P11073-10207™/D6
Draft Standard for Domain Information & Service Model for Service-Oriented Point-of-Care Medical Device Communication

Sponsor

IEEE 11073 Standards Committee
of the
IEEE Engineering in Medicine and Biology Society

Approved <Date Approved>

IEEE-SA Standards Board

445 Hoes Lane
Piscataway, NJ 08854, USA

Abstract: <Select this text and type or paste Abstract—contents of the Scope may be used>

Keywords: <Select this text and type or paste keywords>

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-XXXX-XXXX-X STDXXXXX
Print: ISBN 978-0-XXXX-XXXX-X STDPDXXXXX

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual,

non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this draft standard was completed, the Point of Care Devices Working Group had the following membership:

Jan Wittenber, *Chair*
Paul Schluter and Todd Cooper, *Vice Chair*

Participant1	Participant4	Participant7
Participant2	Participant5	Participant8
Participant3	Participant6	Participant9

The following members of the <individual/entity> balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

[To be supplied by IEEE]

Balloter1	Balloter4	Balloter7
Balloter2	Balloter5	Balloter8
Balloter3	Balloter6	Balloter9

When the IEEE-SA Standards Board approved this standard on <Date Approved>, it had the following membership:

[To be supplied by IEEE]

<Name>, *Chair*
<Name>, *Vice Chair*
<Name>, *Past Chair*
Konstantinos Karachalios, *Secretary*

SBMember1	SBMember4	SBMember7
SBMember2	SBMember5	SBMember8
SBMember3	SBMember6	SBMember9

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

<Name>, *DOE Representative*
<Name>, *NIST Representative*

<Name>
IEEE-SA Content Production and Management
<Name>
IEEE-SA Technical Program Operations

Introduction

This introduction is not part of P11073-10207/D6, Draft Standard for Domain Information & Service Model for Service-Oriented Point-of-Care Medical Device Communication.

TODO IMAGE

This specification comprises the following parts:

— BICEPS Model

A participant information and service-oriented communication model. The participant information model is defined in this specification. It is logically stored in the Medical Device Information Base (MDIB) of the SERVICE PROVIDER. The messages and SERVICEs to interact with the MDIB offered by the PARTICIPANT are defined in the communication model and are also part of this specification.

— BICEPS Discovery

A Discovery model for PARTICIPANTs. In order to support plug-and-play in a service-oriented medical device system (SOMDS) this specification defines how SERVICEs may be discovered in the network by potentiell a SERVICE CONSUMER.

— BICEPS Non-Functional Requirements

A set of non-functional requirements that can be used to define the requirements of SERVICE PROVIDER w.r.t. patient safety, cybersecurity, clinical effectiveness as well as regulatory requirements.

NOTE—The non-normative name of this specification is BICEPS which stands for “Basic Integrated Clinical Environment Protocol Specification”.

1.1 BICEPS Model

The BICEPS model defines a set of communication messages and information elements for the above defined purpose. It should be noted that the BICEPS model does not define a communication message or element for every possible kind of measurement data, setting data, contextual information or remote invocation command.Instead it provides extensibility mechanism as part of the model that make it possible for a PARTICIPANT to convey additional data in a message or to transmit a completely new type of message.

The BICEPS model is closely related to the DIM as defined in IEEE 11073-10201, but it is not an exact copy of it. As the BICEPS model has been designed with the idea in mind to specify only the essential messages that are needed in a distributed system of medical devices in a POC ENVIRONMENT, only a subset of the packages of the IEEE 11073-10201 DIM have been used to be included in the BICEPS model. Those packages as well as those that have not been included into the BICEPS model are shown in Table 1.

The participant information model for PARTICIPANTs is specified in section 5.

The communication model for PARTICIPANTs is specified in section 6.

NOTE—It should be noted that most INFORMATION OBJECTs from the packages in scope have been revised and that even if a package is in scope of the BICEPS model not all objects of the IEEE 11073-10201 DIM package may be part of the BICEPS model.

Table 1— Packages of 11073-10201 DIM that are in-scope and out-of-scope of BICEPS

Packages in scope of BICEPS model	Packages out-of-scope of BICEPS model
Medical	Extended Services
Alert	Communication
Control	
System	
Patient	
Archival	

1.2 BICEPS Discovery

In order to support plug-and-play in a SOMDS, the SERVICEs of SERVICE PROVIDER have to be discovered by potential SERVICE CONSUMERs. BICEPS discovery defines two discovery modes that a SERVICE PROVIDER has to support:

- Explicit discovery
- Implicit discovery

Explicit discovery is based on search messages that are sent to a group of network nodes and that may contain matching criteria of what the SERVICE CONSUMER is interested in.

In contrast to this, implicit discovery is based on messages that are communicated by a SERVICE PROVIDER when it enters or leaves the network or if something in its context has changed substantially.

1.3 BICEPS Non-Functional Requirements

In order to ensure safety, security and effectivity of a SOMDS this specificastion defines non-functional requirements that have to be fulfilled by any transport protocol that should be used to transmit the service messages or discovery messages.

Contents

<After draft body is complete, select this text and click Insert Special->Add (Table of) Contents>

Draft Standard for Domain Information & Service Model for Service-Oriented Point-of-Care Medical Device Communication

2. Overview

2.1 Scope

The scope of this standard is the definition and structuring of information that is communicated in a distributed system of point-of-care medical (PoC) medical devices and medical IT systems that need to exchange data or safely control networked PoC medical devices by defining a participant information model and service model. The definition of transport serialization is outside the scope of this standard.

2.2 Purpose

The standard defines a participant capability and state information model that is derived from the 11073 Domain Information Model (IEEE 11073-10201) as well as a service-based discovery, communication and control model for participants in a service-oriented distributed PoC medical devices and medical IT system. The standard utilizes the 11073 Nomenclature (IEEE 11073-10101) and supports other coding systems for specifying the semantics of an information element.

3. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

[11073-10201]ISO/IEEE 11073 standard, Health informatics – Point-of-Care medical device communication, Part 10201, Domain information model

4. Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause. [1]

4.0. SOMDS: A Service-Oriented Medical Device System (SOMDS) is an instance of a distributed system that implements a service-oriented architecture where the service providers are PoC medical devicesdevices.

4.0.ALARM CONDITION: State of the ALARM SYSTEM when it has determined that a potential or actual HAZARDOUS SITUATIONhazardous situation exists for which OPERATOR operator awareness or response is required. See also [IEC 60601-1-8].

4.0.ALARM SIGNAL: Type of signal generated by the ALARM SYSTEM to indicate the presence (or occurrence) of an ALARM CONDITION. See also [IEC 60601-1-8].

4.0.ALARM SYSTEM: Part of a PoC Medical Device that detects Alert Conditions and - as appropriate - generates Alert Signals. See also [IEC 60601-1-8].

4.0.ALERT: Synonym for the combination of patient-related physiological alarms, technical alarms and equipment user advisory signals.

4.0.CHANNEL: Abstraction for a logical or physical grouping of Metrics that allows hierarchical information organization.

4.0.HAZARDOUS SITUATION: TODO, See also [IEC 60601-1-8]

4.0.INFORMATION OBJECT: An abstract data model applicable to the communication of vital signs information and related patient data. The attributes of an information object definition describe its properties. Each information object definition does not represent a specific instance of real-world data, but rather a class of data that share the same properties.

4.0.MEDICAL DEVICE SYSTEM (MDS) CONTAINMENT TREE: A device configuration and capability description of on MEDICAL DEVICE SYSTEM that represents a POC MEDICAL DEVICE.

4.0.MEDICAL DEVICE SYSTEM (MDS): Abstraction for system comprising one or more medical functions. In the context of this standard, the term is specifically used as an object-oriented abstraction of a PoC Medical Ddevice that provides medical information in the form of information objects that are defined in this standard.

[1] *IEEE Standards Dictionary Online* subscription is available at: http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

4.0.METRIC: Abstraction for a component of a PoC medical device that is able to generate or store direct and derived, quantitative and qualitative biosignal measurement, settings, status, and context data.

4.0.PARTICIPANT: Any network node that is part of a SOMDS and exchange information by means of a service-oriented architecture.

4.0.POC MEDICAL DEVICE: Medical devices that directly interact with, monitor, provide treatment to, or are in some way associated with a single patient. For the purposes of this standard, the scope of POC MEDICAL DEVICES DEVICEs is further limited to patient-connected medical devices that provide support for electronic communications.

4.0.POINT-OF-CARE (POC) ENVIRONMENT: Environment encompassing a particular diagnostic, bed or treatment area which is specific to one patient, and usually including those systems and personnel which are involved in the acute monitoring and treatment of the patient.

4.0.SERVICE CONSUMER: A participant that utilizes at least one SOMDS SERVICE.

4.0.SERVICE PROVIDER: A participant that provides at least one SOMDS SERVICE. A Service Provider can be hosted on the PoC Medical Device itself or on an external appliance (e.g., converter box) or IT system (e.g., aggregator).

4.0.SERVICE: A software system that exposes its capabilities by receiving and/or sending messages on one or several network endpoints. [DPWS 1.1].

4.0.SOMDS SERVICE: A SERVICE that is intended to be used by PARTICIPANTs for retrieving INFORMATION OBJECTs instances or for remote control of a POC MEDICAL DEVICE.

4.1 VIRTUAL MEDICAL DEVICE (VMD): Abstraction for a medical-related subsystem (e.g., hardware or even pure software) of a PoC medical device.

5. Participant Information Model

The scope of this section is the definition of a participant model that allows the description of the capabilities as well as the expression of the current state of a POC MEDICAL DEVICE in a SOMDS.

5.1 General

This specification makes use of the concept of a Medical Device Information Base (MDIB) as defined in IEEE 11073-10201 (section 6) [11073-10201] that represents a structured collection of instances of managed medical objects representing e.g. the vital signs information provided by a particular medical device.

Attributes and hierarchies of the objects in MDIB are defined in this standard. The majority of objects defined here represent generalized vital signs data and support information. Specializations of these objects are achieved by defining appropriate attributes. Object hierarchies and relations between objects are used to express device configuration and device capabilities.

All MDIB INFORMATION OBJECT instances are identifiably by a unique handle. The handle ensures an explicit identification. It has to be unique within one MDIB context.

R0007: ALL handles SHALL be unique within one MDIB of a SERVICE PROVIDER.

The MDIB is logical divided into two parts: The capability description (MDDescription) and the current state of the medical device (MDState). The normative description for both parts can be found in sections 5.2. and 5.3.

