

Health informatics — Vital signs information representation

ICS 35.240.80

National foreword

This Draft for Development is the official English language version of ENV 13734:2000.

This publication is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature due to the limited duration of the European Prestandard. It should be applied on this provisional basis, so that information and experience of its practical application may be obtained.

Comments arising from the use of this Draft for Development are requested so that UK experience can be reported to the European organization responsible for its conversion into a European Standard. A review of this publication will be initiated 2 years after its publication by the European organization so that a decision can be taken on its status at the end of its three-year life. The commencement of the review period will be notified by an announcement in *Update Standards*.

According to the replies received by the end of the review period, the responsible BSI Committee will decide whether to support the conversion into a European Standard, to extend the life of the prestandard or to withdraw it. Comments should be sent in writing to the Secretary of BSI Technical Committee IST/35, Health Informatics, at 389 Chiswick High Road, London W4 4AL, giving the document reference and clause number and proposing, where possible, an appropriate revision of the text.

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

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

Summary of pages

This document comprises a front cover, an inside front cover, the ENV title page, pages 2 to 401, and a back cover.

The BSI copyright date displayed in this document indicates when the document was last issued.

This Draft for Development, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Policy and Strategy Committee on 30 April 2002

© BSI 30 April 2002

ISBN 0 580 38602 3

Amendments issued since publication

Amd. No.	Date	Comments

ICS

Descriptors: medical devices, interoperability,
communications

English version

**Health informatics -
Vital signs information representation**

**Informatique de santé - Signaux
physiologiques, représentation de
l'information.**

**Informatik im gesundheitswesen -
Informationsdarstellung klinischer
vitalparameter**



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Central Secretariat: rue de Stassart, 36 B-1050 Brussels

Table of Contents

Foreword 12

Introduction..... 12

 Rationale for this standard on Vital Signs Information Representation 12

 Content of this standard for Vital signs information representation 13

 Field of Application.....13

1 Scope 17

2 Normative References 17

3 Definitions..... 19

4 Abbreviations 22

5 Requirements 23

6 Communication Parties and Roles 23

7 Domain Information Model 24

 7.1 General 24

 7.1.1 Modelling Concept 24

 7.1.2 Scope of the Domain Information Model 26

 7.1.3 Approach 26

 7.1.4 Extension of the Model..... 26

 7.2 Subject Diagram - Overview 27

 7.3 Model for the Medical Subject..... 29

 7.3.1 Virtual Medical Object 30

 7.3.2 Virtual Medical Device 30

 7.3.3 Channel..... 30

 7.3.4 Metric 30

 7.3.5 Numeric 30

 7.3.6 Sample Array..... 30

 7.3.7 Real Time Sample Array 31

 7.3.8 Time Sample Array 31

 7.3.9 Distribution Sample Array..... 31

 7.3.10 Enumeration 31

 7.3.11 Persistent Metric (PM)-Store..... 31

 7.3.12 Persistent Metric (PM)-Segment 31

7.4 Model for the Alert Subject.....	32
7.4.1 Alert.....	33
7.4.2 Alert Status.....	33
7.4.3 Alert Monitor.....	33
7.5 Model for the System Subject.....	33
7.5.1 VMS.....	34
7.5.2 MDS.....	34
7.5.3 Simple MDS.....	34
7.5.4 Hydra MDS.....	35
7.5.5 Composite Single Bed MDS.....	35
7.5.6 Composite Multiple Bed MDS.....	35
7.5.7 Log.....	35
7.5.8 Event Log.....	35
7.5.9 Battery.....	35
7.6 Model for the Control Subject.....	35
7.6.1 Service and Control.....	37
7.6.2 Operation.....	37
7.6.3 Select Item Operation.....	37
7.6.4 Set Value Operation.....	37
7.6.5 Set String Operation.....	37
7.6.6 Toggle Flag Operation.....	37
7.6.7 Activate Operation.....	38
7.6.8 Limit Alert Operation.....	38
7.7 Model for the Extended Services Subject.....	38
7.7.1 Scanner.....	39
7.7.2 Configurable Scanner.....	39
7.7.3 Episodic Configurable Scanner.....	39
7.7.4 Periodic Configurable Scanner.....	39
7.7.5 Fast Periodic Configurable Scanner.....	40
7.7.6 Unconfigurable Scanner.....	40
7.7.7 Context Scanner.....	40
7.7.8 Alert Scanner.....	40

7.7.9 Operating Scanner	40
7.7.10 Discriminator	40
7.8 Model for the Communication Subject	41
7.8.1 Communication Controller	41
7.8.2 Specialized Communication Controller	41
7.9 Model for the Archival Subject	42
7.9.1 Multipatient Archive	42
7.9.2 Patient Archive	43
7.9.3 Session Archive	43
7.9.4 Physician	43
7.9.5 Session Test	43
7.9.6 Session Notes	43
7.9.7 Ancillary	43
7.10 Model for the Patient Subject	43
7.10.1 Patient Demographics	44
7.11 Domain Information Model - Dynamic Model	44
7.11.1 General	44
7.11.2 MDS Communication Finite State Machine	44
7.11.3 Dynamic Object Relations	45
8 Domain Information Model - Object Definitions	47
8.1 General	47
8.1.1 Notation	47
8.1.2 Common Data Types	47
8.2 Top Object	53
8.2.1 Attributes	54
8.2.2 Behaviour	54
8.2.3 Notifications	54
8.3 Objects in the Medical Subject	54
8.3.1 Virtual Medical Object	54
8.3.2 Virtual Medical Device	55
8.3.3 Channel Object	57
8.3.4 Metric	58
8.3.5 Numeric	63

8.3.6	Sample Array.....	65
8.3.7	Real Time Sample Array.....	69
8.3.8	Time Sample Array	70
8.3.9	Distribution Sample Array.....	72
8.3.10	Enumeration	74
8.3.11	PM-Store	75
8.3.12	PM-Segment.....	78
8.4	Objects in the Alert Subject	79
8.4.1	Alert.....	79
8.4.2	Alert Status	81
8.4.3	Alert Monitor.....	83
8.5	Objects in the System Subject.....	84
8.5.1	Virtual Medical System.....	84
8.5.2	Medical Device System.....	86
8.5.3	Simple MDS	89
8.5.4	Hydra MDS	89
8.5.5	Composite Single Bed MDS.....	90
8.5.6	Composite Multiple Bed MDS	90
8.5.7	Log.....	90
8.5.8	Event Log	92
8.5.9	Battery	93
8.6	Objects in the Control Subject	95
8.6.1	Service and Control Object.....	95
8.6.2	Operation.....	98
8.6.3	Select Item Operation	100
8.6.4	Set Value Operation.....	101
8.6.5	Set String Operation	102
8.6.6	Toggle Flag Operation.....	103
8.6.7	Activate Operation.....	104
8.6.8	Limit Alert Operation	105
8.7	Objects in the Extended Services Subject	107
8.7.1	Scanner	107

8.7.2 Configurable Scanner	108
8.7.3 Episodic Configurable Scanner	110
8.7.4 Periodic Configurable Scanner	111
8.7.5 Fast Periodic Configurable Scanner	112
8.7.6 Unconfigurable Scanner	113
8.7.7 Context Scanner.....	114
8.7.8 Alert Scanner	115
8.7.9 Operating Scanner	116
8.7.10 Discriminator	118
8.8 Objects in the Communication Subject.....	119
8.9 Objects in the Archival Subject.....	119
8.9.1 Multipatient Archive.....	119
8.9.2 Patient Archive	120
8.9.3 Session Archive	121
8.9.4 Physician	122
8.9.5 Session Test.....	123
8.9.6 Session Notes.....	124
8.10 Objects in the Patient Subject	125
8.10.1 Patient Demographics.....	125
9 Service Model for Communicating Systems.....	127
9.1 General.....	127
9.2 Communicating Systems.....	127
9.3 General Service Model Overview	128
9.3.1 Conceptual Architecture of Communicating Systems	128
9.3.2 Operational Services.....	130
9.3.3 Notification Services	130
9.3.4 Services Used by a Manager System.....	130
9.3.5 Services Used by an Agent System	130
9.3.6 Services Used by a Hybrid System.....	130
9.4 General Object Management Services Definition.....	130
9.4.1 Event Report.....	130
9.4.2 GET	131
9.4.3 SET.....	132

9.4.4 ACTION.....	132
9.4.5 CREATE	133
9.4.6 DELETE.....	134
10 The Medical Data Information Base.....	135
11 Conformance Model.....	135
11.1 Applicability.....	135
11.2 Conformance Specification.....	136
11.3 Implementation Conformance Statements	136
11.3.1 General Format.....	136
11.3.2 General Implementation Conformance Statement.....	137
11.3.3 Service Support Implementation Conformance Statement.....	138
11.3.4 DIM MOC Implementation Conformance Statement.....	139
11.3.5 MOC Attribute Implementation Conformance Statement.....	140
11.3.6 MOC Behaviour Implementation Conformance Statement.....	140
11.3.7 MOC Notification Implementation Conformance Statement	141
11.3.8 Levels of Conformance	141
Annex A – Normative, Medical data information base (MDIB) nomenclature, data dictionary and codes	142
A.1 Overview of Annex A.....	142
A.1.1 Introduction	142
A.1.2 Code Assignment to the MDIB Elements.....	143
A.2 Data Dictionary and Codes for Object Oriented Modelling Elements	146
A.2.1 Introduction	146
A.2.3 Object-Oriented Modelling Elements: Inventory Tables.....	148
A.3 Nomenclature, Data Dictionary and Codes for Vital Signs Instrumentation/Devices.....	165
A.3.1 Introduction	165
A.3.2 Base Concepts	166
A.3.3 First Set of Differentiating Criteria.....	167
A.3.4 Second Set of Differentiating Criteria	167
A.3.5 Third Set of Differentiating Criteria	168
A.3.6 Attributes	168
A.3.7 Code Table	168
A.4 Terminology and Codes for Units of Measurement.....	174
A.4.1 Introduction	174

A.4.2 Orders of Magnitude Discriminator.....	174
A.4.3 Units outside of SI.....	175
A.4.4 Units of Measurement	176
A.5 Nomenclature, Data Dictionary and Codes for Metrics (Measurements and Enumerations).....	181
A.5.1 Nomenclature for ECG Measurements.....	181
A.5.2 Nomenclature for ECG Enumerations.....	194
A.5.3 Nomenclature, Data Dictionary and Codes for Haemodynamic Monitoring Measurements	199
A.5.4 Nomenclature, Data Dictionary and Codes for Respiratory Measurements	211
A.5.5 Nomenclature, Data Dictionary and Codes for common Blood Gas, Blood, Urine and other fluid Chemistry Measurements	231
A.5.6 Nomenclature, Data Dictionary and Codes for Fluid Output Measurements	237
A.5.7 Nomenclature , Data Dictionary and Codes for Pumps.....	240
A.5.8 Nomenclature, Data Dictionary and Codes for Neurological Monitoring Measurements.....	251
A.5.9 Nomenclature, Data Dictionary and Codes for Neurophysiologic Enumerations	261
A.5.10 Nomenclature, Data Dictionary and Codes for Stimulation Modes.....	277
A.5.11 Nomenclature, Data Dictionary and Codes for Miscellaneous Measurements	282
A.6 Nomenclature, Data Dictionary and Codes for Body Sites.....	285
A.6.1 Introduction	285
A.6.2 Sites for neurophysiological signal monitoring: location near peripheral nerves.....	285
A.6.3 Sites for neurophysiological signal monitoring: location near muscles.....	294
A.6.4 Nomenclature, Data Dictionary and Codes for sites for EEG-electrode placement on the head.....	314
A.6.5 Sites for EOG signal monitoring	318
A.6.6 Nomenclature, Data Dictionary and Codes for General Neurological Sites for monitoring measurements and drainage	322
A.6.7 Nomenclature, Data Dictionary and Codes for Body sites for CardiovascularMeasurements	323
A.6.8 Nomenclature, Data Dictionary and Codes for miscellaneous body sites used in vital signs monitoring and measurement	326
A.6.9 Nomenclature, Data Dictionary and Codes for Qualifiers of Body Sites Locations.....	334
A.7 Nomenclature, Data Dictionary and Codes for Alerts.....	336
A.7.1 Introduction	336
A.7.2 Diagnostic Pattern Events.....	336
A.7.3 Device and environment related Events	342
A.8 Nomenclature, Data Dictionary and Codes for External Nomenclatures.....	358
A.8.1 Introduction	358

A.8.2 External Nomenclature	358
Annex B – Informative Scenarios.....	361
B.1 Overview.....	361
B.1.2 Introduction	361
B.1.3 Application to the Scope of the Vital Signs Standard.....	361
B.2 Scenario 1 - Data Logger - Single Device.....	363
B.2.1 Overview	363
B.2.2 Examples	363
B.2.3 Data Requirements	364
B.2.4 Communications Requirements.....	365
B.3 Scenario 2 - Data Logger - Multiple Devices.....	365
B.3.1 Overview	365
B.3.2 Examples	365
B.3.3 Data Requirements Introduced	366
B.3.4 Communication Requirements Introduced	366
B.4 Scenario 3 - Real-Time Data Display	367
B.4.1 Overview	367
B.4.2 Examples	367
B.4.3 Data Requirements Introduced	368
B.4.4 Communication Requirements Introduced	368
B.5 Scenario 4 - Patient Alarm Monitoring.....	368
B.5.1 Overview	368
B.5.2 Example.....	368
B.5.3 Data Requirements Introduced	368
B.5.4 Communication Requirements Introduced	369
B.6 Scenario 5 - Remote Control.....	369
B.6.1 Overview	369
B.6.2 Examples	369
B.6.3 Data Requirements Introduced	370
B.6.4 Communication Requirements Introduced	370
B.7 Scenario 6 - Patient Viewing Interoperability.....	370
B.7.1 Overview	370
B.7.2 Examples	370

B.7.3 Data Requirements Introduced	371
B.7.4 Communication Requirements Introduced	371
B.8 Scenario 7 - Patient Monitoring Interoperability	372
B.8.1 Overview	372
B.8.2 Examples	372
B.8.3 Data Requirements Introduced	373
B.8.4 Communication Requirements Introduced	373
B.9 Scenario 8 - Patient Data Exchange (Off-line).....	373
B.9.1 Overview	373
B.9.2 Examples	373
B.9.3 Data Requirements Introduced	374
B.9.4 Communication Requirements Introduced	374
B.10 Scenario 9 - Patient Data Exchange (On-line/Interactive)	374
B.10.1 Overview	374
B.10.2 Examples	375
B.10.3 Data Requirements Introduced	375
B.10.4 Communication Requirements Introduced	375
Annex C – Informative Communicating Systems Example.....	376
C.1 General	376
C.1.1 Disclaimers	376
C.1.2 Purpose	376
C.1.3 Organization	376
C.1.4 Sample Application	376
C.2 The Manager/Agent Framework	376
C.3 Local Agent Initialization	379
C.4 Association.....	382
C.5 Configuration	384
C.5.1 MDS Create Event.....	384
C.5.2 Context Scanner.....	386
C.6 Using the GET Service	389
C.7 Using Scanner Objects for Automated Attribute Access	391
Annex D – Informative Nomenclature Building for the Medical Data Information Base and the Data Dictionary	394

D.1 Introduction.....394

D.2 Categorical Structures and Descriptors.....394

D.3 Possible Uses and Advantages from the Approach of Categorical Structures.....398

D.4 Target Categories in the MDIB.....398

Annex E - Informative, Further References.....400

Foreword

This European Prestandard has been prepared by Technical Committee CEN/TC 251 "Health informatics", the secretariat of which is held by SIS.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this European Prestandard: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

Introduction

Rationale for this standard on Vital Signs Information Representation

Medical technology has developed in the context of rapid advances in the science of information technology and has benefited greatly by incorporating these advances into many measurement and other devices. This has been done in a relatively unstructured manner with many devices being developed in isolation and in a way that precludes communication both between themselves and with hospital computer and data management systems.

