INTERNATIONAL STANDARD

ISO 22166-1

First edition
2021-02

Robotics — Modularity for service robots —

Part 1: General requirements

Robotique — Modularité des robots de service —

Partie 1: Prescriptions générales

Reference number
ISO 22166-1:2021(E)

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

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 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 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.

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

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 has been developed for the rapidly evolving service robotics sector. At present this robotics market covers many small and niche sectors for which it is difficult to develop the specific and wide-ranging components needed. The market sizes and applications are expected to grow significantly, and the number and range of their functions are also increasing. To enable wide-spread and interoperable development of service robots, a common approach for building service robots is needed. This document lays out such common requirements.

On one side, the manufacturer-dependent architectural approaches currently adopted for designing service robots makes design and development difficult and substitution and reuse of modules in upgrading robot products is virtually impossible. On the other side, the research community has developed a vast knowledge base in robot modular design and continues to develop new methods for realising modular approaches, but none have the widespread appeal needed to make significant impact. In these conditions, this document can assist the service robotics manufacturers to produce the quality products at affordable cost demanded by the markets and new approaches are urgently needed to help the markets evolve to meet the global challenges.

An International Standard on robot modularity and robot module interoperability focusing on main issues of safety, security, connectivity (from both hardware and software perspectives) and functionality is pivotal to change the service robotics landscape and speed up the development of the new service robot market sectors. The robot modularity issues in this document are classified into basic modules with hardware and/or software aspects and composite modules. Requirements and guidelines are formulated so that module-based design approaches can be realised allowing application specific service robots and service robot systems meeting customer's requirements to be easily configured. The issues are classified into (a) safety and security, and (b) interoperability guidelines. In addition, the open modular approach realised has to enable modules to be easily substituted by other modules having the same interface specifications but perhaps with enhanced functionalities as needed.

Safety requirements specified in existing safety standards (e.g. ISO 13482, ISO 10218-1, ISO 10218-2, ISO/TS 15066) apply on the system level as well as on the level of a single module. The safety guidelines at the module level of this document are formulated to ensure compliance with the C-type standards for robot system safety. Security issues are also important when adopting an open modularity approaches and hence have been included in this document (e.g. to align with emerging IEC/TC 44 and IEC/TC 65 security related work projects).

Future parts of the ISO 22166 series are intended to include more specific requirements on particular types of robot modules, e.g., basic and composite modules with hardware and/or software aspects, and for particular types of service robots, e.g., mobile servant robots, physical assistant robots, person carrier robots, and service robots in professional environments.

Robotics — Modularity for service robots —

Part 1:
General requirements

1 Scope

This document presents requirements and guidelines on the specification of modular frameworks, on open modular design and on the integration of modules for realising service robots in various environments, including personal and professional sectors.

The document is targeted at the following user groups:

— modular service robot framework developers who specify performance frameworks in an unambiguous way;

— module designers and/or manufacturers who supply end users or robot integrators;

— service robot integrators who choose applicable modules for building a modular system.

This document includes guidelines on how to apply existing safety and security standards to service robot modules.

This document is not a safety standard.

This document applies specifically to service robots, although the modularity principles presented in this document can be utilized by framework developers, module manufacturers, and module integrators from other fields not necessarily restricted to robotics.

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 9787, *Robots and robotic devices — Coordinate systems and motion nomenclatures*

ISO 12100:2010, *Safety of machinery — General principles for design — Risk assessment and risk reduction*

ISO/TR 22100-4, *Safety of machinery — Relationship with ISO 12100 — Part 4: Guidance to machinery manufacturers for consideration of related IT-security (cyber security) aspects*

ISO/IEC 27032, *Information technology — Security techniques — Guidelines for cybersecurity*

IEC 61076-1, *Connectors for electronic equipment-Product requirements — Part 1: Generic specification*

IEC 61984, *Connectors — Safety requirements and tests*

IEC/TS 62443-1-1, *Industrial communication networks — Network and system security — Part 1-1: Terminology, concepts and models*

IEC 62443-2-1, *Industrial communication networks — Network and system security — Part 2-1: Establishing an industrial automation and control system security program*

IEC 62443-3-3, *Industrial communication networks — Network and system security — Part 3-3: System security requirements and security levels*

NIST SP 800-154, *Guide to data-centric system threat modelling*

NIST SP 800-160 vols 1 and 2, *Systems security engineering considerations for a multidisciplinary approach in the engineering of trustworthy secure systems*

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 General terms

3.1.1
abstraction layer
interface to the system that allows some or all of the capabilities of the system to be accessed in a different and generally more abstract manner

Note 1 to entry: An abstraction layer for a module is the same in the case where the system is the module.