5.2 Capability Description

To describe the capabilities of a POC MEDICAL DEVICE a set of Descriptors is defined.

The normative outline of the Descriptor is defined in Annex B.

The Descriptor concept facilitates interoperable communication between PARTICIPANTs. A minimum set of descriptive elements and attributes is declared as mandatory to ensure interoperability and compliance of medical issues. In the specialized Descriptor information elements, some optional elements and attributes are included to describe the system and its capabilities in more detail, but be flexible if the information is not available on all POC MEDICAL DEVICEs. It should be noted that the more information is exposed in the Descriptors, the merrier an interoperable communication is possible.

A main idea of the Descriptor concept is the specialization of this generalized information objects via the CodedValue construct. A CodedValue object has a mandatory element CodeId. The CodeId is a unique identifier of a code in the coding system that be declared by using the CodingSystemId. If no CodingSystemId is declared the implied value references the ISO/IEC 11073-10101:2004 [XXX].

R0008: A SERVICE PROVIDER SHALL use standardized values for CodeId and CodingSystemId in order to specialize an information object if available.

Every Descriptor has an optional attribute IntendedUse. The IntendedUse attributes indicates the intent of the POC MEDICAL DEVICE what the data is allowed to be used for by a SERVICE CONSUMER. IntendedUse can be either be “medical” or “informational”, where “medical” implies that the data may be used by a SERVICE CONSUMER that has a medical intended use. The “medical” category furthermore possesses three subcategories that allow the differentiation of the potential harm to the patient that may result from using the information object. The implied value is “informational”.

R0009: If the IntendedUse of a descriptor is “INFORMATIONAL” the descriptor information and related state information SHALL NOT be used in medical-grade algorithms or applications.

5.2.1 Medical Device System Containment Tree

A MEDICAL DEVICE SYSTEM (MDS) CONTAINMENT TREE expresses one device configuration and hence a set of information provisioning and remote control capabilities of one POC MEDICAL DEVICE.

A MDS CONTAINMENT TREE of a medical device consists of at least one MEDICAL DEVICE SYSTEM (MDS). One derived instance of a MDS is the HydraMDS. A HydraMDS can consist of zero or more VIRTUAL MEDICAL DEVICEs (VMD). The VMDs encapsulate subsystem functionality of the HydraMDS. A VMD can offer zero or more CHANNELs. CHANNELs can consist of zero or more METRICs. A METRIC is a component of a medical device that is able to generate or store direct and derived, quantitative and qualitative biosignal measurement, settings, status, and context data. A CHANNEL is used to group associated METRICs.

NOTE—In contrast to IEEE 11073-10201 [11073-10201] there exists no dedicated CompositeMDS. A CompositeMDS can be modeled by adding multiple MDS to the MDIB and the compositional relationship between the MDS instances is maintained by the system context information of each MDS. Moreover, there

may be a derived MDS descriptor, e.g. SingleBedMDSDescriptor, in a new version of the model that allows the combination of multiple MDS under another MDS.

The normative outline of the AbstractDeviceComponent and its specializations AbstractMDSDescriptor, HydraMDSDescriptor, VMDDescriptor, ChannelDescriptor and AbstractMetricDescriptor are defined in Annex B.

The normative outline of the AbstractMetricDescriptor and its specializations are defined in Annex B.

NOTE—A HydraMDS represents a single- or multipurpose device. In contrast to IEEE 11073-10201 [11073-10201] a singlepurpose MDS is not distinguished from a multipurpose MDS. A HydraMDS may contain zero VMDs in case the capabilities of the submodules should not be exposed to the communication partner, e.g. due to authorization problems.

NOTE—In contrast to IEEE 11073-10201 [11073-10201] at least one CHANNEL is mandatory to group the METRICs of a VMD. It is not allowed to attach a METRIC directly under a VMD.

R0098: A SERVICE PROVIDER SHOULD use the same handle for the same instance of an element of the containment tree with every addition to its MDIB.

R0099: If a SERVICE PROVIDER is not capable of preserving the mapping between handles and elements of the containment tree for a reinsert into the MDIB, e.g. after a power cycle, it MAY reassign a new handle.

NOTE—This may lead to a broken history for that specific element if no other means like production specification can be used by a service cosnumer to identify the instance.

R0102: If a SERVICE CONSUMER detects that the MDIB has been reseted, e.g. after a power cycle, it SHALL not combine current information with prior information based on the handle.

NOTE—As stated in R0099 a SERVICE PROVIDER may have re-assigned the handle during the MDIB reset. Nevertheless, a SERVICE CONSUMER is allowed to combine the information with prior information based on other meta-data.

5.2.1.1 Alerts

If a POC MEDICAL DEVICE possesses an ALERT SYSTEM to detect alert conditions and generate alert signals, this capabilities may also be expressed as part of the MDS CONTAINMENT TREE. The AbstractDeviceComponents can be extended to express the alert handling capabilities by attaching an AlertSystem.

R0012: A SERVICE PROVIDER SHOULD express alerting capabilities using the AlertSystemDescriptor structure.

An AlertSystem consists of zero or more AlertConditions and zero or more AlertSignals that are related to the AlertConditions of the same AlertSystem.

An AlertCondition refers to the object (e.g. AbstractMetricDescriptor or AbstractDeviceComponent) it is monitoring in order to determine the condition that it should alert. Every AlertCondition should be specialized using a CodedValue for the type that describes the condition.

The way the alert is signaled is described by zero or more AlertSignals, where each AlertSignal references to exactly one AlertCondition.

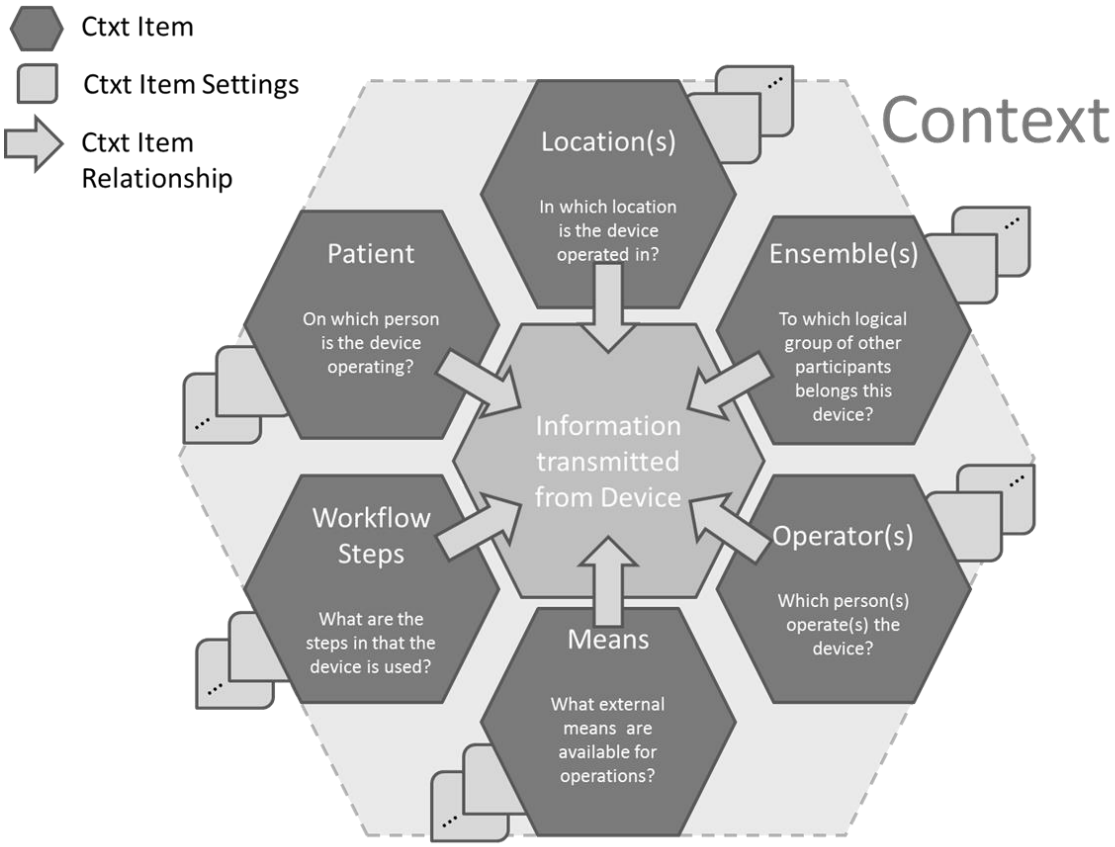
The normative outline of the AbstractAlertDescriptor and its specializations AlertSystem, AlertCondition, and AlertSignal are defined in Annex B.

5.2.1.2 Context Handling

Generated data as well as remote control commands that are intended to modify the state of a POC MEDICAL DEVICE often can only be interpreted by the receiver if this knows the context of generation of the data or the command. For this reason, information about the current context in which the POC MEDICAL DEVICE operates needs to be expressed.

The contextual information may range from just a set of identifiers that identify the context to additional contextual information like patient demographics.

Figure 1— Kinds of contextual information that may be handled by a POC MEDICAL DEVICE.



R0013: A SERVICE PROVIDER MUST NOT express a capability to handle a specific kind of contextual information if it is not able to validate the contextual information either directly or indirectly.

This specification defines six kinds of contextual information (cf. Figure 1) with dedicated descriptors that can be part of a SystemContext:

— PatientContextDescriptor

- LocationContextDescriptor
- WorkflowContextDescriptor
- OperatorContextDescriptor
- MeansContextDescriptor
- EnsembleContextDescriptor

These descriptors are a specialization of an AbstractContextDescriptor.

R0014: If a SERVICE PROVIDER or POC MEDICAL DEVICE is capable of determining to which patient the POC MEDICAL DEVICE is currently connected to, this capability SHOULD be expressed in the SystemContext with a PatientContextDescriptor.

R0015: If a SERVICE PROVIDER or POC MEDICAL DEVICE is capable of determining in which location the POC MEDICAL DEVICE is currently operated, this capability SHOULD be expressed in the SystemContext with a LocationContextDescriptor.

R0016: If a SERVICE PROVIDER or POC MEDICAL DEVICE is capable of determining in which clinical workflow the POC MEDICAL DEVICE is currently participating, this capability SHOULD be expressed in the SystemContext with a WorkflowContextDescriptor.

R0017: If a SERVICE PROVIDER or POC MEDICAL DEVICE is capable of determining who is currently operating the POC MEDICAL DEVICE, this capability SHOULD be expressed in the SystemContext with a OperatorContextDescriptor.