The potential advantages of such communication are becoming more and more obvious. Consequently, there is a pressing need for technical standardization and the development of protocols to enable such communication to exist in an easy and open way, with subsequent clinical, administrative and research benefits.

An essential element is the development of a standard for Vital Signs Representation, which in itself is a pre-requisite for the development of true interoperability.

Interoperability of medical devices raises many technical, safety, and legal issues which are worthy of consideration.

The relevant aspect of this document is the definition of a common device independent representation of Vital Signs Information and the definition of a common model for access of this information. Also the demands resulting from real-time communication have to be considered.

Development of this standard was based upon:

- The ISO Open Systems Interconnection Basic Reference Model.
- Identification and definition of information objects and their relationships based upon a set of scenarios with application of the object-oriented analysis and design methodology. Within CEN/TC251, during the production of this standard, the approach proposed by Coad and Yourdon has been followed.

Furthermore, the discussion and development work was guided by the CEN Technical Report CR1350: 1993 "Investigation of Syntaxes for Existing Interchange Formats to be used in Healthcare" and with consideration of the recently finished documents CEN prENV 12264 "Categorical Structure of Systems of Concepts - Model for Representation of Semantics (MOSE)" and CEN/TC 251/N95-295 Draft First Working Document "Categorical Structure of Systems of Concepts - Medical Devices". Respectively work and documents of the other TC 251 working groups have been considered and referenced.

Work on Standardization of Interoperability for Medical Devices has been carried on by various groups for more than ten years. The development of this standard would not have been possible without previous work done particularly by the IEEE Committee P 1073 and a number of national organizations in the field of Biomedical Engineering and Medical Informatics. Also, significant input has been obtained from other working groups of CEN/TC 251, particularly Working Group 2 dealing with Health Care Terminology, Semantics and Knowledge Bases, Working Group 3 dealing with Health Care Communications and Messages and Working Group 4 dealing with Medical Imaging and Multimedia. Where appropriate, reference is made in the text to their contributions.

Content of this standard for Vital signs information representation

The major parts of this standard are:

- The Domain Information Model (DIM).

It is based upon scenarios for "off-line" data acquisition and archival, e.g. in sleep laboratories, and upon scenarios for departmental information systems in intensive care units with "on-line" communication. This communication takes place between a variety of interconnected medical device systems which may not only acquire data but also perform more or less extended processing.

- The Service Model for these communicating systems and devices.
- The dictionary for information elements of the Medical Data Information Base (MDIB) to be managed and communicated within the system.

This document is structured as follows:

- scope and field of application are defined;
- normative references, definitions and abbreviations are given;
- communication parties and their roles are described;
- the Domain Information Model and the Service Model are specified (in chapters 7 and 8);
- the dictionary for the elements of the MDIB and the methodology for the construction of its nomenclature is described (in chapter 9);
- the Normative Annex A contains the currently defined dictionary;
- the scenarios on which the Domain Information Model is based are given in Informative Annex B;
- a step by step example for communicating systems is given in Informative Annex C;
- other references are provided in Informative Annex D.

In order to understand the essential parts of the standard it is recommended that the step by step example be studied initially.

Field of Application

Figure 0.1 depicts a system where device 1 and device 2 are interconnected by a communication system.

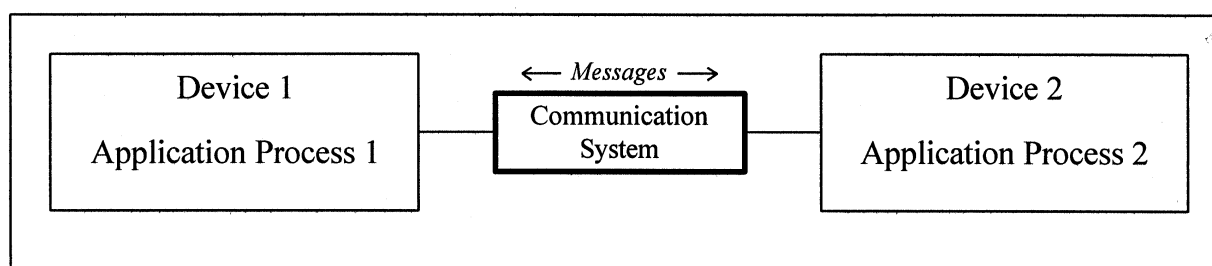


Figure 0.1: Two Communicating Devices

The "activities" (user functionalities) running on these devices, e.g. data acquisition on device 2, data display on device 1 are called application processes. Communication between these systems takes place by means of messages. The messages may contain the data, encoding information and control signals. The communication system block in Figure 0.1 represents hardware and software for interconnection of devices 1 and 2. The set of rules that define content and structure of the messages and their time relationships is called a "protocol" (ISO definition).

In an open communication system devices use a standardized set of common protocols that permits:

- the establishment of a logical connection between application processes;
- the exchange of (vital signs) information in a standardized form irrespective of a specific device type or device architecture so that the information is able to be located and interpreted.

To enable the development of standards for communication processes the ISO Seven Layer Basic Reference Model for Open Systems Interconnection has been established (see ISO/IEC 7498-1: 1994).

Figure 0.2 illustrates the principle of communication between application process 2 and application process 1 through the OSI Seven Layer Model:

- Information exchange between the two application processes takes place vertically through the layers.
- Specific services in each layer prepare the information to be transferred in such a way that the next lower/upper layer is able to transmit/receive the data. The services are activated from the N+1 layer and provided by the N (lower) layer.
- Each protocol layer provides apparent horizontal links between peer-entities of this layer.

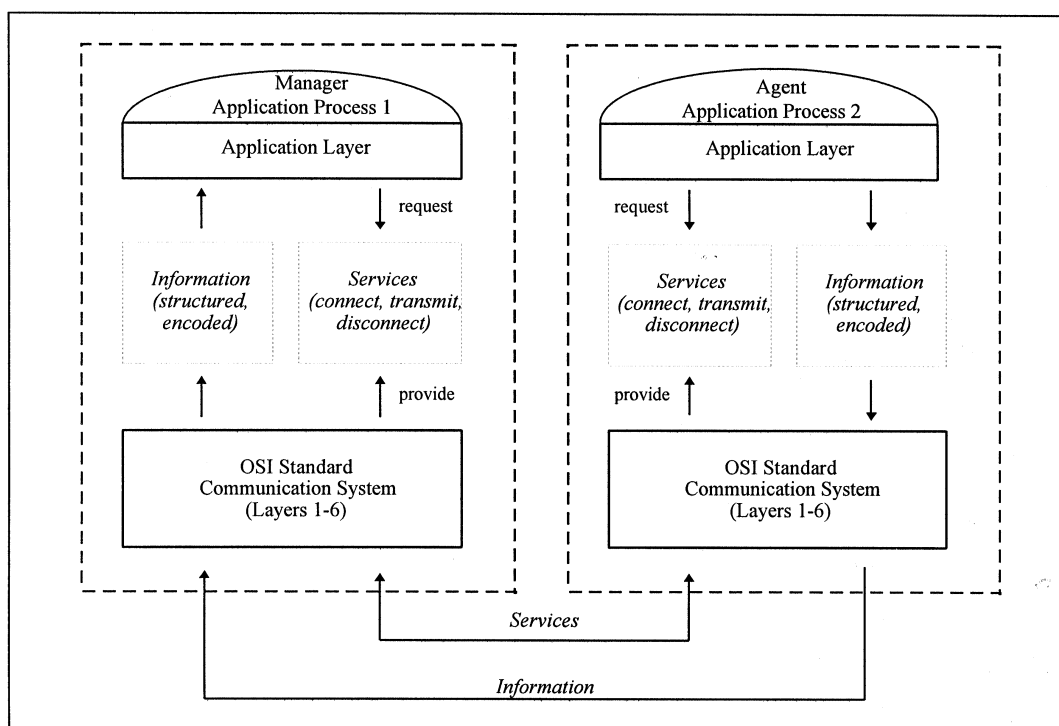


Figure 0.2: Simplified Model of Open Systems Communication

The advantage of the seven layer model is that its structure allows a very complex protocol for communication between applications to be broken down to a hierarchy of simpler protocols between the layers which it is possible to define and standardize. Each layer has only to provide limited functionality.

This standard addresses:

- the interchange of Vital Signs Information between individual devices and between devices and the departmental processing system within the acute care setting; and,
- data acquisition, archival and the exchange of archived data on storage media or via networks, e.g. in the electrophysiology sleep laboratory setting.
- the definition and structuring of information that is communicated or referred to in communication between application entities.

The standard provides a "common" representation of all application entities present in the application processes within the various devices independent of the syntax.

Figure 0.3 illustrates the manager/agent (client/server) framework for communication systems that is defined and used in this standard.

In order to facilitate the development and application of this standard, it is conceptually based on the ISO/OSI Management Framework as an addition to the Basic Reference Model.

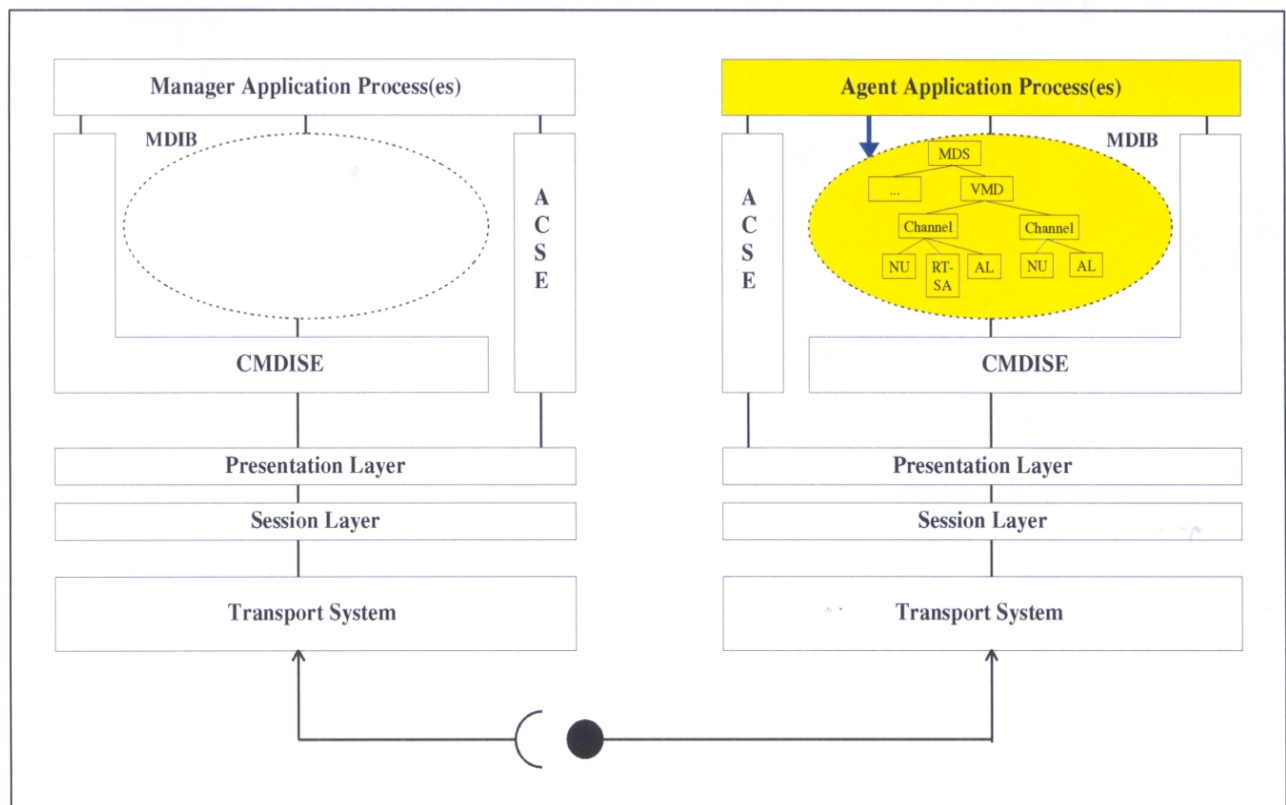


Figure 0.3: Example of a Manager Agent System¹

For example, the agent in Figure 0.3 could be a device or a device system for acquiring vital data (ECG, blood pressure). Its functionality is represented by the agent's application process. The manager may be an archiving system and this functionality is represented by the manager's application process.

The standard defines the structure of the Medical Data Information Base (MDIB) that contains the managed medical objects as defined in the Domain Information Model. These managed medical objects represent the "real world instances" of the application domain. The objects within the MDIB are:

- Medical Device System (MDS);
- Virtual Medical Device (VMD);
- Channel;
- Numeric (measurements, status information, etc.) (NU);
- Real Time Sample Arrays (RTSA);
- Alerts (AL);
- etc.

Details are in Chapter 7 describing the Domain Information Model.

¹ Vital signs are represented by managed medical objects identified within the Medical Data Information Base. Both device information and demographic data need to be transmitted and are therefore part of the MDIB.

A related Medical Data Information Base exists in the manager as well. However, one should be aware of the fact that the MDIB in each device may be different and that the MDIB within the manager may not always contain all the objects as identified within the agent MDIB in the Figure 0.3 and vice versa.

The standard defines the Service Model for object access. The **Common Medical Device Information Service Element CMDISE** provides the services to the objects in the MDIB. The **CMDISE** is a specialization of the ISO common management information service element (**CMISE**).

ACSE is the Association Control Service Element which provides the services to establish logical connections between the agent and the manager. These are defined in ISO/IEC 8649 and 8650. The definition of association control and lower layer communication is outside the scope of the standard.

1 Scope

This standard addresses the definition and structuring of information that is communicated or referred to in communication between application entities.

Within the scope of this standard is:

- the interchange of Vital Signs Information between individual devices and between devices and the departmental processing system within the acute care setting; and,
- data acquisition, archival and the exchange of archived data on storage media or via networks, e.g. in the electrophysiology sleep laboratory setting.

The standard provides a "common" representation of all application entities present in the application processes within the various devices independent of the syntax.

The definition of association control and lower layer communication is outside the scope of the standard.

