BS ISO 22166-201:2024

BSI Standards Publication

Robotics — Modularity for service robots

Part 201: Common information model for modules

Documents
Document Sharing Hub

Full document details: BS ISO 22166-201:2024

National foreword

This British Standard is the UK implementation of ISO 22166-201:2024.

The UK participation in its preparation was entrusted to Technical Committee AMT/10, Robotics.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2024
Published by BSI Standards Limited 2024

ISBN 978 0 539 17452 6

ICS 25.040.30

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2024.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO 22166-201:2024

International Standard

ISO 22166-201

First edition
2024-02

Robotics — Modularity for service robots —

Part 201: Common information model for modules

Robotique — Modularité des robots de service —

Partie 201: Modèle d'information commun pour les modules

Reference number
ISO 22166-201:2024(en)

© ISO 2024

Documents
Document Sharing Hub

Full document details: BS ISO 22166-201:2024

COPYRIGHT PROTECTED DOCUMENT

© ISO 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Documents
Document Sharing Hub

Full document details: BS ISO 22166-201:2024

Contents

Page

Foreword ... **iv**
Introduction ... **v**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **1**
4 Common information model for modules ... **3**
4.1 General ... 3
4.2 Relationship between CIM and specific IMs ... 6
4.3 Class for common information model ... 7
4.3.1 General ... 7
4.3.2 Class for module ID ... 9
4.3.3 Class for properties ... 10
4.3.4 Class for input and output variables ... 11
4.3.5 Class for status ... 13
4.3.6 Class for services ... 14
4.3.7 Class for infrastructure ... 17
4.3.8 Class for safety and security ... 19
4.3.9 Class for modelling ... 24
4.3.10 Class for executable form ... 25
Annex A (normative) **Naming rules** ... **27**
Annex B (normative) **Assignment rule of a module ID** ... **28**
Annex C (normative) **Representation of common information** ... **30**
Annex D (informative) **How to use information models** ... **54**
Annex E (informative) **Format for representation of a class** ... **57**
Bibliography ... **58**

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 ISO 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).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

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), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 299, *Robotics*.

A list of all parts in the ISO 22166 series can be found on the ISO website.

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.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Introduction

This document provides a common information model (CIM) for modules which compose service robots based on ISO 22166-1. Based on the CIM, modules can easily be connected and data exchanged between them. It has been designed to enhance the interoperability, reusability, and composability of modules. CIM is a meta-model and is a base model for all kinds of modules such as hardware modules, modules with hardware and software aspects, software modules, and their composite modules. Hence CIM provides meta characteristics that modules should possess.

Module makers, module integrators, robot makers, robot developers, and robot system integrators are able to use CIM in order to obtain the necessary modules more easily, utilize various modules according to function and budget, and develop a new (composite) module based on CIM. The CIM is able to make design of modules, operation and maintenance of service robots help.

The CIM provides four information types:

— information for module identification;

— information for module selection;

— information for module integration;

— information for module operation and maintenance.

The CIM, as a meta model, consists of number of submodels, which are Module ID, Properties, Input and Output Variables, Status, Services, Infrastructure, Safety and Security Level, Modelling, Executable Form.

The implementation model of CIM will be provided in upcoming standards, with an example being ISO 22166-202 for software modules.

The ISO 22166 series applies to composite modules and various types of service robots.

This document presents requirements and guidelines on the information model of a service robot's modules for ensuring nine principles of modularity presented in ISO 22166-1.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 22166-201:2024

Documents
Document Sharing Hub

Robotics — Modularity for service robots —

Part 201: Common information model for modules

1 Scope

This document specifies requirements and guidelines for the common information model (CIM) for modules of service robots to achieve interoperability, reusability, and composability.

This document specifies the structure of the CIM and details the intended use and meaning of its attributes and subclasses.

This document applies to service robots.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 22166-1:2021, *Robotics — Modularity for service robots — Part 1: General requirements*

IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*

IEEE/Open Group 1003.1-2017, *IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 7*

3 Terms and definitions

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

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

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

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

3.1
information model
IM
abstraction and representation of the entities in a managed environment, their properties, operations, and the way that they relate to each other

[SOURCE: ISO 22166-1:2021, 3.1.11 — modified]

3.2
common information model
CIM
information model that modules most frequently use in service robots

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

3.3
module
component or assembly of components with defined interfaces accompanied with property profiles to facilitate system design, integration, interoperability, and reuse

[SOURCE: ISO 22166-1:2021, 3.3.12]

3.4
module property
attribute or characteristic of a module

[SOURCE: ISO 22166-1:2021, 3.3.14]

3.5
software module
module whose implementation consists purely of programmed algorithms

[SOURCE: ISO 22166-1:2021, 3.4.4]

3.6
hardware module
module whose implementation consists purely of physical parts, including mechanical parts, electronic circuits, and any software, such as firmware, not externally accessible through the communication interface

[SOURCE: ISO 22166-1:2021, 3.4.3]

3.7
module with hardware aspects and software aspects
hardware-software module
module whose implementation consists of physical parts, software, and a communication interface that allows data exchange with other modules

3.8
module manager
functions for lifecycle management of module instances including instantiation of modules

3.9
instance
particular entity instantiated from a specific software module or particular entity of a specific module in hardware aspects

Note 1 to entry: In object-oriented programming, instance means a specific realization of an object.