R0018: If a SERVICE PROVIDER or POC MEDICAL DEVICE is capable of determining which virtual or physical means the POC MEDICAL DEVICE is using, this capability SHOULD be expressed in the SystemContext with a MeansContextDescriptor.

An example for a virtual means is a license from a pool of licenses. An example for a physical means could be a docking station that the POC MEDICAL DEVICE is connected to.

R0019: If a SERVICE PROVIDER or PoC Medical Device is e.g. capable of determining in which logical group the PoC Medical Device is currently operated, this capability SHOULD be expressed in the SystemContext with a EnsembleContextDescriptor.

A logical group may for example be a temporal group based on a treatment session or spatial group that is not affixed to one location, e.g. a set of medical devices that accompanies a patient on a transport.

As not all POC MEDICAL DEVICEs are able to handle all kinds of contextual information, an instance of a AbstractMDSDescriptor is able to express which kinds of contextual information can be handled by the POC MEDICAL DEVICE. Handling contextual information means that the POC MEDICAL DEVICE or the SERVICE PROVIDER is capable of validating that the POC MEDICAL DEVICE is actually operated in the context that is expressed by the contextual information. If the SERVICE PROVIDER is capable of validating the contextual information without the POC MEDICAL DEVICE that this contextual information is related to, this is called direct validation. If the SERVICE PROVIDER utilizes the POC MEDICAL DEVICE for validating contextual information for that medical device, this is called indirect validation.

Direct or indirect validation of contextual information may afford operator interaction or may rely solely on information provided by the act of proposing the contextual information.

5.2.1.3 Remote Control

The Service Control Object (SCO) is the representation of a service controller that comprises offered remote invocation capabilities and on which objects these could be executed. The SCO is an optional component of a HydraMDS.

R0020: A SERVICE PROVIDER SHOULD describe all offered remote invocation capabilities using the SCODescriptor structure.

Every remote invocation capability is described using operation descriptors. The SCO can contain several operation descriptors. These operation descriptors are extensions of AbstractOperationDescriptor. An operation has a mandatory OperationTarget that is a handle reference to one object of MDDescription the operation is targeted to.

The normative outlines of the SCO as well as the OperationDescriptors are defined in Annex B.

5.2.1.4 Localization Support

TODO

dom:LocalizedText

Lang, Ref, Version, Text Content

The normative outlines are defined in Annex B. An informative example for a MDS CONTAINMENT TREE can be found in Annex F.

5.2.2 MDDescription

A set of MDS CONTAINMENT TREEs (see section 5.2.1) that are contained in the MDIB of a SERVICE PROVIDER describe which kind of INFORMATION OBJECT instances of which POC MEDICAL DEVICEs are currently provided by or can be remotely modified through that SERVICE PROVIDER. These MDS CONTAINMENT TREEs are contained in the medical device description MDDescription information element.

R0010: A SERVICE PROVIDER SHALL describe the information provisioning and remote control capabilities of its managed POC MEDICAL DEVICEs using the MDDescription structure.

The normative outline of the MDDescription is defined in Annex B.

NOTE—It should be noted, that the set of MDS Containment Trees in a MDDescription may also be empty at a certain point in time. This may be the case, if for example the Service Provider acts as an aggregator of PoC Medical Devices and none of the devices that it manages are available at that point in time.

Besides a set of MDS CONTAINMENT TREEs, a MDDescription of a SERVICE PROVIDER may contain capability descriptions of POC MEDICAL DEVICEs that do not match the definition of an MDS. Moreover, the MDS CONTAINMEMT TREE MAY contain capability descriptions that are not part of this specification.

R0094: A SERVICE PROVIDER MAY include a capability description of a POC MEDICAL DEVICE in the MDDescription structure using the extension points defined in the MDDescription structure or any of its substructures.

R0011: A SERVICE PROVIDER MUST NOT describe the capabilities of a POC MEDICAL DEVICE using extension elements and attributes that are not part of this standard, if elements or attributes of this standard apply.

5.3 State Expression

The current state of a POC MEDICAL DEVICE at a given point of time that is related to a capability described in the MDS CONTAINMENT TREE and hence intended to be retrieved or modified by a SERVICE CONSUMER is expressed as a set of INFORMATION OBJECT instances in the dynamic partition of MDIB.

R0021: A SERVICE PROVIDER SHALL describe the state of a managed POC MEDICAL DEVICE at a given point in time using the MDState structure.

The normative outline of the MDState is defined in Annex B.

R0022: A SERVICE PROVIDER MAY extend its state expression based on the MDState structure using the extension points defined in the MDState structure or any of its substructures.

5.3.1 Mapping between Descriptor and State

For every descriptor in the MDDescription there SHOULD be at least one corresponding state if information about the state is available. Every state has a mandatory element DescriptorHandle that is a handle reference to the Descriptor the state INFORMATION OBJECT instance is representing the current value for.

R0023: A SERVICE PROVIDER SHOULD have at least one corresponding instance of a specialization of an AbstractState in the MDState partition of the MDIB for each Descriptor that is part of the MDDescription partition of the MDIB.

R0024: Every instance of a specialization of an AbstractState SHALL have a DescriptorHandle that references an existing and corresponding Descriptor in the MDDescription partition of the same MDIB.

The mapping which type of AbstractState is corresponding to which type of Descriptor is given in Table 2.

Table 2— Mapping between Descriptor and State data types. * means that there are multiple derived types possible that are not all listed here.

Descriptor	AbstractState
*MDSDescriptor	ComponentState
VMDDescriptor	ComponentState
ChannelDescriptor	ComponentState
ClockDescriptor	ClockState
*MetricDescriptor	*MetricState
SCODescriptor	-
*OperationDescriptor	*OperationState
AlertSystemDescriptor	AlertSystemState
*AlertConditionDescriptor	*AlertConditionState
AlertSignalDescriptor	AlertSignalState
SystemContext	-
*ContextDescriptor	*ContextState

The semantics of the optional StateHandle to uniquely identify a state is defined as following.

R0096: A State SHALL not own a handle attribute if it is and will be the only state belonging to a descriptor. This is named a single state. Single states are uniquely identifiable by the corresponding descriptor handle.

A state is a multi state if there are or will be multiple states referencing the same descriptor.

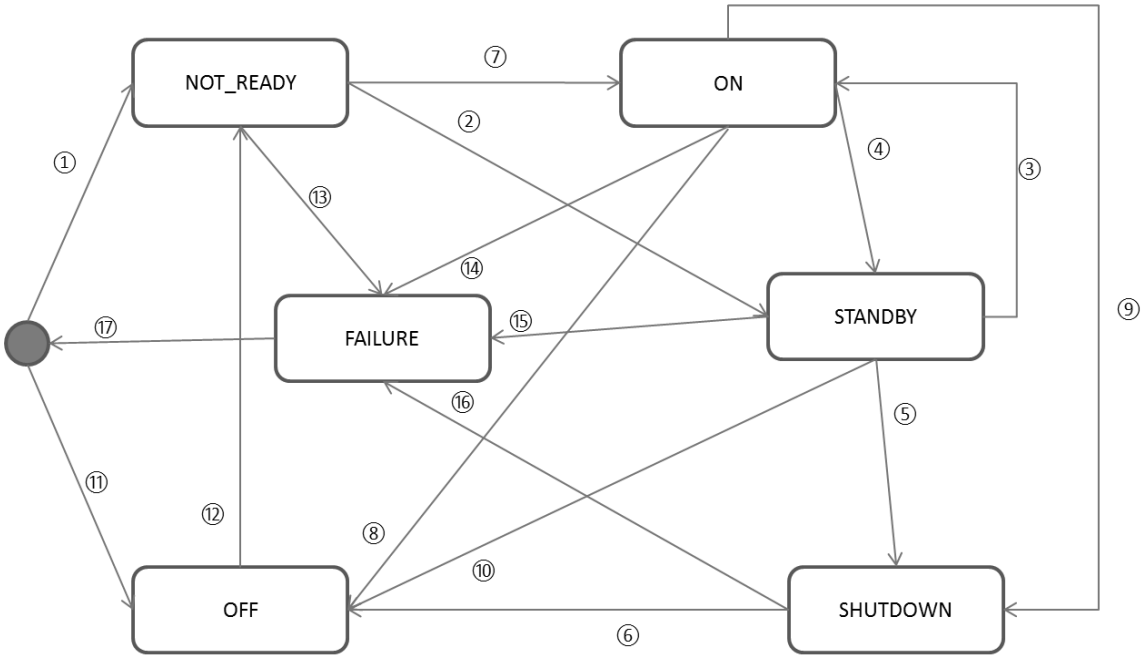
R0097: A state SHALL own a handle attribute if it is a multi state.

5.3.2 State Activation

Every specialization of an AbstractState has some form of indicator that expresses the current operating status of the component the AbstractState is representing.

The potential Activation states of an AbstractDeviceComponent are depicted in Figure 2.

Figure 2—Component activation states



The conditions for the transitions are: The component …

1) … is made available, but needs initialization.
2) … has completed initialization and it does not operate.
3) … has been activated and is operating.
4) … does not operate, but is available for operation.
5) … is intended to be unavailable for operation and needs a shutdown phase.
6) … has been made unavailable for operation.
7) … has completed initialization and is operating.
8) … has been made unavailable for operation.
9) … is intended to be unavailable for operation and needs a shutdown phase.
10) … has been made unavailable for operation.
11) … is intended to be unavailable for its intended use, e.g. turned off, not plugged in.
12) ... is made available, but needs initialization.
13) … has detected a failure during initialization.
14) … has detected a failure during operation.
15) … has detected a failure during standby.
16) … has detected a failure during shutdown.
17) … has been recovered after failure.

Even though the Descriptors, which a State references to, are represented in a form of an MDS CONTAINEMENT TREE the semantics of a state activation of a parent state in the MDS CONTAINMENT TREE is not forwarded down the tree structure to child states as the real semantics highly depends on the POC MEDICAL DEVICE that is represented.

R0025: A SERVICE CONSUMER SHALL interpret the activation information of each AbstractState independently even if the MDS CONTAINEMENT TREE indicates a parent-child relationship.

5.3.2.1 Remote Control

An OperationState has a mandatory element OperatingMode. This has to be used to enable and disable the accessibility of the referenced operation.

R0026: A SERVICE PROVIDER SHALL use the OperatingMode attribute to enable and disable the accessibility of the referenced operation.

