ISO/IEC CD 18038
Sensor Representation in MAR
(Updates)

ISO/IEC JTC 1/SC 24/WG 9 & Web3D Meetings

January 21-24, 2019

Myeong Won Lee (The University of Suwon) and
Gerard J. Kim (Korea University)

Result of CD voting (2018-1-7)

Member responses:	
Votes cast (10)	Australia (SA) China (SAC) Egypt (EOS) France (AFNOR) Japan (JISC) Korea, Republic of (KATS) Russian Federation (GOST R) Switzerland (SNV) United Kingdom (BSI) United States (ANSI)
Comments submitted (2)	Austria (ASI) Malaysia (DSM)
Votes not cast (0)	

Vote	No. of NBs
Approval	6
Disapproval	1
Abstention	3
Comments	2

Comments (1)

Template for comments and secretariat observations

Date:2018-01-07 | Document:

MB/ NC[1]	Line number	Clause/ Subclause	Paragraph/ Figure/Table	Type of comment[2]	Comments	Proposed change
JP0 001				Ge	We will change our votes to approval if our comments are properly considered and adopted.	
JP1 002	Overall			Te	GPS is not a general term, but the US version of GNSS.	Use GNSS instead of GPS.
JP2 003	Overall			Te	GPS MAR world sounds a little bit strange. It would mean MAR world defined with geospatial coordinates.	Rename GPS MAR world.
SA 004	1	03.01.21	1	ed	'Sounds strength' of the environment is a bit unclear, maybe better to be just a device used to detect sound. Assume it would not just be 'sound strength' but may	Cover 'cover, tonality, voice recognition and positional information among others.'
SA 005	1	03.01.23	1	ed	GUID sounds more Microsoft specific, whereas UUID may be a more platform agnostic term to be used.	Use UUID
SA 006	2-5	04.01	4	ed	Types of physical sensor devices include … list is fairly unwieldly, recommend matching this with the types in the following subsection and clause 7.2.10.	Use the same listing as in 4.3.xx and 7.2.10

Comments (2)

SA 007	2-16	04.03	List	ed	Match categories and order of 4.3.xx and 7.2.10 for ease of readability	Reorder
SA 008	Figure	05.02.1	Figure 5.2	ed	Does the name need to be in a set format? (No spaces / Capitalisation / Camel Case etc.) Or is this handled by the GUID?	Change Airconditioner to Air Conditioner
SA 009	Figure	06.01	Figure 6.1	ed	MAR document file should match MAR Data File at 6.1	Change MAR document file to MAR Data File
SA 010	3	07.02.12	Code Block	te	Does this standard only support IPv4?	Clarify, and potentially make this more generic to cater for future protocols
SA 011	Table	A.10	Table A.10	ed	Missing Unit / Data Type (populate with Schema elements from Annex B)	Add Unit / Data Type columns.
SA 012	All of Annex A	A.x	Annex A	te	Should numbers that can have a decimal precision all be of type float? This may be required if developers want to use numbers to a greater precision, where an integer may not be nuanced enough.	Recommend make numbers that might be decimal as floats for future proofing.
SA 013	34	B.1	SoundSensor Type	ed	Consistency with element name 'VCC' / 'Vcc'	Change this in schema to be 'element name ='Vcc'
SA 014	2	Bibliography	Ref 2	ed	Recommend changing Wikipedia reference	Change reference to something else

Status for DIS Text Submission

- Editing meeting for DIS
  - WG9 Web Meeting, October 25, 2018
- Disposition of Comments: prepared
- DIS text: almost prepared
- DIS submission: within 1-2 weeks (February, 2019)

Status for DIS Text Submission

- Editing meeting for DIS
  - WG9 Web Meeting, October 25, 2018
- Disposition of Comments: prepared
- DIS text: almost prepared
- DIS submission: within 1-2 weeks (February, 2019)

Contents of specification (1)

1. Scope
2. Normative references
3. Terms, definitions, acronyms, and abbreviations
4. Concepts
   – Overview
   – Scope of physical sensor representation
   – Physical sensor types
   – Sensor representation
5. Sensor 3D scene graph
   – Definition of a sensor 3D scene graph
   – Physical properties and interfaces with real worlds
6. A system architecture for physical sensor representation
   – A system architecture for physical sensors
   – System framework
7. XML definition of physical sensor representation
   – Structure of physical sensor representation
   – XML schema definition

Contents of specification (2)

8. Conformance
   - Conformance criteria
   - Conformance area

Annex A (Informative) Examples of physical sensor types and parameters

Annex B (Informative) Schema for sensor MAR representation

Annex C (Informative) An example of sensor MAR representation based on sensor MAR schema

Annex D (Informative) Implementation examples of sensor MAR representation

1. Scope

What to focus on

- How to represent physical IoT sensors in a 3D scene
  - Sensor appearance in a scene may or may not be required, depending on the sensor and the application
  - **Precise location and orientation of a IoT sensor**
  - **Unit specification**
  - 3D scene change as a result of each physical sensor
- What to represent about physical IoT sensors
  - 3D scene change
- What to do
  - 3D simulation of each physical IoT sensor in VR
- Reason for doing
  - Physical IoT sensor control and management in a 3D scene
  - Facility management using VR

Use cases

- IoT sensor management in 3D scenes
  - Facility management
    - CCTV management
    - RFID management
    - Security devices management
    - Light management
    - Electric management
- IoT sensor interaction between real and virtual worlds
  - Sensor Input/Output between real and virtual worlds
    - GPS sensor output to 3D scenes
    - Camera sensor output to 3D scenes
    - Motion sensor output to 3D scenes

Scope of sensor representation

- Concepts of physical sensors in MAR
- Requirements for 3D simulation of physical sensors
- How to represent physical sensors in a 3D virtual environment
- How to organize a 3D scene with sensors
- How to define an abstract model for representing physical sensors in a 3D virtual environment
- How to define a system architecture for physical sensors in a 3D virtual environment
- How to use physical sensors in a 3D virtual environment
- Types of physical sensors to be considered for the sensor representation

An abstract data model for physical sensor representation

Define an abstract model for representing visual, functional, and physical properties of sensors in a 3D world

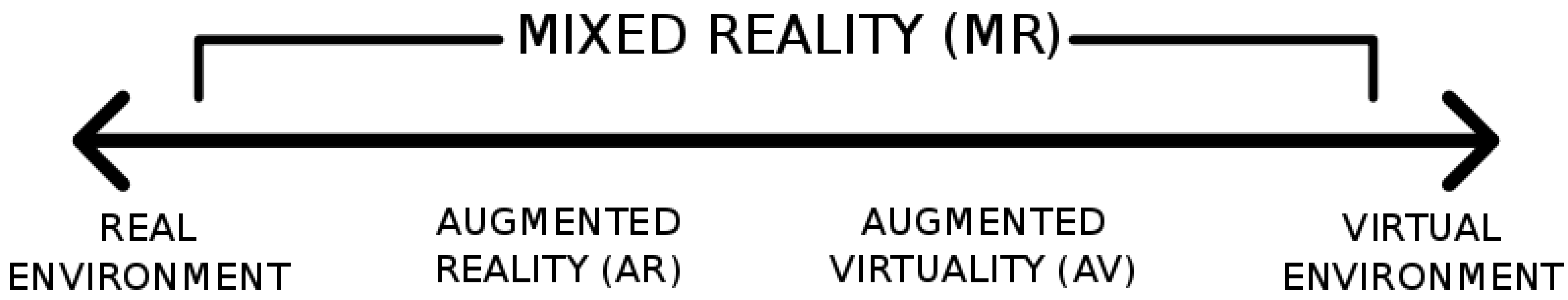
- Visual
  - Visual representation of sensors in 3D worlds
- Functional
  - 3D Simulation for representing the functions of physical sensors in 3D worlds
- Physical
  - Representation of real data stream from sensors in 3D worlds

Data model and interface for physical sensors

- Define a 3D data structure for physical sensors in a 3D scene
- Define the location and orientation of a physical sensor in a 3D scene
- Define visual and functional properties of each physical sensor in a 3D scene
- Define physical properties of each physical sensor in a 3D scene
- Define an interface format for import/export of physical sensor information to/from real worlds
- Define a user interface for controlling physical sensors in a 3D scene

Concepts

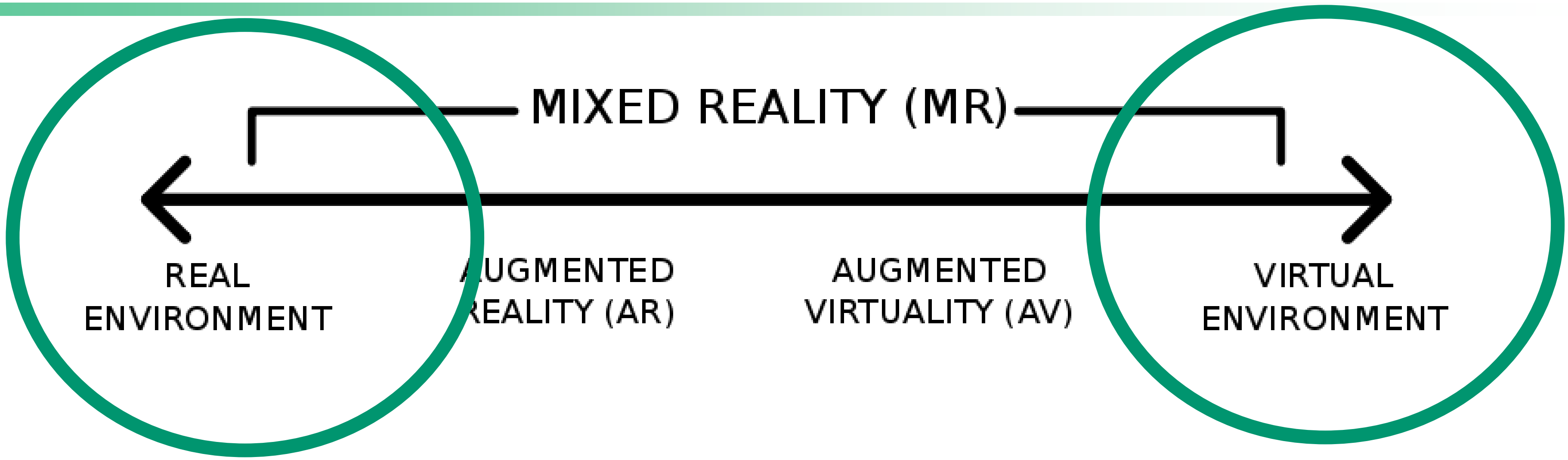
Mixed Reality



- Paul Milgram and Fumio Kishino, 1994
  - A mixed reality as “anywhere between the extrema of the virtuality continuum
  - The Virtuality Continuum extends from the completely real through to the completely virtual environment with augmented reality and augmented virtuality ranging between.

An example mixed reality, Wikipedia, 2012

Sensor representation

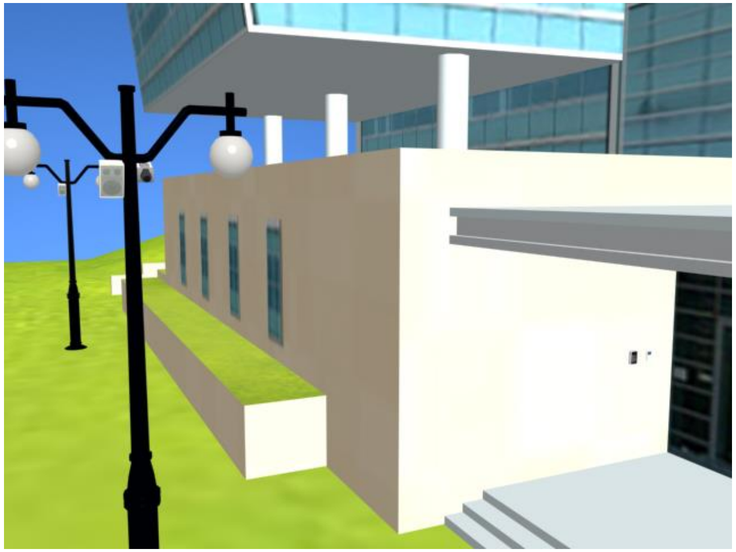


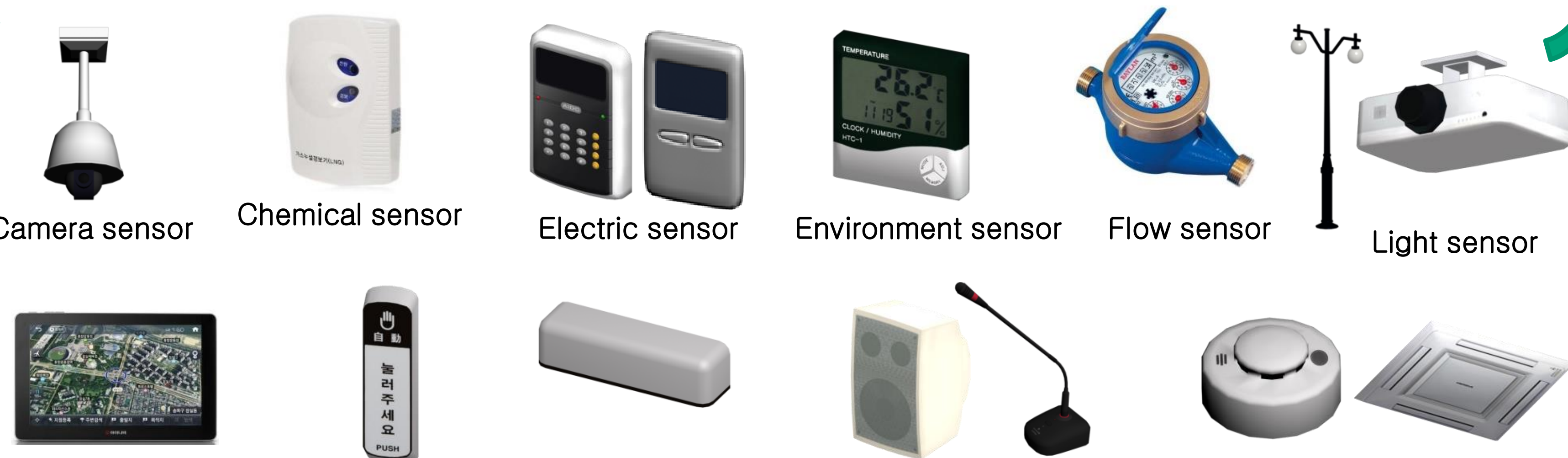
Real world simulation of sensor functions in VR

Sensor Representation

- Definition
  - Integration of real physical sensors and a 3D virtual world
  - A mixed reality world where physical sensors are represented precisely with their physical properties in a 3D virtual world
  - Functional simulation of physical sensors in 3D virtual worlds
- Objectives
  - Exchange IoT sensor information including visual, functional, and physical properties in 3D virtual environments
  - Represent, manage, and control physical IoT sensors with their physical properties in 3D virtual environments

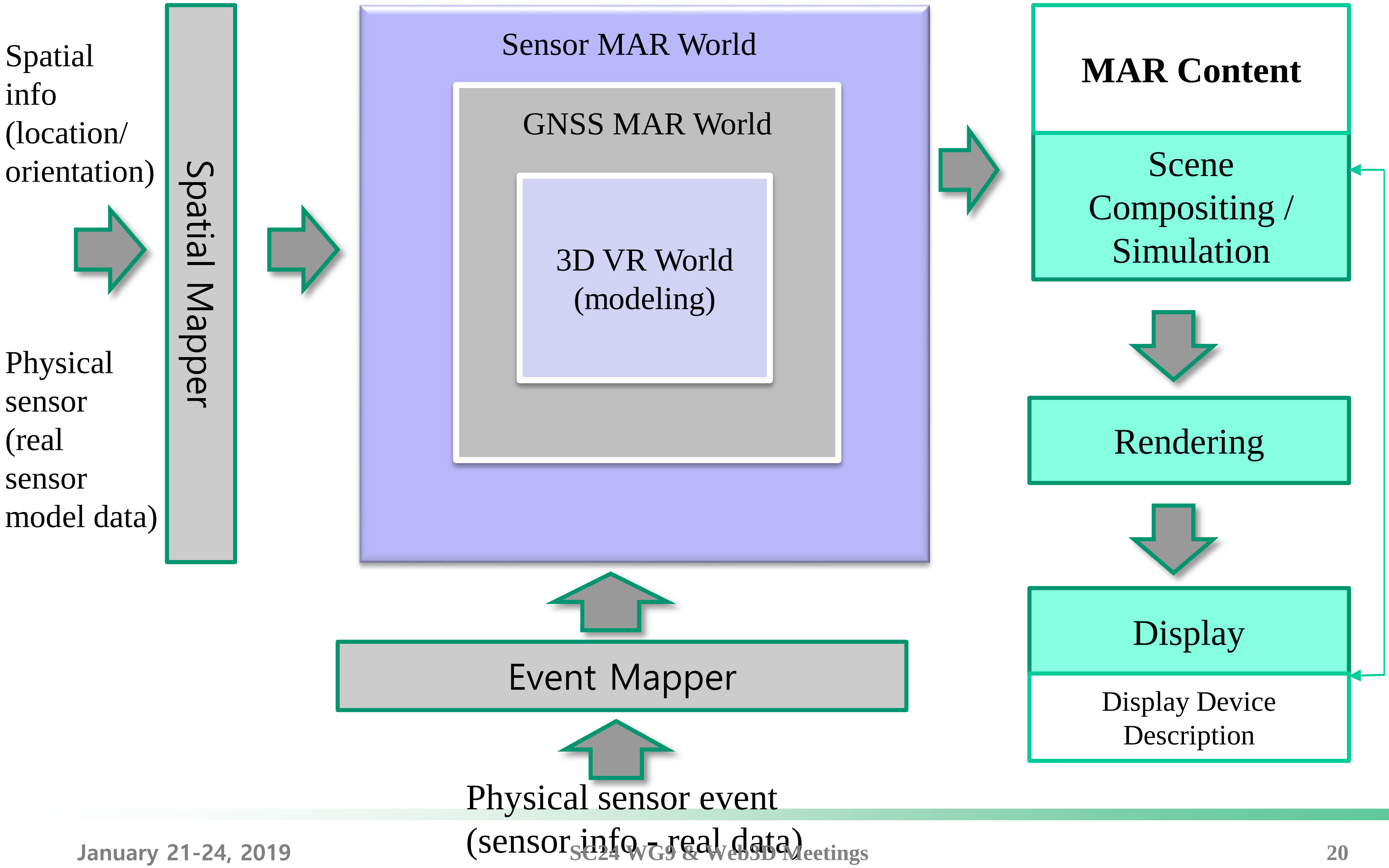
IoT Sensors in VR





Camera sensor
Chemical sensor
Electric sensor
Environment sensor
Flow sensor
Light sensor
Navigation sensor
Pressure sensor
Proximity sensor
Sound sensor
Temperature sensor

Sensor Representation in MAR



Sensor MAR World

3D VR World

3D copied real world
- Shape
- Appearance
- Scaling

GNSS MAR World

GNSS sync with real world
- GPS synchronization
- Exact location and orientation of 3D objects
- Unit specification

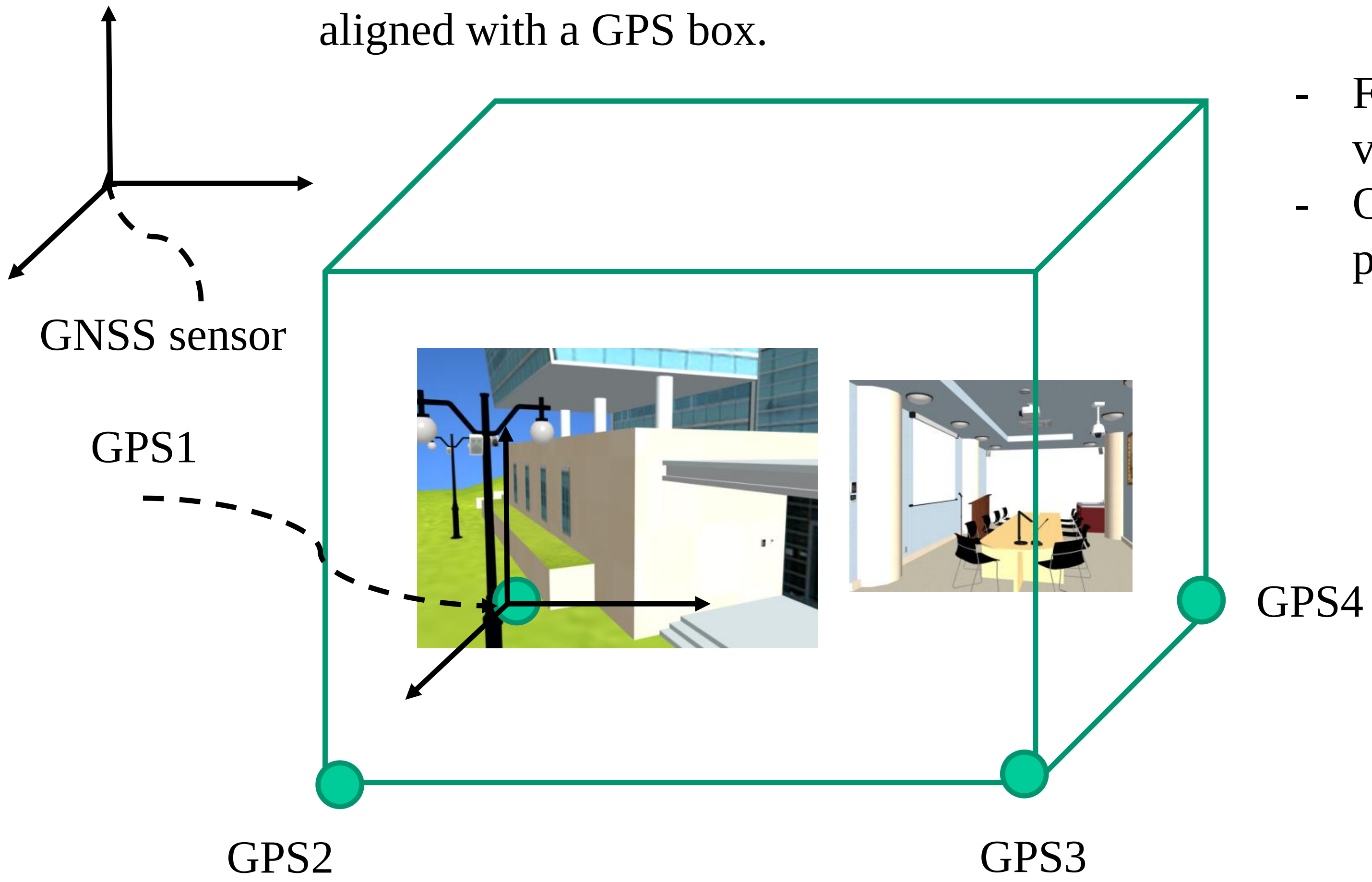
Sensor MAR World

3D copied real world with sensors
- Functional representation of sensors
- Exact location and orientation of sensors

Exact Location and Orientation (1)

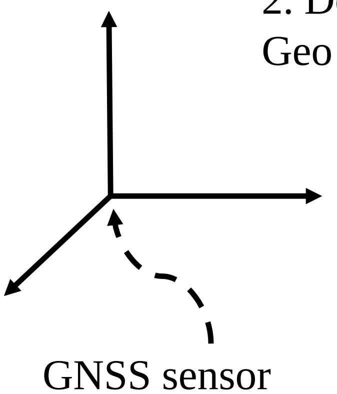
1. Define the Geo coordinate system of a 3D virtual environment aligned with a GPS box.

