INTERNATIONAL STANDARD

ISO/IEC 18038

First edition
2020-04

Information technology — Computer graphics, image processing and environmental representation — Sensor representation in mixed and augmented reality

Reference number
ISO/IEC 18038:2020(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT)

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics, image processing and environmental data representation*.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document defines a representation model for physical sensors to be included in a 3D mixed-reality world. It defines 3D modelling, rendering, simulation, and interfaces for physical sensors. It defines a set of principles, concepts, and functionalities for physical sensors applicable to the complete range of 3D mixed reality standards. It includes the following content:

— terms and definition for sensor interfaces;

— requirements and scope;

— a representation model of physical sensors that can be included in a 3D scene;

— 3D modelling, rendering, and simulation of physical sensors in a 3D scene;

— representation of the attributes of physical sensors in a 3D scene;

— representation of I/O data streaming of physical sensors in a 3D scene;

— representation of the interfaces for controlling physical sensors in a 3D scene;

— functionalities and base components;

— relevant physical sensor properties;

— interfaces with virtual and real worlds;

— use cases.

The objectives of this document are as follows:

— provide a reference model for physical sensor-based 3D mixed-reality applications;

— manage and control physical sensors with their physical properties in 3D mixed reality environments;

— provide an exchangeable information model necessary for transferring and storing data between sensor-based mixed-reality applications;

— support user interfaces with 3D mixed-reality worlds;

— support physical sensor interfaces with 3D mixed-reality worlds.

Information technology — Computer graphics, image processing and environmental representation — Sensor representation in mixed and augmented reality

1 Scope

This document defines the framework and information reference model for representing sensor-based 3D mixed-reality worlds. It defines concepts, an information model, architecture, system functions, and how to integrate 3D virtual worlds and physical sensors in order to provide mixed-reality applications with physical sensor interfaces. It defines an exchange format necessary for transferring and storing data between physical sensor-based mixed-reality applications.

This document specifies the following functionalities:

a) representation of physical sensors in a 3D scene;

b) definition of physical sensors in a 3D scene;

c) representation of functionalities of each physical sensor in a 3D scene;

d) representation of physical properties of each physical sensor in a 3D scene;

e) management of physical sensors in a 3D scene;

f) interface with physical sensor information in a 3D scene.

This document defines a reference model for physical sensor-based mixed-reality applications to represent and to exchange functions of physical sensors in 3D scenes. It does not define specific physical interfaces necessary for manipulating physical devices, but rather defines common functional interfaces that can be used interchangeably between applications.

This document does not define how specific applications are implemented with specific physical sensor devices. It does not include computer generated sensor information using computer input/output devices such as a mouse or a keyboard. The sensors in this document represent physical sensor devices in the real world.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
3D object
collection of vertices in 3D space, connected by various geometric entities such as triangles, lines, curved surfaces, etc.

3.2
augmented reality
AR
interactive experience of a real-world environment whereby the objects that reside in the real world are augmented by computer-generated perceptual information

3.3
camera sensor
sensor (3.26) that detects and converts an optical image into an electronic signal

3.4
closed-circuit television
CCTV
video surveillance which uses video cameras to transmit a signal to a specific place on a limited set of monitors

3.5
chemical sensor
sensor (3.26) that can analyse and provide information about the chemical composition of its environment, that is, a liquid or a gas phase

3.6
electric sensor
sensor (3.26) that examines the change in electrical or magnetic signals based on an environmental input

3.7
environment sensor
sensor (3.26) that monitors relative ambient humidity, illuminance, ambient pressure and ambient temperature

3.8
flow sensor
sensor (3.26) for detecting the rate of fluid flow

3.9
force sensor
sensor (3.26) for force detecting resistor whose resistance changes when force or pressure is applied

3.10
globally navigation satellite system
GNSS
satellite navigation system with global coverage

3.11
globally unique identifier
GUID
unique reference number used as an identifier in computer systems

3.12
light sensor
photodetector which detects changes in quantities of optical signal

3.13
mixed and augmented reality
MAR
integration of real and virtual worlds including *mixed reality* (3.14) and *augmented reality* (3.2)

3.14
mixed reality
MR
merging of real and virtual worlds to generate new environments where physical and synthetic objects co-exist and interact

3.15
mixed reality system
system that can process *mixed reality* (3.14) applications with manipulation functions such as read, write, import, export, modify, display, etc.

3.16
movement sensor
detector to detect a change in position of an object relative to its surroundings or the change in the surroundings relative to an object

3.17
oxygen sensor
electronic device that measures the proportion of oxygen (O_2) in the gas or liquid being analysed

3.18
particle sensor
detector to detect, track, and/or identify high-energy particles

3.19
physical device
real device containing a *sensor* (3.26) which is represented by a virtual device in a virtual environment

3.20
physical sensor
Internet of things (IoT) *sensor* (3.26) which has the functionality of a *physical device* (3.19) in a 3D virtual world

3.21
position sensor
sensor (3.26) that permits position measurement

3.22
pressure sensor
sensor (3.26) that measures pressure, typically of gases or liquids

3.23
programmable logic controller
PLC
digital computer used for automation of typically industrial electromechanical processes, such as control of machinery on factory assembly lines, amusement rides, or light fixtures

3.24
proximity sensor
sensor (3.26) able to detect the presence of nearby objects without any physical contact

3.25
radio frequency identification
RFID
wireless use of electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects

3.26
sensor
device to detect events or changes in its environment and send the information to other electronics

3.27
sound sensor
sensor (3.26) used to detect the sound intensity of the environment

3.28
temperature sensor
sensor (3.26) to detect a change in temperature

3.30
universally unique identifier
UUID
globally unique identifier
128-bit number used to identify information in computer systems

3.32
virtual world
collection of one or more virtual reality (VR) files and other multimedia content that, when interpreted by a VR browser, presents an interactive experience to the user consistent with the author's intent

Note 1 to entry: Virtual reality (VR) is understood as an interactive computer-generated experience taking place in a synthetic and simulated environment.

4 Concepts

4.1 Overview

This clause describes the concepts of sensor-based mixed and augmented reality, including definition, objectives, sensor type, physical sensor representation, system functions for mixed and augmented reality (MAR), MAR objects, MAR scene graph, and MAR world.

A mixed-reality world consists of a 3D virtual world and real-world sensors represented as 3D objects with their physical properties. As a simple example, the conceptual scene of a mixed-reality world is represented in Figure 1. It displays a heritage site represented by a 3D virtual world with a global navigation satellite system (GNSS) sensor and a CCTV sensor. The virtual world is of a real heritage location in a city and the character represents a tourist. GNSS information is displayed for the tourist and a real CCTV device is located at its real physical location at the heritage site.

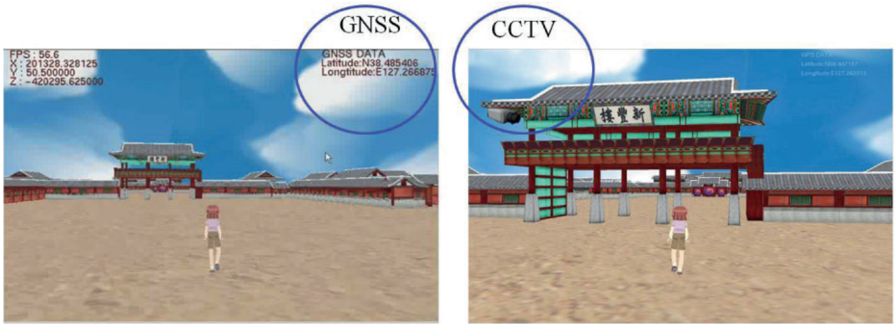


Figure 1 — Example sensor-based mixed-reality world

Once real physical sensors are integrated into a 3D virtual world, their physical properties can be represented precisely in the virtual world. Sensor-based mixed reality is obtained by this convergence of 3D with physical sensors in the real world. For sensor-based mixed reality, sensors in a 3D virtual world are defined, and their information should be able to be transferred between applications, and between a virtual world and a real world. This work is intended to define how to exchange AR/MR application data in heterogeneous computing environments, and how physical sensors can be managed and controlled with their physical properties in a 3D virtual world.

Physical sensors in the real world are many and varied[11][14][18]. In order to control them in a 3D scene, these physical sensors are classified based on their information types and functions. Types of physical sensor devices include acoustic and sound, automotive and transportation, chemical, electric and magnetic, environment and weather, flow and fluid, radiation and particle, navigation, position and angle, speed and acceleration, optical and light, pressure, force and density, thermal and temperature, proximity and presence, and video. Each sensor is represented as a physical device in a 3D scene visually and/or functionally depending on the application and the type of sensor. Figure 2 shows an indoor and an outdoor scene, each with many physical sensors. Each scene represents a corresponding real world. The information and function of each sensor can be represented in the scenes. MAR scenes with physical sensors can be used for representing and simulating the functions of the sensors and, therefore, for managing the sensors using the 3D scene. These can also be used for facility management in a real world[3].

This document focuses on how to represent physical sensors in a 3D scene, what to represent about each physical sensor, what each sensor can do and the reason why each sensor needs this specification. When representing a physical sensor in a 3D scene, having the sensor appear in the scene is optional depending on the type of sensor and the application. Precise location and orientation of a sensor should be able to be represented and units for each sensor should be specified. A 3D scene should be able to be changed by the function of a physical sensor and simulated accordingly. The reason why such representation is needed is to provide a 3D scene with capabilities that can manage and control various physical sensors, for information services or security purposes.

Key

1	camera sensor	4	environment sensor	7	navigation sensor	10	sound sensor
2	chemical sensor	5	flow sensor	8	pressure sensor	11	temperature sensor
3	electric sensor	6	light sensor	9	proximity sensor		

Figure 2 — MAR worlds representation with physical sensors

4.2 Scope of physical sensor representation

Many types of physical sensors are currently in use in the real world and it is expected that the number of sensors and their types will increase with advancements in physical sensor technologies. Physical sensors integrate with 3D virtual worlds by way of information convergence technologies, including mixed and augmented reality. These technologies will be further developed and progressed based on industry need. While physical sensor devices continue to advance technologically, current sensor devices are being integrated into 3D virtual worlds for use in various real-world simulation applications.

It is not easy and, in fact, unnecessary to define every possible type of sensor that can be integrated into a 3D virtual world because these vary and constantly are updated based on the progression of sensor technology. Although the number of types of physical sensors will increase, a common interface for all sensor types is necessary in order to integrate them with a 3D scene. The interface should have the following features:

— the appearance, properties, location and orientation of a physical sensor should be represented in a 3D virtual world. The 3D virtual world should represent a copied scene of a real world;

— the functions of a physical sensor should be visualized or represented in a 3D virtual world;

— all other sensors that cannot be represented in a 3D virtual world visually and/or functionally are excluded.

In order to provide a 3D virtual world with common physical sensor interfaces, an abstracted physical sensor data model is necessary to represent and simulate these physical sensors[13]. This document defines the data model for representing physical sensors in 3D MAR worlds. The data model defines an abstracted interface that can be used for any type of sensors, not including specific attributes of a particular sensor type such as organization of a data stream.

In this document, the scope of sensor representation includes the following topics:

— concepts of physical sensors in a 3D scene;

— how to represent physical sensors in a 3D scene;

— how to organize a 3D scene with physical sensors;

— how to define an abstract model for representing physical sensors in a 3D scene;

— how to define a system architecture for physical sensors in a 3D scene;

— how to use physical sensors in a 3D scene;

— types of physical sensors for sensor representation.

4.3 Physical sensor types

4.3.1 General

Generally, physical sensors and their related devices can be classified as follows:

— acoustic, sound, vibration;

— automotive, transportation;

— camera, image;

— 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, photon;

— pressure;

— force, density, level;

— thermal, heat, temperature;

— proximity, presence.

Each sensor type can be defined based on its physical properties and related devices that can be represented and simulated in a 3D virtual world. Typical parameters for the physical properties of each sensor type are described in Annex A. The sensor types are classified (in alphabetical order) in 4.3.2 to 4.3.17.

4.3.2 Camera sensor

This sensor type integrates real world camera and images into a 3D virtual world. A camera sensor is represented as 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 (Figure 3). An abstract data model concerning the visual, functional, and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

Figure 3 — Camera sensors

4.3.3 Chemical sensor

This sensor type integrates real world chemical detection devices into a 3D virtual world. All chemical detection devices, such as smoke detectors, are included. An abstract data model concerning the visual, functional, and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

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 (Figure 4). The information is provided in the form of a measurable physical signal that is correlated with the concentration of a certain chemical species (termed an analyte). For example, an oxygen sensor (or lambda sensor) is an electronic device that measures the proportion of oxygen (O_2) in the gas or liquid being analyzed.

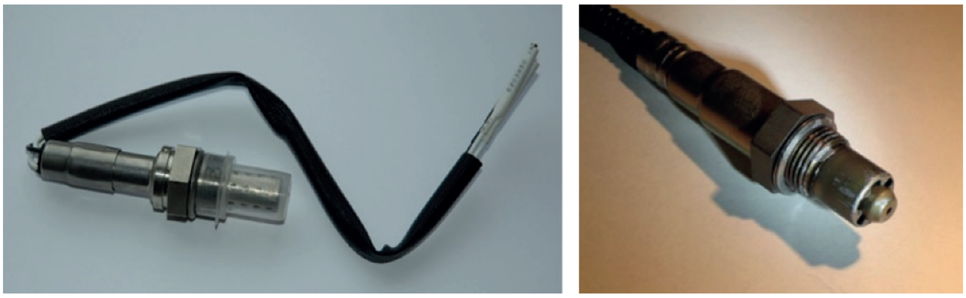


Figure 4 — Chemical sensor (oxygen sensor)

4.3.4 Electric sensor

This sensor type integrates real-world electric and electronic signals into a 3D virtual world. All electrical devices, such as electricity and voltage detectors, are included. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

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 (Figure 5). Its basic function is to detect a fault condition and interrupt current flow. Unmanned security system and RFID sensors are included.

Figure 5 — Electric sensors

4.3.5 Environment sensor

This sensor type integrates real world environmental change in weather, humidity, barometric air pressure as well as air quality measuring into a 3D virtual world. An abstract data model concerning the visual, functional, and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

An environment sensor measures and represents earth surface characteristics and supports the information requirements for effective environment management (Figure 6). As a system, the Earth's environment comprises a collection of interdependent elements such as lithosphere, hydrosphere, biosphere, and atmosphere. A single instrument can be used to measure liquids and solids, and granular, slurry or open-channel flow without changing the transducer. Environment sensors can detect dust, gas, humidity, ambient light and weather.

Figure 6 — Environment sensors

4.3.6 Flow sensor

This sensor type integrates real world flow in air and fluid into a 3D virtual world. All flow detectors, such as air flow and fluid sensors, are included. An abstract data model concerning the visual, functional, and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

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 (Figure 7). Water metering is the process of measuring water use. Water meters can be used at the water source, at a well or throughout a water system, to determine flow through a particular portion of the system.

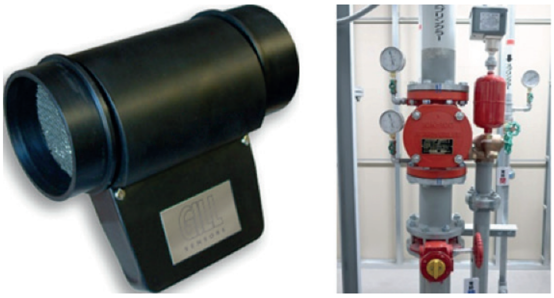


Figure 7 — Flow sensors

4.3.7 Force sensor

This sensor type integrates real-world force, density and level measurements into a 3D virtual world. All force sensors, force transducers and liquid and gas density and level measurement sensors are included. Level sensors detect the level of substances that flow (including liquids), slurries, granular materials and powders. An abstract data model concerning the visual, functional, and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A force-sensing resistor is a device whose resistance changes when force or pressure is applied (Figure 8). 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. In a 3D scene, a force sensor device can be represented with some motion based on its functions.