R0027: A SERVICE PROVIDER SHALL ensure that a DISABLED operation is not accessible for remote invocation.

R0028: A SERVICE PROVIDER MAY enable a referenced operation even if the activation state of the referenced information object instance is indicating a non-operating mode.

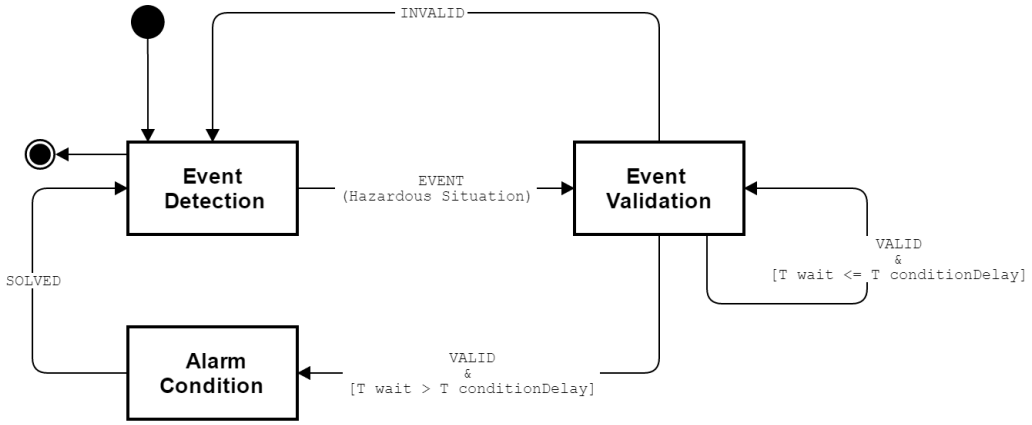
5.3.2.2 Alert Systems

AlertSystemState, AlertConditionState, and AlertSignalState have the mandatory attribute ActivationState that declares whether the element is operating or (temporarily) not.

R0029: A POC MEDICAL DEVICE SHALL flag an AlertCondition as present only if the ActivationState of the corresponding AlertSystemState is "ON" and the condition is present.

The state diagram for an AlertSystem and the relationship to AlertConditions is depicted in Figure 3.

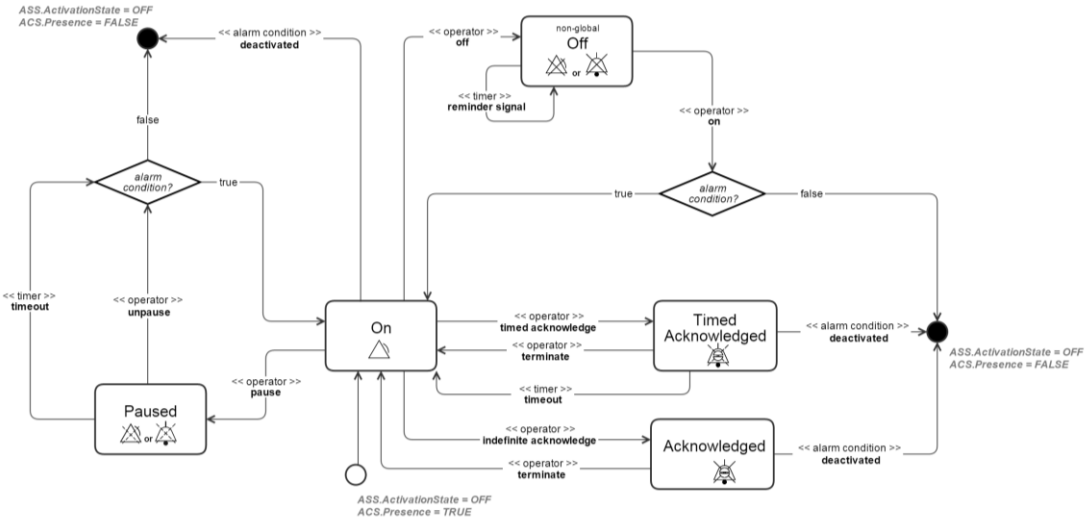
Figure 3—State diagram for Alert Systems. The Alarm Condition state means that the presence attribute of the AlertConditionState is set to true.



An AlertSignalState also possesses the mandatory attribute ActivationState that declares whether a signal will be generated or (temporarily) not if the related condition is present. Moreover, an AlertSignalState also possesses an optional attribute Presence that indicates if the signal is currently generated. For non-latching

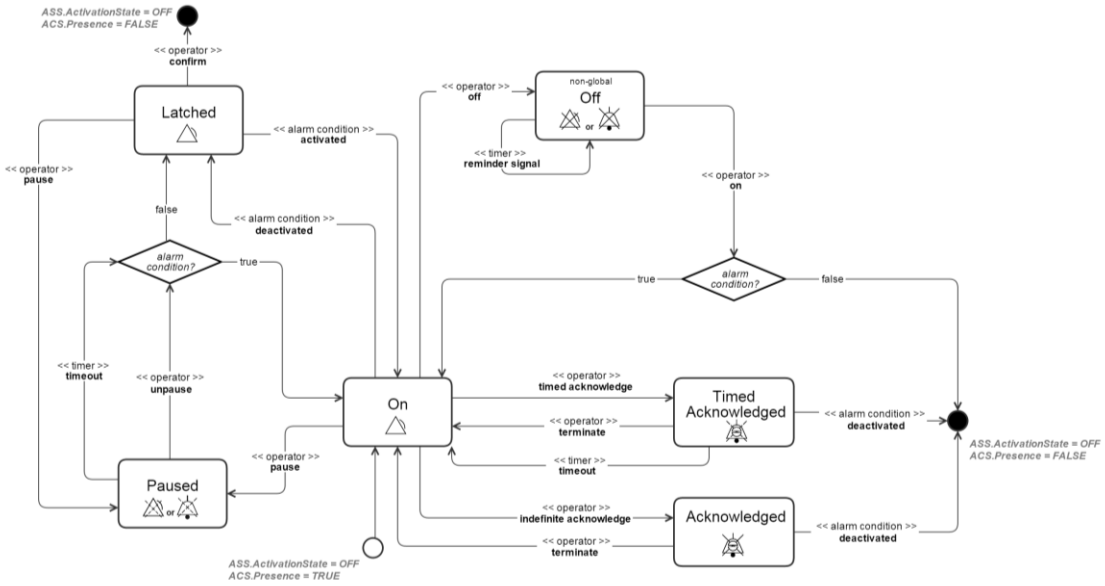
alert signals the state diagram of the Presence attribute as well as the relationship to the AlertConditionState is depicted in Figure 4.

Figure 4— Non-Latching Alert Signals.



For latching alert signals the state diagram of the Presence attribute as well as the relationship to the AlertConditionState is depicted in Figure 5.

Figure 5—Latching Alert Signals



5.3.3 MDIB Instance Versioning

Both AbstractDescriptors and AbstractStates elements have the possibility of versioning by using the attribute DescriptorVersion respectively StateVersion.

As a change in a device description respectively device state is associated with an Event Message (see sections 6.2) it has to be versioned as it may be referenced by a PARTICIPANT later on. Both are of type VersionCounter. An increased VersionCounter indicates a change in device description respectively device state.

R0030: A SERVICE PROVIDER SHALL NOT decrement a VersionCounter during operation.

R0031: A SERVICE PROVIDER SHALL reset the VersionCounter during a restart of the POC MEDICAL DEVICE or of itself, if correct versioning of the MDIB could not be guaranteed.

If a POC Medical Device cannot guarantee correct versioning of the a VersionCounter, e.g. it is not capable of storing the last change in case of power loss, it has to reset the VersionCounter. A reset of a VersionCounter should be handled as any other state change regarding with respect to related elements.

NOTE—It should be noted that by versioning elements in the MDIB also consistency inside the MDIB is assured by the SERVICE PROVIDER.

5.3.3.1 Descriptor Version

The first Descriptor instance that is captured SHALL have the DescriptorVersion number 0. As this is the default value for the attribute, the DescriptorVersion attribute can be omitted if the DescriptorVersion is 0.

R0032: A SERVICE PROVIDER SHALL assign the DescriptorVersion number 0 to the initial instance of a Descriptor.

The DescriptorVersion SHALL be incremented by one with every change of a descriptor.

There are multiple kinds of descriptor changes that have to be recognized by a DescriptorVersion increment:

- A child descriptor is added or deleted.
- The content of an attribute of the descriptor is updated.
- The content of an element of the descriptor with a non-descriptor type is updated.

Incrementing a DescriptorVersion is restricted to the descriptor recognizing one of these conditions. As stated in R0036 this does not cause an increment in its parent.

R0033: A SERVICE PROVIDER SHALL increment the DescriptorVersion by one if a direct child descriptor is added or deleted.

Adding or deleting of a Descriptor only increments the DescriptorVersion of its direct parent. This change does not propagate upwards to other ancestors in the containment tree.

R0034: A SERVICE PROVIDER SHALL increment the DescriptorVersion by one if the content of an attribute of the descriptor is updated.

An update of attribute is defined as an actual change of the value of the attribute or if the content actually is the same but it has been re-determined again.

R0035: A SERVICE PROVIDER SHALL increment the DescriptorVersion by one if the content of an element of the descriptor, including its attributes and child elements if existing, with non-descriptor type is updated.

R0036: A SERVICE PROVIDER MUST NOT increment the DescriptorVersion if an element that has itself a descriptor type is changed.

If an element that has itself a descriptor type is changed the descriptor version of the parent descriptor MUST NOT be incremented.

An informative example for descriptor changes can be found in Annex F.

5.3.3.2 State Version

The first State that is captured SHALL have the StateVersion number 0. As this is the default value for the attribute, the StateVersion attribute can be omitted if the StateVersion is 0.

R0037: A SERVICE PROVIDER SHALL assign the StateVersion number 0 to the initial instance of an AbstractState.

The StateVersion SHALL be incremented by one with every change of a state.

R0038: A SERVICE PROVIDER SHALL increment the StateVersion by one with every change of an AbstractState instance.

There are multiple kinds of state changes that have to be recognized by a StateVersion increment:

— The content of an attribute or element of the AbstractState or its specializations has changed.

— The content of a child element of the AbstractState or its specializations has been changed.

R0039: A Service Provider SHALL increment the StateVersion by one if the content of an attribute of the AbstractState is changed.

R0040: A Service Provider SHALL increment the StateVersion by one if the content of an element of the AbstractState is changed.