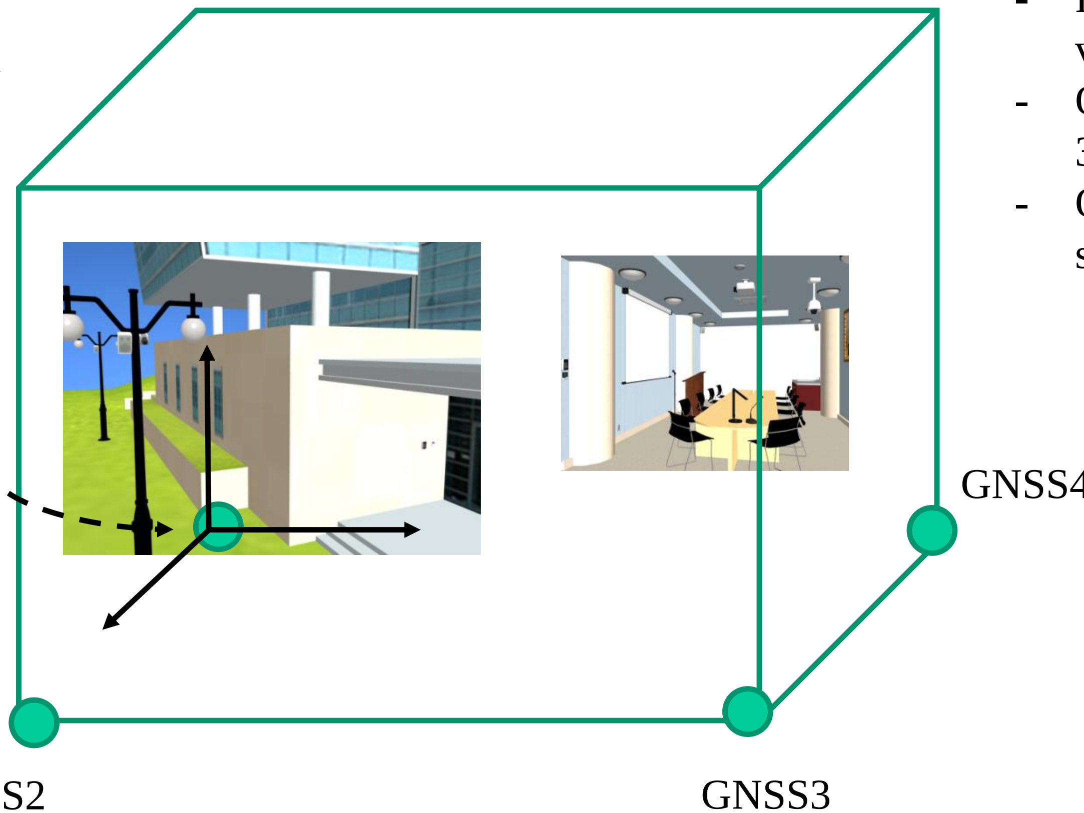
- Four GNSS sensor values
- One Geo origin per 3D scene



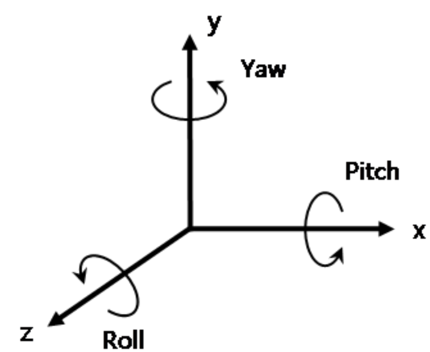
Exact Location and Orientation (2)

2. Define orientation for each sensor in a 3D virtual environment aligned with the Geo coordinate system.





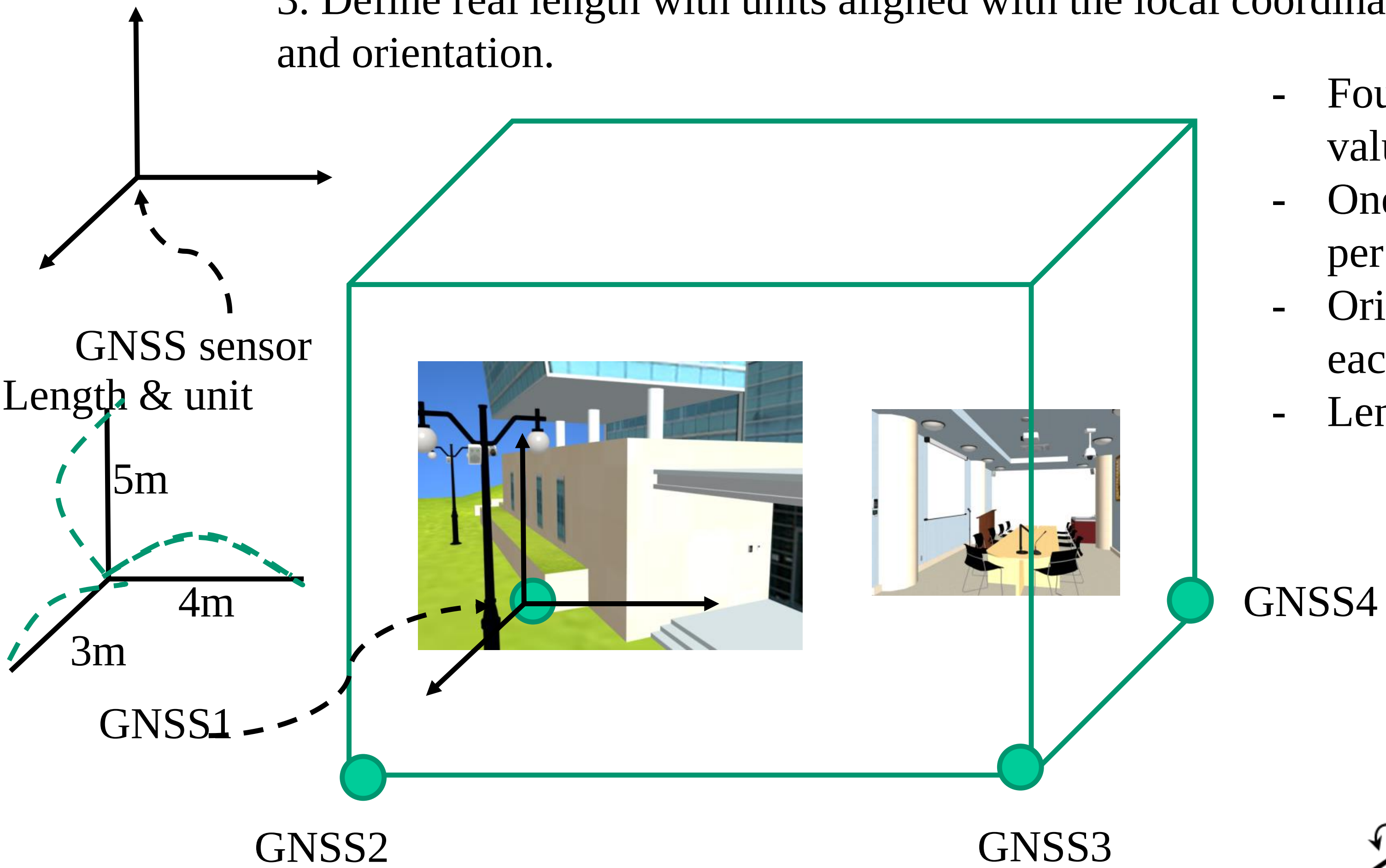
- Four GNSS sensor values
- One Geo origin per 3D scene
- Orientation of each sensor

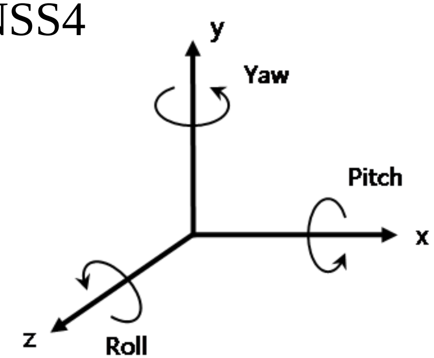


Exact Location and Orientation (3)

3. Define real length with units aligned with the local coordinate system and orientation.

- Four GNSS sensor values
- One Geo origin per 3D scene
- Orientation of each sensor
- Length units





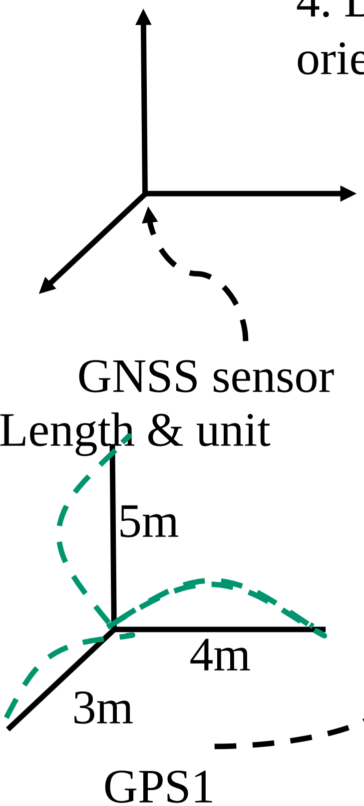
Orientation
- Euler angles, Tait-Bryan angles, Orientation vector

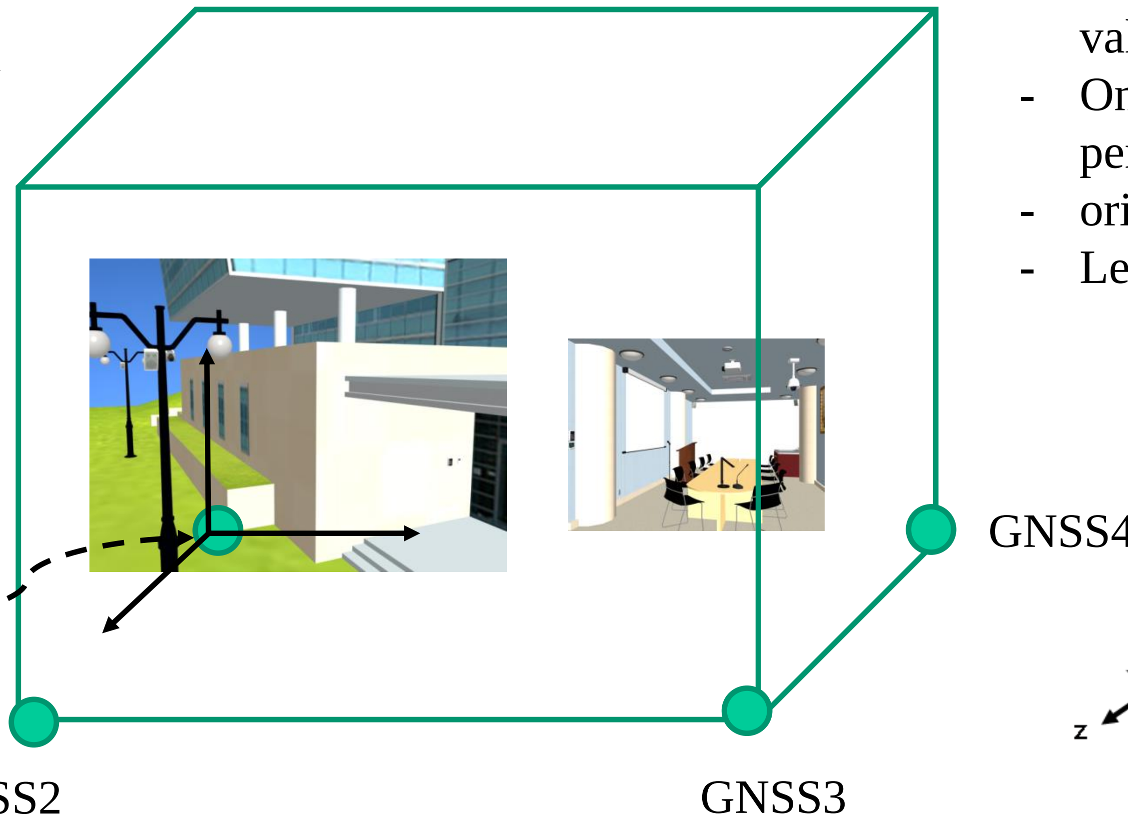
Tait-Bryan angles

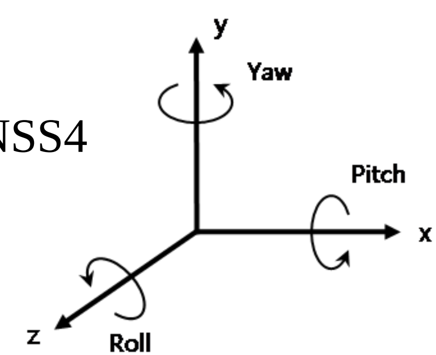
Exact Location and Orientation (4)

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

- Four GNSS sensor values
- One Geo origin per 3D scene
- orientation
- Length units







Local coordinate system → Global coordinate system → Geo based coordinate system → Geo based coordinate system with length unit → Geo based coordinate system Orientation

Sensor Types and Properties

Sensors

- Acoustic, sound, vibration
- Automotive, transportation
- Chemical
- Electric current, electric potential, magnetic, radio
- Environment, weather, moisture, humidity
- Flow, fluid velocity
- Ionizing radiation, subatomic particles
- Navigation instruments
- Position, angle, displacement, distance, speed, acceleration
- Optical, light, imaging, photon
- Pressure
- Force, density, level
- Thermal, heat, temperature
- Proximity, presence

(Wikipedia, 2015-7-6)

Goals

- **Represent, manage, and control physical sensors in 3D scenes.**
- **Define an abstract model for functional simulation of physical sensors in 3D scenes.**

Sensor Types (1)

- Camera Sensor
  - Integrates real world camera and images into a 3D world
  - All camera and image detectors including CCTV and CCD sensors
- Chemical Sensor
  - Integrates real world chemical detection devices into a 3D world
  - All chemical detection devices such as smoke detectors
- Electric and electronic sensors
  - Electric current, electric potential, magnetic, radio
  - Integrates real world electric and electronic signals into a 3D world
  - All electric devices such as electricity and voltage detectors
- Environment sensors, weather sensors
  - Environment, weather, moisture, humidity
  - Integrates real world environmental change in weather and humidity in a 3D world
  - All environmental detectors such as temperature and humidity detectors

Sensor Types (2)

- Flow sensors, fluid sensors
  - Flow, fluid velocity
  - Integrates real world flow in air and fluid into a 3D world
  - All flow detectors such as air flow and fluid sensors
- Force sensors, density sensors, level sensors
  - Force, density, level
  - Integrates real world force, density, and level measurements into a 3D world
  - All force sensors, force transducers, liquid and gas density and level measurement sensors
- Light sensors
  - Optical, light, photon
  - Integrates real world optical, light, and photon measurements into a 3D world
  - All optical, light, and photon detectors
- Navigation sensors
  - Navigation instruments
  - Integrates real world navigation into a 3D world
  - All navigation sensors to detect current position, orientation, and other navigation information

Sensor Types (3)

- Movement sensors
  - Position sensors, angle sensors, displacement sensors, distance sensors, speed sensors, acceleration sensors
  - Integrates real world measurements about position, angle, displacement, distance, speed and acceleration into a 3D world
  - All measurement sensors
- Particle sensors, ion sensors
  - Ionizing radiation, subatomic particles
  - Integrates real world ionizing radiation and subatomic particles into a 3D world
  - All ionizing radiation and subatomic particle detectors
- Position sensors
  - Location sensors
  - Automotive, transportation
  - Integrates real world moving objects into a 3D world
  - Movable devices

Sensor Types (4)

- Pressure sensors
  - Represents sensing real world pressure measurement into a 3D world
  - All pressure sensors and pressure transducers
- Proximity sensors, presence sensors
  - Integrates real world proximity and presence measurements into a 3D world
  - All proximity detectors and presence detectors
- Sound sensors
  - Acoustic, sound, vibration
  - Integrates real world sound into a 3D world
  - Speakers, microphones, hearing aids, vibrating instrumentation aids, ..
- Temperature sensors
  - Thermal, heat, temperature
  - Integrates real world thermal, heat, and temperature measurements into a 3D world
  - All heat detectors and temperature sensors
- Other sensors

Sensor Properties and Interface

Sensor Properties and Interface

- Concepts for defining sensor properties in 3D scenes
  - How to represent and recognize a sensor in a 3D scene (Physical properties)
  - Represent physical sensor properties in a simple form so that any type of sensor can be applied (Physical properties)
  - How to represent streaming information between a physical sensor and a 3D scene (Physical interface)
  - How to connect a physical sensor to its corresponding virtual sensor in a 3D scene (Physical interface)
- Minimum specification for properties and interface in order to
  - Provide a unified interface so that any sensor can be attached to 3D scenes

Sensor Properties and Interface

- Sensor properties
  - Physical properties for interacting with real worlds
    - Global identifier to recognize a sensor with its type in the real world
    - Data type for importing physical sensor information to 3D scenes
    - Data type for exporting internal sensor information to physical sensors
  - Graphical properties for virtual sensors in 3D scenes
- Physical interface
  - Connection information for a physical sensor in real worlds
  - IP address
  - Protocol
  - Port
  - ID/Passwd

Sensor 3D Scene Graph

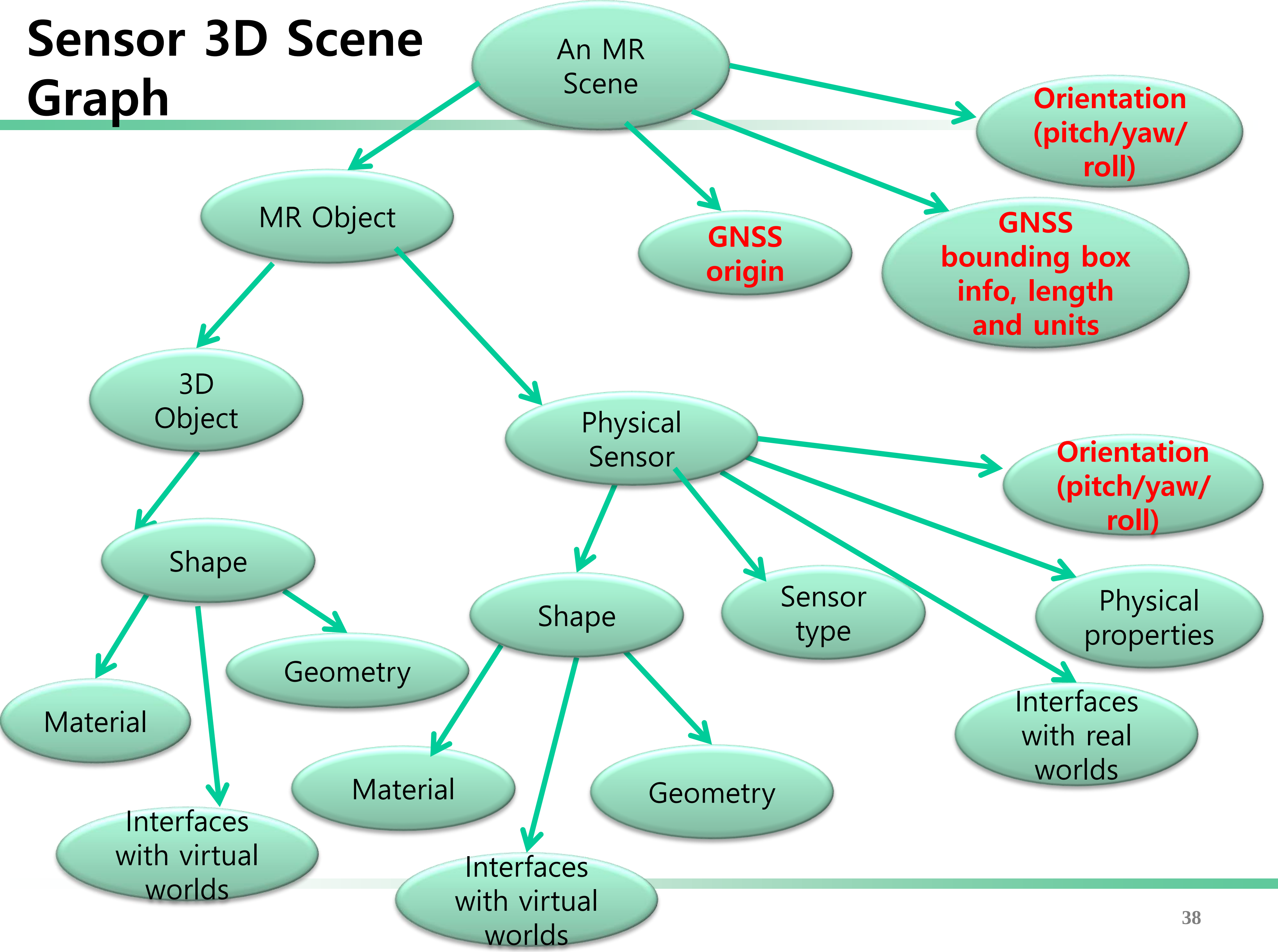
Organization of a Sensor MAR world

- 3D world representation
  - Define a 3D world including real sensor information processing and visualization
- GNSS synchronized 3D worlds
  - Define the data model of a GNSS synchronized 3D world, augmented with real location and orientation information
- Physical sensor devices and properties
  - Define the data model of representing sensor devices and their functional proerties in 3D worlds.
- Interfaces with physical sensor devices
  - Define interfaces for sensor information processing using sensor stream data
  - Define connection or access information to physical sensor devices

