X3D Physical Sensors (updates)

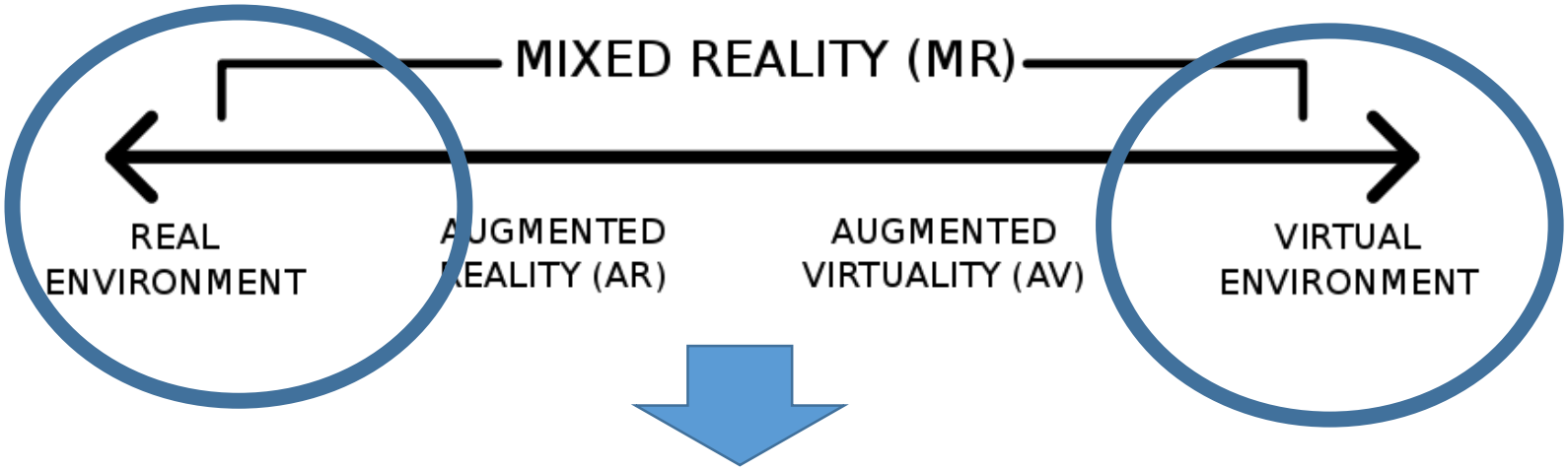
Web3D Korea Standardization Meeting at SIGGRAPH

Los Angeles, CA, USA

July 31, 2017

Myeong Won Lee (The University of Suwon) and Kwan-Hee Yoo (Chungbuk National University)

Physical Sensor Representation in X3D



Real world simulation of sensor functions in X3D scenes

Before and After X3D

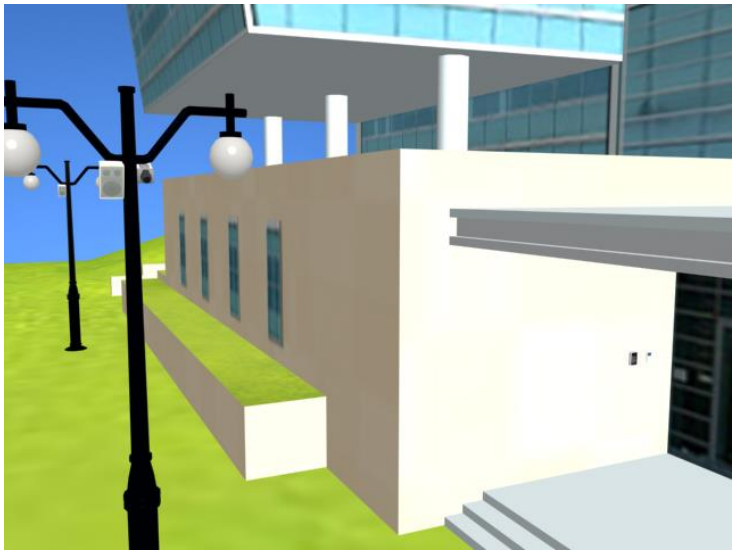
- Before
  - 3D representation of real object appearance
  - Modeling, rendering, and animation for 3D appearance in VR
  - Focus on appearance as it is in the real world
- After
  - 3D simulation and representation of sensor device functions
  - Modeling, rendering, animation, and simulation of real objects in X3D
  - Focus on real world simulation and functional representation in X3D
  - IoT device and sensor representation in X3D

X3D Physical Sensor and SC24 WG9 NP

- ISO/IEC JTC 1/SC 24/WG 9 NWIP
  - Sensor representation in MAR
  - A reference model for physical sensor representation in 3D scenes
  - All sensor types are covered with abstract models and interfaces
  - XML definition of physical sensors

- X3D Physical sensor
  - Physical sensor representation in X3D scenes
  - New proposal for ISO/IEC JTC 1/SC 24/WG6 NWIP
  - Selective sensor types are covered with existing X3D abstract data model and interface
  - X3D based definition of physical sensors requires X3D schema expansion
  - Implementation of X3D physical sensor viewers and simulators

Physical Sensor Representation in X3D



Camera sensor

Chemical sensor

Electric sensor

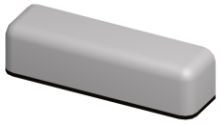
Environment sensor

Flow sensor

Light sensor

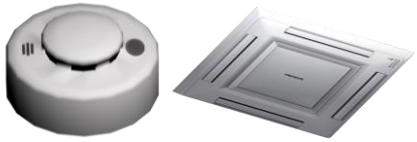
Navigation sensor

Pressure sensor



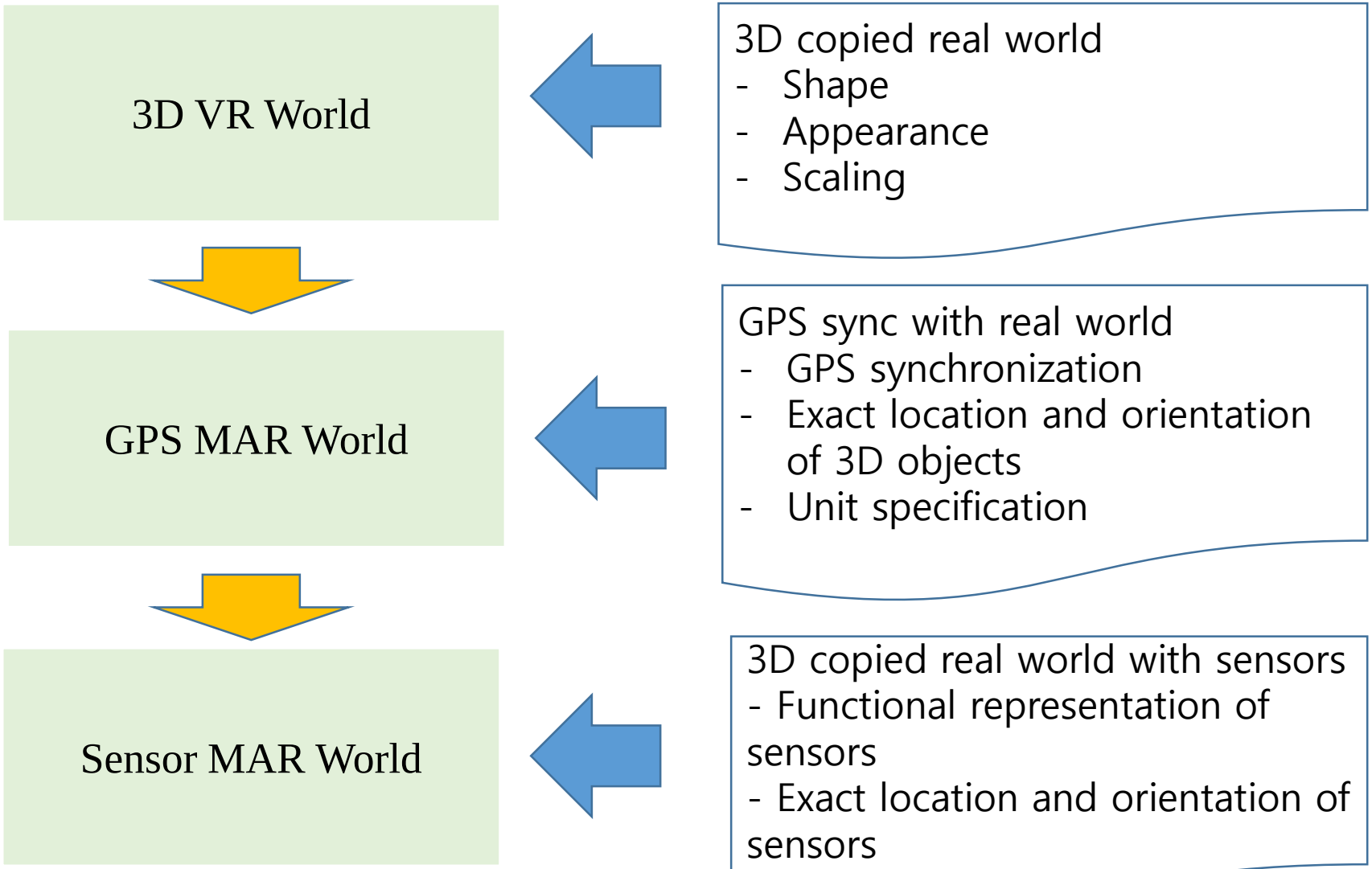
Proximity sensor

Sound sensor



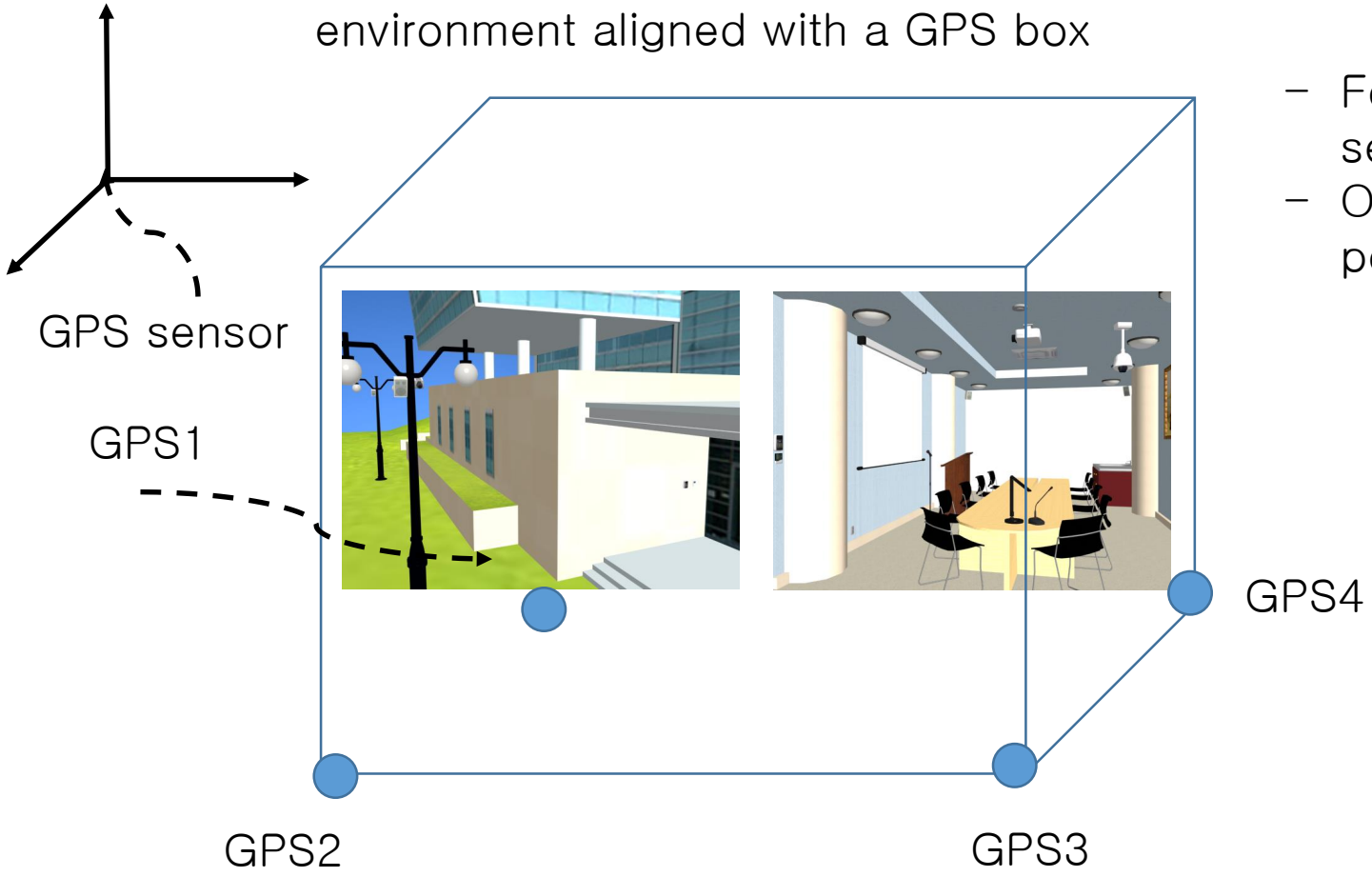
Temperature sensor

Sensor MAR World



Exact Location and Orientation (1)