In order to capture only those changes that are related to exactly one instance of an AbstractState, the StateVersion MUST NOT be changed if an another AbstractState is changed even if that AbstractState is a child of the first AbstractState regarding the hierarchy in the containment tree.

R0041: A SERVICE PROVIDER MUST NOT increment the StateVersion if a child state in the containment tree that has itself a state type is changed.

5.3.3.3 MDDescription Version

The first MDDescription instance that is captured SHALL have the DescriptionVersion number 0. As this is the default value for the attribute, the DescriptionVersion attribute can be omitted if the DescriptionVersion is 0.

R0042: A SERVICE PROVIDER SHALL assign the DescriptionVersion number 0 to the initial instance of an MDDescription.

The MDDescription version SHALL be incremented by one with every change of a Descriptor that is part of that MDDescription container.

R0043: A SERVICE PROVIDER SHALL increment the DescriptionVersion of an MDDescriptor every time any of its children is added or removed or every time the DescriptorVersion of a descendent descriptor changes.

NOTE—The requirements of section 5.3.3.1 do not apply to an MDDescription instance, because MDDescription is not a Descriptor type. Therefore adding a new child to a MDDescription instance causes a total increment of its DescriptionVersion by one, because only R0043 applies and not R0033.

5.3.3.4 MDState Version

The first MDState instance that is captured SHALL have the StateVersion number 0. As this is the default value for the attribute, the StateVersion attribute can be omitted if the StateVersion is 0.

R0044: A SERVICE PROVIDER SHALL assign the StateVersion number 0 to the initial instance of an AbstractState.

The MDState version SHALL be incremented by one with every change of an AbstractState that is part of that MDState container.

R0045: A SERVICE PROVIDER SHALL increment the MDState every time an AbstractState instance in the MDState container is changed.

5.3.3.5 MDIB Version

The MDIB version SHALL be incremented by one with every change of a AbstractState or Descriptor.

R0046: A SERVICE PROVIDER SHALL increment the MDIBVersion by one every time the MDDescription version or MDState version in the MDIB is changed.

R0047: A SERVICE PROVIDER SHALL reset all version counters in the MDIB, if it resets the MDIBVersion.

R0048: A SERVICE CONSUMER SHOULD assume that all version numbers have been reset if it receives on direct request an MDIB Version from a SERVICE PROVIDER that is less than the last it had received.

5.3.4 State Generation Context

As an AbstractState most of the times does have only a meaning if regarded in the context that has been present at the time of capturing the state, the AbstractState as well as the MDState container possesses a StateVersion attribute as well as references to the DescriptorVersion.

As the AbstractContextState is also just an AbstractState it also possesses the StateVersion attribute and a change in the AbstractContextState instance leads to a change in the MDIBVersion as well as the MDState version. Moreover, a AbstractContextState possesses a BindingMDIBVersion and an UnbindingMDIBVersion attribute that allows defining with which version a context state element is associated to the POC MEDICAL DEVICE respectively not associated any longer.

This allows the determination of the associated AbstractContextState instance for all other AbstractState instances that belong to the same MDS.

An informative example for the state generation context can be found in Annex F.

6. Communication Model

The scope of this section is the definition of a set of messages that can be used to transmit the description or the state of a PARTICIPANTs as well as the definition of service interfaces that should be implemented by a SERVICE PROVIDER in order to receive and/or send the message on one or more SERVICE CONSUMERs.

It should be noted that definition of a transport binding is outside the scope of this specification. For this reason, this specification defines requirements that a transport binding has to fulfill in order to be able to transport the information elements defined in this specification.

6.1 General

IEEE 11073-10201 [11073-10201] defines the CMDISE (Common Medical Device Information Service Element) that provides an interface for managing the object instances that are part of the MDIB. Similar to the services of the CMDISE, the managed objects of a SERVICE PROVIDER are accessible through the services via the messages that are defined in this specification.

R0049: A SERVICE PROVIDER MAY offer Services and messages that are not defined in this specification.

Besides those Service and messages that are defined in this specification, a SERVICE PROVIDER may offer additional services with additional messages. This may be used to add device type specific functionality to the SOMDS.

6.2 Messages

R0050: A SERVICE PROVIDER MAY ignore messages from a SERVICE CONSUMER that has not been authenticated.

R0051: A SERVICE PROVIDER MAY decide which requested content it includes in a message based on the identity of the SERVICE CONSUMER.

R0052: A SERVICE PROVIDER SHALL include the current MDIB Version of its MDIB at the time a message is generated for transmission.

6.2.1 Descriptor Request Messages

This section defines the request and response messages that can be exchanged by a SERVICE PROVIDER and SERVICE CONSUMER in order to transmit the capability description of the PARTICIPANT.

The following request messages are defined:

— GetMDDescription

As a response to GetMDDescription, all or a filtered list of MDS CONTAINMENT TREEs that are contained in the MDIB of the SERVICE PROVIDER are returned.

— GetDescriptor

As a response to GetDescriptor, all or a filtered list of Descriptors that are contained in the MDIB of the SERVICE PROVIDER are returned.

— GetContainmentTree

As a response to GetContainmentTree, information about the MDS roots of each MDS CONTAINMENT TREE in the MDIB or the child elements that are defined by the parameter is returned. The information about a containment tree entry contains e.g. the specialization of the element in terms of a CodedValue, the number of subelements, its type and if applicable a reference to its parent element.

R0053: A SERVICE CONSUMER MAY use GetDescriptor and GetContainmentTree if it can expect that the whole description contained in the response to a GetMDDescription is too large to process.

— GetDescriptorsFromArchive

TODO

The normative outline of the messages is defined in Annex C.

6.2.2 State Request Messages

This section defines the request and response messages that can be exchanged by a SERVICE PROVIDER and SERVICE CONSUMER in order to transmit the complete set of AbstractState instances in the MDIB or parts of it.

— GetMDState

As a response to GetMDState, a set of all or a filtered list of AbstractState instances that are contained in the MDIB of the SERVICE PROVIDER is returned.

R0054: A SERVICE PROVIDER MAY not include instances of a ContextState in a response to a GetMDState message even if the handle is part of the request if the contextual information allows to identify the patient.

— GetContextStates

As a response to GetContextStates, a set of all or a filtered list of ContextState instances that are contained in the MDIB of the SERVICE PROVIDER is returned.

— GetStatesFromArchive

TODO

— GetSupportedLanguages

TODO

The normative outline of the messages is defined in Annex C.

6.2.3 Descriptor Event Messages

This section defines the messages that can be exchanged by a service provider and service consumer as an event in order to transmit information about changes in a Descriptor or the containment tree.

— DescriptionModificationReport

A message of type DescriptionModificationReport indicates by its ModificationType, that either an object in the containment tree has been created, updated or deleted.

NOTE—When a descriptor X is added to or deleted from some parent descriptor Y, there shall only be a single DescriptionModificationReport with one ReportDatail containing X. The update of Y by adding X is implied by this message and does not trigger a separate notification.

NOTE—When updating a descriptor triggers a DescriptionModificationReport and this descriptor contains child descriptors, only the descriptor itself shall be transmitted in the ReportPart. All child descriptors shall be stripped from it beforehand. This behaviour is similar to the GetDescriptor message.

An informative example for descriptor modifications can be found in Annex F.

In order to ensure consistency at a service consumer any descriptor that is not derived from an AbstractMDSDescriptor and is added to the MDS Containment Tree the ParentDescriptor has to be provided.

R0055: A SERVICE PROVIDER SHALL include the ParentDescriptor for any AbstractDescriptor that is not derived from AbstractMDSDescriptor if the DescriptionModificationReport comprising a create event is transmitted to a SERVICE CONSUMER.

The normative outline of the messages is defined in Annex C.

6.2.4 State Event Messages

This section defines the messages that can be exchanged by a SERVICE PROVIDER and SERVICE CONSUMER as an event in order to transmit information about changes in an AbstractState.

— EpisodicMetricReport/PeriodicMetricReport

A message that is sent on change or periodically and contains metric states.

— EpisodicAlertReport/PeriodicAlertReport

A message that is sent on change or periodically and contains alert states.

— OperationalStateChangedReport

A message that is sent if the state of a remote operation changed.

— AbstractContextChangedReport

A message that is sent if the state of a context changed. The message does not contain the actual state information, but handles that can be utilized to retrieve the new state via the GetContext message.

The normative outline of the messages is defined in Annex C.

6.2.5 State Modification Request Messages

This section defines additional constraints for the utilization of request messages that modify the state of a POC MEDICAL DEVICE.

R0056: A SERVICE CONSUMER SHALL subscribe to OperationInvokedReport messages of a SOMDS SERVICE before requesting a state modification on that SOMDS SERVICE.

R0057: A SERVICE PROVIDER MAY reject a state modification request if the SERVICE CONSUMER has not subscribed to OperationInvokedReport messages before requesting a state modification on that SOMDS SERVICE.

R0103: A SERVICE CONSUMER SHOULD subscribe to any State that it is intending to modify with a state modification request in order to receive changes to the State.

R0058: A SOMDS SERVICE MAY reject a state modification request if the number of potential concurrent SERVICE CONSUMERs that are allowed to modify the state has is greater than a threshold.

R0059: A SOMDS SERVICE MAY allow more than one SERVICE CONSUMERs to modify the state concurrently.

R0060: A SOMDS SERVICE SHALL ensure that concurrent state modification do not create a HAZARDOUS SITUATION for the patient or operator.

— SetValue

A message that requests the modification of a decimal field of an element.

— SetString

A message that requests the modification of a string field of an element.

— SetRange

A message that requests the modification of a range of an element.

— SetAlertState

A message that requests the modification of an alert state.

— Activate

A message that requests the invocation of a predefined job or function.

— SetContextState

A message that requests the modification of the list of context states.

— AbstractSetResponse

The response to all state modification request messages is derived from this base message type that contains information about first execution state of the request. See OperationInvokedReport in section 6.2.6

The normative outline of the messages is defined in Annex C.

6.2.6 State Modification Event Messages

This section defines the messages that can be exchanged by a Service Provider and Service Consumer as an event in order to transmit information about changes in the execution of a state modification request.