2 Normative References

This document incorporates, by dated or undated reference, provisions from other publications. These normative references are cited in the appropriate place in the text and are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this document only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

CR 1350	1993	Investigation of syntaxes for existing interchange formats to be used in Healthcare
ENV 1068	1993	Medical Informatics - Healthcare information interchange - Registration of coding schemes
ENV 1614	1995	Medical Informatics - System of concepts for systematic names, classification and coding for properties, including quantities, in laboratory medicine
ENV 12264	1997	Medical Informatics - Categorical Structure of Systems of Concepts - Model for Representation of Semantics (MOSE)
ISO 41-0	1992	Quantities and units - Part 0: General principles
ISO 639	1988	Symbols for languages, geographical areas and authorities
ISO 2925	1983	Representation of SI and other units in systems with limited characters sets
ISO 3166	1988	Codes for representation of names of countries
ISO 5218	1977	Information interchange - Representation of human sexes
ISO 7498-1	1994	Information Technology - Open Systems Interconnection - Basic Reference Model
ISO 7498-2	1988	Information processing systems - Open Systems Interconnection - OSI Security Architecture
ISO 7498-3	1980	Information processing systems - Open Systems Interconnection - Naming and Addressing
ISO 7498-4	1980	Information processing systems - Open Systems Interconnection - OSI Management Framework

ISO 8601	1988	Data elements and interchange formats - Information interchange - Representation of dates and times
ISO 8649		Information technology - Open Systems Interconnection - Service definition for the Association Control Service Elements
ISO/IEC 8650-1		Information technology - Open Systems Interconnection - Specification of Protocols for Common Application Service Elements - Part 1: Introduction
ISO/IEC 8650-2		Information technology - Open Systems Interconnection - Specification of Protocols for Common Application Service Elements - Part 2: Association Protocol
ISO/IEC 8824	1990	Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).
ISO/IEC 9545	1989	Information processing systems - Open Systems Interconnection - Application Layer Structure (ALS)
ISO/IEC 9595(E)	1991	Information technology - Open Systems Interconnection - Common Management Information Service (CMIS) Definition
ISO/IEC 10040	1992	Information technology - Open Systems Interconnection - Systems management overview
ISO/IEC 10164-1(E)	1993	Information technology - Open Systems Interconnection - Systems Management: Object Management Function
ISO/IEC 10164-2(E)	1993	Information technology - Open Systems Interconnection - Systems Management: State management function
ISO/IEC 10164-3(E)	1993	Information technology - Open Systems Interconnection - System Management: Attributes for representing relationships
ISO/IEC 10164-4(E)	1992	Information processing systems - Open Systems Interconnection - Systems Management: Alarms reporting function
ISO/IEC 10164-5(E)	1993	Information technology - Open Systems Interconnection - Systems Management: Event Report Management Function
ISO/IEC DIS 10164-6	1991	Information technology - Open Systems Interconnection - Systems Management: Log Control function
ISO/IEC 10164-7(E)	1992	Information technology - Open Systems Interconnection - Systems Management: Security alarm reporting function
ISO/IEC 10164-8(E)	1993	Information technology - Open Systems Interconnection - Systems Management: Security audit trail function
ISO/IEC DIS 10164-9	1993	Information technology - Open Systems Interconnection - Systems Management: Objects and attributes for access control
ISO/IEC DIS 10164-10	1993	Information technology - Open Systems Interconnection - Systems Management: Accounting metering function
ISO/IEC DIS 10164-11	1992	Information technology - Open Systems Interconnection - Systems Management: Workload Monitoring Function
ISO/IEC DIS 10164-12	1992	Information technology - Open Systems Interconnection - Systems Management: Test Management Function

ISO/IEC DIS 10164-13	1993	Information technology - Open Systems Interconnection - Systems Management: Summarization Function
ISO/IEC DIS 10164-14	1993	Information technology - Open Systems Interconnection - Systems Management: Confidence and diagnostic test
ISO/IEC 10165-1(E)	1993	Information processing systems - Open Systems Interconnection - Structure of management information: Management information model
ISO/IEC 10165-2(E)	1992	Information processing systems - Open Systems Interconnection - Structure of management information: Definition of management information

3 Definitions

For the purpose of this document, the following definitions apply:

Term	Definition
Agent:	Device which provides data in a manager/agent communicating system.
Alarm:	Signal which indicates abnormal events occurring to the patient or the device system.
Alert:	Synonym for the combination of patient-related physiological alarms, technical alarms and equipment user advisory signals.
Alert Monitor:	Object representing the output of a device or system alarm processor and as such the overall device or system alarm condition.
Alert Status:	Object representing the output of an alarm process that considers all alarm conditions in a scope that spans one or more objects.
Archival:	Relating to the storage of data over a prolonged period.
Association Control Service Element:	Method used to establish logical connections between medical device systems.
Channel:	Umbrella object in the object model that groups together physiological measurement data and data derived from these data.
Class:	Description of one or more objects with a uniform set of attributes and services including a description of how to create new objects in the class.
Common Medical Device Information Service Element:	With associated protocol provides services that allow access to the object instances in the Medical Data Information Base.
Communication Controller:	Part of a Medical Device System responsible for communications.
Communication Party:	Actor of the problem domain participating in the communication therein.
Communication Role:	Role of a party in a communication situation defining the party's behaviour in the communication. Associated with a communication role is a set of services which the party provides to other parties.
Data Format:	Arrangement of data in a file or stream.
Data Logger:	A medical device which is functioning in its capacity as a data storage and archival system.

Data Agent:	As a medical device a patient data acquisition system which provides the acquired data for other devices.
Data Structure:	Manner in which application entities construct the data set information resulting from the use of an information object.
Dictionary:	Description of the contents of the Medical Data Information Base containing vital signs information, device information, demographics, and other elements of the MDIB.
Discrete Parameter:	Vital signs measurement that it is possible to express as a single numeric or textual value.
Domain Information Model:	The model describing common concepts and relationships for a problem domain.
Event:	Event that is communicated by Event Report service.
Event Report:	Service (provided by the CMDISE) to report an event relating to a managed object instance.
Framework:	A structure of processes and specifications designed to support the accomplishment of a specific task.
Graphic Parameter:	Vital signs measurement that requires a number of regularly sampled data points in order to be expressed properly.
Host system:	Term used as an abstraction of a medical system to which measurement devices are attached.
Information Objects:	Provide an abstract data model applicable to the communication of vital signs information and related patient data. The attributes of an information object definition describe its properties. Each information object definition does not represent a specific instance of real-world data, but rather a class of data which share the same properties.
Instance:	For example: object instance, application instance, information service element instance, VMD instance, class instance, operating instance
Intensive Care Unit:	Unit within a hospital in which critically ill patients are managed using multiple modes of monitoring and vital system support.
Interchange Format:	The representation of the data elements and the structure of the message containing those data elements while in transfer between systems. The interchange format consists of a data set of construction elements and a syntax. The representation is technology specific.
Interoperability:	Idealized scheme whereby medical devices of differing types, models or manufacturers, are capable of working with each other, whether connected to each other directly or through a communication system. The communication system tracks location, connection, disconnection, replacement and re-siting of all devices.
Latency:	In a communications scenario latency is the time delay between sending a signal from one device and receiving it by another device.
Lower Layers:	Layers 1 to 4 of the ISO/OSI reference model. These layers cover mechanical, electrical and general communication protocol specifications.
Manager:	Device which receives data in a manager/agent communicating system.
Manager Agent model:	Communication model whereby one device (agent) provides data and another device (manager) receives data.

Medical Data Information Base:	The concept of an object oriented database storing (at least) vital signs information.
Medical Device System:	Abstraction for system comprising one or more medical devices.
Medical Devices:	Devices, apparatus, or systems used to prevent, measure, diagnose, or treat diseases in humans, and which do not normally enter metabolic pathways. For the purposes of this standard the scope of medical devices is further limited to those patient connected medical devices which provide support for electronic communications.
Monitor:	Patient data acquisition device designed to provide warning of possible patient deterioration.
Object Attributes:	Data which, together with methods, define an object.
Object Class:	A descriptor used in association with a group of objects with similar properties (attributes), common behaviour (operations), common relationships to other objects, and common semantics.
Object Diagram:	Diagram showing connections between objects in a system.
Object Oriented Analysis:	Method of analysis described by Coad and Yourdon but the term Object-Oriented is used widely to describe similar techniques.
Object:	A concept, an abstraction or a thing with crisp boundaries and a meaning for the problem at hand.
Open System:	Set of protocols allowing computers of different origins to be linked together.
Operation:	A function or transformation that may be applied to or by objects in a class. (Sometimes also called service.)
Problem Domain:	The field of health care under consideration in a modelling process.
Protocol:	A standard set of rules describing the transfer of data between devices. It specifies the format of the data, and the signals to start, control and end the transfer.
Scanner:	Observer and "summarizer" of object attribute values.
Scenario:	Formal description of a class of business activities including the semantics of business agreements, conventions, and information content.
Service:	A specific behaviour that a communication party in a specific role is responsible for exhibiting.
Syntax:	The syntax of an Interchange Format (IF) describes the rules for combining the construction elements of the Interchange Format.
System:	Demarcated part of the perceivable universe, existing in time and space, that may be regarded as a set of elements and relationships between these elements.
Timestamp:	An attribute or field in data which denotes the time of data generation.
Top Object:	Ultimate base class for all other objects belonging to one model.
Upper Layers:	Layers 5 to 7 of the ISO/OSI reference model. These layers cover application, presentation, and session specifications and functionalities.
Virtual Medical Device:	An abstract representation of a medical related subsystem of a Medical Device System.
Virtual Medical Object:	An abstract representation of an object in the medical subject of the Domain Information Model.

Vital Sign:	Clinical information relating to one or more patients. Measured by or derived from apparatus connected to the patient, or otherwise gathered from the patient.
Waveform:	Here, used as graphic data, typically data values varying with respect to time. Vital signs data which is usually presented to the clinician in a graphical form.

4 Abbreviations

ACSE	Association Control Service Element
ASN.1	Abstract Syntax Notation One
CEN	European Committee for Standardization
CMDISE	Common Medical Device Information Service Element
DIM	Domain Information Model
DIS	Department Information System
ECG	Electrocardiogram
EEG	Electroencephalogram
FSM	Finite State Machine
HIS	Hospital Information System
ICU	Intensive Care Unit
ID	Identification
IEEE	Institute of Electrical and Electronic Engineers
IF	Interchange Format
ISO	International Standards Organization
LAN	Local Area Network
MDIB	Medical Data Information Base
MDS	Medical Device System
MIB	Management Information Base
OSI	Open Systems Interconnection
PCS	Patient Care System
PDU	Protocol Data Unit
PM	Persistent Metric
PT	Project Team
SA	Sample Array
TC	Technical Committee

VMD	Virtual Medical Device
VMO	Virtual Medical Object
WAN	Wide Area Network
WG	Working Group

5 Requirements

The medical community requires interoperability of medical devices and medical device systems. The standard for Vital Signs Information Representation is a prerequisite for the development of communication protocols and interchange formats in the medical device area.

6 Communication Parties and Roles

Three characteristic clinical settings can be identified for analysis of vital signs processes:

- Data acquisition with optional (pre-) processing and non time-critical information transfer.
- Data acquisition with information transfer and processing in a real-time environment.
- Data reviewing and/or retrieval after initial acquisition. This process is done to review recorded data for medical reports and to produce documentation. It is not time-critical and the archival system can be seen as the data "Agent".

Figure 6.1 and Figure 6.2 illustrate the first two principal system configurations.

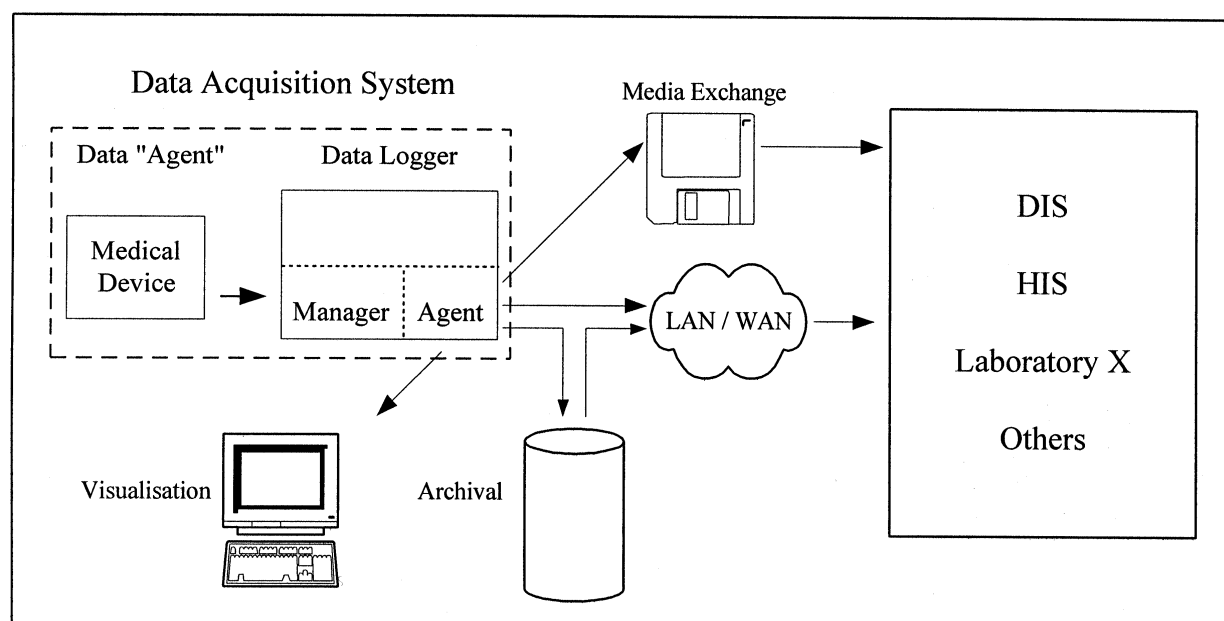


Figure 6.1: Data Acquisition System (non time-critical)

Communication parties represented in Figure 6.1 are the data acquisition system consisting of medical device and data logger and any other information processing system (DIS, HIS, etc.). Within the closed (data acquisition) system pre-processing, measurement, etc. may take place. The data acquisition process and the communication with an external party usually run independently of each other. Analysis and design of the communication process shall facilitate the transfer of acquired information to the other systems.