1. Define the Geo coordinate system of a 3D virtual environment aligned with a GPS box

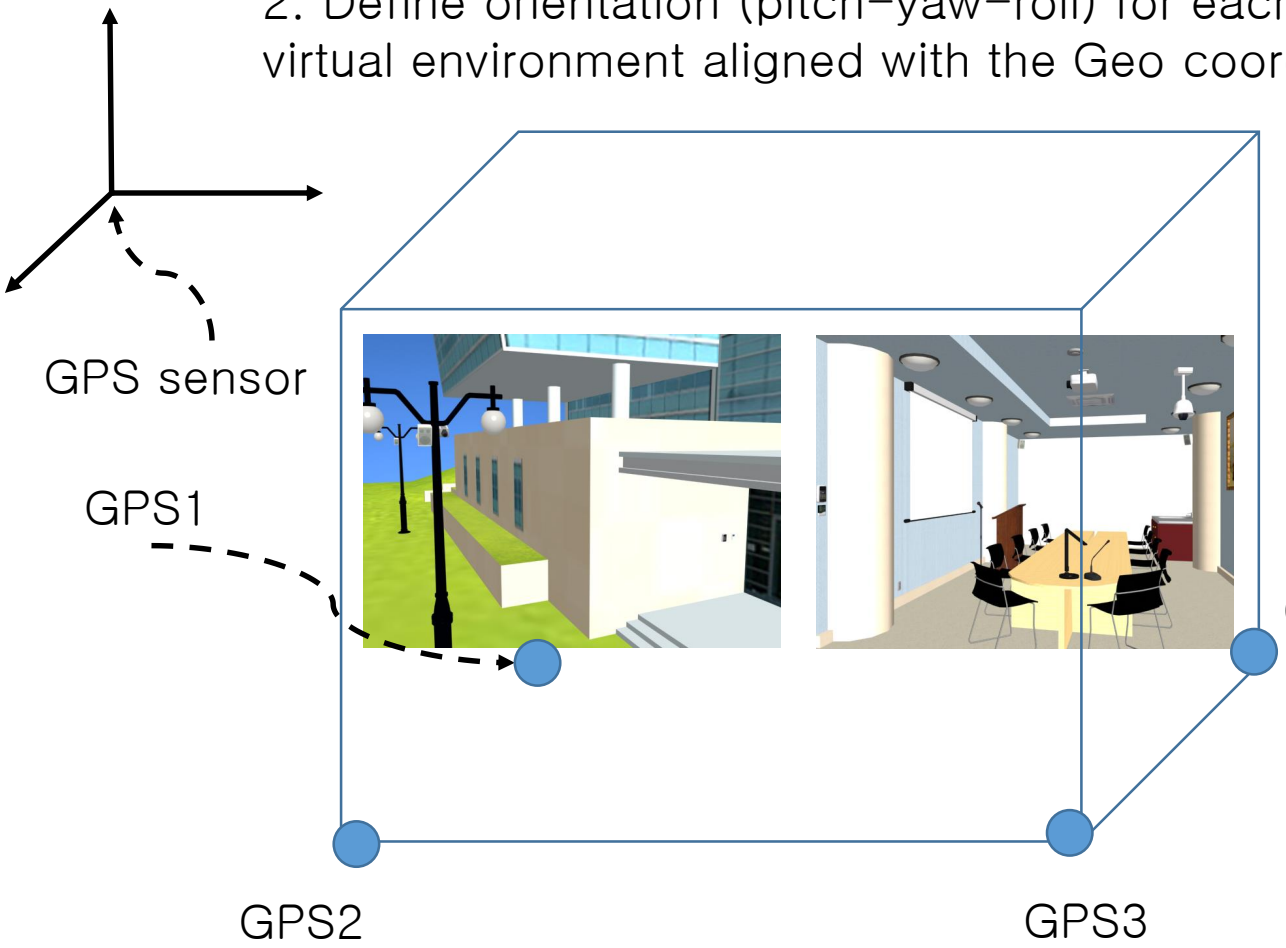


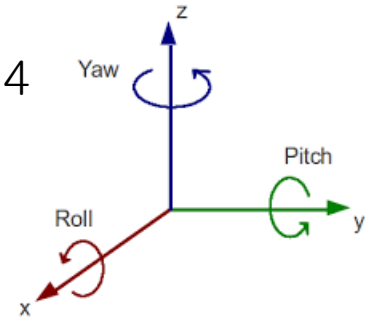
- Four GPS sensor values
- One Geo origin per 3D scene

Exact Location and Orientation (2)

2. Define orientation (pitch-yaw-roll) for each sensor in a 3D virtual environment aligned with the Geo coordinate system.

- Four GPS sensor values
- One Geo origin per 3D scene
- Orientation (pitch-yaw-roll) for each sensor

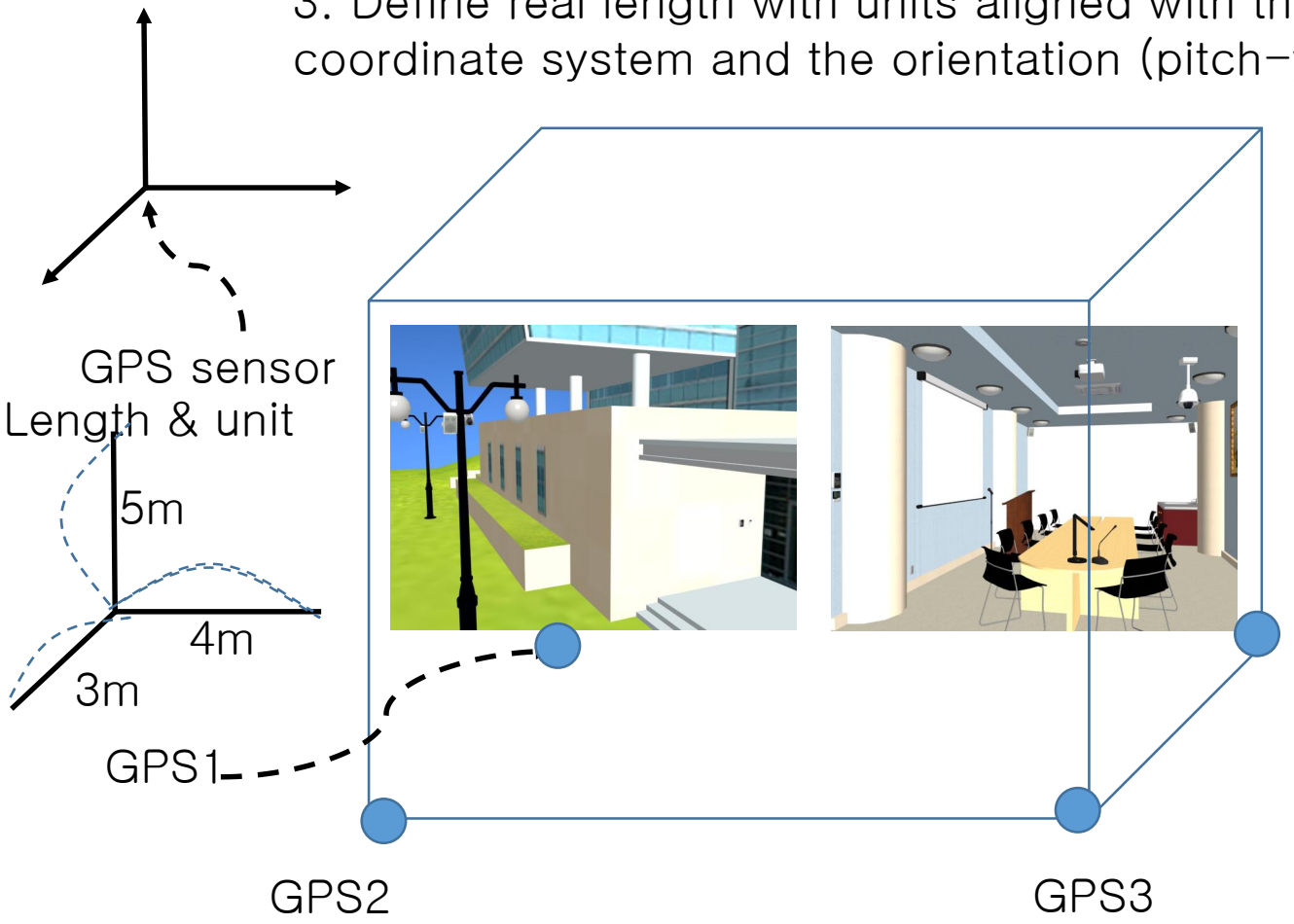


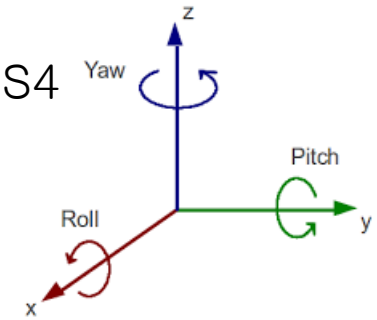


Exact Location and Orientation (3)

3. Define real length with units aligned with the local coordinate system and the orientation (pitch-yaw-roll) frame.

- Four GPS sensor values
- One Geo origin per 3D scene
- Orientation (pitch-yaw-roll) per sensor
- Length units

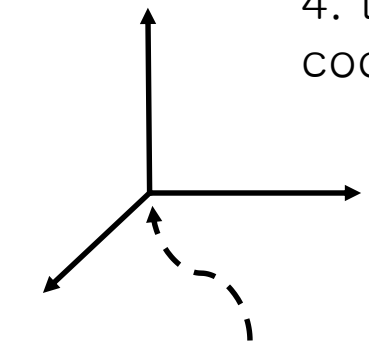


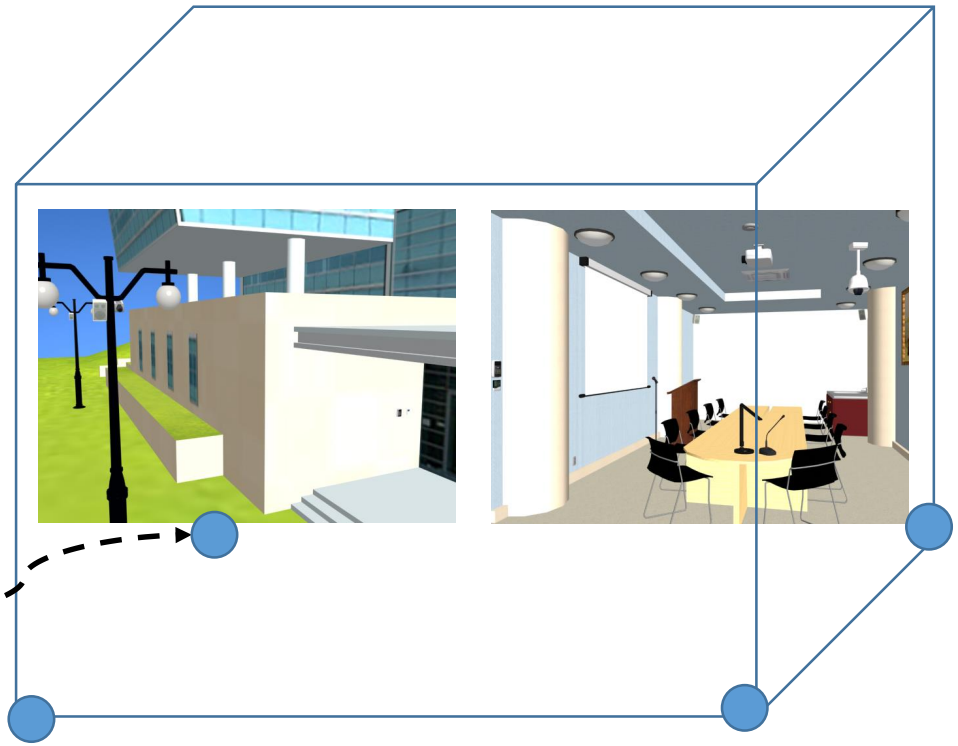


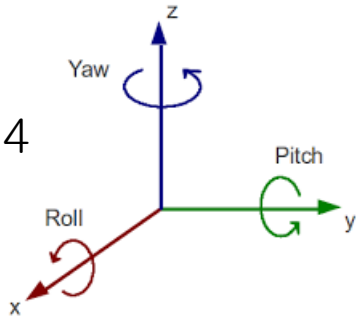
Exact Location and Orientation (4)