Note 2 to entry: In hardware aspects, instance means one of the same modules used in a composite module.

3.10
performance level
PL
discrete level used to specify the ability of safety-related parts of control systems to perform a safety function under foreseeable conditions

[SOURCE: ISO 13849-1:2023]

3.11
safety integrity level
SIL
discrete level (one out of a possible three) for describing the capability to perform a safety function where safety integrity level three has the highest level, and safety integrity level one has the lowest

[SOURCE: IEC 62061:2021, 3.11]

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 22166-201:2024

3.12
executable form
form of the code in which the software or firmware is managed and controlled completely by the operational environment of the module and does not require compilation (e.g. no source code, object code, or just-in-time compiled code)

Note 1 to entry: Its two primary types are compiled programs and scripts.

Note 2 to entry: It can include the set of files and/or directories that make up a complete module or be a single file.

[SOURCE: ISO/IEC 19790:2012, 3.42]

3.13
hardware aspects
information regarding properties and functions necessary for a module and its physical interconnection and regarding the allowed range of physical properties of the operational environment

[SOURCE: ISO 22166-1:2021, 3.3.5]

3.14
software aspects
information regarding the external software properties necessary for a module and its interface and the execution life cycle of that module's function

[SOURCE: ISO 22166-1:2021, 3.3.21]

4 Common information model for modules

4.1 General

Common information for modules shall consist of the items in Table 4.1. The naming of properties and classes used in the information model shall follow the naming rules in Annex A. In Table 4.1, the symbols "M" and "O" denote mandatory and optional, respectively. CIM in Figure 4.1 is described in detail in 4.3. The class models for CIM are drawn based on ISO/IEC 19505-1:2012. The information provided in Table 4.1 and Figure 4.1 can be used in the design, development, operation, and maintenance stages and Annex D shows at which stage the information provided by the information model is used.

NOTE 1 A CIM for managing the common information in Table 4.1 is illustrated in Figure 4.4.

NOTE 2 In this document, CIM representation in XML is used just as examples to help readers understand the meaning of attributes in the class model.

"Module Name" of Item No. 1 in Table 4.1 is the name representing the module. Hereafter, Item No. "N" in Table 4.1 is abbreviated as Item "N".

"Description" of Item 2 provides the overview of the module, what the module is, what it does, and how it is used.

"Manufacturer" of Item 3 provides contact information for the designer, developer, or manufacturer of the module.

"Examples" of Item 4 provides typical use cases of the module.

"Information model version" of Item 5 is the version number of the information model that this document specifies.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Table 4.1 — Common information for modules and the corresponding group tag (used in Figure 4.4)

No.	Item	Common information model [b]	Information models for modules [c]			Related group/tag name (Abbreviation of each group)
			hard-ware-soft-ware module	hardware module	software module	
1	Module Name	M	M	M	M	GenInfo
2	Description	O	O	O	O	
3	Manufacturer	M	M	M	M	
4	Examples	O	O	O	O	
5	Information model version	M	M	M	M	IDnType
6	Module ID	M	M	M	M	
7	Hardware Aspects	O	M	M	-	
8	Software Aspects	O	M	-	M	
9	Module properties [a]	M	M	M	M	Properties
10	Inputs	O	O	O	O	IOVariables
11	Outputs	O	O	O	O	
12	Status	O	O	O	M	—
13	Services (capabilities)	O	M	O	M	Services
14	Infrastructure	O	O	M	M	Infra
15	Safety/security	O	M	O	M	SafeSecure
16	Modelling	O	O	O	O	Modelling
17	ExecutableForm	O	O	O	M	ExecutableForm

[a] It is mandatory only to those which can be influenced (set) from the outside or at least to those which have an expected effect on other modules.

[b] All items are mandatory for CIM.

[c] For information models such as hardware-software modules, hardware modules, or software modules, some types of items can be omitted depending on their functionalities. In particular, for information models of hardware modules and software modules, the items “software aspects” and “hardware aspects” are not included, respectively.

“Module ID” of Item 6 shall be the unique identifier of the module within a system as described in Annex B. Module ID includes information such as which type a module is, hardware-software, hardware, or software, and whether a module is a basic or a composite module.