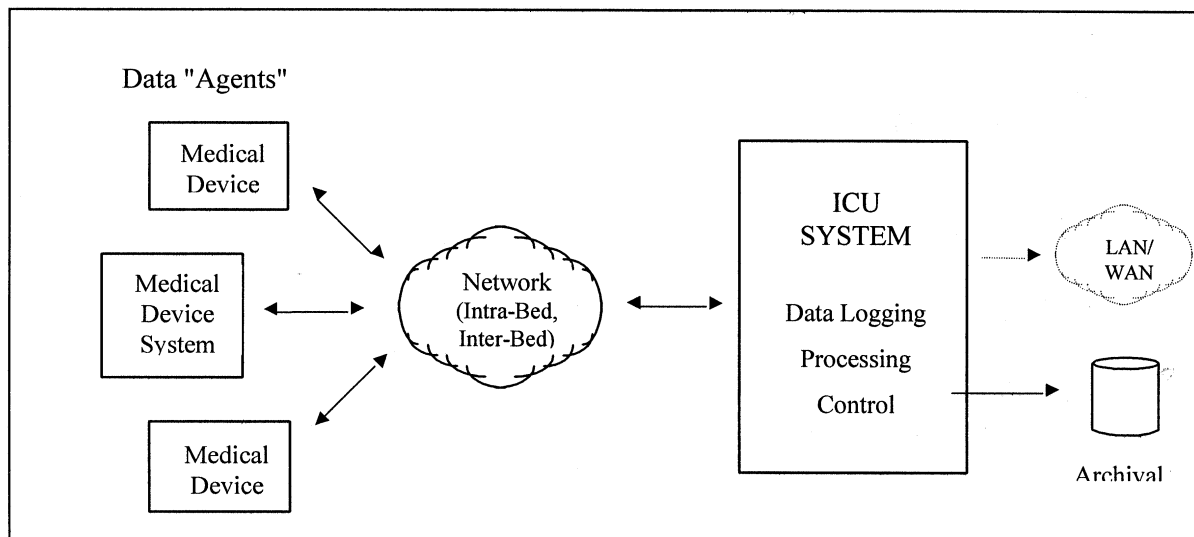


Figure 6.2: Data Acquisition in a Real-Time Environment

Figure 6.2 shows, that communication partners are primarily the medical devices/device systems and the Intensive Care Unit (ICU) system. The ICU system may acquire and process data, may control the flow of data from the devices, and may interact with the devices for function control (e.g. alarm settings, flow rate adjustment). Medical devices and ICU system form as communication partners an *open system*. Analysis and design of their communication processes shall facilitate their interoperability. ICU systems typically provide a means for archiving of data.

7 Domain Information Model

7.1 General

7.1.1 Modelling Concept

The Domain Information Model (DIM) is an object oriented model that consists of objects, their attributes and their methods which are abstractions of real world entities in the domain of (vital signs information communicating) medical devices.

The information model as well as the service model for communicating systems defined and used in this standard are conceptually based on the ISO/OSI systems management model. Objects defined in the information model are considered managed (medical) objects. For the most part they are directly available to management (access) services provided by the Common Medical Device Information Service Element (CMDISE) as defined in this standard.

For communicating systems, the set of object instances available on any medical device that complies with the definitions of this standard forms the Medical Data Information Base. The MDIB is a structured collection of managed medical objects representing the vital signs information provided by a particular medical device. Attribute data types, hierarchies and behaviour of objects in the MDIB are defined in this standard.

The majority of objects defined here represent generalized vital signs data and support information. Specialization of these objects is achieved by defining appropriate attributes. Object hierarchies and relations between objects are used to express device configuration and device capabilities.

EXAMPLE: A generalized object is defined to represent vital signs in the form of a real-time waveform. A set of object attributes is used to specify a particular waveform as an Invasive Arterial Blood Pressure. The position in the hierarchy of all objects defines the sub-system that derives the waveform.

Figure 7.1 shows the relation between managed (medical) objects, MDIB, CMDISE, application processes and communication systems.

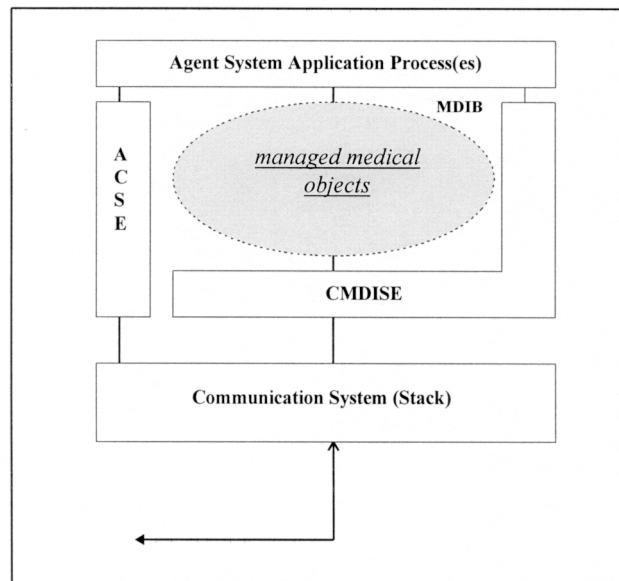


Figure 7.1: Medical Data Information Base in Communicating Systems

In the case of communicating systems, managed medical objects are only accessible through services provided by CMDISE. The way that these objects are stored in the MDIB in any specific system and the way applications and the CMDISE access these objects is an implementation issue and as such not normative.

In the case of a vital signs archived data format that complies with the definitions in this standard, object instances are stored, together with their dynamic attribute value changes, over a certain time period on archival media. Attribute data types and hierarchies of objects in the archival data format are again defined in this standard.

Figure 7.2 shows the relationship between managed medical objects, the data archive and archive access services.

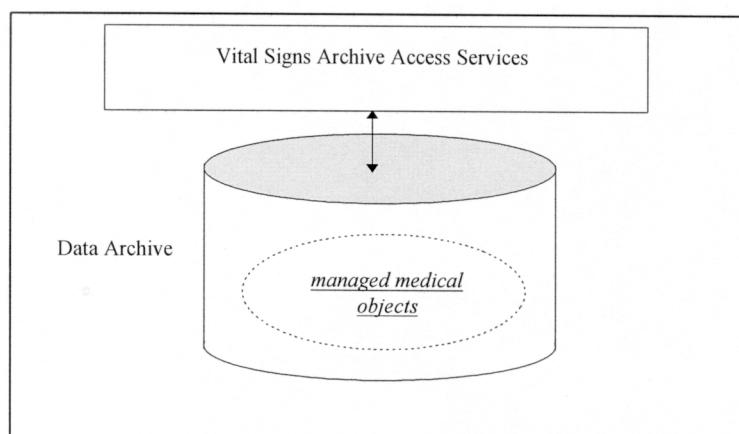


Figure 7.2: Managed Medical Objects in a Vital Signs Archive

In the case of the archived data format the way managed medical objects are stored on a medium is the subject of standardization. The access services are a local implementation issue and as such are not intended to be governed by this standard.

7.1.2 Scope of the Domain Information Model

7.1.2.1 General

Vital signs information objects that are defined in this standard encompass digitized biosignals that are derived by medical measurement devices used for example in anaesthesia, surgery, infusion therapy, intensive care and obstetrical care.

Biosignal data within the scope include direct and derived, quantitative and qualitative measurements, technical and medical alarms and control settings. Patient information relevant for the interpretation of these signals is also defined in the Domain Information Model.

7.1.2.2 Communicating Systems

Communicating systems within the scope of this standard include physiological meters and analysers, especially those providing real-time or continuous monitoring. Data processing capabilities are required for these systems.

Information management objects that provide capabilities and concepts for cost effective communication (specifically data summarization objects) and objects necessary to enable real-time communication are also within the scope of the information model in this standard.

Interoperability issues, specifically lower communication layers, temporal synchronization between multiple devices etc. are outside the scope of this standard.

7.1.2.3 Archived Vital Signs

Context information objects that describe the data acquisition process and organize a vital signs archive are within the scope of the standard.

7.1.3 Approach

For the object oriented modelling the technique described by P. Coad and E. Yourdon is used. The domain is first subdivided into different subjects which permits the organization of the model into smaller packages. Each subject is then defined in the form of object diagrams. Objects are briefly introduced, and their relations and hierarchies are defined in the object diagram.

For the object definitions a textual approach is followed. Attributes are defined in attribute definition tables. Attribute data types are defined using ASN.1. Object behaviour and notifications generated by objects are also defined in definition tables. These definitions directly relate to the service model specified in the standard.

7.1.4 Extension of the Model

It is expected that over time extensions of the model may be needed to account for new developments in the area of medical devices. Also, in special implementations there may be a requirement to model data that are specific for a particular device or a particular application (and that is therefore not covered by the general model).

In some cases it may be possible to use the concept of *external object relations*. Most objects defined in this standard provide an attribute group (the Relationship Attribute Group) that can be used to supply information about related objects that are not defined in the Domain Information Model. This can be done by specifying a relation to an external object and assigning attributes to this relation (see DIM object definitions, External Object Relation List).

In other cases it may be necessary to define completely new objects or to add new attributes, new methods or new events to already defined objects. These extensions are considered private or manufacturer specific extensions. Dealing with these extensions is primarily a matter of an interoperability standard that is based on this standard on vital signs representation.

In general, in an interoperability format (message definitions) objects, attributes and methods are identified by nomenclature codes. The nomenclature code space (code values) leaves room for private extensions. As a general rule, an interoperability standard that is based on this Domain Information Model should be able to deal with private or manufacturer specific extensions by ignoring objects, attributes etc. with unknown identifiers (nomenclature codes).

7.2 Subject Diagram - Overview

The subject diagram organizes the problem domain into packages. It shows the major objects inside each subject and defines the relationships between these subjects.

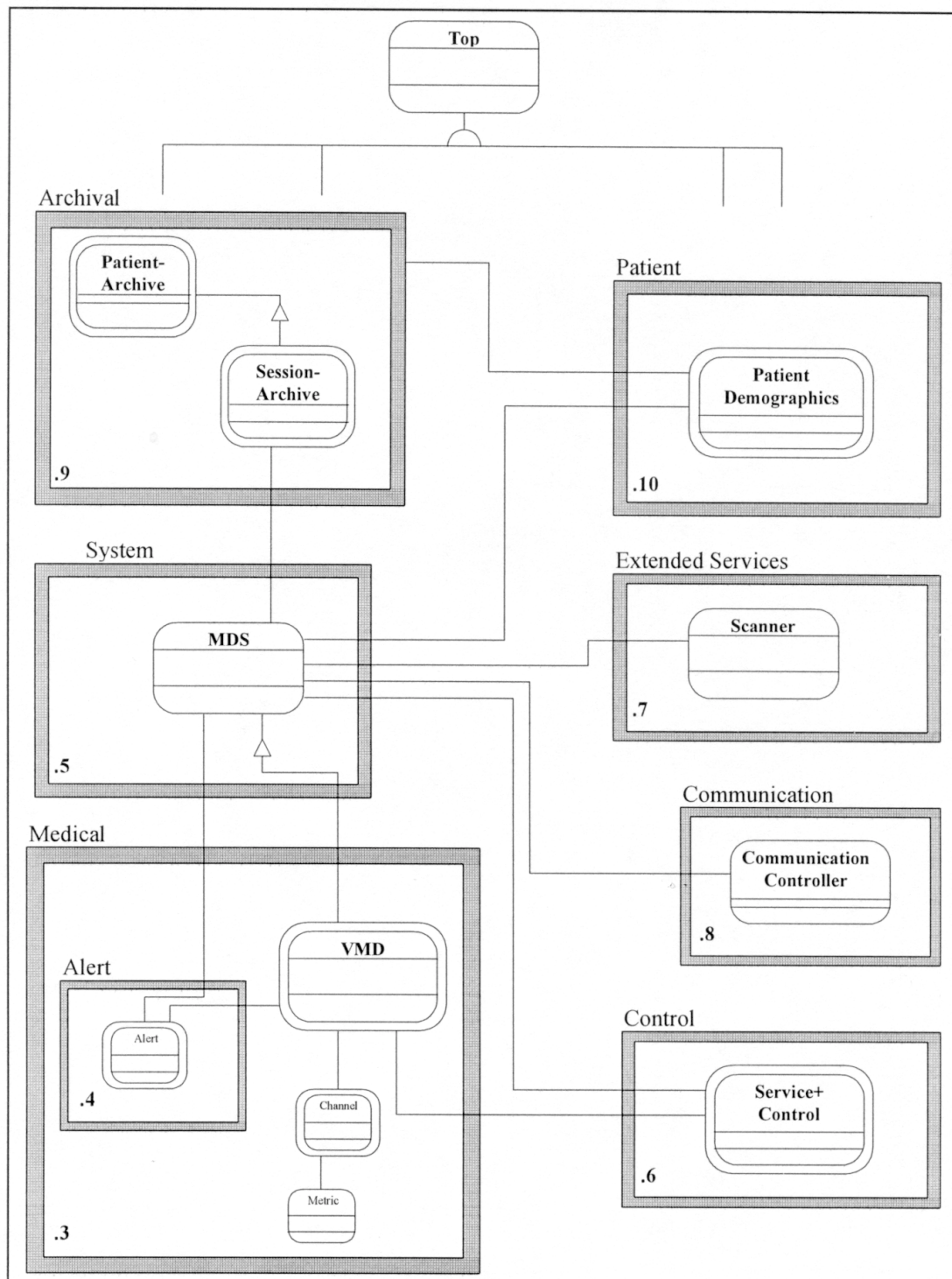


Figure 7.3: Subjects of the Domain Information Model

The subject diagram depicted in Figure 7.3 contains only a small subset of all objects defined in the DIM. Common base objects except Top are not shown in this diagram. Also, not all relations are shown between the different subject areas. Refer to the detailed subject diagrams for more information.

The numbers in the subjects refer to the corresponding subclauses in this model clause and in the object definition clause.

The Top object is an abstract base class and at the same time the ultimate base class for all objects defined in the model. For editorial convenience the modelling diagrams in the document do not show this inheritance hierarchy.

The following subclauses contain the more detailed models for these subjects.

7.3 Model for the Medical Subject

The medical subject deals with the derivation and representation of biosignals and contextual information that is important for the interpretation of measurements.

Figure 7.4 shows the object model of the medical subject.