Figure 8 — Force sensors

4.3.8 Light sensor

This sensor type integrates real-world optical, light and photon measurements into a 3D virtual world. All optical, light and photon detectors are included. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A light sensor is a device that is used to detect light. Photosensors or photodetectors are sensors of light or other electromagnetic energy. In this document, a light sensor includes optical detectors and photo resistors, or light-dependent resistors (LDR), which change resistance according to light intensity (Figure 9). In a 3D scene, the light sensor represents the physical intensity of light based on the functions of the device. In a 3D scene, the light sensor itself is not typically represented. Rather, a 3D object that uses a light sensor, such as a light bulb, fluorescent light, or street light, is represented. It is controlled based on the functions of the included light sensor.

Figure 9 — Light sensors

4.3.9 Movement sensor

This sensor type integrates real world moving objects into a 3D virtual world. Cars, robots, and fans are examples of devices whose movement can be detected. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

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 (Figure 10). This is achieved by measuring optical changes in the field of view. A motion detector can be connected to a burglar alarm that is used to alert a home owner or security service after it detects motion. Such a detector can also trigger a red light camera. Some of these applications include motion-activated outdoor lighting systems, motion sensor street lamps, and motion sensor lanterns.

Figure 10 — Movement sensors

4.3.10 Navigation sensor

This sensor type integrates real world navigation into a 3D virtual world. All navigation sensors that detect current position, orientation and other navigation information are included. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

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 (Figure 11). It is used on ships, aircraft, submarines, guided missiles and spacecraft, for example. In a 3D scene based on the application, the navigation sensor itself is optionally represented.

Figure 11 — Navigation sensors

4.3.11 Particle sensor

This sensor type integrates real-world ionizing radiation and subatomic particles into a 3D virtual world. All ionizing radiation and subatomic particle detectors are included. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A particle sensor is represented by a particle detector in experimental and applied particle physics, nuclear physics and nuclear engineering (Figure 12). 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. Modern detectors are also used as calorimeters to measure the energy of detected radiation. They can also be used to measure other attributes such as momentum, spin, charge, etc., of the particles. Detectors designed for modern accelerators are huge, both in size and in cost. The term “counter” is often used instead of detector when the detector counts the particles but does not resolve their energy or ionization.

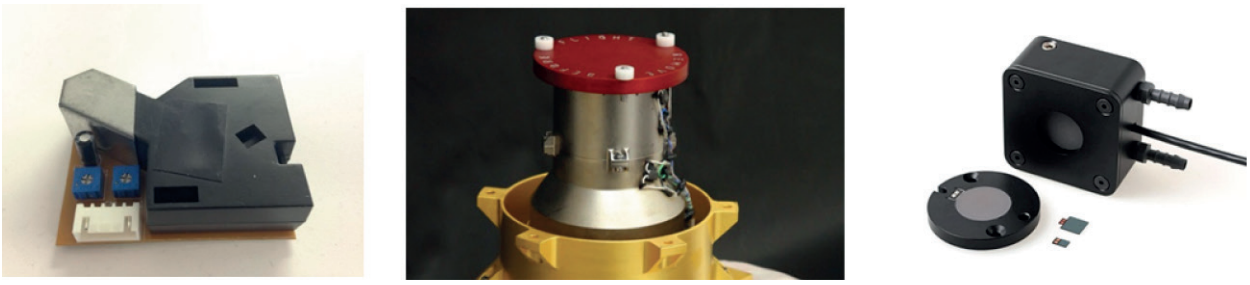


Figure 12 — Particle sensors

4.3.12 Position sensor

This sensor type integrates real-world measurements of position, angle, displacement, distance, speed and acceleration into a 3D virtual world. All measurement sensors (such as position, angle and oscilloscope, displacement, distance, speed and acceleration detectors) are included. An abstract data model concerning the visual, functional, and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A position sensor is any device that permits position measurement (Figure 13). It can either be an absolute position sensor or a relative one (displacement sensor). Position sensors can be linear, angular or multi-axis. In determining a position, the method can use:

— "distance", which would be the distance between two points such as the distance travelled or moved away from some fixed point; or

— "rotation" (angular movement), for example, the rotation of a robot's wheel to determine its distance travelled along the ground.

Either way, position sensors can detect the movement of an object in a straight line using linear sensors or angular movement using rotational sensors.

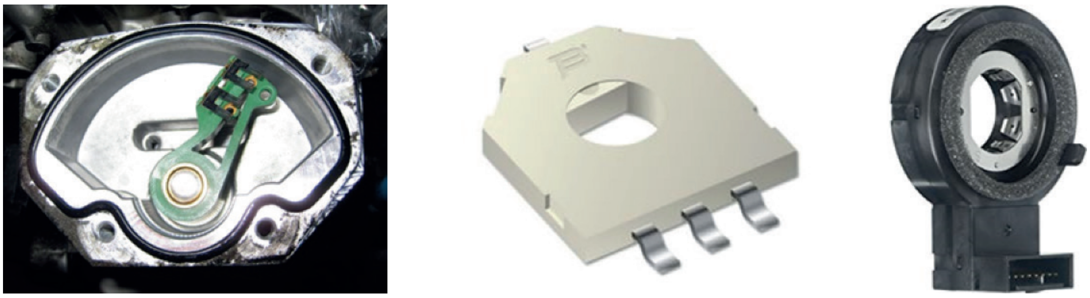


Figure 13 — Position sensors

4.3.13 Pressure sensor

This sensor type integrates real world pressure measurement into a 3D virtual world. All pressure sensors and pressure transducers are included. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined for representing and simulating them in a 3D virtual world.

A pressure sensor measures pressure, typically of gases or liquids (Figure 14). 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 – it generates a signal as a function of the pressure imposed. For the purposes of this document, such a signal is considered to be electrical. Pressure sensors are used for control and monitoring in many applications. Pressure sensors can also be used to indirectly measure other variables such as fluid/gas flow, speed, water level and altitude.

Figure 14 — Pressure sensors

4.3.14 Proximity sensor

This sensor type integrates real-world proximity and presence measurements into a 3D virtual world. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A proximity sensor detects the presence of nearby objects without any physical contact (Figure 15). It typically emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance) and looks for changes in the field or return signal. Proximity sensors are also used in machine vibration monitoring to measure the variation in distance between a shaft and its support bearing.

Figure 15 — Proximity sensors

4.3.15 Sound sensor

This sensor type integrates real-world sound into a 3D virtual world. Devices such as speakers, microphones, hearing aids, and vibrating instrumentation devices are included. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A sound sensor detects and measures sound waves and performs the function of a dynamic microphone (Figure 16). A microphone is an acoustic-to-electric transducer or sensor that converts sound into an electrical signal. Microphones are used in many applications [such as telephones, tape recorders, karaoke systems, hearing aids, motion picture production, live and recorded audio engineering, FRS (family radio service), megaphones, radio and television broadcasting] and to record voice for speech recognition and VoIP. This also includes non-acoustic purposes, such as ultrasonic checking or knock sensors. Most microphones today use electromagnetic induction (dynamic microphone), capacitance change (condenser microphone), piezoelectric generation or light modulation to produce an electrical voltage signal from mechanical vibration.

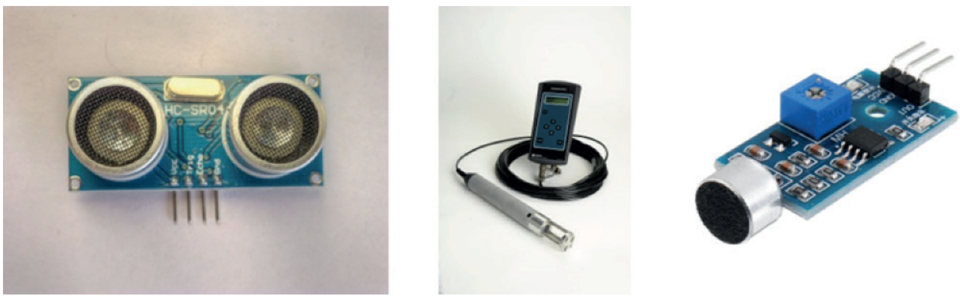


Figure 16 — Sound sensors

4.3.16 Temperature sensor

This sensor type integrates real-world temperature into a 3D virtual world. A thermometer is a device that measures temperature or a temperature gradient. An abstract data model concerning the visual, functional and physical properties of this sensor type should be defined to represent and simulate them in a 3D virtual world.

A thermometer has two important elements:

— a temperature sensor (e.g. the bulb of a mercury-in-glass thermometer or the digital sensor in an infrared thermometer) in which some change occurs with a change in temperature; and

— some means of converting this change into a numerical value (e.g. the visible scale that is marked on a mercury-in-glass thermometer or the digital readout on an infrared model).

A temperature sensor responds to the change of temperature and is used for automizing temperature management (Figure 17). It senses thermal change, produces an electrical signal and is classified as touch and non-touch type. The touch type measures temperature value by touching directly to an object while non-touch type measures thermal radiation from an object.

Figure 17 — Temperature sensors

4.3.17 Other sensors

In addition to the above sensor types, there are many others that are being created and whose use is becoming more common. These should be able to be classified and represented as differentiable sensor types as their typical properties are defined.

4.4 Sensor representation

4.4.1 Overview

Physical sensor-based mixed reality integrates physical sensor devices (or just physical sensors) and their functionalities in 3D virtual worlds. The appearance and physical properties of a physical sensor should be able to be represented in a 3D virtual world. In addition, the physical properties and events should be able to be controlled and managed in the virtual world.

In order to provide a 3D virtual world with the capability of representing physical sensors, mixed-reality systems require the following functions:

— representation of physical sensors in a 3D virtual world: a physical sensor can be represented as a 3D geometric model or as an invisible sensor device at a precise location and orientation;

— representation of visual and functional properties of each physical sensor in a 3D scene;

— representation of physical properties of each physical sensor in a 3D scene;

— control of a physical sensor data stream in a 3D scene;

— interface for controlling physical sensors in a 3D scene.

Figure 18 shows a physical sensor representation model which consists of a sensor MAR world, a spatial mapper and an event mapper. A simulated sensor MAR world organizes MAR content with scene composition. Many types of physical sensors in the real world can be represented and controlled in 3D virtual worlds.

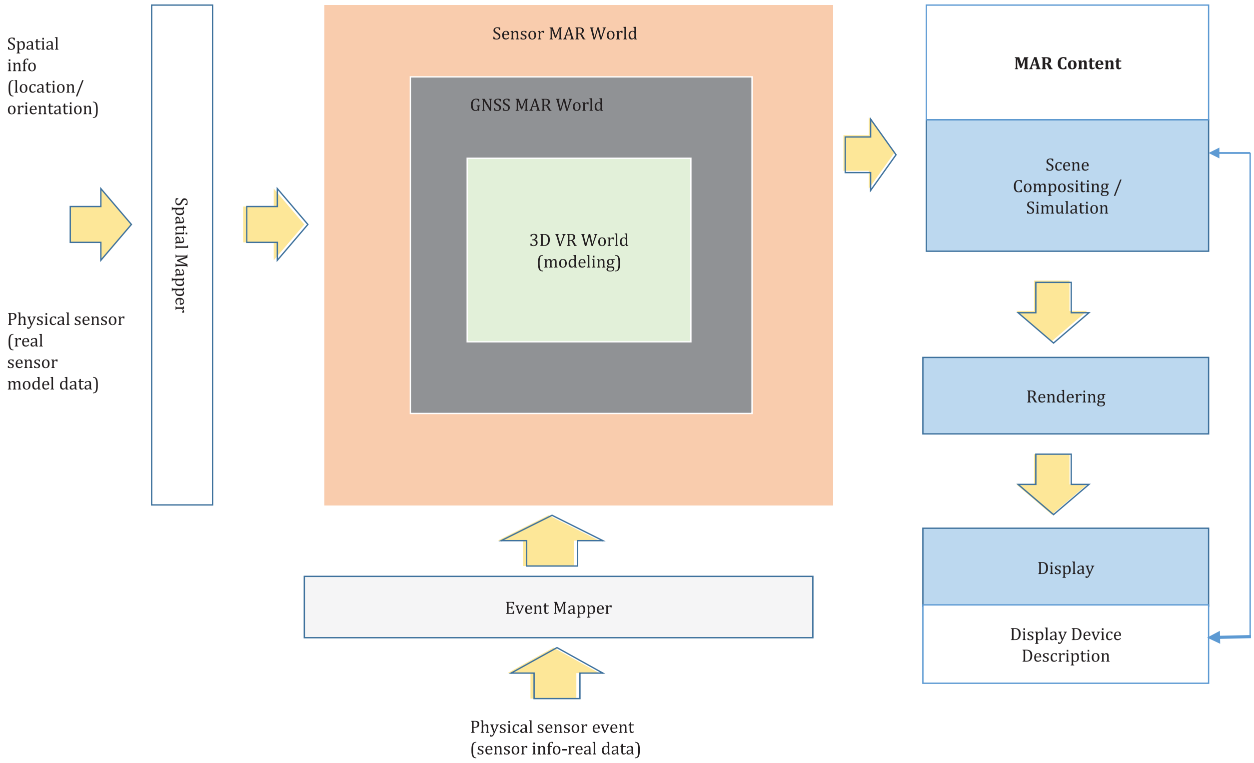


Figure 18 — Physical sensor representation model

Objects in a physical sensor-based 3D MAR world can be modelled by geometric or non-geometric data depending on the properties of the objects. The MAR world should be modelled with precise location and orientation information for all the objects. In other words, the MAR world should be spatially synchronized with its corresponding real world based on location and orientation information. The location can be defined with its GNSS information and the orientation can be represented with the direction of the object. Figure 19 shows the procedure for creating a sensor MAR world in a 3D scene.

A 3D virtual world is first created with 3D modelling, rendering and animation[9][10]. Then, the virtual world is expanded into an MAR world with geospatial coordinates, that is synchronized with its corresponding real world[4]-[7]. The MAR world provides each object with geographical information. The physical sensor-based MAR world is created from the MAR world integrated with physical sensors.

Physical sensor events should be able to be represented in the MAR world. There are many different types of sensors with different types of event information depending on the physical properties of the physical sensor devices. Event information can be represented as text, image, binary stream, etc., based on the type of data stream coming from the sensor. MAR content is composed of 3D virtual objects representing physical objects in the real world and physical information such as events, generated by physical sensors.

Two kinds of interfaces should be included. One is an interface that controls and manipulates each visual MAR object in the 3D scene. The other is an interface to import/export events from physical sensors. These interfaces can be integrated in a 3D scene so physical sensors can be controlled and affect the scene directly.