— OperationInvokedReport

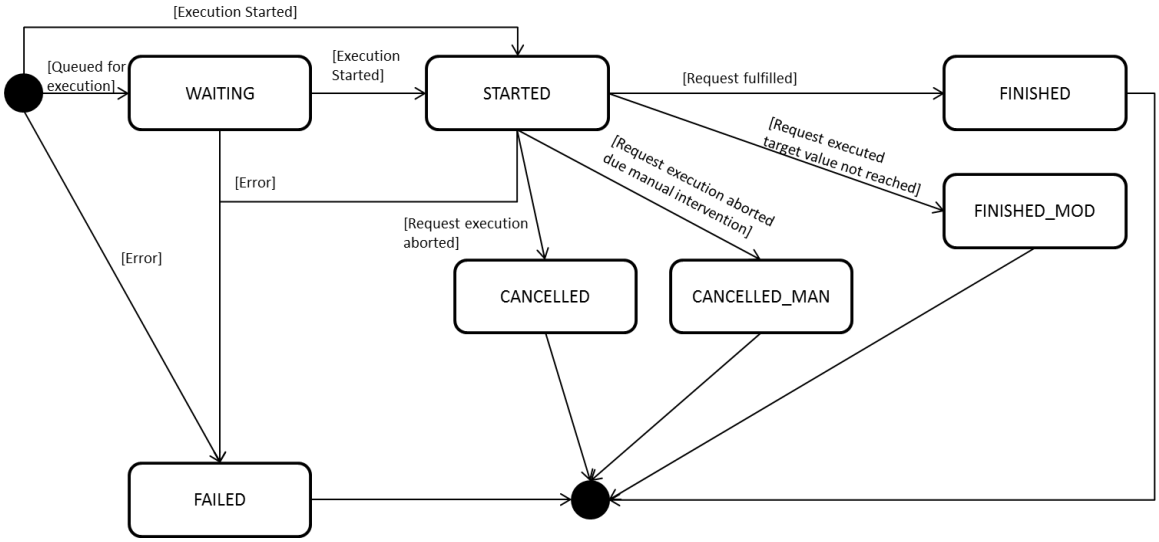
A message of type OperationInvokedReport indicates that there has been a change in the execution of a a state modification request.

R0061: A Service Provider SHALL send an OperationInvokedReport if there has been a change in the execution of a state modification request even if the new state has already been transmitted in an AbstractSetResponse message.

This message SHALL be sent even it has already been communicated in a AbstractSetResponse message in order to allow any subscriber to monitor the progress of the execution of a state modification request.

The state diagram for the execution state of modification request is depicted in Figure 6.

Figure 6—State diagram for the execution state of a modification request



It should be noted that depending on the transport the AbstractSetResponse as well as the OperationInvokedReport messages may arrive in a different order. Logically older execution states for the same modification request may be omitted by a Service Consumer even if they arrive after logically younger execution states.

The normative outline of the messages is defined in Annex C.

6.3 Services

BICEPS services defines a set of services that control access to the managed medical objects of the PARTICIPANT's MDIB. The service interfaces are grouped according to functional groups. As mentioned beforehand, the message component (see section 4.2) comprises the message definitions that can be used to convey data between PARTICIPANTS.

It should be noted that a BICEPS compliant medical device service does not need to implement all BICEPS services.

R0065: A BICEPS compliant Binding SHALL provide means to subscribe to notifications of a SERVICE.

The BICEPS Services specification does not define a means to subscribe to notifications of the service, but rather specifies a requirement for the transport communication protocol to support subscription management.

6.3.1 Get Service

The mandatory GET SERVICE defines an interface that allows a client to retrieve the current description and state of the medical device in a polling mode.

R0062: A SERVICE PROVIDER SHALL provide an instance of a GET SERVCE.

R0063: A BICEPS compliant Binding SHALL provide means to send request messages to a SERVICE.

The normative outline of the service is defined in Annex D.

6.3.2 State Event Report Service

The optional STATE EVENT REPORT SERVICE defines an interface that allows a client to retrieve notifications about changes of a state that is stored in the MDIB.

R0064: A SERVICE PROVIDER MAY provide an instance of the STATE EVENT REPORT SERVICE.

The defined medical device service notifications can be classified either as description update notifications or as state update notifications. The latter can be furthermore distinguished into periodic state update notifications and episodic update notifications.

The normative outline of the service is defined in Annex D.

6.3.3 Descriptor Event Report Service

The optional DESCRIPTOR EVENT REPORT SERVICE defines an interface that allows a client to retrieve notifications about either changes of the description part of the MDIB.

R0104: A SERVICE PROVIDER MAY provide an instance of the DESCRIPTOR EVENT REPORT SERVICE.

The normative outline of the service is defined in Annex D.

6.3.4 Waveform Service

The optional WAVEFORM SERVICE defines an interface that allows a client to retrieve the state of a stream for real time sample array metrics (e.g., waveforms).

R0066: A SERVICE PROVIDER MAY provide an instance of the WAVEFORM SERVICE.

Only the WaveformStream medical device service operation is defined for this BICEPS service.

R0067: A BICEPS compliant BINDING SHALL provide means to subscribe to streams of the service.

The BICEPS Services specification does not define a means to subscribe to streams of the service, but rather specifies a requirement for the transport communication protocol to support subscription management.

The normative outline of the service is defined in Annex D.

6.3.5 Remote Control Service

The optional REMOTE CONTROL SERVICE defines an interface that allows a client to change the state of the medical device.

R0068: A SERVICE PROVIDER MAY provide an instance of the REMOTE CONTROL SERVICE.

It should be noted that all operations are defined as a pair of request and response message. The response message does not contain the resulting new state of the managed medical object caused by the remote invocation, but rather the state of the execution of the remote invocation request. Subsequent state changes of the execution of the remote invocation request are conveyed to a client by notifications. The resulting state of the managed medical object is also conveyed using the respective event notification from the EVENT REPORT SERVICE.

The normative outline of the service is defined in Annex D. An informative example for remote control can be found in Annex F.

6.3.6 Context Service

R0069: A SERVICE PROVIDER MAY provide the CONTEXT SERVICE.

The optional CONTEXT SERVICE possesses a service interface that allows retrieval and modification of contextual information for a POC MEDICAL DEVICE. As contextual information, especially patient data, may be used to identify a person, the operations are encapsuled in this special service. This allows a different handling that may be required w.r.t. security for example due to regulatory requirements like HIPAA.

The normative outline of the service is defined in Annex D.

6.3.7 Archive Service

R0100: A SERVICE PROVIDER MAY provide the ARCHIVE SERVICE.

The optional ARCHIVE SERVICE possesses a service interface that allows retrieval of the history of descriptors and states.

The normative outline of the service is defined in Annex D.

6.3.8 Localization Support Service

R0101: A SERVICE PROVIDER MAY provide the LOCALIZATION SERVICE.

The optional LOCALIZATION SERVICE possesses a service interface that allows retrieval localization information for strings.

The normative outline of the service is defined in Annex D.

6.3.9 Device-specific Services

R0070: A SERVICE PROVIDER MAY provide the device-specific services.

If a medical device service needs to offer data or remotely invocable operations that are not defined in the BICEPS Services specification, it is possible to describe the medical device service operations in the description part of the MDIB. The descriptor allows defining the type of the operation and the interface name where the operation is included. Additional information may be added using the extensibility mechanisms defined in the descriptor.

7. Discovery

BICEPS discovery fosters Plug & Play by defining requirements for a transport communication protocol as well as by defining a set of medical device service operations that allow a SERVICE CONSUMER to retrieve a description of the capabilities of the medical device offered on the network.

An informative example for BICEPS discovery can be found in Annex F.

7.1 Implicit Discovery

Implicit discovery is based on messages that are communicated by a SERVICE PROVIDER when it enters or leaves the network or if something in its context has changed substantially.

R0073: A SERVICE PROVIDER SHALL announce its presence if it is ready to exchange messages with a SERVICE CONSUMER.

R0074: A SERVICE PROVIDER SHOULD announce its upcoming absense if it is switching to a mode where it is not ready to exchange messages with a SERVICE CONSUMER anymore.

R0075: A SERVICE PROVIDER SHALL include an unique identifier in a discovery message that allows its identification by a SERVICE CONSUMER.

R0076: A SERVICE PROVIDER SHALL include an identifier in a discovery message that allows its identification as a BICEPS compliant SERVICE PROVIDER by a SERVICE CONSUMER.

R0078: A BICEPS compliant BINDING SHALL provide means for implicit discovery.

7.2 Explicit Discovery

R0079: A SERVICE PROVIDER SHALL respond to a discovery query with searches for BICEPS compliant devices by the identifier biceps:MedicalDevice.

For that purpose, BICEPS Discovery defines a discovery type that every medical device service that is BICEPS compliant has to listen to in a discovery message.

R0080: A BICEPS compliant BINDING SHALL provide means for explicit discovery.

7.3 Context-based Discovery

For many use cases a SERVICE CONSUMER is not interested in every kind of MDS or every instance of a kind of MDS, but only in those that belong to a specific context. For this reason, the SERVICE CONSUMER should be enabled to determine the context that a MDS is operated in during discovery.

R0072: A SOMDS SERVICE SHOULD include context information into its discovery information for each type of context that it indicates to support in its capability description.

NOTE—It should be noted that a SERVICE PROVIDER may not include context information into implicit discovery messages, e.g. due to privacy issues.

R0081: A SERVICE PROVIDER SHALL respond to a discovery query which searches for a context that is identified in the message.

8. Non-Functional Requirements

This specification defines a set of non-functional requirements that have to be fulfilled by a transport binding.

NOTE—It should be noted, that BICEPS does not define any non-functional requirements data elements on its own as all non-functional requirements are related to the transport communication protocol. Hence, the BICEPS participant information model as well as the communication model rather defines extensibility points that allow embedding of non-functional requirements for a transport communication protocol.

R0082: A POC MEDICAL DEVICE SHOULD include non-functional requirements into the MDDescription structure that describes its capability.

8.1 Patient Safety Considerations

This section defines the requirements that a transport communication protocol has to fulfill in order to ensure patient safety while transporting messages especially focusing on trust establishment between PARTICIPANTs and detection of data corruption during transport.

In order to allow communication between a SERVICE PROVIDER and a SERVICE CONSUMER the risk management of the SERVICE PROVIDER may define that only eligible SERVICE CONSUMERs may be communicated with. Eligible SERVICE CONSUMERs can for example be such SERVICE CONSUMER that can be trusted that they know about the consequences of their behavior in the communication, e.g. if they execute a remote control command.

R0083: A BICEPS compliant Binding SHOULD provide means establishing trust between PARTICIPANTs.

A means for trust establishment can be credentials of a SERVICE CONSUMER that are exchanged during the communication, e.g. a username and password combination or a certificate signed by a trusted authority that defines the qualification of the SERVICE CONSUMER.

