April 25, 2008

Myeong Won Lee, mwlee@suwon.ac.kr

The Specification of Physical Property Units in X3D

1. Introduction

1.1 Name

The name of the component introduced is ***Physical Property Units***. The name is used when referring to the component that specifies the units of length, area, volume, mass, electric current, temperature, substance, luminous intensity, and sound pressure in an X3D virtual world.

1.2 Overview

This document explains the physical property units of measurement in X3D. It discusses the node for defining the component; the physical units of length, area, volume, etc.; and the method of representing the physical property units in an X3D browser. (In this paper, we focus primarily on the Length Unit)

2. Concepts

2.1 Physical Property Units

This concept arises from the need to represent real world units of measurement in virtual worlds. In the current X3D specification, it is difficult to represent a virtual object's physical properties using various units of measurement such as length (meter), mass (kilogram), electric current (ampere), thermodynamic temperature (kelvin), amount of substance (mol), luminous intensity (candela), and sound (decibel). However, there are many applications in a variety of industries that require this representation, either visually or informatively.

For example, in medicine, to represent a 3D visual image of a part of the human body produced by reflected sound waves, we need to specify a sound unit such as decibel (dB). Figure 1 shows an example of a sonogram that requires the specification of sound (dB) or velocity (cm/s) units to display.

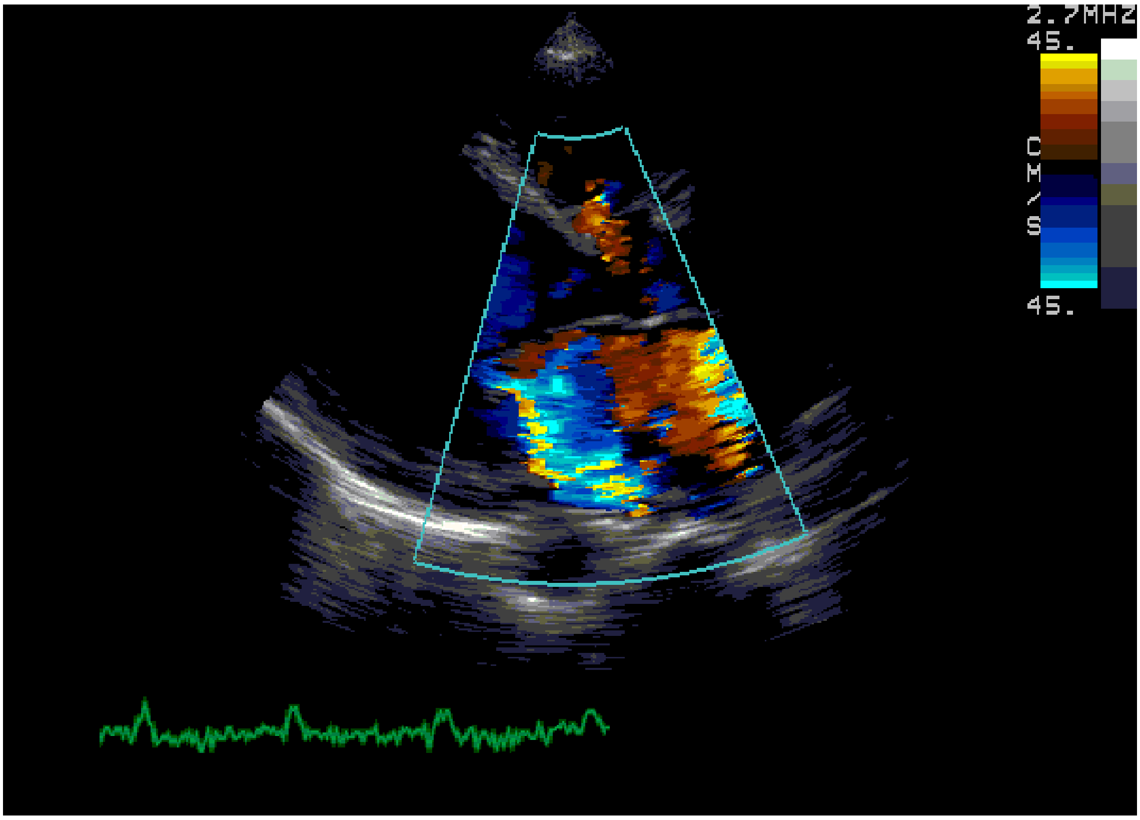


Figure 1 A medical application of sound data (sonogram)

In another example, knowledge of products' real length, area, or volume are factors important in making purchasing decisions at an online shopping mall. With the real length specification, buyers are able to visually determine how two products compare, even though they are offered for sale at different online shops and their display may have been generated by different graphics tools.

The specification defines the units of physical properties to provide realistic representation of objects and valuable information to be used when visualizing objects, comparing objects, and providing physical information scientifically. The physical property units are defined based on SI (International System of Units) and include units of measurement in addition to length, such as area, volume, and sound pressure. Length units allow objects to be visualized according to their scales, while the other units are included to represent objects' properties inside a scene graph so that they can be retrieved whenever required for any given application. Defining physical property units to be used in an X3D world is more powerful, more accurate, and more valid for scientific applications than using a meta tag because the physical units can be specified for many shapes in an X3D scene graph.

The advantages of physical length definition over metadata usage are as follows: The metatada is located in the header of an X3D file, and affects all of a scene graph. The physical length can define the units per each shape, as well as under a scene graph. The physical unit specification is more readable, while the metatada description may not be clear if it represents a scale transformation or it means a unit specification when there are no comments inside the metadata tag. Utilizing the metadata description for other purposes when using X3D files is complex. It provides only limited information, and it is not clear how to use it per shape object in other applications. In contrast, the physical units specification is clear and readable when using it for other applications.

2.2 Physical Length Properties

2.2.1 Physical Length Unit Concept

Current X3D provides only one unit of measurement - the meter. As a result, applications are restricted when representing objects in other real length units such as kilometer, millimeter, or angstrom. There is no choice but to use unscientific representation, especially in astronomical or microbiological units. In addition, it is very difficult to combine two X3D objects in a scene realistically, i.e. physically correctly related to each other, when the objects are generated independently by different graphics tools (see Figure 2).

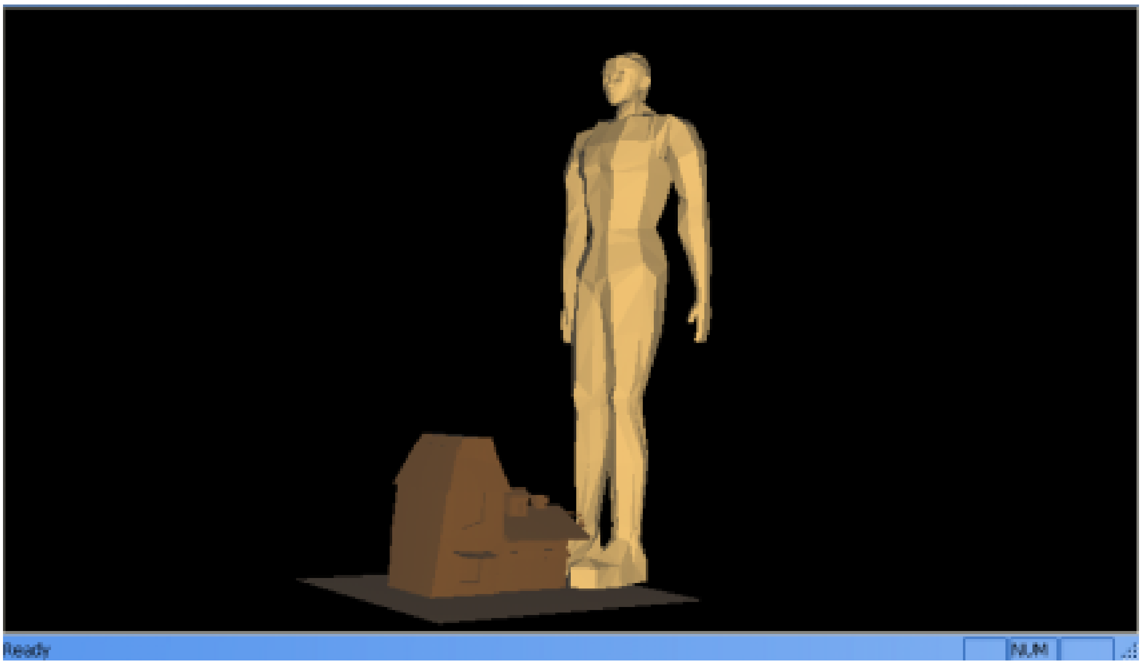


Figure 2 Two X3D files *without* length unit specification

Our specification addresses this issue by allowing for precise representation of objects in X3D worlds using various units of measurement. The physical length unit is designed to be applied not to each object but to the overall scene, i.e. an X3D file. Also, the physical length unit specification is different from scale transformation. We can perform any geometric transformation such as translation, rotation, and scaling within the unit specification.