3.1.2
connector
physical mechanism that enables connection and disconnection between parts of the system

EXAMPLE Communication, powering, mechanical linking.

3.1.3
electrical interface
combination of connectors and the electrical properties for transmitting power, analogue or digital signals

3.1.4
execution life cycle
finite state machine defining all stages of execution of a part's function

3.1.5
error
discrepancy between a computed, observed or measured value or condition, and the true, specified or theoretically correct value or condition

[SOURCE: IEC 60050-192:2015, 192-03-02]

3.1.6
failure
loss of ability to perform as required

[SOURCE: IEC 60050-192:2015, 192-03-01]

3.1.7
fault
inability to perform as required, due to an internal state

[SOURCE: IEC 60050-192:2015, 192-04-01]

3.1.8
function
defined objective or characteristic action of a system or component or module

[SOURCE: ISO/IEC/IEEE 24765, 3.1206-5 — modified.]

3.1.9
functional safety
part of the overall safety relating to the equipment under control (EUC) and the EUC control system that depends on the correct functioning of the electrical, electronic and programmable electronic (E/E/PE) safety-related systems and other risk reduction measures

[SOURCE: IEC 61508-4:2010, 3.1.12]

3.1.10
hardware abstraction layer
HAL
abstraction layer for a component/module that contains hardware aspects, with the abstraction layer providing control of the component/module via a software interface

Note 1 to entry: The purpose of a HAL is usually so that different module implementations can be accessed through the same software interface.

3.1.11
information model
abstraction and representation of the entities in a managed environment, their properties, attributes and operations, and the way that they relate to each other

Note 1 to entry: The information model is independent of any specific repository, usage of software aspects, protocol, or platform.

3.1.12
security
combination of confidentiality, integrity, and availability

[SOURCE: ISO/TR 17522:2015, 3.19]

3.2 Terms related to component

3.2.1
component
part of something that is discrete and identifiable with respect to combining with other parts to produce something larger

Note 1 to entry: Component can be either software or hardware. A component that is mainly software or hardware can be referred to as a software or a hardware component respectively.

Note 2 to entry: Component does not need to have any special properties regarding modularity.

Note 3 to entry: Component and module have been used interchangeably in general terms, but to avoid confusion the term module is used to refer to a component that meets the guidelines presented in this document.

Note 4 to entry: A module is a component, whereas a component does not need to be a module.

3.2.2
software component
component whose implementation consists of a computer programmed algorithm

3.2.3
hardware component
component whose implementation consists of physical elements and possibly any embedded software necessary for its operation

3.3 Terms related to module

3.3.1
composability
ability to assemble modules logically and physically (without need for adaptation of the modules or additional interfacing work) using various combinations into new modules

Note 1 to entry: While 'integration' generally implies significant effort, 'composition' generally implies limited to no effort.

3.3.2
configuration
arrangement of a composite module in terms of the number and type of modules used, the connections between those modules, and the settings for those modules, in order to achieve the desired functionality of the modular robot as a whole

Note 1 to entry: ISO 8373 also defines (joint) configuration but this is a different concept.

Note 2 to entry: This term describes to result of some process, i.e. the state something is in. The process of creating such a state is covered by the term *configuring* (3.3.3).

3.3.3
configuring
setting the number of modules, type of modules, the connections between the modules, and the settings for the modules in order to achieve the desired functionality of a modular service robot as a whole

3.3.4
granularity
degree to which a robot module can be broken down into separate modules

3.3.5
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

Note 1 to entry: Physical interconnection information includes mechanical properties (material, shape, pose, size, forces/torques), electrical and electromagnetic properties, pneumatic and hydraulic properties.

Note 2 to entry: Operational environmental properties include forces, temperature, humidity, vibration and mechanical shock, illumination and noise (sound and electro-magnetic).

3.3.6
infrastructure
structured facilities and resources to support the operation of modules and systems

3.3.7
interface
shared boundary between two or more functional modules, defined by various characteristics pertaining to the functions, signal exchanges, and other characteristics

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.2058, definition 1]

3.3.8
interoperability
capability to communicate, execute programs or transfer data or power among modules or combine modules physically and/or logically in a manner that requires the user to have little or no knowledge of the unique characteristics of the individual modules

3.3.9
interchangeability
module property allowing it to be capable of being used to replace another module

Note 1 to entry: Such interchangeability can relate to modules produced by one manufacturer or from different manufacturers.

3.3.10
mechanical interface
physical means of connection with other modules used for the transmission of physical forces and facilitating module function and/or configuring structure

Note 1 to entry: Transmitted physical forces include forces controlled for an intended purpose as part of planned function, and uncontrolled forces both intentional (e.g. structural support) and unintentional (e.g. cushioning).

