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.

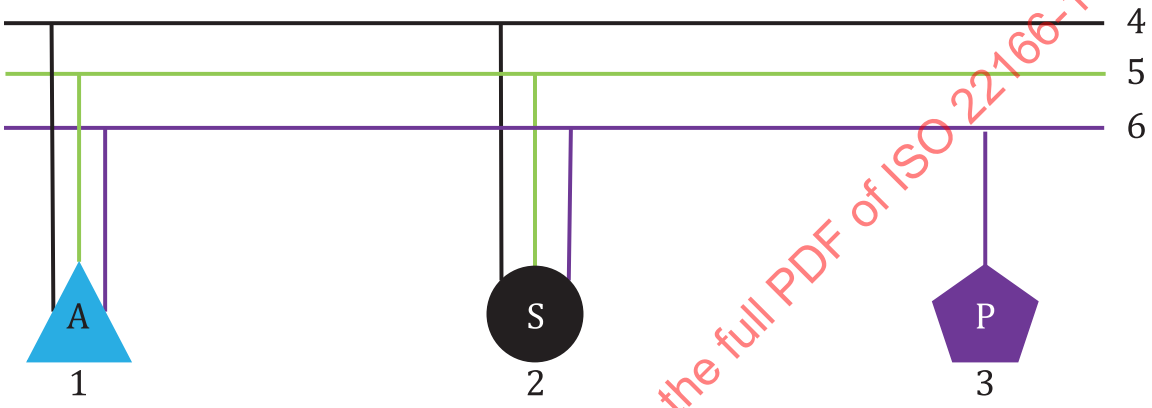
4.4 Electrical interfaces and communication protocols

The electrical interfaces and communication protocols should follow already existing standards.

NOTE 1 Interfaces for data buses and communication networks include hardware and software aspects. A conceptual example of a general interfacing arrangement is shown in Figure 1 covering functionality, powering and the operating environment.

NOTE 2 Table 1 shows some examples of communication protocols. Communications protocols are often implanted in software, but also sometimes in hardware. They represent layer-2 to layer-7 in the OSI reference model as defined in ISO/IEC 7498-1.

The electrical interface hardware should be designed so that the communications do not suffer interference due to close proximity to other electrical wires or devices. Only standardized connectors shall be used.



Key

1	actuator	4	environment
2	sensor	5	functionality
3	power	6	power

Figure 1 — Conceptual example of interface arrangement between modules (elaborated in Clause 6)

Table 1 — Examples of communication protocols able to be used for modules

Reference	Type	Remarks
ISO 11898-1/2 and EN 50325-4/5	CAN and CANopen	The CAN media access unit sub-layer is normally implemented in transceiver ICs, the CAN data link layer protocol and the physical signalling sub-layer are implemented in the CAN protocol controller, and the CANopen application layer is normally implemented in software running on a micro-controller.
ISO/IEC/IEEE 8802-3:2017 IETF 793 or ISO/IEC 14766 (TCP), IETF RFC 768 (UDP), RFC 791 (IPv4), RFC 2460 (IPv6)	Ethernet and TCP/IP	Implemented in PHYs and MAC, which are optionally integrated in technology-specific controllers. (This protocol is widely used worldwide).
IEC 62680 and USB CDC	USB	Many implementations exist and USB communications device class (CDC) is a composite Universal Serial Bus device class.
IEC 61158	Fieldbus	IEC 61158 standardizes commonly used Fieldbus protocols and includes Foundation fieldbus, Profibus, WorldFIP, CC-link, EtherCAT Modbus-RTPS, SERCOS, etc.

4.5 Interchangeability

Interchangeability and re-composition of modules are strongly related to connectivity of modules and can have different levels; the following are considered in this document:

— Level 1: Exchanging of modules is only possible by the manufacturer or robot system integrator

— Level 2: Exchanging of modules by the user is possible when the robot is switched off

— Level 3: Exchanging of modules by the user is possible while the robot is switched on (hot plugging)

— Level 4: Exchanging of modules by the robot itself is possible (hot plugging with activated drives)

Self-configuration (level 3 and 4) may lead to incorrect operation or to hazardous situations. Relevant safety and security issues are discussed in Clause 5. To avoid ambiguity about the state of modules, self-configuration with ongoing robot functions should be avoided.

Module manufacturers shall provide the interchangeability level for modules. The different levels have different implications on requirements with respect to design of connectors, safety and security, durability, documentation of modules, etc., as shown in Table 2.

Table 2 — Recommendations for different levels of interchangeability

Level	Frequency of changes	Design of connectors	Documentation	Safety	Software
1	Low	Can be simplistic with separate connection of mechanical and electrical parts	For readers with technical knowledge	Issues should be covered by a risk assessment performed after exchanging	Installation and configuration can be complex and include manual adjustment
2	Medium - high	Preferably composite plugs	Also for readers without technical knowledge	Safety limits need to be provided for the user. System can do a consistency check on power-up	Organisation in packages, automated solving of dependencies
3	High	Composite plugs with hot-plug functionality	Also for readers without technical knowledge	Safety limits need to be provided for the user. System needs to perform a consistency check while other functions are executed	Automated loading, unloading and switching of modules during runtime
4	High	Composite plugs with hot-plug functionality and large tolerances for automated connection	Also for readers without technical knowledge	Safety limits need to be provided in machine-readable form. System needs to perform a consistency check while other functions are executed	Automated loading, unloading and switching of modules during runtime

An information model for a modular robot system should provide information about the module properties and capabilities of individual modules regarding interchanging and self-configuration.

4.6 Module properties

4.6.1 General

Module properties shall be stored in a module property profile. When a module is transferred for use or reuse, the module profile shall go with the module.

NOTE There is no implication here about the nature of this storage.

4.6.2 Module identification

The module should be named or identified using a string or number code that is published by the manufacturer. In addition, the product itself and the vendor should also be identified using a similar name or identification code. This information can be used in designing a service robot based on modules which (semi-)automatically configure the robot system. If the module uses the data bus, it should transfer its own ID on request to other modules and to the supervisor module.

The module may provide automatic configuration of the robot hardware (including structure) and software.

If the feature to automatically configure itself is available in the module, the minimal information made available to identify the module by the manufacturer should be:

— Module type and/or module ID

— Manufacturer name and/or manufacturer ID

— Module version

— Production date

— Serial number

From a system security point of view, module identification should be verified by a properly designed authentication procedure for security related modules.

4.7 Simulation

If simulation is used to verify the design and function of a module, the limitations and constraints of the models used should be recognized. Especially safety and security should be verified in real-world tests for intended use case applications.

In order to allow proper simulation of a modular system, module manufacturers shall provide relevant information needed for simulation with their modules. It should be specified by the framework designer as to which information is necessary and in which form they should be provided (e.g. on paper or as a parameter file which can be imported into the simulation tool). Information about the module used in the simulation can include:

— its physical characteristics, including its physical properties (e.g. dimensions, mass, density, static and dynamic characteristics, structural strength, etc.), and appearance;

— its electrical characteristics, such as peak and average power requirements for operation;

— the general control algorithms that it can execute;

— the interfaces for input (sensors) or output (actuators) modules, that determine the format and style of the information to be exchanged;

— the method by which a sensor module acquires information from the simulated world; and

— the method by which an actuator module acts upon the simulated world.

NOTE 1 Detailed specifications for the various models are outside the scope of this document.

NOTE 2 It is also possible to provide simulation data within the module template.

4.8 Data types for interoperability

A modular framework shall define data types that can be used within the modular framework and the middleware. This includes the precision of common integer- and real-valued numerical data types as found in IEC/TR 62390.

A modular framework shall also define conventions for commonly used composite data types. Defining the following conventions is recommended. There are also a small number of common composite data-types built upon these [see also OMG RLS (Robotic Localisation Service)], namely:

a) A position in space is defined with respect to some coordinate system that is defined in a fixed position and orientation according to the implementation. A coordinate can be given in Cartesian form with respect to the orthogonal fixed coordinate system using a triple (x, y, z) of real numbers. A pair (x, y) of numbers can also be given, but is interpreted as a triple with $z=0$.

b) Orientations may be specified in one of two ways. A general orientation in three-dimensional space is to be given as a quaternion, which is transferred as a four-tuple of numbers $<c, su, sv, sw>$ where (u, v, w) is the axis of rotation and c and s are the cosine and sine of the half-angle of rotation respectively. Alternatively, a rotation about the z-axis only can be given as a rotation angle about that axis, with the angle measured in radians.

c) The position and orientation of a mobile robot is given by its standard coordinate system as specified in ISO 19649 and ISO 9787.

d) 2D and 3D object geometry data — to be chosen from existing standards.

A modular framework may in addition define guidelines or limitations, how input/output data to/from modules should be structured.

A module manufacturer should pick suitable data types and data structures from the range allowed in the modular framework definition for his module. Information on the data types and data structure shall be stated in the module description (see Annex B for example modules).

5 Provisions for safety and security

5.1 General

This Clause provides guidelines of how requirements from published safety and security standards can be applied to design modules and module-based systems. Its contents shall not be used as a justification to deviate from applicable safety and security standards.

NOTE 1 Safety can be assessed on the level of a single module (normally done by the module manufacturer) and on the level of the service robot system (normally addressed by the integrator).

Safety and security are different design aspects that can influence each other in robot design and in robot module design. A security breach in a robot system and/or robot module can result in safety-related hazards. Therefore, a robot module manufacturer should evaluate, through risk assessment, the potential for security related properties to cause hazards within the intended use cases, which the designer should mitigate through the module design.

A security vulnerability in one module of a service robot system can lead to a security breach for the whole service robot system which can result in hazard(s). A module manufacturer should be aware of security vulnerabilities of a module that can propagate through the robot system. For this reason, safety and security aspects should be addressed to ensure that robot designers take them into account in their modular design.

Existing safety standards apply for robots and robot systems including

— ISO 12100 for risk assessment and risk reduction of machinery,

— ISO 10218-1, ISO 10218-2, and ISO/TS 15066 for industrial robots,

— ISO 13482 for personal care robots,

— IEC/TR 60601-4-1, IEC 80601-2-77 and IEC 80601-2-78 for various aspects of medical robots, and

— ISO 13849-1, IEC 61508 series, and IEC 62061 for functional safety.

In a modular robot system software can be involved in various modules of the robot. ISO/IEC/IEEE 12207:2017 and ISO/IEC/IEEE 15288:2015 have defined the life cycle processes for the development of software to ensure required quality is achieved. IEC 61508-3 specifies safety requirements for software that is part of the safety-related part of the control system. The safety requirements for software apply only to the safety-related parts of the software and are presented in 7.2 and 7.4.

When a modular design approach is adopted, the increased ability to reconfigure a service robot system for multiple applications shall be accounted for in the risk assessment and risk reduction process described in ISO 12100 to ensure that the safety requirements are satisfied even after adding/removing/reconfiguring modules, e.g. by reviewing risk assessment again after a reconfiguration. These requirements can apply on the system level as well as on the module level. In addition to the normal safety-based risk assessment, the designer and/or the service robot integrator shall incorporate a security risk assessment to assess the consequences for safety. For example, a hazardous situation can be mitigated by adding a module. However, after any change in the modular structure of the robot system, the risks after performing a modular reconfiguration should to be assessed again for both safety and security.

The following standards shall be used when evaluating robot system and module security:

— ISO/TR 22100-4 dealing with IT-security aspects for machinery

— ISO/IEC 27032 for general guidelines on cybersecurity

— IEC/TS 62443-1-1 for terminology, concepts and models

— IEC 62443-2-1 dealing with security programs for industrial automation

— IEC 62443-3-3 for security levels in control systems

— NIST SP 800-154 for data-centric system threat modelling

— NIST SP 800-160 volumes 1 and 2 for system security engineering

Figure 2 shows how safety and security risks are related and how they can be addressed. Module manufacturers and integrators (and where applicable module framework designers) shall follow the rules, and requirements established in existing, applicable safety standards. This is indicated in the horizontal line of Figure 2. Security risks for a module shall be assessed with reference to the same intended use, foreseeable misuse, and “limits of the machine” (as per ISO 12100:2010, 5.3) as used in the safety analysis for the module. The process of assessing and mitigating security risks (vertical line of Figure 2) should be performed in an iterative process: either parallel to steps 1 and 2 of Figure 1 in ISO 12100:2010, Clause 4, or at the end of the respective process. The integrated process of assessing and mitigating both security risks and safety risks (diagonal line of Figure 2) should be performed in an iterative process. In cases where implementing security measures conflict with implementing an intended safety function (to meet a safety requirement), mitigating the safety risk shall take precedence, while still mitigating the security risk as far as reasonably possible.