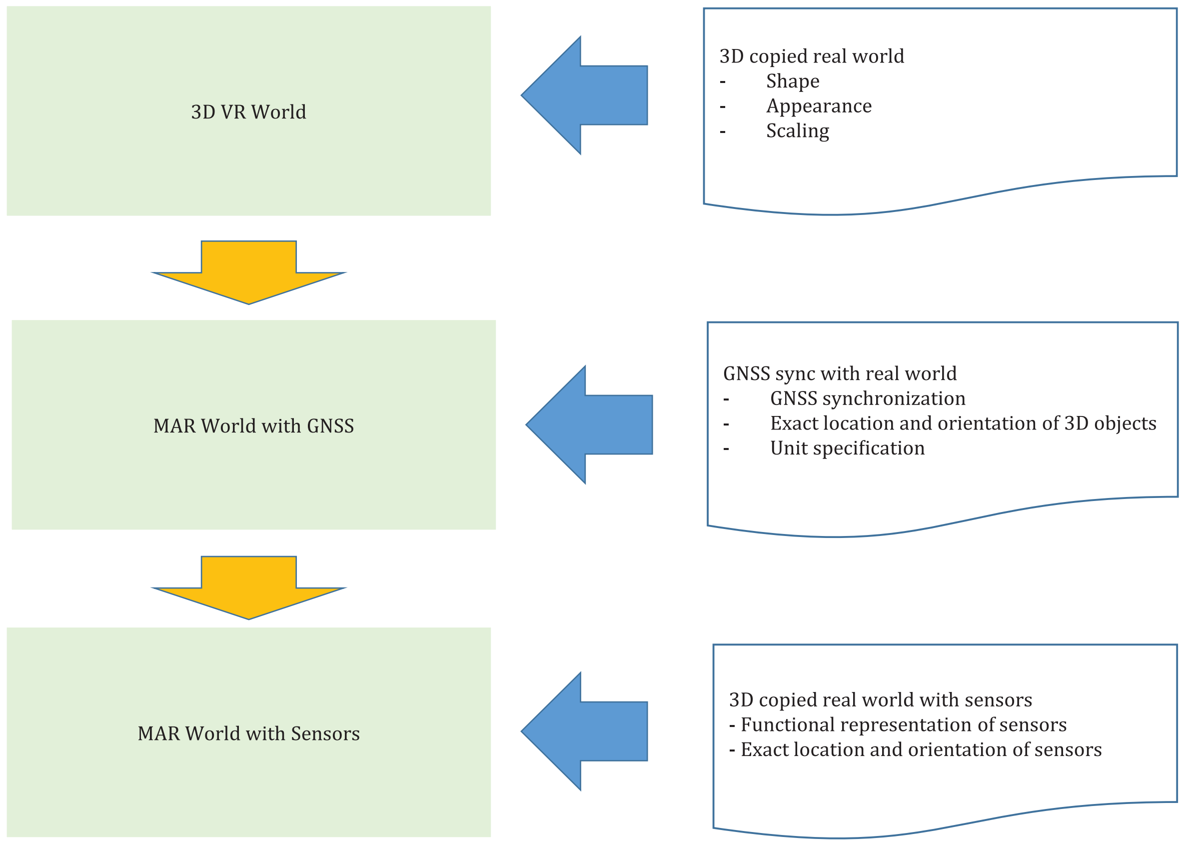


Figure 19 — Creation of a sensor MAR world

4.4.2 Precise location and orientation of a physical sensor

Precise location and orientation of a physical sensor in a 3D MAR world is specified by the following procedure:

Step 1: Define the geo-coordinate system of the 3D virtual environment aligned with a GNSS bounding box (Figure 20). The bounding box defines 8 GNSS positions for a 3D MAR world that represents an indoor or an outdoor space, depending on the application. The 4 GNSS positions at the bottom can be obtained using a general GNSS device, and the 4 GNSS positions at the top can be obtained using the 4 GNSS positions at the bottom plus the length and orientation of the bounding box. Therefore, one geo coordinate system is required per 3D scene. A GNSS sensor defines the origin of the geo coordinate system and the orientation of the 3D MAR space is used to determine the axes of the geo coordinate system.

Step 2: Define the orientation of each sensor in the 3D virtual environment aligned with the geo coordinate system (Figure 21). The location of the sensor is defined in the geo coordinate system. The orientation of the sensor is defined using pitch-yaw-roll values.

Step 3: Define real length with units aligned with the geo coordinate system and orientation-specified pitch-yaw-roll values (Figure 22). The sensor is located at the precise position with real length units.

Figure 23 shows an enhanced graphics pipeline process, including all the steps for locating an IoT sensor in an MAR world.

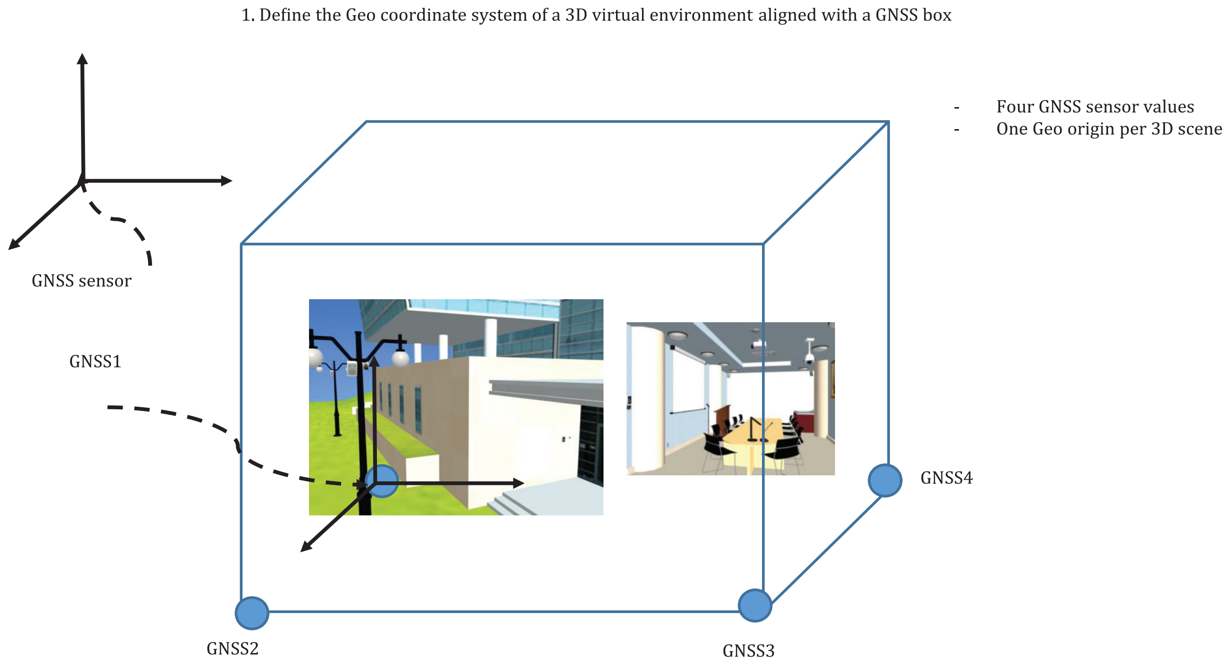


Figure 20 — Definition of a geo-coordinate system per 3D MAR scene

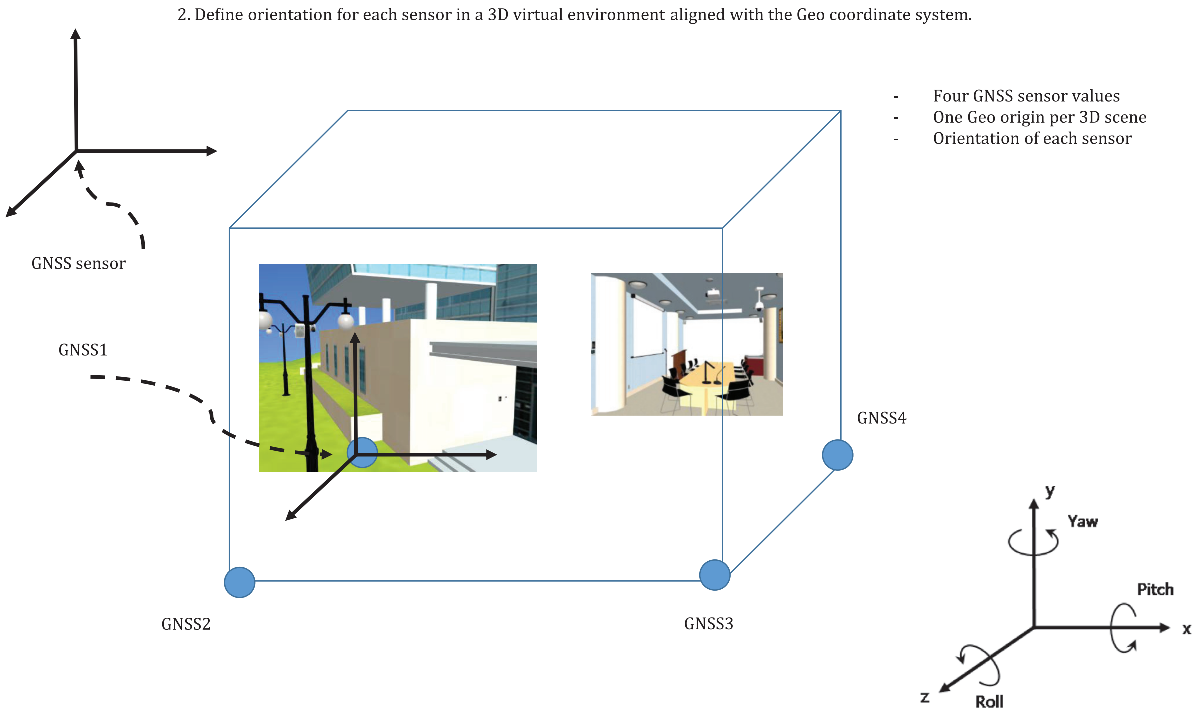


Figure 21 — Definition of orientation for an IoT sensor

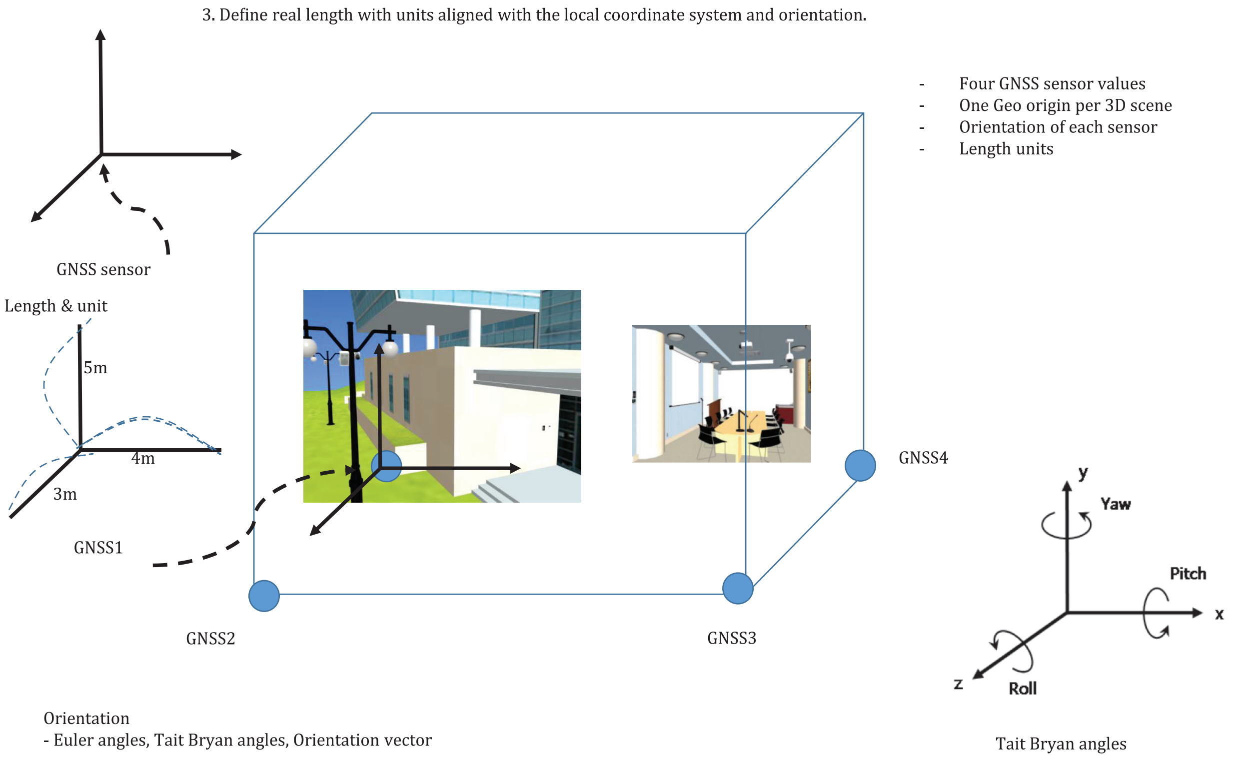


Figure 22 — Definition of real length and units for an IoT sensor

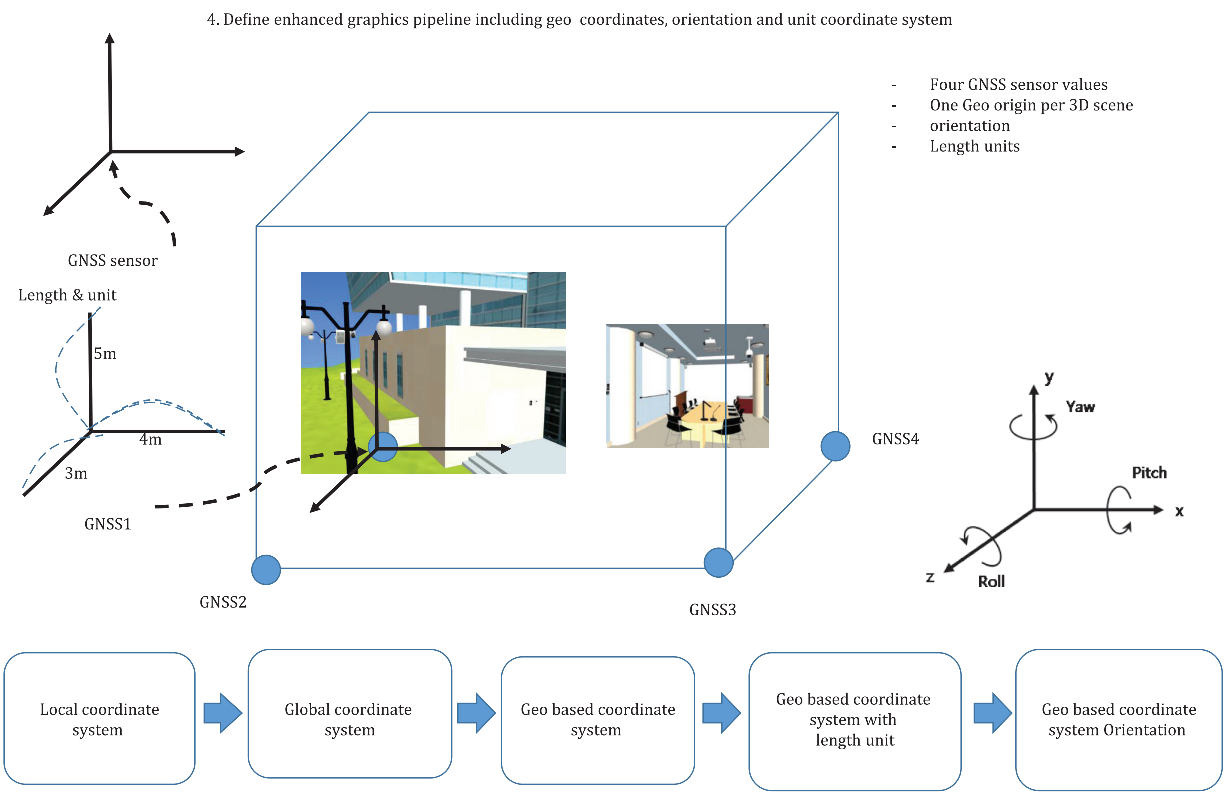


Figure 23 — Enhanced graphics pipeline processing

4.4.3 Sensor properties and interface

After precisely defining a physical sensor's location and orientation, the physical sensor's properties and interface should also be represented. Properties of a sensor are presented functionally in a 3D scene. They can include visual and/or functional information. The properties of the sensor can be represented visually in the scene. However, there can be sensors that do not need visual information, for example, a sound sensor. In this case, only the sound function is represented in the scene. The interface of the physical sensor is defined to provide connection information necessary for importing sensor streaming data from the real world.

4.4.4 Sensor representation data model

A 3D scene is created using a scene graph[9][10]. In order to represent sensors with their functionalities in a 3D scene, the output of graphics pipeline processing for 3D sensors and a 3D scene, and physical properties, should be able to be exchanged between applications and over a network. A 3D data model for sensors can be defined as an extended 3D scene graph. Clause 5 describes the organization of the sensor 3D scene graph.

5 Sensor 3D scene graph

5.1 Definition of a sensor 3D scene graph

A 3D MAR scene that includes sensors is created using a scene graph which consists of 3D scene components and appearance properties. Components include 3D objects and sensors in the 3D MAR scene. The organization of a sensor MAR scene is as follows.