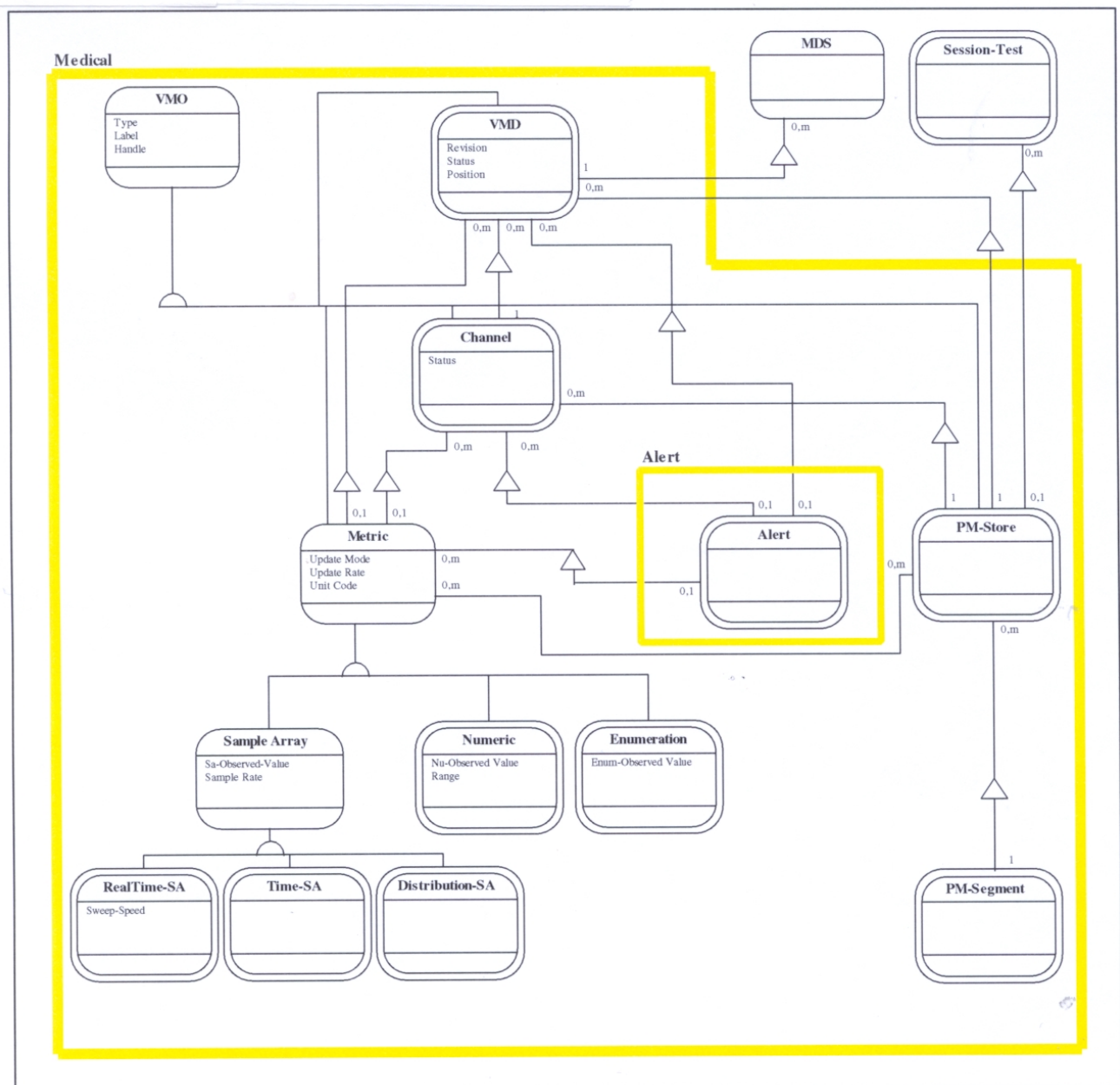


Figure 7.4: Medical Subject Model²

The medical subject model contains the following objects:

² Instances of the Channel object and the Persistent Metric (PM)-Store object shall be contained in exactly one superior object instance. Refer to the Alert subject for information about alarm related objects.

Alarming is considered a medical function, and is therefore contained in the Medical Subject Model. This figure does not show all possible relations between objects in the Medical Subject and the objects in the Alert Subject. The reader is advised to consult the Alert Subject for details.

7.3.1 Virtual Medical Object

The Virtual Medical Object (VMO) is the base class for all medical related objects in the model. It provides consistent naming and identification across the medical subject model.

As a base class, it cannot be instantiated.

7.3.2 Virtual Medical Device

The Virtual Medical Device (VMD) is an abstraction for a medical related subsystem (hardware or even pure software) of a medical device. Characteristics of this subsystem (modes, versions etc.) are captured in this object. At the same time the VMD is a container for objects representing measurement and status information.

EXAMPLE: A modular patient monitor provides measurement modalities in the form of plug in modules. Each module is represented by a VMD object.

7.3.3 Channel

The Channel object is used for grouping metric objects and thus, allows hierarchical information organization. The Channel object is not mandatory for representation of Metric objects in a VMD.

EXAMPLE: A blood pressure VMD may define a Channel object to group together all metrics that deal with the blood pressure (pressure value, pressure waveform). A second Channel object can be used to group together metrics that deal with heart rate.

7.3.4 Metric

The Metric object is the base class for all objects representing direct and derived, quantitative and qualitative biosignal measurement, status and context data.

Specializations of the Metric base class are provided to deal with common *representations* (e.g. single values, array data, status indications) and *presentations* (e.g. on a display) of measurement data.

As a base class, it cannot be instantiated.

7.3.5 Numeric

The Numeric object represents numerical measurements and status information like amplitude measures, counters etc.

EXAMPLE: A heart rate measurement is represented by a numeric object.

NOTE: A *compound* Numeric is defined as an efficient model for e.g. arterial blood pressure which usually has three associated values (systolic, diastolic, mean). The availability of multiple values in a single Numeric (or other Metric) object can be indicated in a special *structure* attribute in the metric object.

7.3.6 Sample Array

The Sample Array (SA) object is the base class for metrics that have a graphical, *curve* type presentation and therefore have their observation values reported as arrays of data points by communicating systems.

As a base class, the Sample Array cannot be instantiated.

7.3.7 Real Time Sample Array

The Real Time Sample Array object is a sample array that represents a real-time continuous waveform. As such it has special requirements in communicating systems, e.g. processing power, low latency and high bandwidth. Therefore it requires the definition of a specialized object.

EXAMPLE: An ECG real-time wave is represented as a Real Time Sample Array object.

7.3.8 Time Sample Array

The Time Sample Array object is a sample array that represents non continuous waveforms (a wave snippet). Within a single observation (a single array of sample values), samples are equidistant in time.

EXAMPLE: Software for ST segment analysis may use the Time Sample Array to represent snippets of ECG real-time waves that just contain a single QRS complex. Within this wave snippet, the software can locate the ST measurement points. It generates a new snippet e.g. every 15 seconds.

7.3.9 Distribution Sample Array

The Distribution Sample Array object is a sample array that represents linear value distributions in the form of arrays containing scaled sample values. The index of a value within an observation array denotes a spatial value, not a time point.

EXAMPLE: An EEG application may use a Fourier transformation to derive a frequency distribution (a *spectrum*) from the EEG signal. It then uses the Distribution Sample Array to represent that spectrum in the MDIB.

7.3.10 Enumeration

The Enumeration object represents status information and/ or annotation information. Observation values may be presented in the form of normative codes (that are included in the nomenclature defined in this standard or in some other nomenclature scheme) or in the form of free text.

EXAMPLE: An ECG rhythm qualification may be represented as an enumeration metric. A ventilator may provide information about its current ventilation mode as an Enumeration object.

7.3.11 Persistent Metric (PM)-Store

The Persistent Metric Store object provides long term storage capabilities for metric data. It contains a variable number of PM-Segment objects that can be accessed only through the PM-Store. Without further specialization the PM-Store is intended to store data of a single metric object only.

EXAMPLE: A device stores the numerical value of an invasive blood pressure on a disk. It uses the PM-Store object to represent this persistent information. Attributes of the PM-Store describe the sampling period, the sampling algorithm and the storage format. Whenever the label of the pressure measurement is changed (e.g. during a wedge procedure) the storage process opens a new PM-Segment to store the updated context data (here: the label).

7.3.12 Persistent Metric (PM)-Segment

The Persistent Metric Segment object represents a continuous time period in which a metric is stored without any changes of relevant metric context attributes (scales, labels etc.).

The PM-Segment is only accessible through the PM-Store object (e.g. for retrieving stored data, the PM-Store object has to be accessed).

7.4 Model for the Alert Subject

The Alert subject deals with objects that represent status information about patient condition and/or technical conditions influencing the measurement or device functioning. Alert related information is often subject to normative regulations and therefore requires special handling.

In the model all alarm related object oriented items are identified by the term *alert*. The term *alert* is used in this document as a synonym for the combination of patient-related physiological alarms, technical alarms and equipment user advisory signals.

An alarm is a signal which indicates abnormal events occurring to the patient or the device system. A physiological alarm is a signal which either indicates that a monitored physiological parameter is out of specified limits or indicates an abnormal patient condition. A technical alarm is a signal which indicates that a device system is not capable of accurately monitoring the patient's condition or is no longer monitoring the patient's condition.

The model defines three different levels of alarming. These levels represent different sets of alarm processing steps, ranging from a simple context free alarm event detection to an intelligent device system alarm process. This process is required to prioritize all device alarms, to latch alarms if needed (a latched alarm does not stop when the alarm condition goes away) and to produce audible and visual alarm indications for the user.

For consistent system wide alarming a particular medical device may either provide no alarming capability or exactly one level of alarming which is dependent on the capabilities of the device. Each level is represented by one specific object class. This means that either zero or one alarm object class (only Alert or only Alert Status or only Alert Monitor, no combinations) is instantiated in the device containment tree. Multiple instances of a class are allowed.

NOTE: Medical device alarming is subject to various national and international safety standards (e.g. IEC 601 series, ISO 9703 series). Considering requirements of current safety standards, objects in this standard define information contents only. Any implementation shall therefore follow appropriate standards for dynamic alarming behaviour.

Figure 7.5 shows the object model of the Alert subject.

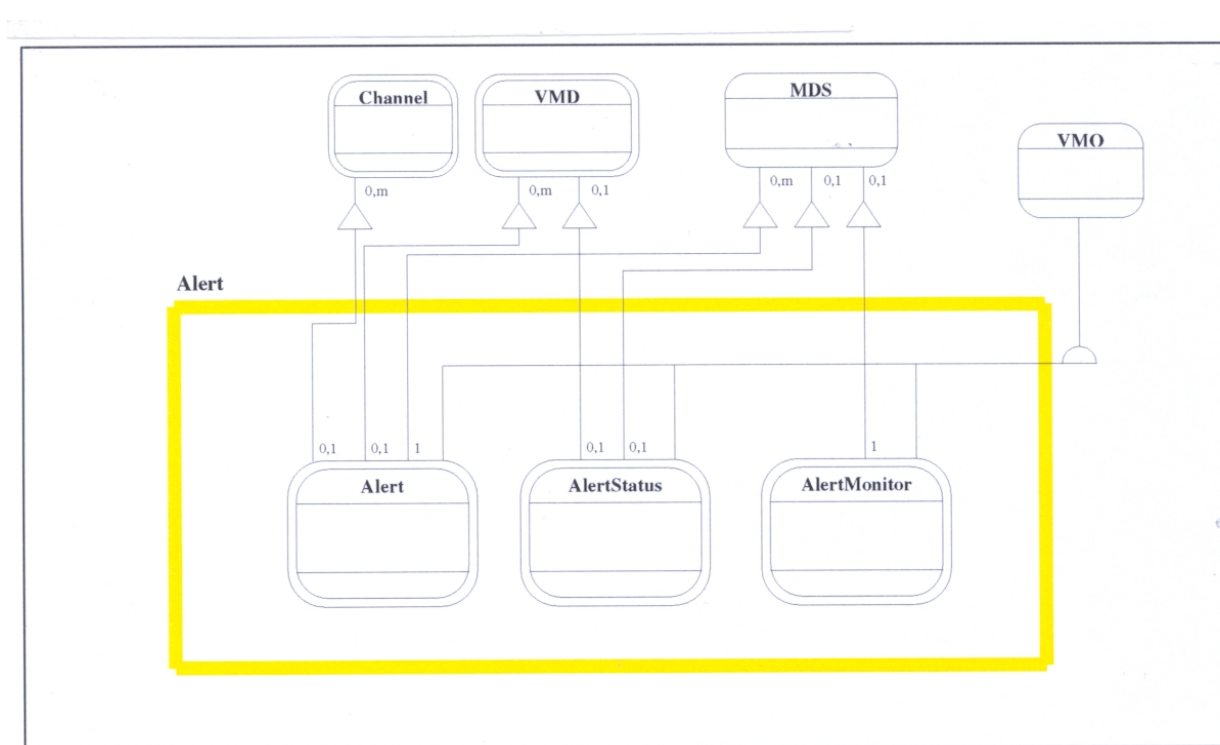


Figure 7.5: Alert Subject Model³

³ Instances of objects in the alert subject area shall be contained in exactly one superior object.

7.4.1 Alert

The Alert object stands for the status of a simple alarm condition check. As such it represents a single alarm only. The alarm can either be a physiological alarm or a technical alarm condition of a related (MDS, VMD, Metric) object. If a device instantiates an Alert object it shall not instantiate the Alert Status or the Alert Monitor objects. A single Alert object is needed for each alarm condition that the device is able to detect.

The Alert object has a reference to an object instance in the medical subject to which the alarm condition relates.

Note: An Alert object instance is not dynamically created or deleted in cases where alarm conditions start or stop. Rather an existing Alert object instance changes attribute values in these cases.

EXAMPLE: An Alert object may represent the status of a process that checks for a limit violation physiological alarm of the Heart Rate signal. In case of a violation of the limit the object generates an event (attribute update) that represents this alarm condition in the form of attribute value changes.

7.4.2 Alert Status

The Alert Status object represents the output of an alarm process that considers all alarm conditions in a scope that spans one or more objects. In contrast to the Alert object it collects all alarm conditions related to a VMD object hierarchy or related to an MDS object and provides this information in list- structured attributes. This allows the implementation of first level alarm processing where knowledge about the VMD or MDS can be used to prioritize alarm conditions and to suppress known false alarm indications.

For larger scale devices without complete alarm processing the Alert Status object greatly reduces the overhead of a large number of Alert object instances.

If a device instantiates an Alert Status object it shall not instantiate the Alert or the Alert Monitor objects. Each VMD or MDS in the MDIB is able to contain at most one Alert Status object instance.

EXAMPLE: An ECG VMD derives a Heart Rate value. As the VMD is able to detect that the ECG leads are disconnected from the patient its Alert Status object reports only a technical alarm and suppresses a Heart Rate limit violation alarm in this case.

7.4.3 Alert Monitor

The Alert Monitor object represents the output of a device or system alarm processor and as such it represents the overall device or system alarm condition. It provides a list of all alarm conditions of the system in its scope. This includes global state information and individual alarm state information that allows the implementation of a safety standard compliant alarm display on a remote system.

If a device instantiates an Alert Monitor object it shall not instantiate the Alert or the Alert Status object. An MDS shall contain not more than one Alert Monitor object instance.

EXAMPLE: A patient monitoring system provides alarm information in the form of an Alert Monitor object to a central station. Alert information include the current global maximum severity audible and visual alarm conditions on the monitor display. The alarm processor operates in a latching mode where physiological alarm conditions are buffered until they are explicitly acknowledged by a user.

7.5 Model for the System Subject

The System subject deals with the representation of devices that derive or process vital signs information and comply with the definitions in this standard.