“Hardware Aspects” of Item 7 and “Software Aspects” of Item 8 relate only to composite modules. If a module is composed of two or more hardware-software modules, software modules, and/or hardware modules, the module IDs are listed in “Hardware Aspects” if they are hardware or hardware-software modules, otherwise (if they are software modules) the module IDs are listed in “Software Aspects”.

“Module properties” of Item 9 are values that are generally used in the initialization of modules. Module properties are classified into mandatory and optional, and shall be listed. The relationship between modules is represented in this item, whose detailed contents will be provided in upcoming standards such as ISO 22166-202.

NOTE 3 Information about the relationships between modules can be provided differently depending on the module type.

NOTE 4 Environment constraints are also considered as properties, examples of which are operating temperature, operating humidity, and maximum allowed mechanical shock. These parameters related to the behaviours of the module can be set in the properties. For example, each coefficient used in the PID (Proportional, Integral and Differential) control algorithm is used once in initialization and is changed and used several times during the execution of related software modules.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

"Inputs/Outputs" of Items 10 and 11 describe the names of variables for data transfer into and/or out of a module.

EXAMPLE 1 Inputs and outputs of a servo control software module are the encoder value and motor control values, respectively.

NOTE 5 Properties are kinds of input values from the viewpoint of modules, but have to be distinguished in that Inputs are related to the environment of the module, but Properties are related to parameters contained in the module. For example, Inputs of the servo control software module are encoder values, but its Properties are P, I, and D coefficients, where P, I, and D denote Proportional, Integral and Differential.

"Status" of Item 12 describes the status of a module that is operating.

NOTE 6 Status is not utilized at design and development stages.

"Services (capabilities)" of Item 13 describes the interfaces that a module provides and utilizes for robot services.

NOTE 7 Service means performing one or more functions of a module for other modules via a pre-described interface.

EXAMPLE 2 Examples of function format for software aspects are shown in Table 4.2. In this example, data types such as int16 and unit 8 are defined in Table C.5.

Table 4.2 — Example of function format for software modules

Name	Arguments	Return value	Description
initialize	Integer init_ival1, Real init_rval2	Integer	Initialization using 2 arguments Return value: (0: success, negative value: error type)
	Integer init_ival1, Real init_rval2, Integer init_ival3	Integer	Initialization using 3 arguments Return value: (0: success, negative value: error type)

EXAMPLE 3 An example of a function format for an electrical/electronic aspect is shown in Table 4.3, where the arguments "connType", "keying", and "busProtocol" mean the type of connector, the gender of the connector, and the protocol type that the module uses. The function is used to check that the peer module utilizes the proper electrical aspects of the module. Values for connector type can be USB-A, RJ45, DB-9, etc. The value for keying is male or female. Values for busProtocol can be USB, Ethernet, EtherCAT, RS232, etc.

Table 4.3 — Example of function format for electrical/electronic adaptability

Name	Arguments	Return value	Description
checkElecConnectivity	String connType, String keying, String busProtocol	Boolean	Check electrical/electronic adaptability using 3 arguments Return value: (True: success, False: error)

EXAMPLE 4 Examples of a function format for mechanical aspects are provided in Table 4.4. As in the electrical/electronic aspect, the function is used to check that the peer module is utilizing the proper mechanical aspects of the module. However, unlike the electrical/electronic aspect, the function format for the mechanical aspect can be more complicated due to a huge variety being used in practice. Consequently, only two simplified categories are listed here, joint and link.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Table 4.4 — Example of function format for mechanical adaptability

Name	Arguments	Return value	Description
linkConnectivity	Real origin, Real mass, Real Inertia[], String shape, Real size, Real axis[], String connection, Real collision[]	Boolean	Check mechanical link adaptability using 8 arguments Return value: (True: success, False: error)
jointConnectivity	Real origin[], Real axis[], Real limits, Real damping, Real friction	Boolean	Check mechanical joint adaptability using 5 arguments Return value: (True: success, False: error)

"Infrastructure" of Item 14 lists hardware and/or software that the modules commonly use or connect to.

EXAMPLE 5 Examples of Infrastructure are power type, middleware type, databus type, and database type.

"Safety/Security" of Item 15 describes the safety-related performance level and the security information provided by the module.

For the Safety/Security of a general service robot, a safety-related performance level is used for the module, which is defined in ISO 13849-1, and the security-related level is listed for the module, where the security level is "0 – 4". Security level "0" means that there are no security measures. The security levels "1-4" are defined in IEC 62443-4-2:2019. However, for specific robot types such as medical robots and physical assistant robots, other safety-related standards and security standards should be utilized. A performance level is generally provided for each single safety function. If a module has several safety functions, the module shall provide a performance level of the module using the combination of the performance levels of all the safety functions of the module or via verification and validation of the module's overall safety-related function. The details are provided in 4.3.8.

"Modelling" of Item 16 provides different kinds of models for simulation or design purposes.

"ExecutableForm" of Item 17 provides program codes executed to achieve or support the module's purpose.