NOTE 2 A module manufacturer can either try a different implementation of the respective safety function, which still meets the safety requirement. Or it can approximate the security measure to extend possibly without hampering the safety function.

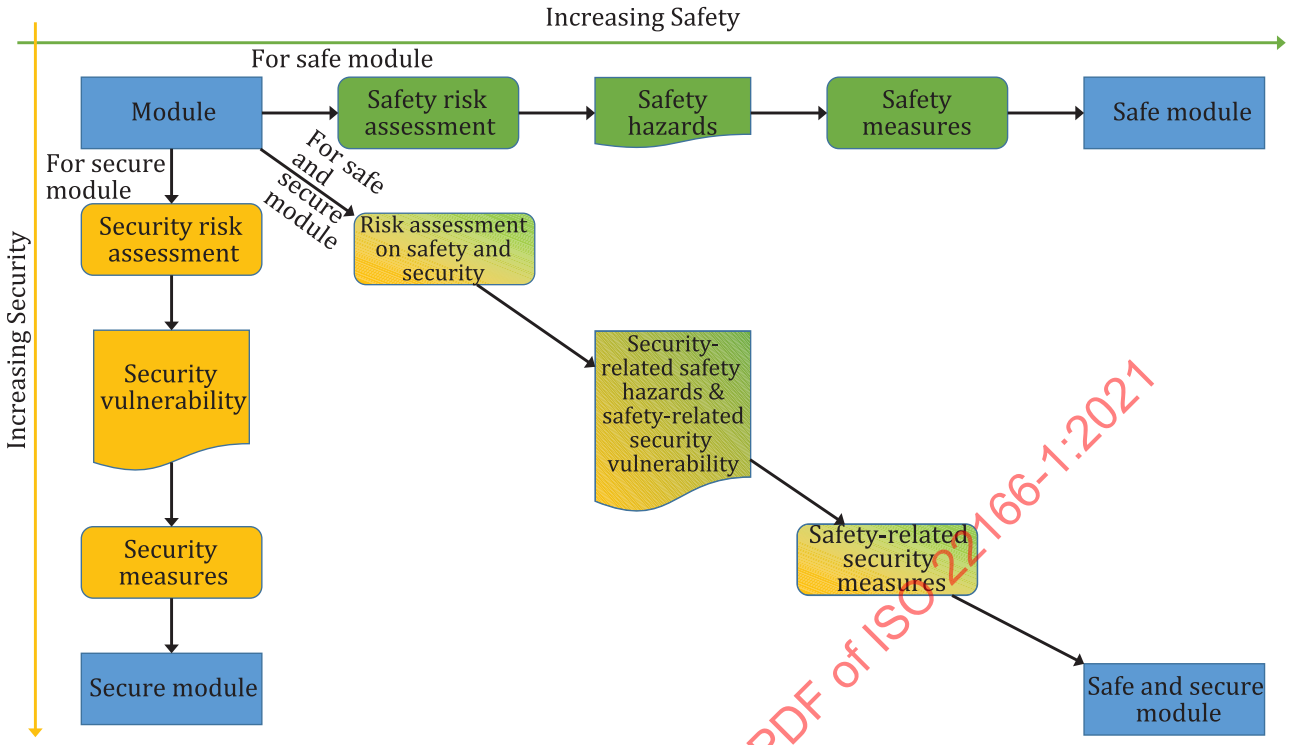


Figure 2 — Safety and security risk considerations for robot modules

A module manufacturer should consider the module's intended use cases and the technology it uses to identify the needed requirements for various application domains (e.g. mechanics, electromagnetics, software, security, environments, biology, chemistry, usability, etc.). In addition, the module manufacturer shall identify and list the considered use cases with concrete limitations and exclusions. While modularity may not be explicitly within the scope of existing standards in the intended application domains, the principles presented in existing standards can be useful to derive appropriate module requirements and testing methods. See Annex D for more detailed information on testing robot modules.

A module integrator should consider the following information:

— Relationship with external system (physical layout, interface, etc.);

— Maintenance guidelines for module and system levels.

5.2 Robot system level safety

The methods used to evaluate the safety of a robot system comprised of robot modules are not different from the methods used to evaluate robot systems without modules, and they are published in existing standards.

The modular service robot framework developers are responsible for:

— Designing the architecture and the composition of the various service robot modules,

— Ensuring proper connection and processing of the safety signals between modules, and

— Evaluating the required safety of the robot for the typical use case applications.

NOTE 1 Typical use case applications can cover different levels of safety; security; combined safety and security; and quality.

NOTE 2 Signals refer both to hardwired signals like emergency stop signals in to or out of a module and also data exchanged over a wired or wireless network, in which case the network should satisfy the functional safety requirements.

The module manufacturers shall be responsible for specifying the intended use case applications for the modules.

The module integrators shall be responsible for:

— Complying with the applicable robot safety standards mentioned in 5.1,

— Accounting for the systems' planned use cases, which is compared for relevance and closeness to the intended use cases presented by the module manufacturer, and

— Complying with safety guidelines specified by the module manufacturer, including the required conditions for use.

If the robot system or its use case applications change, then the risk assessment process shall take in account the changes. Examples are presented in Annex C. If parts of a modular robot system are intended to be changed by the end user, the risk assessment shall cover hazards from possible configurations.

NOTE 3 Applicable limits of use and required safety precautions or follow-up steps are part of the user manual of the robot system.

The modular service robot framework developers shall carry out to design measures to support risk reduction to ensuring the following:

— global safety signals and error states and rules for their use and propagation;

— safety features or a minimum safety performance that all modules should maintain;

— core items to be included in the documentation of safety performance.

5.3 Module level safety

The design of the module shall take into account the applicable published standards for electrical and mechanical safety (see Annex D for testing robot modules). Safety-requirements for software are discussed in 5.4.

If motors are designed to have a stop function, the module should provide a respective safety function in compliance with IEC 61800-5-2.

To prevent failure due to module-to-module communication, the system should use black channel communication (see IEC 62280 and IEC 61784-3). In this case the reliability block of the module should be designed as safety-related and a route should be constructed that guarantees the reliability of the safety function.

Performance levels (PL) or safety integrity levels (SIL) shall be assigned to safety functions implemented. These PL/SIL can be used to evaluate the overall performance of a safety function with respect to the complete robot system. A module manufacturer should publish PL or SIL for all safety functionality shared by the module with other modules. Signals in a module that are not safety-related in that module, may still be useful and reported to other modules for safety-related functions.

The process followed in designing robot modules can be different from normal system design because the module designer/manufacturer does not have the final specific application available at the time of design (only typical use case scenarios are known).

The robot module manufacturer can use the following steps to ensure appropriate and adequate safety design requirements are included:

1. Define intended use cases, and for each use case, describe as many relevant details as possible.

2. A hypothetical robot system design should be carried out for each use case. All foreseeable applications of a module shall be considered. Reasonably foreseeable misuses of a module should also be considered.

   NOTE 1 The assumption that the system has a safety supervisor can be made if necessary (see 7.2 and 7.4).

3. For each intended use case application, potential hazards should be identified (ISO 12100:2010, Annex B, includes a list of potential hazards that should be considered).

4. To perform a safety risk assessment, the module should be considered as an individual module and what potential hazards it can cause in the intended use case applications.

   The safety requirements of the module should be based on some assumed worst use case intended application.

5. The module manufacturer should complete the safety risk assessment for each intended use case, to define appropriate PL for any relevant safety-related functions within the module. It may be appropriate to construct local safety-related functions in the module to take care of this functionality — see "safety supervisor" in 7.2.

NOTE 2 Not all steps may be necessary in all situations.

The module manufactures should document the intended use cases and their assumptions and provide the following information to module integrators:

— Information for use of the module.

— The environmental conditions in which the module may be safely operated.

— Information about the safety related functions in the module.

— Information about data the module provides to other modules, which may be meaningful for safety outside the module (e.g. within the safety supervisor).

NOTE 3 Safety requirements for functionalities such as operator interface and emergency stop are provided in IEC 60204-1 and can relate to modules.

As an example, two possible implementations of a safety system of a mobile robot platform are presented to demonstrate that modules with more (safety related) features are easier to integrate and usable for real-world situations.

EXAMPLE Two composite modules, offered by two different manufacturers have the following features:

— Platform 1 has motors that mechanically limit the maximum speed of the platform to 1 m/s. The platform controller accepts the desired moving speed as an input and outputs the current speed, but both signals are not safety-related and have no performance level rating.

— Platform 2 can reach a maximum speed of 2 m/s. The platform controller offers safety-related speed control with a high-performance level. Therefore, the desired speed input and the current speed output are safety-related.

A robot integrator designs a mobile robot with Platform 1 with a simple safety system comprising laser scanner modules with a fixed protective field, which can stop the vehicle in time, when it runs at 1 m/s.

When using Platform 2 instead of Platform 1, the robot integrator should substantially increase the protective fields of the laser scanner to accommodate to the possible maximum speed of 2 m/s. Instead he decides to use the safety-related speed control function of Platform 2. In this case, maximum protective fields are only necessary, when the platform actually runs at high speeds. For slow docking manoeuvres, the protective fields can be reduced.

The example demonstrates that the system is more adaptable to changing environmental requirements with the required safety features when Platform 2 is used.

NOTE 4 Using speed control and switching of protective fields requires that both the laser scanner module and the safety supervisor module support this functionality.

5.4 General aspects of security

Module-level security should ensure that the individual modules are resistant to unauthorized access to prevent attacks that affect confidentiality, integrity and availability of the module such as:

— unauthorized access to internal data (with possible effects on intellectual property or personal data);

— unauthorized access and altering of the module configuration and internal parameter settings (which may also affect safety);

— attacks causing damage to the module or the modular robot system or preventing normal use.

Tampering in the modules of a robot system should become a safety hazard as the robot system, or parts of it can make uncontrolled movements due to the safety system becoming impaired. To validate the module level security, see Annex D.

Cyber security is an evolving domain and needs to be taken in account for the module design so the latest developments should be considered for inclusion. The design method proposed has a strong similarity with the steps provided in previous sub-clauses for safety.

NOTE 1 No security performance levels are currently available which can be referred to.

Measures to protect a module and a modular service robot should be chosen according to the results of the security risk assessment, taking into account the following:

— Exposure to potential intruders (insiders and outsiders);

— Potential harm that can be caused by unauthorized access (e.g. effects on availability or safety);

— Potential motivation of intruders to gain access (e.g. access to valuable private data).

To achieve system level security, all connected modules that exchange data should be modules which provide sufficient protection for unauthorized access to physical data ports. Internal communication inside a module, communication between modules and communication outside the robot system should be treated differently.

NOTE 2 In almost all modern safety devices and safety-related functions in machinery and robot systems, some kind of communication and (embedded) software is present. Almost any software potentially can be influenced in such a way that the behaviour and safety functionality can be changed. If such a module shares data with other modules or outside the robot system the chances for unauthorized access and influencing can be larger. Clause 7 describes more details.

This document uses combined levels of safety and security to classify a module as follows:

1. No safety or security needed: This non-requirement for combined safety and security can be applicable for small and light robots that cannot harm a person and are not connected to any outside system.

2. Security needed: applicable where the module's intended use includes communications within the robot or with external systems. Clause 6 describes the hardware related security measures and Clause 7 describes the software and communication related measures to achieve the security for a module.

3. In some cases, it is possible to design a safe but potentially unsecure system; it should depend on the application if this is acceptable.

4. Combined safety and security needed: A system can only be considered safe and secure if both safety requirements and security requirements have been addressed. Figure 2 shows the process to follow for ensuring adequate safety and security risk assessment.

