



BSI Standards Publication

Health informatics — Personal health device communication

Part 10207: Domain information and service model for service-oriented point-of-care medical device communication

National foreword

This British Standard is the UK implementation of ISO/IEEE 11073-10207:2019.

The UK participation in its preparation was entrusted to Technical Committee IST/35, Health informatics.

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

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2019
Published by BSI Standards Limited 2019

ISBN 978 0 539 01874 5

ICS 35.240.80

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2019.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Health informatics — Personal health
device communication —**

Part 10207:

**Domain information and service
model for service-oriented point-of-
care medical device communication**

*Informatique de santé — Communication entre dispositifs de santé
personnels —*

*Partie 10207: Informations de domaine et modèle de services pour la
communication orientée services entre dispositifs médicaux sur le site
des soins*



IEEE

Reference number
ISO/IEEE 11073-10207:2019(E)



Documents

Document Sharing Hub
© IEEE 2018



COPYRIGHT PROTECTED DOCUMENT

© IEEE 2018

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

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

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 (see www.iso.org/directives).

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

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.

ISO/IEEE 11073-10207 was prepared by the IEEE 11073 Standards Committee of the IEEE Engineering in Medicine and Biology Society (as IEEE Std 11073-10207-2017) and drafted in accordance with its editorial rules. It was adopted, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Technical Committee ISO/TC 215, *Health informatics*.

A list of all parts in the ISO 11073 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Abstract: Within the context of the ISO/IEEE 11073™ family of standards for point-of-care medical device communication, a Participant Model derived from the ISO/IEEE 11073-10201 Domain Information Model is provided in this standard. The Participant Model specifies the structure of medical information objects. This standard also defines an abstract Communication Model to support the exchange of medical information objects. All elements of the Participant Model and Communication Model are specified using XML Schema. Core subjects of the Participant Model comprise modelling of medical device-related data, e.g., measurements and settings, alert systems, contextual information (e.g., patient demographics and location information), remote control, and archival information. Model extensibility is provided inherently through the use of XML Schema.

Keywords: alert systems, IEEE 11073-10207™, medical device communication, patient, point-of-care, remote control, service-oriented architecture, XML Schema

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

Copyright © 2018 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 21 February 2018. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-4627-3 STD22961
Print: ISBN 978-1-5044-4628-0 STDPD22961

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

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 Notices and Disclaimers Concerning IEEE Standards Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

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; noninfringement; 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, nonexclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, noncommercial 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 nondiscriminatory. 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 standard was submitted to the IEEE-SA Standards Board for approval, the Point of Care Devices Working Group had the following membership:

Jan Wittenber, *Chair*
Stefan Schlichting, *Subgroup Chair*

Bjoern Anderson
Malcolm Clarke
Todd Cooper
Chris Courville
Michael Faughn
Kenneth Fuchs
John Garguilo

Frank Golatowski
David Gregorczyk
Kai Hassing
John Hatcliff
Stefan Karl
Martin Kasparick
Koichiro Matsumoto
Joerg-Uwe Meyer

Stephan Poehlsen
Tracy Rausch
John Rhoads
Paul Schluter
Masato Tanaka
Eugene Vasserman
Stan Wiley

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

Bjoern Andersen
Susan Burgess
Craig Carlson
Keith Chow
Malcolm Clarke
Todd Cooper
Sourav Dutta
Kenneth Fuchs
Joel Goergen
Frank Golatowski

David Gregorczyk
Randall Groves
Kai Hassing
Robert Heile
Werner Hoelzl
Noriyuki Ikeuchi
Atsushi Ito
Stefan Karl
Piotr Karocki

Martin Kasparick
Joerg-Uwe Meyer
Stephan Poehlsen
Beth Pumo
Stefan Schlichting
Eugene Stoudenmire
Walter Struppler
J. Wiley
Jan Wittenber
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 6 December 2017, it had the following membership:

Jean-Philippe Faure, *Chair*
Gary Hoffman, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Chuck Adams
Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Michael Janezic

Thomas Kochy
Joseph L. Koepfinger*
Kevin Lu
Daleep Mohla
Damir Novosel
Ronald C. Petersen
Annette D. Reilly

Robby Robson
Dorothy Stanley
Adrian Stephens
Mehmet Ulema
Phil Wennblom
Howard Wolfman
Yu Yuan

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 11073-10207-2017, Health informatics—Point-of-care medical device communication—Part 10207: Domain Information and Service Model for Service-Oriented Point-of-Care Medical Device Communication.

ISO/IEEE 11073 standards enable communication between medical devices and external computer systems. They provide automatic and detailed electronic data capture of patient vital signs information and device operational data. The primary goals of these standards are to

- Provide real-time, plug-and-play interoperability for patient-connected medical devices.
- Facilitate the efficient exchange of vital signs and medical device data, acquired at the point of care, in all health care environments.

“Real-time” means that data from multiple devices can be retrieved, time correlated, and displayed or processed in fractions of a second. “Plug-and-play” means that all the clinician has to do is make the connection—the systems automatically detect, configure, and communicate without any other human interaction.

“Efficient exchange of medical device data” means that information that is captured at the point of care (e.g., patient vital signs data) can be archived, retrieved, and processed by many different types of applications without extensive software and equipment support and without needless loss of information. The standards are especially targeted at acute and continuing care devices, such as patient monitors, ventilators, infusion pumps, and electrocardiogram (EKG) devices. This family of standards can be layered together to provide connectivity optimized for the specific devices being interfaced.

Note that normative statements of requirements are presented in this standard in the following manner:

Rnnnn: Statement text here.

where "nnnn" is replaced by a number that is unique among the requirements in this standard and thereby forms a unique requirement identifier, for example,

R0007: All HANDLES SHALL be unique within one MDIB sequence of a SERVICE PROVIDER.

Contents

1. Overview	21
1.1 Scope	21
1.2 Purpose	21
2. Normative references.....	22
3. Definitions and notational conventions	23
3.1 Definitions	23
3.2 Notational conventions	25
4. Introduction to BICEPS.....	28
5. Participant Model	30
5.1 Overview	30
5.2 Background.....	30
5.3 Descriptive part.....	34
5.4 State part.....	41
6. Alert signal delegation.....	49
6.1 General	49
6.2 Delegable alert signals.....	49
6.3 Fallback alert signals	52
7. Communication Model.....	53
7.1 General	53
7.2 Message exchange patterns.....	53
7.3 Service Model.....	54
7.4 Message Model.....	59
8. Extension Model.....	62
9. Discovery Model	63
9.1 General	63
9.2 Implicit discovery	63
9.3 Explicit discovery	63
10. Nonfunctional requirements	64
10.1 General	64
10.2 Patient safety considerations.....	64
10.3 Cybersecurity considerations.....	64
10.4 Clinical effectiveness.....	65
10.5 Regulatory considerations	65
11. Conformance	66
11.1 Overview	66
11.2 General format.....	66
11.3 ICS tables.....	67
Annex A (normative) Extension Model	70
A.1 Extension	70
A.2 ExtensionType	71

A.3 MustUnderstand	71
Annex B (normative) Participant Model	72
B.1 AbstractAlertDescriptor	74
B.2 AbstractAlertState	74
B.3 AbstractAlertState/@ActivationState	75
B.4 AbstractComplexDeviceComponentDescriptor	75
B.5 AbstractComplexDeviceComponentDescriptor/AlertSystem	76
B.6 AbstractComplexDeviceComponentDescriptor/Sco	77
B.7 AbstractComplexDeviceComponentState	78
B.8 AbstractContextDescriptor	79
B.9 AbstractContextState	80
B.10 AbstractContextState/@ContextAssociation	81
B.11 AbstractContextState/@BindingMdibVersion	81
B.12 AbstractContextState/@UnbindingMdibVersion	81
B.13 AbstractContextState/@BindingStartTime	82
B.14 AbstractContextState/@BindingEndTime	82
B.15 AbstractContextState/Validator	82
B.16 AbstractContextState/Identification	83
B.17 AbstractDescriptor	83
B.18 AbstractDescriptor/@Handle	84
B.19 AbstractDescriptor/@DescriptorVersion	84
B.20 AbstractDescriptor/@SafetyClassification	84
B.21 AbstractDescriptor/Type	85
B.22 AbstractDeviceComponentDescriptor	86
B.23 AbstractDeviceComponentDescriptor/ProductionSpecification	86
B.24 AbstractDeviceComponentDescriptor/ProductionSpecification/SpecType	87
B.25 AbstractDeviceComponentDescriptor/ProductionSpecification/ProductionSpec	88
B.26 AbstractDeviceComponentDescriptor/ProductionSpecification/ComponentId	88
B.27 AbstractDeviceComponentState	89
B.28 AbstractDeviceComponentState/@ActivationState	90
B.29 AbstractDeviceComponentState/@OperatingHours	90
B.30 AbstractDeviceComponentState/@OperatingCycles	90
B.31 AbstractDeviceComponentState/CalibrationInfo	90
B.32 AbstractDeviceComponentState/NextCalibration	91
B.33 AbstractDeviceComponentState/PhysicalConnector	92
B.34 AbstractMetricDescriptor	93
B.35 AbstractMetricDescriptor/@MetricCategory	94
B.36 AbstractMetricDescriptor/@DerivationMethod	94
B.37 AbstractMetricDescriptor/@MetricAvailability	95
B.38 AbstractMetricDescriptor/@MaxMeasurementTime	95
B.39 AbstractMetricDescriptor/@MaxDelayTime	95
B.40 AbstractMetricDescriptor/@DeterminationPeriod	96
B.41 AbstractMetricDescriptor/@LifeTimePeriod	96
B.42 AbstractMetricDescriptor/@ActivationDuration	96
B.43 AbstractMetricDescriptor/Unit	97
B.44 AbstractMetricDescriptor/BodySite	98
B.45 AbstractMetricDescriptor/Relation	99
B.46 AbstractMetricDescriptor/Relation/@Kind	100
B.47 AbstractMetricDescriptor/Relation/@Entries	100
B.48 AbstractMetricDescriptor/Relation/Code	101
B.49 AbstractMetricDescriptor/Relation/Identification	102
B.50 AbstractMetricState	103
B.51 AbstractMetricState/@ActivationState	104
B.52 AbstractMetricState/@ActiveDeterminationPeriod	104
B.53 AbstractMetricState/@LifeTimePeriod	104

B.54 AbstractMetricState/BodySite.....	105
B.55 AbstractMetricState/PhysicalConnector	106
B.56 AbstractMetricValue	106
B.57 AbstractMetricValue/@StartTime	107
B.58 AbstractMetricValue/@StopTime	107
B.59 AbstractMetricValue/@DeterminationTime	107
B.60 AbstractMetricValue/MetricQuality	107
B.61 AbstractMetricValue/MetricQuality/@Validity	108
B.62 AbstractMetricValue/MetricQuality/@Mode	108
B.63 AbstractMetricValue/MetricQuality/@Qi.....	108
B.64 AbstractMetricValue/Annotation	109
B.65 AbstractMetricValue/Annotation/Type.....	109
B.66 AbstractMultiState	110
B.67 AbstractMultiState/@Handle	111
B.68 AbstractMultiState/Category	111
B.69 AbstractOperationDescriptor	112
B.70 AbstractOperationDescriptor/@OperationTarget	113
B.71 AbstractOperationDescriptor/@MaxTimeToFinish.....	113
B.72 AbstractOperationDescriptor/@InvocationEffectiveTimeout.....	113
B.73 AbstractOperationDescriptor/@Retriggerable	113
B.74 AbstractOperationDescriptor/@AccessLevel	114
B.75 AbstractOperationState	114
B.76 AbstractOperationState/@OperatingMode	115
B.77 AbstractSetStateOperationDescriptor	115
B.78 AbstractSetStateOperationDescriptor/ModifiableData	116
B.79 AbstractState	116
B.80 AbstractState/@StateVersion.....	117
B.81 AbstractState/@DescriptorHandle	117
B.82 AbstractState/@DescriptorVersion	117
B.83 ActivateOperationDescriptor	118
B.84 ActivateOperationDescriptor/Argument	119
B.85 ActivateOperationDescriptor/Argument/ArgName	119
B.86 ActivateOperationDescriptor/Argument/Arg.....	120
B.87 ActivateOperationState	120
B.88 AlertConditionDescriptor.....	121
B.89 AlertConditionDescriptor/@Kind	122
B.90 AlertConditionDescriptor/@Priority.....	122
B.91 AlertConditionDescriptor/@DefaultConditionGenerationDelay	122
B.92 AlertConditionDescriptor/@CanEscalate	122
B.93 AlertConditionDescriptor/@CanDeescalate	123
B.94 AlertConditionDescriptor/Source	123
B.95 AlertConditionDescriptor/CauseInfo	123
B.96 AlertConditionState	124
B.97 AlertConditionState/@ActualConditionGenerationDelay	125
B.98 AlertConditionState/@ActualPriority	125
B.99 AlertConditionState/@Rank	125
B.100 AlertConditionState/@Presence.....	125
B.101 AlertConditionState/@DeterminationTime	125
B.102 AlertSignalDescriptor	126
B.103 AlertSignalDescriptor/@ConditionSignaled.....	127
B.104 AlertSignalDescriptor/@Manifestation.....	127
B.105 AlertSignalDescriptor/@Latching	128
B.106 AlertSignalDescriptor/@DefaultSignalGenerationDelay	128
B.107 AlertSignalDescriptor/@MinSignalGenerationDelay.....	128
B.108 AlertSignalDescriptor/@MaxSignalGenerationDelay	128
B.109 AlertSignalDescriptor/@SignalDelegationSupported.....	128