Classes provided in this document are able to be described using the table format in Annex E.

4.2 Relationship between CIM and specific IMs

The common information model for modules shall be used in information models for all types of modules, which are hardware-software modules, software modules, and hardware modules. Their relationships are shown in Figure 4.1. Hardware-software modules are composed of hardware aspects and software aspects. The relationship between information models of hardware modules, software modules, and hardware-software modules is illustrated in Figure 4.2.

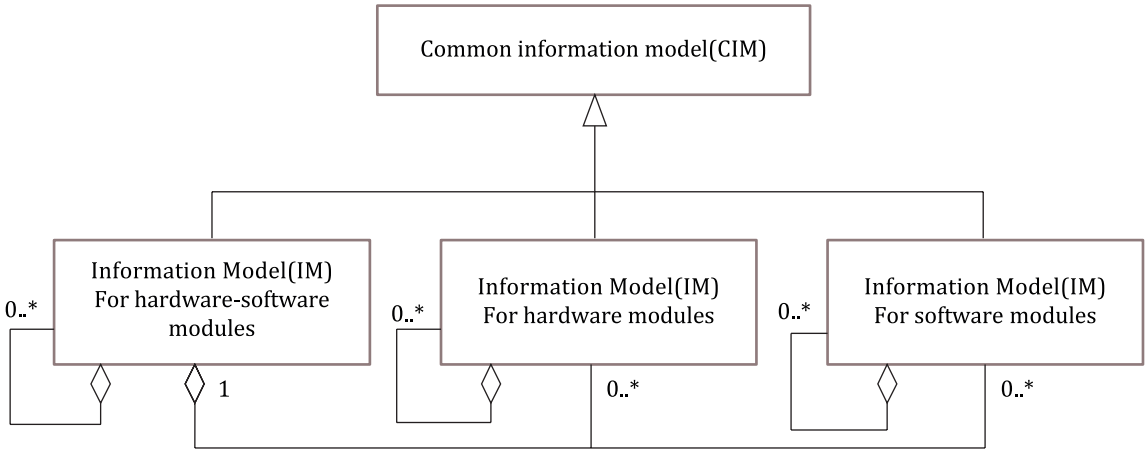

Figure 4.1 — Relationships between information models for modules

© ISO 2024 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO 22166-201:2024

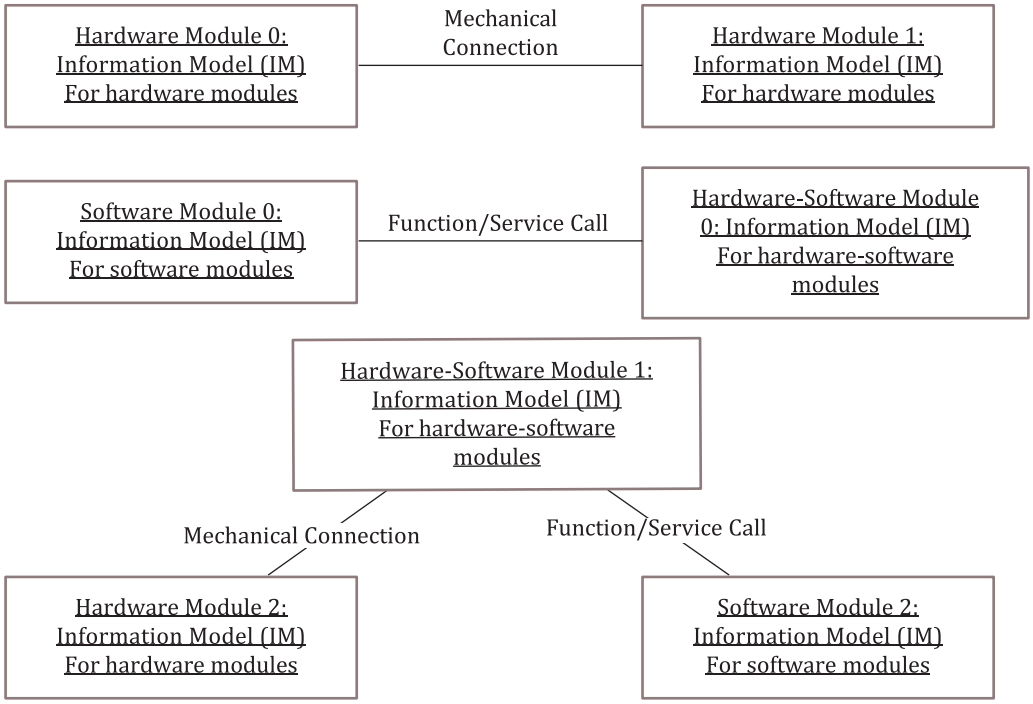

Figure 4.2 — Example of relationships between information models of modules

4.3 Class for common information model

4.3.1 General

The common information model shall be grouped into 9 inner classes as illustrated in Figure 4.3, where eight classes are suggested from Table 4.1 and an additional class, Status, is related to the status-related information of a module. This class is mainly used during operation of the module. The four items of the GenInfo group in Table 4.1, which are Module name, Description, Manufacturer, and Examples, become the member attributes of the class Common Information Model (CIM).