Note 2 to entry: ISO 8373 uses the term for the mechanical interface between a manipulator and the end-effector. In this document, the term is used in a broader sense, including any mechanical interface between robot modules.

3.3.11
modularity
set of characteristics which allow systems to be separated into discrete modules and recombined

3.3.12
module
component or assembly of components with defined interfaces accompanied with property profiles to facilitate system design, integration, interoperability, and re-use

Note 1 to entry: A module may have both hardware and software aspects. It may consist of other components (hardware and software) or other modules (hardware and software).

Note 2 to entry: This neither requires nor prevents the use of Open Source Software to implement parts or all of the open module's functionalities.

Note 3 to entry: Although an open module is conceptually the opposite of a black module, it is still treated as such a black box module for the purpose of this document, i.e. in a robot system conforming to this document, other modules should only communicate with the open module through its official, manufacturer-specified module interface.

Note 4 to entry: An open module is not necessarily a composite module, nor is a composite module necessarily an open module.

3.3.13
package
set of all software binaries, configuration information and support files necessary for a module with software aspects to function as designed

Note 1 to entry: Packages can depend on other packages.

3.3.14
module property
attribute or characteristic of a module

EXAMPLE A module property for hardware can be the torque of an actuator. A module property for software can be the response time to a new command.

3.3.15
module property profile
catalogue of the values of a sub-set of module properties

3.3.16
quality of service
minimum level of performance of a module's service to other modules connected to it for overall operation as intended

3.3.17
reconfiguration
altering the configuration of a modular robot in order to achieve an intended change in the function of the modular robot

3.3.18
reusability
ability to adopt modules, previously designed and produced, to facilitate the development of new modules and robot systems to realise different required functionalities

3.3.19
robot module
module intended to be used as part of a modular robot system

Note 1 to entry: Not all modules used in a modular robot system need to be robot modules, but if the primary intention of a module is use in a modular robot system, then it is a robot module.

Note 2 to entry: Example robot modules are presented in Annex B as being important for service robot modularity.

3.3.20
self-configuration
self-reconfiguration
changing the configuration of a modular robot through an automated process without interaction from outside of the system/subsystem except to initiate the process, if necessary

Note 1 to entry: Typically, mechanical and electrical connections need to be manually (re-)configured, with the automation applying to (re-)configuration of the software aspects.

3.3.21
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

3.4 Terms for classification of modules

3.4.1
basic module
module that is not decomposable into smaller modules

EXAMPLE Basic modules for service robot can be defined as input, processing, output or infrastructure support modules.

3.4.2
composite module
module constructed by two or more modules

Note 1 to entry: A module manufacturer can choose to document the internal structure of its composite module including possibly access to internal interfaces or documented procedures to replace some of the built-in modules. However, in any case the composite module for the purpose of the requirements defined in this document is considered as a "black box module".

3.4.3
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

Note 1 to entry: A hardware module has hardware aspects. It consists of hardware components.

EXAMPLE 1 A mechanical joint with no electronics contained; its' hardware aspects include its size, the kinematic properties, the mounting plates at both ends, the material, stiffness, maximum allowed force and torque, etc.

EXAMPLE 2 An enhanced mechanical joint, that includes a microcontroller, software on the controller and motors to control properties such as stiffness, or damping; its' hardware aspects also include the connector for powering the embedded electronics and embedded motors, including specifying voltage and current limits.

3.4.4
software module
module whose implementation consists purely of programmed algorithms

Note 1 to entry: A software module has software aspects. It consists of software components.

3.5 Characterization of modules regarding principal function

3.5.1
actuator module
actuating module
output module whose primary function is to physically move the robot, or alter the world around the robot, in response to instructions from other modules, with the purpose of achieving the robot system's task(s)

3.5.2
communication module
module that exposes communication interfaces to other mediums or provide means of interconnection between modules

Note 1 to entry: Interfaces to other mediums can be via Wi-Fi, mobile network, Ethernet, etc.

3.5.3
computing module
module that provides computing resources for use by software modules

Note 1 to entry: Computing resources are the hardware for executing the software, and can include distributed modules.

3.5.4
infrastructure module
module that provides facilities and resources to support the operation of other modules

Note 1 to entry: Examples of facilities used by other modules include mechanical frames for physical attachment points and cables for communication and power, where cables can be attached to the frame.

Note 2 to entry: Examples of resources used by other modules include power supplies, memory and processors, and communication bridges (or hubs) among inter-robots or the robot and the servers.