B.110 AlertSignalDescriptor/@AcknowledgementSupported	128
B.111 AlertSignalDescriptor/@AcknowledgeTimeout	129
B.112 AlertSignalState	129
B.113 AlertSignalState/@ActualSignalGenerationDelay	130
B.114 AlertSignalState/@Presence	130
B.115 AlertSignalState/@Location	130
B.116 AlertSignalState/@Slot	130
B.117 AlertSystemDescriptor	131
B.118 AlertSystemDescriptor/@MaxPhysiologicalParallelAlarms	132
B.119 AlertSystemDescriptor/@MaxTechnicalParallelAlarms	132
B.120 AlertSystemDescriptor/@SelfCheckPeriod	132
B.121 AlertSystemDescriptor/AlertCondition	132
B.122 AlertSystemDescriptor/AlertSignal	133
B.123 AlertSystemState	134
B.124 AlertSystemState/@LastSelfCheck	135
B.125 AlertSystemState/@SelfCheckCount	135
B.126 AlertSystemState/@PresentPhysiologicalAlarmConditions	135
B.127 AlertSystemState/@PresentTechnicalAlarmConditions	135
B.128 AlertSystemState/SystemSignalActivation	136
B.129 ApprovedJurisdictions	136
B.130 ApprovedJurisdictions/ApprovedJurisdiction	137
B.131 BaseDemographics	137
B.132 BaseDemographics/Givenname	138
B.133 BaseDemographics/Middlename	138
B.134 BaseDemographics/Familyname	138
B.135 BaseDemographics/Birthname	139
B.136 BaseDemographics/Title	139
B.137 BatteryDescriptor	139
B.138 BatteryDescriptor/CapacityFullCharge	140
B.139 BatteryDescriptor/CapacitySpecified	141
B.140 BatteryDescriptor/VoltageSpecified	141
B.141 BatteryState	142
B.142 BatteryState/@ChargeStatus	143
B.143 BatteryState/@ChargeCycles	143
B.144 BatteryState/CapacityRemaining	144
B.145 BatteryState/Voltage	144
B.146 BatteryState/Current	145
B.147 BatteryState/Temperature	145
B.148 BatteryState/RemainingBatteryTime	146
B.149 CalibrationInfo	146
B.150 CalibrationInfo/@ComponentCalibrationState	147
B.151 CalibrationInfo/@Type	147
B.152 CalibrationInfo/@Time	147
B.153 CalibrationInfo/CalibrationDocumentation	147
B.154 CalibrationInfo/CalibrationDocumentation/Documentation	148
B.155 CalibrationInfo/CalibrationDocumentation/CalibrationResult	148
B.156 CalibrationInfo/CalibrationDocumentation/CalibrationResult/Code	149
B.157 CalibrationInfo/CalibrationDocumentation/CalibrationResult/Value	150
B.158 CauseInfo	150
B.159 CauseInfo/RemedyInfo	150
B.160 CauseInfo/Description	151
B.161 ChannelDescriptor	152
B.162 ChannelDescriptor/Metric	153
B.163 ChannelState	154
B.164 ClinicalInfo	155
B.165 ClinicalInfo/Type	156

B.166 ClinicalInfo/Code.....	157
B.167 ClinicalInfo/Criticality	157
B.168 ClinicalInfo/Description.....	158
B.169 ClinicalInfo/RelatedMeasurement	159
B.170 ClinicalInfo/RelatedMeasurement/@Validity.....	159
B.171 ClinicalInfo/RelatedMeasurement/Value.....	160
B.172 ClinicalInfo/RelatedMeasurement/ReferenceRange	160
B.173 ClinicalInfo/RelatedMeasurement/ReferenceRange/Range	161
B.174 ClinicalInfo/RelatedMeasurement/ReferenceRange/Meaning	162
B.175 ClockDescriptor	163
B.176 ClockDescriptor/@Resolution	163
B.177 ClockDescriptor/TimeProtocol	164
B.178 ClockState	165
B.179 ClockState/@DateAndTime	166
B.180 ClockState/@RemoteSync	166
B.181 ClockState/@Accuracy	166
B.182 ClockState/@LastSet	167
B.183 ClockState/@TimeZone	167
B.184 ClockState/@CriticalUse	167
B.185 ClockState/ActiveSyncProtocol	167
B.186 ClockState/ReferenceSource	168
B.187 CodedValue.....	169
B.188 CodedValue/@Code	170
B.189 CodedValue/@CodingSystem	170
B.190 CodedValue/@CodingSystemVersion	171
B.191 CodedValue/@SymbolicCodeName	171
B.192 CodedValue/CodingSystemName.....	171
B.193 CodedValue/ConceptDescription.....	172
B.194 CodedValue/Translation	172
B.195 CodedValue/Translation/@Code	173
B.196 CodedValue/Translation/@CodingSystem	173
B.197 CodedValue/Translation/@CodingSystemVersion	173
B.198 ContainmentTree.....	173
B.199 ContainmentTree/Entry.....	174
B.200 ContainmentTreeEntry.....	175
B.201 ContainmentTreeEntry/Type.....	176
B.202 DistributionSampleArrayMetricDescriptor.....	177
B.203 DistributionSampleArrayMetricDescriptor/@Resolution.....	178
B.204 DistributionSampleArrayMetricDescriptor/TechnicalRange	179
B.205 DistributionSampleArrayMetricDescriptor/DomainUnit.....	180
B.206 DistributionSampleArrayMetricDescriptor/DistributionRange	181
B.207 DistributionSampleArrayMetricState.....	182
B.208 DistributionSampleArrayMetricState/MetricValue	183
B.209 DistributionSampleArrayMetricState/PhysiologicalRange	184
B.210 EnsembleContextDescriptor	185
B.211 EnsembleContextState	186
B.212 EnumStringMetricDescriptor.....	187
B.213 EnumStringMetricDescriptor/AllowedValue.....	188
B.214 EnumStringMetricDescriptor/AllowedValue/Value	188
B.215 EnumStringMetricDescriptor/AllowedValue/Type	189
B.216 EnumStringMetricDescriptor/AllowedValue/Identification	190
B.217 EnumStringMetricDescriptor/AllowedValue/Characteristic	190
B.218 EnumStringMetricState.....	191
B.219 ImagingProcedure	192
B.220 ImagingProcedure/AccessionIdentifier.....	193
B.221 ImagingProcedure/RequestedProcedureId.....	193

B.222 ImagingProcedure/StudyInstanceUid.....	194
B.223 ImagingProcedure/ScheduledProcedureStepId.....	195
B.224 ImagingProcedure/Modality.....	196
B.225 ImagingProcedure/ProtocolCode.....	197
B.226 InstanceIdentifier.....	198
B.227 InstanceIdentifier/@Root.....	199
B.228 InstanceIdentifier/@Extension.....	199
B.229 InstanceIdentifier/Type.....	200
B.230 InstanceIdentifier/IdentifierName.....	201
B.231 LimitAlertConditionDescriptor.....	202
B.232 LimitAlertConditionDescriptor/@AutoLimitSupported.....	203
B.233 LimitAlertConditionDescriptor/MaxLimits.....	203
B.234 LimitAlertConditionState.....	204
B.235 LimitAlertConditionState/@MonitoredAlertLimits.....	205
B.236 LimitAlertConditionState/@AutoLimitActivationState.....	205
B.237 LimitAlertConditionState/Limits.....	206
B.238 LocalizedText.....	206
B.239 LocalizedText/@Ref.....	207
B.240 LocalizedText/@Lang.....	207
B.241 LocalizedText/@Version.....	208
B.242 LocalizedText/@TextWidth.....	208
B.243 LocationContextDescriptor.....	208
B.244 LocationContextState.....	209
B.245 LocationContextState/LocationDetail.....	210
B.246 LocationDetail.....	211
B.247 LocationDetail/@PoC.....	211
B.248 LocationDetail/@Room.....	211
B.249 LocationDetail/@Bed.....	211
B.250 LocationDetail/@Facility.....	212
B.251 LocationDetail/@Building.....	212
B.252 LocationDetail/@Floor.....	212
B.253 LocationReference.....	212
B.254 LocationReference/Identification.....	213
B.255 LocationReference/LocationDetail.....	214
B.256 MdDescription.....	214
B.257 MdDescription/@DescriptionVersion.....	215
B.258 MdDescription/Mds.....	216
B.259 Mddb.....	217
B.260 Mddb/MdDescription.....	218
B.261 Mddb/MdState.....	218
B.262 MdsDescriptor.....	219
B.263 MdsDescriptor/MetaData.....	220
B.264 MdsDescriptor/MetaData/Udi.....	221
B.265 MdsDescriptor/MetaData/Udi/DeviceIdentifier.....	221
B.266 MdsDescriptor/MetaData/Udi/HumanReadableForm.....	221
B.267 MdsDescriptor/MetaData/Udi/Issuer.....	222
B.268 MdsDescriptor/MetaData/Udi/Jurisdiction.....	222
B.269 MdsDescriptor/MetaData/LotNumber.....	223
B.270 MdsDescriptor/MetaData/Manufacturer.....	223
B.271 MdsDescriptor/MetaData/ManufactureDate.....	224
B.272 MdsDescriptor/MetaData/ExpirationDate.....	224
B.273 MdsDescriptor/MetaData/ModelName.....	224
B.274 MdsDescriptor/MetaData/ModelNumber.....	225
B.275 MdsDescriptor/MetaData/SerialNumber.....	225
B.276 MdsDescriptor/SystemContext.....	226
B.277 MdsDescriptor/Clock.....	227

B.278 MdsDescriptor/Battery	228
B.279 MdsDescriptor/ApprovedJurisdictions	229
B.280 MdsDescriptor/Vmd	229
B.281 MdsState	230
B.282 MdsState/@Lang	231
B.283 MdsState/@OperatingMode	231
B.284 MdsState/OperatingJurisdiction	232
B.285 MdState	232
B.286 MdState/@StateVersion	233
B.287 MdState/State	233
B.288 MeansContextDescriptor	234
B.289 MeansContextState	235
B.290 Measurement	236
B.291 Measurement/@MeasuredValue	236
B.292 Measurement/MeasurementUnit	237
B.293 NeonatalPatientDemographicsCoreData	238
B.294 NeonatalPatientDemographicsCoreData/GestationalAge	239
B.295 NeonatalPatientDemographicsCoreData/BirthLength	239
B.296 NeonatalPatientDemographicsCoreData/BirthWeight	240
B.297 NeonatalPatientDemographicsCoreData/HeadCircumference	240
B.298 NeonatalPatientDemographicsCoreData/Mother	241
B.299 NumericMetricDescriptor	242
B.300 NumericMetricDescriptor/@Resolution	243
B.301 NumericMetricDescriptor/@AveragingPeriod	243
B.302 NumericMetricDescriptor/TechnicalRange	244
B.303 NumericMetricState	245
B.304 NumericMetricState/@ActiveAveragingPeriod	246
B.305 NumericMetricState/MetricValue	246
B.306 NumericMetricState/PhysiologicalRange	247
B.307 NumericMetricValue	248
B.308 NumericMetricValue/@Value	248
B.309 OperatingJurisdiction	249
B.310 OperatorContextDescriptor	249
B.311 OperatorContextState	250
B.312 OperatorContextState/OperatorDetails	251
B.313 OrderDetail	252
B.314 OrderDetail/Start	252
B.315 OrderDetail/End	252
B.316 OrderDetail/Performer	253
B.317 OrderDetail/Service	254
B.318 OrderDetail/ImagingProcedure	255
B.319 PatientContextDescriptor	255
B.320 PatientContextState	256
B.321 PatientContextState/CoreData	257
B.322 PatientDemographicsCoreData	258
B.323 PatientDemographicsCoreData/Sex	259
B.324 PatientDemographicsCoreData/PatientType	259
B.325 PatientDemographicsCoreData/DateOfBirth	260
B.326 PatientDemographicsCoreData/Height	260
B.327 PatientDemographicsCoreData/Weight	261
B.328 PatientDemographicsCoreData/Race	261
B.329 PersonParticipation	262
B.330 PersonParticipation/Role	263
B.331 PersonReference	264
B.332 PersonReference/Identification	264
B.333 PersonReference/Name	265