Sensor 3D Scene Structure (extended)

-An MAR scene
------- Location (GNSS origin)
------- Location Bounding Box (Ground four GNSS position) with length and units
------- Orientation (pitch-yaw-roll)
------- MR object
--------------------3D object
-----------------------------Shape
------------------------------------materials
------------------------------------geometry
------------------------------------interfaces with virtual worlds
-------------------Physical sensor (abstract type)
-----------------------------Shape
------------------------------------materials
------------------------------------geometry
------------------------------------interfaces with virtual worlds
-----------------------------Physical sensor type
-----------------------------Physical properties
-----------------------------Physical interface
-----------------------------**Orientation (pitch, yaw, roll)**

Sensor 3D Scene Graph



Physical Properties of a Physical Sensor

Device

- UUID
- NAME
- EVENT_TYPE
- CONTROL_TYPE
- DESC

Example

```
<UUID>111-111-111</UUID>
<NAME>Air conditioner</NAME>
<EVENT_TYPE>TEMP</EVENT_TYPE>
<CONTROL_TYPE>TEMP</CONTROL_TYPE>
<DESC>sensor type</DESC>
```

Physical Sensor Device Properties

Sensor Device Fields	Properties
UUID	Unique ID for recognizing a device (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 and its sensor type

Physical Interface of a Physical Sensor (connection information)

Connection

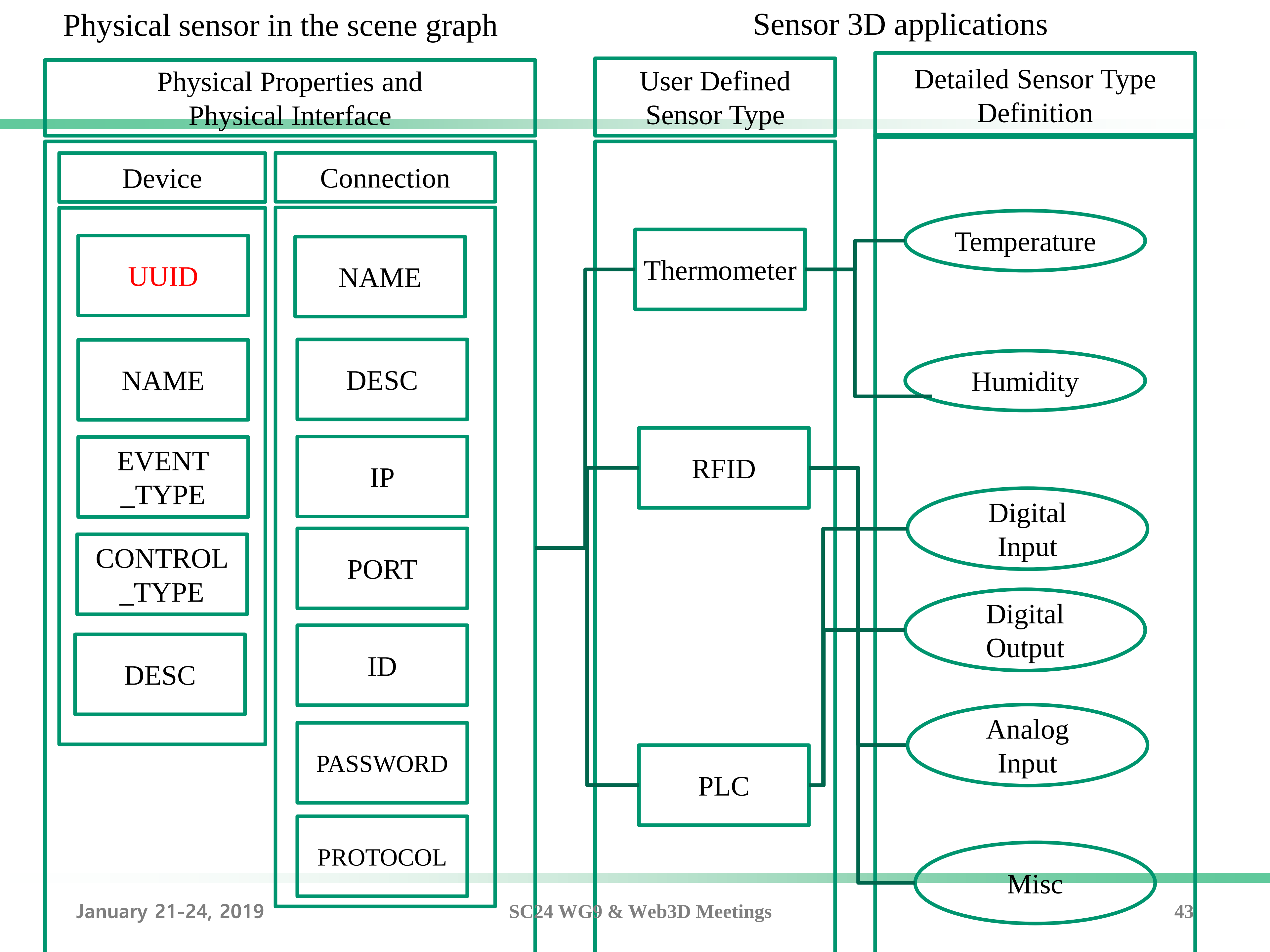
- NAME
- DESC
- IP
- PORT
- ID
- PASSWORD
- PROTOCOL

Example

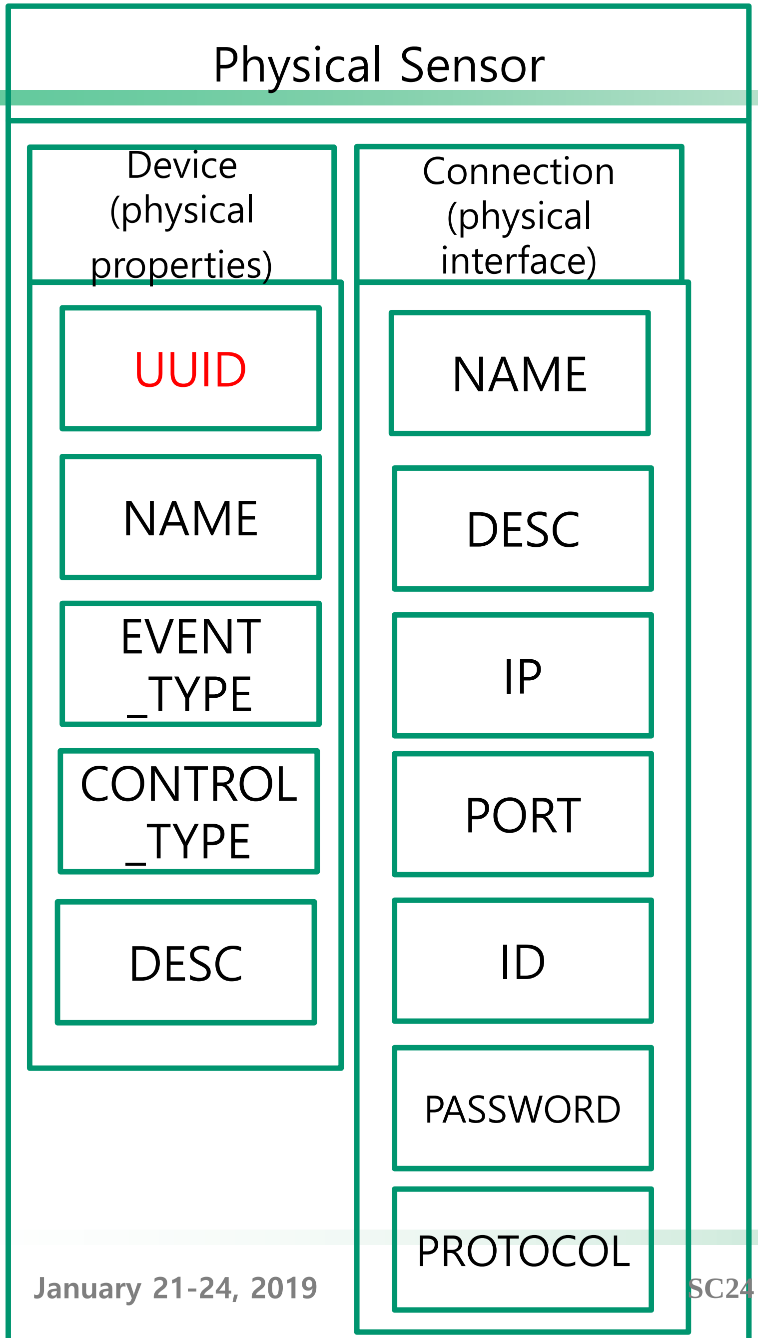
```
<NAME>Air conditioner manager</NAME>
<DESC>Connection info</DESC>
<IP>1.1.1.1</IP>
<PORT>8080</PORT>
<ID>user1</ID>
<PASSWORD>pass1</PASSWORD>
<PROTOCOL>TCP</PROTOCOL>
```

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 in the scene graph
Sensor 3D applications
Physical Properties and Physical Interface
User Defined Sensor Type
Detailed Sensor Type Definition
Device
Connection
UUID
NAME
EVENT _TYPE
CONTROL _TYPE
DESC
NAME
DESC
IP
PORT
ID
PASSWORD
PROTOCOL
Thermometer
RFID
PLC
Temperature
Humidity
Digital Input
Digital Output
Analog Input
Misc

Data Structure of Physical Properties and interfaces of a Physical Sensor Device

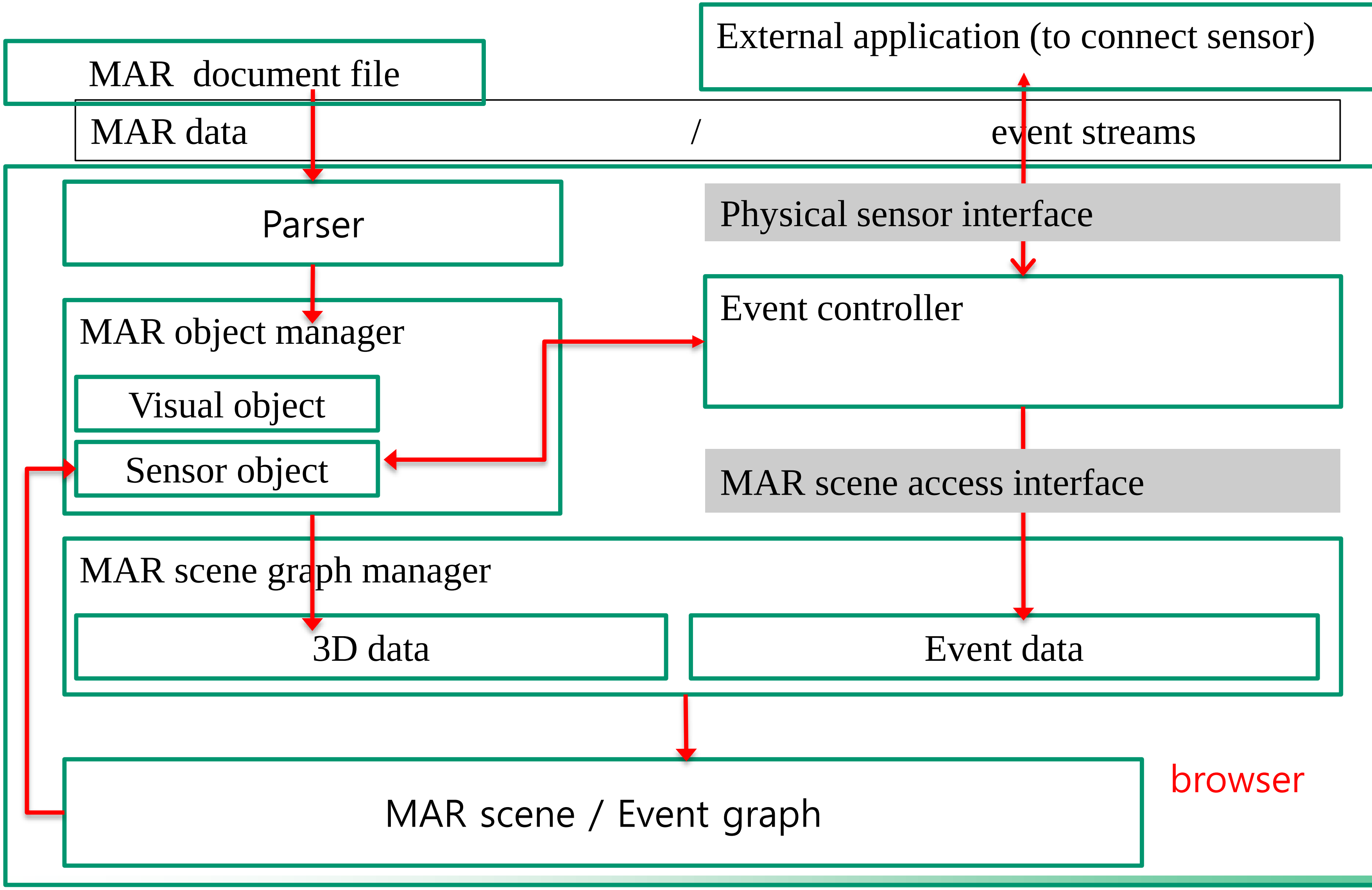


<Physical Properties of a Physical Sensor>
<Device>
<UUID>
<Name>
<EventType>
<ControlType>
<Desc>

<Physical Interface of a Physical Sensor>
<Connection>
<Name>
<Desc>
<IP>
<Port>
<ID>
<Password>
<Protocol>

Architecture of Sensor 3D

Architecture



Functional description of the architecture (1)

- MAR data read from an MAR document file is classified as MAR objects through a parser in a MAR system
- The MAR objects consist of visual objects and sensor objects that are controlled by a MAR object manager
- The MAR object manager manipulates the structures and the properties of visual and sensor objects
- Sensor events coming from external sensor devices are transferred to an event controller via a physical sensor interface
- The event controller transfers the events to sensor objects managed by the MAR object manager to be represented in a 3D scene
- An MAR scene access interface organizes event data so that sensor events transferred from the event controller can be represented in a 3D scene
- In the architecture slide, two arrows from the event controller denote the two cases where an event affects a visual object, and that the event is represented irrespective of a visual object

Functional description of the architecture (2)

- A MAR scene graph manager builds a scene graph that is classified with geometry, properties, and events so that a 3D scene can be organized using visual objects, sensor objects, and sensor events
- When the 3D scene is changed by simulation or a user interface, the program of the external sensor device can be affected by the event controller with the information sent by a sensor object

XML Definition of Sensor Representation

Components of a sensor MR scene (1)

MR → MAR

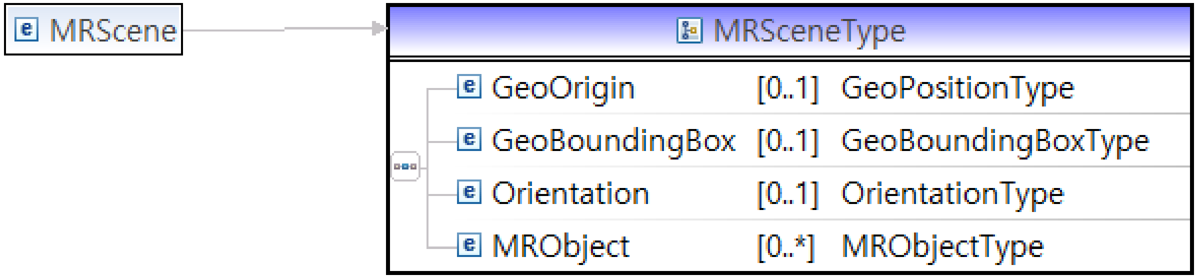
- Components of a mixed reality scene with physical sensors
  - MR Scene
    - An MR Scene consists of a set of MR objects.
    - Also has location and orientation information
  - MR Object
    - An MR Object consists of a 3D object and a physical sensor.
    - Physical sensor is optional.
  - 3D Object
    - A 3D Object has a shape.
  - Shape
    - A Shape consists of Material, Geometry, and Interface with virtual worlds.
  - Physical Sensor
    - A Physical Sensor consists of Abstract Sensor Type, Physical Properties, Physical Interface, and orientation.

Components of a sensor MR scene (2)

- Sensor Type
  - An abstract super-type of all sensor types includes common properties of physical sensors
- Physical Properties
  - Include a Device element which contains child elements including UUID, Name, EventType, ControlType, and Desc
- Physical Interface
  - Includes a Connection element which contains child elements including Name, Desc, IP, Port, ID, Password, Protocol

Basic Schema

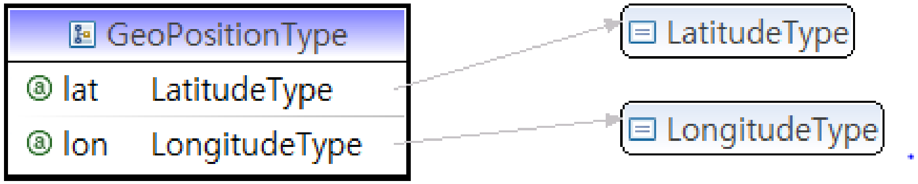
- MRSceneType
  – Consists of location (origin, bounding box), orientation, and a set of MRObjects




```
<complexType name="MRSceneType">
  <sequence>
    <element name="GeoOrigin" type="mar:GeoPositionType" minOccurs="0" />
    <element name="GeoBoundingBox" type="mar:GeoBoundingBoxType"
         minOccurs="0"/>
    <element name="Orientation" type="mar:OrientationType" minOccurs="0" />
    <element name="MRObject" type="mar:MRObjectType" minOccurs="0"
             maxOccurs="unbounded" />
  </sequence>
</complexType>
```

Basic Schema

- GeoPositionType
  - Represents a GNSS position consisting of latitude and longitude values




```xml
<complexType name="GeoPositionType">
    <annotation>
        <documentation> The latitude and longitude of a geo-position
        </documentation>
    </annotation>
    <attribute type="mar:LatitudeType" name="lat" use="required" />
    <attribute type="mar:LongitudeType" name="lon" use="required" />
</complexType>
```

Basic Schema

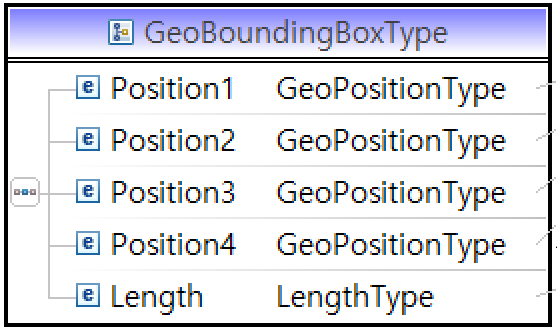
- Latitude & Longitude Type

```
<simpleType name="LatitudeType">
   <annotation>
      <documentation> The latitude is in a decimal degree and in a WGS84 datum. </documentation>
   </annotation>
   <restriction base="decimal">
      <minInclusive value="-90.0" />
      <maxInclusive value="90.0" />
   </restriction>
</simpleType>

<simpleType name="LongitudeType">
   <restriction base="decimal">
      <minInclusive value="-180.0" />
      <maxInclusive value="180.0" />
   </restriction>
</simpleType>
```

Basic Schema

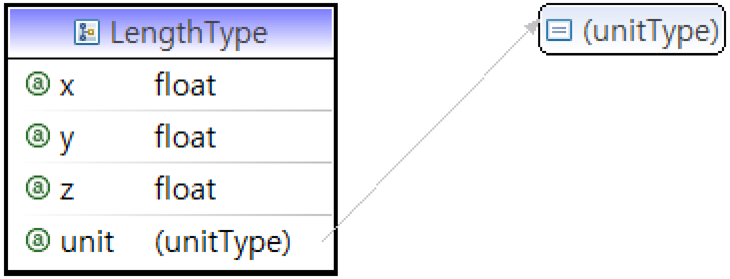
- GeoBoundingBoxType
  - defined by 4 geo-positions and 3 length values with a common unit




```xml
<complexType name="GeoBoundingBoxType">
  <annotation>
     <documentation> A bounding box info defined by 4 geo-positions and
3 lengths with a unit </documentation>
  </annotation>
  <sequence>
     <element name="Position1" type="mar:GeoPositionType" />
     <element name="Position2" type="mar:GeoPositionType" />
     <element name="Position3" type="mar:GeoPositionType" />
     <element name="Position4" type="mar:GeoPositionType" />
     <element name="Length" type="mar:LengthType" />
  </sequence>
</complexType>
```

Basic Schema

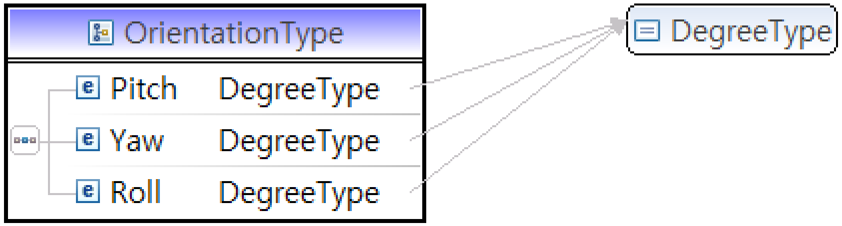
- LengthType
  - defined by 3 length values with a unit




```xml
<complexType name="LengthType">
   <attribute name="x" type="float" use="required" />
   <attribute name="y" type="float" use="required" />
   <attribute name="z" type="float" use="required" />
   <attribute name="unit" use="optional" default="m">
      <simpleType>
         <restriction base="string">
            <enumeration value="pm" /><enumeration value="nm" />
            <enumeration value="um" /><enumeration value="mm" />
            <enumeration value="cm" />…
         </restriction>
      </simpleType>
   </attribute> </complexType>
```