Figure 2 shows the result of two X3D files generated using separate graphics tools and read into a scene together without specifying the length units. As we can see, the coordinates in the X3D files cannot be scaled according to the real sizes of the objects because, since the two X3D objects were generated independently, the X3D browser does not know what their real sizes are from their coordinates.

In contrast, Figure 3 shows the result of two X3D files read into a scene with length unit specifications - the person has a length unit specification of centimeter (10^{-2}), and the house has a length unit specification of meter (10^{0}). An X3D browser can scale the objects' coordinates according to the unit specification. As a result, we can compare their relative sizes as in the real world.

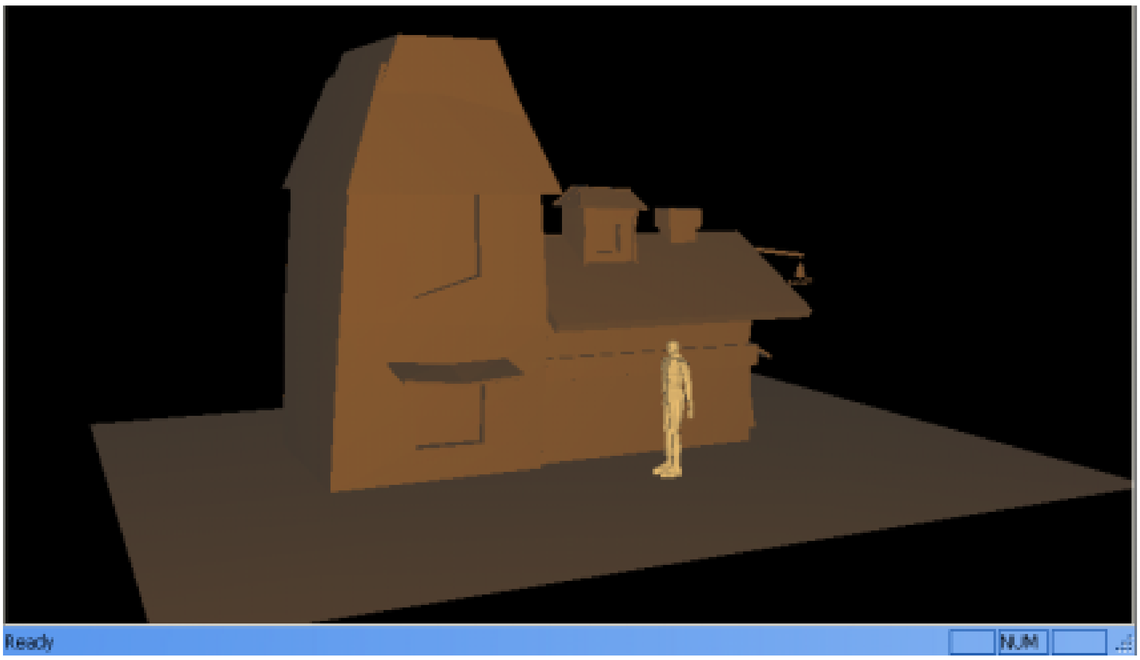


Figure 3 Two X3D files *with* length unit specification

Physical length units refer to all units when measuring an object’s length or size, as well as when comparing objects' lengths.

2.2.2 Definition of Physical Length Properties

The physical length node is defined only once in an X3D file. It is located at the top of a scene graph. Therefore its property affects all of the scene graph. This is different from the Transform Node which can be used multiple times, once for each object represented by a Shape Node. It is also possible to provide LOD (Levels of Detail) functions necessary for objects when the size difference between the objects is large and, therefore, normally difficult to represent together in a scene due to limitations of the display resolution.

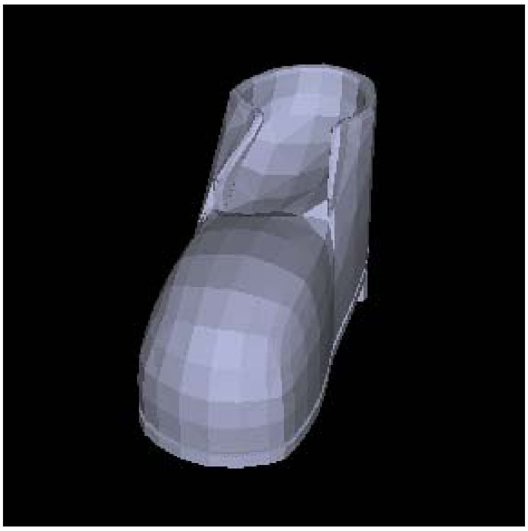
Usage of the physical length unit is as follows:

```
<Scene>
  <physical>
        <length Unit = "MILLI"   basis = ”1”   numeral = ”DEC”/>
    <Transform scale = "15 15 15"   rotation = " 0 1 0 110">
      <Shape>
          <Appearance>
              <Material diffuseColor="0.6588 0.6824 0.8624" />
          </Appearance>
        // Left shoe coordinates generated by 3ds Max
          <IndexedFaceSet coordIndex="   0, 586, 294, -1, 294, 589, 0, -1, 64, 587, 294, -1,
```

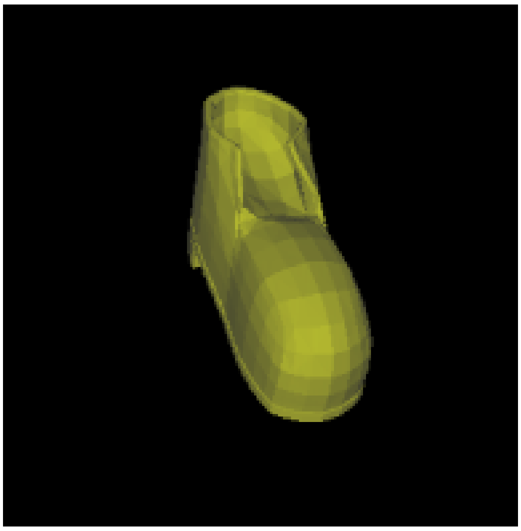
```
                    294, 586, 64, -1, 79, 588, 294, -1, 294, 587, 79, -1, 78, 589, 294, -1,
                    294, 588, 78, -1, 1, 590, 295, -1, 295, 591, 1, -1, 65, 593, 295, -1,
                    295, 590, 65, -1, 66, 592, 295, -1, 295, 593, 66, -1, 2, 591, 295, -1,
                    …………………. ">
          <Coordinate point=" 3.884 -0.239 -1.642, 0.3064 -0.2191 -1.588, -0.5596
                    -0.2191 -1.43, -4.755 -0.5885 -0.7722, 3.551 -0.5164 0.5441,
                    -0.1689 -0.5353 0.8185, -0.9928 -0.559 1.583, -4.655 -0.5884 2.001,
                    3.76 -4.038 -1.642, 0.3064 -4.131 -1.588, -0.5596 -4.131 -1.43,
                    -4.602 -3.405 -0.773, 3.474 -3.633 0.6905, -0.1689 -3.694 0.9864,
                    -0.9928 -3.662 1.761, -4.496 -3.367 2, 4.59 -0.6666 -1.648,
                    0.31 -0.6411 -2.299……………………………………….. "/>
          </IndexedFaceSet>
      </Shape>
    </Transform>
  </physical>
</Scene>
```