- 3D virtual world representation: define a 3D virtual world including real sensor information processing and visualization;
- GNSS synchronized 3D virtual world: define the data model of a GNSS synchronized 3D virtual world, augmented with real location and orientation information;
- physical sensor devices and properties: define the data model of representing sensor devices and their functional properties in a 3D virtual world;
- interfaces with physical sensor devices: define interfaces for sensor information processing using sensor stream data. In addition, define a connection to or access information for a physical sensor device.

A data structure necessary for organizing a sensor MAR scene is defined as follows:

```
- An MAR scene with location and orientation
------- Location (GNSS origin)
------- Location bounding box (four GNSS locations on the bottom), lengths and units
------- Orientation (pitch-yaw-roll)
------- MAR object
---------------------3D object
-----------------------------Shape
------------------------------------materials
------------------------------------geometry
------------------------------------interfaces with virtual worlds
--------------------Physical sensor
-----------------------------Shape
------------------------------------materials
------------------------------------geometry
------------------------------------interfaces with virtual worlds
-----------------------------Physical sensor type
-----------------------------Physical properties
-----------------------------Interfaces with real worlds
--------------------------- orientation (pitch-yaw-roll)--
```

The 3D scene graph for a 3D MAR scene is represented in Figure 24.

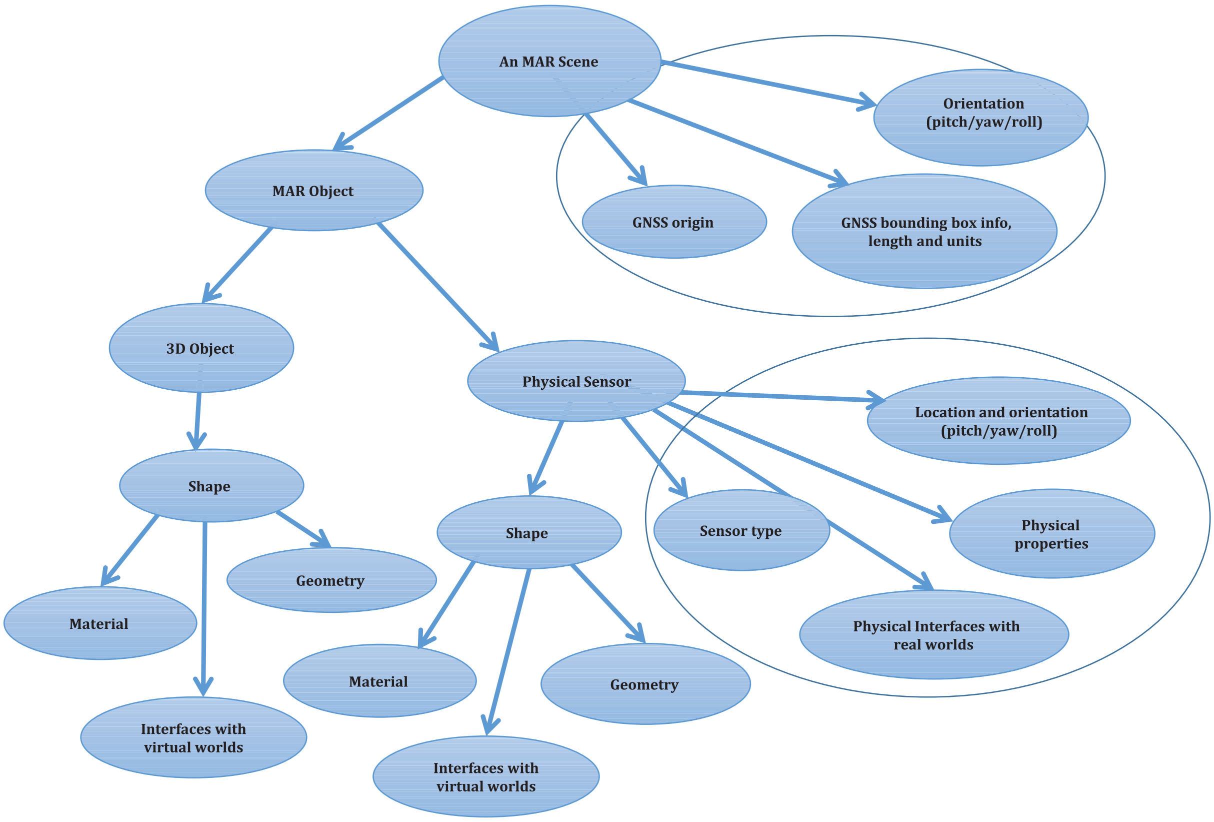


Figure 24 — A sensor 3D MAR scene graph

The hierarchy in Figure 24 represents the data structure of an MAR 3D scene that consists of a set of MAR objects, a GNSS origin, a GNSS bounding box with length and units and orientation. An MAR object is composed of a 3D object and/or physical sensors. The 3D object for the MAR object is located with its location and orientation information in the MAR scene. The shape information of a 3D object and/or a physical sensor includes appearance information. Appearance refers to any sensory information recognized for the object. For example, one MAR object can be represented with its visual shape while another one can be represented with auditory information. Visual appearance is the same as graphical object representation which includes geometry and materials. The GNSS origin is defined for the location of a 3D MAR scene which is a copied real world. For example, a 3D scene of a building or seminar room in the real world has a GNSS origin with its location. For each 3D scene, a GNSS bounding box at the bottom of the space is defined for locating the 3D scene with the corresponding GNSS location.

If an MAR object has a physical sensor, it is defined by its location and orientation, a sensor type, physical properties and interfaces with real worlds. The physical properties include information necessary for import/export of signal data to/from external physical sensor devices. The interfaces with real worlds include connection information necessary for recognizing the sensor device in a network environment.

5.2 Physical properties and interfaces with real worlds

5.2.1 General

In order to represent the function of a physical sensor in a 3D virtual space, the physical properties and the connection information of the physical sensor are controlled and managed. The MAR scene structure includes this information in relation to the physical sensor type and other 3D objects in the scene.

5.2.2 Physical properties of a physical sensor

A physical sensor represents the presence of a physical sensor device and its functions in an MAR scene. The physical sensor device is defined by its properties such as UUID[1], name, event type, control type, and description (Figure 25). The “UUID” field is used to uniquely identify the physical sensor, “name” is the name of the physical sensor, “event type” is the type of input stream from the device, “control type” is the type of control information sent to the device and “description” is the description of the device. Table 1 summarizes the fields necessary to represent the physical properties of a physical sensor device.

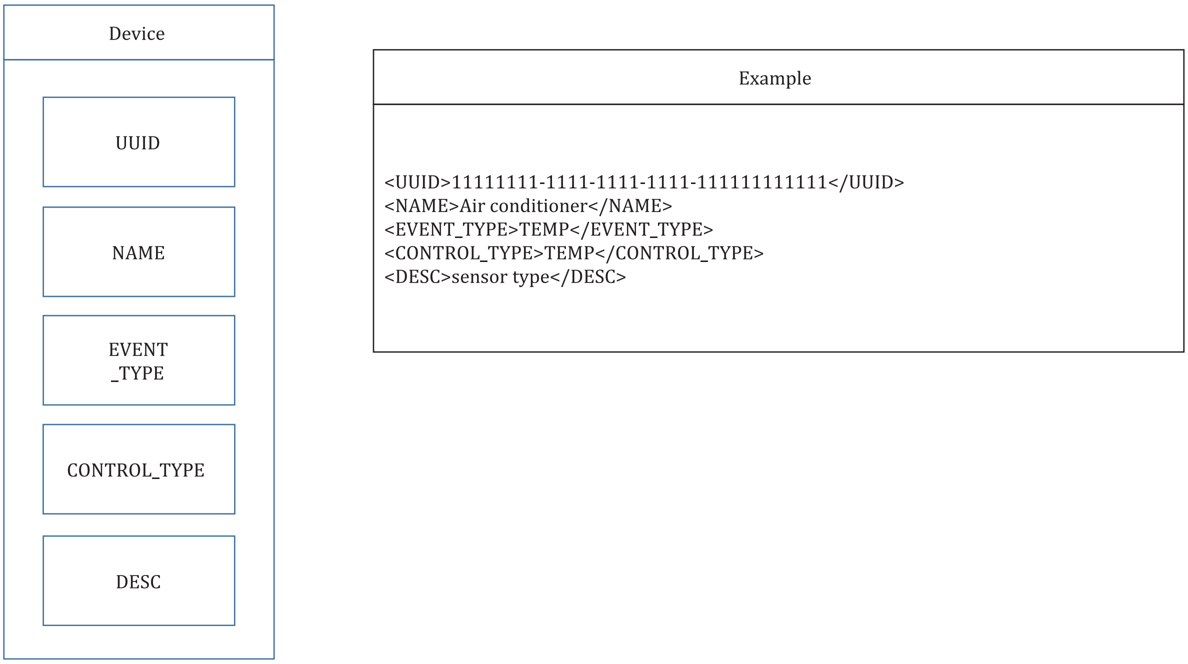


Figure 25 — Physical properties of a physical sensor

Table 1 — Physical sensor device description

Device info fields	Device properties
UUID	Unique ID for recognizing a device
NAME	Device name
EVENT_TYPE	Available data type that can be sent from a physical sensor device
CONTROL_TYPE	Available data type that can send to a physical sensor device
DESC	Additional description of a physical sensor device

5.2.3 Physical interfaces with real worlds of a physical sensor

In the MAR scene structure, physical interfaces with real worlds refers to the definition of connection information from the device to the scene by an application. Relevant fields include name, description, IP, port, ID, password and protocol (Figure 26). The “name” field provides the name of the connection, “desc” describes the connection information, “IP” is the IP address for the device, “port” is the port number for the device connection, “ID” is a user identity, “password” is the corresponding password of the user and “protocol” is the protocol used for the connection to the device. Table 2 summarizes the fields necessary for the physical interfaces of a physical sensor device.

Figure 26 — Physical interfaces with real worlds

Table 2 — Physical interfaces with real worlds

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 identity for accessing a physical sensor device
PASSWORD	User identity password for accessing a physical sensor device
PROTOCOL	Communications protocol

5.2.4 A data structure for the physical properties and interfaces for a physical sensor

Depending on the application, various sensor types can be included when organizing an MAR scene using physical sensors. Sensor types can be defined by expanding the basic data structure of the physical properties and interfaces. This document specifies an abstract data structure for the physical properties and interface in the scene graph while details about specific physical sensors can be defined by user applications. For example, consider an application that has 3 physical sensors: thermometer, RFID (radio frequency identification), and PLC (programmable logic controller) (Figure 27). In this example, user applications can expand the three physical sensor types to detailed sensor data of temperature, humidity, digital input, digital output, analog input, etc.[14].

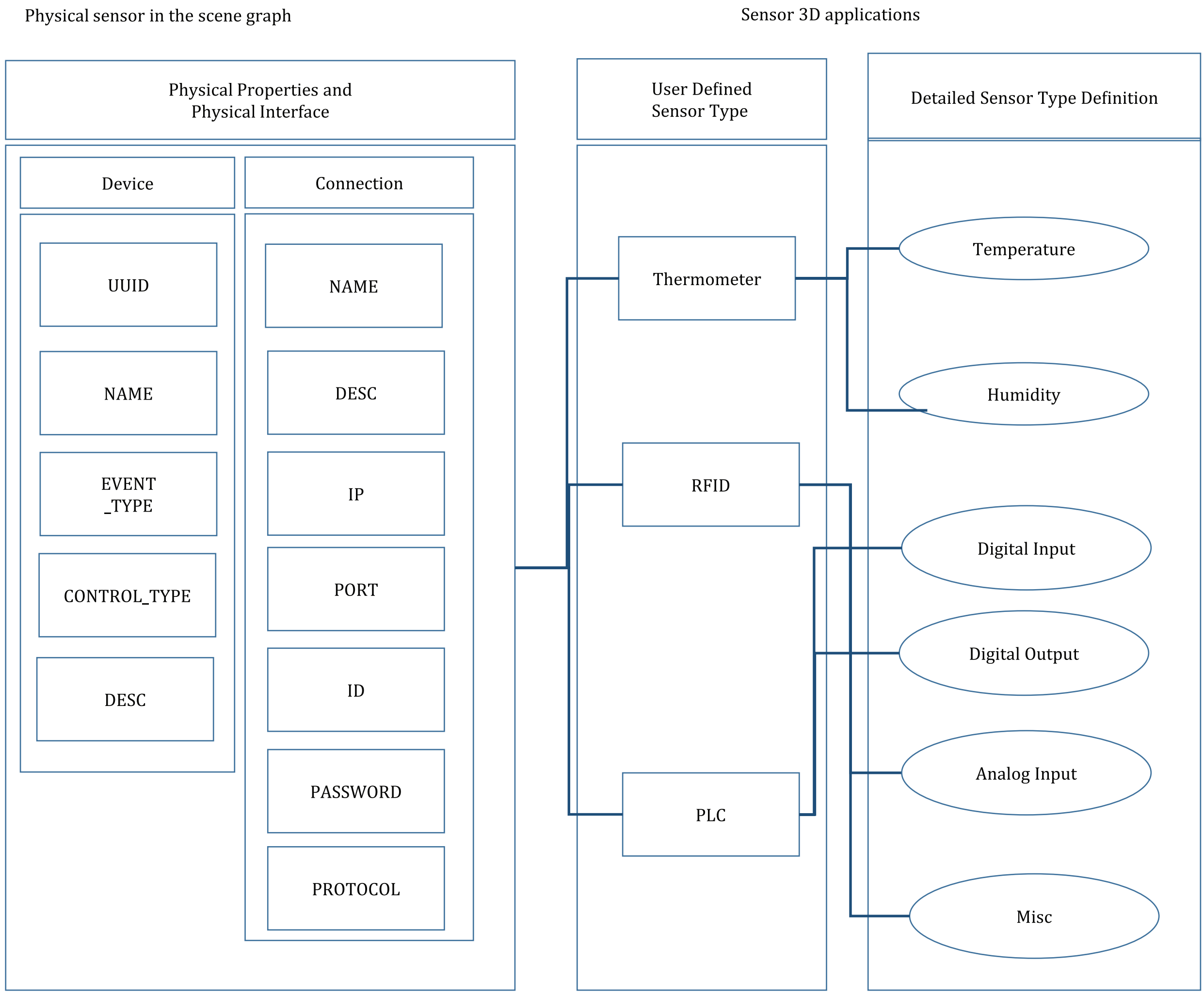


Figure 27 — Relationship between physical properties and user applications

Figure 28 shows the data structure for the physical properties and interfaces of physical sensor devices.