B.334 PhysicalConnectorInfo	265
B.335 PhysicalConnectorInfo/@Number	266
B.336 PhysicalConnectorInfo/Label	266
B.337 Range	267
B.338 Range/@Lower	267
B.339 Range/@Upper	268
B.340 Range/@StepWidth	268
B.341 Range/@RelativeAccuracy	268
B.342 Range/@AbsoluteAccuracy	268
B.343 RealTimeSampleArrayMetricDescriptor	269
B.344 RealTimeSampleArrayMetricDescriptor/@Resolution	270
B.345 RealTimeSampleArrayMetricDescriptor/@SamplePeriod	270
B.346 RealTimeSampleArrayMetricDescriptor/TechnicalRange	271
B.347 RealTimeSampleArrayMetricState	272
B.348 RealTimeSampleArrayMetricState/MetricValue	273
B.349 RealTimeSampleArrayMetricState/PhysiologicalRange	274
B.350 RemedyInfo	274
B.351 RemedyInfo/Description	275
B.352 SampleArrayValue	276
B.353 SampleArrayValue/@Samples	277
B.354 SampleArrayValue/ApplyAnnotation	277
B.355 SampleArrayValue/ApplyAnnotation/@AnnotationIndex	277
B.356 SampleArrayValue/ApplyAnnotation/@SampleIndex	277
B.357 ScoDescriptor	278
B.358 ScoDescriptor/Operation	279
B.359 ScoState	280
B.360 ScoState/@InvocationRequested	281
B.361 ScoState/@InvocationRequired	281
B.362 ScoState/OperationGroup	281
B.363 ScoState/OperationGroup/@OperatingMode	281
B.364 ScoState/OperationGroup/@Operations	282
B.365 ScoState/OperationGroup/Type	282
B.366 SetAlertStateOperationDescriptor	283
B.367 SetAlertStateOperationState	284
B.368 SetComponentStateOperationDescriptor	285
B.369 SetComponentStateOperationState	286
B.370 SetContextStateOperationDescriptor	287
B.371 SetContextStateOperationState	288
B.372 SetMetricStateOperationDescriptor	289
B.373 SetMetricStateOperationState	290
B.374 SetStringOperationDescriptor	291
B.375 SetStringOperationDescriptor/@MaxLength	292
B.376 SetStringOperationState	292
B.377 SetStringOperationState/AllowedValues	292
B.378 SetStringOperationState/AllowedValues/Value	293
B.379 SetValueOperationDescriptor	293
B.380 SetValueOperationState	294
B.381 SetValueOperationState/AllowedRange	295
B.382 StringMetricDescriptor	296
B.383 StringMetricState	297
B.384 StringMetricState/MetricValue	298
B.385 StringMetricValue	299
B.386 StringMetricValue/@Value	299
B.387 SystemContextDescriptor	300
B.388 SystemContextDescriptor/PatientContext	301
B.389 SystemContextDescriptor/LocationContext	302

B.390 SystemContextDescriptor/EnsembleContext	302
B.391 SystemContextDescriptor/OperatorContext	303
B.392 SystemContextDescriptor/WorkflowContext	304
B.393 SystemContextDescriptor/MeansContext	304
B.394 SystemContextState	305
B.395 SystemSignalActivation	306
B.396 SystemSignalActivation/@Manifestation	306
B.397 SystemSignalActivation/@State	307
B.398 VmdDescriptor	307
B.399 VmdDescriptor/ApprovedJurisdictions	308
B.400 VmdDescriptor/Channel	308
B.401 VmdState	309
B.402 VmdState/OperatingJurisdiction	310
B.403 WorkflowContextDescriptor	311
B.404 WorkflowContextState	312
B.405 WorkflowContextState/WorkflowDetail	313
B.406 WorkflowContextState/WorkflowDetail/Patient	314
B.407 WorkflowContextState/WorkflowDetail/AssignedLocation	314
B.408 WorkflowContextState/WorkflowDetail/VisitNumber	315
B.409 WorkflowContextState/WorkflowDetail/DangerCode	316
B.410 WorkflowContextState/WorkflowDetail/RelevantClinicalInfo	317
B.411 WorkflowContextState/WorkflowDetail/RequestedOrderDetail	318
B.412 WorkflowContextState/WorkflowDetail/RequestedOrderDetail/ ReferringPhysician	319
B.413 WorkflowContextState/WorkflowDetail/RequestedOrderDetail/ RequestingPhysician	319
B.414 WorkflowContextState/WorkflowDetail/RequestedOrderDetail/ PlacerOrderNumber	320
B.415 WorkflowContextState/WorkflowDetail/PerformedOrderDetail	321
B.416 WorkflowContextState/WorkflowDetail/PerformedOrderDetail/ FillerOrderNumber	322
B.417 WorkflowContextState/WorkflowDetail/PerformedOrderDetail/ ResultingClinicalInfo	323
B.418 AlertActivation	323
B.419 AlertConditionKind	324
B.420 AlertConditionMonitoredLimits	324
B.421 AlertConditionPriority	324
B.422 AlertConditionReference	325
B.423 AlertSignalManifestation	325
B.424 AlertSignalPresence	325
B.425 AlertSignalPrimaryLocation	326
B.426 CalibrationState	326
B.427 CalibrationType	326
B.428 CodeIdentifier	326
B.429 ComponentActivation	327
B.430 ContextAssociation	327
B.431 DerivationMethod	327
B.432 EntryRef	328
B.433 GenerationMode	328
B.434 Handle	328
B.435 HandleRef	328
B.436 LocalizedTextContent	329
B.437 LocalizedTextRef	329
B.438 LocalizedTextWidth	329
B.439 MdsOperatingMode	330
B.440 MeasurementValidity	330
B.441 MetricAvailability	331
B.442 MetricCategory	331
B.443 OperatingMode	331
B.444 OperationRef	332
B.445 PatientType	332

B.446 QualityIndicator	332
B.447 RealTimeValueType	333
B.448 ReferencedVersion	333
B.449 SafetyClassification	333
B.450 Sex	334
B.451 SymbolicCodeName	334
B.452 Timestamp	334
B.453 TimeZone	335
B.454 VersionCounter	335
B.455 ContainmentTreeInfo	335
B.456 ContainmentTreeInfo/@HandleRef	335
B.457 ContainmentTreeInfo/@ParentHandleRef	336
B.458 ContainmentTreeInfo/@EntryType	336
B.459 ContainmentTreeInfo/@ChildrenCount	336
B.460 MddbVersionGroup	336
B.461 MddbVersionGroup/@MddbVersion	336
B.462 MddbVersionGroup/@SequenceId	337
B.463 MddbVersionGroup/@InstanceId	337
Annex C (normative) Message Model	338
C.1 Activate	339
C.2 Activate/Argument	339
C.3 Activate/Argument/ArgValue	339
C.4 ActivateResponse	340
C.5 DescriptionModificationReport	340
C.6 DescriptionModificationReport/ReportPart	341
C.7 DescriptionModificationReport/ReportPart/@ParentDescriptor	342
C.8 DescriptionModificationReport/ReportPart/@ModificationType	342
C.9 DescriptionModificationReport/ReportPart/Descriptor	342
C.10 DescriptionModificationReport/ReportPart/State	343
C.11 EpisodicAlertReport	344
C.12 EpisodicComponentReport	344
C.13 EpisodicContextReport	345
C.14 EpisodicMetricReport	346
C.15 EpisodicOperationalStateReport	346
C.16 GetContainmentTree	347
C.17 GetContainmentTree/HandleRef	347
C.18 GetContainmentTreeResponse	348
C.19 GetContainmentTreeResponse/ContainmentTree	348
C.20 GetContextStates	349
C.21 GetContextStates/HandleRef	349
C.22 GetContextStatesByFilter	350
C.23 GetContextStatesByFilter/@ContextType	350
C.24 GetContextStatesByFilter/Filter	350
C.25 GetContextStatesByFilterResponse	351
C.26 GetContextStatesByFilterResponse/ContextState	352
C.27 GetContextStatesByIdentification	353
C.28 GetContextStatesByIdentification/@ContextType	353
C.29 GetContextStatesByIdentification/Identification	354
C.30 GetContextStatesByIdentificationResponse	355
C.31 GetContextStatesByIdentificationResponse/ContextState	356
C.32 GetContextStatesResponse	357
C.33 GetContextStatesResponse/ContextState	358
C.34 GetDescriptor	359
C.35 GetDescriptor/HandleRef	359
C.36 GetDescriptorResponse	360

C.37 GetDescriptorResponse/Descriptor	360
C.38 GetDescriptorsFromArchive	361
C.39 GetDescriptorsFromArchive/DescriptorRevisions	362
C.40 GetDescriptorsFromArchive/TimeFrame	362
C.41 GetDescriptorsFromArchive/Handle	363
C.42 GetDescriptorsFromArchiveResponse	363
C.43 GetDescriptorsFromArchiveResponse/Descriptor	364
C.44 GetLocalizedText	365
C.45 GetLocalizedText/Ref	365
C.46 GetLocalizedText/Version	366
C.47 GetLocalizedText/Lang	366
C.48 GetLocalizedText/TextWidth	366
C.49 GetLocalizedText/NumberOfLines	367
C.50 GetLocalizedTextResponse	367
C.51 GetLocalizedTextResponse/Text	368
C.52 GetMdDescription	369
C.53 GetMdDescription/HandleRef	369
C.54 GetMdDescriptionResponse	369
C.55 GetMdDescriptionResponse/MdDescription	370
C.56 GetMdib	370
C.57 GetMdibResponse	371
C.58 GetMdibResponse/Mdib	371
C.59 GetMdState	372
C.60 GetMdState/HandleRef	372
C.61 GetMdStateResponse	373
C.62 GetMdStateResponse/MdState	373
C.63 GetStatesFromArchive	374
C.64 GetStatesFromArchive/StateRevisions	374
C.65 GetStatesFromArchive/TimeFrame	375
C.66 GetStatesFromArchive/Handle	375
C.67 GetStatesFromArchiveResponse	376
C.68 GetStatesFromArchiveResponse/State	377
C.69 GetSupportedLanguages	377
C.70 GetSupportedLanguagesResponse	378
C.71 GetSupportedLanguagesResponse/Lang	378
C.72 ObservedValueStream	379
C.73 ObservedValueStream/Value	379
C.74 ObservedValueStream/Value/@Metric	380
C.75 ObservedValueStream/Value/@StateVersion	380
C.76 ObservedValueStream/Value/Value	380
C.77 OperationInvokedReport	381
C.78 OperationInvokedReport/ReportPart	382
C.79 OperationInvokedReport/ReportPart/@OperationHandleRef	382
C.80 OperationInvokedReport/ReportPart/@OperationTarget	382
C.81 OperationInvokedReport/ReportPart/InvocationInfo	383
C.82 OperationInvokedReport/ReportPart/InvocationSource	383
C.83 PeriodicAlertReport	384
C.84 PeriodicComponentReport	385
C.85 PeriodicContextReport	385
C.86 PeriodicMetricReport	386
C.87 PeriodicOperationalStateReport	387
C.88 Retrievability	387
C.89 Retrievability/By	388
C.90 SetAlertState	388
C.91 SetAlertState/ProposedAlertState	389
C.92 SetAlertStateResponse	389

