ISO/~~DIS~~FDIS 22166-201~~:2022(E)~~

~~Date: 2022-11-05~~

ISO/TC 299

Secretariat: SIS

Date: 2023-10-21

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

FDIS stage

Contents ~~Page~~

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 ~~documents~~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).

Field Code Changed

~~Attention is drawn~~ISO draws attention to the possibility that ~~some of~~ the ~~elements~~implementation of this document may ~~be~~involve the ~~subject~~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. ~~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).~~

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.

Field Code Changed

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~~www.iso.org/members.html.

Introduction

This document provides a common information model (CIM) for modules which compose service robots based on ISO 22166-1~~ Modularity for Service Robots Part 1 - General Requirements.~~. 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 ~~following~~ four information types:

— ~~— Information~~information for module identification;

— ~~— Information~~information for module selection;

— ~~— Information~~information for module integration;

— ~~— Information~~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 ~~document~~ for software modules.

~~Future editions and parts of this family of the International Standard may include more specific requirements and guidelines on modules with software aspects and modules with hardware aspects. Thus, the family~~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.

~~The current version of ISO 22166 consists of the following parts, under the general title "~~

Robotics - — Modularity for service robots — " —

— — Part 1: General requirements

— — Part 201: Common information model for modules

— — Part 202: Information model for software modules

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, in whole or in part, are normatively referenced in this document and are indispensable for its application. 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.

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

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:~~2015~~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

[~~Modified~~SOURCE: IEC 62061:2021, 3.11]

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]

~~5~~4 Common information model for modules

~~5.1~~4.1 General

Common information for modules shall consist of the items in ~~Table 4.1.~~Table 4.1. The naming of properties and classes used in the information model shall follow the naming rules in ~~Annex A.~~Annex A. In ~~Table 4.1,~~Table 4.1, the symbols "M" and "O" denote mandatory and optional, respectively. CIM in ~~Figure 4.1~~Figure 4.1 is described in detail in ~~4.3.~~4.3. The class models for CIM are drawn based on ISO/IEC 19505-1:2012. The information provided in ~~Table 4.1~~Table 4.1 and ~~Figure 4.1~~Figure 4.1 can be used in the design, development, operation, and maintenance stages and ~~Annex D~~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~~Table 4.1 is illustrated in ~~Figure 4.4.~~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~~Table 4.1 is the name representing the module. Hereafter, Item No. "N" in ~~Table 4.1~~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.

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

No.	Item	Common information model [b]	Information models for modules [c]			Related group/tag name (Abbreviation of each group)
			hardware-software 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.~~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 document. .

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.

-"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

5

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.

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.

5.24.2 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.

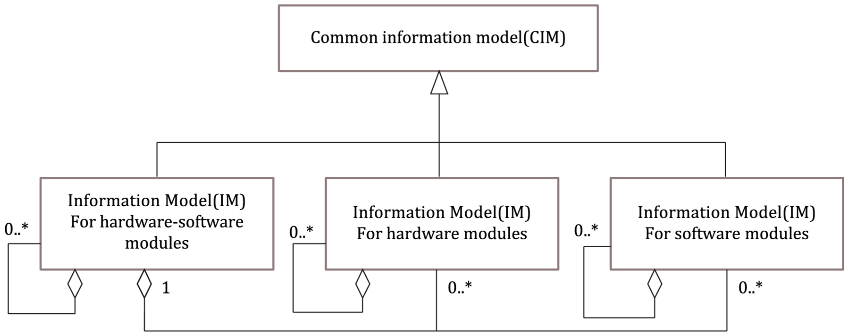

Figure 4.1 — Relationships between information models for modules

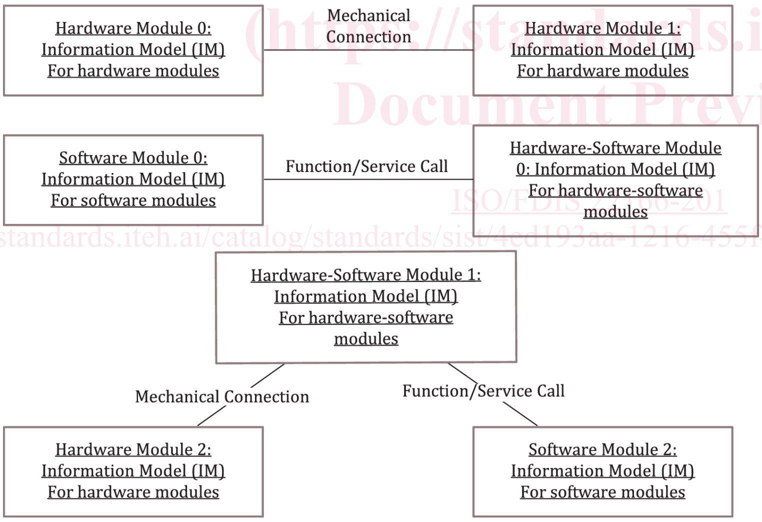

Figure 4.2 — Example of relationships between information models of modules

5.34.34.3 Class for common information model

5.3.14.3.1 General

The common information model shall be grouped into 9 inner classes as illustrated in Figure 4.3, where seven 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,Table 4.1, which are Module name, Description, Manufacturer, and Examples, become the member attributes of the class Common Information Model (CIM).

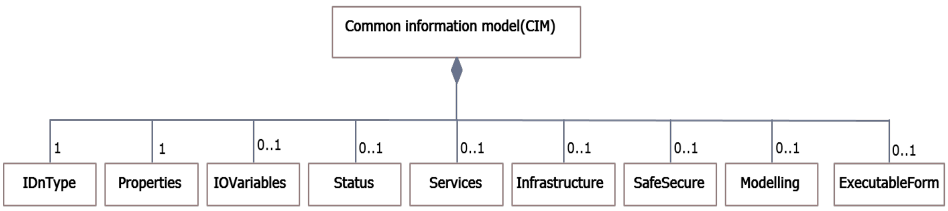

Figure 4.3

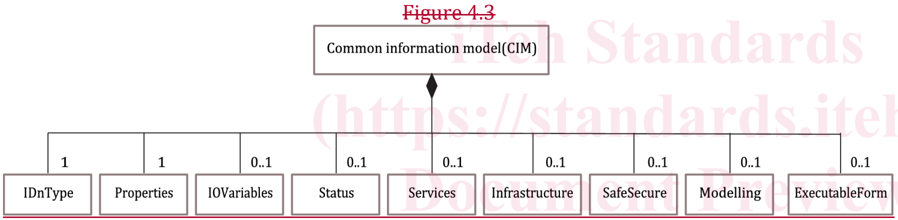

Figure 4.3 — Classes of the common information model

The class CIM (Common Information Model) shall have attributes given in Figure 4.4Figure 4.4 and Table 4.5Table 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.Annex C. IDnType, Properties, IOVariables, Status, Services, Infrastructure, SafeSecure, Modelling, and ExecutableForm in Figure 4.4Figure 4.4 refer to the class described in 4.3.24.3.2 to 4.3.10.4.3.10.

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.

NOTE 2: The four items moduleName, description, manufacturer, and examples are attributes of class CIM (Common Information Model)..

8