4. Define enhanced graphics pipeline including geo coordinates, orientation, and unit coordinate system

- Four GPS sensor values
- One Geo origin per 3D scene
- One relative compass frame
- Length units







Local coordinate system → Global coordinate system → Geo based coordinate system → Pitch-yaw-roll and unit coordinate system

Objectives of X3D Physical Sensors

- Define X3D objects with physical sensors using physical properties
- Representation of physical properties of each physical sensor
- Provide physical sensor interfaces for event processing to/from each physical sensor in X3D

Physical Sensors

- Camera Sensor
- Chemical Sensor
- Electric Sensor
- Environment Sensor
- Flow Sensor
- Force Sensor
- Light Sensor
- Movement Sensor
- Navigation Sensor
- Particle Sensor
- Position Sensor
- Pressure Sensor
- Proximity Sensor
- Sound Sensor
- Temperature Sensor
- Other sensors

Sensor representation and simulation with physical functions in an X3D scene

GOAL

Physical Sensors in X3D

- Camera Sensor
- Chemical Sensor
- Electric Sensor
- Environment Sensor
- Flow Sensor
- Force Sensor
- Light Sensor
- Movement Sensor
- Navigation Sensor
- Particle Sensor
- Position Sensor
- Pressure Sensor
- Proximity Sensor
- Sound Sensor
- Temperature Sensor
- Other Sensors

Sensor representation and simulation with physical functions in an X3D scene

GOAL

Camera Sensor

- Requires a 3D object to represent the camera sensor in an X3D scene and a camera functions user interface to control and change the 3D scene
- Simulation procedure
  - Process the state of the camera with On/Off
  - Represent and process a camera event
  - Represent and process physical functions of the camera, such as play, pause, zoom, etc.
- Examples
  - CCTV, phone camera, standard camera

Environment Sensor

- Requires a 3D object to represent the environment sensor in an X3D scene and a functions user interface to display and control the sensor and to change the scene
- Simulation procedure
  - Process the state of the environment sensor device with On/Off
  - Acquire and represent changing weather information from the device, including temperature and humidity
  - Represent and process physical functions of the device
- Examples
  - Thermometer, hygrometer

Light Sensor

- Requires a 3D object to represent the light sensor in an X3D scene and a light functions user interface to control and change the 3D scene
- Simulation procedure
  - Represent the state of the physical light sensor
  - Represent light information, such as physical intensity, location, orientation, material, etc.
  - Provide interfaces for event processing to/from a physical light
- Examples
  - Fluorescent light, streetlight

Navigation Sensor

- Requires a 3D object to represent the navigation sensor in X3D and a functions user interface to control and change the 3D scene
- Simulation procedure
  - Process the state of the navigation sensor with On/Off
  - Acquire navigation information events such as GPS
  - Represent and control navigation information from the sensor
- Examples
  - Magnetic compass, GPS

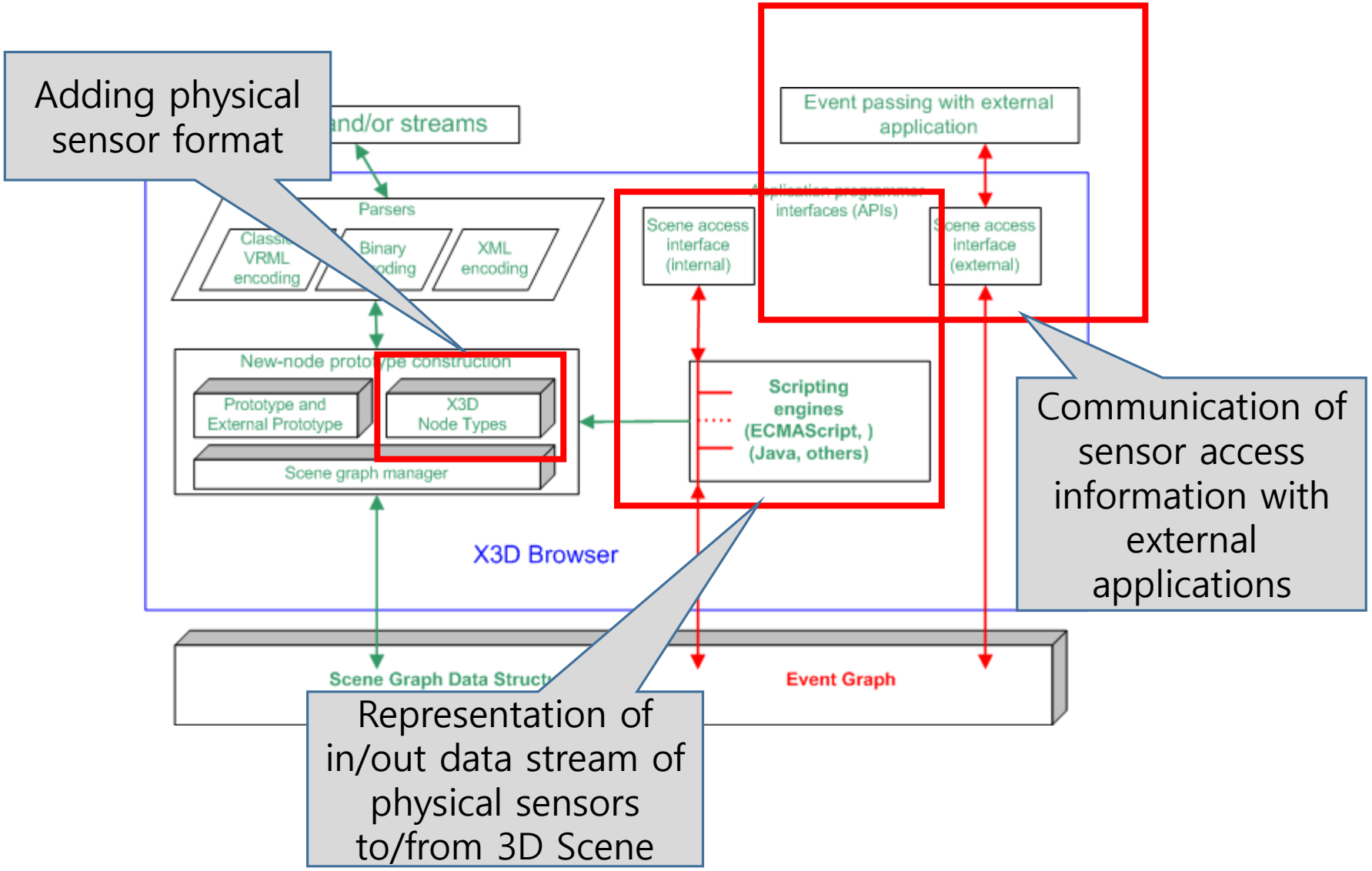
Position Sensor

- Requires a 3D object to represent the position sensor in an X3D scene and a functions user interface to control and change the 3D scene
- Simulation procedure
  - Process the state of the position sensor with On/Off
  - Acquire location information events
  - Represent and control location information from the position sensor
- Examples
  - Range finder, telemeter, angle finder, measuring instrument

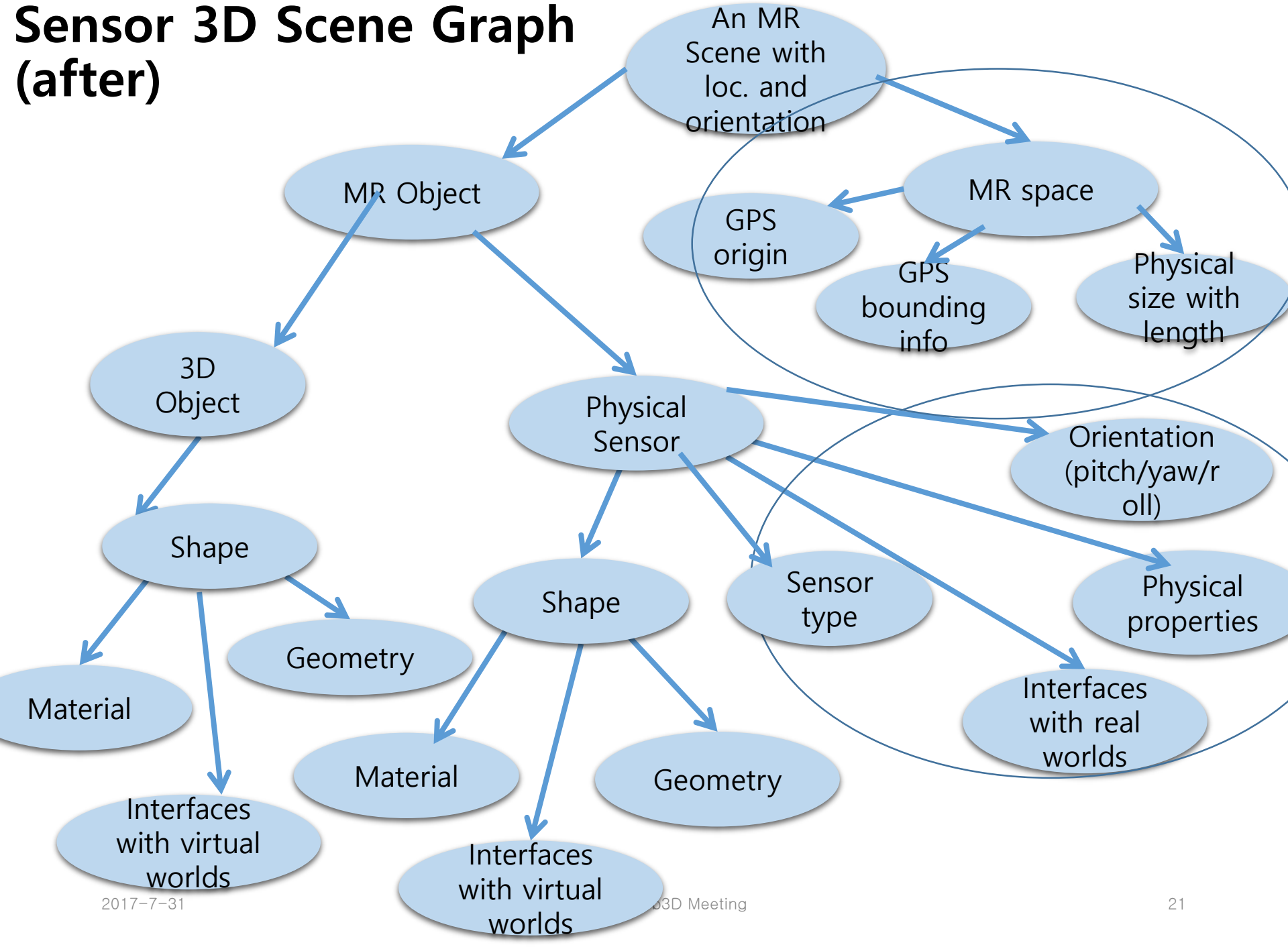
Sound Sensor

- Requires a 3D object to represent the sound sensor in an X3D scene and a sound functions user interface to control and change the 3D scene
- Simulation procedure
  - Process the state of the sound with On/Off
  - Represent and control sound device events
  - Represent sound information through the sound device
- Examples
  - Speaker, microphone

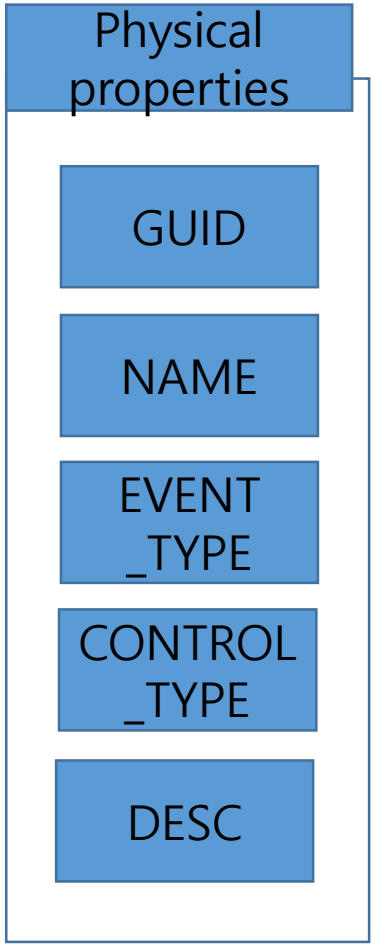
X3D Physical Sensor Architecture



Sensor 3D Scene Graph (after)



Physical Properties of a Physical Sensor (device)