NOTE 3 A system that has only hardware and no software, like a safety switch, this is one of the few exceptions that can be considered safe without being secure.

5.5 Steps to design security into a module

The robot module manufacturer should use the following steps to ensure appropriate and adequate security design requirements:

1. Define the use cases for the module from a security perspective. These use cases can be similar to the ones defined for safety, but can differ as well.
2. Foreseeable intended uses and potential applications of a module shall be considered.

   NOTE The assumption that the system has a security supervisor (see 7.2 and 7.4) can be made if necessary.
3. The designer should complete the security risk assessment for each use case to derive security requirements that the module should maintain for facilitating module and system security.
4. The software in the module should comply with the security requirements of 7.4.
5. Check the guidelines for secure data exchange are provided as presented in 7.4.
6. Check the hardware guidelines of 5.7 to protect the module against unauthorized access.
7. The security risk assessment should be conducted on each module independently and the consequences assessed in use-case scenarios defined in step 2.

5.6 Physical security of modules

A module manufacturer should consider the following aspects of physical security for module design while an integrator should consider them for the modular robot system:

— Security of communication ports

— Physical access to internal components from the outside

NOTE Modules can pass through bus systems to neighbouring modules. Thus, security breaches can propagate from one module to another.

Module manufacturers and integrators shall consider the following measures to restrict access to a communication port of a module or system, for example:

— Latch sensor (no security is required but a need to know if in an open or closed state)

— Mechanical lock with a physical key

— Mechanical lock with a latch actuator

Modules with no security measures should only be acceptable in a protected environment like an internal research lab environment or for small and light service robots.

5.7 Cyber security of modules

A module (or its firmware and software) should

— prohibit unauthorized tampering;

— provide security for the data stored, processed and exchanged by the module;

— provide communication security.

NOTE 1 The necessity to apply cyber security measure depends on the intended use case of the module.

The cybersecurity of a module should be designed to achieve the following security objectives: confidentiality, integrity, and availability. In carrying out the cyber security risk assessment and risk reduction, the following should be considered:

— Examples of measures for confidentiality: Secure booting, authentication (e.g. passwords), data encryption

— Examples of measures for integrity: Access control and permission, checksums

— Examples of measures for integrity and of confidentiality: Data security, Secure communication

— Examples of measures for availability: Secure code updates, adequate communication bandwidth, redundancy

NOTE 2 General security aspects for industrial automation systems are covered by the IEC 62443 series of standards. A standard is under development that is aimed at covering security aspects of machines; IEC/TR 63074 presents security aspects related to functional safety of control systems.

6 Hardware aspects in module design

6.1 General

This Clause describes the requirements and guidelines to enable the interoperability and reusability of modules with hardware aspects including hardware modules. For modules with hardware aspects, Table 3 shows the main connectivity issues that shall be considered for realizing an effective modular design framework and so achieving the interoperability, safety and security requirements as presented in this document. Examples of modules with hardware aspects are shown together with how the connectivity issues shall be addressed. Designing a hardware module or a module with hardware aspects, such as an actuator, can require considerations of safety, security, power, signalling as well as the mechanical structure.

NOTE The connectivity issues of hardware modules or modules with hardware aspects can be either physical (e.g. Power, Data) or can refer to more abstract interactions (e.g. Safety, Security, Environment, Mechanics). If a module is for example connected to Security, this module has security issues or exchanges data in some way with other security-related modules.

Table 3 — Connectivity issues for modular framework via example modules

Module\Interaction	Environment	Mechanics	Data	Power	Security	Safety
Actuator (A)	✓	✓	✓	✓	✓	✓
Power supply (P)			(✓)	✓	✓	✓
Sensor (S, digital/analogue)	✓	✓	✓	✓	✓	✓
Computing with software (CS)			✓	✓	✓	✓
Supervisor (SU)	✓		✓	✓	✓	✓
User interface (UI)	✓		✓	✓	✓	✓

6.2 Requirements and guidance for hardware aspects of modules

6.2.1 Mechanical interfaces

6.2.1.1 General

The module shall be provided with specifications of its mechanical interfaces and connectors, for example:

— Specification of connectors and interfaces;

— Specification of connectors with placeholders (blind or empty connectors for connectivity not needed by the module);

— Specification of multiple physical sizes of connectors for different requirements of physical durability and size, e.g. for an intended use for different parts of a manipulator;

— Specification of mechanical link for loop of databus and/or power;

— Specification of interfaces which allow data buses or power being looped through the module, even if the module itself does not require connection to them (e.g. in case of a mechanical link).

NOTE 1 While integration of connectors in the module is advised, it presents a specific design challenge to combine the physical motion of attaching and detaching a module mechanically, while at the same time connecting and disconnecting the connectors for data, power, safety and security to maintain their intended operations. If this robustness design specification does not fulfil the requirements for the intended use, the module can impose a safety and performance risk(s), eventually resulting in malfunction and failure.

Proper testing and validation of the attachment and detachment of the module to other modules should be performed. If applicable, information for use shall state that testing and validation is required. The module shall be accompanied with the information to enable testing and validation of the attachment and detachment of the module to other modules. Information for use shall include information to ensure connectivity and functionality between modules with hardware aspects, for example:

— module alignment, module positioning, and module locking with the desired stiffness in static and planned dynamic motion cases;

— that data, signal and power connections are not damaged during the mechanical positioning and locking process of the module interface;

— the mechanical connection/locking mechanisms to achieve the specified accuracy and stiffness in the mechanical connector for the module's intended use cases.

Considering the module's intended use and the specified use cases, the number of times the module can be disconnected and reconnected for the module's specified life. For example, the following designs can be used as applicable:

— Using a progressive interference fit to bring physical contact points together without damage;

— Using coaxial and/or conical structures to lessen angular or lateral motion during mating to avoid wear, tear, and damage to contact points;

— Using toroidal structures to create multiple contact points to increase the distribution of physical connection to achieve the improved accuracy in the mechanical connection;

— Using compliant materials and structures to create mechanical connections that are more compliant to distribute mechanical forces through an extended structure to avoid a single point failure.

NOTE 2 Interface standards for industrial robots can be used for modules of a service robot, for example: ISO 9409-1, -2 and ISO 11593.

Module manufacturers should make the specification of mechanical interfaces available so that it can be used by other module manufacturers or integrators. This can include:

— CAD data of mechanical parts;

— Manufacturer and part number of plugs;

— Pin assignments.

6.2.1.2 Connection accuracy and reliability

In service robot modularity design, connections between modules should have following connectivity characteristics for intended use, examples of which are shown in Figure 3:

— Power

— Data

— Security

— Safety

— Mechanics

The safety-related modules shall ensure safety for connection between them. Modules shall specify the accuracy for connection between them.

a) Connection between joints **b) Connection between joint and end-effector**

Key

1 mechanics
2 data
3 power
4 security
5 safety

Figure 3 — Example connectivity characteristics needed through a modular joint

Information for use shall include the specification of the module's connectors for reliability, including:

— amount of translation and rotation to play after joint is locked. That is, no play shall be possible once joint is locked;

— robustness and reliability parameters of the module interface. Wear and tear of the mechanical surfaces for alignment and the integrated connections for power, data, and safety in order to withstand a minimum number of cycles for attachment/detachment;

— durability properties of the module interface. In use cases where modules are frequently changed or where extreme conditions like dirt and overloading are likely, the module shall have been verified and validated for the specified number of cycles. The number of minimum cycles should be specified by the manufacturer.

Module manufacturers should at least fulfil the requirements defined in the modular framework or define their own specifications to meet their intended use case applications.

6.2.1.3 Connection stiffness

The module and the module interface should have sufficient stiffness to transfer static and dynamic forces and torques from module to module via verification and validation process, commonly called the design envelope. To limit the amount of geometric deformation of a module with respect to the other end of the module, the maximum loading torque and force can be specified over the three axes (x, y, z) at the module interface or at the other end of the module.

Forces and/or torques that can be applied shall be specified for the module, including its interface, so that following requirements are met:

— Maximum geometric deformation at the other end of the physical module is less than a specified amount.

— Maximum rotational deformation at maximum twisting loading is less than a specified amount.

6.2.1.4 Mechanical connectors and connections

The module should be able to be attached with minimal or no tools if possible. If the module is relatively small, manual (i.e. unassisted hand) attachment should be possible. For heavier modules, where a hoisting support should be employed, the mechanical interface should be designed to accept mountings involving more force, bumping of interfaces, or high-speed contacts, etc. without damage being caused to the mechanical interface.

Specified testing should be recommended by the manufacturer for assessing the durability of the attachment/detachment method adopted in the particular use case applications (see Annex D).

If connectors are integrated in the module, proper instructions for safe attachment/detachment of the module should be provided.

NOTE A special kind of connector is advised for single cable solutions especially in motor drive and motion control applications where the connector integrates the mechanical interface, power, data, and safety signals, and that proper instructions for safe connecting/disconnecting are provided.

Electrical connectors shall comply with requirements presented in IEC 61076-1 and/or IEC 61984 as appropriate.

Selection and positioning individual connectors should ensure that the following is maintained:

— the resulting force/motion trajectories are within the specified limits;

— power supply requirements for electrical, pneumatic, hydraulic and mechanical forms of energy;

— requirements on data communications and their integrity; and

— compliance with published relevant safety requirements (see Clause 5).

The mechanical loads and forces on electrical, pneumatic, or hydraulic connectors should be considered as part of the process of designing the module's integration details. These should ensure the following:

— Correct physical interaction of different connectors dimensionally and electrically;

— Minimisation of EMC/EMI between different connectors in the interface;

— No leakage of liquids or gasses in the case of fluid power transfer through integrated connectors in the interface.

6.2.2 Interfacing for power supply

Power supplies provide power or energy to all actuators. Manufacturers should select the appropriate power type, such as electrical (AC or DC), pneumatic or hydraulic, e.g., manufacturers should focus on supply voltages that are widely used such as 5 V, 12 V, 24 V or 48 V.

Manufacturers shall specify the rating and maximum output loading capacity of power supplies. Modules should be designed to have a minimum reserve for attachment of additional modules. If modules can be rearranged arbitrarily, it cannot be determined in advance what the maximum power is that can be drawn through a certain module.

EXAMPLE Each arm joint needs 5 A current, so if six of them are connected serially in an arm, the first module needs to be capable of passing 30 A.

Electrical power supplies can have a battery or other energy storage systems, and can work together with a power management system to implement intelligent functions.

6.2.3 Other aspects for module description

For each kind of module with hardware aspects, the important features or data should be specified, examples of which are:

— Kinematic and dynamic properties like geometric parameters, mass, centre of mass, moment of inertia and coordinate transformation;

— The ingress protection (IP) classification as defined in IEC 60529.

If relevant, the following aspects related to the operational environment should be defined:

— Operating environment conditions like range of temperature and humidity;

— Bio-compatibility for applications involving contacts with persons.

NOTE Bio-compatibility issues include cytotoxicity, sensitisation, irritation/intracutaneous, acute systemic toxicity, sub-chronic toxicity, genotoxicity, implantation, hemo-compatibility, chronic toxicity, carcinogenicity and bio-degradability.

For sensors and actuators, module specific features should be described, examples of which are:

— Accuracy and resolution;

— For sensors: sensitivity, sensing range, frequency response if any, and pose in internal coordinate system if applicable;

— For actuators: accuracy, maximum and rated power/torque, maximum and rated speed, and pose in internal coordinate system if applicable.

7 Software aspects in module design

7.1 General

This clause describes the requirements and guidelines that are designed to enable the interoperability and reusability of modules with software aspects, considering the special needs of software modules that can be used in a service robot system. An information model is used to achieve interoperability and reusability. Therefore, an appropriate information model should be provided with a module. As the internal details of such modules are not in the focus of this document, this clause focuses on interfaces between modules, which define the external inputs and external outputs of modules. As different modules with the same functionality should be interchangeable, the types of data flowing into and out of such modules needs to be defined by specifying which communication models are allowed for the used application layer services. Examples of the communication models are the publish/subscribe model; the client/server model; and the blackboard shared memory model (see Table 4). Software modules for robots can be developed based on a middleware framework, examples of which include ROS, OpenRTM, OPRoS, and OROCOS. Safety and security aspects of modules with software aspects are presented in Clause 5.