C.93 SetComponentState	390
C.94 SetComponentState/ProposedComponentState	391
C.95 SetComponentStateResponse	392
C.96 SetContextState	392
C.97 SetContextState/ProposedContextState	393
C.98 SetContextStateResponse	394
C.99 SetMetricState	394
C.100 SetMetricState/ProposedMetricState	395
C.101 SetMetricStateResponse	396
C.102 SetString	396
C.103 SetString/RequestedStringValue	397
C.104 SetStringResponse	397
C.105 SetValue	397
C.106 SetValue/RequestedNumericValue	398
C.107 SetValueResponse	398
C.108 SystemErrorReport	399
C.109 SystemErrorReport/ReportPart	399
C.110 SystemErrorReport/ReportPart/ErrorCode	400
C.111 SystemErrorReport/ReportPart/ErrorInfo	401
C.112 WaveformStream	401
C.113 WaveformStream/State	402
C.114 AbstractAlertReport	403
C.115 AbstractAlertReport/ReportPart	404
C.116 AbstractAlertReport/ReportPart/AlertState	404
C.117 AbstractComponentReport	405
C.118 AbstractComponentReport/ReportPart	405
C.119 AbstractComponentReport/ReportPart/ComponentState	406
C.120 AbstractContextReport	407
C.121 AbstractContextReport/ReportPart	407
C.122 AbstractContextReport/ReportPart/ContextState	408
C.123 AbstractGet	409
C.124 AbstractGetResponse	409
C.125 AbstractMetricReport	410
C.126 AbstractMetricReport/ReportPart	411
C.127 AbstractMetricReport/ReportPart/MetricState	411
C.128 AbstractOperationalStateReport	412
C.129 AbstractOperationalStateReport/ReportPart	413
C.130 AbstractOperationalStateReport/ReportPart/OperationState	413
C.131 AbstractReport	414
C.132 AbstractReportPart	414
C.133 AbstractReportPart/SourceMds	415
C.134 AbstractSet	415
C.135 AbstractSet/OperationHandleRef	415
C.136 AbstractSetResponse	416
C.137 AbstractSetResponse/InvocationInfo	416
C.138 InvocationInfo	417
C.139 InvocationInfo/TransactionId	417
C.140 InvocationInfo/InvocationState	418
C.141 InvocationInfo/InvocationError	418
C.142 InvocationInfo/InvocationErrorMessage	419
C.143 RetrievabilityInfo	419
C.144 RetrievabilityInfo/@Method	420
C.145 RetrievabilityInfo/@UpdatePeriod	420
C.146 TimeFrame	420
C.147 TimeFrame/@Start	420
C.148 TimeFrame/@End	421

C.149 VersionFrame.....	421
C.150 VersionFrame/@Start	421
C.151 VersionFrame/@End.....	421
C.152 DescriptionModificationType	422
C.153 InvocationError	422
C.154 InvocationState	423
C.155 RetrievabilityMethod	423
C.156 TransactionId	423
Annex D (informative) Examples.....	424
D.1 Containment tree.....	424
D.2 MDIB versioning.....	424
D.3 MDIB sequencing.....	429
D.4 BICEPS discovery	430
D.5 Remote control.....	432
D.6 Alert signal generation.....	433
Annex E (informative) Bibliography.....	435

Health informatics—Point-of-care medical device communication

Part 10207: Domain Information and Service Model for Service-Oriented Point-of-Care Medical Device Communication

1. Overview

1.1 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 devices and medical information technology (IT) systems in which medical data needs to be exchanged or networked point-of-care medical devices need to be controlled. The standard provides a Participant Model and Communication Model derived from the IEEE 11073™ Domain Information Model (IEEE 11073-10201™ DIM). Furthermore, it utilizes the IEEE 11073 Nomenclature (IEEE 11073-10101™) and supports other coding systems to convey the semantics of any information elements.¹

The definition of network transport mechanisms is outside the scope of this standard.

1.2 Purpose

The purpose of this standard is to enable vendors of point-of-care medical devices and medical IT systems to constitute networked medical device applications from the perspective of domain experts. Therefore, this standard provides tools to virtually model point-of-care medical devices. Moreover, it defines a service-oriented architecture to access those networked-based point-of-care medical devices.

¹ Information on normative references can be found in Clause 2.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used; therefore, 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.

IEC 60050-482:2004, International Electrotechnical Vocabulary – Part 482: Primary and secondary cells and batteries.²

IEEE Std 11073-10101a™-2015, Health informatics—Point-of-care medical device communication—Part 10101: Nomenclature, Amendment 1: Additional Definitions.^{3,4}

ISO/IEEE 11073-10101:2004, Health informatics — Point-of-care medical device communication — Part 10101: Nomenclature.⁵

ISO/IEEE 11073-10201:2004, Health informatics — Point-of-care medical device communication — Part 10201: Domain information model.

² IEC publications are available from the International Electrotechnical Commission (<http://www.iec.ch>) and the American National Standards Institute (<http://www.ansi.org/>).

³ The IEEE standards or products referred to in Clause 2 are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

⁴ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁵ ISO publications are available from the International Organization for Standardization (<http://www.iso.org/>) and the American National Standards Institute (<http://www.ansi.org/>).

3. Definitions and notational conventions

3.1 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.⁶ Any word in capital letters refers to the technical use defined in this clause. In any other case, no special spelling is given.

ALERT CONDITION: State of the ALERT SYSTEM when it has determined that a potential or actual HAZARDOUS SITUATION exists for which OPERATOR awareness or response is required.

NOTE—See also IEC 60601-1-8 (2014) [B1].^{7,8}

ALERT SIGNAL DELEGATION: Capability of a POC MEDICAL DEVICE to let another PARTICIPANT generate a POC MEDICAL DEVICE's ALERT SIGNAL as primary ALERT SIGNAL in order to indicate the presence of an ALERT CONDITION on the POC MEDICAL DEVICE.

ALERT SIGNAL: Type of signal generated by the ALERT SYSTEM to indicate the presence (or occurrence) of an ALERT CONDITION.

NOTE—See also IEC 60601-1-8 (2014) [B1].

ALERT SYSTEM: Part of a POC MEDICAL DEVICE that detects ALERT CONDITIONS and, as appropriate, generates ALERT SIGNALS.

NOTE—See also IEC 60601-1-8 (2014) [B1].

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

ATTRIBUTE: XML attribute as defined in eXtensible Markup Language .

NOTE—See XML [B5].

BASIC INTEGRATED CLINICAL ENVIRONMENT PROTOCOL SPECIFICATION (BICEPS): Non-normative name of IEEE Std 11073-10207.

BICEPS BINDING: Specification that defines how to connect BICEPS to a transport protocol layer.

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

CODE: Identifier used to semantically describe an entity within a CODING SYSTEM.

CODED VALUE: Value that utilizes a CODING SYSTEM and a CODE to semantically describe a certain object within the MEDICAL DATA INFORMATION BASE (MDIB). In general, CODED VALUES are based on standardized terminologies to increase interoperability.

CODING SYSTEM: Set of CODEs.

⁶ *IEEE Standards Dictionary Online* is available at <http://dictionary.ieee.org>.

⁷ The numbers in brackets correspond to the numbers of the bibliography in Annex E.

⁸ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

CONTAINMENT TREE ENTRY: Node of the CONTAINMENT TREE that encloses descriptive and state information.

CONTAINMENT TREE: Device configuration and capability description of an MEDICAL DEVICE SYSTEM (MDS) that represents a POC MEDICAL DEVICE and is conceivable as a four-layered tree.

CONTEXT: Abstraction of a component of a POC MEDICAL DEVICE that defines the relationship of a POC MEDICAL DEVICE with its usage environment. Technically, a CONTEXT can be understood as a token that is shared between two or more POC MEDICAL DEVICES to let them know they are “talking about the same things.”

DELEGABLE ALERT SIGNAL: ALERT SIGNAL that takes part in the ALERT SIGNAL DELEGATION process. It is controlled by a PARTICIPANT.

DEVICE COMPONENT: Term used to describe the characteristics of a medical-related component of a POC MEDICAL DEVICE. A DEVICE COMPONENT can be a physical component that is integrated inside the device, a removable physical component, or a nonphysical component that allows physiological measurement data and its derived data to be grouped in a hierarchical information organization.

NOTE—See also pm:AbstractDeviceComponentDescriptor and pm:AbstractComplexDeviceComponentDescriptor.

ELEMENT: XML element as defined in eXtensible Markup Language.

NOTE—See XML [B5].

FALLBACK ALERT SIGNAL: ALERT SIGNAL that is emitted by a POC MEDICAL DEVICE when a DELEGABLE ALERT SIGNAL is not confirmed by the generating PARTICIPANT in time, i.e., it exceeds a confirmation timeout.

HANDLE: Term used to uniquely identify a certain object within the MEDICAL DATA INFORMATION BASE (MDIB) without designating any semantics.

HAZARDOUS SITUATION: Situation when people are exposed to a hazard or when property or the environment is threatened. A HAZARDOUS SITUATION exists when a vulnerable entity is exposed to a hazard.

NOTE—See also ISO 14971:2007 [B2].

MEDICAL DATA INFORMATION BASE (MDIB): Structured collection of any data objects that are provided by a particular POC MEDICAL DEVICE. MDIB includes descriptive and state information.

MEDICAL DEVICE SYSTEM (MDS): Abstraction of a system comprising one or more medical functions. In the context of IEEE Std 11073-10207, the term is specifically used as an object-oriented abstraction of a POC MEDICAL DEVICE that provides medical information in the form of information objects that are defined in IEEE Std 11073-10207.

MESSAGE: Set of data in a specific format that is exchanged between two PARTICIPANTS.

METRIC: Abstraction of a component of a POC MEDICAL DEVICE that is able to generate or store direct and derived, quantitative and qualitative biosignal measurements, settings, and status values.

PARTICIPANT: Any network node that is part of a SERVICE-ORIENTED MEDICAL DEVICE SYSTEM (SOMDS) and exchanges information by means of a service-oriented architecture. A PARTICIPANT can be either a SERVICE PROVIDER or a SERVICE CONSUMER.

POC ENVIRONMENT: Environment encompassing a particular diagnostic, bed, or treatment area that is specific to one patient and usually including the systems and personnel that are involved in the acute monitoring and treatment of the patient.

POC MEDICAL DEVICE: Point of care (POC) medical device that directly interacts with, monitors, provides treatment to, or is in some way associated with a single patient. For IEEE Std 11073-10207, the scope of POC MEDICAL DEVICES is further limited to patient-connected medical devices that provide support for electronic communication.

RESPONSIBLE ORGANIZATION: Entity that is accountable for the use and maintenance of a POC MEDICAL DEVICE.

SERVICE CONSUMER: Participant that utilizes at least one SERVICE.

SERVICE INTERFACE: Set of SERVICE OPERATIONS that is provided by a SERVICE to describe its capabilities.

SERVICE OPERATION: Single operation that can be executed remotely at a SERVICE PROVIDER.

SERVICE PERSONNEL: Individuals or entity accountable to the RESPONSIBLE ORGANIZATION for installing, assembling, maintaining, or repairing a POC MEDICAL DEVICE.

SERVICE PROVIDER: Participant that provides at least one SERVICE. A SERVICE PROVIDER can be hosted on the POC MEDICAL DEVICE itself or on an external appliance (e.g., converter box) or information technology (IT) system (e.g., aggregator).

SERVICE: Part of a software system in a SERVICE-ORIENTED MEDICAL DEVICE SYSTEM (SOMDS) that exposes functional capabilities on a communication backbone.

SERVICE-ORIENTED MEDICAL DEVICE SYSTEM (SOMDS): Instance of a distributed system that implements a service-oriented architecture composed of SERVICE PROVIDERS and SERVICE CONSUMERS as defined for IEEE Std 11073-10207.

TYPE: Complex or simple type definition as defined in eXtensible Markup Language Schema.

NOTE—See XML Schema [B8].

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

3.2 Notational conventions

3.2.1 General

This standard defines a Participant Model and a Communication Model that comprises a Message Model, Service Model, and Discovery Model. Both the Participant Model and Communication Model can be extended using the mechanisms defined in the Extension Model.