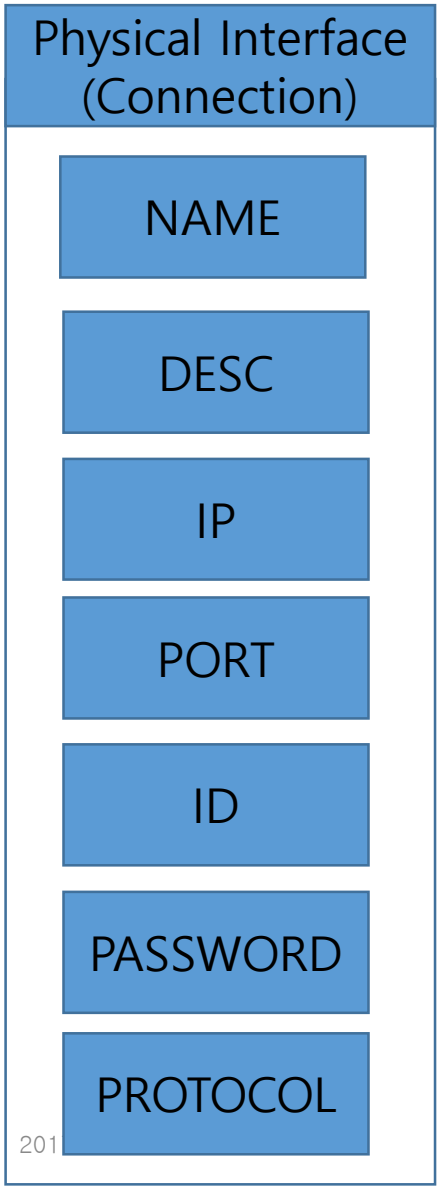
Example

```
<GUID>111-111-111</GUID>
<NAME>Airconditioner</NAME>
<EVENT_TYPE>TEMP</EVENT_TYPE>
<CONTROL_TYPE>TEMP</CONTROL_TYPE>
<DESC>sensor type</DESC>
```

Physical Sensor Device Properties

Device Properties Fields	Description
GUID	Unique ID for recognizing a device (Globally Unique Identifier, an implementation of Universally Unique identifier (UUID))
NAME	Device name
EVENT_TYPE	Available data type that can access a physical sensor device
CONTROL_TYPE	Available data type that can send to a physical sensor device
Description	Additional description of a physical sensor device

Physical Interface of a Physical Senor (Connection Information)

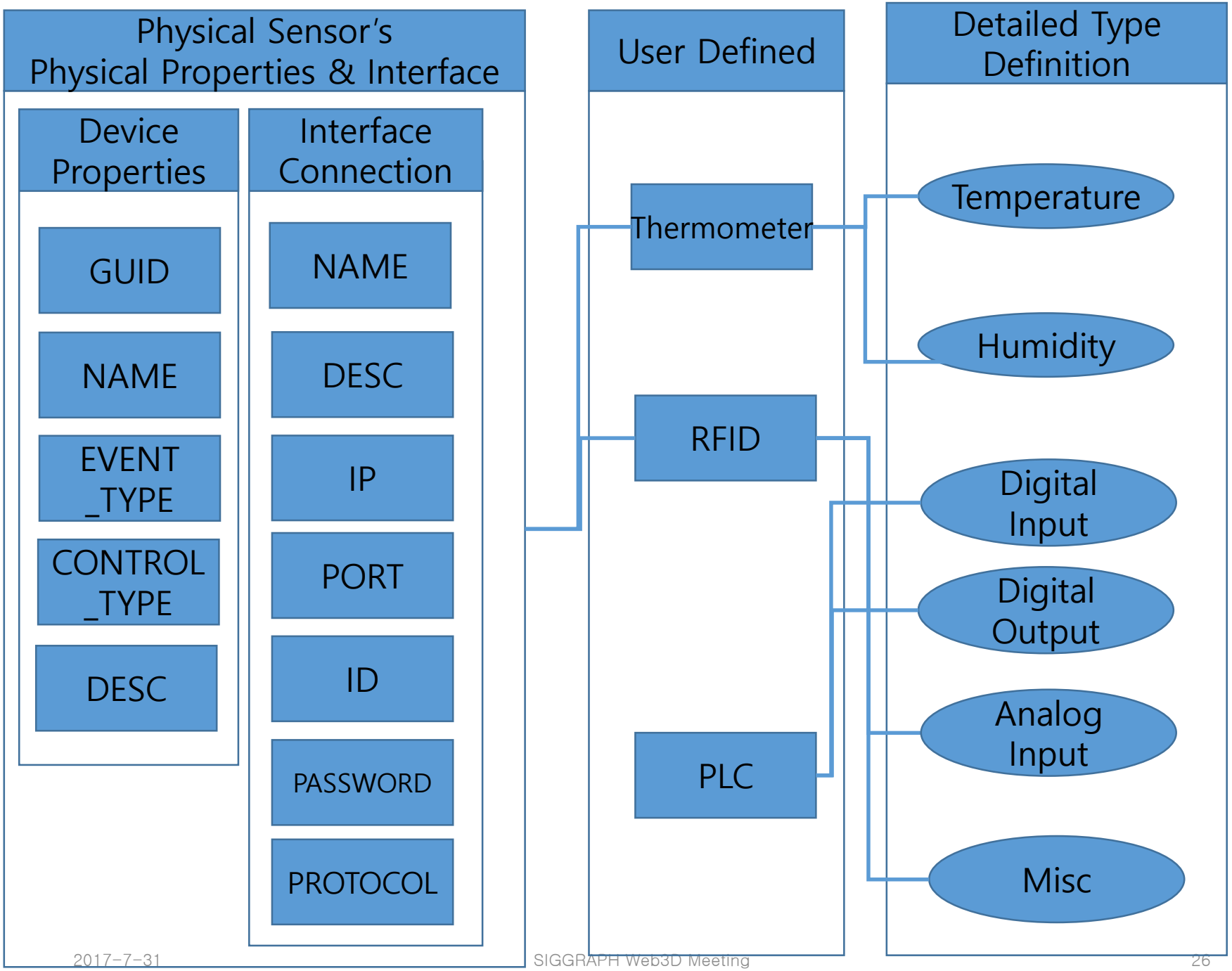


Example

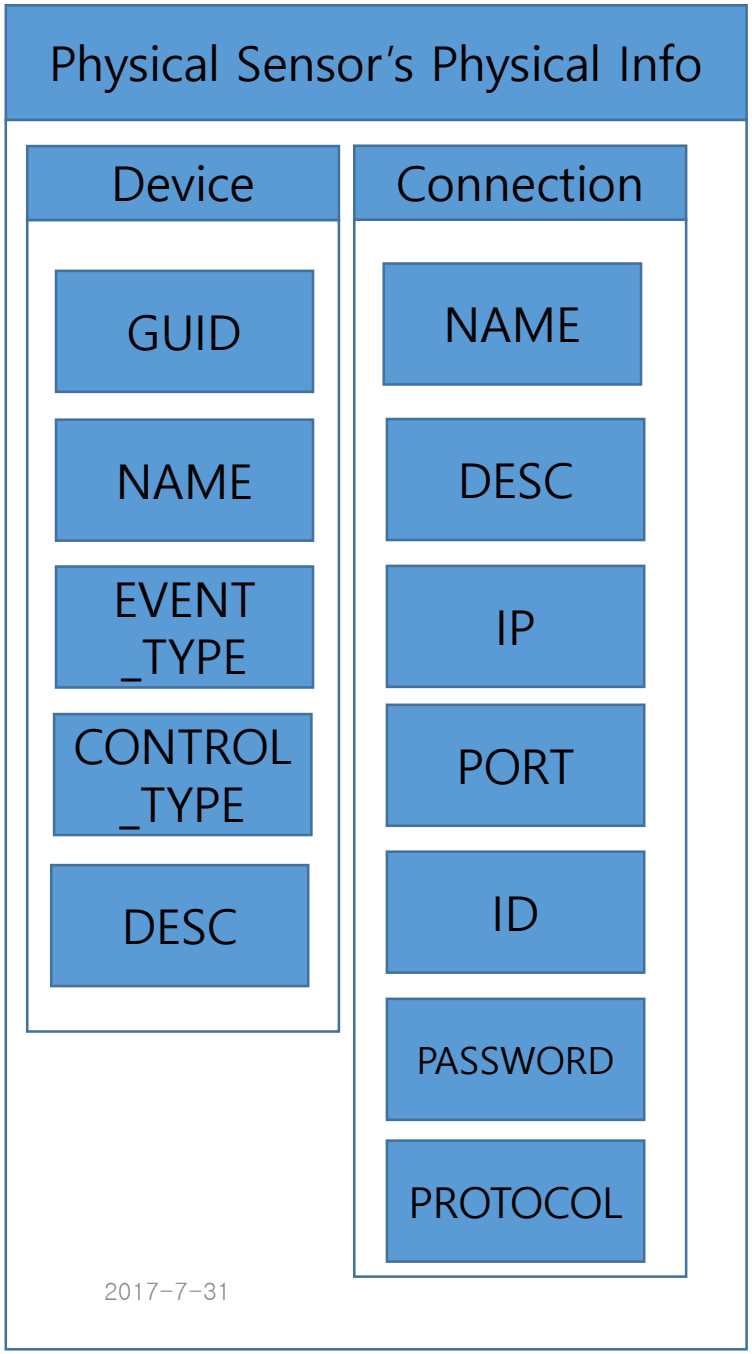
```
<NAME>Airconditioner manager</NAME>
<DESC>Connection info</DESC>
<IP>1.1.1.1</IP>
<PORT>8080</PORT>
<ID>user1</ID>
<PASSWORD>pass1</PASSWORD>
<PROTOCOL>TCP</PROTOCOL>
```

Physical Sensor Connection Description

Connection Info. Fields	Physical Sensor Device Connection Information
NAME	Name related to connection information
DESC	Description of connection information
IP	IP address for a physical sensor device
PORT	Port for a physical sensor device
ID	User account for accessing a physical sensor device
PASSWORD	User account password for accessing a physical sensor device
PROTOCOL	Communication protocol

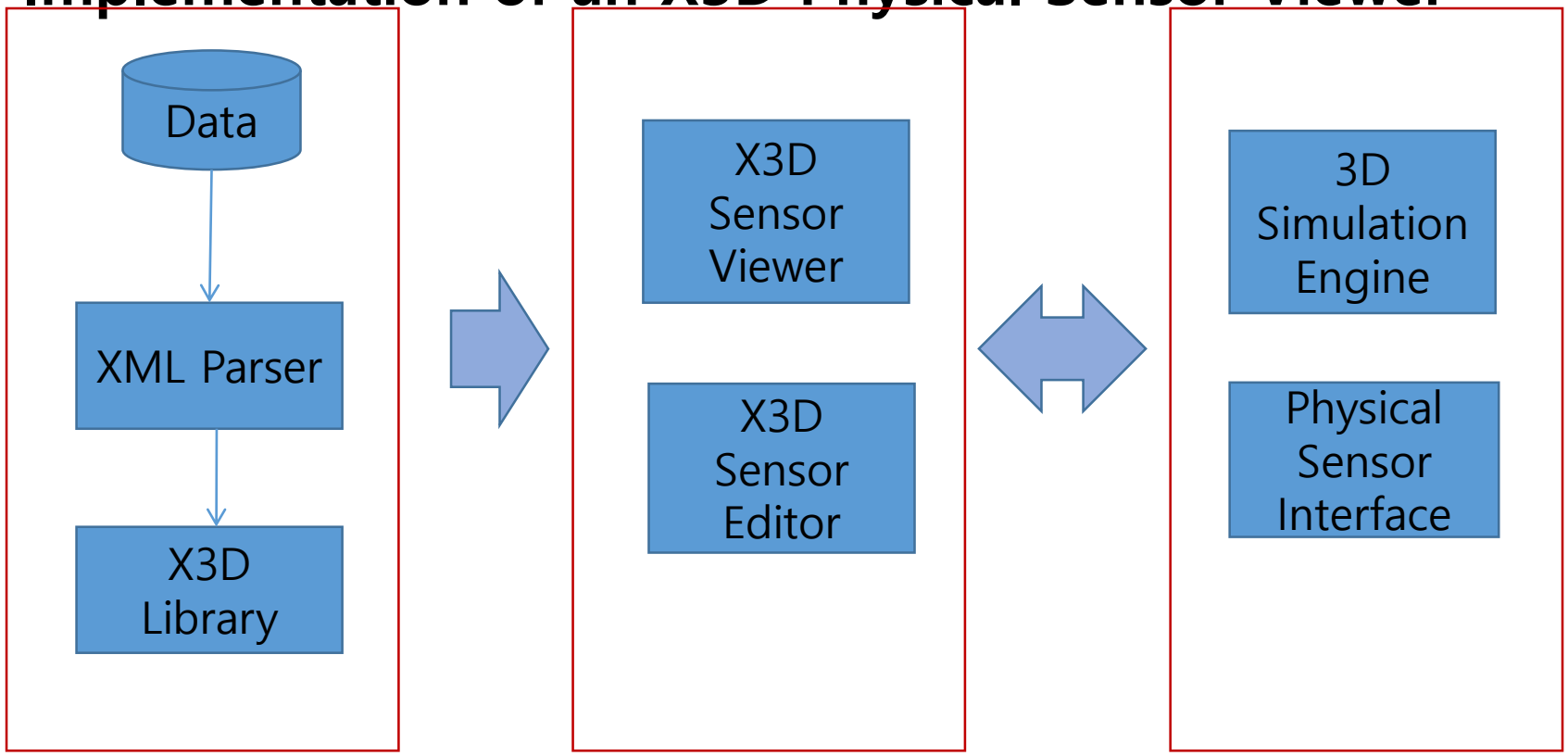

Physical Sensor's
Physical Properties & Interface
Device
Properties
GUID
NAME
EVENT
_TYPE
CONTROL
_TYPE
DESC
Interface
Connection
NAME
DESC
IP
PORT
ID
PASSWORD
PROTOCOL
User Defined
Thermometer
RFID
PLC
Detailed Type
Definition
Temperature
Humidity
Digital
Input
Digital
Output
Analog
Input
Misc