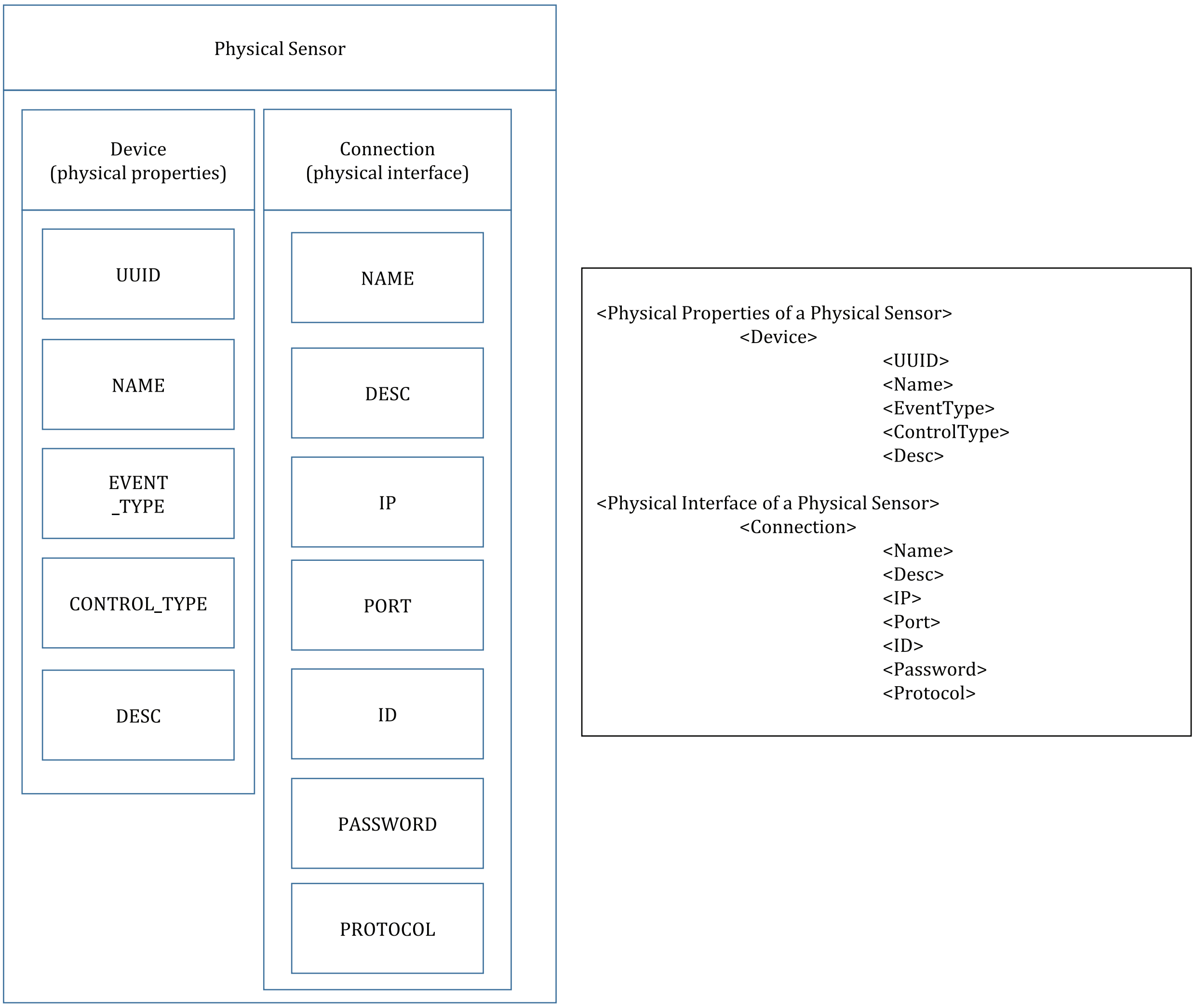


Figure 28 — Data structure of physical properties and interfaces for a physical sensor

6 System architecture for physical sensor representation

6.1 System architecture for physical sensors

When a 3D MAR scene is generated, an MAR scene graph manager controls all necessary data for the scene. The 3D scene can be stored in an MAR data file. The file contains all data for generating and transferring a 3D MAR scene through a network. The MAR data file specifies a 3D MAR scene using the following components:

— MAR data file;

— MAR parser;

— MAR object manager;

— MAR scene graph manager;

— MAR scene and event graph;

— MAR scene access interface;

— event controller;

— physical sensor interface.

Figure 29 shows the overall architecture for physical sensor representation in a 3D scene. A parser reads the data from an MAR data file and interprets it as various MAR objects. MAR objects consist of visual objects and sensor objects that are controlled by an MAR object manager. Events generated by an external device program are transferred to an event controller through a physical sensor interface. The event controller sends the events to be represented by the scene to a sensor object that is controlled by the MAR object manager. An MAR scene access interface receives and processes event data so that sensor events obtained from the event controller can be represented in the scene. In Figure 29, one of the arrows leading from the event controller indicates that an event affects a visual object, and the other arrow indicates that an event is represented independently of any visual object in a scene. In both cases, events can be represented in an MAR scene. An MAR scene graph manager generates and processes a scene graph which is defined with geometry, properties, and events using visual objects, sensor objects, and sensor events. When a change occurs in a scene from a user interface, it is transferred to the event controller by the MAR object manager and then to the external device program.

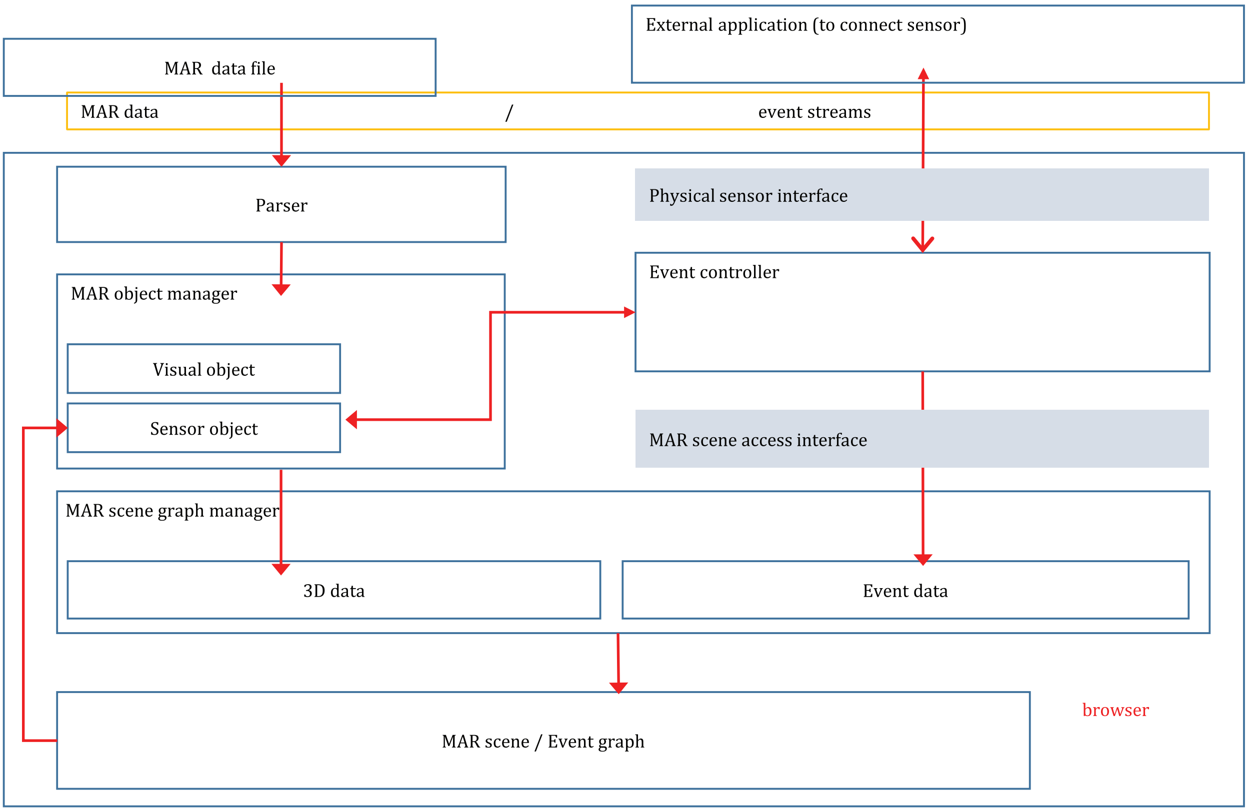


Figure 29 — System architecture for physical sensor representation

6.2 System framework

6.2.1 General

Conceptually, the system of an MAR world with physical sensors includes five components necessary for processing the representation and the functions of physical sensors in a 3D virtual world (Figure 30).

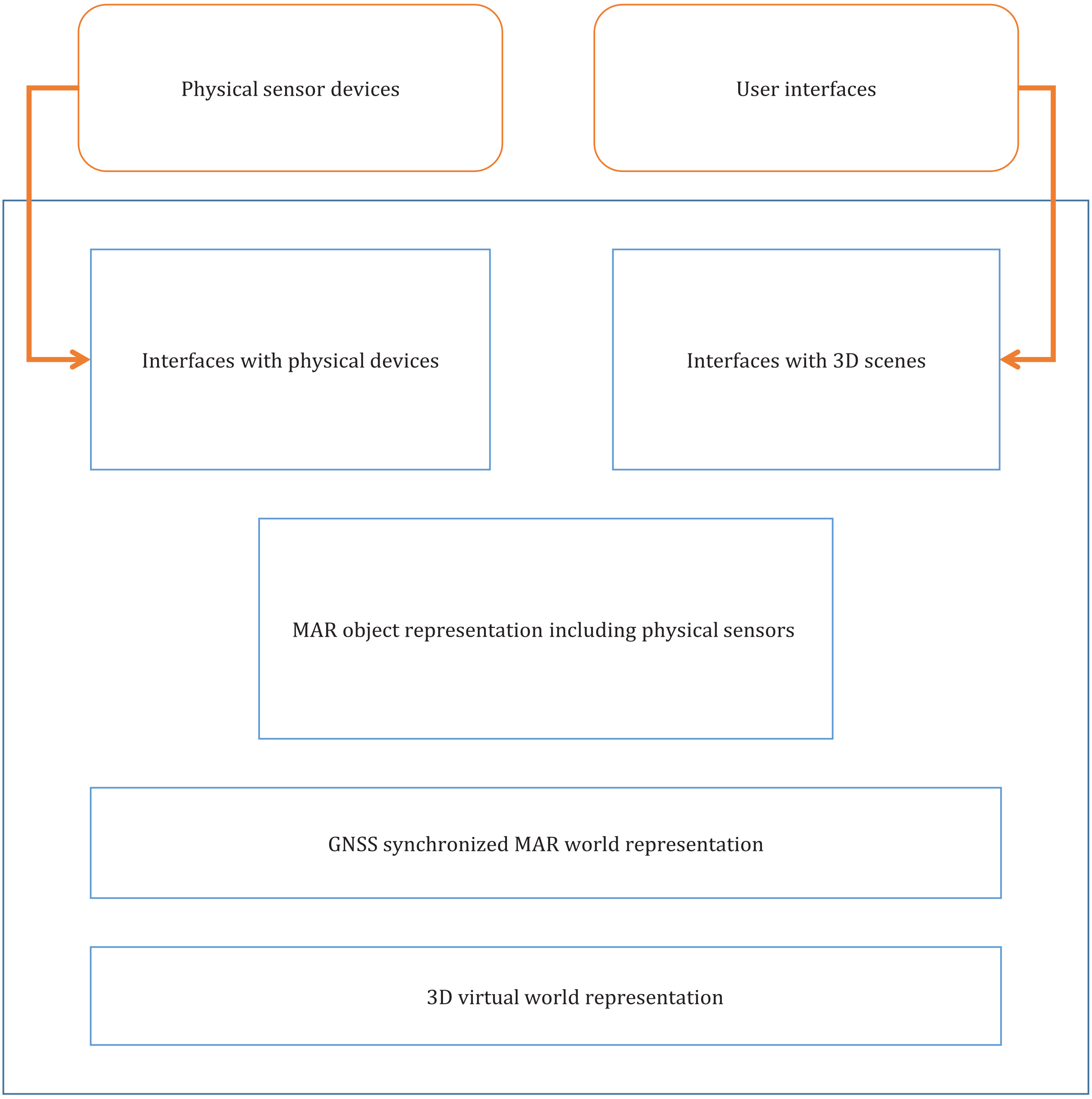


Figure 30 — System framework for an MAR world with physical sensors

6.2.2 3D MAR world representation

An MAR world with physical sensors is organized using a 3D virtual world simulating a real-world application. 3D modelling and rendering functions are basic to an MAR system and should be included. A 3D virtual world that simulates a real world should be defined first and 3D modelling and rendering functions for physical sensors in the world should also be provided. The 3D virtual world model can be of any size, such as the size of a room, a building, a street or a city, depending on the application.

6.2.3 GNSS synchronized 3D virtual worlds

A 3D virtual world that simulates a real world should be synchronized with GNSS information so that all objects in the real world can be placed at their correct locations in the 3D virtual world[19]. The reason for using GNSS information for all objects in a 3D virtual world is to define and to recognize real-world objects with their precise locations in the 3D virtual world. A method of generating a GNSS synchronized 3D virtual world should be provided. In addition to the overall GNSS synchronized world, all 3D objects, including physical sensors, can have their own GNSS information if necessary. GNSS and orientation information should be able to be defined per object in the MAR world.

6.2.4 Sensor devices and their properties

The appearance of a physical sensor device is represented with a 3D object in an MAR world. In addition, its properties should also be represented, including visual or other sensory properties and physical properties. There are many types of physical sensors in the real world. Each physical sensor should be able to be represented as a physical sensor type along with its appearance and physical properties.

6.2.5 Interfaces with 3D sensor objects

User interfaces for manipulating and controlling physical sensors should be included in the MAR world. These affect visual and other sensual information about physical sensors in the MAR world. With user interfaces, the appearance of physical sensors can be variously represented.

6.2.6 Interfaces with physical sensor devices

Interfaces for importing and exporting information to and from physical sensor devices should be represented in the MAR world. A mechanism for receiving events from each sensor device should be provided in order to represent physical properties of the sensor in the MAR world. All events from physical sensor devices should be able to be represented.

7 XML definition of physical sensor representation

7.1 Structure of mixed and augmented reality scene

7.1.1 MARScene

MARScene consists of a set of MARObjects, GNSS information and location and orientation information.

7.1.2 MARObject

MARObject consists of a 3D object and physical sensors. Physical sensor and GNSS information are optional.

7.1.3 3D object

A 3D object has a shape.

7.1.4 Shape

Shape consists of Material, Geometry and InterfaceWithVirtualWorlds. Physical properties are defined differently depending on each sensor type.

7.1.5 Physical sensor

Physical sensor consists of Shape, Sensor type, Physical properties, Physical interface, and Location and Orientation.

7.1.6 Sensor type

Sensor type is an abstract super-type of all sensor types that includes properties common to all physical sensors.

7.1.7 Physical properties

Physical properties contain child elements including UUID, Name, EventType, ControlType, and Desc.

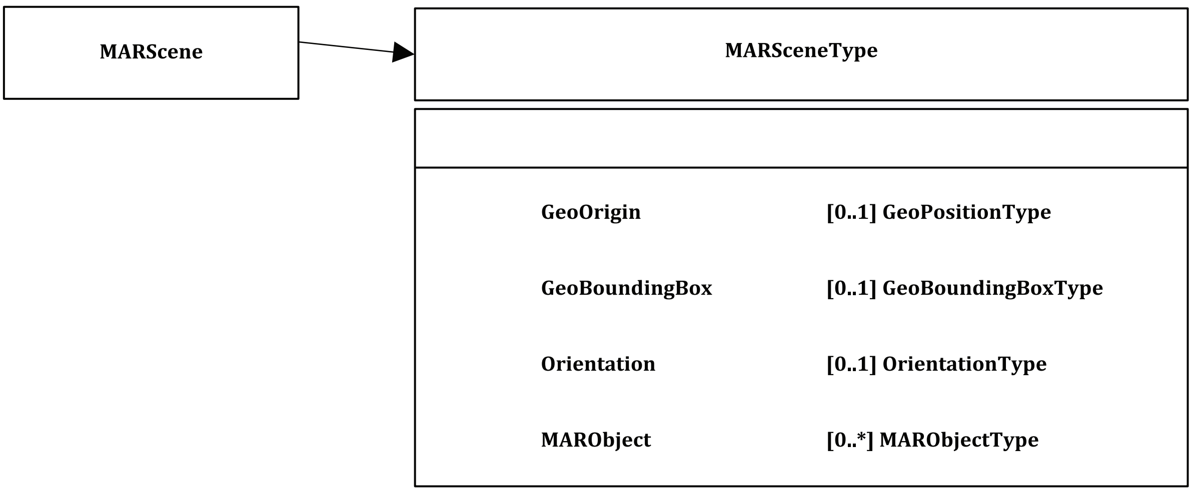
7.1.8 Physical interface

Physical interface has a Connection element which contains child elements including Name, Desc, IP, Port, ID, Password, and Protocol.

7.2 XML schema definition

7.2.1 MARSceneType

MARSceneType consists of Location (origin, bounding box), Orientation and a set of MARObjects (Figure 31).