3.5.5
sensing module
input module for collecting data about the world around the robot or the state of the robot for use by other modules to support the robot system in performing its task(s)

3.5.6
supervisor module
software module which checks the status of other modules and can control the transition from one state to other state to make the operation sequence of modules proper

4 General provisions

4.1 General

This clause introduces the essential concepts behind the use of modularity in service robotics. For describing these concepts SysML (OMG SysML) should be used, which defines types of diagrams within a general-purpose modelling language for systems engineering applications and which also supports specification, analysis, design, verification and validation. Manufacturers should perform verification and validations processes for satisfying the principles of modularity.

4.2 Generic principles of modularity

4.2.1 General

This subclause explains generic principles that a module's design should follow. While these principles are partly presented as recommendations, a module designer shall:

— document whatever modular approach has been chosen; and

— provide all necessary information for integrators to use the module.

These principles can apply generically to modules with hardware or software aspects. In this Clause the term module, unless stated otherwise, is used in its widest sense to refer to basic or composite modules.

4.2.2 Composability

Modules shall be designed in a way that they can be assembled logically and physically into composite modules for performing more sophisticated operations, while maintaining operational and safety requirements. Composing should be possible based on information provided on interfaces so that information of internal structure is not necessary. Modules can be organized in data banks or repositories to make re-use more practicable. This is further discussed in 7.2.2.

4.2.3 Integrability

Hardware aspects and software aspects of modules shall be designed in a way that they can be integrated to form larger systems to perform intended target services or functions. To allow the linking of modules in a reliable manner, appropriate interfaces should be designed. Safety aspects of systems made from modules are discussed in Clause 5.

4.2.4 Interoperability

Modules shall be designed in a way that they can be linked to work with other modules. They should be easy to connect and should allow to share power and data via appropriate connectors. To allow exchange of data, interfacing protocols shall be defined and implemented at appropriate levels, as specified in Clause 7.

4.2.5 Module granularity

The function of modules should be achieved with an appropriate level of granularity in a modular framework: Basic modules and composite modules. Examples of basic modules and composite modules are presented in Annex B.

4.2.6 Platform independence

Modules should be designed in a way that they can be implemented on different service robots or combined with different sets of modules without significant modification. Software modules should be generated in such a way that they should run on different platforms such as embedded computing systems, Linux, Windows or real-time operating systems with minimum modifications. Modules with hardware aspects which are used in different service robot system should be operated on different platforms.

4.2.7 Openness

Openness in this document shall include mechanical and electrical interfaces for modules with hardware aspect and software interfaces among modules should include the following: a defined reference architecture consisting of modules with hardware and software aspects, and their design together with their safety, security and testing methods.

The re-use of modules should be supported by the provision of relevant information such as their dependencies and incompatibilities to integrators.

NOTE Relevant information can include source code, documents, computer-aided design (CAD) models, circuit diagrams, design experience, system architectures, software hierarchies, interfacing specifications, etc.

4.2.8 Reusability

Reusability is the ability of modules to be used and re-used in different platforms via appropriately defined interfaces. Interfaces of modules shall be designed in a way, that modules can be re-used. Relevant interfaces allowing re-use can include software interfaces, connectors between modules, and linkages between hardware aspects of modules.

Where appropriate, reusability should be supported by managing builds, configurations and reconfiguration options, upgrading possibilities and overall maintenance requirements of modules.

4.2.9 Safety

Modules should be designed to comply with relevant safety standard in all safety-related applications. They should furthermore be designed to support the overall safety of a modular system. Manufacturers of modules should provide necessary information to support integrators in safety design of system.

4.2.10 Security

Modules with software aspects or communication interfaces should be designed to prevent attempts to access with them by unauthorised methods or persons. They should furthermore be designed to support the overall security of a modular system.

4.3 Abstraction

An abstraction layer should be used to define standard interfaces between hardware and software in order to:

— support interoperability and reusability;

— simplify simulation and modelling;

— foster independence of implementation and platform-independence.

NOTE 1 For example, an infrared sensing software module and an ultrasonic sensing software module may be used together to get the distance from a robot to a nearby object. These two modules can read the distance values from the infrared sensor and the ultrasonic sensor using their device drivers, respectively. In this case, the two modules may not be reusable and interoperable because each module uses its own device driver even though the same data is used. To ensure the reusability of the two modules to read the distance value, one abstracted device driver is necessary, after which a different sensing module can be used because of the abstraction layer even though many manufacturers provide different types of distance measuring sensors.

NOTE 2 Software aspects in software modules use the abstraction layer to access hardware devices such as servo motors and laser sensors.

NOTE 3 The use of a hardware abstraction layer or another form of device driver is optional in this document (see 7.3). If a particular implementation of a modular robot system can be achieved by directly calling software functions of device drivers, then this is allowed. Abstraction can include the use of translation techniques where underlying communication technologies are different.