Data Structure of Physical Sensor Properties in a Scene



<Physical Properties of a Physical Sensor>
<Device properties>
<GUID>
<Name>
<EventType>
<ControlType>
<Desc>
<Physical Interface of a Physical Sensor>
<Connection>
<Name>
<Desc>
<IP>
<Port>
<ID>
<Password>
<Protocol>

Implementation of an X3D Physical Sensor Viewer



- X3D document parsing
- Generate X3D geometric data for 3D representation using X3D library

- Display X3D geometric data
- Edit X3D and physical sensors

- Include all libraries for displaying X3D data
- Physical sensors interface
- UI library

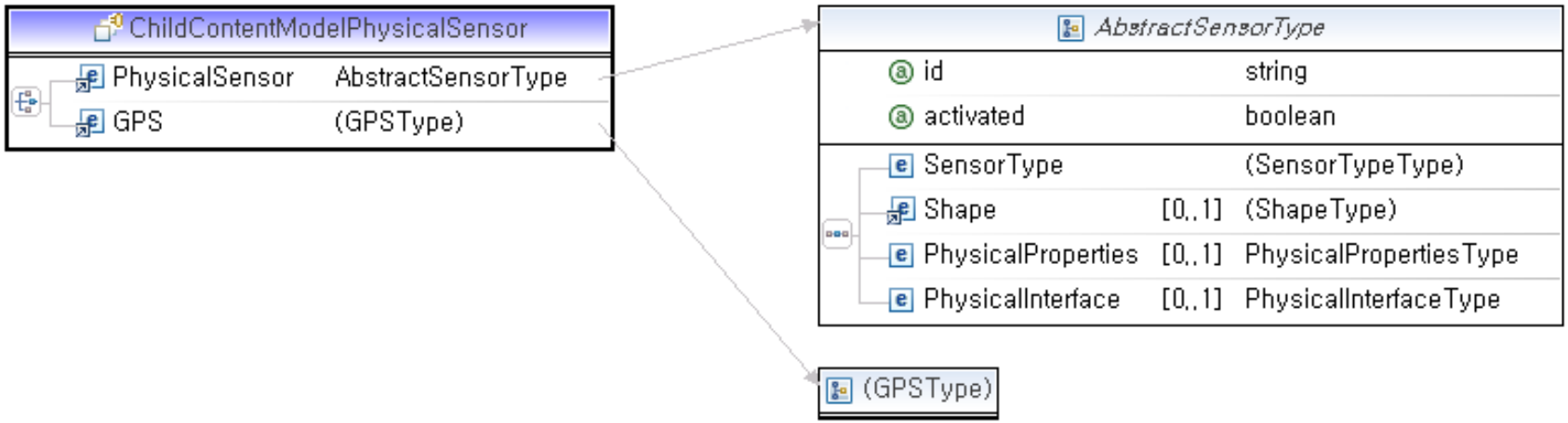
X3D Schema Redefinition for Physical Sensors (1)

```
<xs:redefine schemaLocation="x3d-3.2.xsd">
  <xs:annotation>
    <xs:appinfo>We extend the group GroupingNodeChildContentModel
      and type X3DGroupingNode by adding a reference to the group
      ChildContentModelPhysicalSensor.</xs:appinfo>
    <xs:documentation source="http://..."/>
  </xs:annotation>
  <xs:group name="ChildContentModel">
    <xs:annotation>
      <xs:appinfo>ChildContentModel is the child-node content model
corresponding to X3DChildNode, combining all profiles.  ChildContentModel
can contain most nodes, other Grouping nodes, Prototype declarations and
ProtoInstances in any order and any combination</xs:appinfo>
      <xs:documentation
source="http://www.web3d.org/x3d/specifications/ISO-IEC-FDIS-19775-1.2-
X3D-AbstractSpecification/Part01/components/group.html#
GroupingAndChildrenNodes">children</xs:documentation></xs:annotation>
```

X3D Schema Redefinition for Physical Sensors (2)

```
        <xs:choice>
            <!-- ChildContentModelCore omitted here since included in
                 X3DNode type definition -->
            <xs:group ref="ChildContentModel"/>
            <!-- new content model for physical sensor components -->
            <xs:group ref="ChildContentModelPhysicalSensor"/>
        </xs:choice>
    </xs:group>
</xs:redefine>
```

X3D Schema Redefinition for Physical Sensors (3)



X3D Schema Redefinition for Physical Sensors (4)

```
<xs:group name="ChildContentModelPhysicalSensor">
    <xs:annotation>
        <xs:appinfo>Child-node content model corresponding to Physical Sensor
                                and GPS.</xs:appinfo>
        <xs:documentation source="http://..."/>
    </xs:annotation>
    <xs:choice>
        <xs:element ref="PhysicalSensor"/>
        <xs:element ref="GPS"/>
    </xs:choice>
</xs:group>

<xs:element name="PhysicalSensor" type="AbstractSensorType">
    <xs:annotation>
            <xs:appinfo>Child-node content model representing a physical
                                sensor.</xs:appinfo>
            <xs:documentation source="http://..."/>
        </xs:annotation>
</xs:element>
```

X3D Abstract Physical Sensor Type (1)

```
<xs:complexType name="AbstractSensorType">
   <xs:complexContent mixed="false">
      <xs:extension base="X3DChildNode">
         <xs:sequence>
            <xs:element name="SensorType">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:enumeration value="Camera" />
                     <xs:enumeration value="Electric" />
                     <xs:enumeration value="Environment" />
                     <xs:enumeration value="Light" />
                     <xs:enumeration value="Movement" />
                     <xs:enumeration value="Navigation" />
                     <xs:enumeration value="Position" />
                     <xs:enumeration value="Sound" />
                     <xs:enumeration value="Others" />
                  </xs:restriction>
               </xs:simpleType>
            </xs:element>
```

X3D Abstract Physical Sensor Type (2)

```
            <xs:element ref="Shape" maxOccurs="1" minOccurs="0" />
            <xs:element name="PhysicalProperties"
                  type="PhysicalPropertiesType" minOccurs="0" />
            <xs:element name="PhysicalInterface"
                  type="PhysicalInterfaceType" minOccurs="0" />
      </xs:sequence>
         <xs:attribute name="id" type="ID" />
         <xs:attribute name="activated" type="xs:boolean" />
      </xs:extension >
   </xs:complexContent>
</xs:complexType>
```

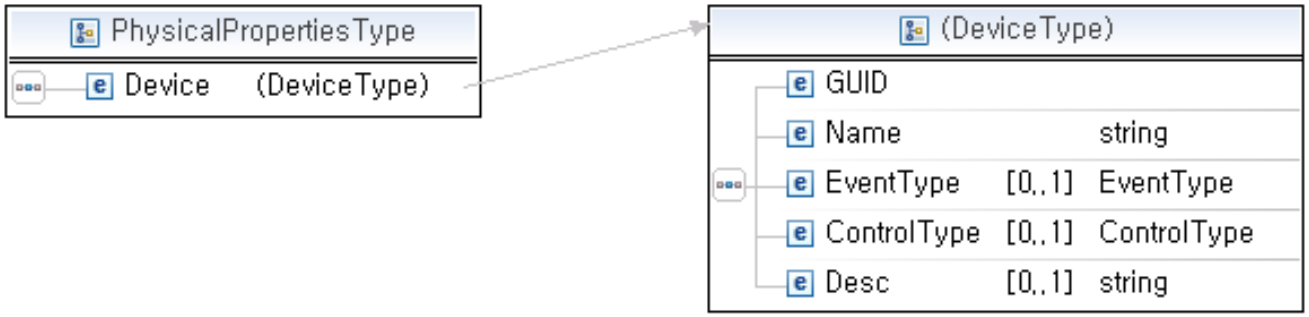
X3D Physical Sensor Attribute Type (1)

```
<xs:complexType name="IntWithUnitType">
   <xs:simpleContent>
      <xs:extension base="xs:integer">
         <xs:attribute name="unit" type="xs:string" />
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

<xs:complexType name="IntRangeWithUnitType">
   <xs:attribute name="min" type="xs:integer" />
   <xs:attribute name="max" type="xs:integer" />
   <xs:attribute name="value" type=”xs:integer" />
   <xs:attribute name="unit" type="xs:string" />
</xs:complexType>
```

X3D Physical Sensor Attribute Type (2)

```
<xs:complexType name="FloatWithUnitType">
   <xs:simpleContent>
      <xs:extension base="xs:float">
         <xs:attribute name="unit" type="xs:string" />
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

<xs:complexType name="FloatRangeWithUnitType">
   <xs:attribute name="min" type="xs:float" />
   <xs:attribute name="max" type="xs:float" />
   <xs:attribute name="value" type="xs:float" />
   <xs:attribute name="unit" type="xs:string" />
</xs:complexType>

<xs:complexType name="StringWithUnitType">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute name="unit" type="xs:string" />
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

X3D Physical Properties Type (1)

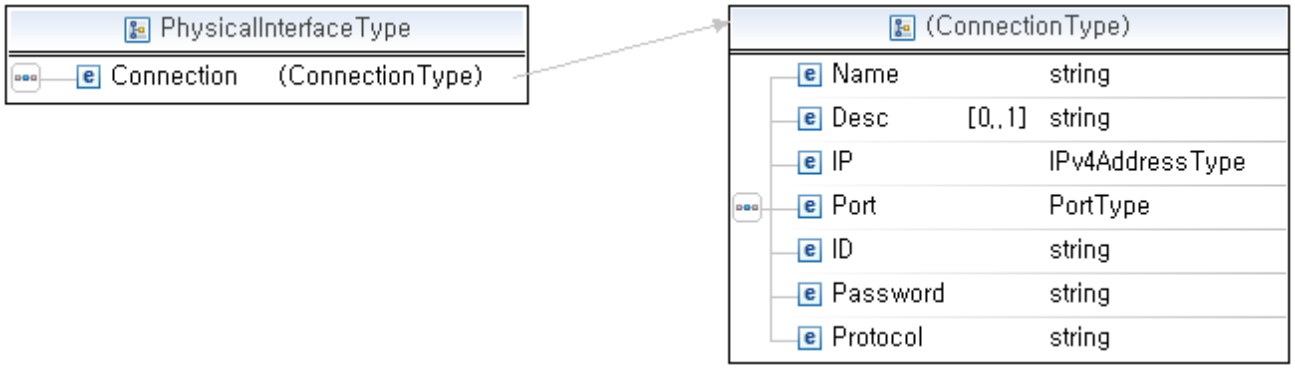



```
<xs:complexType name="PhysicalPropertiesType">
    <xs:sequence>
        <xs:element name="Device">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="GUID" type="GUIDType" />
                    <xs:element name="Name" type="xs:string" />
                    <xs:element name="EventType" type="EventType" minOccurs="0" />
                    <xs:element name="ControlType" type="ControlType" minOccurs="0" />
                    <xs:element name="Desc" type="xs:string" minOccurs="0" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

X3D Physical Properties Type (2)

```
<xs:simpleType name="GUIDType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{4}-[0-9a-fA-F]{12}" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="EventType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Temp" />
        <xs:enumeration value="EventTypeName1" />
        <xs:enumeration value="EventTypeName2" />
        <!-- Other values can be added -->
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ControlType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Temp" />
        <xs:enumeration value="ControlName1" />
        <xs:enumeration value="ControlName2" />
        <!-- Other values can be added -->
    </xs:restriction>
</xs:simpleType>
```

X3D Physical Interface Type (1)




```
<xs:complexType name="PhysicalInterfaceType">
    <xs:sequence>
        <xs:element name="Connection">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Name" type="xs:string" />
                    <xs:element name="Desc" type="xs:string" minOccurs="0" />
                    <xs:element name="IP" type="IPv4AddressType" />
                    <xs:element name="Port" type="PortType" />
                    <xs:element name="ID" type="xs:string" />
                    <xs:element name="Password" type="xs:string" />
                    <xs:element name="Protocol" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

X3D Physical Interface Type (2)