Basic Schema

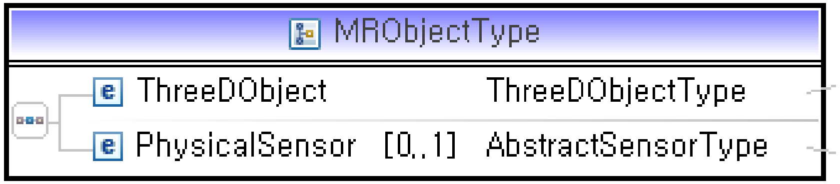
- OrientationType
  - defined by 3 decimal degrees (pitch, yaw, roll)




```xml
<complexType name="OrientationType">
   <sequence>
      <element name="Pitch" type="mar:DegreeType" />
      <element name="Yaw" type="mar:DegreeType" />
      <element name="Roll" type="mar:DegreeType" />
   </sequence>
</complexType>
<simpleType name="DegreeType">
   <restriction base="decimal">
      <minInclusive value="0.0" />
      <maxInclusive value="360.0" />
   </restriction>
</simpleType>
```

Basic Schema

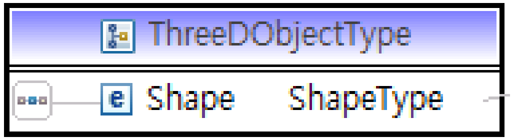
- MRObjectType
  - Consists of ThreeDObject and PhysicalSensor
  - PhysicalSensor can be omitted




```
<complexType name="MRObjectType">
    <sequence>
        <element name="ThreeDObject" type="mar:ThreeDObjectType" />
        <element name="PhysicalSensor" type="mar:AbstractSensorType"
            minOccurs="0" />
    </sequence>
</complexType>
```

Basic Schema

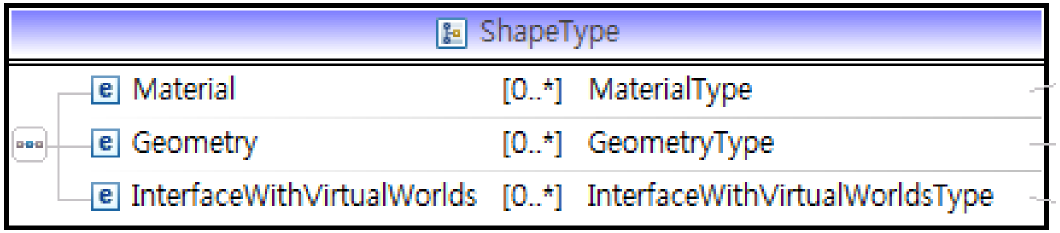
- ThreeDObjectType
  - Has a Shape element




```
<complexType name="ThreeDObjectType">
    <sequence>
        <element name="Shape" type="mar:ShapeType" />
    </sequence>
</complexType>
```

Basic Schema

- ShapeType
  - Consists of Material, Geometry, and InterfaceWithVirtualWorlds elements




```
<complexType name="ShapeType">
   <sequence>
      <element name="Material" type="mar:MaterialType"
               maxOccurs="unbounded" minOccurs="0" />
      <element name="Geometry" type="mar:GeometryType"
               maxOccurs="unbounded" minOccurs="0" />
      <element name="InterfaceWithVirtualWorlds"
               type="mar:InterfaceWithVirtualWorldsType"
               maxOccurs="unbounded" minOccurs="0" />
   </sequence>
</complexType>
```

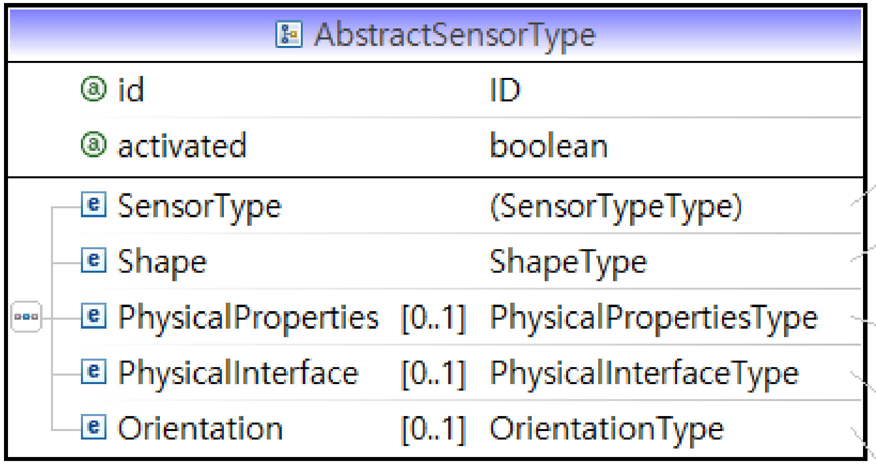
Basic Schema

- IntWithUnit, IntRangeWithUnit, FloatWithUnit, FloatRangeWithUnit

```
<complexType name="IntWithUnit">
   <simpleContent>
      <extension base="integer">
         <attribute name="unit" type="string" />
      </extension>
   </simpleContent>
</complexType>
<complexType name="IntRangeWithUnit">
   <attribute name="min" type="integer" /><attribute name="max" type="integer" />
   <attribute name="value" type="integer" />
   <attribute name="unit" type="string" />
</complexType>
<complexType name="FloatWithUnit">
   <simpleContent>
      <extension base="float">
         <attribute name="unit" type="string" />
      </extension>
   </simpleContent>
</complexType>
<complexType name="FloatRangeWithUnit">
   <attribute name="min" type="float" /><attribute name="max" type="float" />
   <attribute name="value" type="float" />
   <attribute name="unit" type="string" />
</complexType>
```

Abstract Sensor Type

- AbstractSensorType
  - An abstract super-type of all sensor types, which includes common properties of physical sensors
    - id: sensor identification
    - activated: activated status of a physical sensor
    - SensorType: {Type1, …, Type15}
    - Shape: *ShapeType*
    - PhysicalProperties: common properties
    - PhysicalInterface: interface with real world
    - Orientation: pitch, yaw, roll
  - Other physical properties are defined in each sensor type.



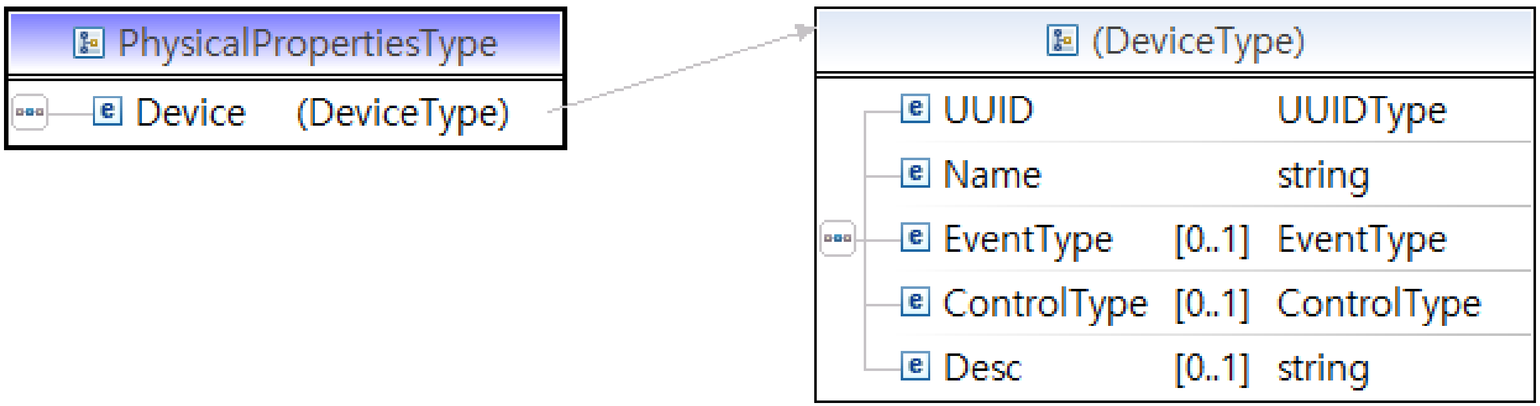
Abstract Sensor Type

- AbstractSensorType

```
<complexType name="AbstractSensorType">
  <sequence>
     <element name="SensorType">
        <simpleType>
           <restriction base="string">
              <enumeration value="Camera" /><enumeration value="Chemical" />
              <enumeration value="Electric" /><enumeration value="Environment" />
              <enumeration value="Flow" /><enumeration value="Force" />
              <enumeration value="Light" /><enumeration value="Movement" />
              <enumeration value="Navigation" /><enumeration value="Particle" />
              <enumeration value="Position" /><enumeration value="Pressure" />
              <enumeration value="Proximity" /><enumeration value="Sound" />
              <enumeration value="Temperature" /><enumeration value="Others" />
           </restriction>
        </simpleType>
     </element>
     <element name="Shape" type="mar:ShapeType" />
     <element name="PhysicalProperties" type="mar:PhysicalPropertiesType"
           minOccurs="0" />
     <element name="PhysicalInterface" type="mar:PhysicalInterfaceType" minOccurs="0"/>
     <element name="Orientation" type="mar:OrientationType" minOccurs="0" />
  </sequence>
  <attribute name="id" type="ID" />
  <attribute name="activated" type="boolean" />
</complexType>
```

Physical Properties

- PhysicalPropertiesType
  - Has a Device element
  - Device element contains child elements including
    - UUID
    - Name
    - EventType
    - ControlType
    - Desc



Physical Properties

- PhysicalPropertiesType

```
<complexType name="PhysicalPropertiesType">
  <sequence>
    <element name="Device">
      <complexType>
        <sequence>
          <element name="UUID" type="mar:UUIDType" />
          <element name="Name" type="string" />
          <element name="EventType" type="mar:EventType" minOccurs="0"/>
          <element name="ControlType" type="mar:ControlType"
             minOccurs="0"/>
          <element name="Desc" type="string" minOccurs="0" />
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
```

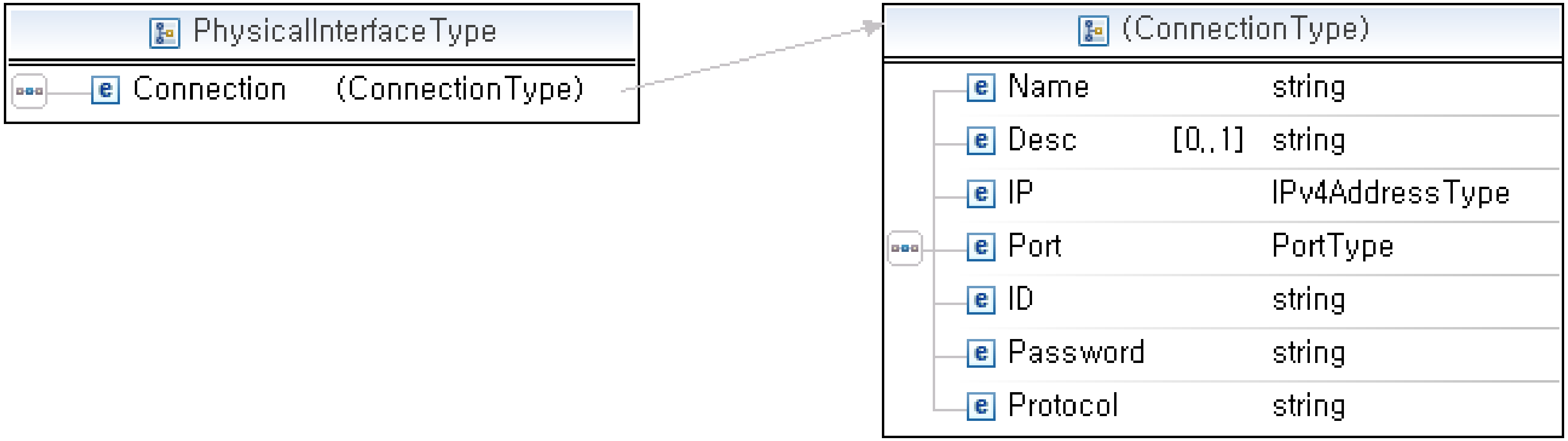
Physical Properties

- UUIDType, EventType, ControlType

```xml
<simpleType name="UUIDType">
  <restriction base="string">
    <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}" />
  </restriction>
</simpleType>
<simpleType name="EventType">
  <restriction base="string">
    <enumeration value="Temp" />
    <enumeration value="EventTypeName1" />
    <enumeration value="EventTypeName2" />
    <!-- Other values can be added -->
  </restriction>
</simpleType>
<simpleType name="ControlType">
  <restriction base="string">
    <enumeration value="Temp" />
    <enumeration value="ControlName1" />
    <enumeration value="ControlName2" />
    <!-- Other values can be added -->
  </restriction>
</simpleType>
```

Physical Interface

- PhysicalInterfaceType
  - Has a Connection element
  - Connection element contains child elements including
    - Name
    - Desc
    - IP
    - Port
    - ID, Password
    - Protocol



Physical Interface

- PhysicalInterfaceType

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

Physical Interface

- IPv4AddressType, PortType

```
<simpleType name="IPv4AddressType">
   <restriction base="string">
       <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])" />
   </restriction>
</simpleType>

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

Conformance

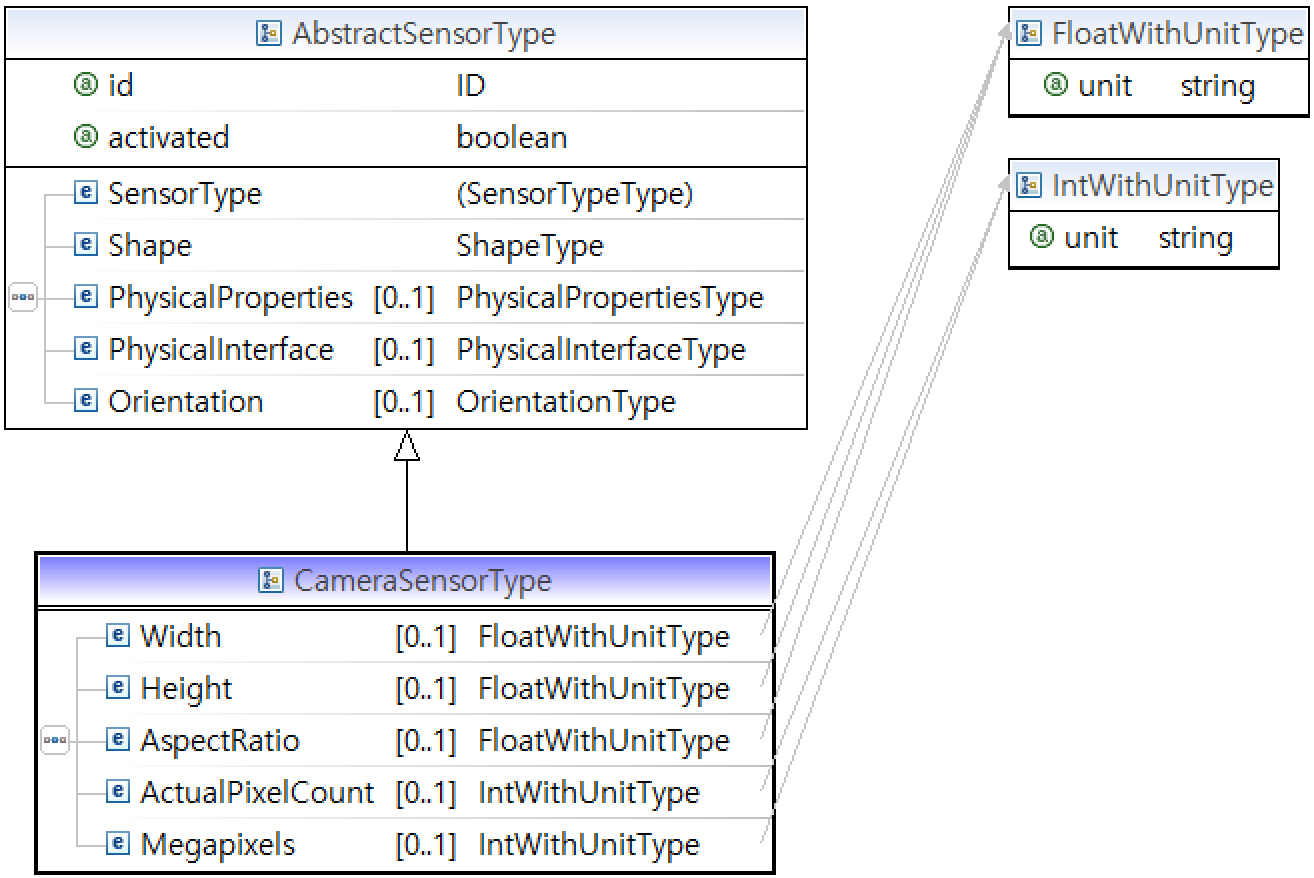
- Sensor data structure and device parameters to be included for sensor representation in a MAR world
- MAR file converters that can generate an exchangeable data file defining an MAR scene from an application
- MAR viewers that can display MAR worlds and MAR simulated worlds
- MAR editors that can generate and update a MAR world visually and interactively

Annex A

Examples for Physical Sensor Types

Type 1: Camera Sensor

- CameraSensorType
  – Includes elements that represent the properties of Width, Height, Aspect Ratio, Aspect Pixel Count, etc.



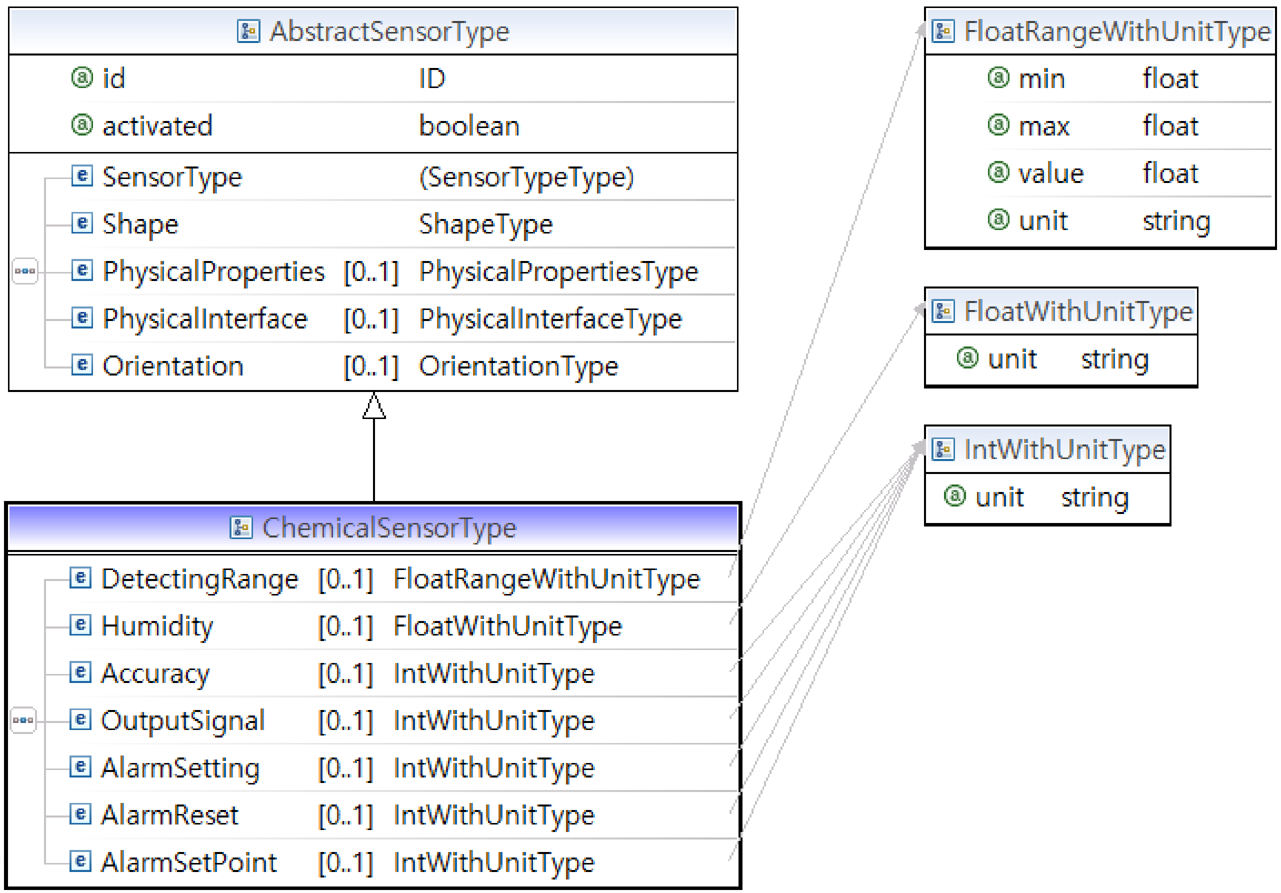
Type 1: Camera Sensor

- CameraSensorType

```xml
<complexType name="CameraSensorType">
   <complexContent>
      <extension base="mar:AbstractSensorType">
         <sequence>
            <element name="Width" type="mar:FloatWithUnit" minOccurs="0" />
            <element name="Height" type="mar:FloatWithUnit" minOccurs="0" />
            <element name="AspectRatio" type="mar:FloatWithUnit"
               minOccurs="0" />
            <element name="ActualPixelCount" type="mar:IntWithUnit"
               minOccurs="0" />
            <element name="Megapixels" type="mar:IntWithUnit" minOccurs="0" />
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

Type 2: Chemical Sensor

- ChemicalSensorType
  - Includes elements that represent the peoperties of Detecting Range, Humidity, Accuracy, Output Signal, Alarm Setting, etc.