Table 4 — Software communication interface models for different purposes

No.	Information type	Supported information exchange model	Note
1	Data	Publish/subscribe model	Data can be transmitted via one or more communication models. Data is exchanged between modules, between integrated development environment (or tools) and module.
		Client/server model	
		Black board (shared memory) model	
2	Package	Client/server model	Files are exchanged between integrated development environment (or tools) and module.

7.2 Information model

7.2.1 General

Modules with software aspects for service robot system shall provide the software interface to access input/output data, invoke services, or process events. Hence a software component internally provides some functionalities such that data can be modified via a communication API or message format, or a remote service is invoked, and the results are returned; and the proper process is operating at the occurrence of events, where the remote data and the remote service are provided by other software components.

Also, modules with software aspects can have access to hardware components and be able to read the modules' profiles to initialize and properly operate them. This can be directly, or via a device driver or a HAL which allows software modules access to the hardware components without modification of the module's code.

In addition, message formats are provided for the control and maintenance of software, such as the downloading and uploading of files (e.g., software, profile, and application packages), and execution control of software modules (e.g., start, stop, suspend, resume, and so on).

NOTE Software modules can be defined using existing specifications such as OMG RoIS (Robotic Interaction Service) for service interface or OMG RLS (Robotic Localization Service) for representing locations and coordinate systems.

7.2.2 Model for exchange of information among modules

This model shall be used to exchange information among modules, where information includes the values of variables, invocation of services, process of events, and the contents of files such as the executable code of software components, a profile, or a package. The variables, the services, and the events are defined in modules with software aspects. The type of variables is classified into periodic variables or aperiodic variables and the type of services is classified into blocking (or synchronous) services or nonblocking (or asynchronous) services.

The protocols among two or more modules with software aspects are not specified because many internationally standardized and de-facto communication protocols exist. Note that remote access to the remote hosts is performed using the message format in this clause which is provided by the middleware. The middleware can also support exchange of information among software modules in the local host.

NOTE Local host and remote host here means the computing module as a software module that is logged into at present and other computing module the software wants to connect to via communication protocols, respectively.

The model for exchange of information among modules with software aspects shall support the following:

a) Read and write data;

b) Invoke services;

c) Enrolment and processing of events;

d) Quality of Services as required for items a) to c). (e.g. safety-related values, real-time characteristics, security).

The response time in real-time cases should include the overall data transmission and service invocation times.

The model should support at least one of the following preferred methods to read and write data in instances of other software modules:

— Request with response, request without response;

— Subscribe/publish;

— Black board (via shared memory).

Manufacturers can adopt other methods as they emerge, but interoperability requirements shall be provided within the module template.

The module manufacturer should design the message format for the exchange of information to satisfy the followings:

— Support a middleware.

— Support the encoding/decoding rules for exchange of information among two or more middleware.

— Support information for this Sub-clause can be found in 7.2.3, 7.2.4 and 7.4.2.

7.2.3 Model for access to properties and its access

A module with software aspects shall use its properties' values for proper execution and the setting of values for its initialization. The module shall have the following properties:

a) Manufacturer's information for the module;

b) Execution environment, for example, OS type, execution type (periodic, sporadic, non-real-time, real-time, etc), execution period if execution type is periodic, etc.;

c) Supported communication mode (e.g., publish/subscribe, client/server, black board, etc.);

d) Security level (confidentiality, Integrity, Authentication, the numbers of bits in the key);

e) Safety-related information (e.g. required PL or SIL markings).

In addition, the module should have following properties:

f) (Blocking or nonblocking) service invocations provided externally;

g) Information provided externally;

h) Initialization values necessary for proper execution; and

i) Appropriate software and hardware requirements for ensuring module operation and safety.

If a module requires a specific sequence of events and/or commands to become properly initialized or if modules require a specific sequence of switching-on events, a module such as a supervisor module shall manage those sequences.

EXAMPLE 1 All wheel modules should be put into correct operation before the upper parts of the robot system are put into operation.

EXAMPLE 2 A laser sensor module or a camera module should be initialized and operational before it can be used for safe navigation.

These sequences at the service robot level need to be implemented and configured by the system integrator and may be controlled by a module such as a supervisor module.

A module with software aspects shall provide the functions to read a profile, set properties from the profile to the software components, and write the modified properties to a profile, where properties are specified The module shall use functions defined by the model to read the profile in order to initialise the software components, provide the services of the software components, and access data stored. This module shall support the following:

— Set property value(s)

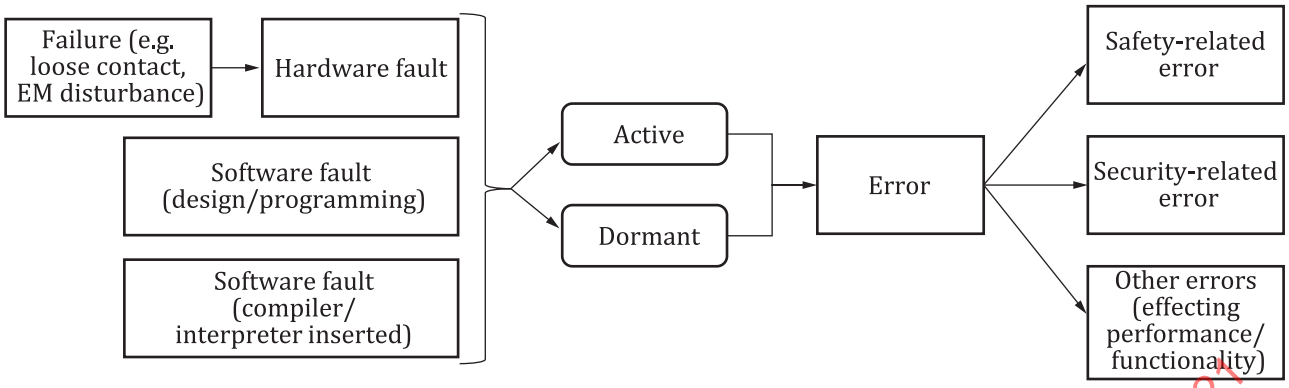
— Get property value(s)

7.2.4 Model for error handling and recovering

Errors in modules can cause a service robot to malfunction or not operate normally. These errors can lead a robot service to a hazardous situation. To avoid such scenarios, faults shall be detected as soon as possible, and it shall be ensured they are corrected and hazardous situations are prevented.

NOTE 1 An error is the explicit representation of a fault.

Failures shall be separated into safety-related ones and non-safety-related ones, as shown in Figure 4; this can be carried out via the safety/security manager. Note that safety-related failures can occur as a result of non-safety-related ones, depending on the application and operational environments.



NOTE Errors can be caused by software or hardware faults; the latter being caused by hardware failures. Faults can be dormant, that is not affecting system operation, until activated by some trigger, such as a specific combination of system states.

Figure 4 — Safety-related and non-safety-related failures

A module with software aspects handling errors shall support the following to handle and recover from error situations:

— Send and receive, error status and error recovery data to/from external modules (Figure 6) such as a safety manager module (see 7.4);

— Classify errors into safety-related errors, security-related and other errors. which are specified by application;

NOTE 2 Non-safety errors are included in other errors.

— Support the execution life cycle (Figure 6) for safety (see 7.3);

— Provide handling methods for unknown errors.

Module designers should define appropriate reactions according to the classified error types. For safety-related errors it can be required to propagate promptly the error to the system level (e.g. a safety manager module). Also, security-related errors can require handling on the system level (e.g. a security manager module). Other errors are handled on the lower level as possible (e.g. in the module itself).

Modules designated to identify, and handle errors should have sufficient reliability. The performance level of such a module should be at least as high as the required performance level of any safety function associated with handled errors.

If there are two or more external modules able to handle the same errors, those modules should have priorities set for sending of the response/command to the errors.

7.2.5 Interoperation of software modules

Modules should be able to communicate and interact with modules developed by different manufacturers.

The following items shall be provided in a module data sheet for ensuring effective interoperability between service robot modules:

a) Information exchanged among modules when needed (see 7.2.2);

b) Information for management of the module (see 7.4.2);

c) Information used in the property profile for the module (see 7.2.3); and

d) Information for error handling and recovering (see 7.2.4).

The following item should be provided for ensuring effective interoperability and reusability between service robot modules:

e) Defined information model between modules and middleware (see 7.2.2).

The following item can be provided for ensuring effective interoperability and reusability between service robot modules:

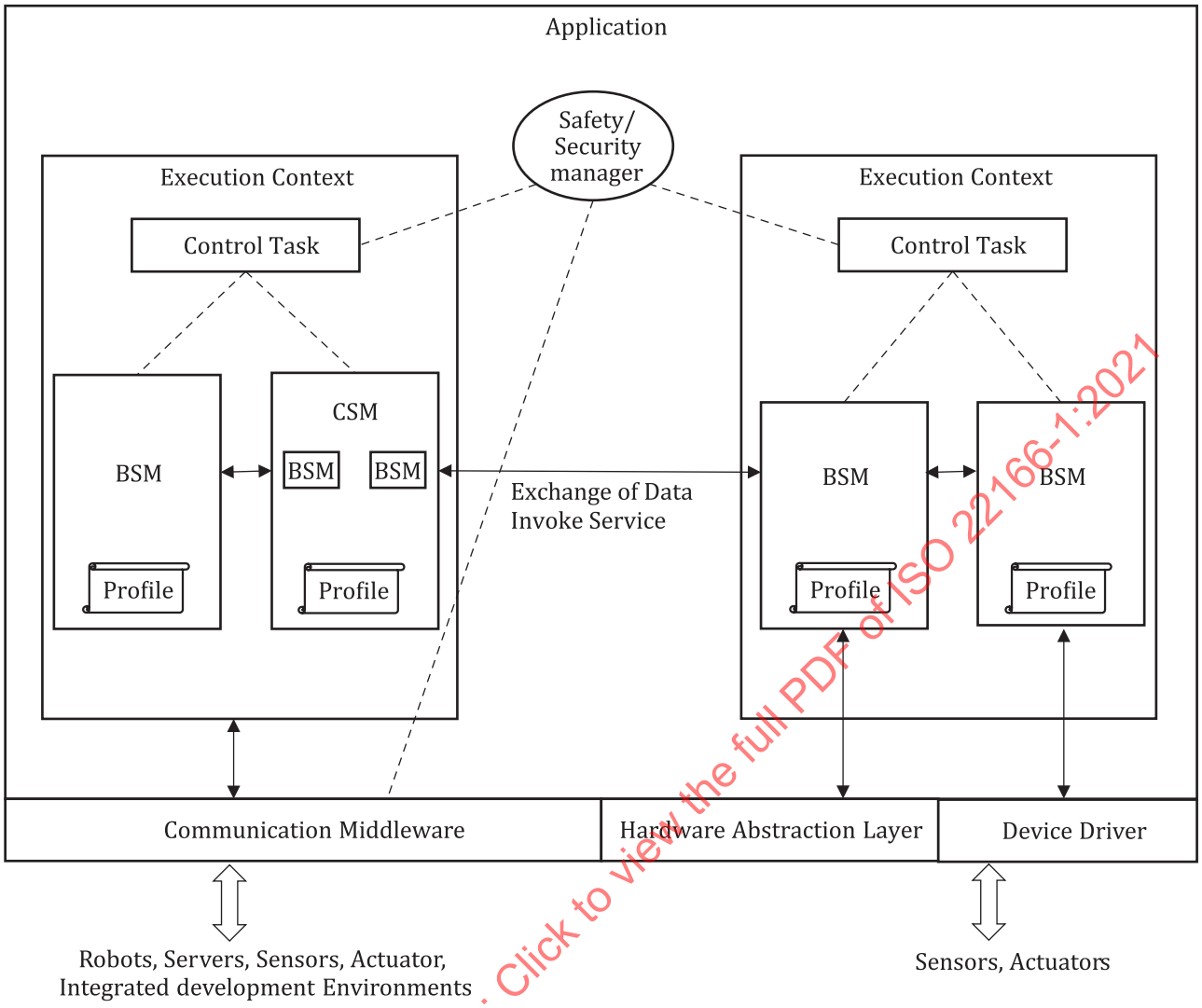
f) Defined model for the hardware abstraction layer or device driver.