```
<xs:simpleType name="IPv4AddressType">
    <xs:restriction base="xs:string">
        <xs:pattern value="((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-
9])₩.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PortType">
    <xs:restriction base="xs:unsignedShort">
        <xs:minInclusive value="1" />
    </xs:restriction>
</xs:simpleType>
```

X3D Physical Sensors Example (1)

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D version="3.2" profile="Immersive"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="x3d-3.2-ext.xsd">
   <head>
      <meta name='filename' content='chair.x3d' />
   </head>
   <Scene>
      <Viewpoint position="0.0 0.0 1.0" description="1M view" />
      <NavigationInfo type='"EXAMINE" "ANY"' />
      <Background groundColor="0.05 0.1 0.3" skyColor="0.05 0.1 0.3 " />
      <Transform scale="1 1 1" translation="-900 600 -900">
         <Shape>
            <Appearance>
               <Material diffuseColor="0.0902 0.05882 0" />
            </Appearance>
```

X3D Physical Sensors Example (2)

```
            <IndexedFaceSet
               coordIndex=" 20, 19, 18, -1, 20, 18, 17, -1, 21, 20,
                  22, 21, 16, -1, 22, 16, 15, -1, 23, 22, 15, -1, 23, 15, 14, -1">
               <Coordinate point="448.6 326.6 -55.04, 448.6 289.1 -65.81" />
            </IndexedFaceSet>
         </Shape>
         <PhysicalSensor activated="true" id="id0"
xsi:type="CameraSensorType">
            <SensorType>Camera</SensorType>
            <Shape>
               <Sphere radius='10' />
               <Appearance>
                  <Material diffuseColor="0.0 0.0 1.0" />
               </Appearance>
            </Shape>
```

X3D Physical Sensors Example (3)

```
<PhysicalProperties>
   <Device>
      <GUID>21EC2020-3AEA-4069-A2DD-08002B30309D</GUID>
      <Name>Camera-101</Name>
      <EventType>Temp</EventType>
      <ControlType>Temp</ControlType>
      <Desc>Description of sensor device and type</Desc>
   </Device>
</PhysicalProperties>
<PhysicalInterface>
   <Connection>
      <Name>Camera manager</Name>
      <Desc>Connection info</Desc>
      <IP>111.111.222.222</IP>
      <Port>8080</Port>
      <ID>user1</ID>
      <Password>passwd1</Password>
      <Protocol>TCP</Protocol>
   </Connection>
</PhysicalInterface>
```

X3D Physical Sensors Example (4)

```
        <Width unit="mm">1.0</Width>
        <Height unit="mm">1.0</Height>
        <AspectRatio unit="percent">10.0</AspectRatio>
        <ActualPixelCount>10</ActualPixelCount>
        <Megapixels>10</Megapixels>
    </PhysicalSensor>
</Transform>
<Transform translation='-2.4 0.2 1.0' rotation='0.0 0.707 0.707 0.9'>
    <Shape>
        <Box />
        <Appearance>
            <Material diffuseColor='0.0 0.0 1.0' />
        </Appearance>
    </Shape>
```

X3D Physical Sensors Example (5)

```
<PhysicalSensor activated="true" id="id1"
   xsi:type="MovementSensorType">
   <SensorType>Movement</SensorType>
   <Shape>
      <Sphere radius='2.3' />
      <Appearance>
         <Material diffuseColor="1.0 1.0 1.0" />
      </Appearance>
   </Shape>
   <PhysicalProperties>
      <Device>
         <GUID>11EC2020-3AEA-4069-A2DD-08002B30309D</GUID>
         <Name>Movement-101</Name>
         <EventType>Temp</EventType>
         <ControlType>Temp</ControlType>
         <Desc>Description of sensor device and type</Desc>
      </Device>
   </PhysicalProperties>
```

X3D Physical Sensors Example (6)

```
        <PhysicalInterface>
            <Connection>
                <Name>Movement manager</Name>
                <Desc>Connection info</Desc>
                <IP>111.111.222.222</IP>
                <Port>8080</Port>
                <ID>user1</ID>
                <Password>passwd1</Password>
                <Protocol>TCP</Protocol>
            </Connection>
        </PhysicalInterface>
    </PhysicalSensor> <GPS />
</Transform>
```

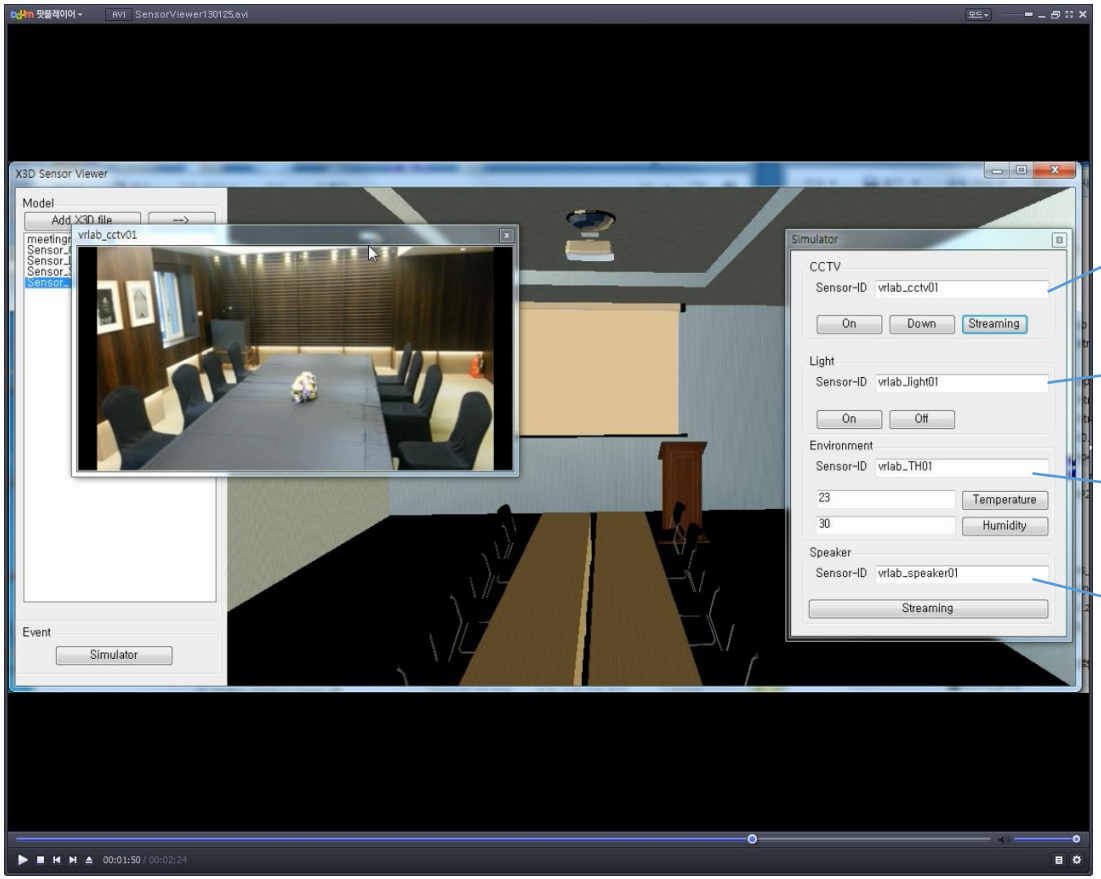
X3D Physical Sensors Example (7)

```
        <Shape>
            <Cone />
            <Appearance>
                <Material diffuseColor='0.0 1.0 0.0' />
            </Appearance>
        </Shape>
        <PhysicalSensor activated="true" id="id2"
xsi:type="NavigationSensorType">
            <SensorType>Electric</SensorType>
            <Shape>
                <Sphere radius='15.5' />
                <Appearance>
                    <Material diffuseColor="0.0 1.0 1.0" />
                </Appearance>
            </Shape>
        </PhysicalSensor>
```

X3D Physical Sensors Example (8)

```
        <PhysicalSensor activated="true" id="id3"
xsi:type="ProximitySensorType">
            <SensorType>Proximity</SensorType>
            <Shape>
                <Sphere radius='10' />
                <Appearance>
                    <Material diffuseColor="0.0 0.0 1.0" />
                </Appearance>
            </Shape>
        </PhysicalSensor>
    </Scene>
</X3D>
```

X3D Physical Sensor Viewer (1)