This is a simple example including a Length Node. The unit field denotes the unit of measurement, in this example millimeter (10^{-3}); the basis field denotes the basic scale unit, in this example 1 for metric system; and the numeral field denotes a numeral system, in this example decimal. The left shoe also has scale transformation by [15 15 15]. This is to scale the size of the shoes when looking in the scene independent of the physical unit defined. Figure 4a shows the left shoe defined by the above X3D. The right shoe is similarly defined (Figure 4b). Then, Figure 4c shows the right shoe read into the scene using an Inline Node after the left shoe was displayed. The same results are obtained even when different physical length units are defined for the shoes, such as centimeter for the left shoe and meter for the right shoe

(a)



(b)



(c)

Figure 4 (a) Left shoe (b) Right shoe

(c) Integration of separately designed models with Inline Node

Normally an inaccurate scene such as that shown in Figure 5 is obtained if we read multiple X3D objects together using Inline Nodes. Here, the three objects - table, cup, and chair - overlap without relative scalability in the scene. However, this phenomena is avoided when we make use of the physical length unit.

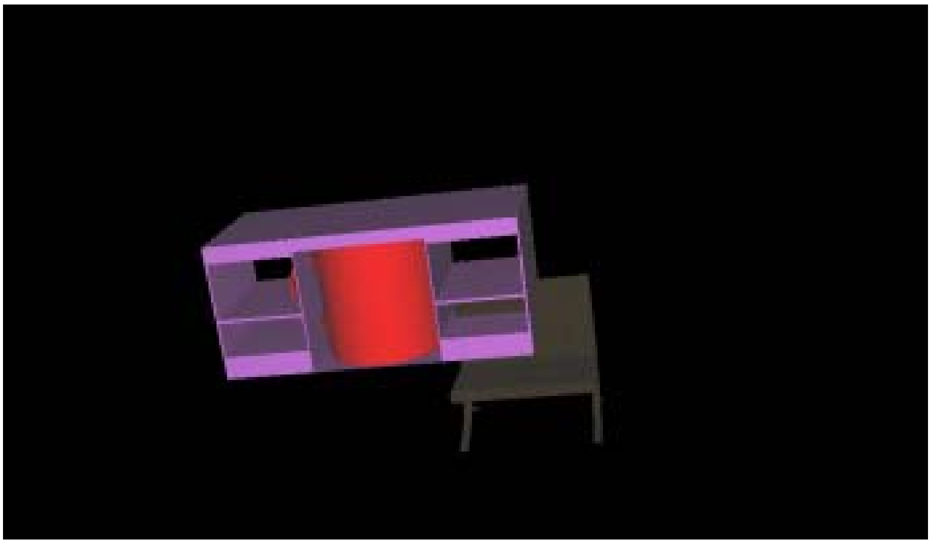


Figure 5 Combination of three X3D Objects *without* Physical Length Unit

The next example shows that multiple objects in a file can be defined with the physical length unit, and also that multiple objects with different physical length units in multiple X3D files can be scaled into a scene according to the units defined. Within the units defined, we can use any number of Transform Nodes to scale or to do any other transformation the same as in the original X3D. The transformation parameters assigned in the Transform Nodes are applied according to the physical length unit. In the following X3D files, the first defines two objects with a common physical length unit in a single file, and the second defines a single object with a different physical length unit in another file.

```
// Two objects defined with a common physical length unit in an X3D file
```

```
<Scene>

  <physical>

            <length Unit = "CENTI"   basis = "1"   numeral= "DEC" />

    <Transform scale="1 1 1"   translation = "0 0 0">

        <Shape>

            [………] // Coordinates for a chair

        </Shape>

      </Transform>

    <Transform scale="1.5 1.5 1.5"   translation = "8 -4 10" >

        <Shape>

            [………] // Coordinates for a table

        </Shape>

      </Transform>

   </physical>

 </Scene>
```

```
// One object defined with a different physical length unit in an another X3D file

<Scene>

   <physical>

            <length Unit = "MILLI"   basis = "1"   numeral= "DEC" />

    <Transform scale="10 10 10"   translation = "10 5 10">
```

```
      <Shape>
          [……..] // Coordinates for a cup
      </Shape>
    </Transform>
  </physical>
</Scene>
```

In the first X3D file, multiple objects, including a chair and a table, are defined with centimeter (10^{-2}) as their physical length units. In the second Shape node, the table is scaled by 1.5 times, and translated by 8cm, -4cm, and 10cm according to x, y, and z axes respectively in the Transform node. In the second X3D file, a cup was defined with millimeter (10^{-3}) as its physical length unit. The cup was scaled by 10 times and then translated by 10mm, 5mm, and 10mm according to x, y, and z axes respectively in the Transform node. In Figure 6, we see the three objects combined in a scene while realizing their relative sizes. In this way, with the unit definition, a precise scaling factor is applied and a precise comparison in length or size can be made.

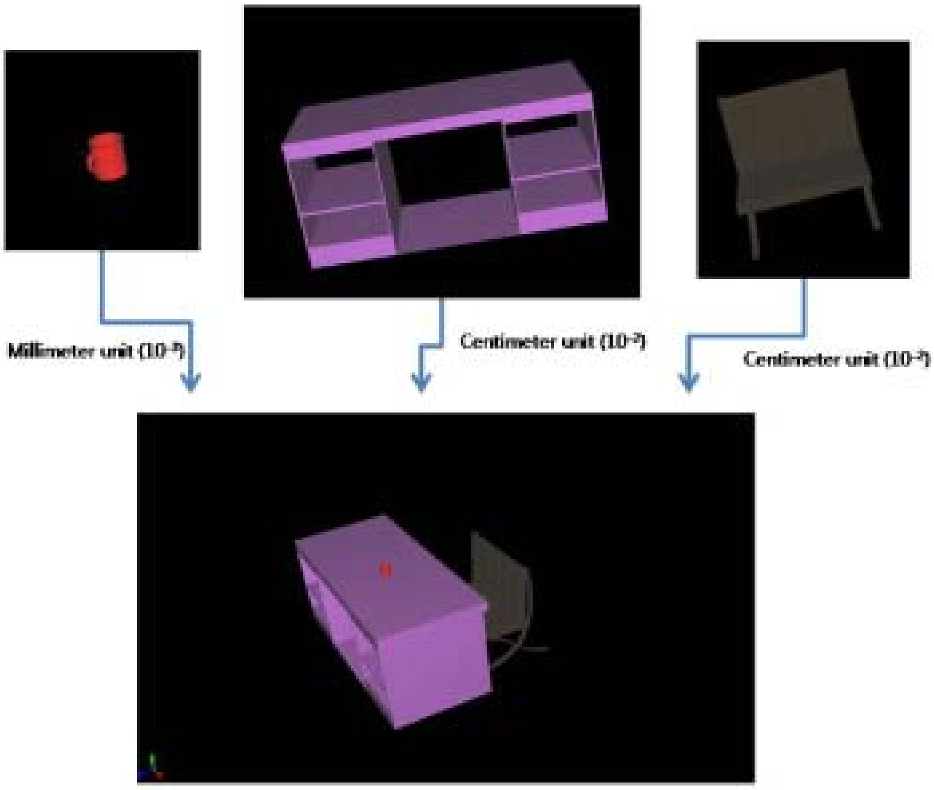


Figure 6 Combining three X3D objects in a scene with Physical Length Units

2.4 Consideration of duplicate unit definition

When two X3D files are read into an X3D browser sequentially to form a scene, the unit specified in the first X3D file is applied to the scene. In this case, an X3D object is not visible in the scene when the length or size difference is larger than the resolution set for the display. Usually, smaller objects are difficult to see when the length difference between two objects is over 10^3 with a display resolution of approximately $10^3 * 10^3$.