NOTE The property profile is stored in the profile repository.

7.3 Architectural model for software modules

7.3.1 General

An architectural model for software modules shall include execution context and control tasks. The model used for safety and security should include a safety manager and a security manager. Software modules and their interrelationships are illustrated in Figure 5, which shows an example of several interconnected software modules. Some of the modules are basic software modules, whereas others are composite as they can be decomposed into smaller modules. Two execution contexts are shown, which are essentially isolated threads of control which can be hosted on different processors (or not). A separate security and safety manager observes the behaviour of the module as a whole, and communications with other modules takes place through the hardware abstraction interface/device driver and the communication middleware. The safety or security managers are implemented separately as independent modules. The safety manager shall only receive relevant data from safety-related software modules.



Key

BSM Basic Software Module

CSM Composite Software Module

- - - - - - control path

Figure 5 — Software framework architecture for service robot modularity

A profile repository manages the profiles used by modules.

An execution context is the element which consists of one or more software modules and one control task. The control task coordinates the software modules in the execution context and manages their real-time constraints if any.

An application is an element which controls the robot system according to user needs and consists of one or more execution contexts. The application utilizes the application package, which includes the software modules, the values and steps for initialization, and related-resources for execution of the application.

Abstraction mechanisms, such as the hardware abstraction interface, help software modules access the hardware independently of hardware-dependent characteristics. Software modules can read/write data from/to the corresponding hardware through the abstraction mechanism, which enables the portability of the software modules. The modules including the software modules access the sensing/

actuating parts using the abstraction mechanism to get data from the devices and pass data to other modules.

The communication middleware empowers software modules and software components to exchange information. The middleware can look after files related to the software modules, components and application and upload/download relevant needed files from/to the server and/or the robot. The communication middleware is able to be implemented in the execution context according to the information exchange model shown in Table 4. Note that the middleware is not defined in this document.

The security manager shall manage security issues which have occurred in both the software modules as well as in other parts as needed. For example, the security manager can monitor and control risks such as access by unauthorized users.

The safety manager shall manage safety-related issues which occur in all software modules as well as other parts as needed. For example, the safety manager should monitor the execution state of software modules, detect whether a limit is violated or if the robot is entering into a hazardous situation, and if this is the case, bring the robot into a safe state.

7.3.2 Requirements for software modules

Modules with software aspects comprise executable code and a profile, where the profile stores values of the module properties for supporting the module's proper execution.

EXAMPLE 1 Module properties examples: version number, OS type, service methods provided, execution type such as periodic execution, sporadic and non-real-time and relevant hardware-related module properties. Module properties' values examples: values for initialization, values necessary for execution of the software module such as OS type, supported communication protocols, and supported service types and event types.

EXAMPLE 2 A basic software module example: distance calculation module, which reads the measured distance data via the hardware abstraction interface from appropriate hardware (such as an ultrasonic sensor, an infrared sensor, or a laser sensor), converts the data to data with the correct standard format, and send the converted data to other software modules. An example of more complex modules would be a stereo distance measuring module or an object detection module running on top of an image stream received from a sensing (camera) module.

EXAMPLE 3 A typical example of a composite software module is a manipulation software module consisting of basic software modules such as actuator controlling modules, axis synchronization module, an inverse kinematics module, and a path planning module, which are described as examples in Annex B.

The software module shall be designed to satisfy the following requirements:

a) Support exchange of information with other modules via a defined information model (see 7.2.2);

b) Support Quality of Service (e.g. real-time characteristics) provision if specified;

c) Have a unique identifier and obtain values of module properties necessary for proper operational and interoperability;

   EXAMPLE 4 Information for reusability, interoperability, and composability of the software module examples are OS type, communication protocol type, the interface type for the service and the data type used.

d) Create one or more instances with unique identifiers for each software module in the application;

e) Be controlled by the control task which manages its execution life cycles as shown in Figure 6;

f) Support modular-level safety depending on error types likely to occur in the software module, its module properties profile and its connection with other modules;

g) Support modular-level security if the module can access the external modules;

h) Have a profile which includes values of module properties defined in 7.2.3;

i) Support software platform-independence.

NOTE 1 This document allows software modules or software components within modules to be executed via a variety of programming languages, under different operating systems, with different document file formats or databases.

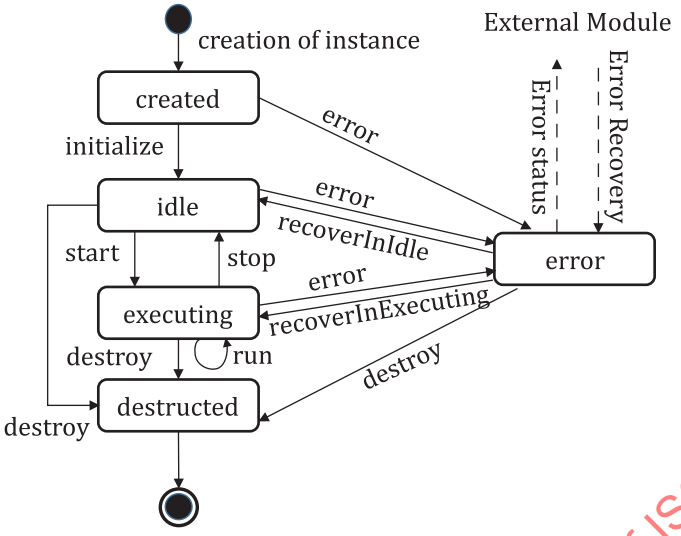


Figure 6 — Execution life cycle of a software module including error handling

Software modules should comply with the execution life cycle shown in Figure 6 which is operating the following behaviours: When a software module is created, the module enters into the 'created' state. Event 'initialize' occurs when the software module is initialized. Event 'start' occurs when the execution of the software module is started. Event 'stop' occurs when execution of the software module is stopped and event 'run' occurs at every given period. Event 'destroy' occurs when the software module is unloaded from memory after the completion of its execution or removed. Event 'error' is generated when an error occurs at any state of the software module. Events 'recoverInIdle', and 'recoverInExecuting' are generated for the error to be recovered in the corresponding states of created, idle, or executing. In particular, 2 types of events, 'recoverInIdle' and 'recoverInExecuting' are designed for the error recovery operation. Note that each event causes the related function to be called; periodically a real-time module performs the related functions using the event 'run'.

NOTE 2 When a safety-related software module is in an 'error' state, safety-related errors driving safety-related failures are sent to other external modules which process the errors and return the proper recovery values. An example of an external module can be a software module or a module that can handle errors to avoid entrance into hazardous situations. The typical example is a safety manager as shown in Figure 5.

For errors processed in the safety-related part of the control system, error recovery procedures (especially recoverInExecuting) should follow ISO 12100 and ISO 14118 to prevent hazards due to unexpected start-up.

7.4 Safety/Security-related requirements for modules with software aspects

7.4.1 General

The safety-related software modules shall be designed based on Clause 5. The security of such modules is related to cyber security and is described in 5.7 to 5.10. This Sub-clause describes the safety/security manager module (see Figure 5) to manage safety/security issues that cannot be handled inside the module. Note that the safety/security manager module can be implemented as one integrated module or two independent modules, namely a safety module and a security module. The modules can also be implemented as redundant architectures to satisfy the relevant PL/SIL.

The security manager module is a module that manages the security of the robot and its modules and can set or implement the security policy to manage responses to security problems. Note that security problems can occur when one module exchange data, such as values and files, with the external modules

or when unauthorized users gets the right to be able to access the robot without valid permission, etc. When one module exchanges data with an external or internal module, the corresponding data should not be eavesdropped on nor modified by appropriate cyber security measures such as encryption and authentication. When programs or profiles are downloaded or control commands are received from message senders outside the robot, authorization of message senders shall be monitored and controlled by the safety/security manager module.

7.4.2 Interaction with safety/security manager modules

Safety-related modules shall provide the following information to the safety manager module to handle modular software safety:

— Error information provided by the module

— Error recovery information that the module receives

The safety manager module comprehensively shall handle the error information received from each safety-related module and provides information to perform a stop or safe operations for each module. The stop operation can be divided into stopping the robot operation and stopping the operation of the modules related to a specific event. Stopping and restarting should follow applicable safety standards such as ISO 12100 (general) and IEC 60204–1 (for stopping), and ISO 14118 (for start-up).

While one module exchanges data with an external or internal module using communication methods, integrity and authentication should be verified. In particular, integrity and authentication should be verified while communicating between external development/monitoring tools and servers. In the case of using a fieldbus that does not support security, physical security should guarantee that only authorized users can physically access the fieldbuses. In addition, the cyber security should guarantee the transmission of the following data if necessary:

— Package, software modules, and their profiles

— Control of the execution status of each software module

— Input and output data of modules

The security manager should be operated in conjunction with the safety manager so that it can operate according to the robot's own policy even if the robot does not communicate with the outside due to denial of service attacks or other similar problems. Hence the safety/security manager module should have the following functions:

— The security manager sends safety-related information to the safety manager if the security manager detects safety-related security problems.

— The safety manager controls modules according to the specified safety policy, which is pre-set.

8 Information for use

8.1 General

Module manufacturers shall provide sufficient documentation with their modules, such that third parties can make use of the module (such as integrate into a larger system, or create other modules that interoperate with the provided module) based on the provided documentation. Module manufacturers should provide list of standards the module complies with and provide all documentation required by those standards. This Clause contains the requirements for additional documentation to support modularity.

The service robot integrator should complete the information for use for the service robot system with all information necessary for the user of the system. This should include the following:

— Providing manuals for the complete system.

— Adding or replacing warning signs and other markings and indications on the robot.

— Providing a system plan which indicates all modules from which the robot is built with their connections.

Service robot integrators should make the information for use of each module in the robot available to the user of the service robot system.

Service robot integrators shall state in their documentation, which modifications (e.g. exchange of modules) of the service robot system are allowed by the user.

Information for use consists of information for proper use of a module to perform its intended tasks with the module. The user may be, but is not limited to, a robot manufacturer, a module designer, a module tester, or involved in module maintenance.

Markings, symbols and written warnings should be understandable and unambiguous to provide details of the module. For basic modules, information should include type of module (input, computing, processing, infrastructure and output). For composite modules, sufficient detail on the different basic and commonly used modules used should be provided.

Signs such as pictograms can be used for explicit representation warnings or for illustrating the operating environments. All printed markings should be legible and durable. Marking regarding safety shall follow respective requirements and guidelines from existing safety standards. Pictograms should be preferred over written warnings, if possible, to make easier use of the module in different regions.

The module manufacturer should provide both printed and electronic versions of the information for use, and consider human factors and usability of the documents.

The description of a module should use the Robot Module Template specified in Annex A. Additional information not covered by the template should be presented in a similar format, where applicable.

Where indications are provided, they shall be described either by markings on the module or in the documentation of the module.

8.2 Markings or Indications

Markings on the module should exist with recognizable patterns on the outside of the module. The marking should be as detailed as needed but as a minimum, should comprise the name or the equivalent mark of the module supplier, and the model or type number of the module and markings for normal use, including all markings or indications required by published and relevant safety standards.