Figure 7.6 shows the object model for the System subject:

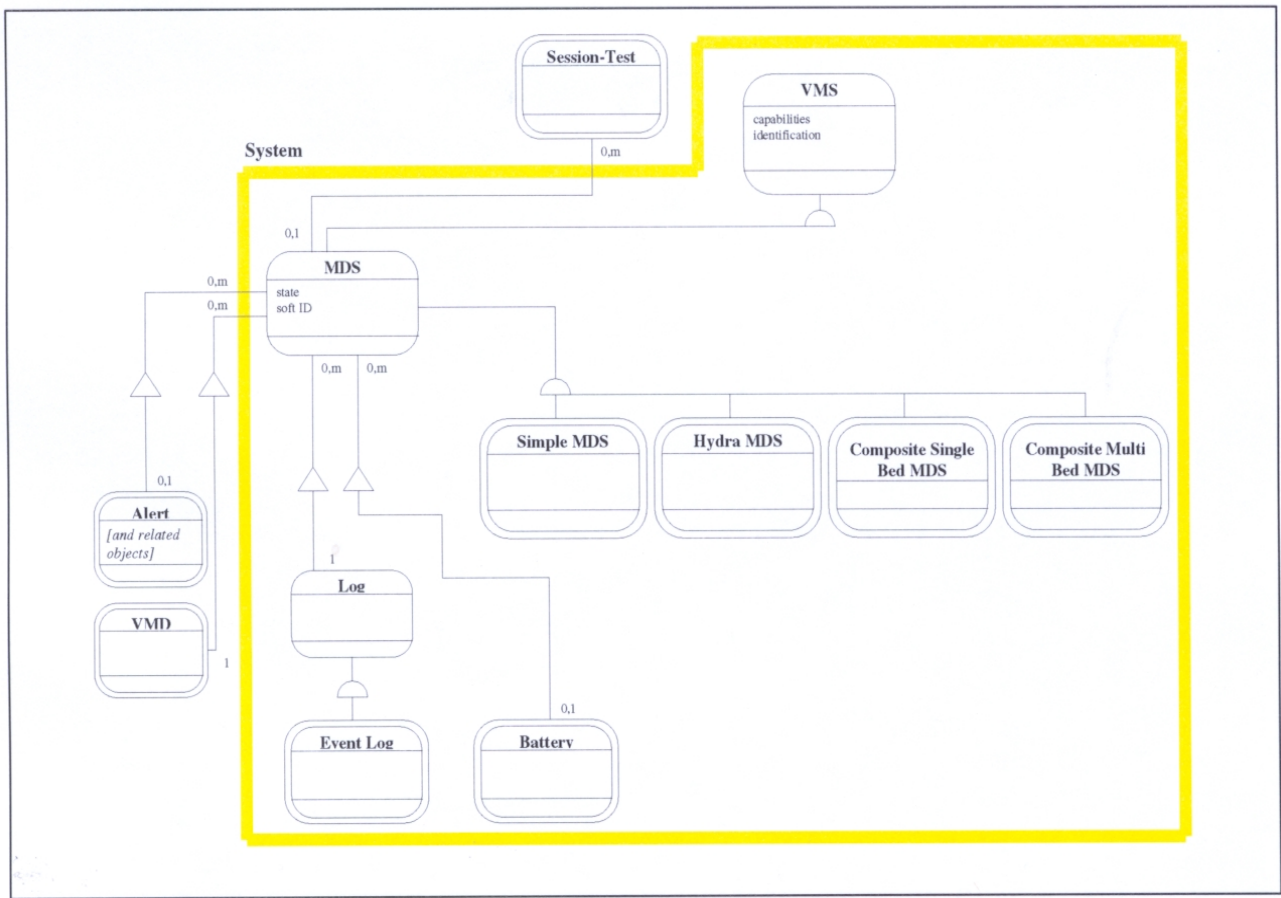


Figure 7.6: System Subject Model

The model contains the following objects:

7.5.1 VMS

The Virtual Medical System (VMS) is the abstract base class for all System type objects in this model. It provides consistent naming and identification of system related objects.

As a base class, the VMS cannot be instantiated.

7.5.2 MDS

The Medical Device System object (MDS) is an abstraction of a device that provides medical information in the form of objects that are defined in the Medical subject of the Domain Information Model.

The MDS is the top-level object in the device's MDIB and represents the instrument itself. Further specializations of this class are used to represent differences in complexity and scope.

As a base class, the MDS cannot be instantiated.

7.5.3 Simple MDS

The Simple MDS object represents a medical device that contains a single VMD instance only (single purpose device).

7.5.4 Hydra MDS

The Hydra MDS object represents a device that contains multiple VMD instances (multi- purpose device).

7.5.5 Composite Single Bed MDS

The Composite Single Bed MDS represents a device that contains (interfaces) one or more Simple or Hydra MDS objects at one (bed) location.

7.5.6 Composite Multiple Bed MDS

The Composite Multiple Bed MDS represents a device that contains (interfaces) multiple MDS objects at multiple (bed) locations.

7.5.7 Log

The Log object is a storage container for important local system notifications and events. It is possible to define specialized classes for specific event types.

As a base class, the Log object cannot be instantiated.

7.5.8 Event Log

The Event Log object is a general log object that stores system events in a free text representation.

EXAMPLE: An infusion device may want to keep track of mode and rate changes by remote systems. Whenever a remote operation is invoked, it creates an entry in its Event Log.

7.5.9 Battery

For battery powered devices, some battery information is contained in the MDS in the form of attributes. If the battery sub-system is either capable of providing more information (smart battery) or is *manageable*, then a special Battery object is provided.

7.6 Model for the Control Subject

The Control subject contains objects that allow remote measurement control and device control.

The model for remote control defined in this standard provides the following benefits:

- A system that allows remote control is able to explicitly register which attributes or features it is possible to access or modify by a remote system
- For attributes that it is possible to remotely modify, a list of possible legal attribute values is provided to the controlling system.
- It is not mandatory that a remote controllable item corresponds to an attribute of an (e.g. medical) object.
- Dependence of a controllable item on internal (system) states is modelled.
- A simple locking/ transaction scheme allows to handle transient states during remote control

There are at least two different uses of remote control which are considered:

- Automatic control may be done by some process running on the controlling device. This process has to be able to discover automatically how it can modify/ access the controllable items to provide it's function.
- It is also possible to use remote control to present some form of control interface to a human operator. For this, descriptions of functions, and possibly help information, need to be provided.

The basic concept presented here is based on *Operation objects*. An Operation object allows modification of a *virtual attribute*. This virtual attribute may e.g. be a measurement label, a filter state (on/off), a gain factor etc. The attribute is called *virtual* because it need not correspond to any attribute in other objects instantiated in the system.

Different specializations of the Operation object define *how* the virtual attribute is modified. A *Select Item operation* e.g. allows to choose from a given list of possible values for the attribute. A *Set Value* operation allows setting of the attribute to a value from a defined range with a specific step width (resolution).

The idea is that the Operation object provides all necessary information about legal attribute values. Furthermore, the Operation defines various form of text string to support a human user of the operation. It also contains grouping information that allows logical grouping of multiple Operation objects together when they are presented as part of a human interface.

Operation objects cannot directly be accessed by services defined in the Service model. Instead, all controls shall be routed through the Service and Control object (SCO). This object supports a simple locking mechanism to prevent side effects caused by simultaneous calls.

The SCO groups together all Operation objects that belong to a specific entity (MDS/ VMD). The SCO also allows feedback to a controlled device, e.g. for a visual indication that the device is currently remote controlled.

Figure 7.7 shows the object model for the Control subject:

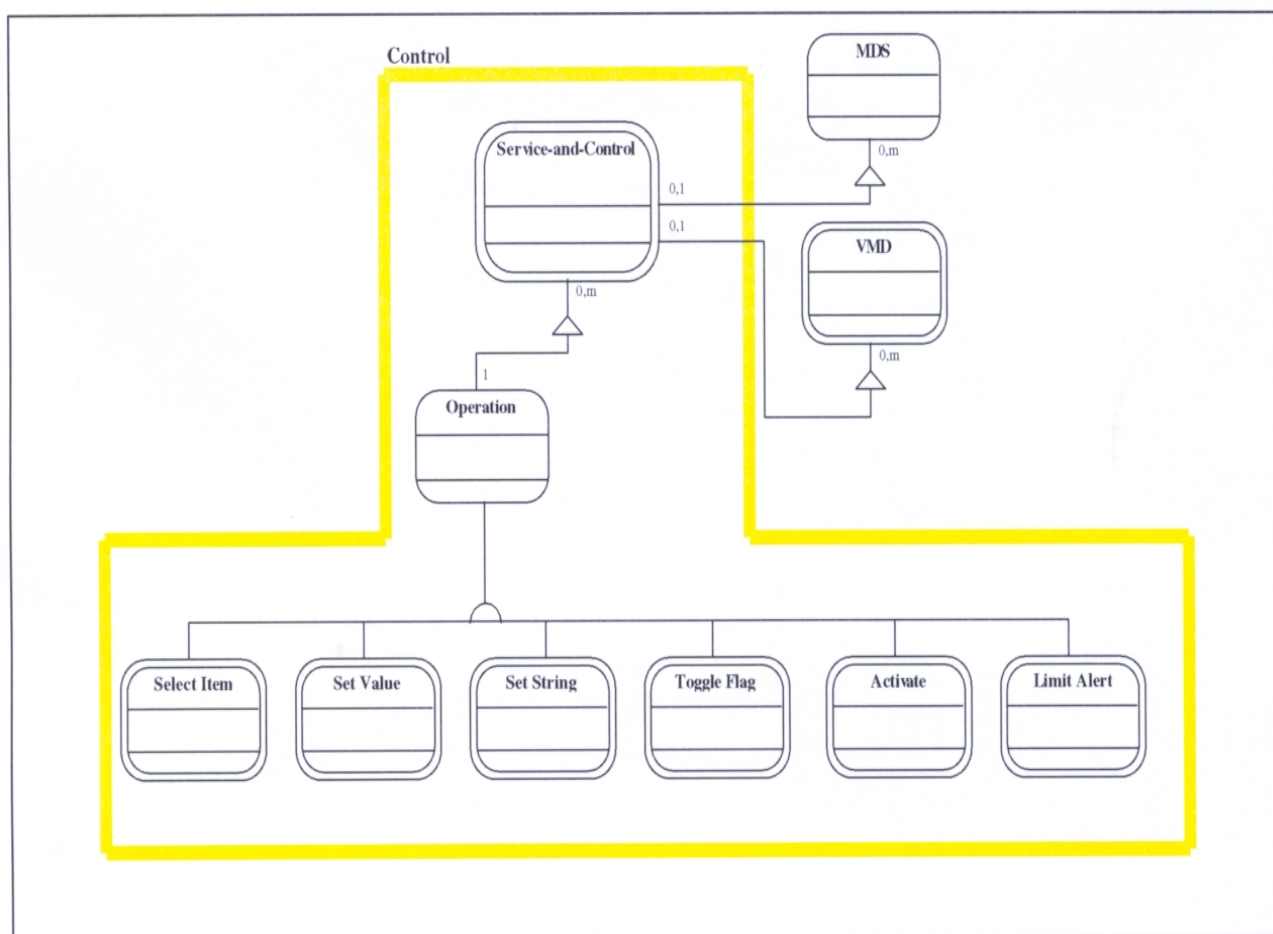


Figure 7.7: Control Subject Model

The model contains the following objects:

7.6.1 Service and Control

The Service and Control object (SCO) is responsible for managing all remote control capabilities that are supported by a medical device.

Remote control in medical device communication is a safety and security sensitive area. The SCO provides means for:

- a) Simple transaction processing.
 This prevents inconsistencies when a device is controlled from multiple access points (local and remote) and during processing of control commands.
- b) State indications.
 This allows local and remote indication of ongoing controls.

7.6.2 Operation

The Operation class is the (abstract) base class for classes that represent remote controllable items. Each Operation object allows to modify some specific item (a *virtual* attribute) in a specific way defined by the Operation object. Operation objects are not directly accessible by services defined in the service model. All controls shall be routed through the (parent) SCO object to allow a simple form of transaction processing.

The set of Operation objects instantiated by a particular medical device defines the complete (remote) control interface of the device. This way a host system is able to discover the remote control capabilities of a device in the configuration phase.

7.6.3 Select Item Operation

The Select Item operation allows to select one item out of a given list.

EXAMPLE: The Invasive Pressure VMD may allow modification of its label. It uses a Select Item operation for this. The list of legal values supplied by the operation may be e.g. {ABP, PAP, CVP, LAP}. By invoking the operation, a user is able to select one value out of this list.

7.6.4 Set Value Operation

The Set Value operation allows to adjust a value within a given range with a given resolution.

EXAMPLE: A measurement VMD may allow adjustment of a signal gain factor. It uses the Set Value operation for this. The operation provides the supported value range and step width within this range.

7.6.5 Set String Operation

The Set String operation allows to set the contents of an opaque string variable of a given maximum length and format.

EXAMPLE: An infusion device may allow a remote system to set the name of the infused drug in free text form. It defines an instance of the Set String operation for this function. The operation specifies the maximum string length and the character format, so that the device is able to show the drug name on a small display.

7.6.6 Toggle Flag Operation

The Toggle Flag operation allows operation of a toggle switch (with two states, e.g. on/off).

EXAMPLE: An ECG VMD may support a line frequency filter. It uses the Toggle Flag operation for switching the filter on or off.

7.6.7 Activate Operation

The Activate operation allows a defined activity to be started (e.g. a zero pressure).

EXAMPLE: The zero procedure of an invasive pressure VMD may be started with an Activate operation.

7.6.8 Limit Alert Operation

The Limit Alert operation allows adjustment of the limits of a limit alarm detector and to switch the limit alarm on or off.

7.7 Model for the Extended Services Subject

The Extended Services subject contains objects that provide extended (medical) object management services that allow efficient access to medical information in communicating systems. This is achieved by a set of objects that package attribute data from multiple objects in a single event message and by another object that allows to specify a configurable filter for event messages that are generated by objects in the MDIB.

The objects providing extended services are conceptually derived from ISO/OSI system management services defined in the ISO/IEC 10164 series of standards (specifically parts 5 and 13). The definitions have been adapted to and optimized for specific needs in the area of vital signs communication between medical devices.

Figure 7.8 shows the object model for the Extended Services subject.