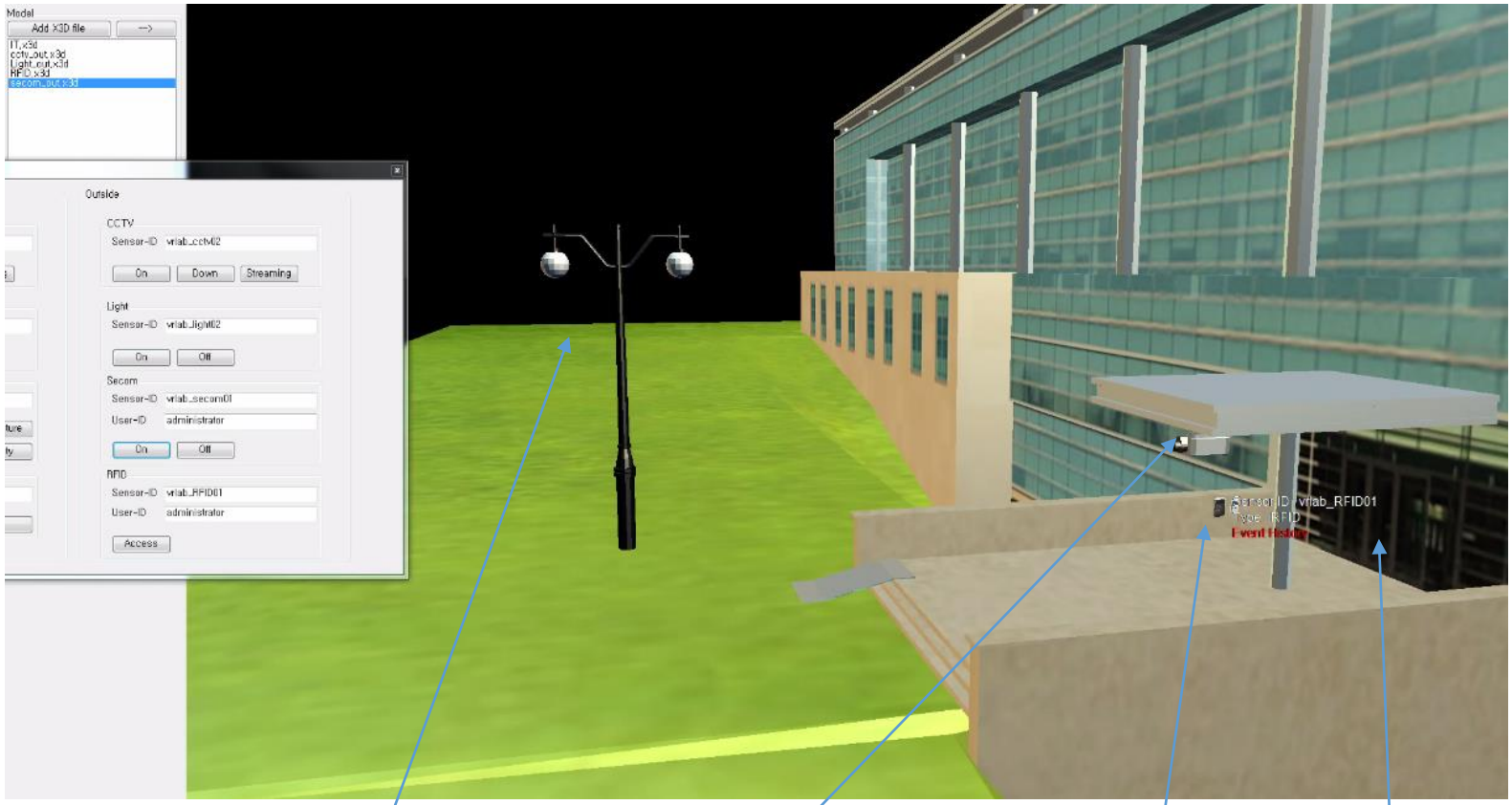
Camera sensor

Light sensor

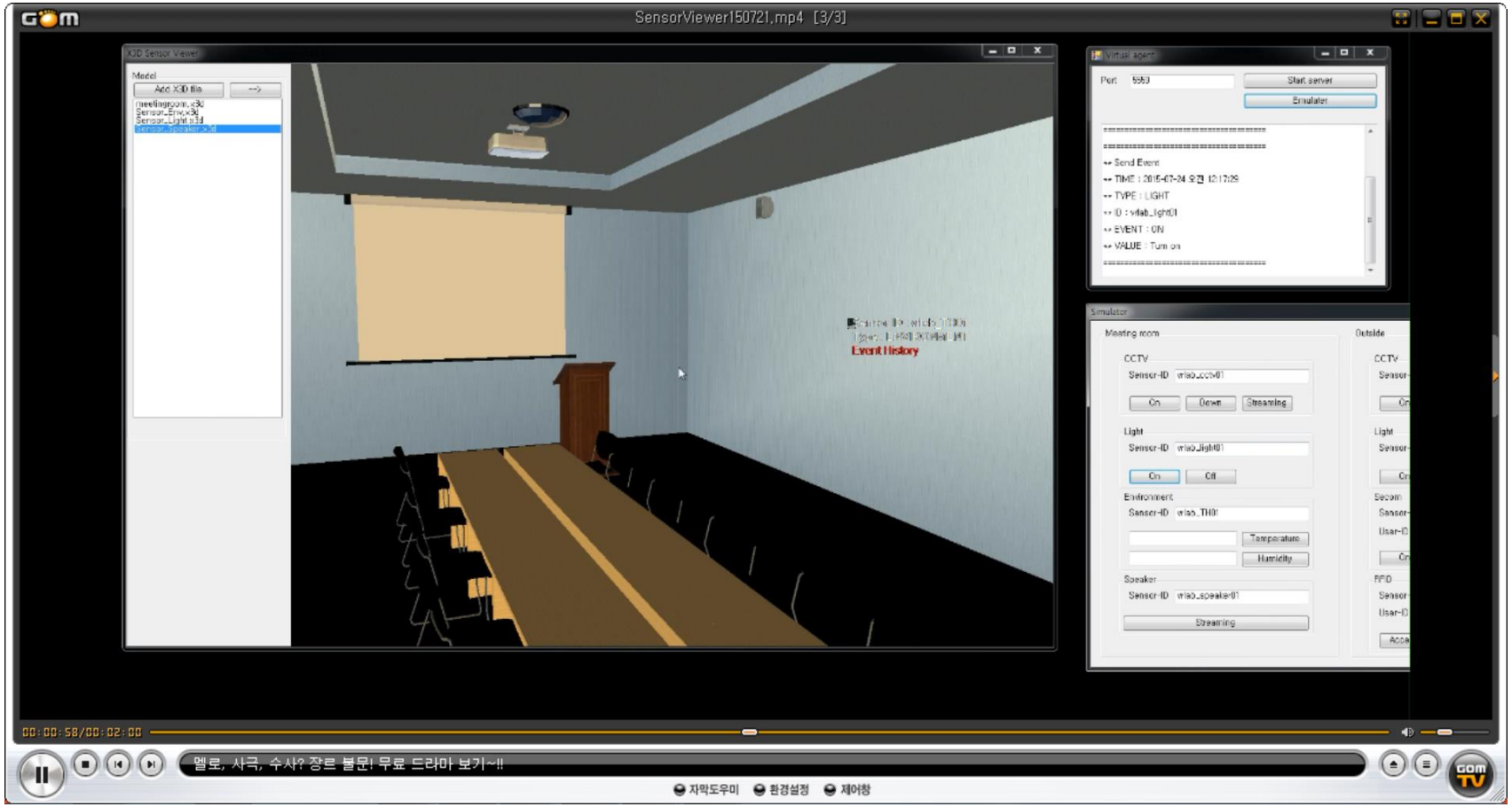
Environment sensor

Sound sensor

X3D Physical Sensor Viewer (2)

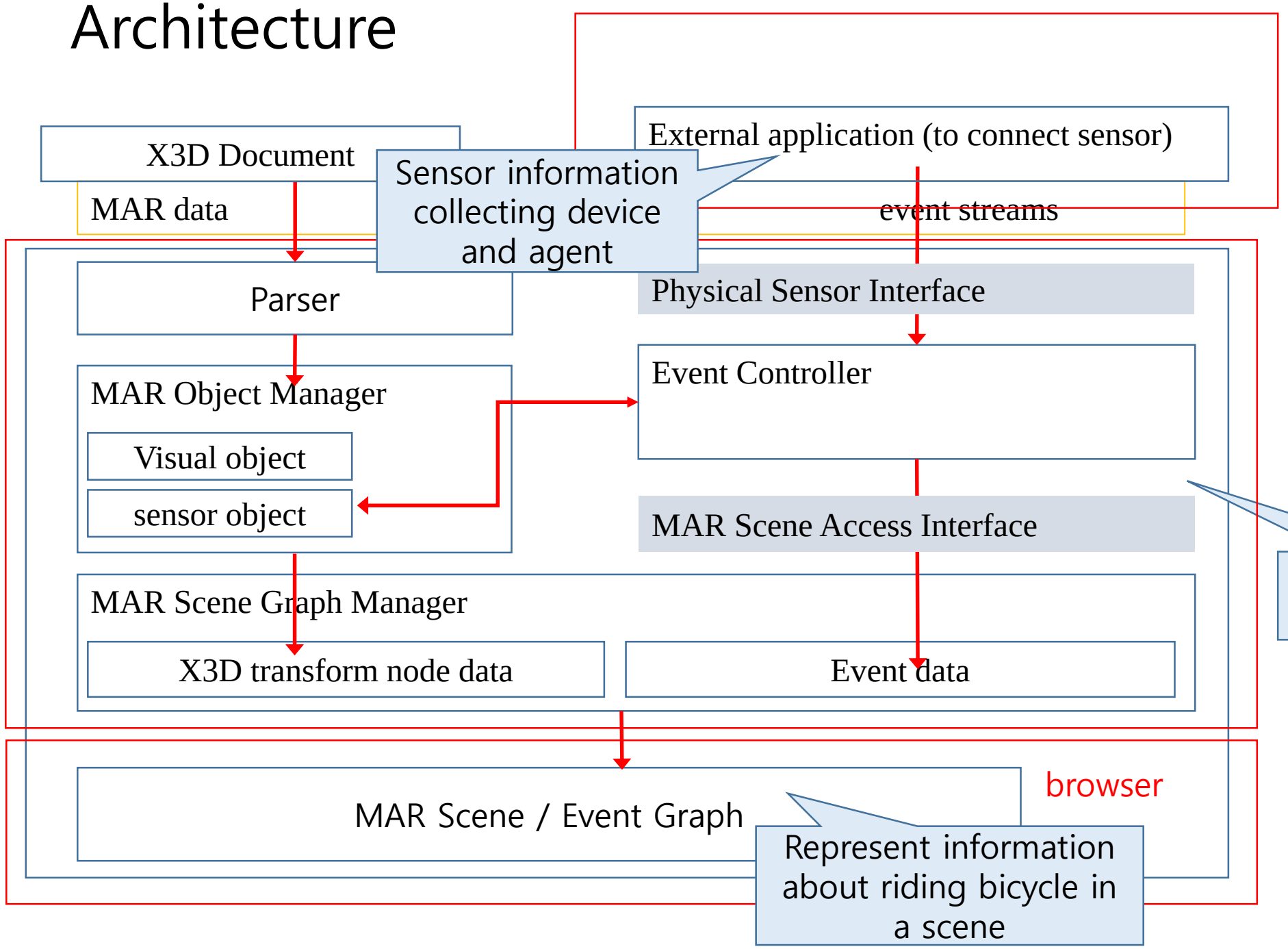


X3D Physical Sensor Viewer (3)



Implementation of Bike Sensor Representation

Architecture



Operation

Test using Cycling computer (Sensor values are obtained by GARMIN)

ANT+ communication

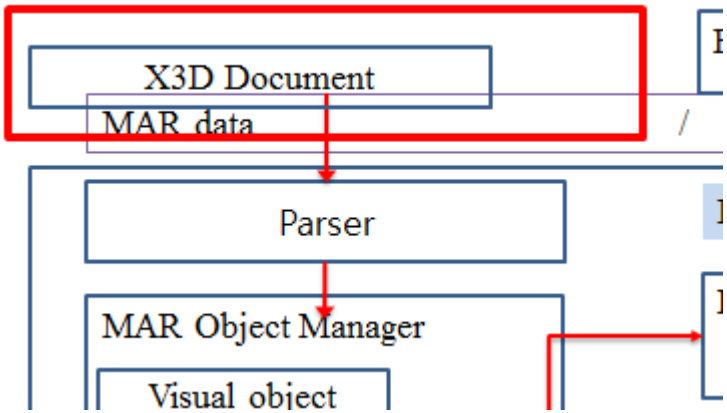
Collect sensor information using mobile devices considering movement

External application (to connect sensor)

Transfer event information to a scene using socket communication

MAR Scene / Event Graph

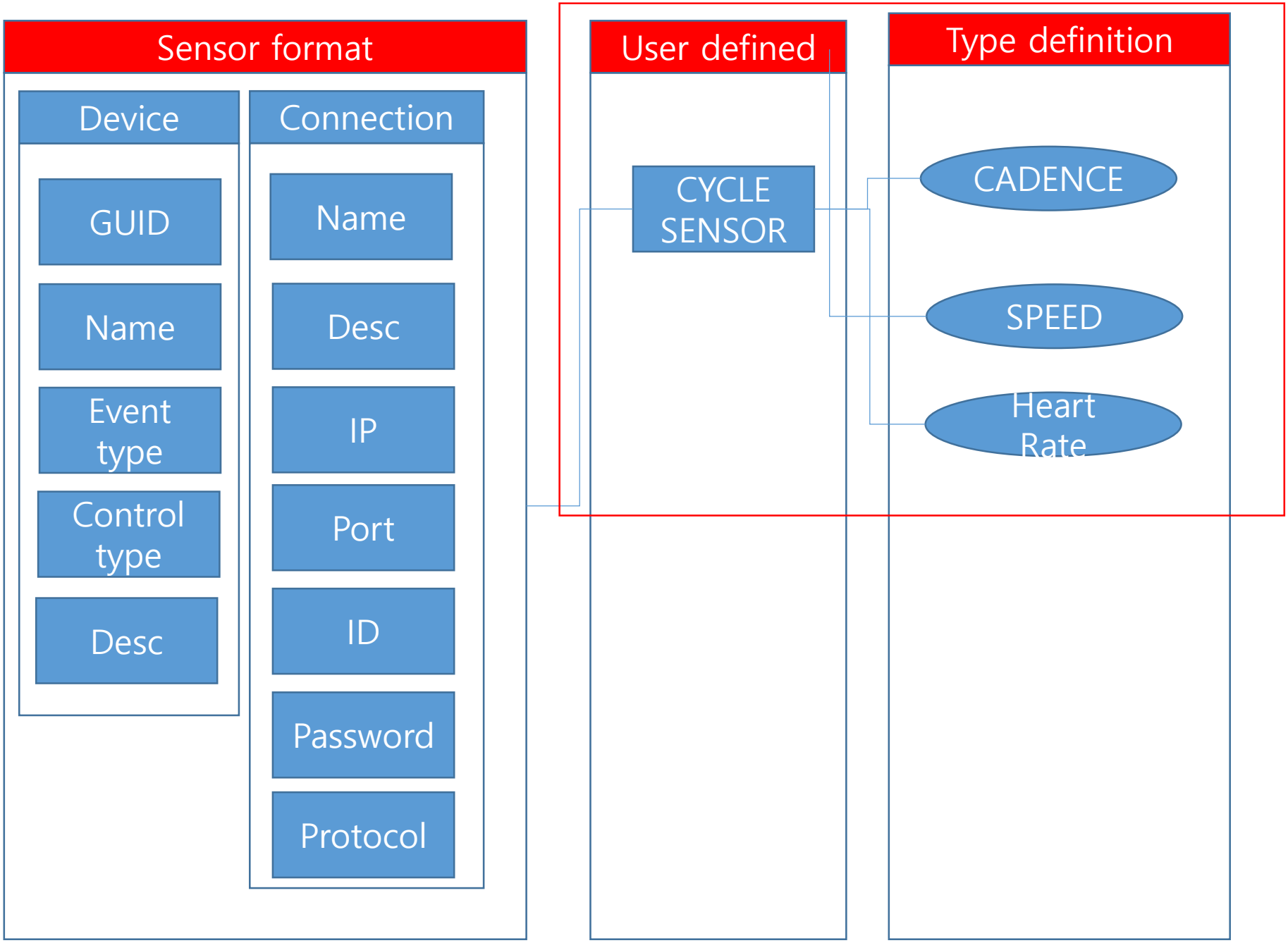
X3D Document



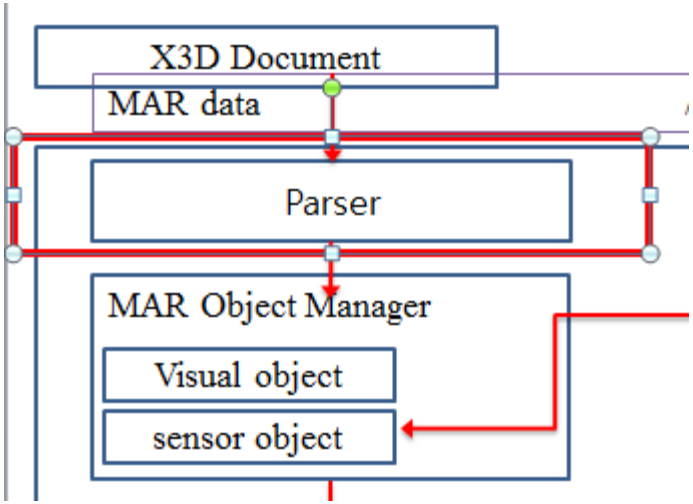
Specify bike sensor definition, sensor availability, event type in an X3D document

```
<Sensor_Node available="true" type="CYCLE" id="vrlab_Cycle">
  <server type="TCP" ip="192.168.0.11" port="5559" id="anonymous" passwd="" />
  <event_list available-event="CADENCE, SPEED, GPS" />
</Sensor_Node>
```