For a module with hardware aspects, the marking should be visible, legible, and indelible and shall at least comprise the following:

— Manufacturer's name

— Serial number

— Certification Markings for safety and security, if appropriate

Software modules shall at least contain the following information in their documentation, such as user manual or text files on electronic storage media used in distribution of the software module:

— Manufacturer's name

— Software module type and version number

— Operating system type

— Serial number

8.3 Information for users

The Information for users for the module should be provided for its proper and intended usage. The Information for users should comprise the following:

a) Detailed description of the module

- Instruction for use of the module
- Brief description of basic modules and/or composite modules contained
  - Description for a module with hardware aspects
    - Manufacturer's name and contact details, Country of manufacturer
    - Module type and version number
    - Inter-connectivity features included in the module (direction of connectors, pin assignments, etc.)
    - Serial number, if necessary
    - Rated values for supplies (e.g., electricity voltage supply or rated range in volts (DC/AC), rated frequency if necessary, pneumatic pressure, etc.)
    - Rated power in watts or rated current in amperes
    - Type of communication if used
    - Safety Certification Markings, if appropriate
    - Security features, if appropriate
    - Mass (in kg) and 3D dimensions (in mm)
  - Description for a module with software aspects
    - Manufacturer's name and contact details, Country of manufacturer
    - Software module type and version number
    - Operating system type and details
- Serial number, if necessary
- Checklist for usage of the module. e.g. type of modules suitable for interfacing, supported hardware modules (e.g. for appropriate mechanical/electrical interfacing) or compatible software modules
- Operating environments for modules
- Installation method for software modules, if any
- Details for connecting to other modules
- List of principles from Clause 4 followed by the module

b) Suitable use case applications, including their safety and security-related information, if any

c) Details for setting and adjusting module property values

d) List of replaceable basic modules and composite modules, if any

e) List of known faults or errors

f) Battery charging method, if relevant

g) Information to handle and transport module with details of grasping and handling points

h) List of expendables and their maintenance cycles

Safety-related information necessary for maintaining safe function when integrating modules should be provided, if appropriate, in a structured and well-defined format.

8.4 Information for service

The information for service should include instruction for maintaining correct operation of the module with details of tasks requiring specific technical knowledge or specialist skills, and hence need to be carried out by appropriate persons (e.g. maintenance staff, specialists, etc).

The information for service should include the following:

a) Detailed description of the module and its maintenance requirements

b) Information on the physical operational environment as appropriate (e.g. luminous intensity for vision module, contaminants in atmosphere, extreme in temperature, etc)

c) Information (as applicable) on:

- Instructions for setting up, maintenance schedule and nominal operational parameters
- Sequence of operation(s) to check for maintenance
- Frequency of inspections
- Frequency and method of functional testing of the modules
- Guidance on the adjustment, maintenance, and repair when needed
- Recommended spare parts list for modules with hardware aspects
- List of tools required and supplied

d) Detailed mechanical diagrams and electrical block diagram

e) List of known faults or errors and their description

f) List of expendables and their maintenance cycles

Annex A
(informative)

Robot module template

A.1 General template

Various modules are introduced within this document, and for the sake of uniformity, module descriptions should follow a common module template so that a normative format may evolve. Table A.1 shows the Robot Module Template which should be used by manufacturers to describe details of their modules. In Table A.1, italicized text indicates information which should be included in each part of the template. Manufacturers should use the template to describe details of their modules. Additional information may also be provided as appropriate.

Table A.1 — Description of the standard Robot Module Template

Module name: *A natural language name of a specific module or class of modules.*
Description: *Overview of module, what the module is, what it does and how it can be used in intended application scenarios: describe the application scenarios of the robotic module, so that validation tests can be performed, if necessary.*
Manufacturer: *Contact information for the developer(s) of the module. This can include details of the designer, the manufacturer, or the vendor organisations.*
Module ID: *Manufacturer's unique product reference number for module*
Examples: *Typical use case examples of the module*
Hardware aspects: *Summary details regarding hardware aspects, see Clause 6 (via examples if possible)*
Software aspects: *Summary details regarding software aspects, see Clause 7 (via examples if possible)*
Module properties: *List of module properties (see Clauses 6 and 7)*
Inputs: *List of module inputs*
Outputs: *List of module outputs*
Function/Functionality: *A description of the way that the module accepts inputs, and processes them to determine its outputs. The use of suitable diagrams (using for example the line, circle or SysML methods presented in Annex C) to illustrate the functionality is recommended.*
Infrastructure: *The type of infrastructure support and/or the environmental protection provided (e.g., power lines, database management system, data bus with or without safety/security, IP protection, etc)*
Safety: *Safety-related requirements for module level and system level safety (e.g., to meet required performance levels) (See Clauses 5.1–5.3)*

Table A.1 *(continued)*

Security: *Security requirements for module level and system level security (e.g. against unauthorized access, or to guarantee an appropriate level of privacy, etc.). This security requirements should include hardware and software perspectives. (See Clauses 5.1, 5.4–5.7)*
Modelling: *Mathematical or physical description of module applied to various test scenarios (e.g. virtual module model)*

A.2 Hardware-specific extensions to the robot module template

For the common template description, see Table A.1; Table A.2 shows additional information that should be provided for modules with hardware aspects.

Table A.2 — Additional information for modules with hardware aspects

Properties: *List of properties of modules with hardware aspects, such as physical size, interface type, mechanical and electrical characteristics*
Input: *List of inputs, such as digital/analogue sensor and command signals, as well as other communications between modules, etc.*
Output: *List of outputs, such as digital/analogue outputs, angle/position/speed/torque outputs, etc.*
Functionality: *For modules with hardware aspects, this relates mainly to interchangeability and interoperability. It is suggested that the module should be divided according to functional viewpoints such as its internal elements/structures, connectivity to external modules and to relevant features in the operational environment including people.*
Infrastructure: *Infrastructure requirements: The module's requirements from the rest of the system, like available power, structural support, heat dissipation, etc.* *Environmental constraints: The module's limits for external conditions such as temperature, humidity, maximum allowed mechanical shock, etc., both when turned off and when under operation.*
Modelling: *Mathematical or physical description of module dynamics applied to various purposes, such as performance simulation, functional evaluation and validation scenarios.*

Annex B
(informative)

Robot module examples

B.1 Examples of modules with hardware aspects

B.1.1 Actuated rotating joint

Module name: Actuated rotating joint
Description: The modular robot joint connects two consecutive links and offers one rotational degree of freedom motion. The modular joint consists of motor, reduction gear, power line, signal line, and control circuit. The joint can be driven by electrical power. The joint can perceive its rotational angle and the torque with internal sensors.
Manufacturer: ISO Inc.
Module ID: Joint J001
Examples: The joint can be used for moving a sensor or can be combined with other joints (e.g. 6 or 7 degrees of freedom) to form a manipulator.
Hardware aspects: — Connection Flange type 001B on both ends with connectors for power, CAN-bus, Safety-Torque-Off — Service port for direct USB access to the integrated electronics — IP-Rating: IP 54
Software aspects: Communication protocol: CANOpen

Module properties: — Size: Ø80 mm × 70 mm — Weight:1,2 kg — Reducer ratio: 1:30 — Joint range: ±270° — Maximum joint speed: 90°/s — Maximum torque: 100 Nm/20 Nm in forward/backward directions — Joint stiffness: max. 0,5 mm/1° shift at maximum load — Rated torque: 10 Nm — Current rating of connector: 10 A — Power consumption: 50 W — Accuracy: ±0,5° — Repeatability: ±0,3° — Limits [torque (Nm), position (rad), speed (rad/s)]
Inputs: — Position (rad), speed (rad/s), torque (Nm) command — Signals for safe functions — Control related parameters
Outputs: — Actual position (rad), speed (rad/s), torque (Nm) — Status, warnings, error, current, voltages, temperature, diagnostic information
Functionality: The joint can be used in position mode, velocity or force modes. It can be set to provide warnings to go into stop mode when limits are exceeded (no safety function). Internal configuration (CAN ID, limits, etc.) can be accessed via USB.
Infrastructure: — Power supply: 24 V dc (18 V–30 V), 50 W — Operational conditions: +5 ° to +35 °Celsius. Humidity <90 % non-condensing
Safety: Safe functions are provided according to IEC 61800-5-2. For protection of the module (no safety function), the module stops and goes to an error state in case of: overload (mechanical, electrical) failure, encoder sensor failure, overheating. Electrical power supply for service robot for warehouse logistics should be validated to check if it complies with the basic principles of integrability, interchangeability and safety, etc.
Security: Flange 001B and the cover of the USB port require standard tools for access.
Modelling: See model files for kinematic and dynamic models. The static model parameters include holding, rated, and stalling torques; the dynamic model parameters include speed, acceleration, and bandwidth.

B.1.2 Power supply

Module name: Power supply battery module
Description: Battery module with power management system, providing 24 V DC output.
Manufacturer: ISO Inc.
Module ID: Power supply P001
Examples: The power supply can be used in a mobile robot platform or on an exoskeleton
Hardware aspects: Power connector (2 pin) Data I/O-connector (4 pin) IP-Rating: IP65
Software aspects: Communication protocol: RS232, Battery management SW including alarm
Module properties: — Rated specifications: 24 V, 5 A continuous, 20 A max. — Capacity: 5 Ah — Power output: 25 V (full), 21 V (power management switches off) — Charging: 28 V to 35 V, up to 5 A current intake
Inputs: Charging power input Battery on/off
Outputs: Power output Battery error
Functionality: Battery needs to be switched on via digital input to provide power. Digital output signals for low power warning and errors
Infrastructure: — Operational conditions: +5 ° to +35 °Celsius. Humidity <90 % non-condensing
Safety: For protection of the module (no safety function), the module stops and goes to an error state in case of: overload, overheating, low power, deeply discharged
Security: N/A
Modelling: Please visit the website (with URL link provided) to download the behaviour model for use case scenarios.

B.2 Examples of modules with software aspects

B.2.1 Recognition

Module name: Vision recognition module

General description:

This module can be used for face recognition. Often building a database is included within an enhanced face recognition module. In a dynamic module, hardware, such as cameras and, 3D scanners are used for providing an on-the-fly data stream. The result of the module varies, such as a matched ratio between the given goal data and the registered data in the database, or ID numbers or the name of the best item matched from the database.

Manufacturer:

ISO Inc.

Module ID:

VRM0001

Examples:

Face recognition

Hardware aspects:

None

Software aspects:

Get data (input image), face recognition, put the result (name of recognized face)

Module properties:

— The database location, e.g. path, IP and Port number, or a URL

— Types of used recognition categories, e.g. eye, frontal face, full body, upper-body, etc

— Image size (pixels)

— Number of image frames per second (if input is a kind of moving image)

Inputs:

Image or image stream

Outputs:

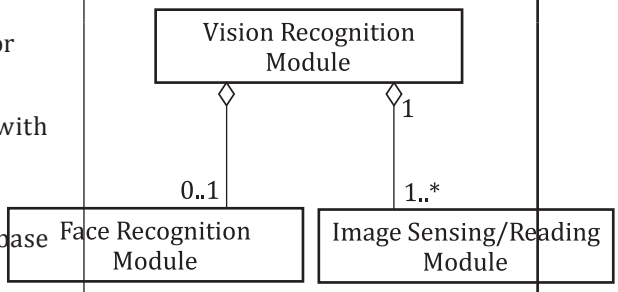
Result of the visual recognition with specified confidence (or accuracy), e.g., name of recognised person such as James, Eve, Adam, etc. (in face recognition)

Function/Functionality:

— Get image (or image stream) data from the camera module for human recognition

— Detect a face from the image data; Take a picture of the face with its identification number; Extract feature points of the faces; Compute the distances between them

— Find the face photo (or feature points of faces) from the database having the closest match to this computed value; Return the ID number of the selected face photo



Infrastructure:

Middleware, databases

Safety:

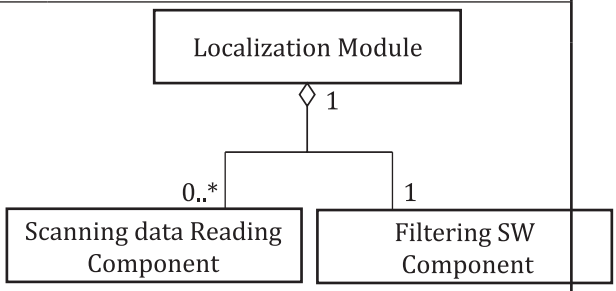
Not applicable to intended use case scenarios

Security:

Authentication, confidentiality of database for privacy

Modelling: Not applicable

B.2.2 Localisation

Module name: Localisation Module
Description: A personal robot has to know its own pose (position and orientation) within the reference coordinate system, a process, which is called localisation. The localisation module uses a laser scanning module in order to get the pose.
Manufacturer: ISO Inc.
Module ID: ID provided by manufacturer
Examples: Laser scan-based localisation
Hardware aspects: None
Software aspects: Get data (input image), computing and comparing with references such as land marks, put the pose Filtering software
Properties: — The number and types of sensing modules used in the module (e.g. angle and number of beams of the laser scanner, etc.) — Location for landmarks or map information or hazardous zone information, e.g. path, IP and Port number, or a URL
Inputs: Scanning data (from laser scanning module) Motion data from a wheel control module such as distance moved and orientation
Outputs: Pose of the robot with confidence (or accuracy)
Function/Functionality: — Get data from laser scanning module — Get data from wheel control module — Get the estimator of pose using data and filter — Compare the estimator with the reference poses — Update the pose 
Infrastructure: Middleware
Safety: Warning or stop according to applicable standard, if the robot enters into a hazardous zone
Security: Authentication
Modelling: Not applicable

B.3 Examples of commonly used composite modules

B.3.1 General

All service robots have high-level functionalities that can be identified. These functionalities include human-robot interfaces, navigation and localization, manipulation, travelling from one place to another, and ensuring safety according to applicable safety standards. Robot modules can be commonly used to composite modules implementing such typical high-level functions. This Clause presents modules at a higher level not previously discussed but considered important to realise a wide variety of service robot applications.