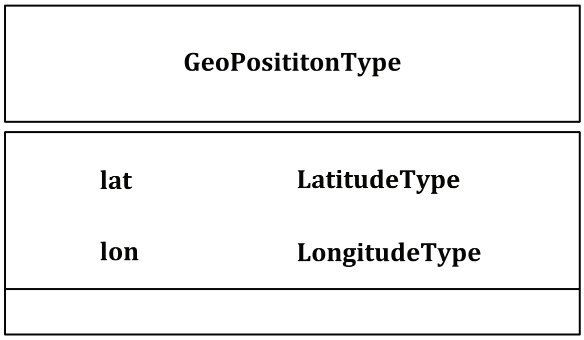



```
<complexType name="MARSceneType">
  <sequence>
     <element name="GeoOrigin" type="mar:GeoPositionType" minOccurs="0" />
     <element name="GeoBoundingBox" type="mar:GeoBoundingBoxType"
     minOccurs="0"/>
     <element name="Orientation" type="mar:OrientationType" minOccurs="0" />
     <element name="MARObject" type="mar:MARObjectType" minOccurs="0"
     maxOccurs="unbounded" />
  </sequence>
</complexType>
```

Figure 31 — MARScene type

7.2.2 GeoPositionType

GeoPositionType represents a GNSS position consisting of latitude and longitude values (Figure 32).

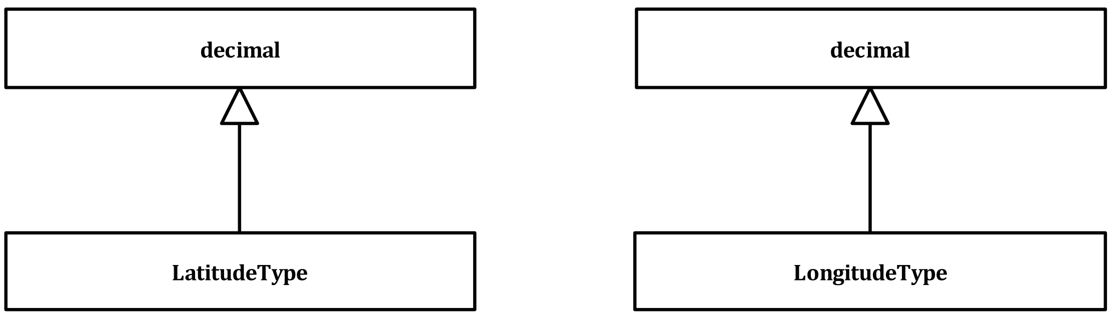



```
<complexType name="GeoPositionType">
    <annotation>
        <documentation> The latitude and longitude of a geo-position
    </documentation>
    </annotation>
    <attribute type="mar:LatitudeType" name="lat" use="required" />
    <attribute type="mar:LongitudeType" name="lon" use="required" />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

Figure 32 — GeoPosition type

7.2.3 LatitudeType and LongitudeType

Latitude and longitude types are defined by two values respectively (Figure 33).

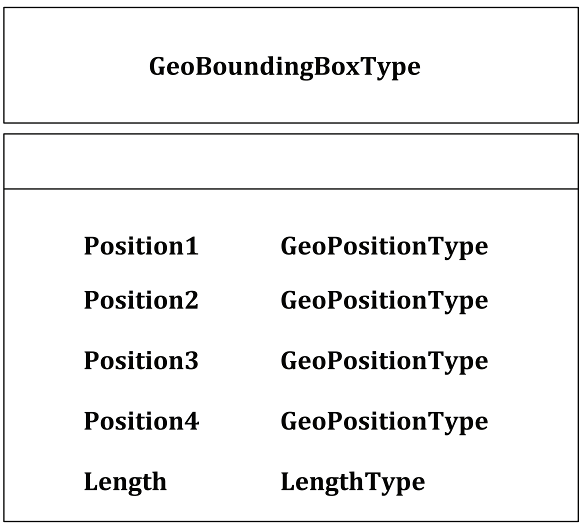



```
<simpleType name="LatitudeType">
    <annotation>
        <documentation> The latitude is in a decimal degree and in a WGS84 datum.
        </documentation>
    </annotation>
    <restriction base="decimal">
        <minInclusive value="-90.0" />
        <maxInclusive value="90.0" />
    </restriction>
</simpleType>
<simpleType name="LongitudeType">
    <restriction base="decimal">
        <minInclusive value="-180.0" />
        <maxInclusive value="180.0" />
    </restriction>
</simpleType>
```

Figure 33 — Latitude and longitude types

7.2.4 GeoBoundingBoxType

GeoBoundingBoxType is defined by 4 geo-positions and 3 length values with a common unit (Figure 34).




```
<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>
```

Figure 34 — GeoBoundingBox type

7.2.5 LengthType

LengthType is defined by three length values with a unit (Figure 35).

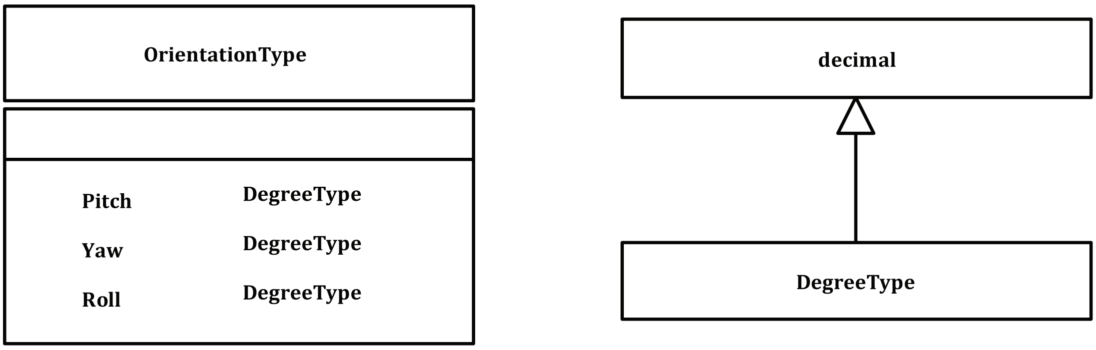
LengthType	
x	**float**
y	**float**
z	**float**
unit	**(unitType)**

```
<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>
```

Figure 35 — Length type

7.2.6 OrientationType

OrientationType is defined by three decimal degrees (pitch, yaw, roll) (Figure 36).

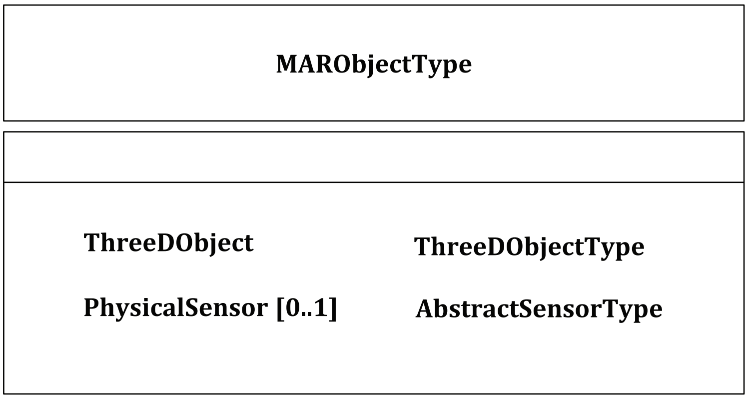


```
<complexType name="OrientationType">
     <sequence>
          <element name="Pitch" type="mar:DegreeType" />
          <element name="Yaw" type="mar:DegreeType" />
          <element name="Roll" type="mar:DegreeType" />
     </sequence>
</complexType>
<simpleType name="DegreeType">
     <restriction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

Figure 36 — Orientation type

7.2.7 MARObjectType

MARObjectType consists of ThreeDObject and PhysicalSensor elements (Figure 37). PhysicalSensor is optional.

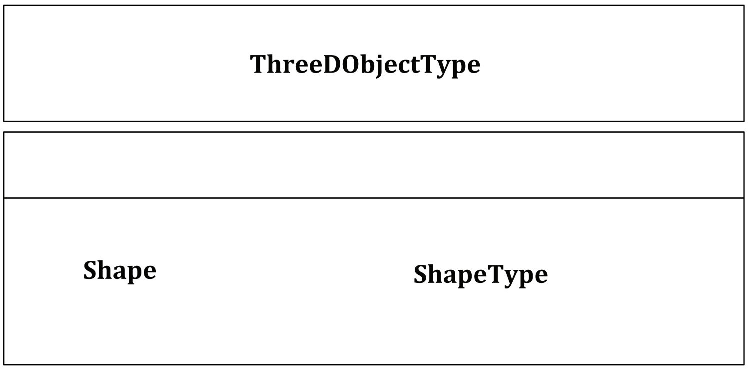



```
<complexType name="MARObjectType">
     <sequence>
          <element name="ThreeDObject" type="tns:ThreeDObjectType" />
          <element name="PhysicalSensor" type="tns:AbstractSensorType"
              maxOccurs="1" minOccurs="0" />
     </sequence>
</complexType>
```

Figure 37 — MARObject type

7.2.8 ThreeDObjectType

ThreeDObjectType consists of a Shape element (Figure 38).




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

Figure 38 — 3D object type

7.2.9 ShapeType

ShapeType consists of Material, Geometry and InterfaceWithVirtualWorlds elements (Figure 39).

ShapeType		
Material	**[0..*]**	**MaterialType**
Geometry	**[0..*]**	**GeometryType**
InterfaceWithVirtualWorlds	**[0..*]**	**InterfaceWithVirtualWorldsType**

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

Figure 39 — Shape type

7.2.10 AbstractSensorType

AbstractSensorType is a super-type for all sensor types. It includes properties of sensors common to all sensors (Figure 40). Id is an identifier for each sensor. “Activated” denotes the active status of a sensor. Shape is geometry data. SensorType has the following enumeration values: {Type1, Type2, ..., Type15}. PhysicalProperties has common properties. Physical properties of each sensor are defined per sensor type. Physical interface has an interface with the real world. Orientation has pitch, yaw, and roll information.

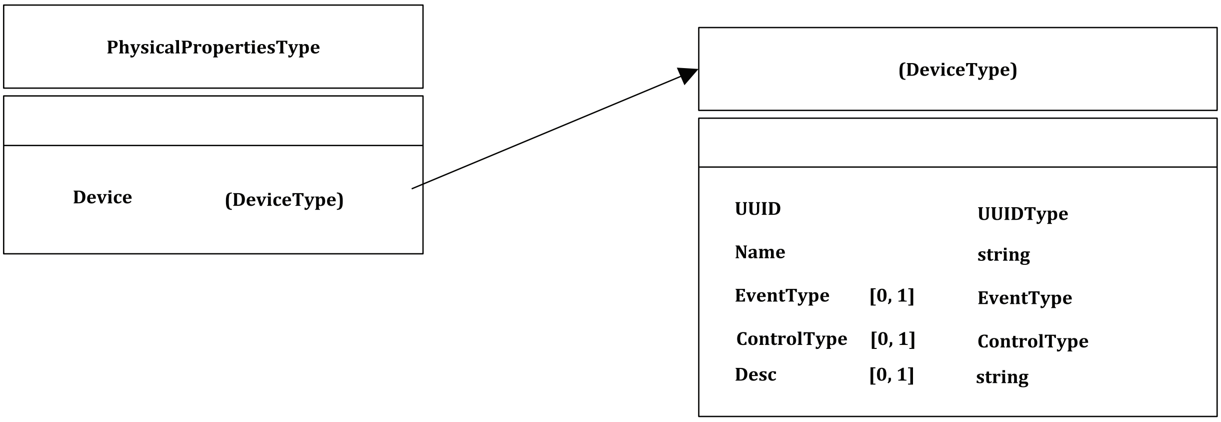
AbstractSensorType		
id		**ID**
activated		**boolean**
SensorType		**(SensorTypeType)**
Shape	**[0, 1]**	**ShapeType**
PhysicalProperties	**[0, 1]**	**PhysicalPropertiesType**
PhysicalInterface	**[0, 1]**	**PhysicalInterfaceType**
Orientation	**[0, 1]**	**OrientationType**

```
<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" minOccurs="0"/>
      <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>
```

Figure 40 — Abstract sensor type

7.2.11 PhysicalPropertiesType

PhysicalPropertiesType has a Device element that contains child elements including UUID, Name, EventType, ControlType, and Desc (Figure 41).

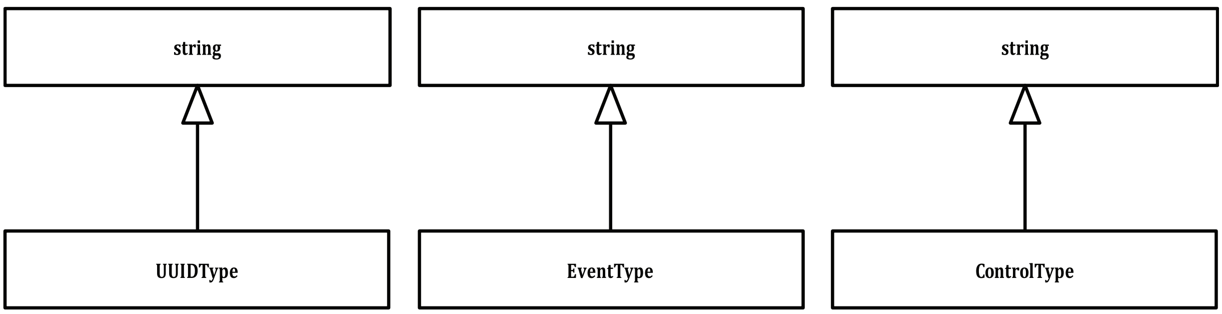



```
<complexType name="PhysicalPropertiesType">
   <sequence>
       <element name="Device">
          <complexType>
             <sequence>
                <element name="UUID" type="mar:UUIDType" />
                <element name="Name" type="string" />
                <element name="EventType" type="mar:EventType"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

Figure 41 — Physical properties type

7.2.12 UUIDType, EventType and ControlType

UUIDType, EventType and ControlType are defined as in Figure 42.

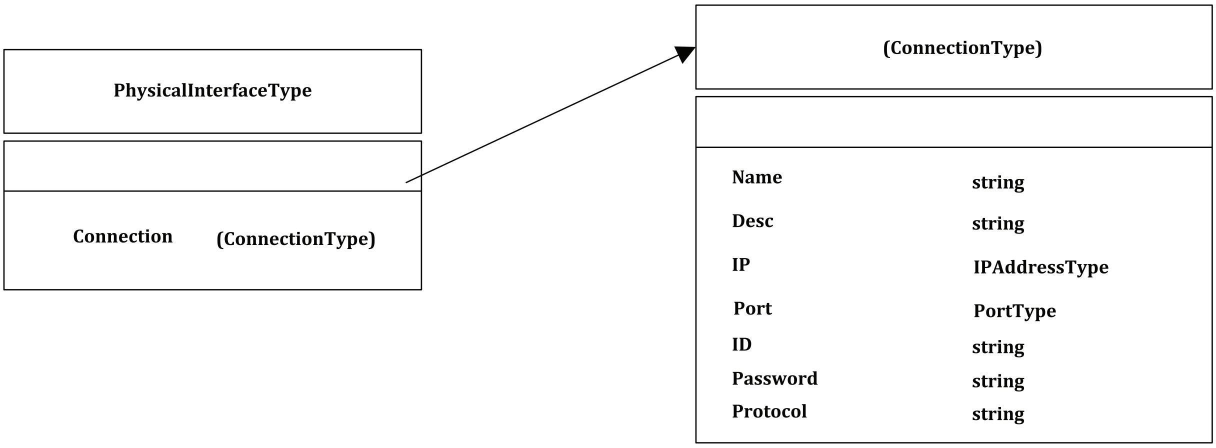