3. Abstract Type

X3DPhysicalNode : X3DNode {

SFNode [in,out] metadata NULL [X3DMetadataObject]

}

This abstract type node is the base type node for all physical nodes.

4. Node Reference

4.1 Physical Node

Physical : X3DPhysicalNode {

SFNode	[in,out]	Length	NULL [Length]
SFNode	[in,out]	Area	NULL [Area]
SFNode	[in,out]	Volume	NULL [Volume]
SFNode	[in,out]	Mass	NULL [Mass]
SFNode	[in,out]	Time	NULL [Time]
SFNode	[in,out]	Current	NULL [Current]
SFNode	[in,out]	Temperature	NULL [Temperature]
SFNode	[in,out]	Substance	NULL [Substance]
SFNode	[in,out]	Luminous	NULL [Luminous]
SFNode	[in,out]	SoundPressure	NULL [SoundPressure]
SFNode	[in,out]	metadata	NULL [X3DMetadataObject]

}

The Physical Node specifies the physical property of an object in an X3D scene. The field value can be NULL. If the value is not NULL, the field should include an appropriate node which defines the physical property. If the field value is Length, a Length Node should be defined. If the field value is NULL, the object is not transformed by any length unit, and, therefore, indicates that the object is not represented by any physical unit of measurement. As a result, an object is described based only on its coordinate values included in its X3D file.

The physical property units follow the SI (International System of Units) metric system (Table 1).

Table 1 Basic International System of Units (SI)

Name	**Symbol**	**Quantity**
metre	m	Length
kilogram	kg	Mass
second	s	Time
ampere	A	electric current
kelvin	K	thermodynamic temperature
mole	mol	amount of substance
candela	cd	luminous intensity

4.2 Length Node

Length : X3DPhysicalNode {

SFString [in,out] unit "UNI"
[“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”]

SFFloat [in,out] basis “1” [-∞ , ∞]

SFString [in,out] numeral “DEC”

[“SCIEN” | “DEC” | “ENGIN” | “ARCH” | “FRAC”]

SFNode [in,out] metadata NULL [X3DMetadataObject]

}

The Length Node specifies various real-world length units in an X3D file. The default length unit is meter. For example, a Physical Node can be defined as follows:

```
<physical>
        <length unit = "INCH"   basis = ”1”   numeral = ”DEC” />
</physical>
```

In this example, the unit of measurement changes from one meter to one inch. The unit specification provides the X3D world with precise and realistic modeling. The coordinates of the X3D world are transformed according to the unit specified.

4.2.1 Unit Field

The unit field specifies the unit for representing the length of an object. Without the unit field defined, the unit of measurement defaults to meter (UNI) (Table 2). Only the length units defined in the SI metric system have restrictions in representing virtual worlds precisely. Other units based on the SI measurement can be defined for denoting area or volume, for example. In addition, various units necessary for defining astronomical, medical, and microscopic objects can also be included (Table 3 and Table 4).

Table 2 Unit scale dictionary for powers of 10

Label	Symbol	Concept definition	Code
YOTTA	Y	10^{24}	1
ZETTA	Z	10^{21}	2
EXA	E	10^{18}	3
PETA	P	10^{15}	4
TERA	T	10^{12}	5
GIGA	G	10^{9}	6
MEGA	M	10^{6}	7
KILO	k	10^{3}	8
HECTO	h	10^{2}	9
DECA	da	10^{1}	10

UNI	1	10^{0}	11
DECI	d	10^{-1}	12
CENTI	c	10^{-2}	13
MILLI	m	10^{-3}	14
MICRO	µ	10^{-6}	15
NANO	n	10^{-9}	16
PICO	p	10^{-12}	17
FEMTO	f	10^{-15}	18
ATTO	a	10^{-18}	19
ZEPTO	z	10^{-21}	20
YOCTO	y	10^{-24}	21

vTable 3 Other units of measurement used with SI

Label	Symbol	Concept definition	Code
ANG	A	10^{-10}	22
NMILE	nm	1852 x m	23

AU	au	1.50 x 10^{11} m	24
LY	ly	9.46 × 10^{15} m	25
PASEC	pc	3.085678 × 10^{16} m	26
KPASEC	kpc	3.085678 × 10^{19} m	27

Table 4 Yard-pound system

Label	Symbol	Concept definition	Code
INCH	in	0.0254 X m	28
LINK	lk	0.201168 X m	29
FEET	ft	0.3048 X m	30
YARD	yd	0.9144 X m	31
ROD	rd	5.0292 X m	32
CHAIN	ch	20.1168 X m	33
FLONG	fl	201.168 X m	34
MILE	mi	1,609.344 X m	35
LEAGU	lg	4,828.032 X m	36

Units of measurement are different in different countries and parts of the world. The unit field allows the inclusion of custom units of measurement in the virtual environment (Table 4). Users can define whatever units are required in their applications (Table 5). The combination of the unit field and the basis field allows users to specify any necessary unit system in the X3D file.

Table 5 User defined units

Label	Symbol	Concept definition	Code
USER	user	1 Meter	37

4.2.2 Basis Field

The basis field specifies the scale of measurement for the user defined length unit, based on meter equal to 1. In other words, the basic user defined unit is the value calculated by the meter. For example, one inch is 25.4 millimeters ($25.4 * 10^{-3}$ meter) in the metric system, so the basis field would have a value of 0.001.

```
<Scene>

<Viewpoint position="0.0 0.0 1.0" description="1M view"/>

  <physical>

              <length unit = "USER"   basis = ”0.001”   numeral = ”DEC”/>

  </physical>
```

```
  <Transform scale="1 1 1" >

      <Shape>

          ...............

      </Shape>

    </Transform>

</Scene>
```

Figure 7 shows the result of specifying the same length by using different values in the length unit and basis fields. One model (left) has the length unit field defined as *millimeter* and the basis field as *1*, the other (right) has the length unit field defined as *USER* and the basis field as *0.001*. Both models are rendered the same size in the scene.