A typical scenario that may lead to a situation where patient safety cannot be ensured is if the data exchanged between the PARTICIPANTs gets corrupted. For this reason, a compliant binding shall provide means to ensure that the communication partner may be able to detect corrupted data.

R0084: A BICEPS compliant binding SHALL provide means for allowing the detection of corrupted State in a message.

R0085: A BICEPS compliant binding SHOULD provide means for the detection of corrupted remote control commands.

It should be noted that also the Descriptors may contain information that lead to critical situations and therefore a compliant binding has to provide also means to allow the detection of corrupted data in a message.

R0086: A BICEPS compliant binding SHALL provide means for allowing the detection of corrupted Descriptors in a message.

R0086: A BICEPS compliant binding SHALL provide means to allow the detection of the lost of connection of a PARTICIPANT to another PARTICIPANT.

8.2 Cybersecurity Considerations

This section defines the cybersecurity requirements that are related to message exchange between PARTICIPANTs. Out of scope of this specification are requirements related to general cybersecurity requirements of the device itself.

As PARTICIPANTs may exchange information that identifies a person or is linkable to a person (e.g. Personally Identifiable Information, Protected Health Information) while communicating that information, cybersecurity means may have to be applied to ensure that a transport complies with national regulations.

R0087: A BICEPS compliant binding SHALL provide means that allows confidential transport of a message between two PARTICIPANTs.

Even if data does not get corrupted by a failure during transport, the intentional or unintentional modification of data transported in a message by a third party may lead to patient harm. For this reason, means to ensure integrity of the messages have to be provided by a compliant binding.

R0088: A BICEPS compliant binding SHALL provide means that ensures integrity of a message during transport between two PARTICIPANTs.

For logging systems and for access control mechanism it may be needed that a receiver of a message is able to determine the originator of a message.

R0089: A BICEPS compliant binding SHALL provide means that allows determining the PARTICIPANT that originates a message.

8.3 Clinical Effectiveness

This section defines the requirements related to the clinical effectiveness of a SOMDS.

For clinical purposes the correlation of data from multiple SERVICE PROVIDER may be required. For this reason, time information may be required to ensure that the sequence of data is interpreted correctly.

R0090: A BICEPS compliant binding SHALL provide means that allows synchronizing time in the SOMDS.

R0091: A BICEPS compliant binding SHOULD provide means that allows synchronizing time between the PARTICIPANTs and an external clock.

For clinical purposes, information exchanged between two PARTICIPANTs may have to be transported with certain guarantees regarding the maximum latency and jitter. For this purpose, a compliant binding has to provide means to define and monitor the fulfillment of the requirements.

R0092: A BICEPS compliant binding SHOULD provide means to define Quality-of-Service metrics for a communication between two PARTICIPANTs.

8.4 Regulatory Considerations

This section defines requirements related to regulatory requirements that have not been addressed by the previous sections.

In a SOMDS with independent components of different manufactures logging of data is expected to be required by regulatory bodies. It is not within the scope of this document to define the type of data logging (e.g. distributed vs. central logging), but to ensure that the transport binding is capable of fulfilling the required information.

R0093: A BICEPS compliant binding SHALL provide means that allows to distinguish unique messages in a sequence of messages with potential duplicates.

9. Conformance

A conformant implementation SHALL satisfy all the SHALL or REQUIRED level requirements defined herein.

TODO Add table with all SHALL requirements

Annex A

(normative)

Constants

Annex B

(normative)

Participant Information Model

See
https://sourceforge.net/p/opensdc/bicepsmodel/ci/068dd618aea7a53793b25e35214d429e5f94b5f6/tree/BICEPS_DomainModel.xsd?format=raw

Annex C

(normative)

Message Model

See
https://sourceforge.net/p/opensdc/bicepsmodel/ci/068dd618aea7a53793b25e35214d429e5f94b5f6/tree/BICEPS_MessageModel.xsd?format=raw

Annex D

(normative)

Service Model

Get Service [mandatory]

- GetMdib

- GetMdDescription

- GetMdState

Containment Tree Service [optional]

This service SHOULD be used by a service consumer if it is offered by a Service Provider.

- GetContainmentTree

- GetDescriptor

Set Service [optional]

If a device is intended to be remote controlled, a device SHALL implement this service interface.

- SetRange

- SetValue

- SetString

- SetAlertState

- Activate

- SetComponentState

- OperationInvokedReport

State Event Service [optional]

- PeriodicMetricReport

- EpisodicMetricReport

- PeriodicAlertReport

- EpisodicAlertReport

- PeriodicOperationalStateReport

- EpisodicOperationalStateReport

- PeriodicComponentReport

- EpisodicComponentReport

- SystemErrorReport

Context Service [optional]

If a device supports context handling, a device SHALL implement this service interface.

- SetContextState

- GetContextStates

- EpisodicContextReport

- PeriodicContextReport

Description Event Service [optional]

If a device may change its description, a device SHALL implement this service interface.

- DescriptionModificationReport

Waveform Service [optional]

If a device streams waveforms, a device SHALL implement this service interface.

- WaveformStream

Localization Service [optional]

If multiple languages are supported, a device SHOULD implement this service interface

- GetLocalizedText

- GetSupportedLanguages

Archive Service [optional]

If history of descriptors and/or states is supported, a device SHOULD implement this service interface

- GetDescriptorsFromArchive

- GetStatesFromArchive

Annex E

(normative)

Extension Model

element **Extension**

properties	content complex
used by	complexTypes **AbstractDescriptor AbstractGet AbstractGetResponse AbstractMetricValue AbstractReport AbstractReportPart AbstractSet AbstractSetResponse AbstractState Annotation ArgumentType BaseDemographics CalibrationInfo CauseInfo ClinicalInfo CodedValue ContainmentTree ContainmentTreeEntry EnumNomenRef ImagingProcedure InstanceIdentifier LocationDetail MdDescription Mdib MdState Measurement MeasurementState Order OrderDetail PersonReference Range RemedyInfo Retrievability SystemMetaData**
annotation	documentation The Extension element is a container to collect extensions of any kind. In the scope of SDC extensions should be added to any meaningful MDIB and message model entities (any component descriptors and states, patient data objects, coded values, get request and response messages, etc.). For better distinction extensions SHOULD be wrapped into container elements belonging to a namespace where the extension is specified. Example: <![CDATA[<tns:Extension xmlns:tns="http://extension-point-uri/15/03" xmlns:ext="http://concrete-extension-namespace"> <ext:AdditionalInfo><!-- ... --></ext:AdditionalInfo> </tns:Extension>]]> Here ext:AdditionalInfo is the wrapper element containing the extension's information.

attribute **MustUnderstand**

type	**xsd:boolean**
properties	default false
annotation	documentation In cases where an extension modifies the meaning of the element that contains it, a MustUnderstand attribute is senseful. This means that the data cannot savely be processed unless the application knows the meaning of the extension. A MustUnderstand signed extension could be look like this: <![CDATA[<tns:Extension xmlns:tns="http://extension-point-uri/15/03" xmlns:ext="http://concrete-extension-namespace"> <ext:AdditionalInfo tns:MustUnderstand="true"><!-- ... --></ext:AdditionalInfo> </tns:Extension>]]> Extensions are not required to provide a MustUnderstand attribute. If no MustUnderstand attribute is present, false is assumed. The MustUnderstand attribute is conceptually borrowed from SOAP header's "mustUnderstand" attribute.

Annex F

(informative)

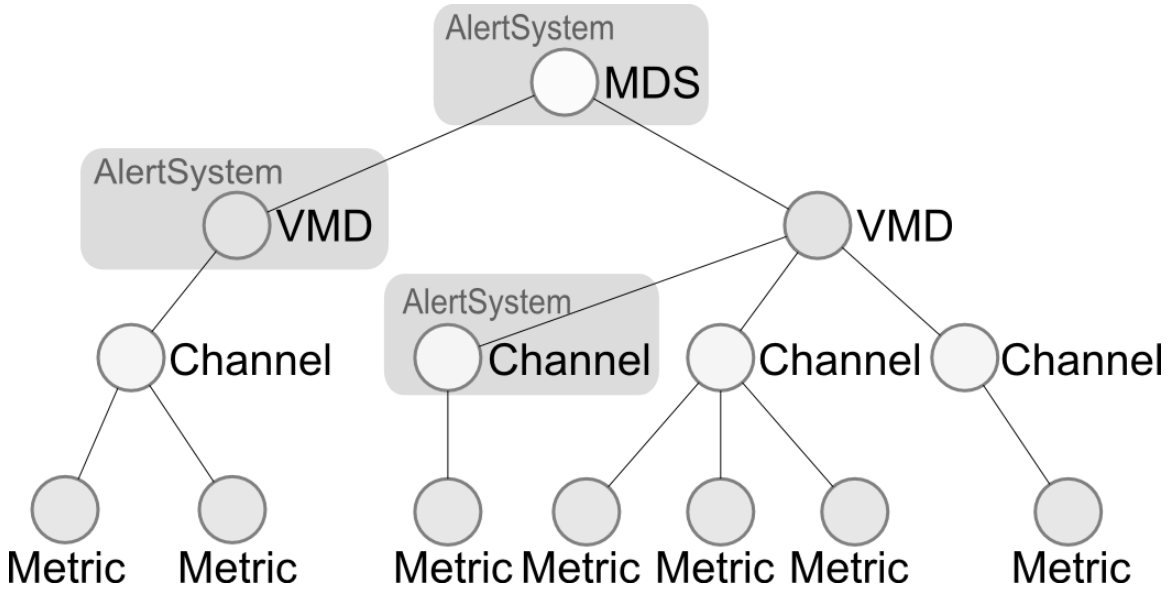
Examples

This section comprises examples for the participant and communication model as well as for the discovery mechanisms.

MDS Containment Tree

Fehler! Verweisquelle konnte nicht gefunden werden. shows a possible containment tree. It does not illustrate the possibility to group multiple MDSs in one MDDescription.

Figure 7— Exemplary instance of a MDS Containment Tree



Descriptor Versioning

If for example a MDSDescriptor is added to the MDDescription of an MDIB, the DescriptionVersion is incremented. If afterwards a MetricDescriptor inside of the containment tree of that MDSDescriptor changes, the DescriptionVersion of the MDDescription as well as the DescriptorVersion of the MetricDescriptor is incremented. If the MDSDescriptor is removed from the MDDescription than the DescriptionVersion is incremented again.