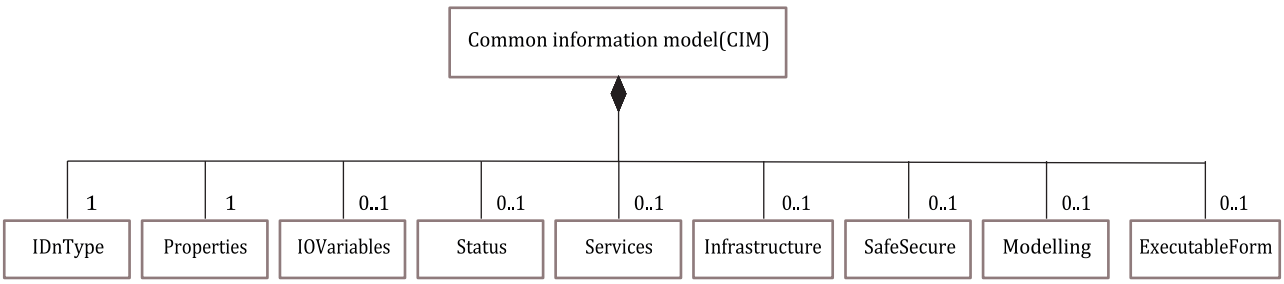

Figure 4.3 — Classes of the common information model

The class CIM (Common Information Model) shall have attributes given in Figure 4.4 and Table 4.5 and its attributes are based on ISO 22166-1:2021, Clauses 4 to 7. Values of attributes such as ModuleName, Description, Manufacturer, and Examples are provided in Annex C. IDnType, Properties, IOVariables, Status, services, infrastructure, SafeSecure, modelling, and ExecutableForm in Figure 4.4 refer to the class described in 4.3.2 to 4.3.10.

NOTE 1 Attributes can be declared as one of the following: private (-), protected (*), or public (+). The name of an attribute is given first and the data type of the attribute is defined next. The separation symbol between an attribute's name and its data type is a colon (:). If attributes are declared as public, it is not necessary to define the functions to access those attributes.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

NOTE 2 The four items moduleName, description, manufacturer, and examples are attributes of class CIM.

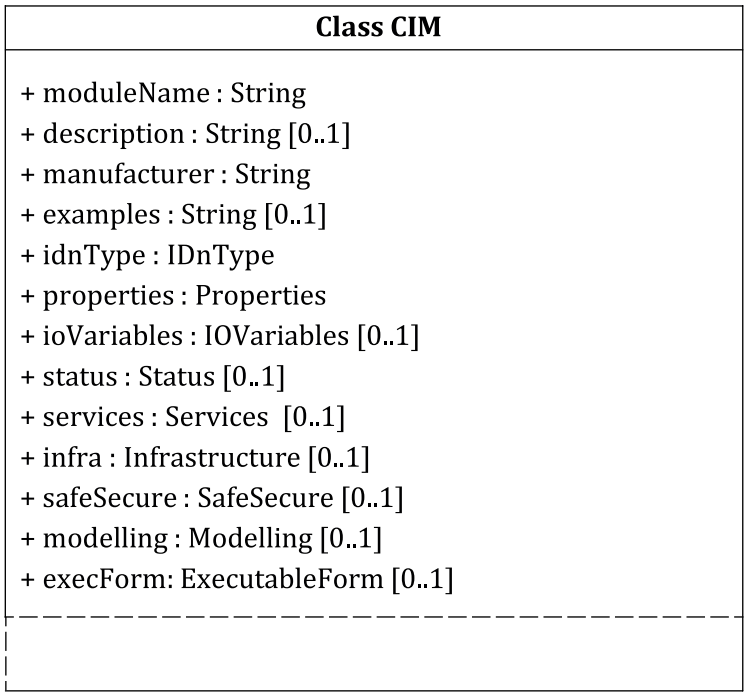

Figure 4.4 — Class diagram of class CIM

Table 4.5 — Class CIM

Description: The root class for the CIM (Common Information Model).				
Derived From: N/A				
Attributes:				
moduleName	String	M	1	The name representing the module
description	String	O	1	The overview of the module, what the module is, what it does, and how it is used.
manufacturer	String	M	1	The contact information for the designer, developer, or manufacturer of the module
examples	String	O	1	Typical use cases of the module
idnType	IDnType	M	1	The module ID includes a unique identifier, instance ID, and owned software/hardware aspects as their IDs. See the details in 4.3.2.
properties	Properties	M	1	A list of configurable parameters of the module. See the details in 4.3.3.
ioVariables	IOVariables	O	1	Information for Input and Output Variables of a module. See the details in 4.3.4
status	Status	O	1	Information for the status and health condition of a module. See the details in 4.3.5
services	Services	O	1	The interfaces that a module provides and utilizes for robot services. See the details in 4.3.6.
infra	Infrastructure	O	1	Information for the infrastructure of a module includes Electric Power, Data Bus, Database, and Ingress Protection. See the details in 4.3.7