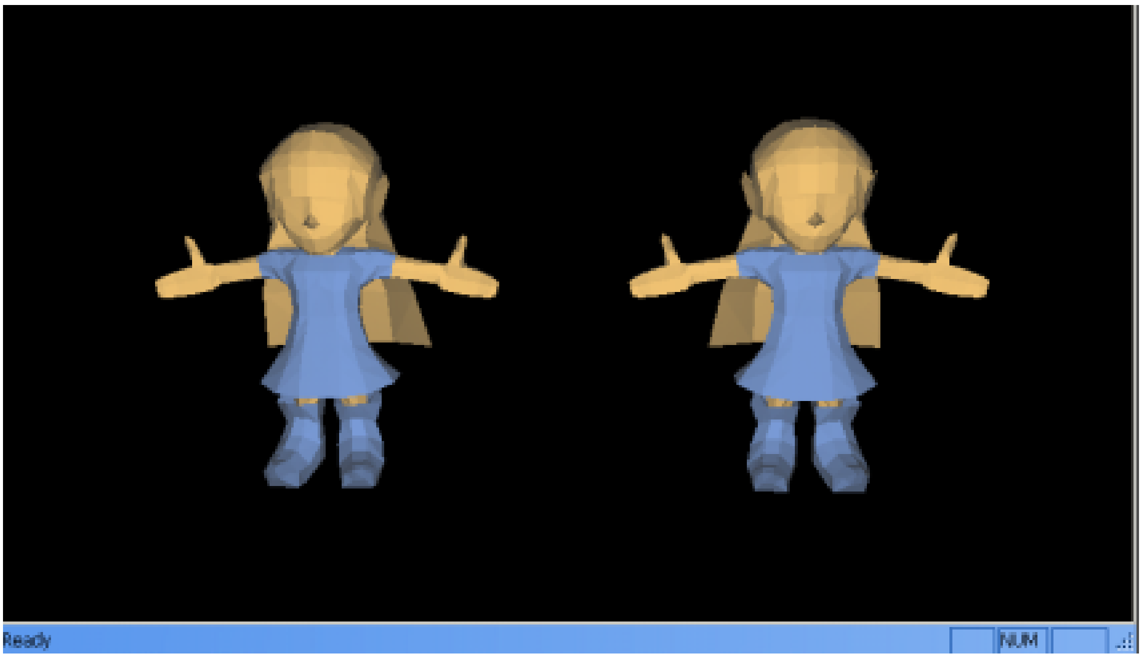


Figure 7 3D models with different unit specifications

4.2.3 Numeral Field

The numeral field represents the format for displaying the unit specification. It is intended to specify the unit precisely and accurately in the X3D world. There are five possible values for the numeral field, representing standard formats used in the areas of architecture, science, mathematics, and engineering (Table 5).

Table 6 Numeral representation in various application areas

Numeral type	**Numeral value**	**Examples**
Scientific	SCIEN	1.5000E+00, 2.0039E+00, 0.0000E+00
Decimal	DEC	1.5, 2, 0
Engineering	ENGIN	1.5”, 2”, 0”
Architectural	ARCH	$1\frac{1}{2}$”, 2”, 0”
Fractional	FRAC	$1\frac{1}{2}$, 2, 0

A graphics modeler such as 3ds Max, Maya, or AutoCAD includes the specification of physical length for a specific side of an object. In addition, there are cases that require various mathematical notations. However, current X3D browsers cannot accurately represent these objects in the scene because X3D does not include such notations. The numeric field allows the X3D user to include the mathematical notations in the scene.

```
<Scene>

  <Viewpoint position="0.0 0.0 1.0" description="1M view"/>

  <physical>

            <length unit ="UNI" basis="1" numeric="FRAC"/>

  </physical>

    <Transform scale=" 1/3  1/3  1/3   " >

      <Shape>

          [………….. ]

      </Shape>

    </Transform>

 <Transform scale=" 1    1    1 " >

      <Shape>

          […..………..]

      </Shape>

    </Transform>

 </Scene>
```

Above is an example of defining fractional notation in the numeric field. It describes the number 1/3 directly as a fraction instead of as 0.333333… in an original X3D description. This description allows the transfer of accurate object scale representation directly in a file without round off error, especially important in areas of application such as architecture or medicine.

4.3 Area Node

```
Area : X3DPhysicalNode {
   SFString  [in,out]        unit          "UNI2"
                             ["CENTI2"|"UNI2"|"FT2"|" YD2"|" HA"|"ACRE"|]
   SFFloat   [in,out]   value        "1"      [-∞ , ∞]
   SFNode    [in,out]   metadata     NULL     [X3DMetadataObject]
}
```

The Area node defines the unit for the area composing an object. This node can be applied to a Shape node below a Transform node. It can include the numeric data value of the area for an object. The area unit is dependent on the length unit and can be applied to two-dimensional objects. The default unit for the Area node is square meter (m^2).

4.3.1 Unit Field

The Unit field specifies the unit for an area. If this field is not defined, the default unit is square meter (m^2).

Table 7 Area units

Label	Symbol	Concept definition	Code

CENTI2	cm^2	$0.01m^2$	1
UNI2	m^2	$1\ m^2$	2
FT2	$foot^2$	$0.092904m^2$	3
YD2	yd^2	$0.836135m^2$	4
HA	ha	$10000m^2$	5
ACRE	acre	$4046.927283m^2$	6

4.4 Volume Node

```
Volume : X3DPhysicalNode {
  SFString   [in,out]     unit         "UNI3"
                                       ["CENTI3"|"UNI3"|"FT3"|" YD3"|"DL"|"ML"|"L"]
  SFFloat    [in,out]     value        "1"        [-∞ , ∞]
  SFNode     [in,out]     metadata     NULL       [X3DMetadataObject]
}
```

The Volume node specifies the unit for an object’s volume. The node can be applied to a Shape node below a Transform node. It can include the numeric data value of an object’s volume. The unit of a volume is dependent on the length unit and can be applied to a three-dimensional object.

4.4.1 Unit Field

The Unit field specifies the unit of a volume. If this field is not defined, the default unit is cubic meter (m^3).

Table 8 Volume units

Label	**Symbol**	**Concept definition**	**Code**
CENTI3	Cm^3	$0.000001m^3$	1
UNI3	M^3	$1\ m^3$	2
FT3	$Foot^3$	$0.028321m^3$	3
YD3	$Yard^3$	$0.769231m^3$	4
DL	dℓ	$0.0001m^3$	5
ML	mlℓ	$0.000001m^3$	6
L	ℓ	$0.001m^3$	7

4.5 Mass Node

```
Mass : X3DPhysicalNode {
  SFString   [in,out]      unit                    "KG"
                                                   ["MG"|"G"|"KG"|"LB"|"TON"|"GRAIN"|"OZ"]
```

SFFloat [in,out] value "1" [-∞ , ∞]

SFNode [in,out] metadata NULL [X3DMetadataObject]

}

The Mass node specifies the unit for an object's mass. It can be applied to a Shape node below a Transform node and can include the numeric data value for representing an object's mass. The mass unit follows the specification of the International System of Units.

4.5.1 Unit Field

Table 9 Mass units

Label	**Symbol**	**Concept definition**	**Code**
MG	Mg	0.000001kg	1
G	G	0.001kg	2
KG	Kg	1kg	3
LB	Lb	0.453599kg	4
TON	Ton	1000kg	5
GRAIN	Grain	0.000065kg	6
OZ	Oz	0.02835kg	7

4.6 Time Node

Time : X3DPhysicalNode {

```
  SFString   [in,out]   unit       "S"    ["S"|"MIN"|"S"]
  SFFloat    [in,out]   value      "1"    [-∞ , ∞]
  SFNode     [in,out]   metadata   NULL   [X3DMetadataObject]
```

}

The Time node specifies the unit for time. It can be applied to a Shape node below a Transform node, and can include the numeric data value for an object's time-related data. The time unit follows the specification of the International System of Units. The default time unit is second (sec).

4.7 Current Node

Current : X3DPhysicalNode {

```
  SFString   [in,out]   unit       "A"    ["MA"|"A"]
  SFFloat    [in,out]   value      "1"    [-∞ , ∞]
  SFNode     [in,out]   metadata   NULL   [X3DMetadataObject]
```

}

The Current node specifies the electric current unit for an object. It can be applied to a Shape node below a Transform node, and can include the numeric data value for an object's current. The current unit follows the specification of the International System of Units. Its default unit is ampere (A).

4.8 Temperature Node

Temperature : X3DPhysicalNode {

SFString [in,out] unit "K" ["OC"|"0F"|"K"|"0E"]

SFFloat [in,out] value "1" [-∞ , ∞]

SFNode [in,out] metadata NULL [X3DMetadataObject]

}

The Temperature node specifies the unit for an object's or an environment's temperature. It can be applied to a Shape node below a Transform node. It can include the numeric data value for representing an object's or environment's temperature. The temperature unit follows the specification of the International System of Units.

4.8.1 Unit Field

The Unit field specifies the temperature unit for an object. If the field is not defined, the default unit of temperature is Kelvin (K).

