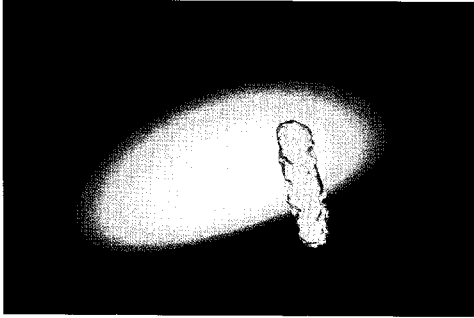


그림 15: 길이 단위 적용 전

이와 같이, 그동안 미생물에서 천체까지 모든 크기의 물체를 표현할 때 기존의 X3D 뷰어에서 발생할 우려가 있는 부동소수점 오류 없이, 표현하기 힘들었던 실세계 대부분의 물체를 실제 단위의 크기로 물체들을 비교할 수 있게 되었다. 그림 17은 앞에서 기술한 물리 단위 스키마 정의를 이용하여 미생물인 클라미도모나스와 박테리아의 물리 단위를 기술한 것이다.

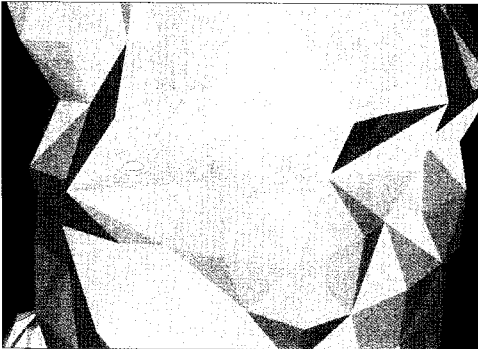


그림 16: 길이 단위 적용 후

```
// 클라미도모나스 단위 지정
<physical>
  <length unit="MILLI" basis="1" numeral="DEC"/>
</physical>

// 박테리아 단위 지정
<physical>
  <length unit="MICRO" basis="1" numeral="DEC"/>
</physical>
```

그림 17: 미생물의 물리 단위 정의

그림 18에서는 하드디스크를 사용하고 모니터는 센티미터를 사용하여 단위를 정의한 또 다른 예를 보여 준다. 이와 같이 길이 단위를 다른 모델에 적용시키도 상대

적인 크기를 보여주게 되므로 cm로 정의된 모니터와 mm로 정의된 하드디스크를 한 화면에서 비교할 수 있다. 그림 20은 하드디스크와 모니터에 길이 단위를 정의한 것이다. 그림 20과 그림 21은 60,000km 크기의 토성과 250,000 hectometer (25,000km) 크기의 천왕성의 경우를 설명한다.

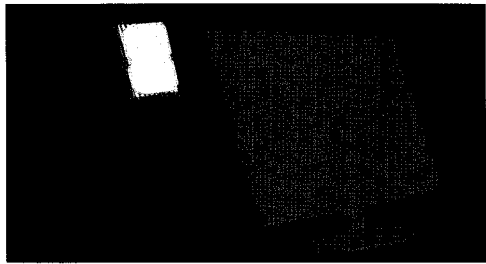
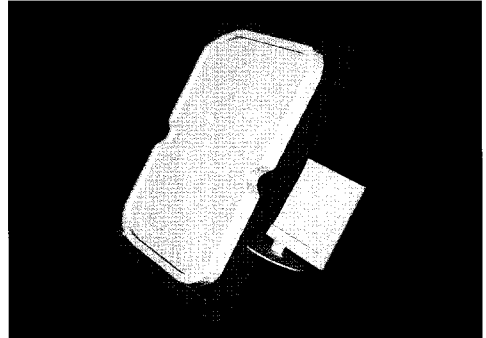
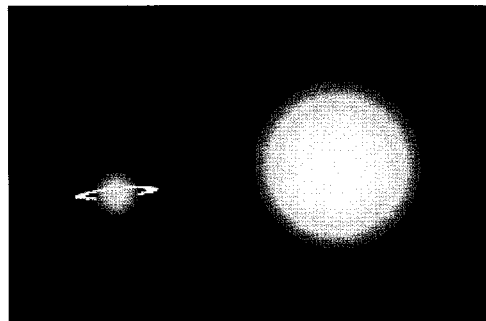


그림 18: 길이 단위 적용 전(위), 적용 후(아래)

```
// 하드디스크 길이 단위 지정
<physical>
  <length unit="MILLI" basis="1" numeral="DEC"/>
</physical>

// 모니터 길이 단위 지정
<physical>
  <length unit="CENTI" basis="1" numeral="DEC"/>
</physical>
```