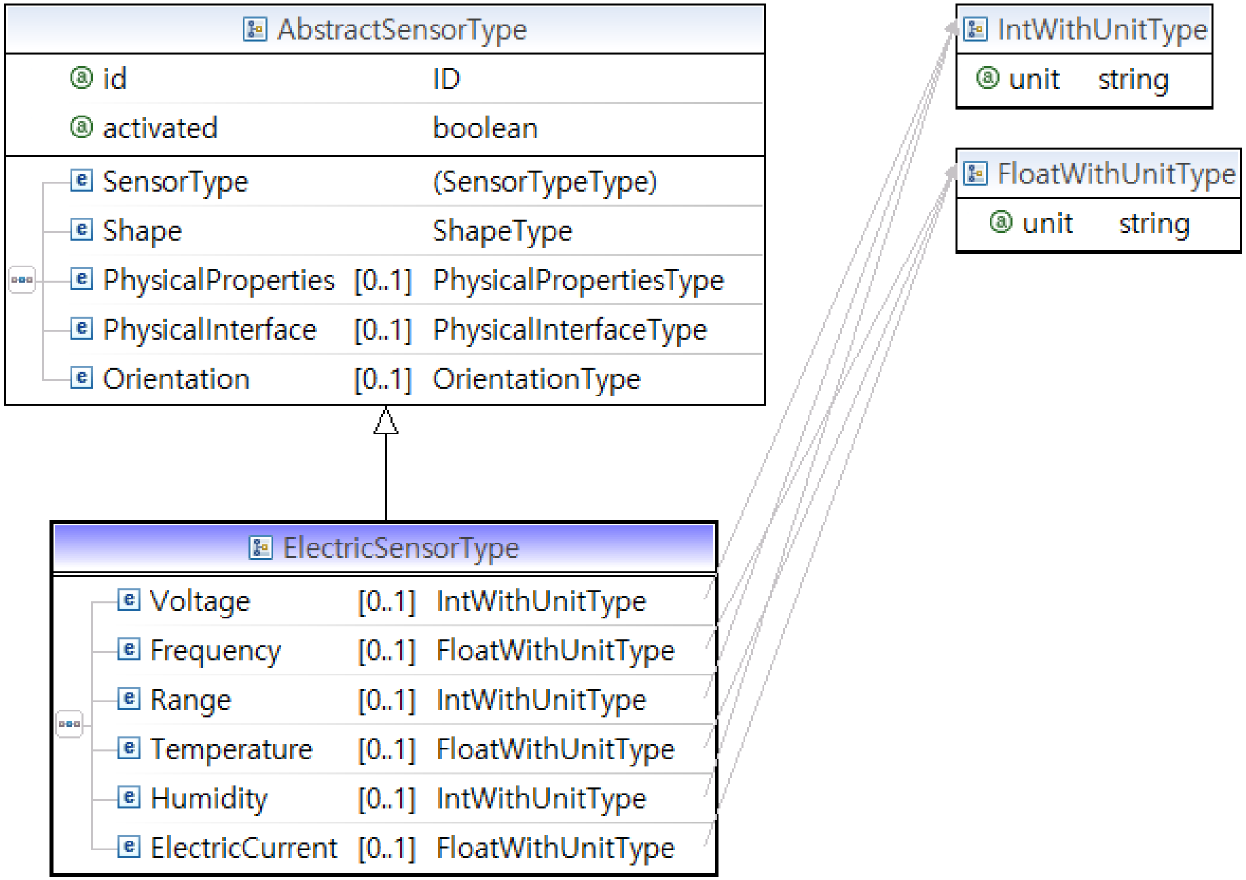
Type 2: Chemical Sensor

- ChemicalSensorType

```
<complexType name="ChemicalSensorType">
   <complexContent>
      <extension base="mar:AbstractSensorType">
         <sequence>
            <element name="DetectingRange" type="mar:FloatRangeWithUnit"
            minOccurs="0" />
            <element name="Humidity" type="mar:FloatWithUnit" minOccurs="0" />
            <element name="Accuracy" type="mar:IntWithUnit" minOccurs="0" />
            <element name="OutputSignal" type="mar:IntWithUnit" minOccurs="0" />
            <element name="AlarmSetting" type="mar:IntWithUnit" minOccurs="0" />
            <element name="AlarmReset" type="mar:IntWithUnit" minOccurs="0" />
            <element name="AlarmSetPoint" type="mar:IntWithUnit" minOccurs="0" />
      </sequence>
      </extension>
   </complexContent>
</complexType>
```

Type 3: Electric Sensor

- ElectricSensorType
  - Includes elements that represent the properties of Voltage, Frequency, Range, Temperature, Humidity, Electric Current, etc.



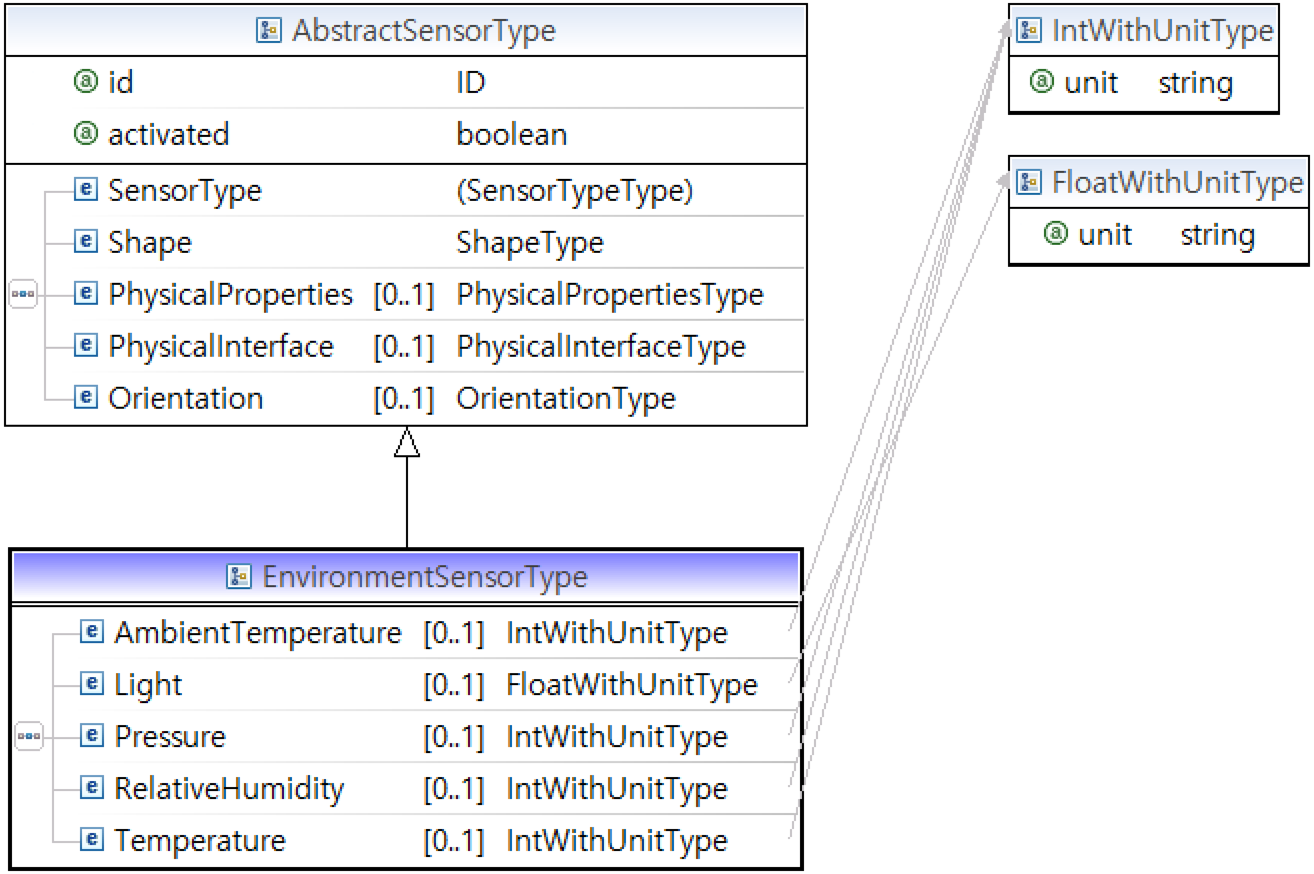
Type 3: Electric Sensor

- ElectricSensorType

```
<complexType name="ElectricSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="Voltage" type="mar:IntWithUnit" minOccurs="0" />
        <element name="Frequency" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="Range" type="mar:IntWithUnit" minOccurs="0" />
        <element name="Temperature" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="Humidity" type="mar:IntWithUnit" minOccurs="0" />
        <element name="ElectricCurrent" type="mar:FloatWithUnit"
          minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 4: Environment Sensor

- EnvironmentSensorType
  - Includes elements that represent the properties of Ambient Temperature, Light, Pressure, Relative Humidity, etc.



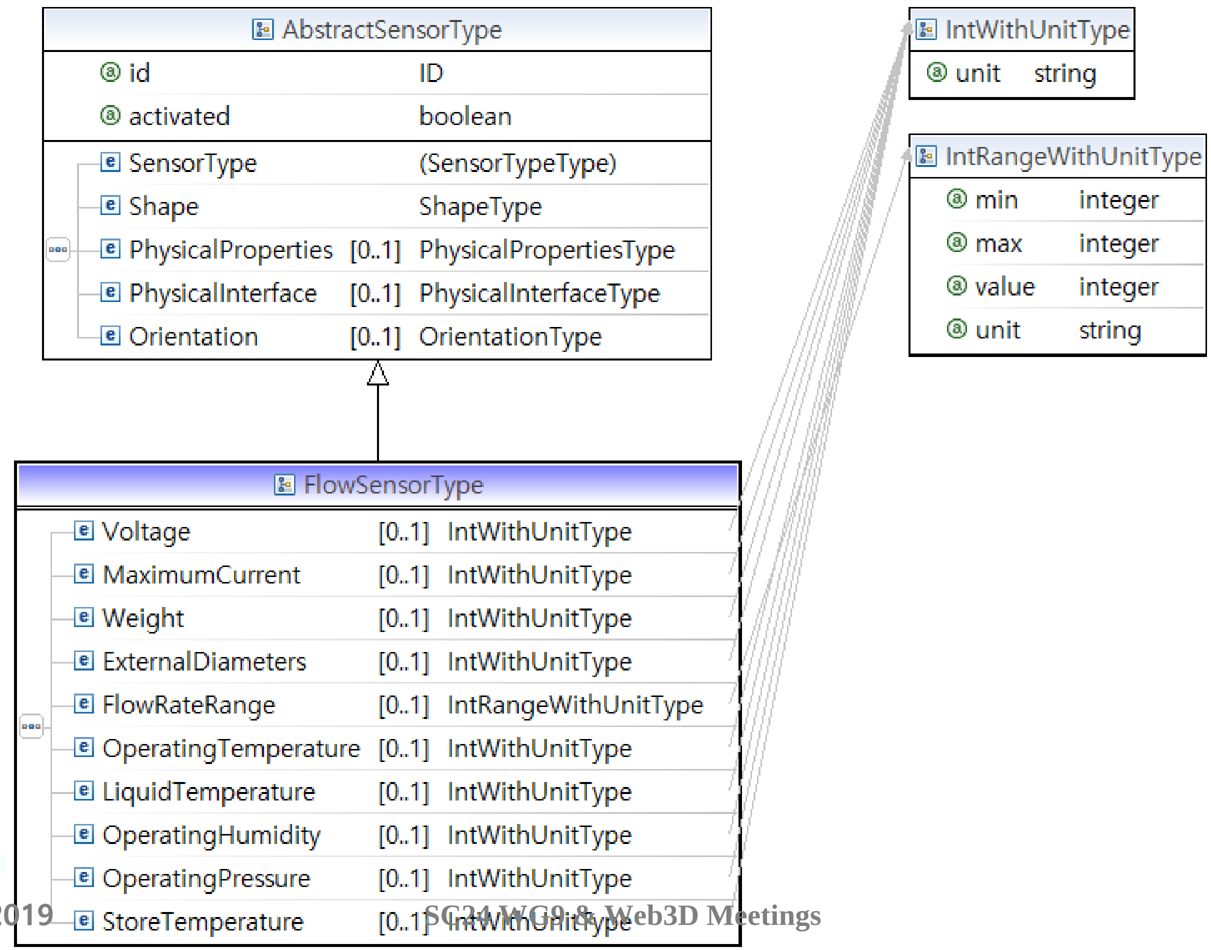
Type 4: Environment Sensor

- EnvironmentSensorType

```
<complexType name="EnvironmentSensorType">
  <complexContent>
     <extension base="mar:AbstractSensorType">
        <sequence>
           <element name="AmbientTemperature" type="mar:IntWithUnit"
              minOccurs="0" />
           <element name="Light" type="mar:FloatWithUnit" minOccurs="0" />
           <element name="Pressure" type="mar:IntWithUnit" minOccurs="0" />
           <element name="RelativeHumidity" type="mar:IntWithUnit"
              minOccurs="0"/>
           <element name="Temperature" type="mar:IntWithUnit" minOccurs="0" />
        </sequence>
     </extension>
  </complexContent>
</complexType>
```

Type 5: Flow Sensor

- FlowSensorType
  - Includes elements that represent the properties of Voltage, Maximum Current, Weight, External Diameters, Flow Rate Range, etc.



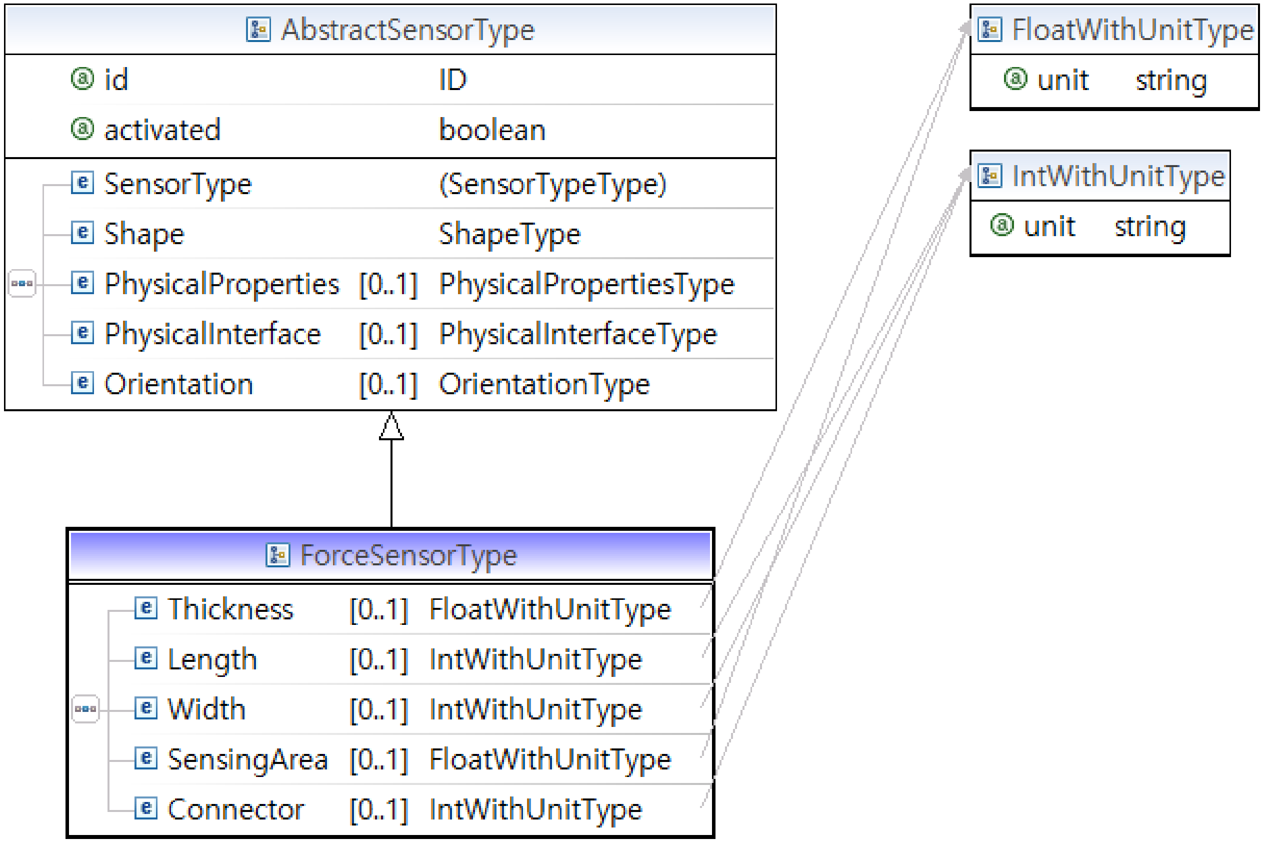
Type 5: Flow Sensor

- FlowSensorType

```
<complexType name="FlowSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
     <sequence>
       <element name="Voltage" type="mar:IntWithUnit" minOccurs="0" />
       <element name="MaximumCurrent" type="mar:IntWithUnit" minOccurs="0" />
       <element name="Weight" type="mar:IntWithUnit" minOccurs="0" />
       <element name="ExternalDiameters" type="mar:IntWithUnit" minOccurs="0" />
       <element name="FlowRateRange" type="mar:IntRangeWithUnit" minOccurs="0" />
       <element name="OperatingTemperature" type="mar:IntWithUnit"
          minOccurs="0"/>
       <element name="LiquidTemperature" type="mar:IntWithUnit" minOccurs="0" />
       <element name="OperatingHumidity" type="mar:IntWithUnit" minOccurs="0" />
       <element name="OperatingPressure" type="mar:IntWithUnit" minOccurs="0" />
       <element name="StoreTemperature" type="mar:IntWithUnit" minOccurs="0" />
     </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 6: Force Sensor

- ForceSensorType
  - Includes elements that represent the peoperties of Thickness, Length, Width, Sensing Area, Connector, etc.



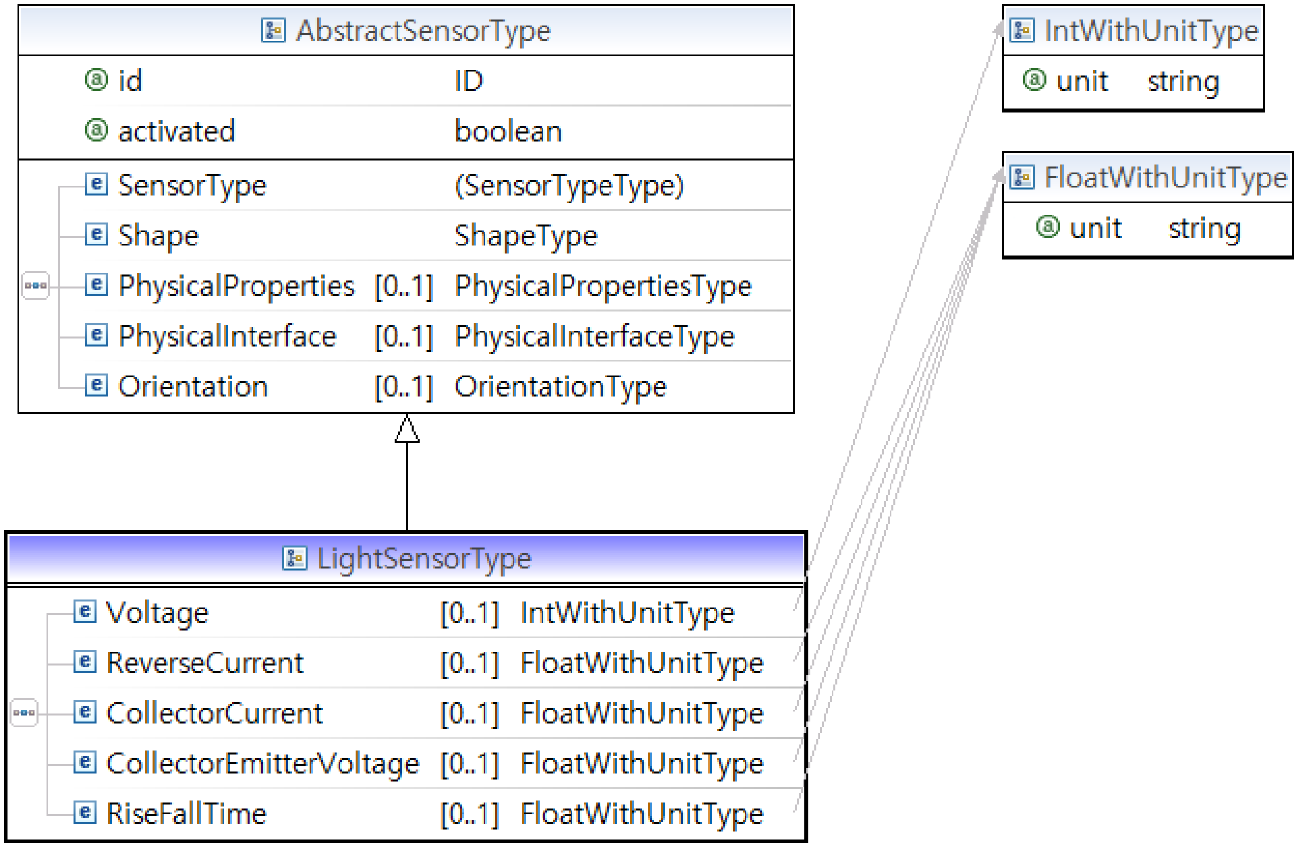
Type 6: Force Sensor

- ForceSensorType

```xml
<complexType name="ForceSensorType">
  <complexContent>
     <extension base="mar:AbstractSensorType">
        <sequence>
           <element name="Thickness" type="mar:FloatWithUnit" minOccurs="0" />
           <element name="Length" type="mar:IntWithUnit" minOccurs="0" />
           <element name="Width" type="mar:IntWithUnit" minOccurs="0" />
           <element name="SensingArea" type="mar:FloatWithUnit" minOccurs="0" />
           <element name="Connector" type="mar:IntWithUnit" minOccurs="0" />
        </sequence>
     </extension>
  </complexContent>
</complexType>
```

Type 7: Light Sensor

- LightSensorType
  - Includes that represent the properties of Voltage, Reverse Current, Collector Current, Collector Emitter Voltage, Rise Fall Time, etc.