Table 10 Temperature unit

Label	**Symbol**	**Concept definition**	**Code**
0C	℃	274.15K	1

0F	°F	255.927778K	2
K	K	1K	3
0R	°R	0.538889K	4
TON	Ton	1000kg	5
GRAIN	Grain	0.000065kg	6
OZ	Oz	0.02835kg	7

4.9 Substance Node

```
Substance : X3DPhysicalNode {
  SFString  [in,out]   unit       "MOL"   ["MOL"]
  SFFloat   [in,out]   value      "1"     [-∞ , ∞]
  SFNode    [in,out]   metadata   NULL    [X3DMetadataObject]
}
```

The Substance node specifies the unit for the substance of an object. It can be applied to a Shape node below a Transform node. It can include the numeric data value for an object's substance. The substance unit follows the specification of the International System of Units. The default unit is mol.

4.10 Luminous Node

```
Luminous : X3DPhysicalNode {
    SFString   [in,out]   unit       "CD"   ["CD"|"LM"|"LUX"]
    SFFloat    [in,out]   value      "1"    [-∞ , ∞]
    SFNode     [in,out]   metadata   NULL   [X3DMetadataObject]
}
```

The Luminous node specifies the luminous intensity for an object. It can be applied to a Shape node below the Transform node. It can include the numeric data value for the object's luminous intensity. The luminous unit follows the specification of the International System of Units. The default unit is candela (cd).

4.11 SoundPressure Node

```
SoundPressure : X3DPhysicalNode{
    SFString   [in,out]   unit       "DB"   ["FB"|"PA"]
    SFFloat    [in,out]   value      "1"    [-∞ , ∞]
    SFNode     [in,out]   metadata   NULL   [X3DMetadataObject]
}
```

The SoundPressure node specifies the unit for an object's sound pressure. It can be applied to a Shape node below the Transform node. It can include the numeric data value for the object's sound pressure. The sound pressure unit follows the specification of the International System of Units. The default unit is Decibel (dB).

Annex

1. Unit specification for the solar system using astronomical units (AU)

When we represent the solar system, we must consider the physical scale using the original X3D (Figure A.1). We must know the scaling factor applied to the coordinate values of the astronomical models in order to retrieve exact size or length because the current X3D has only one default unit, meter. Otherwise, the X3D would have to represent the coordinate values using very large numbers such as 149.60 x 10^9 meters, which corresponds to one AU (Astronomical Unit).

In Figure A.1, if the coordinates in the X3D file do not have the exact numeric value specified according to the meter unit, we cannot know the exact real distance or size of the solar system. Although the X3D file can have the meter-scaled coordinates defined, the data would consist of very large numbers. Otherwise, we should calculate the scales defined in all the Transform nodes using the coordinates in the X3D file, although the objects were generated according to exact scales in a modeler. This is difficult for browswer users because browsers usually do not provide such real length calculation since the X3D does not define the real units.

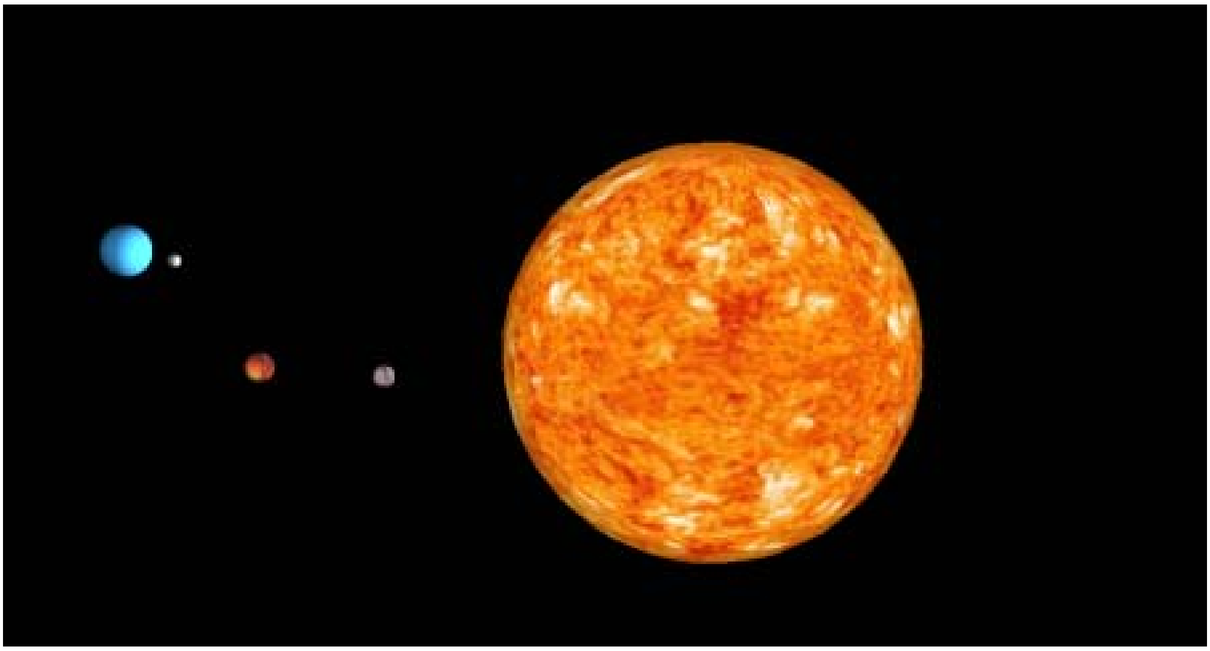


Figure A.1 Representation of the solar system *without* Length Unit (Octaga Player)

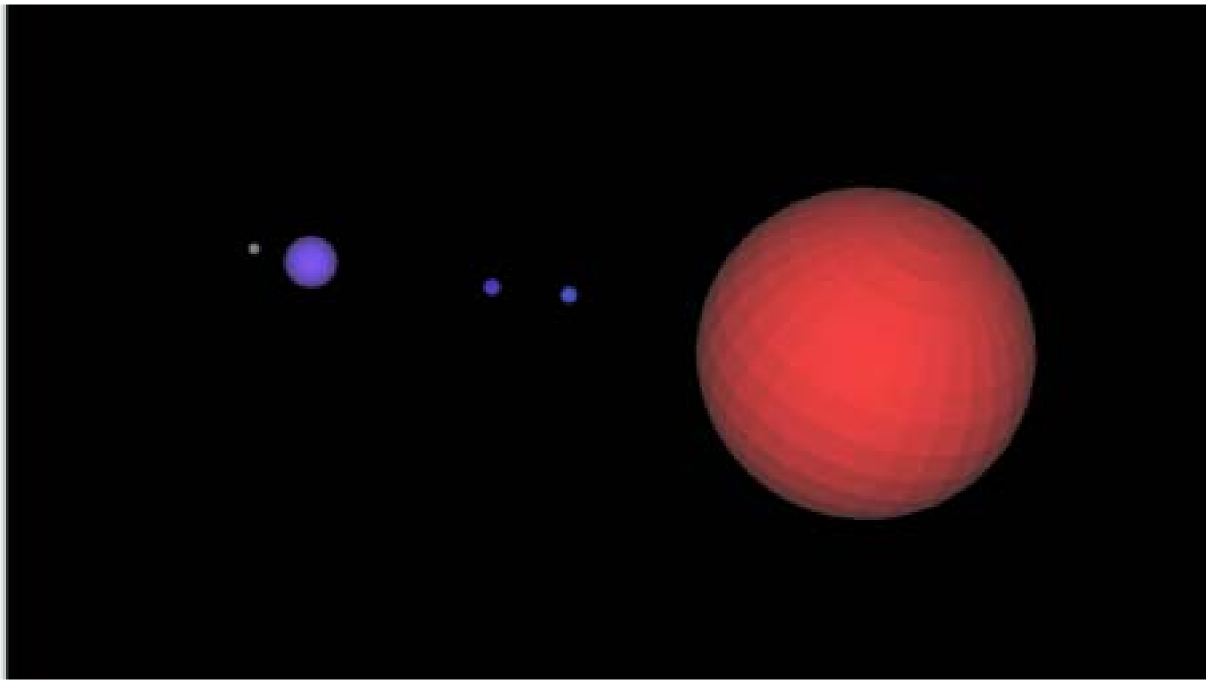


Figure A.2 Representation of the solar system *with* Length Unit, AU

Figure A.2 shows the result of using the astronomical unit (AU) in the X3D scene. Users can represent the real sizes of planets or satellites using the astronomical units. In the figure, the scene is represented by combining the sun and all the planets by Inline nodes after each of them has been generated with its respective length unit. In this case, it is easy for browser users because they do not need to calculate scaled values for the coordinates according to the meter unit. The astronomical unit is defined as in Figure A.3.

```
<Physical>

    <Length    unit = “AU”    basis = “1”    numeral = “DEC”>

</Physical>
```