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Table 4.5 *(continued)*

safeSecure	SafeSecure	0	1	The safety level and the security level that the module provides. See the details in 4.3.8.
modelling	Modelling	0	1	The modelling-related information that the module supports for simulation. See the details in 4.3.9.
execForm	ExecutableForm	0	1	The program-related information executed to achieve or support the purpose of the module. See the details in 4.3.10.

4.3.2 Class for module ID

Information for a Module ID shall be defined in class IDnType which shall have attributes given in Figure 4.5 and Table 4.6. Values of the attribute "moduleID" and other attributes in Table 4.6 are provided via Annex B and Annex C. When the same type of modules are used in a composite module, the modules shall be identified individually. For this purpose, an Instance ID (or IID) is needed. An IID for a software module is dynamically assigned by the module manager. An IID for a module with hardware aspects is statically assigned by the module manufacturer. An IID shall be unique if it is used. The information model version is the version number of the information model used when the module is specified. The information model version is upgraded whenever attributes (including methods) of the classes presented in this document are updated.

NOTE 1 A module with hardware aspects is a hardware module or a module with hardware and software aspects.

The attributes "hwAspects" and "swAspects" are lists of module IDs for hardware aspects and software aspects, respectively.

On the instantiation of this class, the instanceID shall be set using its moduleID. If a module is added to a composite module, the composite module shall check whether or not its moduleID is duplicated in the composite module. If duplicated (or the same module exists), the module manager/composite module manufacturer shall assign a new instance ID to the module.

NOTE 2 The Module ID is provided by the manufacturer of a module.

NOTE 3 The moduleID including IID is used in order to access the module. The default IID is "0".

NOTE 4 The informationModelVersion is "1.0" when the information model defined in this document is used.

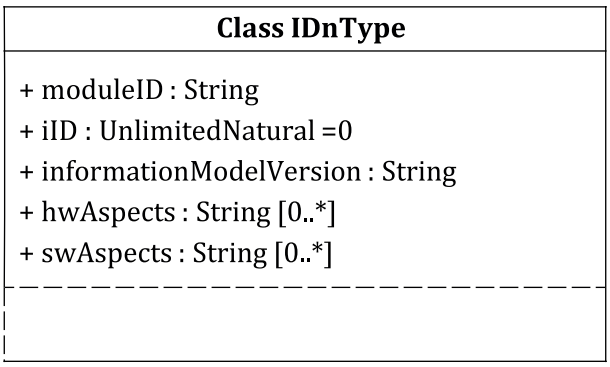

Figure 4.5 — Class diagram of class IdnType

Table 4.6 — Class IDnType

Description: Information for Module ID shall be defined in class IDnType.				
Derived From: N/A				
Attributes:				
moduleID	String	M	1	See Annex B for the moduleID format
iID	UnlimitedNatural	M	1	Instance ID, default value = 0.

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Table 4.6 *(continued)*

informationModelVersion	String	M	1	Version number of the information model
hwAspects	String	O	N	List of module IDs for hardware modules and modules with hardware aspects and software aspects
swAspects	String	O	N	List of module IDs for software modules

4.3.3 Class for properties

Information for properties of a module shall be defined in class Properties which shall provide the properties of a module as in Figure 4.6 and Tables 4.7 to 4.9, which include information necessary for the execution of the module and information about the operating environment or condition of the module. Figure 4.6 is the class diagram for class Properties and shows the relationship between classes given in Table 4.7 to Table 4.9. These detailed attributes such as the member name and the related data type shall be defined using values of the tag "Property" in Annex C.