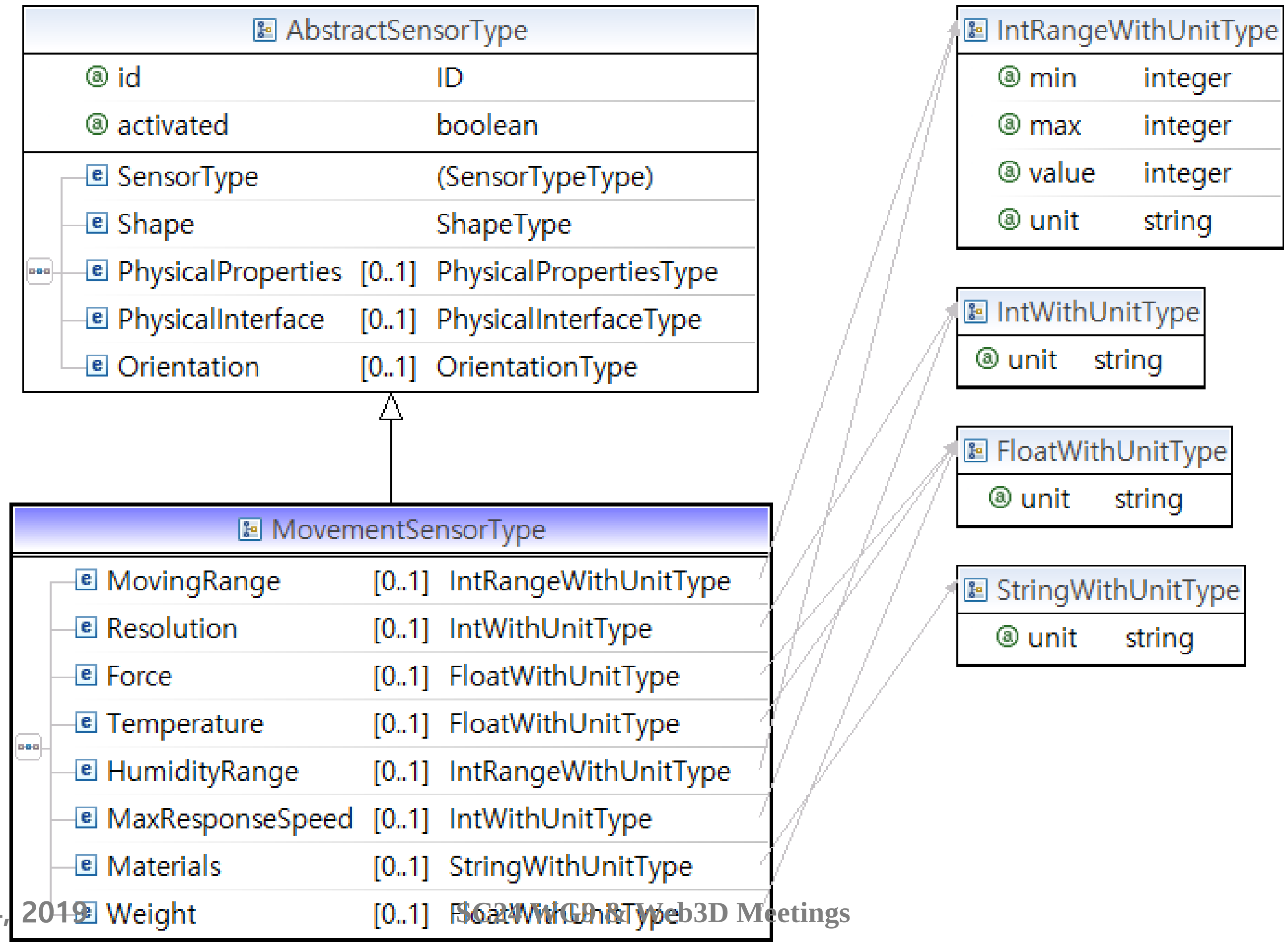
Type 7: Light Sensor

- LightSensorType

```
<complexType name="ForceSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="Voltage" type="mar:IntWithUnit" minOccurs="0" />
        <element name="ReverseCurrent" type="mar:FloatWithUnit" minOccurs="0"/>
        <element name="CollectorCurrent" type="mar:FloatWithUnit"
            minOccurs="0" />
        <element name="CollectorEmitterVoltage" type="mar:FloatWithUnit"
            minOccurs="0" />
        <element name="RiseFallTime" type="mar:FloatWithUnit" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 8: Movement Sensor

- MovementSensorType
  - Includes elements that represent the properties of Moving Range, Resolution, Force, Temperature, HumidityRange, etc.



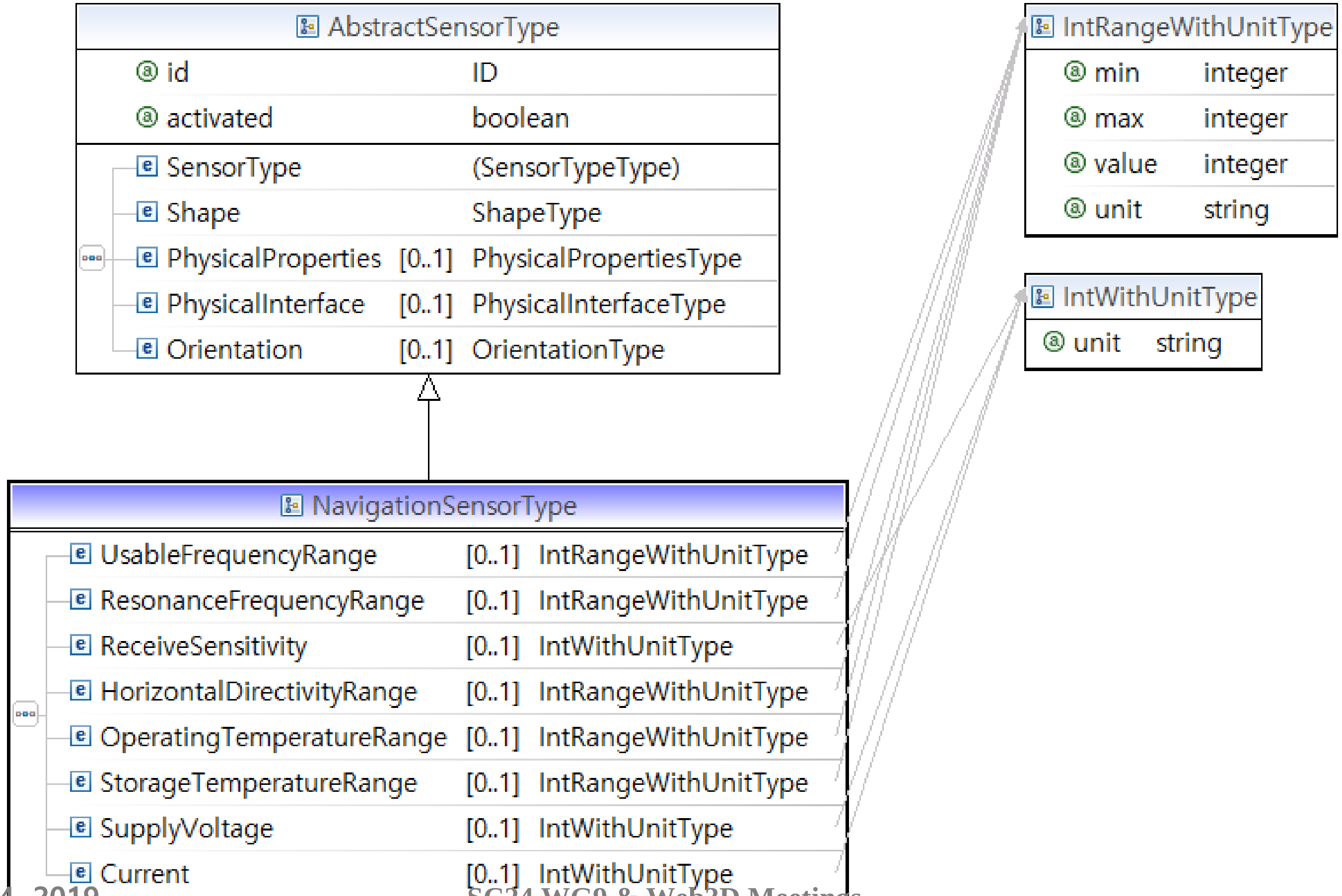
Type 8: Movement Sensor

- MovementSensorType

```
<complexType name="MovementSensorType">
   <complexContent>
      <extension base="mar:AbstractSensorType">
         <sequence>
            <element name="MovingRange" type="mar:IntRangeWithUnitType" minOccurs="0" />
            <element name="Resolution" type="mar:IntWithUnitType" minOccurs="0" />
            <element name="Force" type="mar:FloatWithUnitType" minOccurs="0" />
            <element name="Temperature" type="mar:FloatWithUnitType" minOccurs="0" />
            <element name="HumidityRange" type="mar:IntRangeWithUnitType" minOccurs="0"/>
            <element name="MaxResponseSpeed" type="mar:IntWithUnitType" minOccurs="0" />
            <element name="Materials" type="mar:StringWithUnitType" minOccurs="0" />
            <element name="Weight" type="mar:FloatWithUnitType" minOccurs="0" />
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

Type 9: Navigation Sensor

- NavigationSensorType
  - Includes elements that represent the properties of Usable Frequency Range, Resonance Frequency Range, Receive Sensitivity, etc.



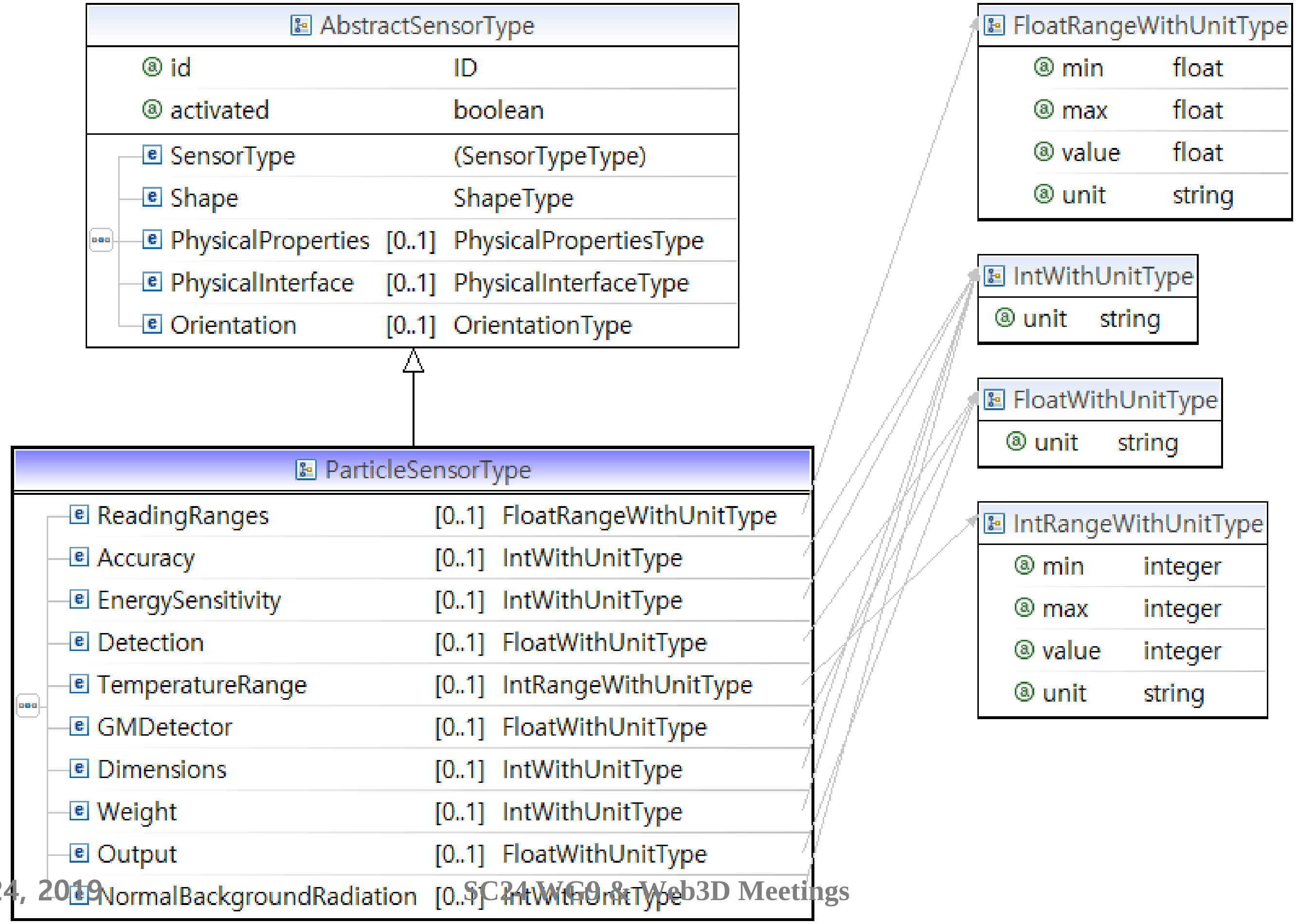
Type 9: Navigation Sensor

- NavigationSensorType

```
<complexType name="NavigationSensorType">
   <complexContent>
      <extension base="mar:AbstractSensorType">
         <sequence>
            <element name="UsableFrequencyRange" type="mar:IntRangeWithUnitType"
               minOccurs="0" />
            <element name="ResonanceFrequencyRange" type="mar:IntRangeWithUnitType"
               minOccurs="0" />
            <element name="ReceiveSensitivity" type="mar:IntWithUnitType"
               minOccurs="0" />
            <element name="HorizontalDirectivityRange" type="mar:IntRangeWithUnitType"
               minOccurs="0" />
            <element name="OperatingTemperatureRange" type="mar:IntRangeWithUnitType"
               minOccurs="0" />
            <element name="StorageTemperatureRange" type="mar:IntRangeWithUnitType"
               minOccurs="0" />
            <element name="SupplyVoltage" type="mar:IntWithUnitType" minOccurs="0" />
            <element name="Current" type="mar:IntWithUnitType" minOccurs="0" />
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

Type 10: Particle Sensor

- ParticleSensorType
  - Includes elements that represent the properties of Reading Ranges, Accuracy, Energy Sensitivity, Detection, etc.



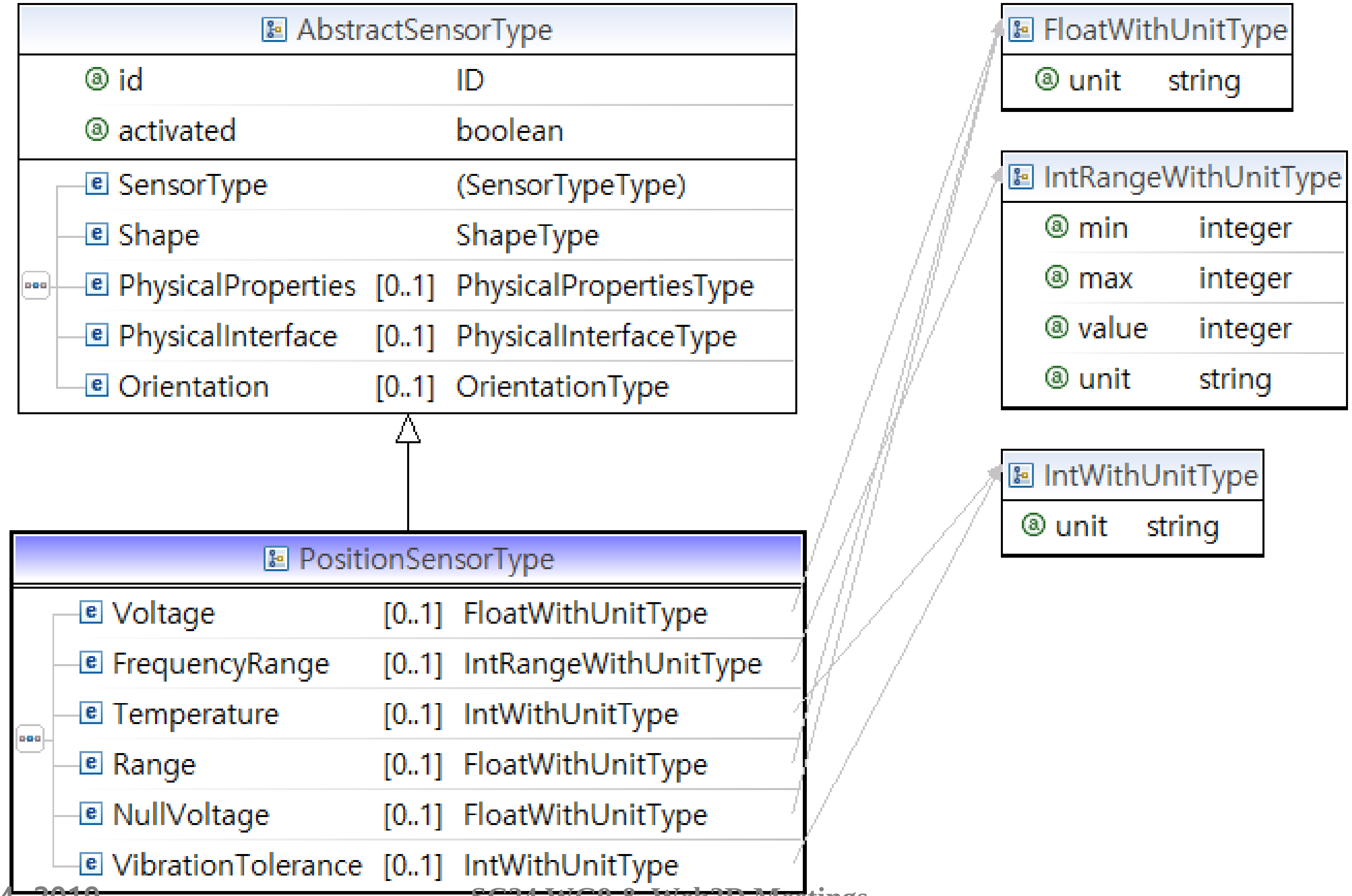
Type 10: Particle Sensor

- ParticleSensorType

```
<complexType name="ParticleSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="ReadingRanges" type="mar:FloatRangeWithUnit"
            minOccurs="0" />
        <element name="Accuracy" type="mar:IntWithUnit" minOccurs="0" />
        <element name="EnergySensitivity" type="mar:IntWithUnit" minOccurs="0"/>
        <element name="Detection" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="TemperatureRange" type="mar:IntRangeWithUnit"
            minOccurs="0" />
        <element name="GMDetector" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="Dimensions" type="mar:IntWithUnit" minOccurs="0" />
        <element name="Weight" type="mar:IntWithUnit" minOccurs="0" />
        <element name="Output" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="NormalBackgroundRadiation" type="mar:IntWithUnit"
            minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 11: Position Sensor

- PositionSensorType
  - Includes elements that represent the properties of Voltage, Frequency Range, Temperature, Range, etc.