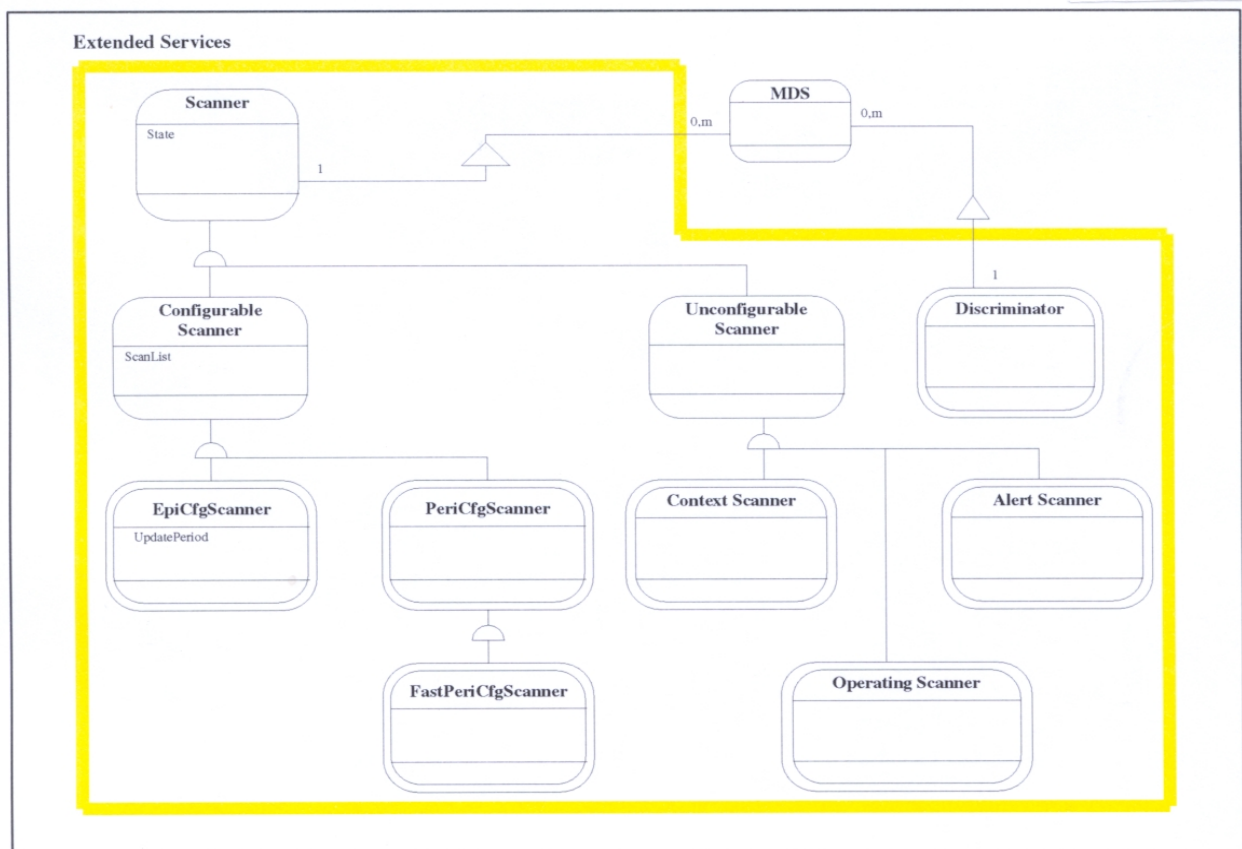


Figure 7.8: Extended Services Subject Model

The model contains the following objects:

7.7.1 Scanner

A Scanner object is an observer and "summarizer" of object attribute values. It observes attributes of managed (medical) objects and generates summaries in the form of notification event reports. These event reports contain data from multiple objects which provides a better communication performance compared to separate polling commands (GET service) or multiple individual event reports from all object instances.

Objects derived from the Scanner class may either be instantiated by the agent system itself or by the manager system (dynamic scanner creation by using the CREATE service).

As a base class, the Scanner object cannot be instantiated.

7.7.2 Configurable Scanner

The Configurable Scanner has a special attribute (the *ScanList* attribute) that allows to configure which object attributes are scanned. The ScanList may be modified either by the agent system (auto-configuration or pre-configuration) or by the manager system (full dynamic configuration by using the SET service).

A Configurable Scanner may support different granularity for scanning:

- a) Attribute Groups (defined sets of attributes)

The ScanList contains the Ids of attribute groups, all attributes in the group are scanned.

- b) Individual attributes

The ScanList contains the Ids of all attributes that are scanned.

In order to deal efficiently with optional object attributes the attribute group scan granularity is recommended for Configurable Scanners.

As a base class, the Configurable Scanner cannot be instantiated.

7.7.3 Episodic Configurable Scanner

The Episodic Configurable Scanner (EpiCfgScanner) is responsible for observing attributes of managed medical objects and for reporting attribute changes in the form of *unbuffered* event reports.

The unbuffered event report is triggered only by object attribute value changes. If the EpiCfgScanner uses attribute groups as the granularity for scanning, the event report contains all attributes of the scanned object that belong to this attribute group if one or more of these attributes changed their value.

EXAMPLE: A medical device provides heart beat detect events in the form of an Enumeration object. A display application creates an instance of the EpiCfgScanner object and adds the observed value of the Enumeration object to the ScanList. The scanner instance afterwards sends a notification whenever the Enumeration object reports a heart beat.

7.7.4 Periodic Configurable Scanner

The Periodic Configurable Scanner (PeriCfgScanner) is responsible for observing attributes of managed medical objects and for periodically reporting attribute values in the form of *buffered* event reports. A buffered event report contains the attribute values of all available attributes that are specified in the scan list, independent of attribute value changes.

If the Scanner operates in a special *superpositive* mode, the buffered event report contains all value changes of attributes that occurred in the reporting period, otherwise the report only contains the most recent attribute values.

EXAMPLE: A data logger creates an instance of the PeriCfgScanner object and configures the Scanner so that it sends an update of the observed value attributes of all Numeric objects in the MDIB every 15 seconds.

7.7.5 Fast Periodic Configurable Scanner

The Fast Periodic Configurable Scanner (FastPeriCfgScanner) is a specialized object class for scanning the observed value attribute of the RT-SA object. This special scanner object is further optimized for low latency reporting and efficient communication bandwidth utilization which is required to access real-time waveform data.

EXAMPLE: A real-time display application (manager system) wants to display ECG waveforms. It creates a FastPeriCfgScanner on the agent system (server device) and requests periodic updates of all ECG leads.

7.7.6 Unconfigurable Scanner

The Unconfigurable Scanner object scans a pre- defined set of managed medical objects which cannot be modified. This means that an Unconfigurable Scanner typically is a reporting object that is specialized for one specific purpose.

As a base class, the Unconfigurable Scanner cannot be instantiated.

7.7.7 Context Scanner

The Context Scanner is responsible for observing device configuration changes. After instantiation, the Context Scanner is responsible for announcing the object instances in the device's MDIB. The scanner provides the object instance containment hierarchy and static object attribute values.

In case of dynamic configuration changes the Context Scanner sends notifications about new object instances or deleted object instances.

EXAMPLE: A data logger creates an instance of the Context Scanner object in an agent MDIB to receive notifications about MDS configuration changes when new measurement modules are plugged in (new VMD instance) or when such a module is unplugged (VMD instance deleted).

7.7.8 Alert Scanner

The Alert Scanner is responsible for observing the alert related attribute groups of objects in the Alert subject. As alarming in general is security sensitive, the scanner is not configurable (all or no alert objects are scanned).

The Alert Scanner sends event reports periodically, so that time-out conditions can be checked.

7.7.9 Operating Scanner

The Operating Scanner is responsible for providing all information about the operating and control system of a medical device.

This means the scanner maintains the configuration of Operation objects contained in SCO objects (sending CREATE notifications for Operation objects), it scans transaction handling related SCO attributes and it scans Operation object attributes. Because SCO objects and Operation objects may have dependencies, the scanner is not configurable.

7.7.10 Discriminator

The Discriminator object controls the forwarding of events generated by objects in the MDIB between communicating systems.

EXAMPLE: A data logging application in a manager device does not need to receive attribute update event reports from the MDS object. It creates a Discriminator object to suppress forwarding of these events over the communication link to conserve bandwidth.

7.8 Model for the Communication Subject

The Communication subject deals with objects that enable and support basic communication.

Figure 7.9 shows the object model for the Communication subject:

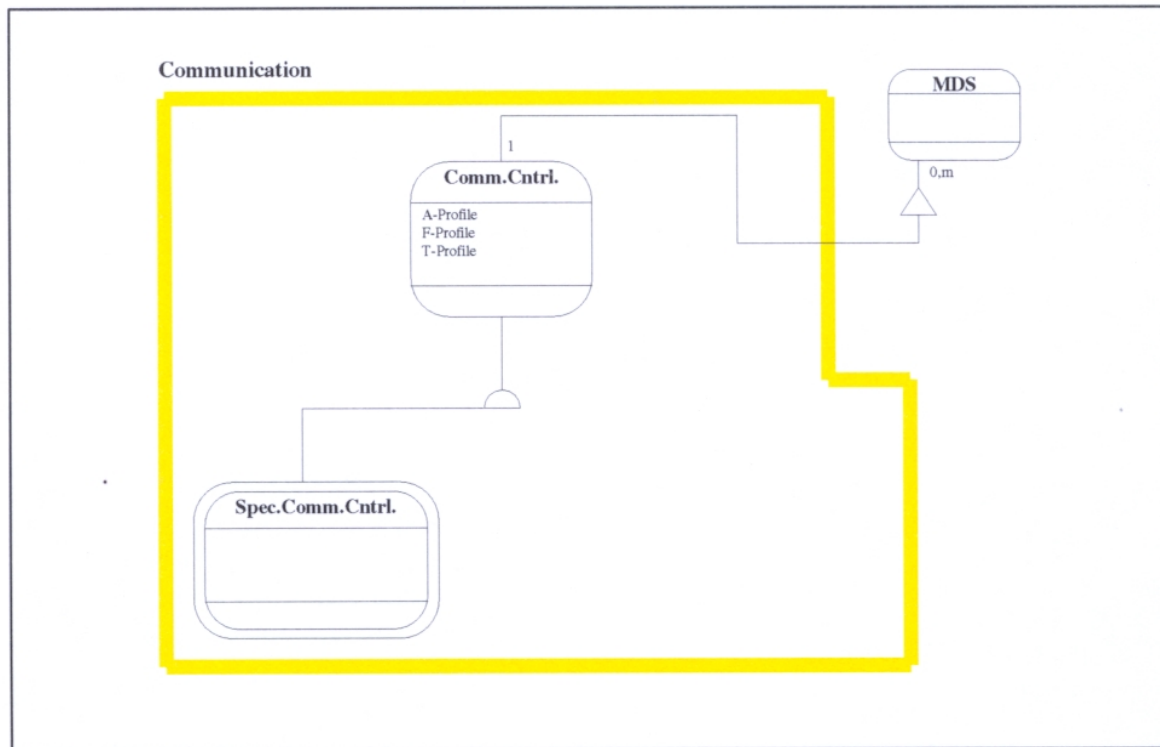


Figure 7.9: Communication Subject Model

The model contains the following objects:

7.8.1 Communication Controller

The Communication Controller represents the upper layer and lower layer communication profile of a medical device. It contains information about the application profile (CMDISE), the format profile (message formats) and the transport profile (specifics of the lower layer implementation).

As a base class, the Communication Controller cannot be instantiated.

7.8.2 Specialized Communication Controller

Dependent on the communication software (upper layer and lower layer *stack*) and communication hardware different specializations of the Communication Controller are needed.

NOTE: Specializations of the Communication Controller object are not within the scope of this standard. It is the role of an interoperability standard based on this model to define Communication Controller specializations (representing the set of valid profiles).

7.9 Model for the Archival Subject

The Archival subject deals with storage and representation of biosignals, status and context information in an on-line or an off-line archive.

Figure 7.10 shows the object model of the Archival subject.

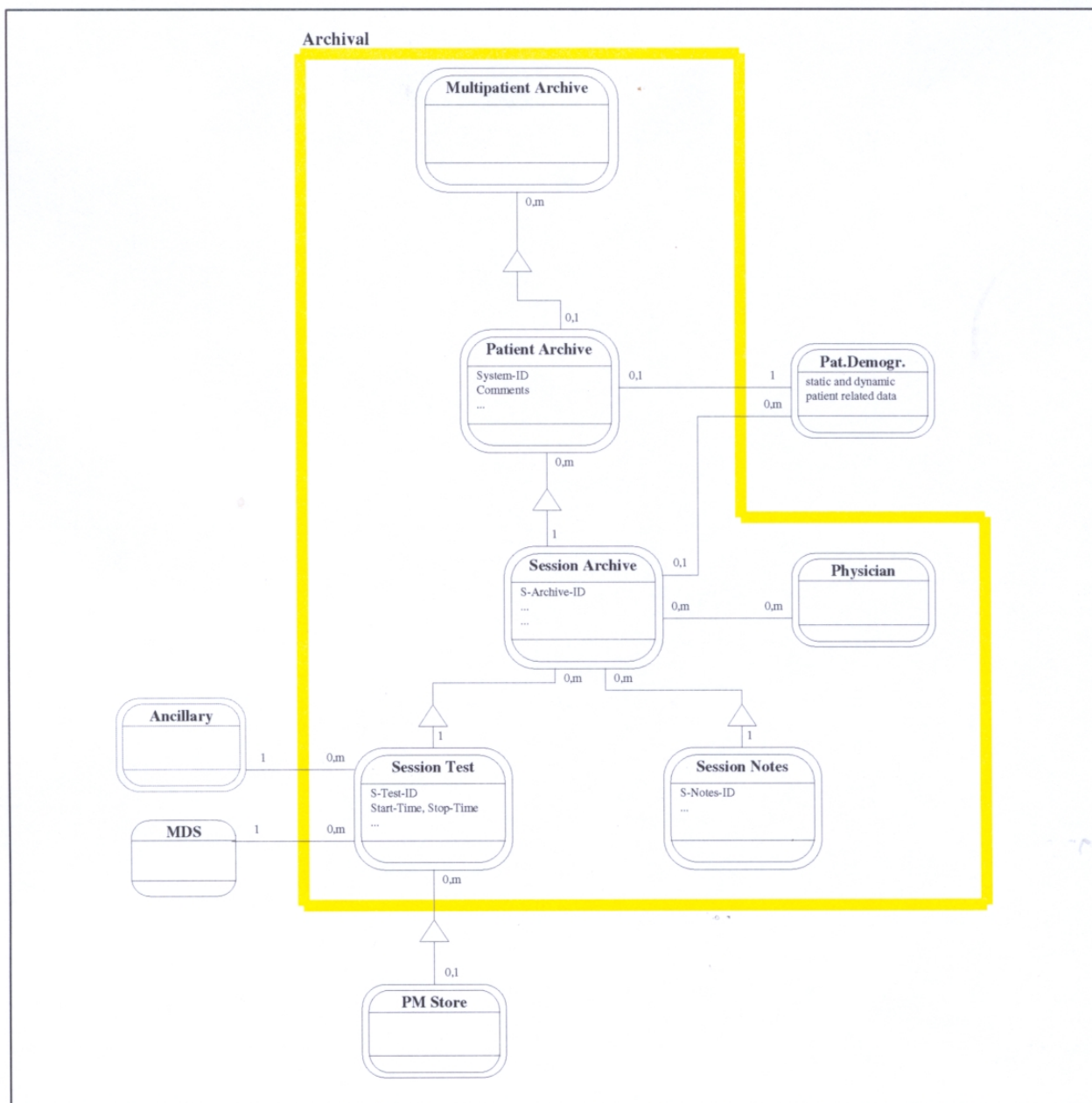


Figure 7.10: Archival Subject Model

The model contains the following objects:

7.9.1 Multipatient Archive

The Multipatient Archive object groups together multiple Patient Archives referring to different patients.

EXAMPLE: A drug study may be documented in the form of a Multipatient Archive containing multiple Patient Archives that show how the drug affected the monitored vital signs.

7.9.2 Patient Archive

The Patient Archive object groups patient related information (e.g. vital signs data, treatment data and patient demographics) together in a single archive object. The object relates to static (invariant) data in a Patient Demographics object only.

EXAMPLE: A hospital may store data about multiple visits of a single patient in a Patient Archive that contains a number of Session Archives, each documenting vital signs information recorded during a specific visit in a hospital department.

7.9.3 Session Archive

The Session Archive object represents a patient visit or a continuous stay in the a hospital or hospital department. Diagnostic treatments performed during this time period are represented by Session Test objects contained in the Session Archive. The Session Archive object refers to dynamic (variant) data in a Patient Demographics object.