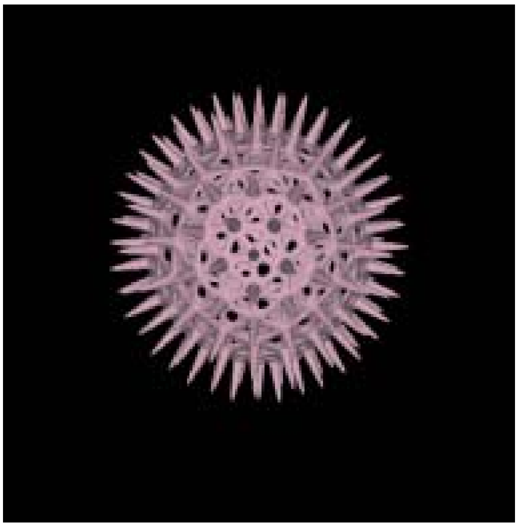
Figure A.3 An astronomical unit definition

2. Unit specification for a microorganism

In microbiology, an microorganism is defined as an organism smaller than 0.1 mm and which cannot be seen with the naked eye. Usually, it is defined by micrometer unit. If the unit for the microorganism is not specified, it is time-consuming to know the size of the microorganism from its X3D file. In addition, it is difficult to differentiate size between two microorganisms if they were generated by different modelers without considering the units.

Figure A.4 represents a microorganism defined by micrometer unit, and Figure A.5 represents a microorganism defined by millimeter. They were generated by 3ds Max. In separate views, they look similar in size. When they are read into a common scene, however, they should be scaled. Otherwise, without specifying their units, it is difficult to know their size differences. However, it is very simple to determine their relative sizes, as in Figure A.6 ,when we define their units as in Figure A.7 and Figure A.8.

(Figure A.4) 10 Micrometer unit



(Figure A.5) 0.1 Millimeter unit

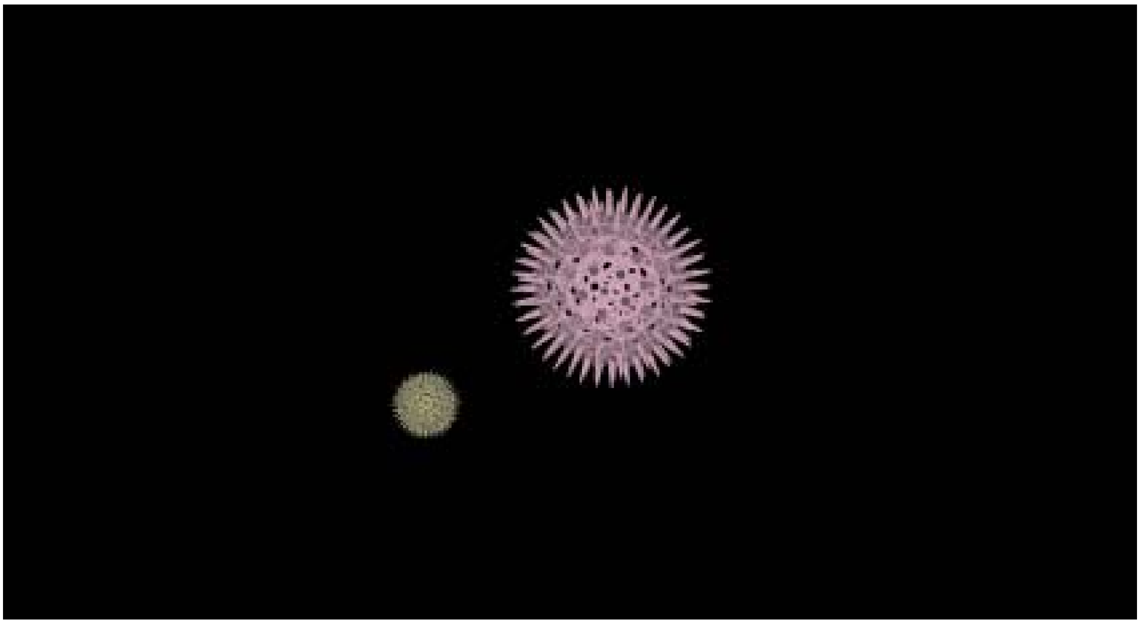


Figure A.6 Two microorganisms specified in different units

```
<Physical>

    <Length    unit = “MICRO”    basis = “1”    numeral = “DEC”>

    <Transform scale = “10 10 10” >

</Physical>
```

Figure A.6 Unit specification for a microorganism in micrometer

```
<Physical>

    <Length    unit = “MILLI”    basis = “1”    numeral = “DEC”>

    <Transform scale = “0.1 0.1 0.1” >

</Physical>
```

Figure A.7 Unit specification for a microorganism in millimeter

3. Unit specification for architecture design

The physical length unit is necessary to represent an architecture design accurately. Problems may occur with the X3D without unit specification when the design drawing has units and measurement values originally defined in CAD. Figure A.8 shows CAD examples which have real length values. Figure A.9 shows their X3D representations after specifying their length units as in Figure A.10.