Class diagrams, sequence diagrams, component diagrams, and state machine diagrams follow the Unified Modeling Language (UML) notation. Beyond UML diagrams, the following subclauses describe notational conventions used to describe and reference XML Schema types.

3.2.2 XML Schema namespaces

Annex A, Annex B, and Annex C include an XML Schema [B8] to describe an Extension Model, the object hierarchy, and ATTRIBUTES of an MDIB instance as well as a set of MESSAGE structures that is used to request MDIB objects. The MDIB object hierarchy and ATTRIBUTES are also known as the Participant Model. The set of MESSAGE structures is also known as the Message Model. All models possess dedicated XML namespaces (see XML Namespaces [B6]). These are defined in Table 1.

Table 1—Namespace mapping of Extension, Participant, and Message Models

Namespace prefix	URI	Model part
ext	http://standards.ieee.org/downloads/11073/11073-10207-2017/extension	Extension Model
pm	http://standards.ieee.org/downloads/11073/11073-10207-2017/participant	Participant Model
msg	http://standards.ieee.org/downloads/11073/11073-10207-2017/message	Message Model

This standard also uses the XML namespace prefix “xsd” to point to the XML Schema namespace specified in XML Schema [B8]:

`http://www.w3.org/2001/XMLSchema`

3.2.3 XML Schema referencing

References to TYPE or ELEMENT definitions of the XML Schema are made by using the corresponding QName (see XML Namespaces [B6]). To reference a nested TYPE definition, the QName is enhanced with a chain of QNames and unqualified ATTRIBUTE names, each separated by XPath delimiters (see XPath [B7]). Examples:

- `pm:Handle` points to the *Handle* TYPE defined in the Participant Model.
- `msg:OperationInvokedReport/msg:ReportPart/@OperationTarget` points to the ATTRIBUTE *OperationTarget* that is defined as part of the *ReportPart*, which is included in the *OperationInvokedReport*. Since the QName prefix is `msg`, the referenced TYPE refers to the Message Model.
- `./@ActivationDuration` refers to an ATTRIBUTE within an arbitrary TYPE definition.

Throughout the XML Schema, a camel case notation is used:

- Every artifact begins with a capital letter (e.g., “Handle”)
- Names are separated by a capital letter (e.g., “ActivationState”)
- Abbreviations are denoted in camel case (e.g., “VmdDescriptor”)

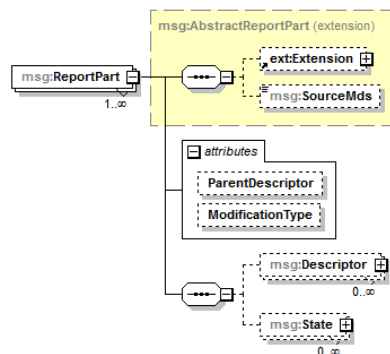
3.2.4 XML Schema type description

Any XML Schema type that is defined in Annex A, Annex B, and Annex C, follows a certain layout. The subclause title corresponds to the type name. Beneath the title the XML Schema type is defined, which is in accordance to XML Schema either “element,” “attribute,” “simpleType,” or “complexType.” Every type is explained in more detail by a properties table (see Table 2).

Table 2—Description of XML Schema type properties

Property name	Description
Type	Type name, including inheritance information if applicable.
Children	List of child elements of the XML Schema type.
Used by	A list of elements and complex types in which the XML Schema type is used.
Attributes	List of attributes defined by the XML Schema type.
Documentation	General description of the data type.
Constraints	Special restrictions to contents, e.g., valid values for enumerations.
Properties	If given, the minimum and maximum occurrence of the XML Schema type.
Use	Usage: “opt” for optional, “req” for required.

Beyond those properties, element and complex type definitions include a picture that describes the structure of the data type graphically. An example is given in Figure 1. The box on the left side describes the data type name and its cardinality. Connected by a solid line are the super types in a yellow box, followed by the attributes and child elements. There are two kinds of boxes. Solid lined boxes are required data fields and dotted lined boxes are optional data fields. If a child element count has a particular range, this range is noted as a cardinality at the bottom right of the box. A plus sign at the right of a solid-lined or dotted box signalizes the existence of child element or attribute definitions that are not shown for sake of brevity.

**Figure 1—Example XML Schema element type structure**

4. Introduction to BICEPS

BICEPS comprises three major parts: a Participant Model, a Communication Model, a Discovery Model as well as nonfunctional requirements. Altogether, these parts are also known as BICEPS. Figure 2 illustrates a decomposition of BICEPS and brings them into logical relation. The diagram shows two perspectives: from the medical domain, known as “Domain Perspective,” and from the IT network, known as “Network Perspective.” The Domain Perspective deals with modeling of medical device systems from a domain point of view, whereas the Network Perspective deals with exchanging medical device information in an abstract communication backbone.

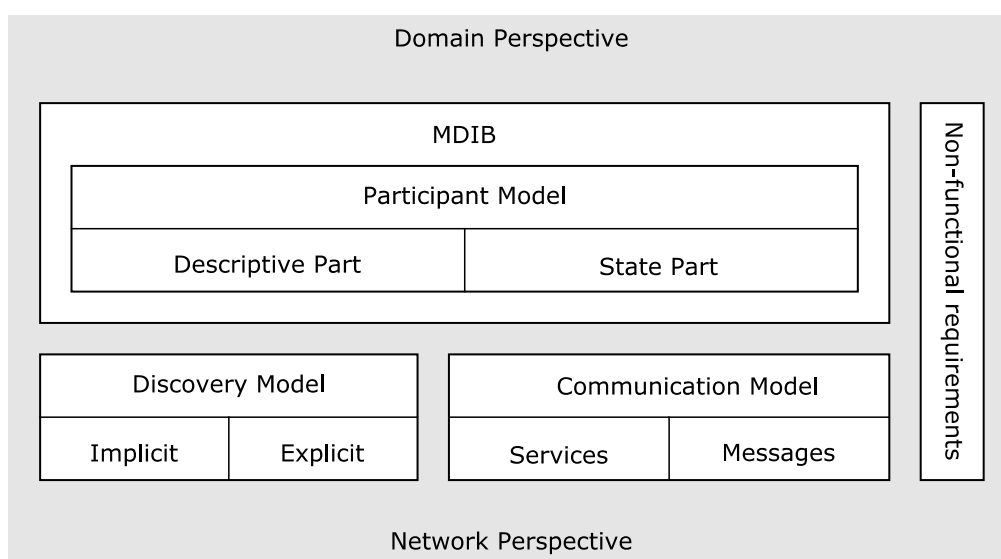


Figure 2—BICEPS component decomposition

The domain perspective pertains to the MDIB, which includes the Participant Model (see Clause 5). The Participant Model consists of a descriptive (mainly static) and a state part (volatile). By defining the MDIB, a SERVICE PROVIDER is able to define a virtual representation of its measurements, settings, alert systems, and contextual information. The Participant Model does not define a communication MESSAGE or ELEMENT for every possible kind of measurement data, setting data, contextual information or remote invocation command, but rather provides an extensibility mechanism that makes it possible for a PARTICIPANT to convey additional data in a MESSAGE or to transmit a completely new type of MESSAGE.

The Participant Model is closely related to the 11073 Domain Information Model (IEEE 11073-10201 DIM), but it is not an exact copy of it. As the Participant Model has been designed with the idea in mind to specify only the essential data structures and 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 are used to be included in the Participant Model. Those packages as well as those that have not been included into BICEPS are shown in Table 3. Whenever this specification reuses a term from IEEE 11073-10201, it is considered semantically the same, though described in another notation than, e.g., ASN.1.

Table 3—IEEE 11073-10201 DIM packages that are in scope and out of scope of BICEPS

IEEE 11073-10201 DIM packages	Covered by BICEPS?
Medical	yes
Alert	yes
Control	yes
System	yes
Patient	yes
Archival	yes
Extended Services	no
Communication	no

The network perspective pertains to the Communication Model (see Clause 7) and Discovery Model (see Clause 9). Both conflate into a SOMDS to provide service discovery, data exchange and remote control between PARTICIPANTS.

Access to the MDIB is established by using a set of SERVICE OPERATIONS. These operations can be used by SERVICE CONSUMERS to exchange MESSAGES with SERVICE PROVIDERS by either requesting them explicitly or subscribing to events that occur on a SERVICE PROVIDER side.

In order to support plug-and-play in a SOMDS, the SERVICES of a SERVICE PROVIDER have to be discovered by potential SERVICE CONSUMERS. The BICEPS Discovery Model defines two discovery modes:

- Explicit discovery
- Implicit discovery

Explicit discovery is based on search messages that are sent to a group of network nodes and that might contain matching criteria of what a SERVICE CONSUMER is interested in. Moreover, plug-and-play is only possible if a suitable data transport foundation is defined. The discovery elements in this standard shape the requirements to a transport layer, only.

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.

BICEPS also defines a set of nonfunctional requirements that can be used to specify the SERVICE PROVIDER's requirements with regard to patient safety, cybersecurity, and clinical effectiveness as well as regulatory requirements (see Clause 10).

5. Participant Model

5.1 Overview

The scope of this clause is the definition of a model that allows the description of the capabilities as well as the expression of the current state of a POC MEDICAL DEVICE, which acts as a PARTICIPANT in a SOMDS. The complete model is defined in Annex B.

5.2 Background

5.2.1 General

BICEPS makes use of the MDIB concept as defined in IEEE 11073-10201 (Clause 6). The MDIB represents a structured collection of instances of managed medical objects representing, e.g., the vital signs information provided by a particular medical device. BICEPS defines the MDIB as a twofold database: the descriptive part and the state part (see Figure 3). The descriptive part comprises a hierarchy of descriptors to establish relationships between managed medical objects. Moreover, it includes the mostly static attributes that are expressed by a POC MEDICAL DEVICE. The state part is the dynamic counterpart of the descriptive part in the MDIB hierarchy. It includes volatile information (like observed measurement values) in form of a flat list of elements corresponding to the objects in the description.

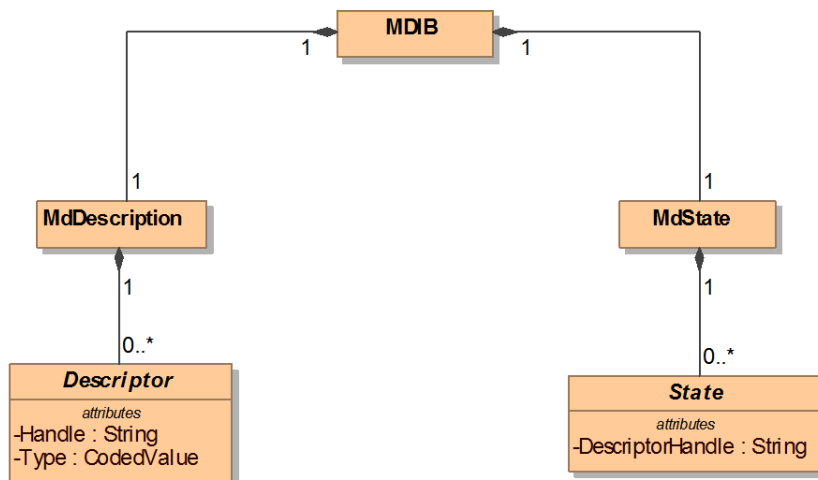


Figure 3—Descriptive and state part of the MDIB
(Every descriptor is referenced by DescriptorHandle in State.)

5.2.2 Handle

Any pm:AbstractDescriptor or pm:AbstractState that is included in an MDIB instance, is identifiable by a HANDLE (also see pm:Handle) or reference to an object with a HANDLE (see pm:HandleRef). It allows for explicit identification.

R0007: All HANDLES SHALL be unique within one MDIB sequence of a SERVICE PROVIDER.

NOTE—The MDIB sequence is defined in 5.2.5.6.

R0105: A HANDLE SHALL consist of characters that match only valid Unicode codes greater than U+0020, except for U+FFFD (replacement character).

R0098: If a SERVICE PROVIDER removes and reinserts a CONTAINMENT TREE ENTRY from / into the CONTAINMENT TREE within one MDIB sequence, and the CONTAINMENT TREE ENTRY has the same HANDLE, it SHALL be of the same XML Schema datatype.

R0137: A SERVICE PROVIDER SHALL NOT change the HANDLE of a CONTAINMENT TREE ENTRY within one MDIB sequence.