State Generation Context

If a SERVICE CONSUMER receives an instance of a AbstractContextState CS with a version C1 at time t1 and the instance of a AbstractContextState CS is changed at time t4 then the version C2=C1+1 holds. The MDIBVersion at time t1 is M1 and the MDIBVersion at time t4 is M4. If the AbstractContextState is associated with the PoC Medical Device starting from MDIBVersion M1, the BindingMDIBVersion at time time t1 and t4 is M1.

If a SERVICE CONSUMER receives another AbstractState AS with version S1 at time t2 and the instance of that AbstractState AS is changed at time t3 then the version S2=S1+1 holds. If that instance of an AbstractState AS is changed again at time t5 then the version S3=S2+1 holds.

The MDIBVersion at time t1 is M1, at time t2 is M2,…, and at time t5 is M5.

The contextual information for the interpretation of AbstractState AS at time t3 is the instance of AbstractContextState CS with BindingMDIBVersion M1 < M3 and UnbindingMDIBVersion NULL < M3.

Descriptor Changed Events

Adding or deleting subtrees of descriptors in the Mdib can be captured in DescriptionModificationReports in different ways.

For example, there might be a medical device, that has pluggable connectors for different probes. When a probe is attached, the device adds a new channel "Ch" to an existing VMD in its MDIB. Finally a numeric metric descriptor "N" is added to the channel to represent the measured value. There are three possibilities how to use the DescriptionModificationReports for these changes:

a) Create a single DescriptionModificationReports, that contains two ReportDetails, where the first ReportDetail contains "Ch" only, the second ReportDetail contains "N". Changing the order of the ReportDetails is forbidden here, because the report parts are processed sequentially and adding "N" to a channel, that not yet exists, is an invalid operation.

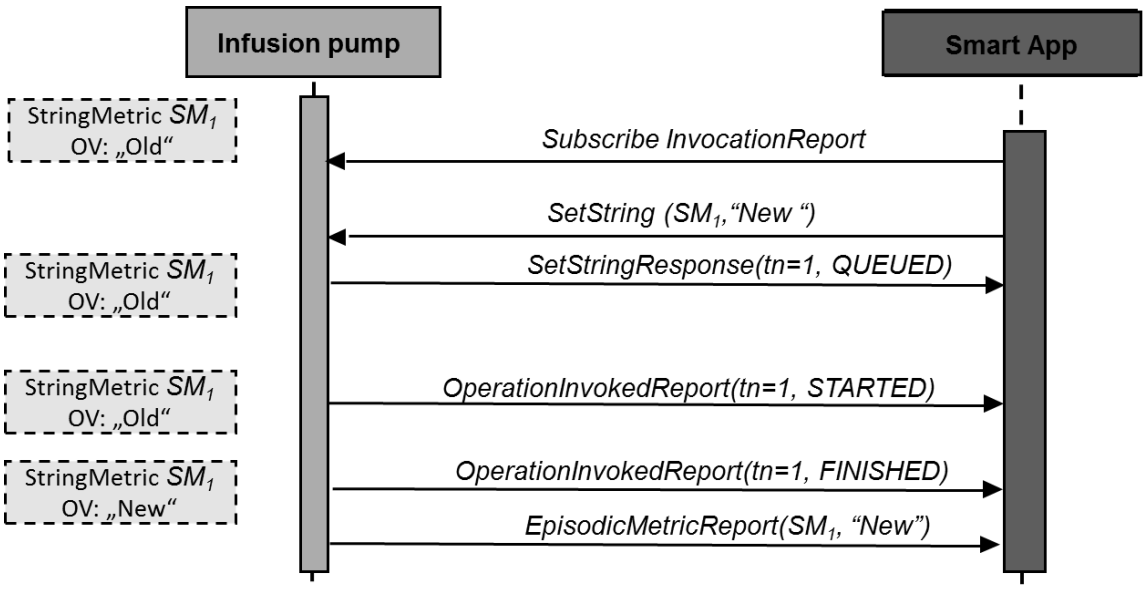
b) First add "N" to "Ch" to form the structure "ChN". Then put a single ReportPart into a DescriptionModificationReports, that contains the combined structure "ChN".

c) Create two DescriptionModificationReports, where each contains a single ReportDetail. The first report contains "Ch", the second report contains "N". While this variant is technically possible, it is strongly discouraged, because each message is sent separately over the network and causes unneccessary processing overhead.

Remote Control

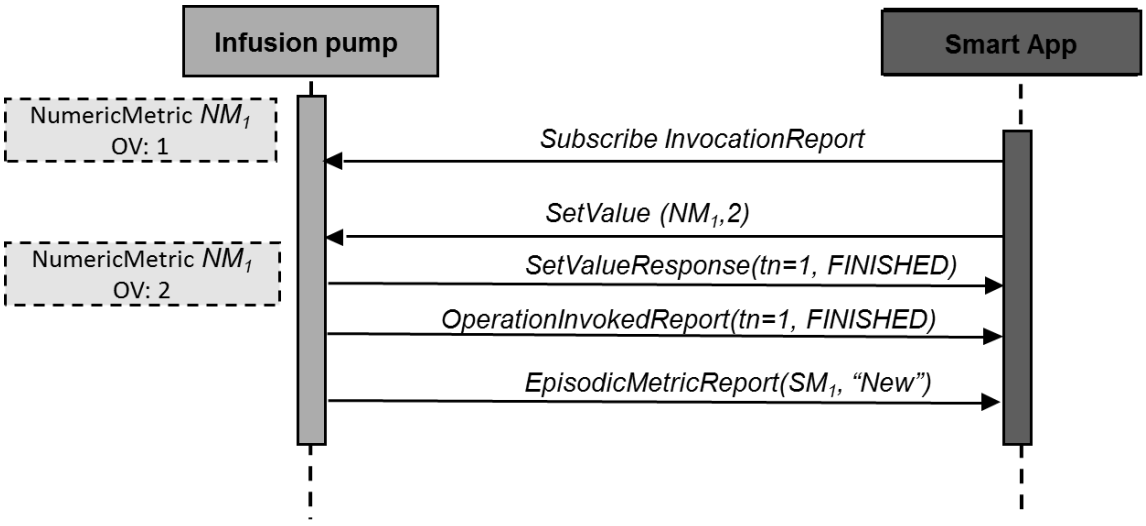
In **Fehler! Verweisquelle konnte nicht gefunden werden.** an exemplary remote invocation of a SetValue medical device service operation by a Smart App on an infusion pump is depicted. The Smart App has discovered the medical device service beforehand, and subscribes to OperationInvokedReport as well as the EpisodicMetricReport. Afterwards, it sends a SetString request to the BICEPS Set Service of the infusion pump in order to change the current value “Old” to the requested value “New”. The SetStringResponse contains the first state of the execution of the remote invocation: “Queued”. After some time, the MDIB of the infusion pump starts to process the remote invocation request and changes the value to the requested one. It then sends two event reports via its BICEPS Event Report Service: an OperationInvokedReport with the state set to “Finished” and an EpisodicMetricReport containing the new state of the StringMetric.

Figure 8—Exemplary remote invocation of a SetString medical service operation by a smart app on an infusion pump.



In **Fehler! Verweisquelle konnte nicht gefunden werden.** an exemplary remote invocation of a SetValue medical device service operation by a Smart App on an infusion pump is depicted. The Smart App has discovered the medical device service beforehand, and subscribes to OperationInvokedReport as well as the EpisodicMetricReport. Afterwards, it sends a SetValue request to the BICEPS Set Service of the infusion pump in order to change the current value ‘1’ to the requested value ‘2’. The SetValueResponse contains immediately the state “FINISHED” as the infusion pump has directly processed the remote invocation request and changed the value to the requested one. It then sends two event reports via its BICEPS Event Report Service: an additional OperationInvokedReport with the state set to “Finished” and an EpisodicMetricReport containing the new state of the NumericMetric.

Figure 9—Exemplary remote control invocation of a SetValue medical device service operation by a smart app on an infusion pump.



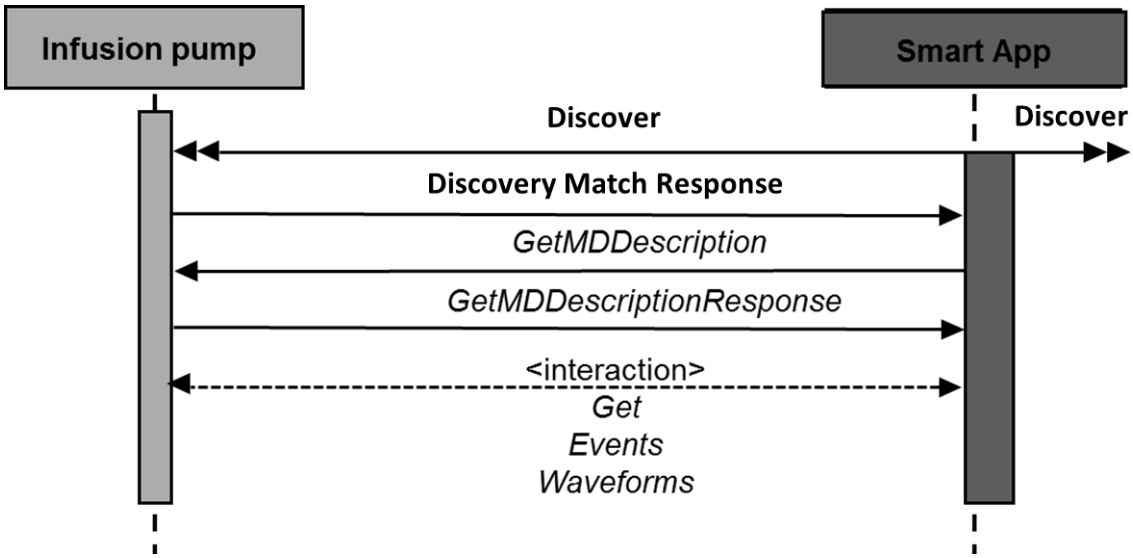
Discovery

An exemplary explicit discovery sequence is depicted in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

The set of medical device service operations is closely related to the descriptive component of the BICEPS-MIM MDIB component and the related messages to retrieve the description are explained in section 6.2.1.

Fehler! Verweisquelle konnte nicht gefunden werden. depicts a clinical application client that searches for a participant in the distributed system that offers certain capabilities. In this case it looks for a possibility to control an infusion pump.

Figure 10—Exemplary explicit discovery sequence



Annex G

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[IEC 60601-1-8] IEC 60601-1-8/AMD1:2012 TODO

[DPWS 1.1] TODO