```
<complexType name="PhysicalInterfaceType">
  <sequence>
    <element name="Connection">
      <complexType>
        <sequence>
          <element name="Name" type="string" />
          <element name="Desc" type="string" minOccurs="0" />
          <element name="IP" type="mar:IPAddressType" />
          <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>
```

Figure 42 — UUID, Event, and Control types

7.2.13 PhysicalInterfaceType

PhysicalInterfaceType has a Connection element that contains child elements including Name, Desc, IP, Port, ID, Password, and Protocol (Figure 43).

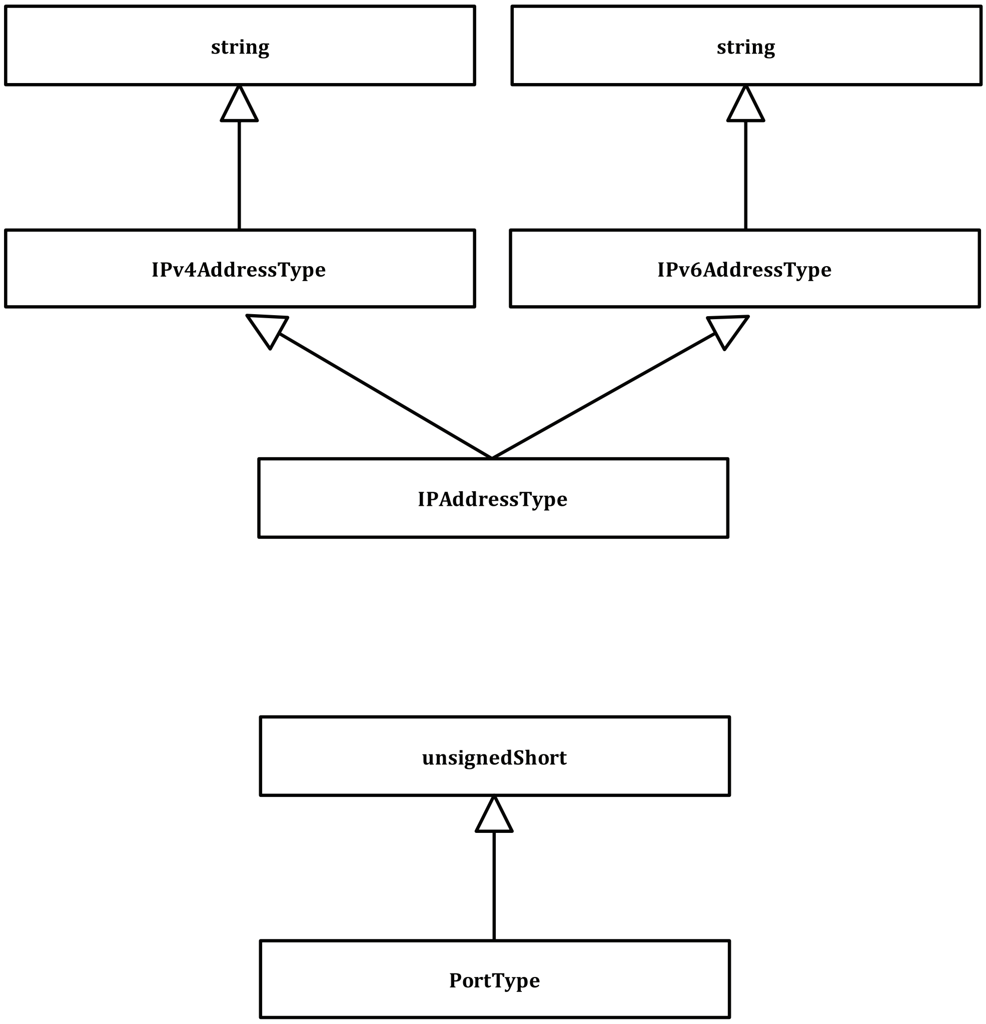



```
<complexType name="PhysicalInterfaceType">
  <sequence>
    <element name="Connection">
      <complexType>
        <sequence>
          <element name="Name" type="string" />
          <element name="Desc" type="string" minOccurs="0" />
          <element name="IP" type="mar:IPAddressType" />
          <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>
```

Figure 43 — Physical Interface type

7.2.14 IPAddressType and PortType

IPAddressType and PortType are defined as in Figure 44.