R0138: If a SERVICE PROVIDER removes a CONTAINMENT TREE ENTRY from the CONTAINMENT TREE and reinserts the same CONTAINMENT TREE ENTRY into the CONTAINMENT TREE within one MDIB sequence, the CONTAINMENT TREE ENTRY SHALL have the same handle.

NOTE—From R0098 it follows that a CONTAINMENT TREE ENTRY that disappears and appears again can change any of its ATTRIBUTES and ELEMENTs except for the XML Schema datatype.

R0099: If a SERVICE PROVIDER removes and reinserts the same CONTAINMENT TREE ENTRY of an element in the CONTAINMENT TREE beyond one MDIB sequence, it SHOULD use the same HANDLE for that CONTAINMENT TREE ENTRY.

NOTE—R0099 implies that with a new MDIB sequence a handle might be assigned to another element in the CONTAINMENT TREE. Therefore, after an MDIB sequence has changed, a SERVICE CONSUMER cannot make any assumptions regarding the previous relation between a HANDLE and an element in the CONTAINMENT TREE.

R0102: If a SERVICE CONSUMER detects that an MDIB sequence has changed, it SHALL not combine current CONTAINMENT TREE information with prior CONTAINMENT TREE information based on the HANDLE.

NOTE—As stated in R0099, a SERVICE PROVIDER might have reassigned a HANDLE during an MDIB sequence change. Nevertheless, a SERVICE CONSUMER is allowed to combine the information with prior information based on other metadata. However, to make sure that no obsolete information is utilized, it is strongly encouraged to re-request the whole MDIB after a MDIB sequence change.

5.2.3 Coded value

A HANDLE does not convey any semantical information. To semantically describe objects within the MDIB, this standard introduces CODED VALUEs (see pm:CodedValue). A CODED VALUE consists of a CODING SYSTEM (see pm:CodedValue/@CodingSystem) and a CODE (see pm:CodedValue/@Code).

R0008: A SERVICE PROVIDER SHOULD use standardized values for CODE and CODING SYSTEM in order to specialize a CONTAINMENT TREE ENTRY if available.

R0128: A SERVICE PROVIDER SHOULD use the ISO/IEEE 11073-10101 and IEEE 11073-10101a-2015 nomenclature whenever there is an appropriate CODE available.

NOTE—The pm:CodedValue ELEMENT defines the ISO/IEEE 11073-10101 nomenclature as the default CODING SYSTEM.

5.2.4 Instance identifier

5.2.4.1 General

Instance identifiers are defined in `pm:InstanceIdentifier`. Every instance identifier is associated with a single object or entity within a given virtual or physical system and is constituted from a root and an extension. The root value references a system, in which the extension values are unique and valid.

5.2.4.2 Encoding of unknown elements

In situations where an extension is known, but the root is not known, the implied value of a nonexistent root is a null-flavor “unknown.”

R0135: A PARTICIPANT SHALL encode the instance identifier root by the URI “biceps.uri.unk” if and only if its value is unknown.

5.2.5 MDIB versioning

5.2.5.1 General

The MDIB provides various version ATTRIBUTES to allow detection of outdated data and to allow provision of a CONTAINMENT TREE history (accessible through the ARCHIVE SERVICE).

NOTE—An informative example for change tracking can be found in D.2.

5.2.5.2 Descriptor version

Any TYPE that is derived from `pm:AbstractDescriptor`, possesses the `pm:AbstractDescriptor/@DescriptorVersion` ATTRIBUTE. The following requirements describe in which cases a descriptor version change is applied.

R0033: A SERVICE PROVIDER SHALL increment `pm:AbstractDescriptor/@DescriptorVersion` by 1 if a direct child descriptor is added or deleted.

NOTE—From R0033 follows that adding or deleting of a descriptor only increments the descriptor version of its direct parent. The version increment does not propagate upwards to other ancestors in the CONTAINMENT TREE.

R0034: A SERVICE PROVIDER SHALL increment `pm:AbstractDescriptor/@DescriptorVersion` by 1 if the content of a child ELEMENT or an ATTRIBUTE of the descriptor have changed and if the child ELEMENT itself is not derived from `pm:AbstractDescriptor`.

NOTE 1—A child ELEMENT refers to the child in an XML document tree.

NOTE 2—`pm:AbstractDescriptor` is exempted from R0034 because every parent descriptor owns a dedicated descriptor version that is not meant to be affected by updates in any child descriptor. If `pm:AbstractDescriptor` would not be exempted from R0034, then any update of a child descriptor would propagate changes up to the root descriptor of the MDIB.

5.2.5.3 State version

Any TYPE that is derived from pm:AbstractState, possesses the pm:AbstractState/@StateVersion ATTRIBUTE. The following requirements describe in which cases a state version change is applied.

R0038: A SERVICE PROVIDER SHALL increment pm:AbstractState/@StateVersion by 1 if the content of a child ELEMENT or an ATTRIBUTE of the state has changed.

NOTE—A child ELEMENT refers to the child in an XML document tree.

5.2.5.4 MdDescription version

The descriptive part of the MDIB possesses the pm:MdDescription/@DescriptionVersion ATTRIBUTE.

R0043: A SERVICE PROVIDER SHALL increment pm:MdDescription/@DescriptionVersion every time any descendent descriptor of pm:MdDescription is added or removed, and every time pm:AbstractDescriptor/@DescriptorVersion of a descendent descriptor changes.

5.2.5.5 MdState version

The state part of the MDIB possesses the pm:MdState/@StateVersion ATTRIBUTE.

R0045: A SERVICE PROVIDER SHALL increment the pm:MdState/@StateVersion every time pm:AbstractState/@StateVersion of a descendent state in pm:MdState changes.

5.2.5.6 MDIB version

The MDIB version follows the concept of WS-Discovery's AppSequence ELEMENT (see 5.3 in WS-Discovery [B3]) and is described in pm:MdibVersionGroup. The sequence identifier defined in pm:MdibVersionGroup is used to discriminate different sequences of versions. In addition to that, the instance identifier can be used to establish an order between different sequences.

R0046: A SERVICE PROVIDER SHALL increment pm:MdibVersionGroup/@MdibVersion by 1 every time pm:MdDescription/@DescriptionVersion or pm:MdState/@StateVersion have changed.

NOTE—Devices that are not capable of maintaining the pm:MdibVersionGroup/@MdibVersion number can reset it after each power cycle.

R0047: If a SERVICE PROVIDER resets pm:MdibVersionGroup/@MdibVersion, all other pm:VersionCounter instances SHALL also be reset.

R0117: A SERVICE PROVIDER SHALL assign a new MDIB sequence identifier after it has reset pm:MdibVersionGroup/@MdibVersion.

NOTE 1—A new MDIB sequence identifier is defined as a unique identifier that has never been used before by a SERVICE PROVIDER.

NOTE 2—The MDIB sequence identifier and instance identifier are explained in pm:MdibVersionGroup.

5.3 Descriptive part

5.3.1 General

The descriptive part of the MDIB describes the capabilities a POC MEDICAL DEVICE possesses. Any TYPE that constitutes a part of the hierarchy in the MDIB, is derived from a descriptor TYPE (see pm:AbstractDescriptor). The descriptor concept facilitates interoperable communication between PARTICIPANTS by introducing typed device information based on standardized nomenclatures. A minimum set of descriptive ELEMENTs and ATTRIBUTEs is declared as mandatory to help ensure interoperability and compliance of medical issues. Derived descriptor TYPEs possess some optional ELEMENTs and ATTRIBUTEs, which are included to describe the system and its capabilities in more detail.

R0010: A SERVICE PROVIDER SHALL describe the information provisioning and remote control capabilities of its represented POC MEDICAL DEVICES using the pm:MdDescription structure.

NOTE—The set of MDS CONTAINMENT TREES in pm:MdDescription might be empty at a certain point in time. This is the case if, e.g., the SERVICE PROVIDER acts as an aggregator of POC MEDICAL DEVICES and none of the devices it manages are available at that point in time.

Figure 4 shows the general structure of a CONTAINMENT TREE as a UML class diagram. It visualizes the hierarchical relationship between the descriptors that are defined in this standard. SystemContext and AlertSystem include further child descriptors. For the sake of simplicity these children are omitted in Figure 4 and moved to Figure 5 and Figure 9.

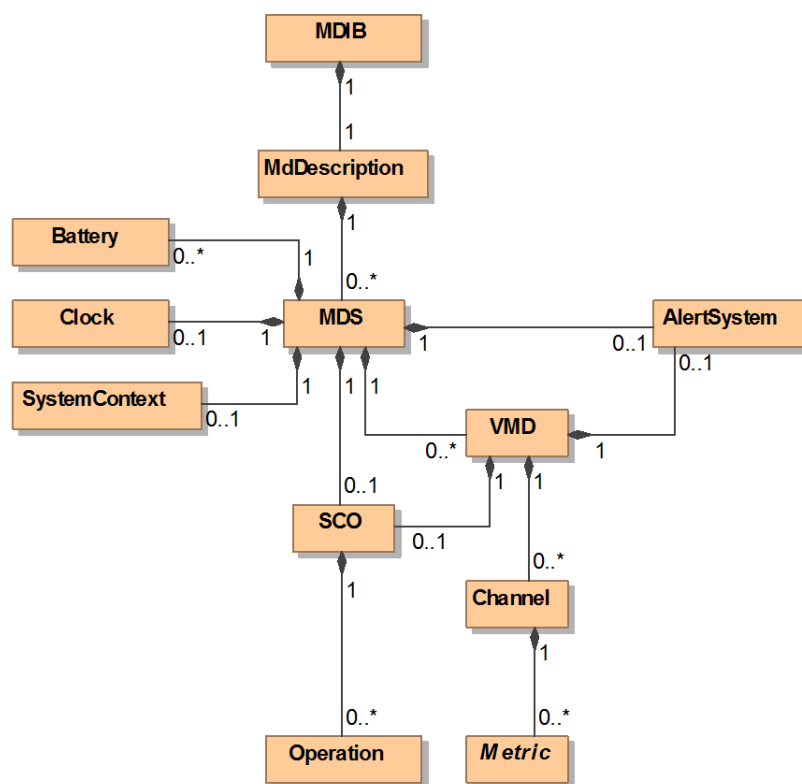


Figure 4—General BICEPS CONTAINMENT TREE overview

5.3.2 MDIB

The MDIB is the top level ELEMENT that includes the structural descriptor information. Beyond the structural descriptor information, the MDIB also includes corresponding states (omitted in Figure 4).

5.3.3 MdDescription

MdDescription is a container to group MDS descriptors, see pm:MdDescription.

5.3.4 MDS

A pm:MdDescription MAY possess zero or more pm:MdsDescriptor objects. The pm:MdsDescriptor object is depicted in Figure 3 as MDS.

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

5.3.5 Clock

An MDS MAY support a clock object to provide timestamp and time synchronization information (see pm:MdsDescriptor/pm:Clock).

NOTE—In general, POC MEDICAL DEVICES will possess a clock object; otherwise, they will not be able to keep or provide time information.

5.3.6 Battery

An MDS MAY support zero or more battery objects to provide battery-related information (see pm:MdsDescriptor/pm:Battery).

5.3.7 SystemContext

A POC MEDICAL DEVICE MAY provide contextual information to let SERVICE CONSUMERS verify if a requested SERVICE PROVIDER share the same working context in terms of, e.g., a location or patient that the POC MEDICAL DEVICE is associated with. Figure 5 shows an overview of available contexts and their relationship.

NOTE—Except for the patient and location context, all context descriptors can be present multiple times. Thereby it is possible to represent, for example, different perspectives on a workflow (multiple descriptors), each of which can contain multiple workflow steps (multiple states). An example is one pm:WorkflowContextDescriptor with pm:WorkflowContextDescriptor/pm:Type representing a hospital stay with one pm:WorkflowContextState where pm:WorkflowContextState/pm:Category represents an emergency care procedure and one pm:WorkflowContextState where pm:WorkflowContextState/pm:Category represents an Surgical Procedure. In this example, the first pm:WorkflowContextState has a pm:WorkflowContextState/@ContextAssociation of “Dis” as this workflow state has been finished already and the later has a pm:WorkflowContextState/@ContextAssociation of “Assoc” as this workflow state is currently executed.