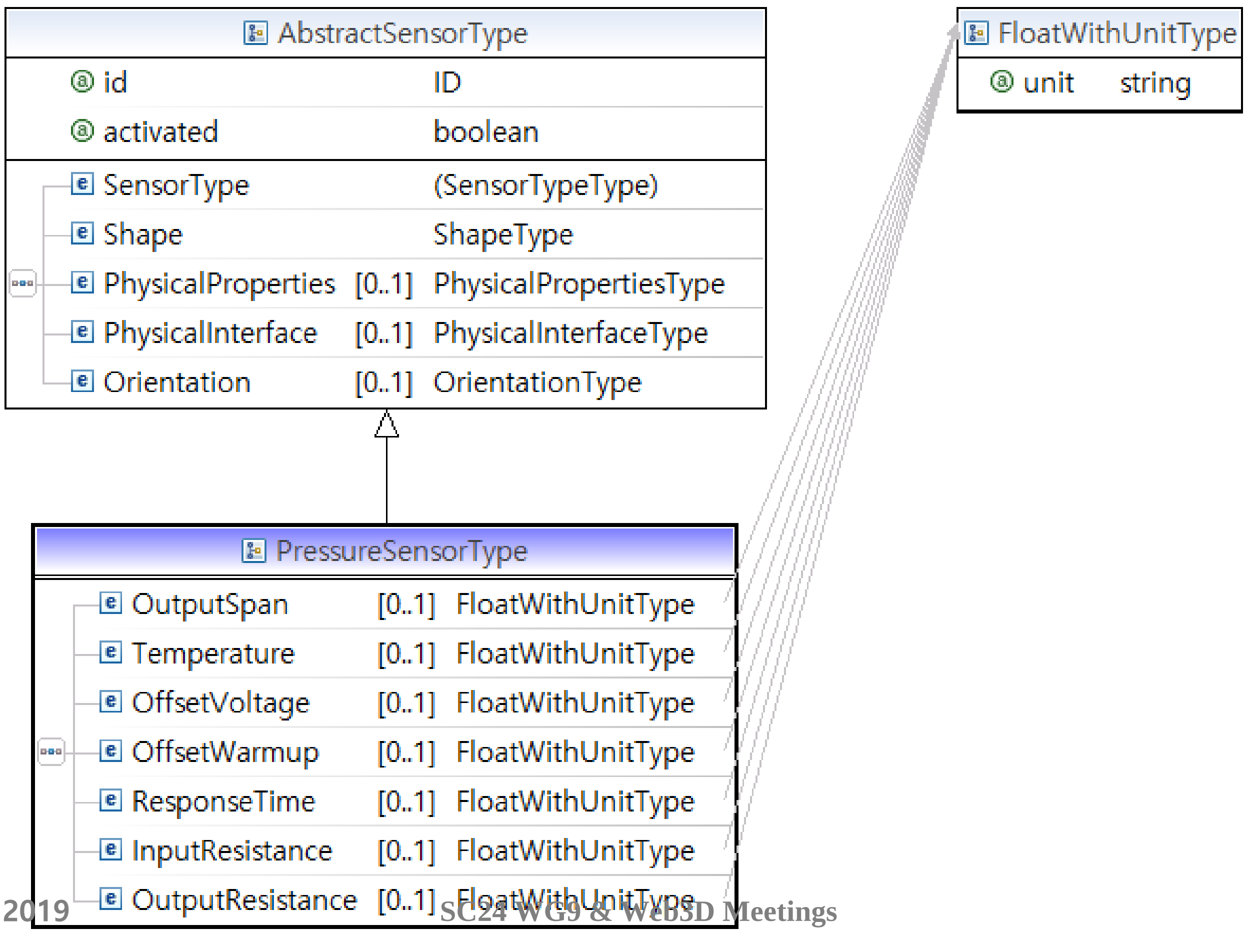
Type 11: Position Sensor

- PositionSensorType

```
<complexType name="PositionSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="Voltage" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="FrequencyRange" type="mar:IntRangeWithUnit"
            minOccurs="0" />
        <element name="Temperature" type="mar:IntWithUnit" minOccurs="0" />
        <element name="Range" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="NullVoltage" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="VibrationTolerance" type="mar:IntWithUnit"
            minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 12: Pressure Sensor

- PressureSensorType
  - Includes elements that represent the properties of Output Span, Temperature, Offset Voltage, Offset Warmup, Response Time, etc.



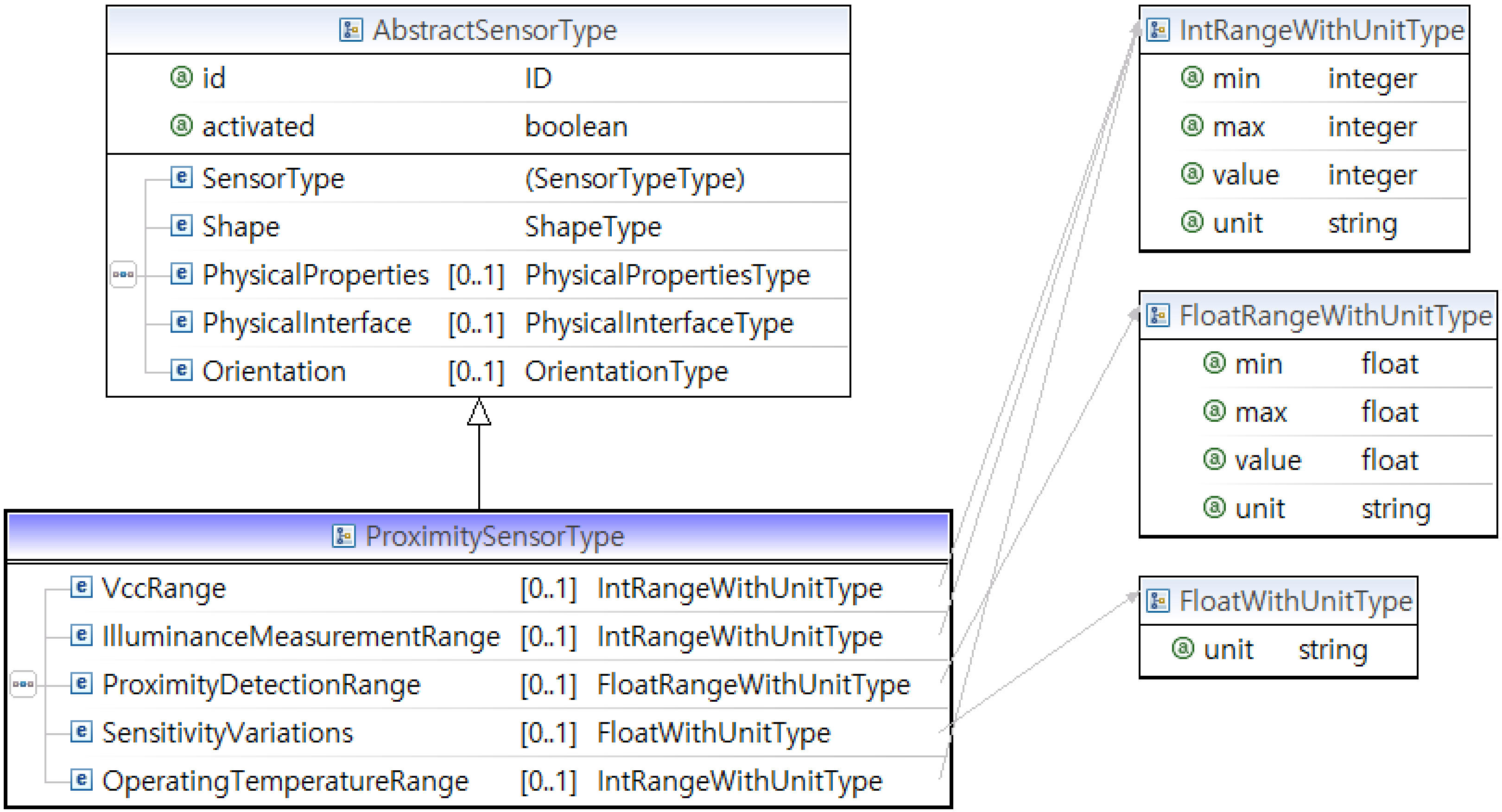
Type 12: Pressure Sensor

- PressureSensorType

```
<complexType name="PressureSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="OutputSpan" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="Temperature" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="OffsetVoltage" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="OffsetWarmup" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="ResponseTime" type="mar:FloatWithUnit" minOccurs="0" />
        <element name="InputResistance" type="mar:FloatWithUnit"
           minOccurs="0" />
        <element name="OutputResistance" type="mar:FloatWithUnit"
           minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 13: Proximity Sensor

- ProximitySensorType
  - Includes elements that represent the properties of Vcc Range, Illuminance Measurement Range, Proximity Detection Range, etc.



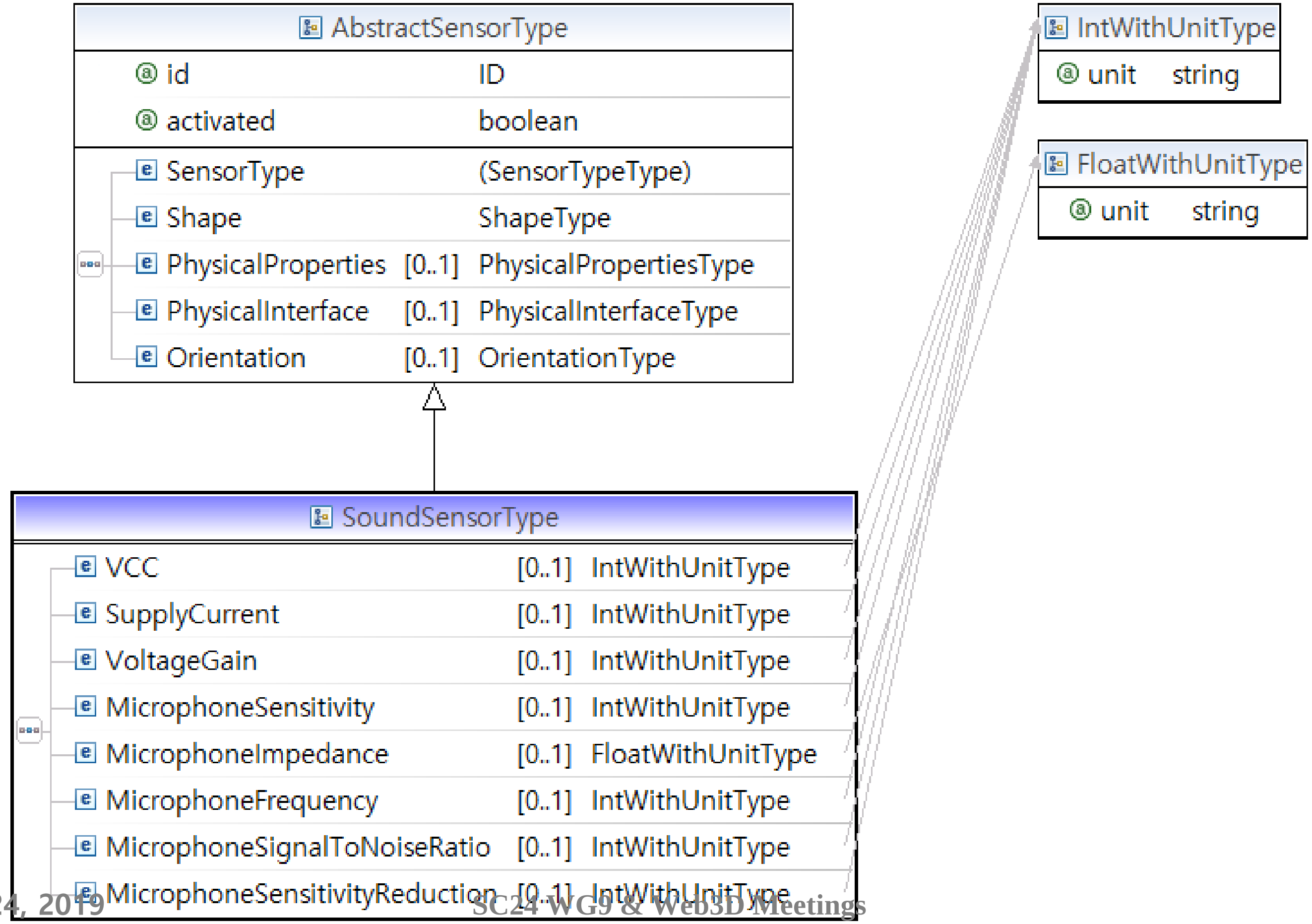
Type 13: Proximity Sensor

- ProximitySensorType

```xml
<complexType name="ProximitySensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="VccRange" type="mar:IntRangeWithUnitType" minOccurs="0" />
        <element name="IlluminanceMeasurementRange" type="mar:IntRangeWithUnitType"
            minOccurs="0" />
        <element name="ProximityDetectionRange" type="mar:FloatRangeWithUnitType"
            minOccurs="0" />
        <element name="SensitivityVariations" type="mar:FloatWithUnitType"
            minOccurs="0" />
        <element name="OperatingTemperatureRange" type="mar:IntRangeWithUnitType"
            minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 14: Sound Sensor

- SoundSensorType
  - Includes elements that represent the properties of VCC, Supply Current, Voltage Gain, Microphone Sensitivity, Impedance, Frequency, etc.



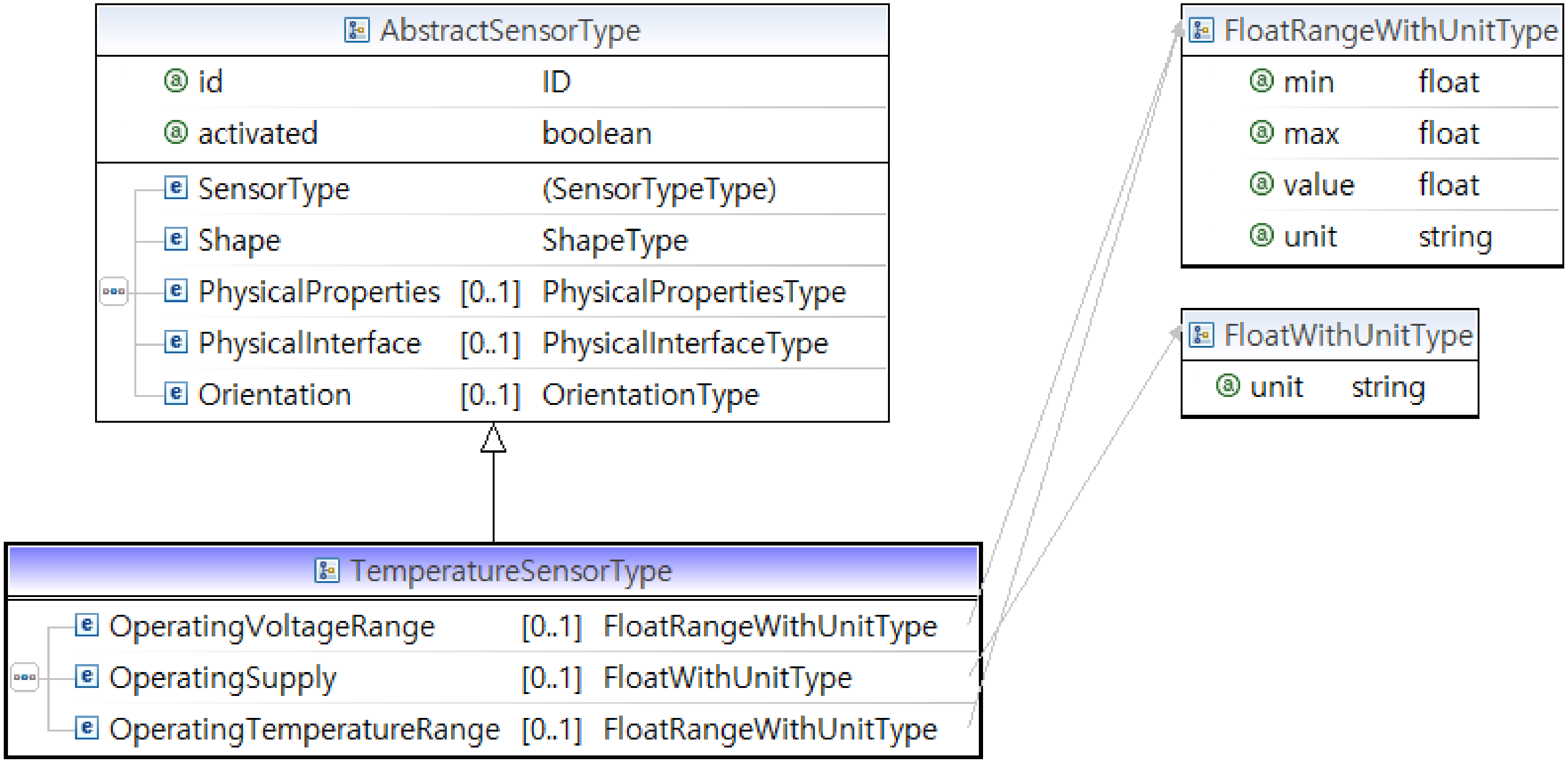
Type 14: Sound Sensor

- SoundSensorType

```
<complexType name="SoundSensorType">
  <complexContent>
    <extension base="mar:AbstractSensorType">
      <sequence>
        <element name="VCC" type="mar:IntWithUnit" minOccurs="0" />
        <element name="SupplyCurrent" type="mar:IntWithUnit" minOccurs="0" />
        <element name="VoltageGain" type="mar:IntWithUnit" minOccurs="0" />
        <element name="MicrophoneSensitivity" type="mar:IntWithUnit"
          minOccurs="0" />
        <element name="MicrophoneImpedance" type="mar:FloatWithUnit"
          minOccurs="0" />
        <element name="MicrophoneFrequency" type="mar:IntWithUnit"
          minOccurs="0" />
        <element name="MicrophoneSignalToNoiseRatio" type="mar:IntWithUnit"
          minOccurs="0" />
        <element name="MicrophoneSensitivityReduction" type="mar:IntWithUnit"
          minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Type 15: Temperature Sensor

- TemperatureSensorType
  - Includes elements that represent the properties of Operating Voltage Range, Operating Supply, and Operating Temperature Range



Type 15: Temperature Sensor

- TemperatureSensorType

```
<complexType name="TemperatureSensorType">
  <complexContent>
     <extension base="mar:AbstractSensorType">
        <sequence>
          <element name="OperatingVoltageRange" type="mar:FloatRangeWithUnitType"
              minOccurs="0" />
          <element name="OperatingSupply" type="mar:FloatWithUnitType"
              minOccurs="0" />
          <element name="OperatingTemperatureRange" type="mar:FloatRangeWithUnitType"
              minOccurs="0" />
        </sequence>
     </extension>
  </complexContent>
</complexType>
```

Annex C

An example of sensor MAR representation based on sensor MAR schema

Sensor MAR Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<MRScene xmlns="http://suwon.ac.kr/mwlee/MAR"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://suwon.ac.kr/mwlee/MAR MRSensor.xsd ">
  <GeoOrigin lat="37.335760" lon="126.584224" />
  <GPSBoundingBox>
     <Position1 lat="37.335800" lon="126.584300" />
     <Position2 lat="37.335800" lon="126.584400" />
     <Position3 lat="37.335900" lon="126.584300" />
     <Position4 lat="37.335900" lon="126.584400" />
     <Length x="10.0" y="5.5" z="7.5" unit="m" />
  </GPSBoundingBox>
  <Orientation>
     <Pitch>22.5</Pitch> <Yaw>152.0</Yaw> <Roll>280.5</Roll>
  </Orientation>
  <MRObject>
     <ThreeDObject>
        <Shape>
           <Material/> <Geometry/> <InterfaceWithVirtualWorlds/>
        </Shape>
     </ThreeDObject>
     <PhysicalSensor activated="true" id="idvalue1" xsi:type="CameraSensorType">
        <SensorType>Camera</SensorType>
        <Shape>
           <Material /> <Geometry /> <InterfaceWithVirtualWorlds />
        </Shape>
```

Sensor MAR Example (cont'd)

```
      <PhysicalProperties>
         <Device>
            <GUID>21EC2020-3AEA-4069-A2DD-08002B30309D</GUID>
            <Name>Camera-101</Name>
            <EventType>Temp</EventType> <ControlType>Temp</ControlType>
            <Desc>Description of camera sensor device and type</Desc>
         </Device>
      </PhysicalProperties>
      <PhysicalInterface>
         <Connection>
            <Name>Camera sensor manager</Name>
            <Desc>Connection info</Desc>
            <IP>202.22.118.243</IP><Port>8080</Port>
            <ID>user1</ID><Password>passwd1</Password>
            <Protocol>TCP</Protocol>
         </Connection>
      </PhysicalInterface>
      <Orientation>
         <Pitch>120.0</Pitch> <Yaw>65.0</Yaw> <Roll>250.5</Roll>
      </Orientation>
      <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>
</MRObject>
```

Sensor MAR Example (cont'd)

```
<MRObject>
  <MRObject>
   <ThreeDObject>
     <Shape>
        <Material /> <Geometry /> <InterfaceWithVirtualWorlds />
     </Shape>
   </ThreeDObject>
   <PhysicalSensor activated="true" id="idvalue2" xsi:type="ChemicalSensorType">
     <SensorType>Chemical</SensorType>
     <Shape>
        <Material />
        <Geometry />
        <InterfaceWithVirtualWorlds />
     </Shape>
     <PhysicalProperties>
        <Device>
           <GUID>3F2504E0-4F89-41D3-9A0C-0305E82C3301</GUID>
           <Name>Chemical-101</Name>
           <EventType>Temp</EventType>
           <ControlType>Temp</ControlType>
           <Desc>Description of chemical sensor device and type</Desc>
        </Device>
     </PhysicalProperties>
```

Sensor MAR Example (cont'd)

```
      <PhysicalInterface>
         <Connection>
            <Name>Chemical sensor manager</Name>
            <Desc>Connection info</Desc>
            <IP>202.22.103.34</IP>
            <Port>8088</Port>
            <ID>user2</ID>
            <Password>passwd2</Password>
            <Protocol>UDP</Protocol>
         </Connection>
      </PhysicalInterface>
      <Orientation>
         <Pitch>32.5</Pitch>
         <Yaw>270.0</Yaw>
         <Roll>120.5</Roll>
      </Orientation>
      <MovingRange min="10" max="50" unit="mm" />
      <Resolution unit="um">30</Resolution>
      <Force unit="N">10.5</Force>
      <Temperature unit="C">25.0</Temperature>
      <HumidityRange min="20" max="70" unit="%" />
      <MaxResponseSpeed unit="m/min">10</MaxResponseSpeed>
      <Materials unit="No">40</Materials>
      <Weight unit="g">50</Weight>
   </PhysicalSensor>
</MRObject>
```

Sensor MAR Example (cont'd)

```
<MRObject>
  <MRObject>
  <ThreeDObject>
     <Shape>
        <Material /> <Geometry /> <InterfaceWithVirtualWorlds />
     </Shape>
  </ThreeDObject>
  <PhysicalSensor activated="true" id="idvalue3" xsi:type="ElectricSensorType">
     <SensorType>Electric</SensorType>
     <Shape>
        <Material />
        <Geometry />
        <InterfaceWithVirtualWorlds />
     </Shape>
     <PhysicalProperties>
        <Device>
           <GUID>3F2504E0-4F89-41D3-9A0C-0305E82C3301</GUID>
           <Name>Electric-3</Name>
           <EventType>Temp</EventType>
           <ControlType>Temp</ControlType>
           <Desc>Description of electric sensor device and type</Desc>
        </Device>
     </PhysicalProperties>
```

Sensor MAR Example (cont'd)