```
<simpleType name="IPAddressType">
  <union memberTypes="mar:IPv4AddressType mar:IPv6AddressType" />
</simpleType>
<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="IPv6AddressType">
  <restriction base="string">
    <pattern value="(([0-9a-fA-F]{1,4}:){7,7}[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:)
{1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,5}(:[0-9a-fA-F]
{1,4}){1,2}|([0-9a-fA-F]{1,4}:){1,4}(:[0-9a-fA-F]{1,4}){1,3}|([0-9a-fA-F]{1,4}:){1,3}
(:[0-9a-fA-F]{1,4}){1,4}|([0-9a-fA-F]{1,4}:){1,2}(:[0-9a-fA-F]{1,4}){1,5}|[0-9a-fA-F]
{1,4}:((:[0-9a-fA-F]{1,4}){1,6})|:((:[0-9a-fA-F]{1,4}){1,7}|:)|fe80:(:[0-9a-fA-F]{0,4})
{0,4}%[0-9a-zA-Z]{1,}|::(ffff(:0{1,4}){0,1}:){0,1}((25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}
[0-9]).){3,3}(25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])|([0-9a-fA-F]{1,4}:){1,4}:((25[0-
5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9]).){3,3}(25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9]))" />
  </restriction>
</simpleType>
<simpleType name="PortType">
    <restriction base="unsignedShort">
        <minInclusive value="1" />
    </restriction>
</simpleType>
```

Figure 44 — IPAddress and Port types

Annex B defines an XML schema to represent the sensor-based MAR scene graph. Annex C contains an example of schema expansion for representing the physical properties of sensor types based on the schema defined in Annex B. Annex D shows an example of representing sensors in an MAR virtual environment using XML based on the sensor MAR schema defined in Annexes B and C. Annex E contains two use cases as examples of simulating and visualizing the functions of sensors using a 3D scene.

8 Conformance

8.1 Conformance criteria

The conformance of sensor representation is focused on defining common usage and exchange of sensor information in a 3D scene between applications and between heterogeneous computing environments. The following points should be considered:

— sensor information data using 3D scenes specifies location with orientation, appearance, and functional parameters of each sensor;

— sensors are classified based on their representative sensor types. Each sensor is recognized as a specific sensor type along with its functional parameters;

— a sensor is defined as an object that consists of appearance data and incoming and outgoing signals or streams for representing functional parameters;

— sensors and 3D data are organized based on the hierarchy of the MAR data structure with sensors that can be represented and recognized in a 3D scene.

8.2 Conformance area

Conformance should be considered for the following components, including sensor and 3D information.

— sensor information data sets including 3D environment data that is generated and transferred;

— 3D sensor viewers that display sensors and 3D environments;

— 3D sensor editors that generate, display, simulate and modify 3D scenes with sensors.

Annex A
(informative)

Examples of physical sensor types and parameters

A.1 General

This annex includes examples of physical sensor types and parameters with units and data type.

For each sensor type below, typical parameters for controlling that type of sensor are shown in the corresponding table. User interfaces should be provided for these, including controls for turning the sensor on/off and for initiating sensor type-specific actions.

For each sensor type, the physical functions of the sensor can be represented and simulated with parameters, as referenced in the following typical procedure:

— in a 3D scene, turn on the sensor using a control;

— represent and change the parameter values of the sensor using various types of controls;

— display changes to the scene based on changing parameter values; and

— turn off the sensor using a control.

Specifics and deviations from this general procedure are described in the section for the specific sensor type.

A.2 Camera sensor

Typical parameters for controlling camera devices are shown in Table A.1. User interfaces should be provided for these, including controls for camera orientation and zoom. When several camera sensors are used in a scene, controls should also be provided for selecting, displaying and controlling specific camera scenes.

Once the camera device is turned on, typical actions include changing the orientation of the camera and zooming in/out.

In Tables A.1 to A.15, the symbols used are as follows:

— int: integer number;

— float: floating point number;

— string: data type that specifies a sequence of characters in computer programming.

Table A.1 — Parameters of a camera sensor device

Parameter	Unit	Data type
Width	mm	float
Height	mm	float
Diagonal	mm	float
Aspect ratio	—	int
Pixel count	—	int
Megapixels	—	int

Table A.1 *(continued)*

Parameter	Unit	Data type
Horizontal pixels	—	int
Vertical pixels	—	int
Pixel size	μm	float
Frame rate	fps (frames per second)	int
Dynamic range	dB	float
Supply voltage	V	float
Power consumption	mW	int
Operating temperature	°C	float

A.3 Chemical sensor

Typical parameters for controlling chemical sensors are shown in Table A.2.

Once the chemical sensor is turned on, typical actions include controlling the proportion of a gas or a liquid, such as oxygen.

Table A.2 — Parameters of a chemical sensor device

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	—	int
Target gas	—	string
Supply voltage	V	float
Power consumption	mW	int
Operating temperature	°C	int
Operating humidity	%	int

A.4 Electric sensor

Typical parameters for controlling electric sensors are shown in Table A.3.

Once the electric sensor is turned on, typical actions include providing parameter values.

Table A.3 — Parameters of an electric sensor device

Parameter	Unit	Data type
Voltage	V	float
Frequency	Hz	float
Range	mm	int
Operating temperature	°C	int
Operating humidity	%	int
Accuracy	%	float

A.5 Environment sensor

The physical functions of an environment sensor are represented and simulated with parameters (Table A.4) as referenced in the following procedure:

— in a 3D scene, turn on an environment sensor using a control;

— represent the parameter values of the sensor whenever they change according to each sensor device connected; and

— turn off the sensor using a control.

Table A.4 — 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

A.6 Flow sensor

Typical parameters for controlling flow sensors are shown in Table A.5.

Once the flow sensor is turned on, typical actions include representing parameter values as they change.

Table A.5 — 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

A.7 Force sensor

Typical parameters for controlling force sensors are shown in Table A.6.

Once the force sensor is turned on, typical actions include representing parameter values as they change.

Table A.6 — Parameters of a force sensor

Parameter	Unit	Data type
Thickness	mm	float
Length	mm	int
Width	mm	int
Sensing area	mm^2	float

Table A.6 *(continued)*

Parameter	Unit	Data type
Connector	pin	int
Diameter	mm	float
Sensitivity	N (g, kg)	float
Repeatability	%	float
Operating temperature	°C	int

A.8 Light sensor

Typical parameters for controlling light sensors are shown in Table A.7.

Once the light sensor is turned on, typical actions include representing parameter values as they change. With some light devices such as street lights, the 3D scene should change automatically, i.e. the device turns on/off automatically based on the detection of light by the sensor.

Table A.7 — 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

A.9 Movement sensor

Typical parameters for controlling movement sensors are shown in Table A.8.

Once the movement sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, depending on the application, the movement sensor itself may be represented. A 3D object can represent the movement sensor and act like a sensor in the scene. Otherwise, the 3D scene can change according to the resulting views or sound controlled by an interface for detected movement parameters.

Table A.8 — 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	—	string
Weight	g	float
Detection distance	m	float
Field of view (horizontal)	°	float

Table A.8 *(continued)*

Parameter	Unit	Data type
Field of view (vertical)	°	float

A.10 Navigation sensor

Typical parameters for controlling navigation sensors are shown in Table A.9.

Once the navigation sensor is turned on, typical actions include representing parameter values and displaying changes to the scene based on parameter values from changing GNSS information and displaying the path to the changed location. A 3D object that uses a navigation sensor, such as a smart phone or vehicle navigation system, can be represented. Otherwise, a 3D scene will be navigated and displayed according to the view controlled by an interface for navigation parameters.

Table A.9 — 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

A.11 Particle sensor

Typical parameters for controlling particle sensors are shown in Table A.10.

Once the particle sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, particles have commonly been used when representing natural phenomena such as rain and snow. Depending on the application, the particle sensor itself can be represented - a 3D object can represent the particle sensor and act like a sensor in the scene. Otherwise, a 3D scene will change based on the resulting views or sound controlled by an interface for detected particle parameters.

Table A.10 — 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/l, $\mu g/m^3$	int
Supply voltage	V	float
Operating temperature	°C	int
Operating humidity	%	int

A.12 Position sensor

Typical parameters for controlling position sensors are shown in Table A.11.

Once the position sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, modelling of the position sensor itself is not necessary. Rather, it is necessary to simulate the functions of the sensor and display the change in the scene due to the sensor operation. If the position sensor is modelled, a 3D object is recognized and simulated in the scene based on the parameter values of the sensor.

Table A.11 — 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

A.13 Pressure sensor

Typical parameters for controlling pressure sensors are shown in Table A.12.

Once the pressure sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, modelling of the pressure sensor itself is not necessary. Rather, it is necessary to simulate the functions of the sensor and display the change in the scene due to the sensor operation. If the pressure sensor is modelled, a 3D object is recognized and simulated in the scene based on the parameter values of the sensor.

Table A.12 — 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	Ω	float
Output resistance	Ω	float
Pressure range	kPa	float
Supply voltage	V	float
Accuracy	%	float
Sensitivity	V/kPa	float

A.14 Proximity sensor

Typical parameters for controlling proximity sensors are shown in Table A.13.

Once the proximity sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, modelling of the proximity sensor itself is not necessary. Rather, it is necessary to simulate the functions of the sensor and display the change in the scene due to the sensor operation. If the proximity sensor is modelled, a 3D object is recognized and simulated in the scene based on the parameter values of the sensor.

Table A.13 — Parameters of a proximity sensor

Parameter	Unit	Data type
Supply voltage	mV	int
Illuminance measurement range	V	int
Proximity detection range	μW/cm^2	float
Sensitivity variations	%	float
Operating temperature	°C	int

A.15 Sound sensor

Typical parameters for controlling sound sensors are shown in Table A.14.

Once the sound sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, modelling of the sound sensor itself depends on the application. If the scene includes the sensor, its functions should be represented and simulated. The resulting change in the scene due to the sound sensor should be represented and controlled. If the sound sensor is modelled, a 3D object is recognized and simulated in the scene based on the parameter values of the sensor.

Table A.14 — 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

A.16 Temperature sensor

Typical parameters for controlling temperature sensors are shown in Table A.15.

Once the temperature sensor is turned on, typical actions include representing parameter values as they change. In a 3D scene, modelling of the temperature sensor itself depends on the application. If the scene includes the sensor, its functions shall be represented and simulated. The resulting change in the scene due to the temperature sensor should be represented and controlled. If the temperature sensor is modelled, a 3D object is recognized and simulated in the scene based on the parameter values of the sensor.

Table A.15 — 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

Annex B
(informative)

Schema for sensor MAR representation

The schema available at http://standards.iso.org/iso-iec/18038/ed-1/en is a schema defined by XML for the sensor-based MAR scene graph.

Annex C
(informative)

Example XML schema extension for physical sensor representation

The physical properties and I/O streaming data types for a sensor type are variable because individual sensor products can have their own set of parameters. It is necessary to have a minimum of common abstract types for the schema definition and to have schema expansion and data types for all parameters necessary for the specific sensor to be used in an MAR system.

The schema available at http://standards.iso.org/iso-iec/18038/ed-1/en is an example of schema expansion for representing the physical properties of sensor types based on the schema defined in Annex B.

Annex D
(informative)

Example of sensor MAR representation based on the sensor MAR schema

The schema available at http://standards.iso.org/iso-iec/18038/ed-1/en is an example of representing sensors in an MAR virtual environment using XML based on the sensor MAR schema defined in Annexes B and C.

Annex E
(informative)

Implementation examples of sensor MAR representation

E.1 General

In this annex, two use cases are demonstrated as examples of simulating and visualizing the functions of sensors using a 3D space. The first uses indoor sensor devices (camera, environment, light, and sound sensors) and the second uses outdoor sensor devices (camera, light, and proximity sensors).

E.2 Indoor sensor representation

This use case includes a camera sensor, a light sensor, an environment sensor, and a sound sensor. The sensors and their functions are represented in a 3D virtual world. The purpose of the representation is to simulate and monitor the functions of the sensors visually and remotely in a network environment. Figures E.1 to E.4 show scenes with the sensors operating and streaming data.

Details of this use case are as follows.

— A 3D scene represents a real world where sensors are managed and controlled. Sensors are optionally represented as a 3D shape in the scene, depending on the application and sensor type. In this use case, all 4 sensors have 3D shapes, and are represented visually in the scene.

— The functions of the light sensor are visualized in the scene. For example, “On” makes the scene bright based on the intensity of the light sensor. The intensity can be controlled and displayed using a user interface that can send signals to the light sensor.

— The functions of the environment sensor are displayed in the scene. For example, information about temperature and humidity are displayed in the scene in a text form. In other instances, the information can display inside the 3D shape of the sensor device represented in the scene. "On" and "Off" signals from the MAR scene to the environment sensor can be represented in the scene.

— The CCTV camera sensor captures changes in the real world for a security purpose. An online remote system is needed to monitor the functions of the camera sensor visually. The functions of the camera sensor can include capturing and sending video. The system can display the video and send a signal to the camera sensor so that the sensor can stop or change the direction of the CCTV.

— The functions of the sound sensor are represented in the scene. For example, the speaker receiving sound signals through a sound sensor generates the sound with its sound function. Volume control is another example of a function. "On" and "Off" signals from the MAR scene to the sound sensor can be represented in the scene.

— All data, including appearance and function data, for sensors should be structured so as to be stored, used, transferred and exchanged between applications and between heterogeneous computing environments.

— All data generated during the operation of the sensors, including appearances and function data should be able to be managed with the MAR data structure defined in this document.

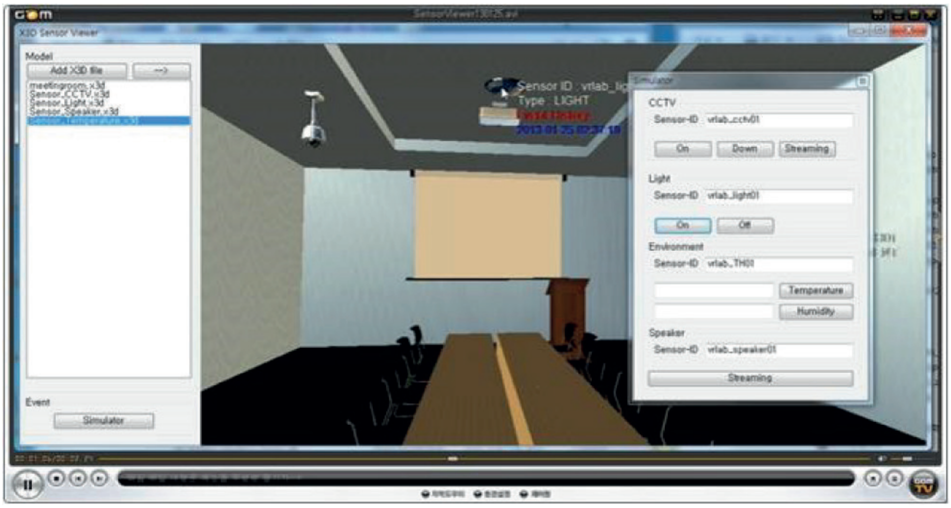


Figure E.1 — Indoor simulated sensors and their representation in a 3D scene: Light sensor

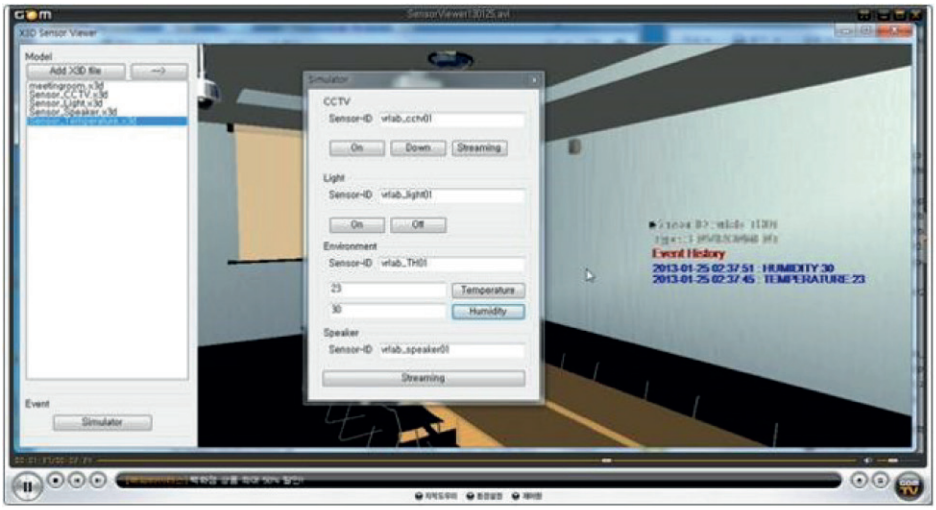


Figure E.2 — Indoor simulated sensors and their representation in a 3D scene: Environment sensor

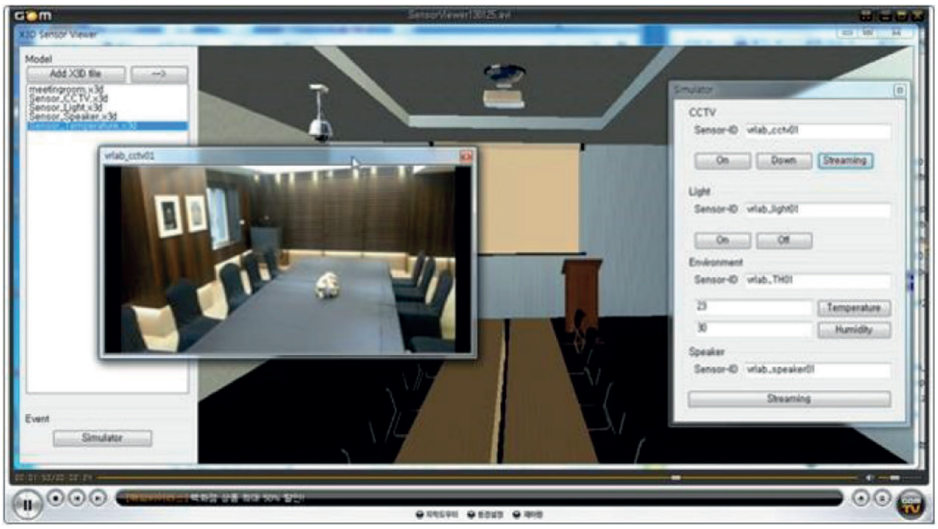


Figure E.3 — Indoor simulated sensors and their representation in a 3D scene: Camera sensor

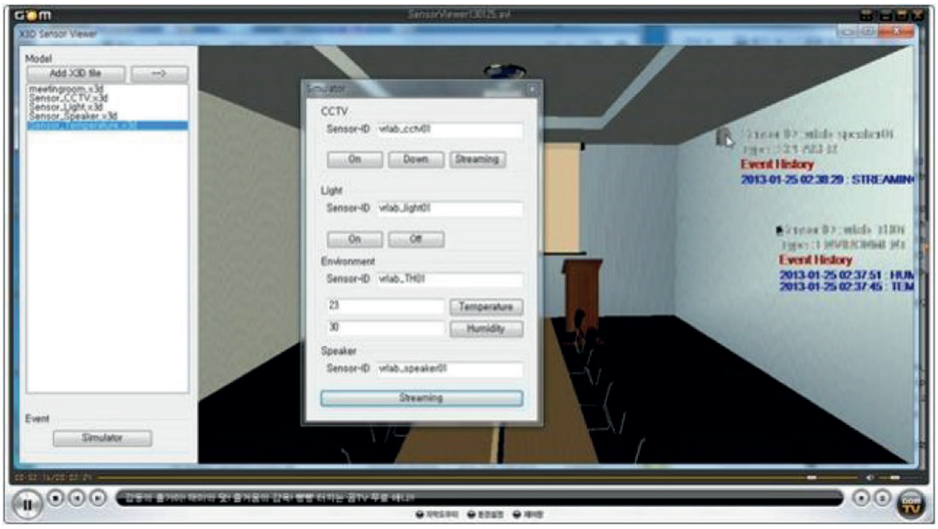


Figure E.4 — Indoor simulated sensors and their representation in a 3D scene: Sound sensor

E.3 Outdoor sensor representation

This use case includes a camera sensor, a light sensor, and a proximity sensor. The sensors and their functions are represented in a 3D virtual world. Figures E.5 and E.6 show scenes with the sensors operating and streaming data.

Details of this use case are as follows.

— A 3D scene represents a real world where sensors are managed and controlled. Sensors are optionally represented as a 3D shape in the 3D scene, depending on the application and sensor type. In this use case, all three sensors have 3D shapes, and are represented visually in the scene.

— The CCTV camera sensor captures changes in the real world for a security purpose. An online remote system is needed to monitor the functions of the camera sensor visually. The functions of the camera sensor can include capturing and sending video. The system can display the video and send a signal to the camera sensor so that the sensor can stop or change the direction of the CCTV.

— The functions of the light sensor are visualized in the scene. There is also an invisible proximity sensor inside the light sensor. Functions for the light sensor such as “On” or “Off” are represented in the scene for the light sensor based on a signal from the sensor device. In addition, the light can be turned on by a proximity sensor when a live object approaches the sensor. The intensity can be controlled and displayed using a user interface that can send signals to the light sensor.

— In addition, there is a visible proximity sensor with a touch sensor device represented in a 3D shape (such as RFID at an entrance door). The functions of this proximity sensor differ from the proximity sensor inside the light sensor. For example, for this proximity sensor, the door opens when a live object approaches the sensor. "On" and "Off" signals from the MAR scene to the environment sensor can be represented in the scene.

— All data, including appearance and function data, for sensors should be structured so as to be stored, used, transferred, and exchanged between applications and between heterogeneous computing environments.

— All data generated during the operation of the sensors, including appearance and function data, should be able to be managed with the MAR data structure defined in this draft.

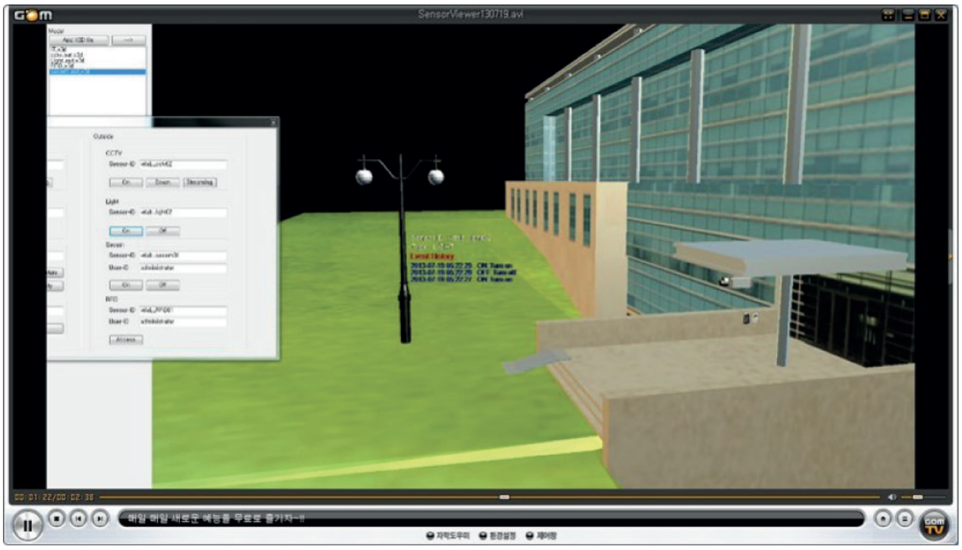


Figure E.5 — Outdoor simulated sensors and their representation in a 3D scene: Camera and light sensors

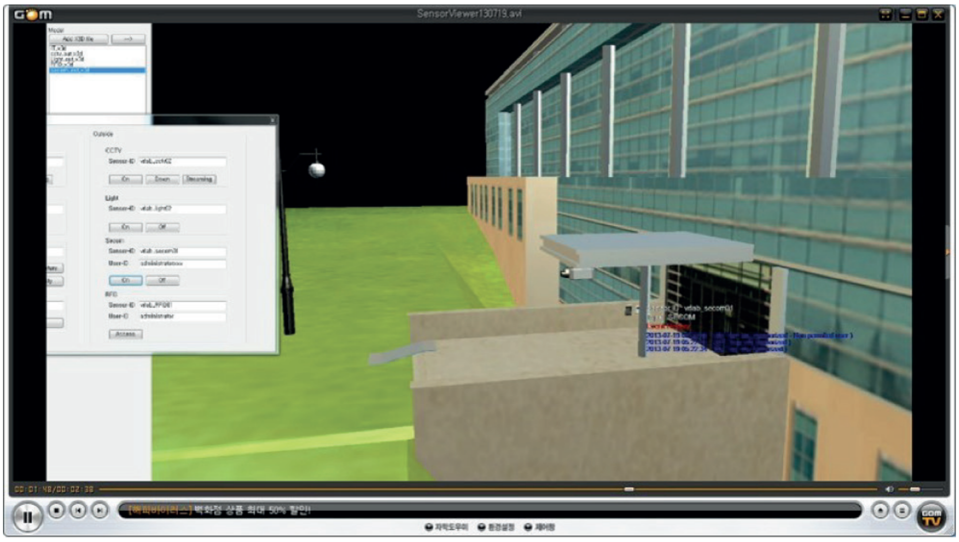


Figure E.6 — Outdoor simulated sensors and their representation in a 3D scene: Proximity and sound sensors

Bibliography

[1] ISO/IEC 11578, *Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)*

[2] ISO/IEC 14772-1, *Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding*

[3] ISO 16739-1, *Industry Foundation Class (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema*

[4] ISO/IEC 18023-1, *Information technology — SEDRIS — Part 1: Functional specification*

[5] ISO/IEC 18023-2, *Information technology — SEDRIS — Part 2: Abstract transmittal format*

[6] ISO/IEC 18025, *Information technology — Environmental Data Coding Specification (EDCS)*

[7] ISO/IEC 18026, *Information technology — Spatial Reference Model (SRM)*

[8] ISO/IEC 18039, *Information technology — Computer graphics, image processing and environmental data representation — Mixed and augmented reality (MAR) reference model*

[9] ISO/IEC 19775-1, *Information technology — Computer graphics, image processing and environmental data representation — Extensible 3D (X3D) — Part 1: Architecture and base components*

[10] ISO/IEC 19775-2, *Information technology — Computer graphics, image processing and environmental data representation — Extensible 3D (X3D) — Part 2: Scene access interface (SAI)*

[11] Broring Arne, Echterhoff Johannes, Jirka Simon, Simonis Ingo, Everding Thomas, Stasch Christoph, Liang Steve, Lemmens Rob", New Generation Sensor Web Enablement", *Sensors* 2011, pp.2652-2699

[12] Bănică Florinel-Gabriel, Chemical Sensors and Biosensors: Fundamentals and Applications. Chichester, UK: John Wiley & Sons. 2012

[13] Reed Carl, Botts Mike, Percivall George, Davidson John, OGC® Sensor Web Enablement: Overview And High Level Architecture, Open Geospatial Consortium, 2013-06-17

[14] Platt Charles, Encyclopedia of Electronic Components **Volume 3**: Sensors, Maker Media, Inc., 2016

[15] Simonis Ingo, OGC® Sensor Web Enablement Architecture, Open Geospatial Consortium Inc., 2008-08-20

[16] Lein James K., Environmental Sensing Analytical Techniques for Earth Observation, Springer, 2012

[17] Botts Mike, Robin Alexandre, OGC Sensor ML: Model and XML Encoding Standard, Open Geospatial Consortium Inc., 2014-02-04

[18] Botts Mike, Robin Alexandre, OpenGIS Sensor Model Language (SensorML) Implementation Specification, Open Geospatial Consortium Inc., 2007-07-17

[19] Groves Paul D., Principles of GNSS, Inertial, and Multisensor Integrated Navigations Systems, Second Edition (GNSS Technology and Applications), Artech House, 2013

[20] IETF RFC 1630, *Universal Resource Identifier in WWW*

[21] IETF RFC 2141, *Uniform Resource Name in WWW*

ICS 35.140

Price based on 61 pages