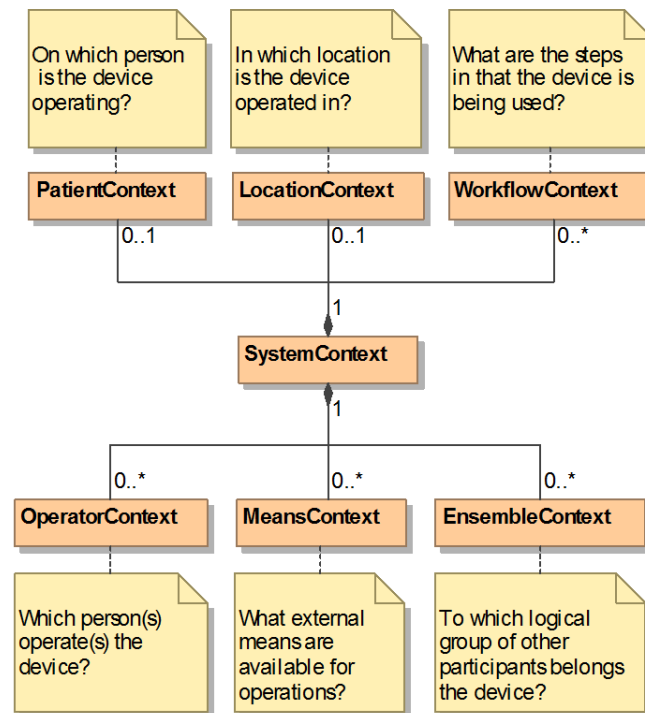


Figure 5—Contextual information that may be handled by a POC MEDICAL DEVICE

BICEPS defines six flavors of contextual information. For each contextual information a dedicated descriptor is included in pm: MdsDescriptor/pm:SystemContext:

- PatientContext (see pm:PatientContextDescriptor)
- LocationContext (see pm:LocationContextDescriptor)
- WorkflowContext (see pm:WorkflowContextDescriptor)
- OperatorContext (see pm:OperatorContextDescriptor)
- MeansContext (see pm:MeansContextDescriptor)
- EnsembleContext (see pm:EnsembleContextDescriptor)

R0013: A SERVICE PROVIDER SHALL 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.

NOTE 1—A context is called being handled, if a POC MEDICAL DEVICE or SERVICE PROVIDER is capable of validating, that it is operated in the context provided by the information accessible in the MDIB. Validation of contextual information may afford operator interaction or may rely solely on information provided by the act of proposing the contextual information. Example: a surgeon (i.e., an operator) has to confirm patient data (provided in the MDIB by a patient context) on the graphical user interface of a POC MEDICAL DEVICE. Confirming the patient data can be considered as a kind of validation.

NOTE 2—As not all POC MEDICAL DEVICES are able to handle all kinds of contextual information, an MDS is able to express which kinds of contextual information can be handled by the POC MEDICAL DEVICE. If a SERVICE PROVIDER is capable of validating the contextual information without the POC MEDICAL DEVICE that the 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.

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

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

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

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

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

NOTE—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 MDS context with a `pm:EnsembleContextDescriptor`.

NOTE—A logical group is, e.g., a temporal group based on a treatment session or spatial group that is not affixed to one location.

R0106: Any context descriptor SHALL NOT be removed from or added to the MDIB during runtime, except if the hosting MDS is removed or added to the MDIB.

NOTE 1—The existence of context descriptors defines whether a SERVICE PROVIDER supports processing a particular kind of contexts. Either a SERVICE PROVIDER is able to process a kind of context or it is not. Therefore, it makes no sense to remove or add context descriptors during runtime, except in the case when a whole MDS is removed or added to the MDIB.

NOTE 2—By not allowing context descriptors to appear or disappear during runtime, an implementation can make assumptions on the availability of contexts. This can ease implementation.

5.3.8 SCO

A POC MEDICAL DEVICE allows SERVICE CONSUMERS to request remote control commands by means of the service control object (SCO). The SCO MAY possess a set of operations that the SERVICE PROVIDER offers to SERVICE CONSUMERS to allow modification of MDIB objects. The class diagram of remote control commands is given in 5.3.9.

5.3.9 Operation

The SCO includes zero or more operations. The underlying Operation types are depicted in Figure 6, whereby

- Operation is the base type, defined in `pm:AbstractOperationDescriptor`
- SetValue is defined in `pm:SetValueOperationDescriptor`
- SetString is defined in see `pm:SetStringOperationDescriptor`

- Activate is defined in pm:ActivateOperationDescriptor
- SetContextState is defined in see pm:SetContextStateOperationDescriptor
- SetMetricState is defined in pm:SetMetricStateOperationDescriptor
- SetComponentState is defined in see pm:SetComponentStateOperationDescriptor
- SetAlertState is defined in see pm:SetAlertStateOperationDescriptor

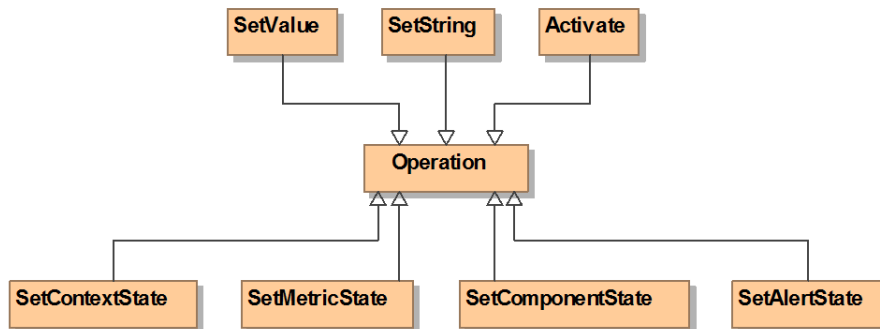


Figure 6—Operation specializations used to remote control a POC MEDICAL DEVICE

R0011: A SERVICE PROVIDER SHOULD describe all offered remote invocation capabilities using the pm:ScoDescriptor structure in pm:MdsDescriptor/pm:Sco.

5.3.10 VMD

As defined in ISO/IEEE 11073-10201, the VMD is an abstraction for a medical-related subsystem (e.g., hardware or even pure software) of a POC MEDICAL DEVICE (see pm:VmdDescriptor). The VMD MAY capture zero or more pm:ChannelDescriptor objects.

NOTE—As defined by the XML Schema, BICEPS allows a fixed CONTAINMENT TREE only, i.e., an MDIB always provides a fixed MDS to VMD to channel structure. Metrics can only be appended in channels.

5.3.11 Channel

Channel is defined in pm:ChannelDescriptor and is used to group metric objects. A channel MAY contain zero or more pm:AbstractMetricDescriptor objects.

NOTE—Since only channels are able to possess metrics, at least one channel is required to include metrics in the MDIB. This is in contrast to ISO/IEEE 11073-10201, in which the “Channel object is not mandatory for representation of Metric objects in a VMD.”

5.3.12 Metric

5.3.12.1 General

Metric is a generalized type for all objects representing direct and derived biosignal measurements and settings. The set of Metric types is illustrated in Figure 7, whereby

- Metric is the base type, defined in pm:AbstractMetricDescriptor
- StringMetric is defined in pm:StringMetricDescriptor

- EnumMetric is a subtype of StringMetric and defined in pm:EnumStringMetricDescriptor
- NumericMetric is defined in pm:NumericMetricDescriptor
- RealTimeSampleArrayMetric is defined in pm:RealTimeSampleArrayMetricDescriptor

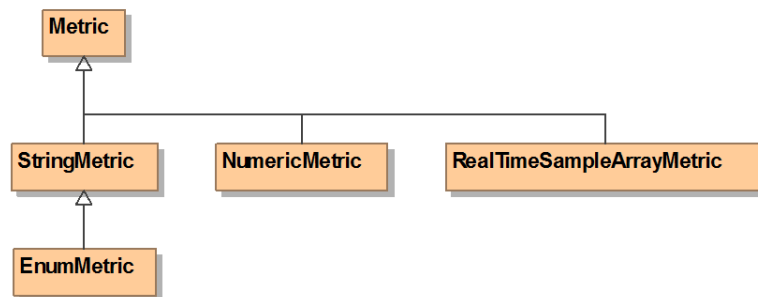


Figure 7—Metric specializations used to constitute biosignal measurements and settings

5.3.12.2 Time-related attributes

Metric values come with different time-related ATTRIBUTES to derive different temporal relationships between sequences of measurement. In Figure 8, vertical lines that are connected with horizontal lines, illustrate time periods. Single vertical lines illustrate points in time. DeterminationPeriod is the maximum time distance between two consecutive measured metric values (expressed by time points t and $t+1$, with $t < t+1$). In between DeterminationPeriod, there are time points and periods that designate specific parts of the determination process. A normative outline is given in Annex B:

- pm:AbstractMetricDescriptor/@DeterminationPeriod
- pm:AbstractMetricValue/@StartTime
- pm:AbstractMetricValue/@StopTime
- pm:AbstractMetricValue/@DeterminationTime
- pm:AbstractMetricDescriptor/@MaxMeasurementTime
- pm:AbstractMetricDescriptor/@MaxDelayTime

Beyond those ATTRIBUTES, pm:AbstractMetricState/@ActiveDeterminationPeriod designates a temporary or more-frequent changing determination period. The pm:AbstractMetricState/@ActiveDeterminationPeriod ATTRIBUTE overrides the pm:AbstractMetricDescriptor/@DeterminationPeriod period.

LifeTimePeriod is defined in pm:AbstractMetricDescriptor/@LifeTimePeriod. It designates the time period until a measured metric value is not usable anymore and can be overwritten by pm:AbstractMetricState/@LifeTimePeriod.

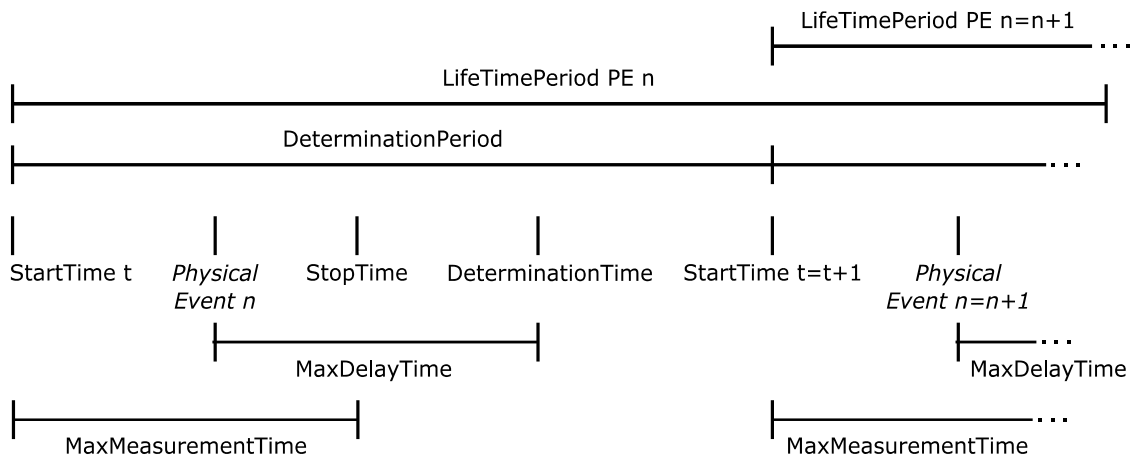


Figure 8—Logical relationship between time points and time periods provided by METRICs
 (PE refers to “Physical Event.”)

5.3.12.3 Detect obsolete metric values

A SERVICE CONSUMER MAY check if a METRIC value is outdated based on `pm:AbstractMetricDescriptor/@DeterminationPeriod` and `msg:Retrievability/@UpdatePeriod`. Table 4 illustrates in which relation of `pm:AbstractMetricDescriptor/pm:Availability` and `msg:Retrievability` the determination period and update period can be used to determine if a METRIC value is outdated.

NOTE—It is not sensible for a SERVICE CONSUMER to use `LifeTimePeriod` to determine if a METRIC value update has not been received. `LifeTimePeriod` is used only to determine if a METRIC is not useful anymore in terms of the physiological measure (also see `pm:AbstractMetricDescriptor/@LifeTimePeriod`).