```
        <PhysicalInterface>
           <Connection>
              <Name>Electric sensor manager</Name>
              <Desc>Connection info</Desc>
              <IP>202.20.119.10</IP>
              <Port>8084</Port>
              <ID>user3</ID>
              <Password>passwd3</Password>
              <Protocol>TCP</Protocol>
           </Connection>
        </PhysicalInterface>
        <Orientation>
           <Pitch>110.5</Pitch>
           <Yaw>25.0</Yaw>
           <Roll>220.0</Roll>
        </Orientation>
        <Voltage unit="V">10</Voltage>
        <Frequency unit="Hz">70.0</Frequency>
        <Range unit="mm">10</Range>
        <Temperature unit="C">60.0</Temperature>
        <Humidity unit="percent">70</Humidity>
        <ElectricCurrent unit="V">10.0</ElectricCurrent>
     </PhysicalSensor>
  </MRObject>
</MRScene>
```

Annex D

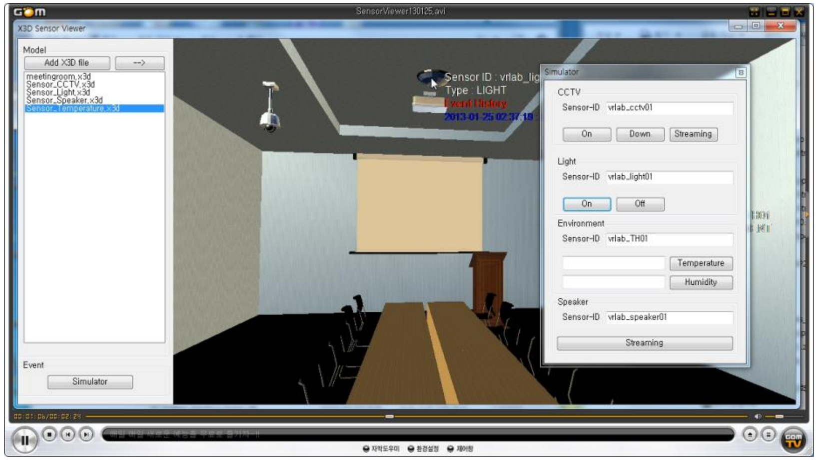
Implementation examples of sensor MAR representation

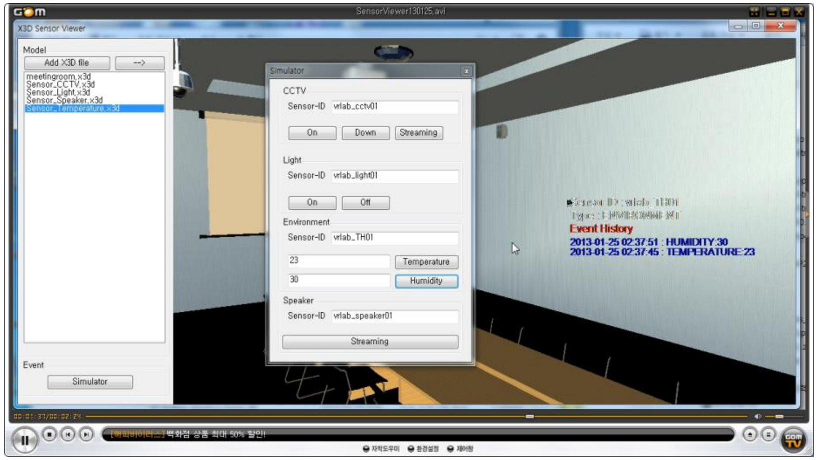
Applications

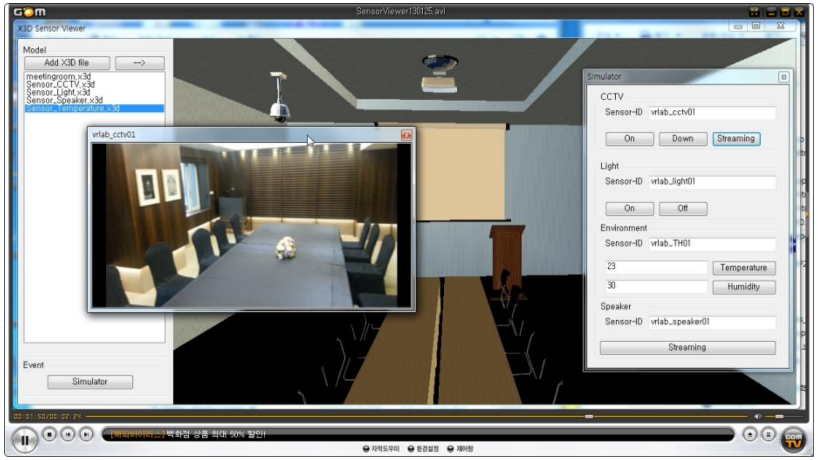
- Indoor 3D simulation for some sensor devices
  - Camera sensor
  - Environment sensor
  - Light sensor
  - Sound sensor

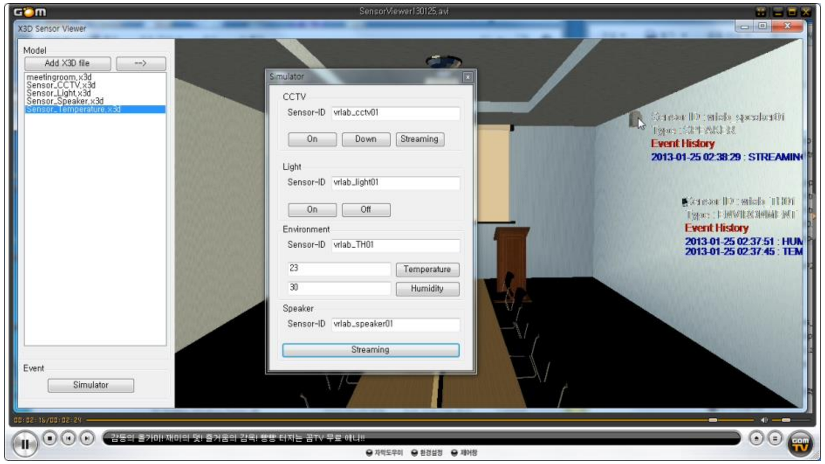
- Outdoor 3D simulation for some sensor devices
  - Camera sensor
  - Electric sensor
  - Light sensor
  - Proximity sensor

Indoor sensor simulation

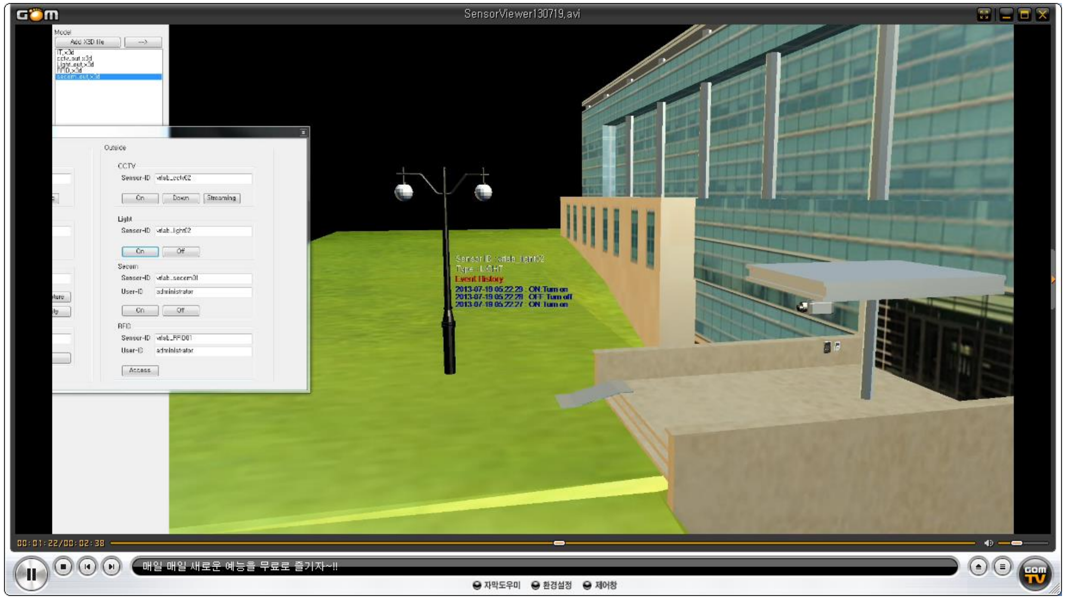


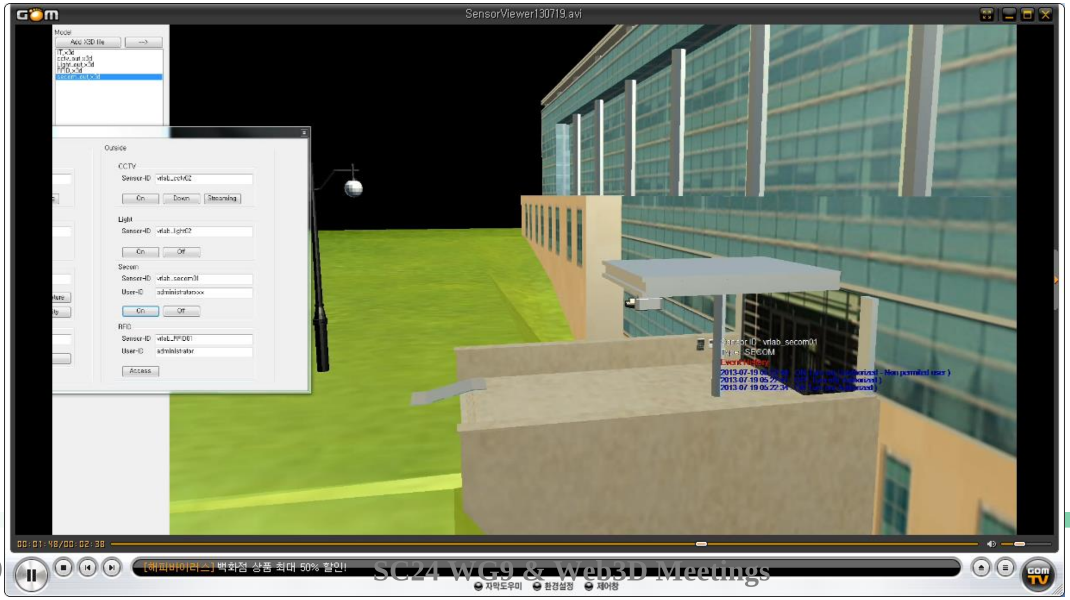






Outdoor sensor simulation





Physical Sensor Types Concepts and Examples (Annex A)

Type 1: Camera sensor

- A camera sensor represents a camera device that converts an optical image into an electronic signal. It is used for digital cameras, phone cameras, camera modules, and other imaging devices, including CCTV.
- The physical functions of a camera sensor can be represented and simulated using defined parameters. The physical functions should be able to be controlled directly in a 3D scene.
- A manipulation interface should be provided to control the representation of the physical functions. Manipulation in a 3D scene includes: turn on a camera sensor using a graphical button, display the camera scene, change the orientation of the camera and then re-display the camera scene, enlarge or reduce the camera scene using a zoom button, turn off the camera sensor using a graphical button.

Example parameters of a camera sensor

Parameter	Unit	Data type
width	mm	float
height	mm	float
diagonal	mm	float
aspect ratio	n/a	int int
pixel count	n/a	int
megapixels	n/a	Int
horizontal pixels	n/a	int
vertical pixels	n/a	int
pixel size	µm	float
frame rate	fps (frames/sec)	int
dynamic range	dB	float
supply voltage	V	float
power consumption	mW	int
operating temperature	°C	float

Type 2: Chemical Sensor

- A chemical sensor is a self-contained analytical device that provides information about the chemical composition of its environment, that is a liquid or a gas phase. The information is provided in the form of a measurable physical signal that is correlated with the concentration of a certain chemical species.
- The physical functions of a chemical sensor can be represented and simulated with parameters as follows:
- In a 3D scene, turn on a chemical sensor using a graphical button, control the proportion of gas or liquid, such as oxygen, using a user interface, display the results, and turn off the chemical sensor using a graphical button.

A three-wire oxygen sensor suitable for use in a Volvo 240 or similar vehicle

Example parameters of a chemical sensor

Parameter	Unit	Data type
detection range	ppm	float
accuracy	%	Int
output signal	mA	int
alarm setting	ppm	int
alarm reset	ppm	int
alarm set point	n/a	int
target gas	n/a	string
supply voltage	V	float
power consumption	mW	int
operating temperature	˚C	int
operating humidity	%	int

Type 3: Electric Sensor

- An electric sensor is a manually or automatically operated electrical switch designed to protect an electrical circuit from damage caused by overload or short circuit. Its basic function is to detect a fault condition and interrupt current flow. Unmmaned security system and RFID sensors are included in the type of electric sensors.
- The physical functions of an electric sensor can be represented and simulated with parameters as follows:
- In a 3D scene, turn on an electric sensor using a graphical button, control the value of each parameter using a user interface, display the action of the sensor, and turn off the electric sensor using a graphical button.

Example parameters of an electronic sensor

Parameter	Unit	Data type
voltage	V	float
frequency	Hz	float
range	mm	int
operating temperature	℃	int
operating humidity	%	int
accuracy	%	float

Type 4: Environment Sensor

- An environment sensor measures and represents earth surface characteristics and supports the information requirements for effective environment management. As a system, the Earth's environment comprises a collection of interdependent elements such as lithosphere, hydrosphere, biosphere, and atmosphere.
- The physical functions of an environment sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on an environment sensor using a button control, represent the parameter values of the sensor whenever they change according to each sensor device connected, and turn off the sensor using a button control.

Example parameters of an environment sensor

Parameter	Unit	Data type
ambient temperature	˚C	int
light	lx	float
pressure	hPa	int
relative humidity	%	int
temperature	˚C	int

Type 5: Flow Sensor

- A flow sensor is a device that senses the rate of fluid flow. Typically, a flow sensor is the sensing element used in a flow meter, or flow logger, to record the flow of fluids. Flow measurement is necessary for representing the function of a flow sensor. An example of a flow sensor is a water meter.
- The physical functions of a flow sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on a flow sensor using a butto control, represent the parameter values of the sensor whenever they change according to each sensor device connected, and turn off the sensor using a button control.

Example parameters of a flow sensor

Parameter	Unit	Data type
voltage	V	int
maximum current	mA	int
weight	g	int
external diameters	mm	int
flow rate	L/min	int
operating temperature	˚C	int
liquid temperature	˚C	int
operating humidity	%RH	Int
operating pressure	mPa, kPa	float

Type 6: Force Sensor

- A force-sensing resistor is a device whose resistance changes when force or pressure is applied. It is also known as a "force sensitive resistor". Force-sensing resistors consist of a conductive polymer which changes resistance in a predictable manner following application of force to its surface.
- The physical functions of a force sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on a force sensor using a button control, represent the parameter values of the sensor whenever they change according to each sensor device connected, and turn off the sensor using a button control.

Example parameters of a force sensor

Parameter	Unit	Data type
thickness	mm	float
length	mm	int
width	mm	int
sensing area	mm^2	float
connector	pin	int
diameter	mm	float
sensitivity	N (g, kg)	float
repeatability	%	float
operating temperature	˚C	int

Type 7: Light Sensor

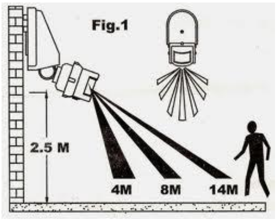
- A light sensor is a device that is used to detect light. Photosensors or photodetectors are sensors of light or other electromagnetic energy. A light sensor includes optical detectors and photo resistors; or Light Dependent Resistors (LDR), which change resistance according to light intensity.
- The physical functions of a light sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on a light sensor using a button control, represent the parameter values of the sensor, and change the intensity of light in a scene based on the sensor devices connected, using an interface such as a scroll bar, and turn off the sensor using a button control.

Example parameters of a light sensor

Parameter	Unit	Data type
voltage	V	float
reverse current	mA	float
collection current	mA	float
collector emitter voltage	V	float
rise/fall time	ms	float
measurement range	lux	float
operating temperature	˚C	int

Type 8: Movement Sensor

- A typical example of a movement sensor is an electronic motion detector which contains a motion sensor that transforms the detection of motion into an electric signal. This is achieved by measuring optical changes in the field of view. A motion detector may be connected to a burglar alarm that is used to alert a home owner or security service after it detects motion.
- The physical functions of a movement sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on a movement sensor using a button control, represent the parameter values of the sensor, display the change to the scene based on changing parameter values from each connected sensor device, and turn off the sensor using a button control.



Example parameters of a movement sensor

Parameter	Unit	Data type
moving range	mm	int
resolution	µm	int
force	N	float
operating temperature	˚C	float
operating humidity	%	int
maximum response speed	m/min	int
materials	n/a	string
weight	g	float
detection distance	m	float
field of view (horizontal)	°	float
field of view (vertical)	°	float

Type 9: Navigation Sensor

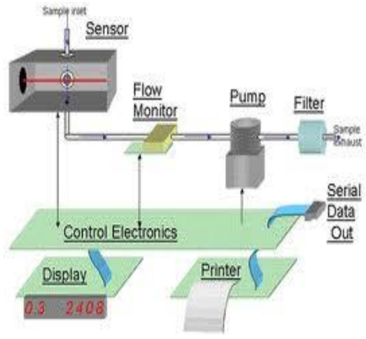
- A navigation sensor is a component of an inertial navigation system (INS) that uses a computer, motion sensors (accelerometers), and rotation sensors (gyroscopes) to continuously calculate via dead reckoning the position, orientation, and velocity (direction and speed of movement) of a moving object without the need for external references.
- The physical functions of a navigation sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on a navigation sensor using a button control, represent the parameter values of the sensor, display the change to the scene based on the parameter values from changing GPS information from each connected sensor device, display the path of the changed location, and turn off the sensor using a button control.

Example parameters of a navigation sensor

Parameter	Unit	Data type
usable frequency range	Hz	int
resonance frequency range	Hz	int
receive sensitivity	dB	int
horizontal directivity range	dB, kHz	int
operation temperature	˚C	int
storage temperature	˚C	int
supply voltage	V	int
current	mA	int

Type 10: Particle Sensor

- A particle sensor is represented by a particle detector in experimental and applied particle physics, nuclear physics, and nuclear engineering. It is also called a radiation detector, and is used to detect, track, and/or identify high-energy particles, such as those produced by nuclear decay, cosmic radiation, or reactions in a particle accelerator.
- The physical functions of a particle sensor are represented and simulated with these parameters as referenced in the following procedure.
- In a 3D scene, turn on a particle sensor using a button control, represent the parameter values of the sensor, display the change in the scene based on changing parameter values from each connected sensor device, and turn off the sensor using a button control.

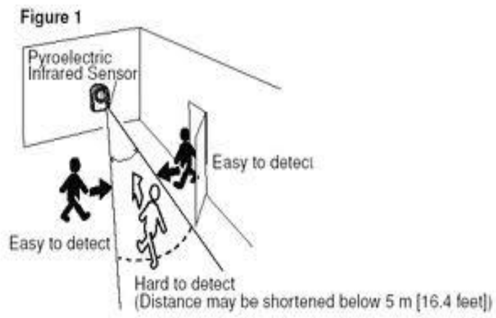


Example parameters of a particle sensor

Parameter	Unit	Data type
reading ranges	mR	float
accuracy	%	int
energy sensitivity	CPM	int
detection	meV	float
GM detector	mg/cm^2	float
dimensions	mm	Int
weight	g	int
output	mm	float
normal background radiation	CPM	int
detectable particle size	µm	float
detection range (concentration range)	pcs/liter, µg/m3	int
supply voltage	V	float
operating temperature	˚C	int
operating humidity	%	int

Type 11: Position Sensor

- A position sensor is any device that permits position measurement. It can either be an absolute position sensor or a relative one (displacement sensor). Position sensors can be linear, angular, or multi-axis.
- The physical functions of a position sensor are represented and simulated with these parameters as follows:
- In a 3D scene, turn on a position sensor using a button control, represent the parameter values of the sensor, display the change in the scene based on changing parameter values from each connected sensor device, and turn off the sensor using a button control.

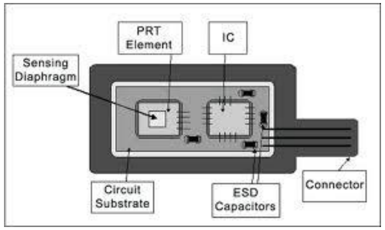


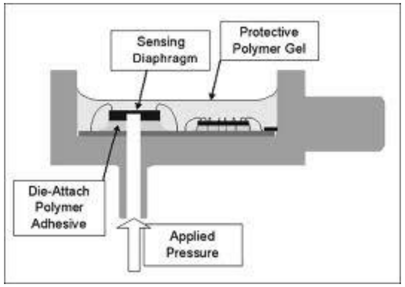
Example parameters of a position sensor

Parameter	Unit	Data type
voltage	V	float
frequency range	Hz	int
temperature	˚F	int
range	˚C	float
null voltage	%	float
vibration tolerance	Hz	int

Type 12: Pressure Sensor

- A pressure sensor measures pressure, typically of gases or liquids. Pressure is an expression of the force required to stop a fluid from expanding and is usually stated in terms of force per unit area. A pressure sensor usually acts as a transducer.
- The physical functions of a pressure sensor are represented and simulated with parameters as follows:
- In a 3D scene, turn on a pressure sensor using a button control, represent the parameter values of the sensor, display the change in the scene based on changing parameter values from each connected sensor device, and turn off the sensor using a button control.
-



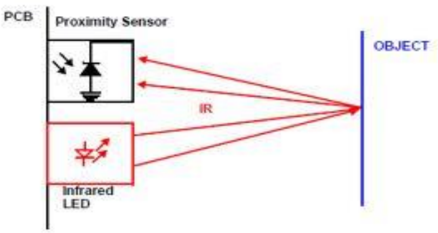


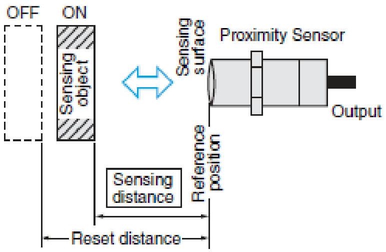
Example parameters of a pressure sensor

Parameter	Unit	Data type
output span	mV	float
operating temperature	˚C	float
offset voltage	V	float
offset warm-up	˚C	float
response time	uS	float
input resistance	ohm	float
output resistance	ohm	float
pressure range	kPa	float
supply voltage	V	float
accuracy	%	float
sensitivity	V/kPa	float

Type 13: Proximity Sensor

- A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact. A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation, and looks for changes in the field or return signal.
- The physical functions of a proximity sensor can be represented and simulated with parameters as follows:
- In a 3D scene, turn on a proximity sensor using a button control, represent the parameter values of the sensor, and display the change in the scene due to the parameter values whenever they change according to each sensor device connected, and turn off the sensor using a button control.



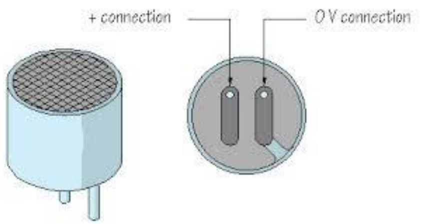


Example parameters of a proximity sensor

Parameter	Unit	Data type
supply voltage	mV	int
illuminance measurement range	V	int
proximity detection range	μW/cm²	float
sensitivity variations	%	float
operating temperature	˚C	int

Type 14: Sound Sensor

- A sound sensor detects and measures sound waves and performs the function of a dynamic microphone. A microphone is an acoustic-to-electric transducer or sensor that converts sound into an electrical signal.
- The physical functions of a sound sensor can be represented and simulated with parameters as follows:
- In a 3D scene, turn on a sound sensor using a button control, represent the parameter values of the sensor, and display the change to the scene based on the parameter values whenever they change from each sensor device connected, and turn off the sensor using a button control.



Example parameters of a sound sensor

Parameter	Unit	Data type
supply voltage	V	int
supply current	mA	int
voltage gain(A)	dB	int
microphone sensitivity	dB	int
microphone impedance	KΩ	float
microphone frequency	HZ	int
microphone S/N ratio	dB	int
microphone sensitivity reduction	dB	int

Type 15: Temperature Sensor

- A thermometer is a device that measures temperature or temperature gradient using a variety of different principles.
- The physical functions of a temperature sensor can be represented and simulated with parameters as follows:
- In a 3D scene, turn on a sound sensor using a button control, represent the parameter values of the sensor, and display the change in the scene based on the parameter values whenever they change from each sensor device connected, and turn off the sensor using a button control.

Example parameters of a temperature sensor

Parameter	Unit	Data type
supply voltage	V	float
supply current	mA	float
operating temperature	˚C	float
temperature accuracy	˚C	float
temperature resolution	˚C	float

Type 16: Other Sensors

- Sensors other than those classified as among the 15 basic sensor types should be able to be represented in a 3D scene.
- Examples:
  - Heartbeat sensor
  - Molecular sensor
  - Nanosensor