그림 19: 하드디스크와 모니터의 길이 단위 정의



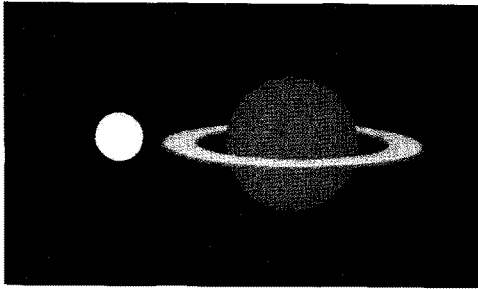


그림 20: 길이 단위 적용 전(위)과 적용 후 (아래)

```
// 천왕성의 길이 단위 지정
<physical>
  <length unit="HECTO" basis="1" numeral="DEC"/>
</physical>

// 토성의 길이 단위 지정
<physical>
  <length unit="KILO" basis="1" numeral="DEC"/>
</physical>
```

그림 21: 천왕성과 토성의 길이 단위 정의

[4] 김수현, 이명원, “웹3D에서의 물리적 측정 단위 컴포넌트 명세”, 김수현, 이명원, *정보과학회논문지: 컴퓨팅의 실제 및 레터*, 제 15 권, 제 6 호, pp.454-460, 2009.6.

[5] 정철희, 임창혁, 이민근, 이명원, “가상환경에서의 실세계 물리적 길이 비교”, *한국컴퓨터그래픽스학회 논문지*, 제13권, 제3호, pp. 19-24, 2007년 9월.

[6] 이명원, 임창혁, 이용덕, “가상 객체의 실제 길이 표현을 위한 다중 레벨”, *한국컴퓨터그래픽스학회 논문지*, 제 13권, 제3호, pp. 25-31, 2007년 9월.

[7] DICOM Standard, “Digital Imaging and Communications in Medicine”, *The National Electrical Manufacturers Association(NEMA)*, 2009.

[8] “SI Units,” *The International System of Units (SI)*, *International Bureau of Weights and Measures*, February 2007.

6. 결론

현재 X3D 명세서에는 실세계 모든 물체의 3D 데이터 표현을 위한 많은 컴포넌트들이 존재하나 실세계에서와 같이 물체의 물리적 속성을 나타낼 수 있는 단위에 대한 컴포넌트가 아직 정의되어 있지 않다. 본 연구는 실세계 물체를 가상환경에 표현할 때 물체의 물리적 속성 정보를 나타내기 위해서 국제 단위 표준인 SI 단위계를 사용하기 위한 스키마 정의에 중점을 두었다. 본 논문에서의 물리 단위 스키마와 단위 정의 방법은 현재 ISO/IEC JTC1 SC24 와 Web3D Consortium에서 표준화가 진행 중이다. 가상환경에서의 물리 단위 국제표준화에 의해서 3D 데이터를 이용하는 모든 정보 서비스와 비즈니스 분야에서 3D 물체의 정확한 속성 표현이 가능하게 되어 CAD, 의료, 이러닝, GIS 등 많은 영역에서 새로운 디지털 정보 서비스를 제공하게 될 것이다.

참고 문헌

[1] ISO/IEC 19775-1:2004(E), *Information technology - Computer graphics and image processing - Extensible 3D (X3D), Part 1: Architecture and base components*, 2004.

[2] ISO/IEC 19776-1:2005(E), *Extensible 3D (X3D) encodings, Part 1: Extensible Markup Language (XML) encoding*, 2005.

[3] Don Brutzman and Leonard Daly, *Extensible 3D Graphics on Web Authors*, *Morgan Kaufman*, 2007.

〈저자소개〉



김이현

- 2010년 수원대학교 학사
- 관심분야 : 컴퓨터그래픽스, 가상현실, 서버 프로그래밍



박창섭

- 1995년 한국과학기술원 전산학과 학사
- 1997년 한국과학기술원 전산학과 석사
- 2002년 한국과학기술원 전자전산학과 박사
- 2002년~2005년 (주)KT R&D 센터 책임연구원
- 2005년~2009년 수원대학교 인터넷정보공학과 전임강사, 조교수
- 2009년~현재 동덕여자대학교 컴퓨터학과 조교수
- 관심분야: XML data management, Semantic Web, Information Retrieval



이명원

- 1981년 서울대학교 학사
- 1984년 서울대학교 대학원 계산통계학과 (전산전공) 석사
- 1990년 The Univ. of Tokyo 대학원 정보과학과 박사
- 1984년~1986년 (주)테이콤 연구소 연구원
- 1990년~1993년 일본 Kubota Corporation 연구원, The Univ. of Tokyo 연구원
- 1993년~1996년 (주)KT 멀티미디어연구소 선임연구원
- 1996년~2002년 수원대학교 컴퓨터학과 전임강사, 조교수
- 2002년~현재 수원대학교 인터넷정보공학과 부교수
- 관심분야: 컴퓨터그래픽스, 가상현실, 컴퓨터애니메이션