Bike sensor definition



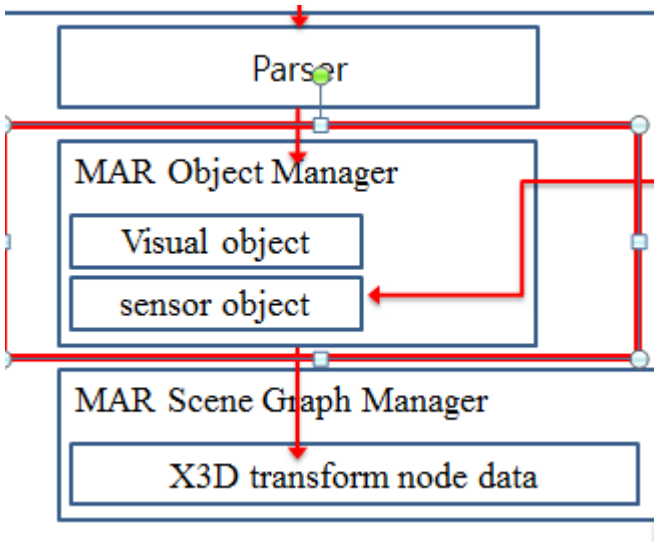
Parser



Bike sensor information specified in the X3D document is transferred to the MAR object manager in the Parser

```
#region Set Datta

public void AddX3DData(int nIndex , string strDir, string strFileName, DataSet dsData)
{
    m_Scene.AddX3D(nIndex, strDir, strFileName, dsData);
}

#endregion
```

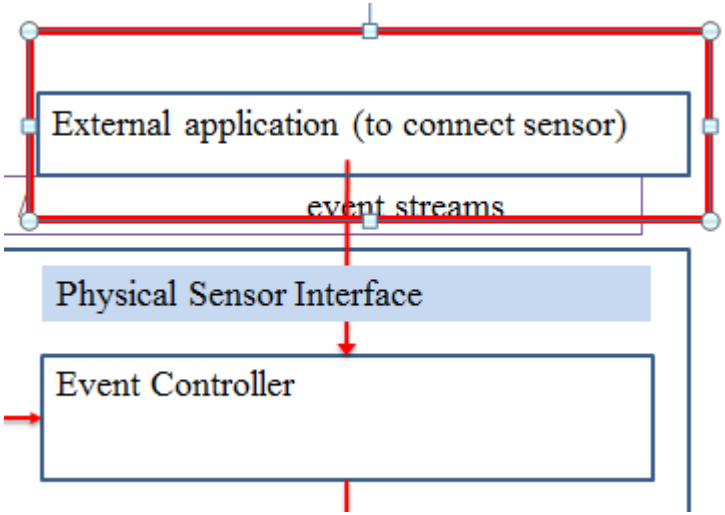
Mar Object Manager



MAR object manger transfers sensor information transformed in the Parser to the Scene Graph

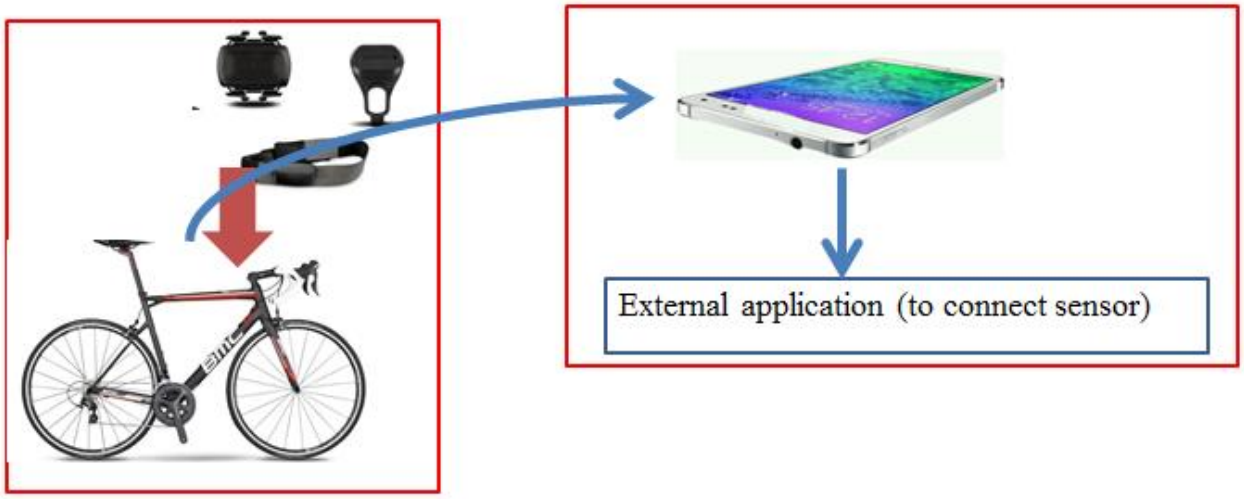
```
if (dsData.Tables.Contains("Sensor_Node") == true)
{
    string Sensor_Node_Id = dsData.Tables["Sensor_Node"].Rows[0]["Sensor_Node_Id"].ToString();

    bSensor = true;
    strSensorID = dsData.Tables["Sensor_Node"].Rows[0]["ID"].ToString();
    strSensorType = dsData.Tables["Sensor_Node"].Rows[0]["type"].ToString();
```

External Application

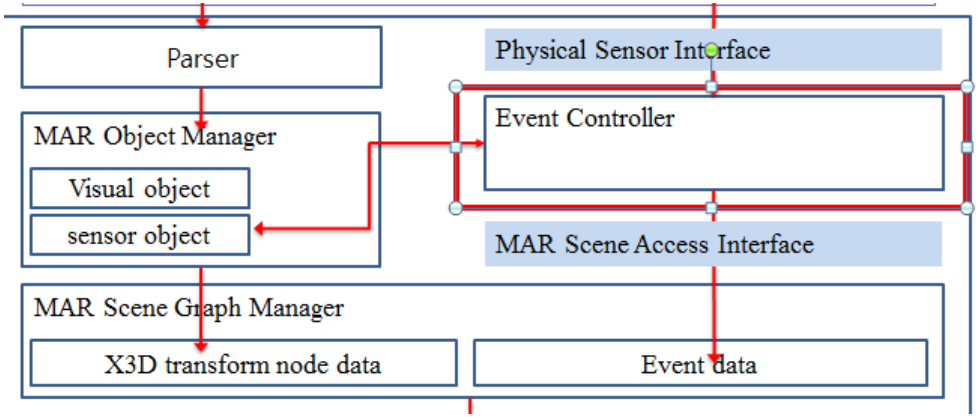


Application collects event generated by bike sensors and transfers it to the event controller

Event generated from sensors is collected by the mobile device and is transferred to the event controller by the application on the device



Event Controller

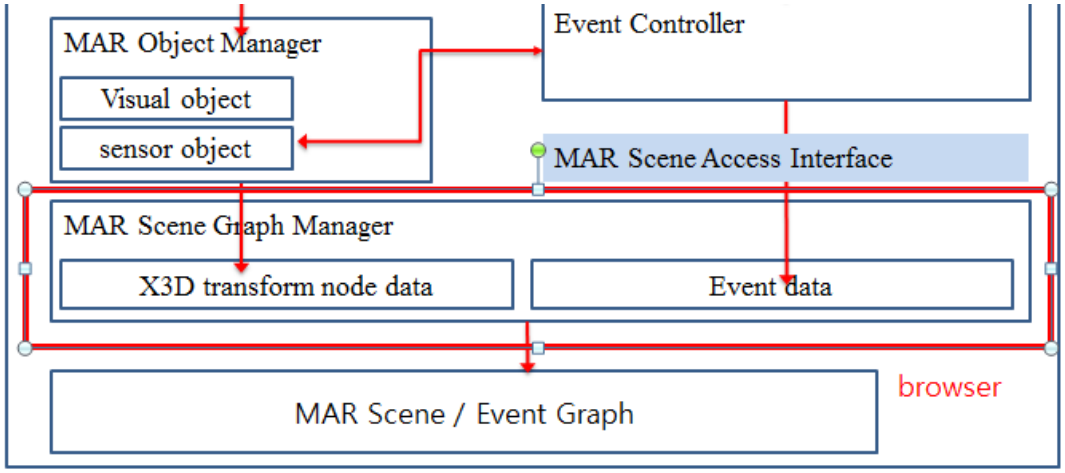



```
public void SetEvent(DataTable dtEvent)
{
    string ID = dtEvent.Rows[0]["ID"].ToString();
    string TYPE = dtEvent.Rows[0]["TYPE"].ToString();
    string EVENT = dtEvent.Rows[0]["EVENT"].ToString();
    string VALUE = dtEvent.Rows[0]["VALUE"].ToString();

    if (m_dicEvent.ContainsKey(ID) == true)
    {

        string USER = string.Empty;
        if (dtEvent.Columns.Contains("USER") == true)
        {
            USER = dtEvent.Rows[0]["USER"].ToString();
            dtEvent.Columns.Remove("USER");
        }

        m_dicEvent[ID].Rows.Add(dtEvent.Rows[0].ItemArray);
```

The event controller receives event generated from bike sensors, saves the event, and transfer it to an object.

Sensor data is processed in relation to SensorID, Event type (CADENCE,SPEED,GPS) defined in the X3D document.

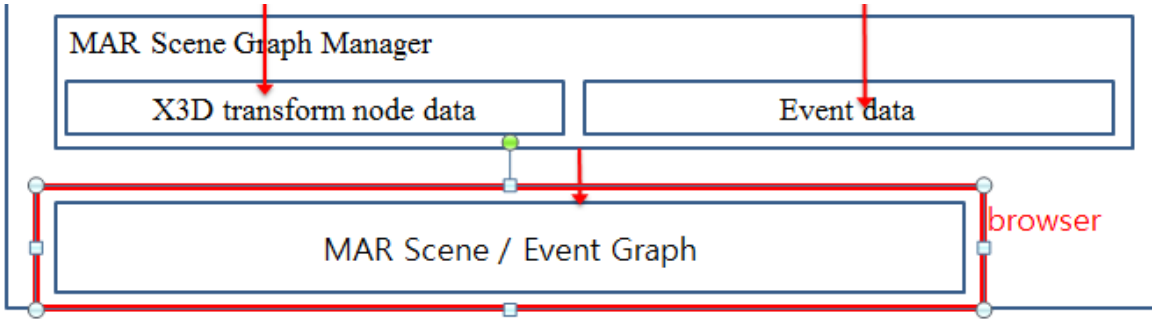
Mar Scene Graph Manager



MAR scene graph manager integrates and manages sensor data and represents it to the scene

```
// 기하데이터
public List<X3DTranform> m_lTransform = new List<X3DTranform>();

// 센서 정보
private bool m_bSensor = false;
public bool IsSensor
{
    get { return m_bSensor; }
    set { m_bSensor = value; }
}

private string m_strSensorID;
public string SensorID
{
    get { return m_strSensorID; }
    set { m_strSensorID = value; }
}

private string m_strSensorType;
public string SensorType...

public List<string> m_lEventList = new List<string>();
```

Mar Scene / Event Graph




```
public override bool Draw()
{
    //DrawTile();

    m_Camera.Update(new Vector2(m_Mainclass.EngineInput.MouseRightMovement.X, -m_Ma

    if(m_dtModel != null )
    {

        foreach(DataRow dr in  m_dtModel.Rows )
        {
            X3DModel model = (X3DModel)dr["MODEL"];
            int nIndex = int.Parse(dr["INDEX"].ToString());

            bool bPick = false;


            foreach (X3DTranform transform in model.m_lTransform)
            {
```

MAR scene and event graph represents bike 3D and sensor data received by bike sensors.

Bike Sensor Representation Video (Implementation Results)

- Bike sensor information: time, latitude, longitude, elevation, cadence, temperature

Conclusions

- X3D based physical sensor representation
- Extended data definition for representing and simulating physical sensors in X3D
- Scene graph definition including physical sensors
- Definition of physical properties and interface of each physical sensor
- Implementation of a physical sensor viewer and user interface for each type of physical sensor
- Implementation of bike sensors representation