Table 4—Metric availability and retrievability matrix to detect outdated METRICs

Retrievability	Availability	
	Continuous	Intermittent
Periodic	The full set of all periodic METRICs is sent via <code>msg:PeriodicMetricReport</code> , based on <code>msg:Retrievability/@UpdatePeriod</code> .	The full set of all periodic METRICs is sent via <code>msg:PeriodicMetricReport</code> , based on <code>msg:Retrievability/@UpdatePeriod</code> .
Episodic	A METRIC update is expected every time its determination period ends. The update is receivable by <code>msg:EpisodicMetricReport</code> .	The determination period is not an indicator for expected updates. It is up to the SERVICE CONSUMER how to react on unexpectedly missing METRIC updates.

5.3.13 AlertSystem

MDS and VMD objects MAY include an alert system, which in turn detects alert conditions and generates alert signals. Figure 9 shows the relation of alert systems, alert conditions and alert signals, whereby

- AlertSystem is defined in `pm:AlertSystemDescriptor`
- AlertCondition is defined in `pm:AlertConditionDescriptor`
- LimitAlertCondition is a subtype of AlertCondition and defined in `pm:LimitAlertConditionDescriptor`
- AlertSignal is defined in `pm:AlertSignalDescriptor`

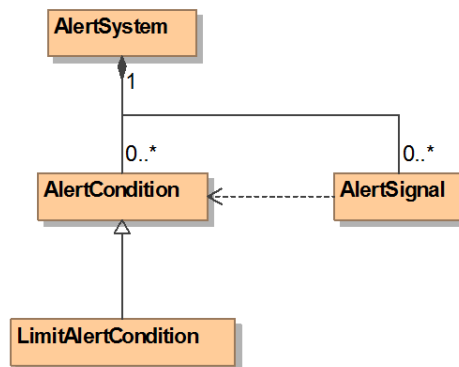


Figure 9—Alert system entities and their relationships

5.4 State part

5.4.1 General

The state part of the MDIB describes the current state of a POC MEDICAL DEVICE at a given point in time. Any TYPE that constitutes such volatile information is derived from a common state TYPE (see pm:AbstractState).

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

5.4.2 Mapping between descriptor and state

R0023: For each state in pm:MdState a SERVICE PROVIDER SHALL provide a descriptor in pm:MdDescription. Every state in pm:MdState SHALL reference its descriptor by using the pm:AbstractState/@DescriptorHandle ATTRIBUTE.

In general, every concrete descriptor/state pair matches the following naming scheme to establish a relationship: [Object Name]State is related to [Object Name]Descriptor. Examples:

- pm:NumericMetricState is related to pm:NumericMetricDescriptor.
- pm:ChannelState is related to pm:ChannelDescriptor.
- pm:PatientContextState is related to pm:PatientContextDescriptor.

5.4.3 Single state vs. multi-state

Descriptors are allowed to possess either exactly one state (i.e., a single state, see pm:AbstractState), or zero or more states. In the latter case, a state is called a multi-state (see pm:AbstractMultiState). Figure 10 depicts the single state relationship at the top and the multi-state relationship the bottom of the image. It also shows that multi-states are a specialization of single states.

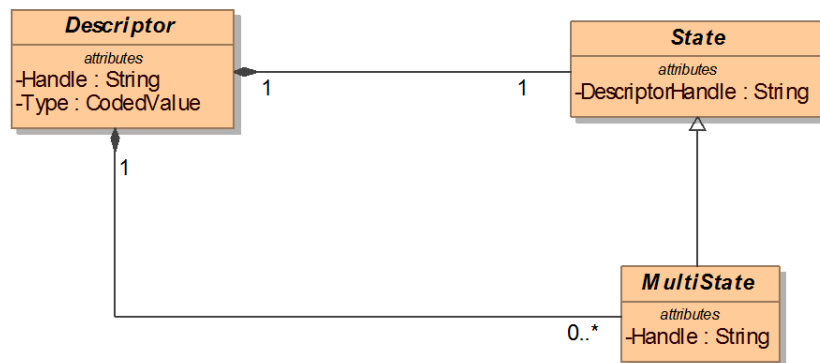


Figure 10—Relationship between descriptors, states, and multi-states

R0097: Any state that is derived from `pm:AbstractMultiState` SHALL possess a **HANDLE** that uniquely identifies the state.

NOTE 1—This is equivalent to descriptors, whereby descriptor and state HANDLES are disjunctive due to R0007.

NOTE 2—BICEPS introduces context states as the only multi-states (see `pm:AbstractContextState`).

NOTE 3—Single states need no **HANDLE** because they are uniquely identifiable by their corresponding descriptor (see `pm:AbstractState/@DescriptorHandle`).

5.4.4 Context state

R0125: A SERVICE PROVIDER SHALL use `pm:AbstractContextState/@ContextAssociation` to announce which context state it is currently associated with.

R0124: A SERVICE PROVIDER SHALL be associated with a maximum of one patient context state of a particular patient context descriptor at a time.

R0133: A SERVICE PROVIDER SHALL be associated with a maximum of one location context state of a particular location context descriptor at a time.

NOTE—This allows a SERVICE PROVIDER to be associated with multiple workflow steps, ensembles, means or operators. In contrast, a device can only be in one location and associated with only one patient.

5.4.5 BindingMdbVersion and UnbindingMdbVersion

Context states enclose the ATTRIBUTES `pm:AbstractContextState/@BindingMdbVersion` and `pm:AbstractContextState/@UnbindingMdbVersion`, which provide version information on when a context state was associated and disassociated (see `pm:AbstractContextState`). These referenced version numbers SHOULD be used to gain contextual information of any other states in the MDIB by comparing `pm:AbstractContextState/@BindingMdbVersion` and `pm:AbstractContextState/@UnbindingMdbVersion`.

A state *S* can be considered as bounded to a specific context state *CS*, if

$$\text{BindingMdbVersion}(\text{CS}) \leq \text{MdbVersion}(\text{S}) < \text{UnbindingMdbVersion}(\text{CS})$$

5.4.6 Device component and metric activation

DEVICE COMPONENTs and METRICs come with an activation state (see pm:ComponentActivation) to describe the current operating status of a CONTAINMENT TREE ENTRY within the MDIB. Figure 11 illustrates the finite state machine that SHOULD be adopted for pm:ComponentActivation. This state machine is only a recommendation of how transitions between states can be modeled.

NOTE—Devices can decide to not provide every state of pm:ComponentActivation and will therefore support only a subset of the finite state machine given in Figure 11.

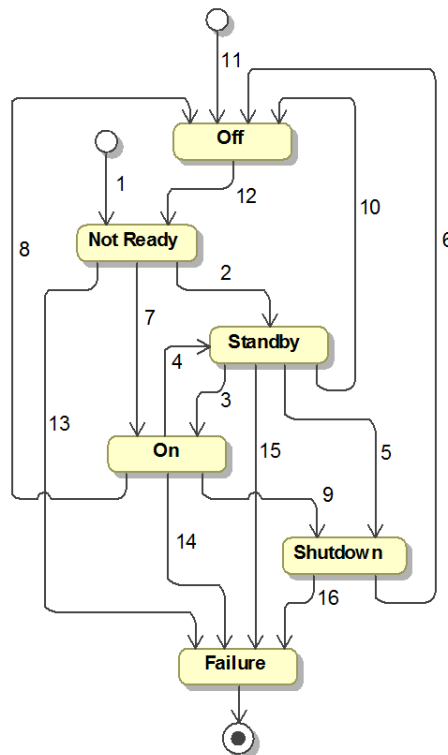


Figure 11 —Finite state machine of pm:ComponentActivation

The following semantics are applied to the transition numbers in Figure 11: the DEVICE COMPONENT or METRIC

- 1) Is made available, but needs initialization.
- 2) Has completed initialization and does not operate at the moment.
- 3) Has been activated and is operating now.
- 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 or 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.

NOTE—A shutdown phase could be for example a monitoring device that is about to be turned off, but still exposes its SERVICES on the network.

R0025: If pm:AbstractDeviceComponentState/@ActivationState is intended to be switched to “Off,” then every activation state of the component’s children SHALL be made inactive in advance. Inactive means that any pm:AbstractDeviceComponentState/@ActivationState is “Off,” any pm:AbstractAlertState/@ActivationState is “Off,” and any pm:AbstractOperationState/@OperationMode is “NA” (not available).

NOTE—From this it follows that any child component cannot be activated if a parent component is in state “Off.”

R0110: A pm:ComponentActivation MAY start in another state than the initial ones defined in Figure 11.

NOTE—This depends on which state is observed by a SERVICE PROVIDER the first time it builds the MDIB.

R0131: If a component is recovered from a failure, its activation state is REQUIRED to start in an initial state as depicted in Figure 11.

R0132: If a component is recovered from a failure, the MDIB is REQUIRED to have a new MDIB sequence identifier assigned.

5.4.7 Metric activation state

Some METRIC categories (pm:AbstractMetricDescriptor/pm:MetricCategory) enclose a specific interpretation of the pm:ComponentActivation states and SHALL be applied according to Table 5, Table 6, and Table 7.

NOTE—METRIC categories that are not listed in Table 5, Table 6, and Table 7 are not affected by special interpretation. In that case the description in pm:ComponentActivation applies.

Table 5—Component activation of measurement METRICs

pm:AbstractMetricDescriptor/pm:MetricCategory equals “Msrmt”	
pm:AbstractMetricDescriptor/ pm:ActivationState is	Interpretation
On	The measurement is performed according to pm:AbstractMetricDescriptor/@DeterminationPeriod.
NotRdy	The measurement is currently initializing. As no measurement is performed during that phase, there is no determined value available.
StndBy	The measurement has been initialized, but is not being performed. Examples: the whole MDS is in standby mode or the current configuration, e.g., ventilation mode, does not include the determination of the measurement. As no measurement is performed during that phase, there is no new determined value available, but the last determined value could be present.
Shtdn	The measurement is currently de-initializing. As no measurement is performed during that phase, there is no determined value available.
Off	The measurement is not performed and is de-initialized. As no measurement is performed during that phase, there is no determined value available.
Fail	The measurement sensor has failed. As no measurement can be performed during that phase, there is no determined value available.

Table 6—Component activation of setting METRICS

pm:AbstractMetricDescriptor/pm:MetricCategory equals “Set”	
pm:AbstractMetricDescriptor/ pm:ActivationState is	Interpretation
On	The setting is applied and according to pm:AbstractMetricDescriptor/@DeterminationPeriod reapplied.
NotRdy	The setting is currently initializing. As the setting is not applied during that phase, there is no value available.
StndBy	The setting has been initialized, but it is not being applied. Examples: the whole MDS is in standby mode or the current configuration does not include the application of the setting. If the setting has been initialized, there is a value that is applied if the encompassing METRIC is activated.
Shtdn	The setting is currently de-initializing. As the setting is not applied during that phase, there is not value available.
Off	The setting is not applied and is de-initialized. As the setting is not applied during that phase, there is no value available.
Fail	The setting cannot be applied due to a failure. As the setting cannot be applied during that phase, there is no value available.

Table 7—Component activation of calculation METRICS

pm:AbstractMetricDescriptor/pm:MetricCategory equals “Clc”	
pm:AbstractMetricDescriptor/ pm:ActivationState is	Interpretation
On	The calculation is performed according to pm:AbstractMetricDescriptor/@DeterminationPeriod.
NotRdy	The calculation is currently initializing. As no calculation is performed during that phase, there is no determined value available.
StndBy	The calculation has been initialized, but is not being performed. Examples: the whole MDS is in standby mode or that the current configuration does not include the determination of the calculation. As no calculation is performed during that phase, there is no a new determined value available, but the last determined value could be present.
Shtdn	The calculation is currently de-initializing. As no calculation is performed during that phase, there is no determined value available.
Off	The calculation is not performed and is de-initialized. As no calculation is performed during that phase, there is no determined value available.
Fail	The calculation algorithm has failed. As no calculation can be performed during that phase, there is no determined value available.

5.4.8 SCO operation activation state

Any state that is derived from pm:AbstractOperationState, has a mandatory ATTRIBUTE pm:AbstractOperationState/@OperatingMode of TYPE pm:OperatingMode to control the accessibility of a service operation.

R0026: A SERVICE PROVIDER SHALL use pm:AbstractOperationState/@OperatingMode to enable or disable the accessibility of the referenced operation according to the description in pm:OperatingMode.