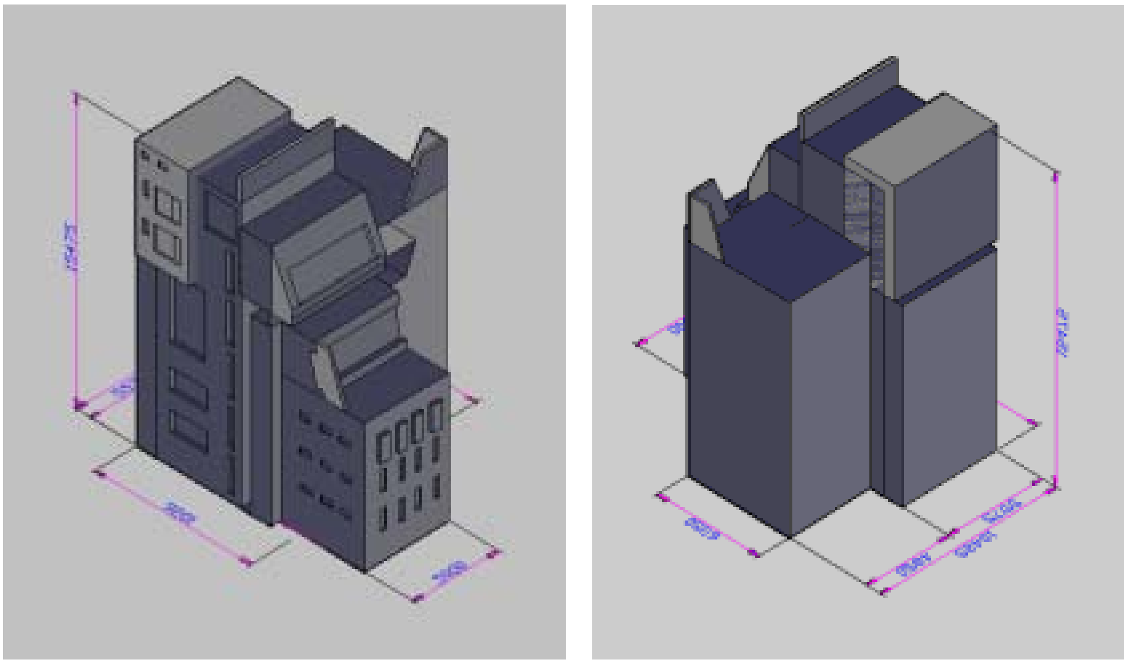


Figure A.8 CAD examples

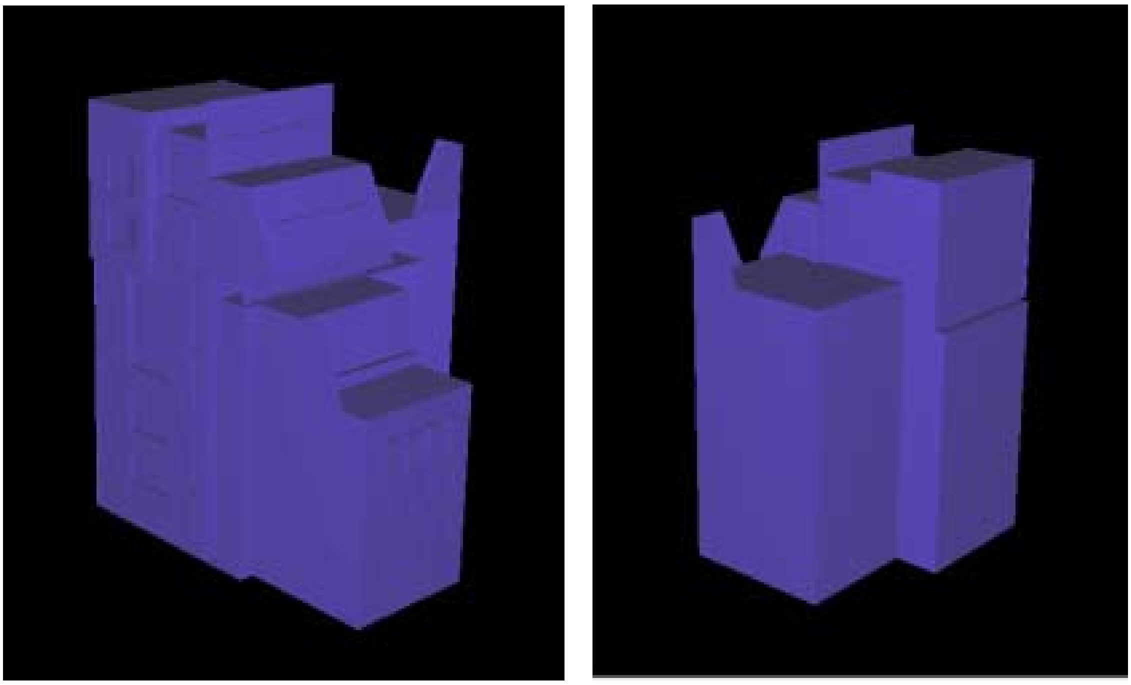


Figure A.9 X3D representations of CAD Examples

<Physical>

<length unit = “MILLI” basis = “1” numeral = “DEC”>

</Physical>

Figure A.10 Unit specification for the CAD examples

4. Anthropometric unit specification

Recently, many countries have constructed anthropometric databases including specific lengths of human bodies. Examples of such public enterprises are Size Korea, Size USA, and Size Japan. In the anthropometric areas, unit specification is very important in order to show precise distance between two feature points on a human body. Figure A.11 shows an example of standardized facial measurement. Recently, 3D scanners are being used for measuring human bodies, with the millimeter unit usually used for 3D scanners. Without unit specification, we cannot be sure how the coordinates of the scanners should be scaled for matching the real size of the face. With the unit specification, we can obtain an accurate measurement for a face (Figure A.12). The units for the human face are defined as in Figure A.13. The X3D unit specification can bring about anthropometry with 3D scanners as an important Web3D application area.

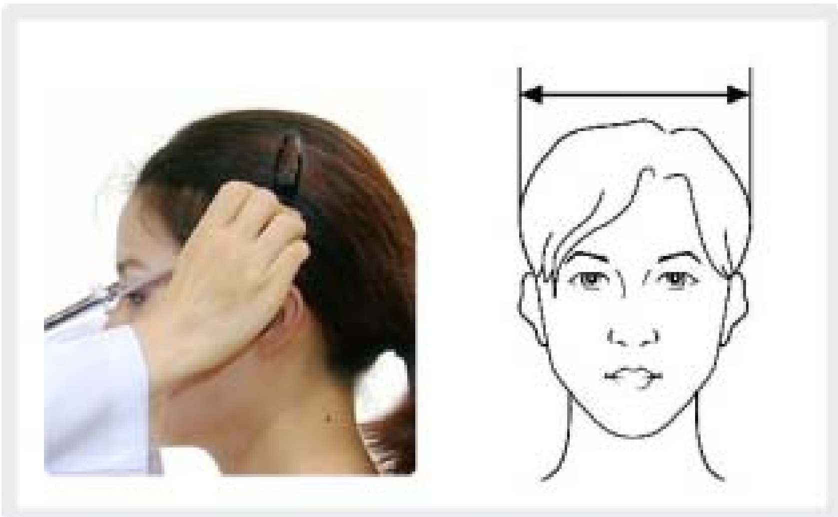


Figure A.11 Facial measurement

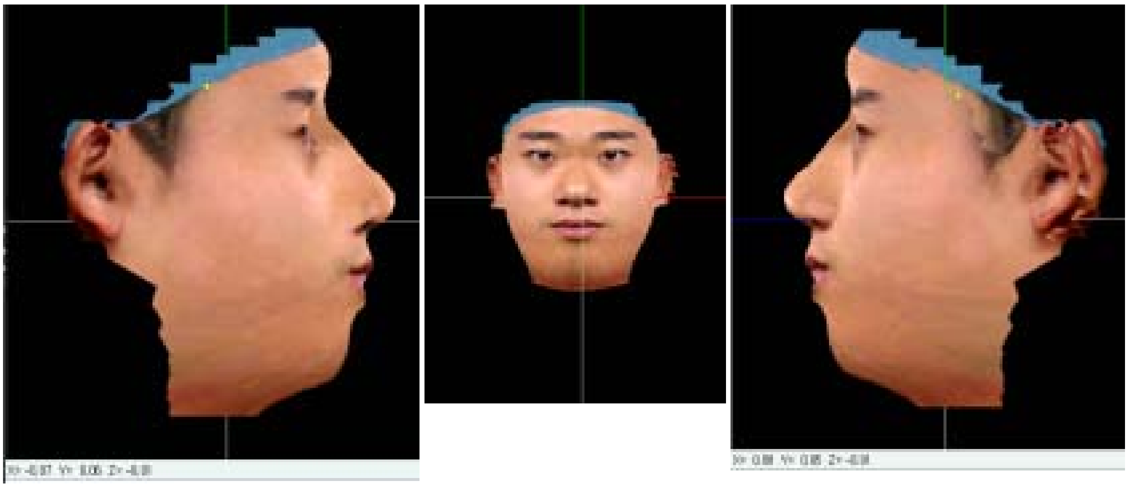


Figure A.12 A facial model with unit specification millimeter

```
<Scene>
    <Physical>
            <length unit = "MILLI"   basis = ”1”   numeral= ”DEC” />
     <Transform >
         <Shape>
             [……...] // Facial data
         </Shape>
       </Transform>
    </physical>
 </Scene>
```

Figure A.13 Unit definition for a human face