Composite modules are a combination of modules containing mechanical, electronics, and software parts. Such modules generally have a more complex nature like for example a modular manipulator with multiple degrees of freedom and integrated controllers, actuators, sensors, control software, safety functions, etc.

For any composite module, the minimal functions presented in the template should be specified in the module's property profile and in the module's input and output definitions. Inputs and outputs are generally data communicated to/from the module. The template for every commonly used module should provide a brief overview and minimal specification. Every manufacturer of such modules may add more functions as needed.

B.3.2 Manipulator module

Manipulation is a complex motion that can involve different levels of modularity such as individual joint control, coordinating manipulation with travel and use of different end effectors.

Module Name: Manipulator module
Description: An assembly of rigid link segments connected via joints forming an articulated system for manipulation of the end effector with all parts connected with defined mechanical interfaces. If it is the intention of the manufacturer to use a module in a collaborative application, then additional safety-related functions may be required in the device, for example to deal with elderly people or in a professional environment.
Manufacturer: Contact information
Module ID: Manufacturers unique product reference numbers for this module configuration
Example: A 6 degree of freedom robot manipulator to which a 2-finger end effector can be attached. Modular design enables users to reconfigure a manipulator to have 4–7 degrees of freedom to meet specific requirements. The manipulator in this example has an ultrasonic sensor capable of detecting objects at a distance of 20 cm or less.
Hardware Aspects: — Mounting, casing, motors, mechanical interface
Software aspects: — Kinematic module — Communication protocol implementation — Arm control module — Joint control coordination module
Module properties: — Degree of freedom: types of joints (2D, 3D, rotary/prismatic), manipulator configuration — Joint ranges: motion range, motion tolerances — Length and location (or type) of link module that links the joints — Payload at a range of poses: Allowable weight (kg) or force (N) at end effecter under static and dynamic conditions — Operational range of the arm (m × m × m) relative to the arm reference — Maximal speed (m/s) and acceleration (m/s^2) at end effector (may depend on pose)
Inputs: The manufacturer should define an enumerated list of commands for example: — Command operational position for pose, velocities — Move to spatial pose specified in quaternion — Force/speed limiting at end effecter

Outputs:

Actual spatial pose defined in x, y, z (meters) and orientation in quaternions

— Actual spatial velocity of the end effecter

— Actual and projected spatial envelope

— Actual joint velocities, acceleration and force (torque) of individual (m/s, rad/s)

— Operational status, warnings, errors, actual currents, voltages, temperatures

Function/Functionality:

— Forward kinematics, inverse kinematics, motion planning, dynamics

— Start/stop motion (enable, disable)

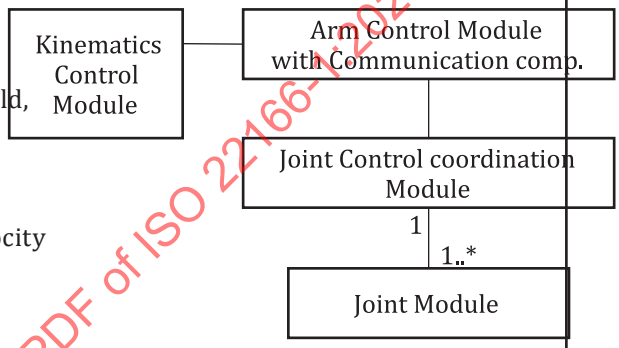
— Overload detection, status detection (OK/error), brake/hold, enable/disable functions

— Provide, stop, homing position interfaces

— Provide set/get force/torque, set/get position, set/get velocity interfaces based on an abstract interface

— Send force/torque values for all joints periodically to the joints, if required

— Motion and envelope forecasting from trajectory generator for improved performance



Infrastructure:

— Link/joint frames to provide mechanical support

— Quicklock to attach a gripper

— Power supply

— Communications databus

— Local and/or distributed controllers of the manipulator

Safety: The module follows applicable safety standards, as stated in Clause 5 (for example IEC 61508-3 or IEC 60204-1).

The protective stop function of the module complies with PL d according to ISO 13849-1.

Module safety: The module provides the following safety functions

— Collision force limiting having PL b (sensitive skin)

— Overload limiting

— Speed limiting control according to ISO 10218 with respect to speed control

System safety: The module provides the following safety-related information:

— Module Status

— Motion and envelope forecasting from trajectory generator for reducing collision risk (PL a)

— Stopping distance to be calculated in 3D space at rated speeds

— End effector speed to be set

— Specification of internal errors likely to cause major breakdowns or performance degradation

Security: The module can provide one or more of the following security functions — All module-module communications following the guidelines presented in Clause 7 — All inputs using error detection mechanisms — Accepting goal inputs only from authorized providers — Motion command information for use including authorization
Modelling: — Virtual static and dynamic model of manipulator including end effectors, joint dynamics and envelope

B.3.3 Mobile platform module

Mobility is a complex motion that can involve different level of modularity such as different locomotion configurations, different traveling behaviours, and coordinating manipulation with travel.

Module Name: Mobile platform module
General description: A locomotion module includes: — The motion system can contain a suspension system, steering system, drive mechanism system — Payload compartment — Locomotion methods: traditional wheels, omni-wheels, ball wheels, a variety of legs and legged configurations, hybrid locomotion methods, climbing, crawling, swimming, etc.
Manufacturer: Contact information
Module ID: Manufacturer's unique product reference numbers for this module configuration
Examples: Wheeled mobile base with emergency buttons, laser rangefinder and bumpers
Hardware aspects: — Actuator module — Bumper module — Battery module — Mobile base control hardware — Laser rangefinder module — Structural part module — Hardware aspects of communication module
Software aspects: — Mobile base control software — Position sensing software — Communication software — Battery management module — Actuator control module — Touch sensing software — Coordination module — Safety manager

Module properties:

— Mechanical configuration: Type/number of wheels, wheel arrangement and configuration and overall dimensions

— Payload: limits of load that can be carried within specified environmental conditions (e.g. mass, dimensions, temperature, etc.)

— Travel speeds: maximum speeds in foreseen operational scenarios, straight, turning, flat/slopes, unloaded, fully loaded

— Weight, centre of gravity (COG) in the unloaded condition

— Maximum slope angle and maximum step height in the unloaded and fully loaded conditions

— Battery duration and recharge time for full service

Inputs:

The manufacturer should provide an enumerated list for commands to be used for a mobile module:

— Motion speed and direction

— Control related parameters: surface conditions, obstacles, and environment

— Obstacle detection (digital and/or analogue)

— Protective and/or emergency stop

Outputs:

— Actual spatial pose defined in x, y, z (metres) and orientation in quaternions

— Motor rotational information: direction, angular

— Acceleration, motor torque/currents

— Status (OK/error), warnings, current, voltages, temperature

— Errors, safety-related operational conditions

— Safety performance level (PL level depending on the suggested use case application)

— Non-safe ultrasonic short-range obstacle detection

— Status of obstacle detection

— Status of pinching detection

— Status of protective and/or emergency stop

Function/Functionality:

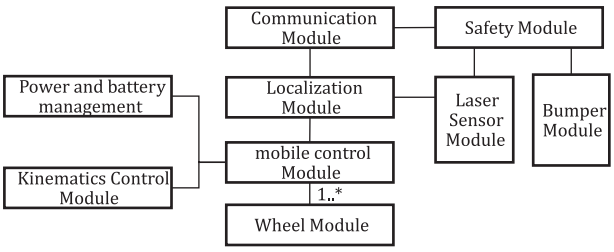
— Local motion control and kinematics

— Enable/disable functions

— Emergency braking

— Status check of internal modules

— Power- and battery management



Infrastructure:

— Chassis frame to provide mechanical support

— Power supply rail

— Communication

Safety: The module follows applicable safety standards, as stated in Clause 5 (for example IEC 61508-3 or IEC 60204-1). — Selectable stopping distances at specific velocities for configured system — The robot provides integrated safety circuits to connect safety-related sensors and modules. — PL a to e for each safety function provided by the module — Performance levels provided by the module: Emergency stop (PL d), Protective stop input (PL d)
Security: All module to module communications should follow guidelines presented in Clause 5 — Error detection mechanism to ensure communication data integrity — Goal locations are only accepted from authorized providers/modules — Motion command has to include authorization information for use
Modelling: Virtual static and dynamic model of mobile platform

B.3.4 Human robot interaction module

Human robot interaction (HRI) modules provide a means for humans to interact with the robot, become aware of robot's intentions, and provide commands or information to the robot.

Module Name: Human robot interaction module
Description: The HRI module can have the following functions: — Detection/Identification of a person — Interaction with the person (user) via speech, sound, light, touch screens
Manufacturer: Contact information
Module ID: Manufacturer's unique product reference numbers for this module configuration
Examples: Only spoken messages and information is sent to a user who is first recognized and identified by the camera. — HRI module can have sub-modules including a face recognition module, a speaker module — A coordination software module is used to manage the sequence of the sub-modules' interfaces or data — The TTS module translates text to speech — Light to indicate status and intended motion
Hardware aspects: — Speaker module — Touch screen — Light
Software aspects: — Coordination software module — TTS software module — Touch screen interaction software — Face recognition/identification module

Module properties:

— TTS software module, message format to play on speaker

— Face recognition module, database format

— API for touch screen

— Status and error information

— Operating conditions such as environment temperature and humidity range

Inputs:

— Message to play in the speaker

— User inputs from touch screen

Outputs:

— Result of person detection/identification

— Status (connected to database and to recognition server)

— Errors (system or person detection)

— Pass-through of inputs from user on touch screen

Function/Functionality:

Face recognition module, via the recognition module:

— Operational mode

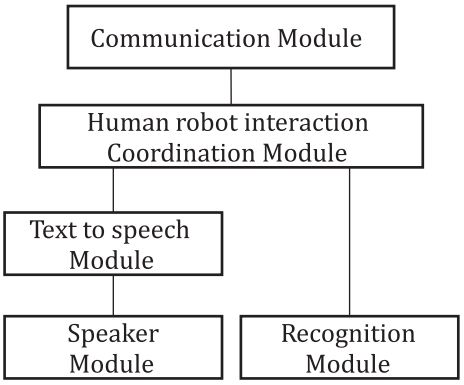
— Speech software module, convert text into speech via the speaker

— Coordination software module, manages the sequence the interfaces and data.

— Process user interaction via touchscreen

— Gesture identification module

— Voice command identification module



Infrastructure:

Middleware, external recognition server, database with pictures and messages

Safety:

The module follows applicable safety standards, as stated in Clause 5 (for example IEC 61508-3 or IEC 60204-1).