The following examples are given in XML and have to be transformed as in Figure 4.6:

```
<!-- example for hardware-software module -->
<Property name="maxRatedCurrent" value="15" type="float32" unit="ampere" description =
"maximum of rated current for motor" />
<Property name="angleResolution" value="1" type="float32" unit="degree" description =
"Angle Resolution of Lidar Sensor" />
<Property name="minOpRange" value="0.05" type="float32" unit="meter" description =
"minimum operation range of Lidar Sensor" />
<Property name="maxOpRange" value="10" type="float32" unit="meter" description = "maximum
operation range of Lidar Sensor" />
<Property name="commModule" value="Ethernet microUSB" type="array of string" unit="null"
description = "communication protocol type of Lidar Sensor" />
<!-- example for software module -->
<Property name="Weight" value="150" type="float32" unit="gram" description = "weight of
Lidar Sensor" />  <-- example for hardware-software module -->
<Property name="maxRadius" value="2" type="float32" unit="meter" description = "maximum
radius for Obstacle Avoidance" />
<Property name="minRadius" value="0.5" type="float32" unit="meter" description = "minimum
radius for Obstacle Avoidance" />
<!-- example for hardware module -->
<Property name="origin" type="array of float32" unit="cm", "radian" description = "pose
of the inertial reference frame, relative to the link reference frame including xyz and
rpy" value="[0 0 0 0 0 0]" />
<Property name="mass" type="float32" unit="gram" description = "mass of a link" value =15
/>
<Property name="inertia" type="array of float32" unit="g-cm2" description = "inertial
properties of the link, 3x3 matrix" value ="[1 0 0; 0 1 0; 0 0 1]" />
<Property name="shape" type="string" unit="" description = "shape of the visual object"
value =cylinder />
<Property name="size" type="3x1 array of float32" unit="cm" description = "three side
lengths of the box" value ="[100; 200; 100]" />
```

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

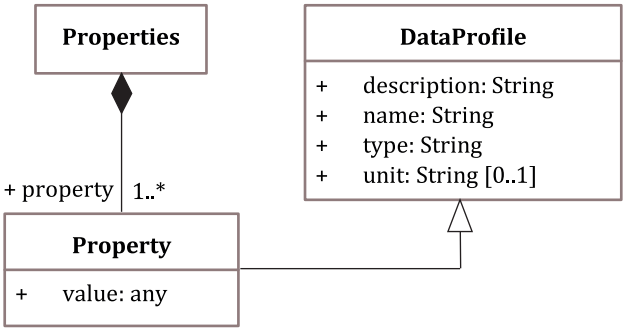

Figure 4.6 — Class diagram for class Properties

Table 4.7 — Class DataProfile

Description: Class for data profiles				
Derived From: N/A				
Attributes:				
description	String	M	1	Explanation of the property
name	String	M	1	Name of the property
type	String	M	1	Data type of the property
unit	String	O	1	Unit of the property. If no unit, unit is null.

Table 4.8 — Class Property

Description: Class for a property				
Derived From: DataProfile				
Attributes:				
value	any	M	1	Value of the property

Table 4.9 — Class Properties

Description: Defines information for properties of a module				
Derived From: N/A				
Attributes:				
property	Property	M	N	List of attributes generated according to the class Property

4.3.4 Class for input and output variables

Information for Input and Output Variables of a module shall be defined in the class IOVariables as in Figure 4.7 and Tables 4.10 to 4.15. The class IOVariables shall provide variables used within the module, which include information necessary for the execution of the module. These variables shall be defined using the class Variable specified in Table 4.12. The class Variable inherits from the class VariableProfile in Table 4.11, which, in turn, inherits from the class DataProfile in Table 4.7. For storing additional information, the class NVList in Table 4.14 can be utilized as a container. The NVList is a list of NameValue objects, where the class NameValue is defined in Table 4.15. The values IN, OUT, and INOUT are defined as enumeration data types in Table 4.10. The detailed attributes such as the member name and the related data type shall be defined using values of the tag IOVariables as in Annex C. Figure 4.7 shows the relationships between classes for the class IOVariables.

The following examples are given in XML and have to be transformed to the form of Figure 4.12:

```
 <IOVariables>
     <Input name="controlValue" type="float32" unit="ampere" description="current control to
motor" /> <!-- IOType=IN -->
```

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

```
 <Output name="encoder" value= 0 type ="uint32" unit = "none" description = "absolute
encoder's value" /> <!-- IOType=OUT -->
     <Inout name="state" type="uint8" value= 0 unit = "none" description = "status of motor"/>
<!-- IOType=INOUT -->
 </IOVariables>
```

From these tags, the following member variables and their data types are generated.

+ controlValue : float32 // or public float32 controlValue; current control command to motor, input

+ encoder=0 : uint32 // or public uint32 encoder=0; value of absolute encoder, output

+ state=0 : uint8 // or public uint8 state=0; reset or read status of motor, input and output