— Assessment of the development of the module for functional safety (IEC 61508 series)

— Alert messages for the user (according to relevant ISO standards)

— Human factors and usability

Security:

— Module only accessible via secured data

— Means to prevent abuse of the face recognition by ensuring a minimum confidence level

— Access to database and recognition server via secure connection

Modelling: Not applicable

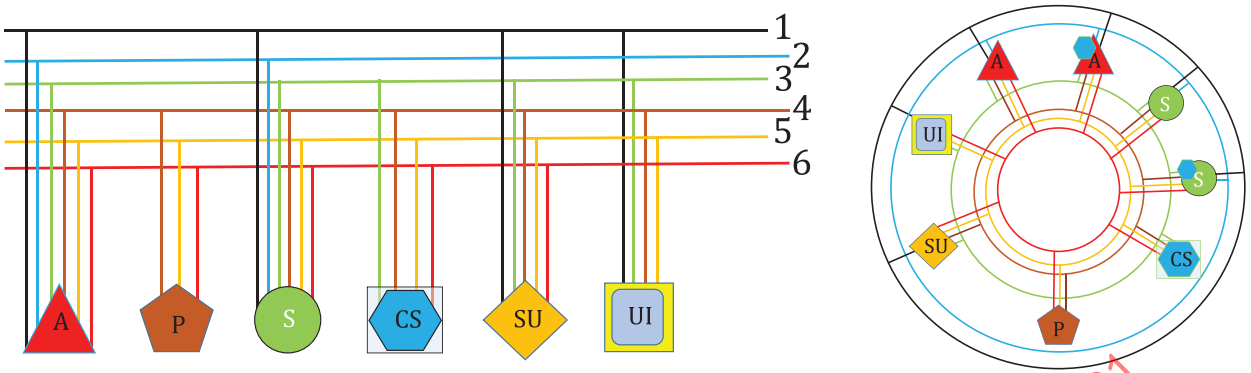
Annex C
(informative)

Use case examples of modularity for service robots

C.1 General

In the following sections, typical examples for the modular design of service robots are presented, which adopt the concepts and guidelines presented in this document; these cover hardware design, software design as well as the safety and security aspects presented in Clauses 5 to 7. In Clause C.2, a basic mobile robot system which adopted modular design is presented and the concepts of modularity are used to include advanced functionalities such as mobile manipulation to provide a variety of service features. In addition, a physical assistant robot in personal care application is presented in C.3.

The connectivity issues when configuring modules with hardware aspects can be presented diagrammatically. Two example methods, namely the line and circle methods based on Virk[36], and, Norman[37] respectively, are shown in Figure C.1 and can be used to illustrate connectivity between modules in the design of a service robot. Here a set of commonly used module icons are defined and connected to form application specific designs.



a) Line method **b) Circle method**

Key

1	environment	4	power
2	mechanics	5	security
3	data	6	safety

sensor (S)

computing module with software (CS)

sensor with processing

power supply (P)

software (SW)

supervisor (SU)

actuator (A)

user interface (UI)

actuator with processing

Figure C.1 — Connectivity diagrams for robot modularity and example modules

The line method presents the defined module icons whose connectivity to other modules is illustrated as conventional databus line diagrams via the specified interaction variables; these comprise safety, security, power, data communications (represented in various ways such as a specific digital databus, or as simple analogue or digital signal lines), mechanical interfaces and appropriate protections for the operational environment (e.g., water, dust, vibrations, etc.). The circle method also presents modular connectivity details via a circular format. Both methods are interchangeable and allow specific functionalities to be designed and presented by connecting individual blocks via interfaces for the relevant interoperability requirements.

Many modules can comprise intelligent features in which case they should require a range of data connectivity functionalities for the interoperability requirements; For example, the power supply module can have intelligent power management features and so data connectivity is likely to be needed; this can be achieved via a range of protocols (e.g. CAN, I2C, TCP/IP, USB).

There are other methods and approaches to represent the modularity aspects within a system depending on application (e.g. SysML).

C.2 Modularity for mobile robot systems

An example of a service robot with modular design is a delivery robot based on a mobile platform that can operate in crowded environments to hand objects to humans and is composed of a mobile platform and various sensors used to identify objects. Its main behaviours using Brook's approach[38], are the following:

— Move to a specific desired location

— Identify objects and avoid potential hazards during travel

Figure C.2 a) shows a configuration of modules with hardware and software aspects for a delivery service robot that can navigate with an obstacle avoidance behaviour in crowded facilities while ensuring security of data being accessed by unauthorised persons via encryption. Figure C.2 b) shows a visual example. This use case scenario should foresee safety considerations, security considerations and safety-related security considerations. To meet the safety requirement, the modules utilized should allow the needed safety requirements to be satisfied for the resulting delivery service robot in the application sector. The delivery service robot has different types of modules with hardware aspects, comprising wheel modules, actuator modules, a LIDAR module, a 2D image camera module, an infrared camera module, a computing module, and a power supply module. The four actuated wheels are installed on a mobile platform and controlled by a computing module for performing the desired traveling motions. Note that modules should follow the procedures suggested in Clause 5 where the safety, security and safety-related security risk assessment corresponding to the specific application of the delivery robot. The hardware aspect-related information (or properties) should be provided to ensure proper operation of the corresponding software modules. In particular, mechanical modules for delivery of packages should be well-organized to allow easy re-configuration. Static objects present in the operational environment should be identified for carrying out appropriate behaviours such as turning toward the goal or obstacle avoidance. Dynamic safety-related objects (such as humans) should involve more stringent safety requirements.

Figure C.2 c) illustrates a configuration of software modules that can achieve the desired behaviours using modules with hardware aspects, shown in Figure C.2 a). Note for convenience, the mapping between hardware and software aspects of modules is not focused upon here. Rather the overall functionality of the various software modules needed within use case scenario are highlighted in Figure C.2 c). Software modules for a delivery service robot can be largely classified as follows:

1) an identification module,

2) one or more data exchanging modules,

3) a security module,

4) a navigation module,

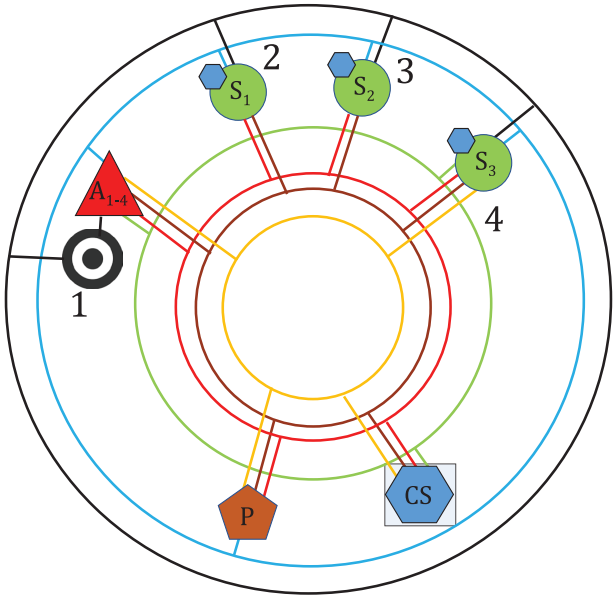
5) an obstacle avoidance module,

6) a mobility control module, and

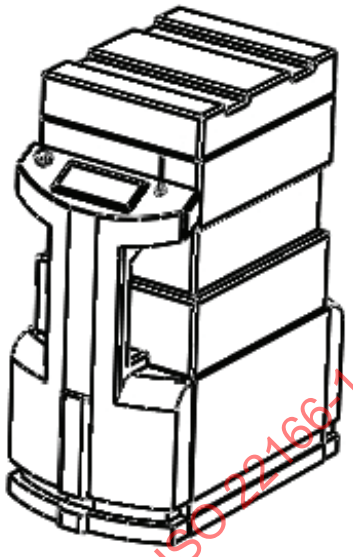
7) a safety module.

The identification module can consist of a personal identification module for face recognition of authorised individuals and an object identification module for recognising safety-related objects. The data exchange module is used to exchange data among modules within the delivery robot, servers, and other robots as appropriate. Commands such as movement to a target location and identification of a specific person are received via the data exchange module. Hence commands should be encrypted/protected and have an authority to be delivered and read by modules having the correct authority. If the decryption fails or the authority to the given command is incorrect, the security module should raise alerts, monitor progress of the operations, and perform suitable security measures. If the occurred security situation can lead to safety issues, the module should notify the safety module to ensure appropriate safety measures can be implemented. The navigation module consists of a mapping module, a localization module, and a path planning module. The navigation module sends the next

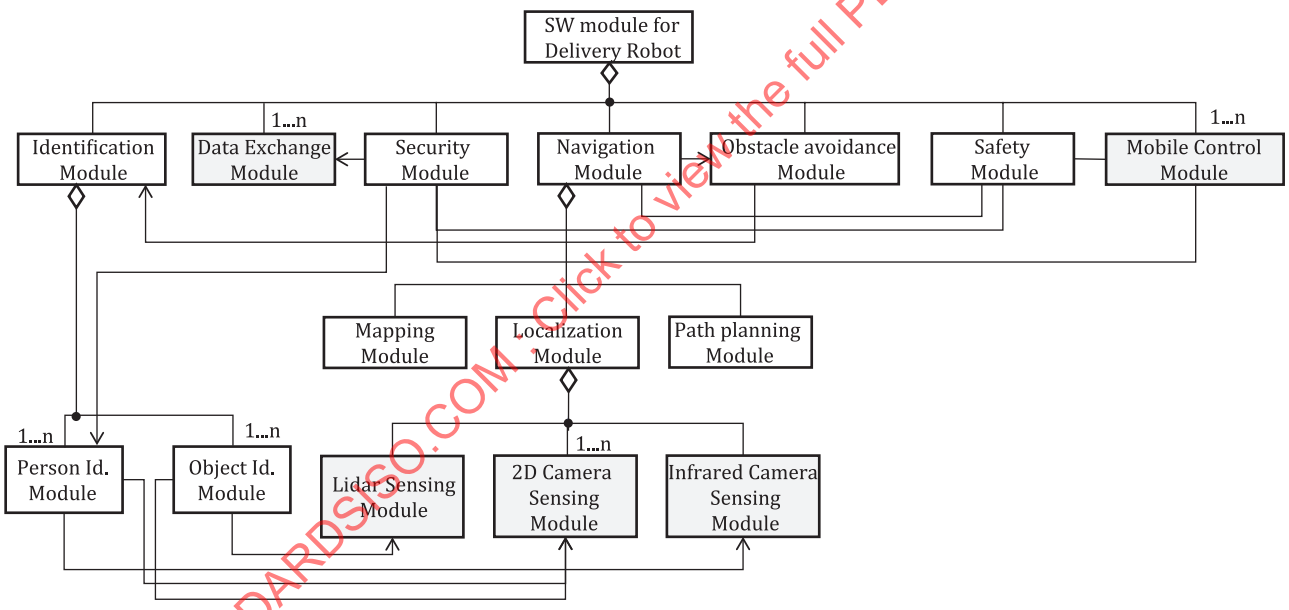
waypoint to the mobility control module. In addition, the navigation module checks whether or not the robot is operating in a hazardous area and then should send the alarm notifications to the safety module if the robot entered into a hazardous situation. The obstacle avoidance module should provide the navigation module with information about obstacles for the robot to avoid. Of course, the obstacle avoidance module can be included in the navigation module. The safety module should manage the safety related hazards for the robot including those related to security as determined by the safety and security risk considerations. The safety module should collect and analyse safety-related data from the identification module, the data exchange modules, the security module, the navigation module, and the mobility control module. A mobility control module includes the software module for controlling the four actuators ($A_{1\text{-}4}$) in the wheel modules. According to the analysed results, appropriate security, safety and security-related safety measures should be considered and performed. The localization module generates the current pose of the robot using sensing modules, which are chosen here to comprise a LIDAR sensing module, a 2D camera sensing module, and an Infrared camera sensing module to obtain the needed sensing information using appropriate software modules. Note that the property files of the software modules used for the delivery service robot should be provided to ensure operation as planned. The shading in boxes shown in Figure C.2 c) represents software modules communicating with modules having hardware aspects.



a) Modules with hardware aspects



b) Example of image of delivery robot



c) Software modules

Key

1 wheel
2 infrared camera
3 LIDAR sensor
4 2D camera

Figure C.2 — Example of design of delivery robot with a mobile platform

To expand the mobile delivery robot platform to carry out mobile manipulator behaviours, it is possible to include pick-and-place operations to the mobility operations so that enhanced robot can take objects