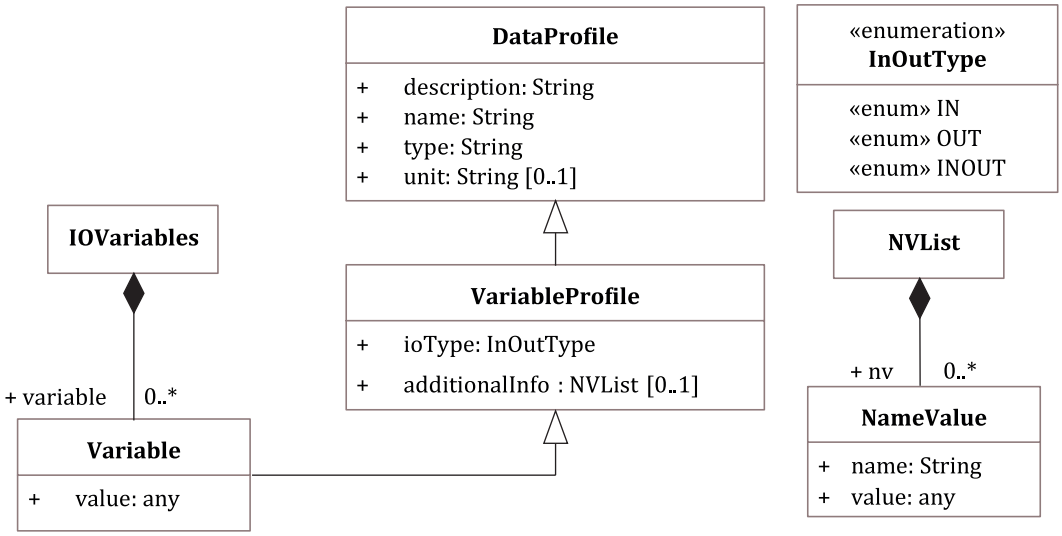

Figure 4.7 — Relationships between classes for class IOVariables

Table 4.10 — Enumeration InOutType

Description: Types of input and output variables	
Attributes:	
IN	Used for variables to which external modules write data
OUT	Used for variables from which external modules reads data
INOUT	Used for variables which external modules read from or write data to.

Table 4.11 — Class VariableProfile

Description: Class for a profile of variables				
Derived From: DataProfile (see Table 4.7)				
Attributes:				
ioType	InOutType	M	1	One of {IN, OUT, INOUT}. See Table 4.10
additionalInfo	NVList	O	1	The class NVList serves as a container for additional information across various classes. See Table 4.14

Table 4.12 — Class Variable

Description: Class for a variable				
Derived From: VariableProfile (see Table 4.11)				
Attributes:				
value	any	M	1	Value of the variable

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024

Table 4.13 — Class IOVariables

Description: Class for input and output variables of a module				
Derived From: N/A				
Attributes:				
variable	Variable	O	N	List of attributes generated according to the tag Variable

Table 4.14 — Class NVList

Description: A list for NameValue for additional information across various classes.				
Derived From: N/A				
Attributes:				
nv	NameValue	O	N	A list of NameValue. See Table 4.15

Table 4.15 — Class NameValue

Description: A list for NameValue for additional information across various classes.				
Derived From: N/A				
Attributes:				
name	String	M	1	Name of additional information
value	any	M	1	Value of additional information

4.3.5 Class for status

Information for the status and health condition of a module shall be defined in class Status as in Figure 4.8 and Table 4.17. Class Status provides the status of a module which represents the health condition of the module if the hardware information model exists, the status of the execution life cycle if the software information model exists, and the error type that may have occurred in operation.

Error numbers shall use those defined in IEEE/Open Group 1003.1-2017. Additional error numbers may be defined for each software module, which shall not conflict with error numbers in POSIX.1003.1.

NOTE 1 Examples of HealthCond are as follows: GH (good health), H1 (Health1), H2 (Health2), and Fa (Failure).

NOTE 2 Examples of errors used in ErrorType are as follows: Operation not permitted, Authentication Error, Bad Parameter, Unsupported Service, out of Range, and Precondition not met. Values of ErrorType will be defined in detail in ISO 22166-202 and other documents.

NOTE 3 ExeStatus is defined in ISO 22166-1 and shall be one of following: CREATED, IDLE, EXECUTING, DESTRUCTED, and ERROR. These values are defined as enumeration data types in Table 4.16.

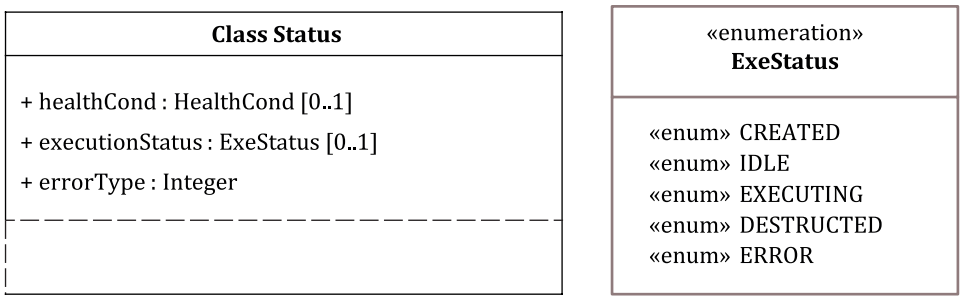

Figure 4.8 — Class diagram for class Status

© ISO 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 22166-201:2024