7.9.4 Physician

The Physician object represents the physician responsible for the set of diagnostic and therapeutic activities during the time period represented by the Session Archive object.

7.9.5 Session Test

The Session Test object contains vital signs information of a single patient that is recorded during a single examination or diagnostic treatment. The object contains vital signs metrics in form of PM-Store objects. It also may contain information about equipment that was used for recording (in form of relations to MDS and Ancillary objects).

EXAMPLE: Vital signs information recorded during a ECG stress test examination is organized in a Session Test object.

7.9.6 Session Notes

The Session Notes object is a container for diagnostic data, patient care details and treatment related information in form of textual data.

7.9.7 Ancillary

The Ancillary object is not further defined in this standard. The object is present in the model to indicate, that information from sources other than devices within the scope of this standard are permitted to be included in (or referenced by) the Session Test object.

EXAMPLE: Image data that complies with the MEDICOM (ENV12052:1995) or DICOM (Digital Imaging and Communication in Medicine) standard are permitted to be included in the Session Test as ancillary data.

7.10 Model for the Patient Subject

The Patient subject deals with all patient related information which is relevant in the scope of this standard, but which is not vital signs information modelled in the Medical subject.

Figure 7.11 shows the object model for the Patient subject:

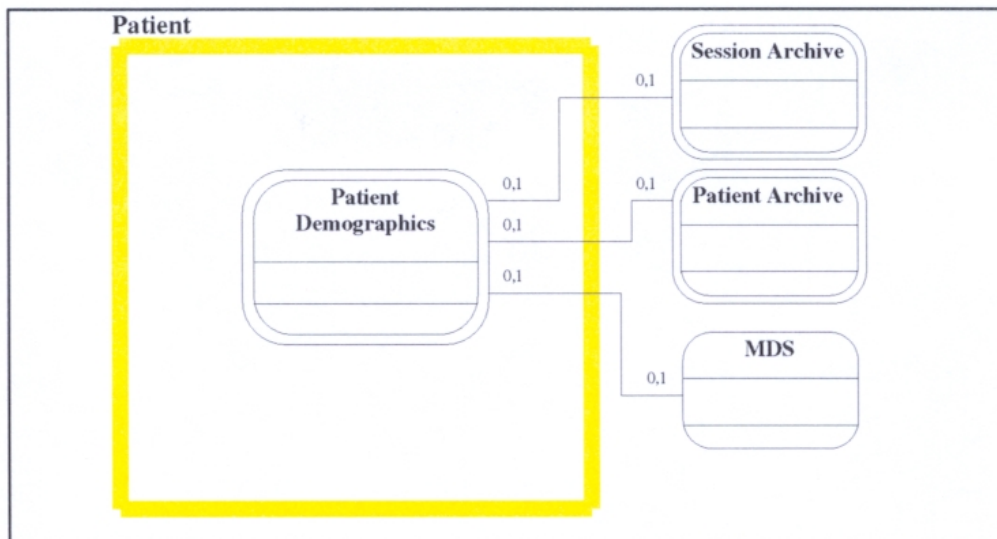


Figure 7.11: Patient Subject Model

The model contains the following objects:

7.10.1 Patient Demographics

The Patient Demographics object stores patient census data.

This standard provides minimal patient information as typically required by medical devices. A complete patient record is outside the scope of this standard.

7.11 Domain Information Model - Dynamic Model

7.11.1 General

This subclause defines global dynamic system behaviour. Note that dynamic object behaviour resulting from invocation of object management services is part of the object definitions.

7.11.2 MDS Communication Finite State Machine

Figure 7.12 shows the MDS finite state machine (FSM) for a communicating medical device that complies with the definitions in this standard. The FSM is used to synchronize the operational behaviour of manager (client) systems and agent (server) systems.

After power up, the device performs all necessary local initializations (boot phase) and ends up in the disconnected state, where it waits for connection events.

When a connection event is detected, the device tries to establish a logical connection (an association) with the other device. A manager (client) system is the association requester, an agent (server) system is the association responder. Basic compatibility checks are performed in the associating state.

After successful association, configuration data (the MDIB structure) is exchanged by the use of services and extended services (in particular the Context Scanner) as defined in this standard. Additional information (e.g. MDS attributes) is supplied that allows further compatibility and state checks.

After configuration, medical data is exchanged by using services and extended services as defined in this standard. Dynamic re-configuration is allowed in the operating state. If the device or the type of re-configuration does not allow dynamic handling in the operating state, a special state is provided for this.

If an event indicates an intention to disconnect, the disassociating state is entered.

The diagram does not show error events. Fatal error events take the state machine out of the operating state.

NOTE: This state machine describes the behaviour of the MDS communication system only. Usually the device must perform its medical function independent of the communication system.

The FSM is considered a part of the Medical Device System (MDS) object. The MDS status attribute reflects the state of the machine. The MDS may announce state changes in the form of attribute change event reports.

An interoperability standard using the definitions in this standard defines normative state transition tables based on this MDS FSM model.

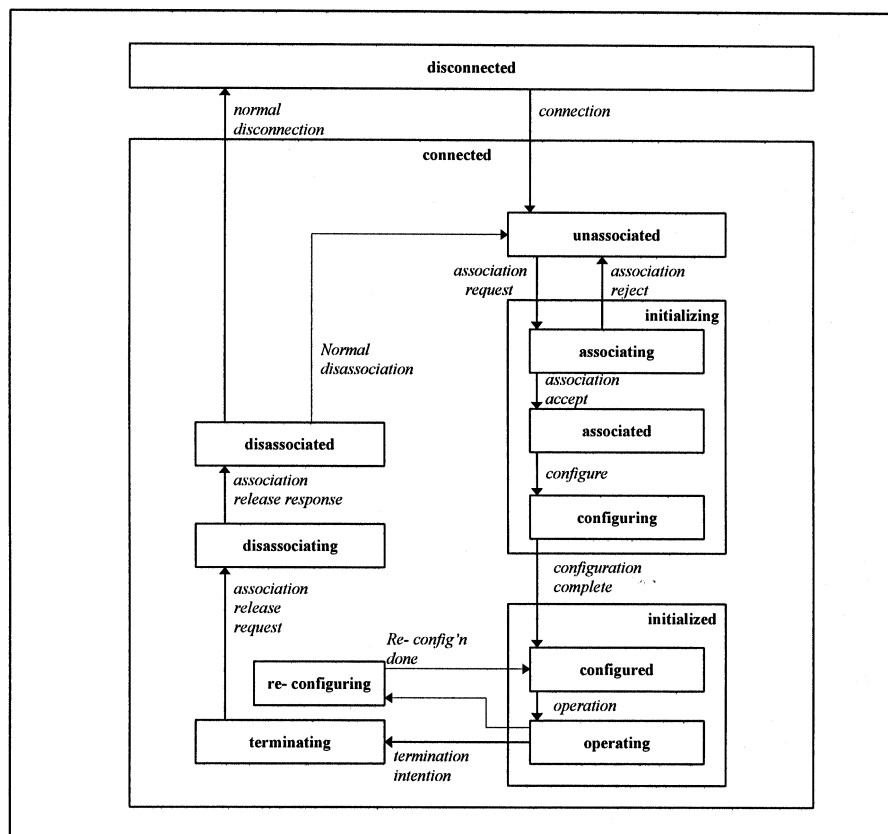


Figure 7.12: MDS Finite State Machine

7.11.3 Dynamic Object Relations

This sub-clause deals with relations between managed medical objects (objects which are defined as managed objects in this standard).

Generally, the relationships between object instances that are defined in the subject models are dynamic.

EXAMPLE: A modular patient monitor is modelled as an MDS. Measurement modules are modelled as VMDs. If a new module is connected to the monitor, there is also a new relationship between the MDS and the new VMD instance.

Communicating agent systems (agents) use services defined in this standard to announce configuration change events to other connected systems. These manager systems (managers) modify their view of the agent MDIB.

A vital signs information archive however does not only have to update its configuration, it has to permanently store these connection and disconnection events.

EXAMPLE: An instance of the Session Archive object represents the stay of a patient in the ICU. During that period, new devices are connected to the patient to increase the number of recorded vital signs. They are removed again as soon as the patient's condition stabilizes. The Session Archive shall not delete recorded data when the recording device is disconnected.

Thus, in certain applications (e.g. archival applications) object instance relationships have associated information that must be captured.

Whenever required, the relationships themselves can be considered to be special managed objects as shown in Figure 7.13.

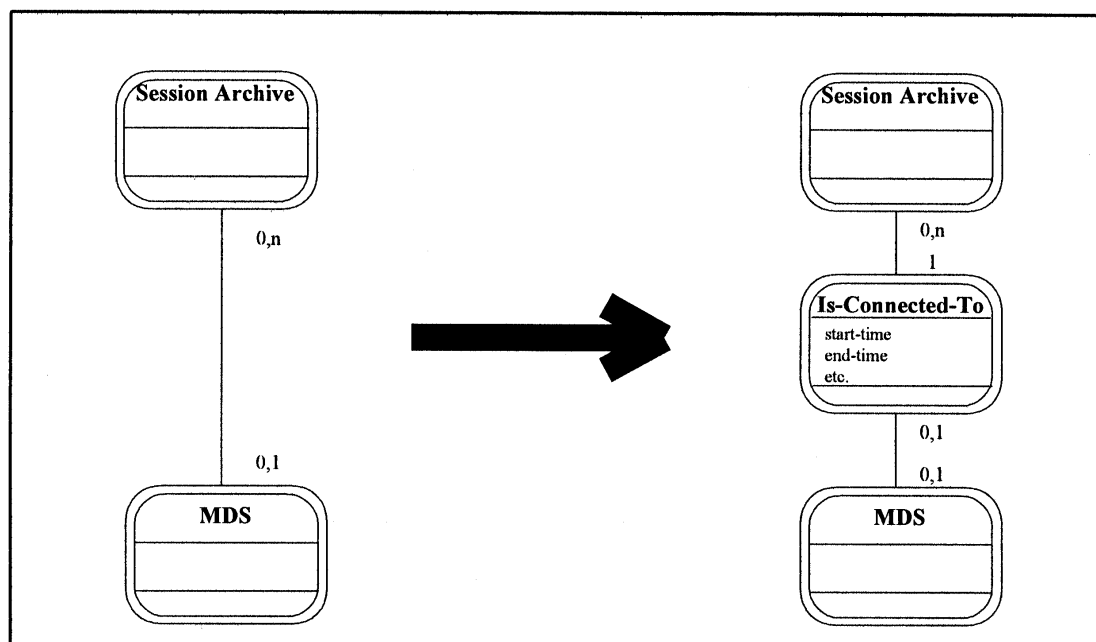


Figure 7.13: Example of a Relationship Represented by an Object

The example in Figure 7.13 shows a relation between a Session Archive and an MDS. The relation is represented as an object. This object has attributes that provide information e.g. about time of connection and disconnection

Modelling the relations as objects has the advantage that information can be defined in the form of attributes, it is not necessary to assign the attributes to one or both objects that are related.

The following subclauses define how dynamic object relations are handled by systems that comply with the definitions in this standard.

7.11.3.1 Dynamic Object Relations in Communicating Systems

Relations between object instances that are defined in the subject models are considered *configuration* information. The Context Scanner object provides configuration information in the form of object create notifications and object delete notifications. No means for persistent storage of past (old) configuration information is defined for communicating systems.

Other relations between object instances (e.g. to reference a *source* signal in derived data) are specified in the form of attributes of the corresponding objects (e.g. the *Vmo-Source-List* attribute of the Metric object). Dynamic changes of these attributes are announced by attribute change notification events.

7.11.3.2 Dynamic Object Relations in Archival Systems

An archival system may need to provide persistent storage for configuration information. In this case the corresponding relations are considered to be represented by objects, as previously shown.

An archival system that uses a data format in compliance with the definitions in this standard has to provide means to store (archive) the attributes of dynamic relationship objects.

8 Domain Information Model - Object Definitions

8.1 General

This section contains the object definitions for all objects in the Domain Information Model. The subjects defined in the model are used to categorize objects. For each object class, attributes, behaviour and notifications are defined.

8.1.1 Notation

Each object is defined in a separate clause. Subclauses define attributes, behaviour and notifications for the objects.

The object is defined in a section as follows:

Object:	<i>Defines the name of the object</i>
Description:	<i>"Gives a short (informative) textual description of the object"</i>
Derived From:	<i>Defines (potential) base classes of the object</i>
Name Binding:	<i>Defines the attribute that uniquely identifies an instance of the object in a given context (for manageable objects this is the <u>handle</u> attribute and the context is the device system, single MDS context).</i>
Registered as:	<i>Defines a term that is defined in the nomenclature to allow unique identification (object ID, code) of the object.</i>

Object attributes are defined in a subclause. A table defines attribute names, (unique) attribute IDs, attribute data types and certain qualifiers. The qualifiers have the following meaning:

M:	attribute is mandatory
O:	attribute is optional
C:	attribute is conditional, availability of attribute depends on a pre-defined condition

Unless otherwise noted, the attribute definition tables do not show inherited attributes again. This means that the attribute lists of all base classes have to be checked for a complete list of object attributes.

Attributes are assigned to (or grouped together in) *attribute groups*. This makes it possible to classify attributes according to their use (e.g. static context information, dynamic context information, value observations). The grouping also makes it possible to effectively deal with *optional* attributes: A GET service facilitates the retrieval of all members of the group, so an application is able to determine, which of the attributes are actually present in a particular object instance.

Attribute groups may be *extensible*. This means that a derived object class is able to add additional members to an inherited attribute group.

Attribute groups are also defined in a table that specifies group identification and the list of group members. Inherited attribute groups are not shown in this table again unless these groups are extensible.

Special methods or functions that are provided by an object class are defined in the *behaviour* subclause. These methods can be invoked by the CMDISE ACTION service.

Events generated by an object class (other than a generic *Attribute Change Notification*) are defined in the *notifications* subclause. An object reports these events by using the CMDISE EVENT REPORT service.

8.1.2 Common Data Types

This subclause defines a set of ASN.1 data types that are used in the object definitions.

8.1.2.1 Integer and Bit String Data Types

For representing integer numbers, the object definitions use fixed size data types only. The bit string data type represents a bit field where each single bit has a defined meaning (flag fields). The following integer types and bit string types are used:

```
--
-- 8 bit unsigned integer
--
INT-U8 ::= INTEGER (0..255)
--
-- 8 bit signed integer
--
INT-I8 ::= INTEGER (-128..127)
--
-- 16 bit unsigned integer
--
INT-U16 ::= INTEGER (0..65535)
--
-- 16 bit signed integer
--
INT-I16 ::= INTEGER (-32768..32767)
--
-- 32 bit unsigned integer
--
INT-U32 ::= INTEGER (0..4294967295)
--
-- 32 bit signed integer
--
INT-I32 ::= INTEGER (-2147483648..2147483647)
--
-- 16 bit wide bit string
--
BITS-16 ::= BIT STRING (SIZE(16))
--
-- 32 bit wide bit string
--
BITS-32 ::= BIT STRING (SIZE(32))
```

NOTE: When interpreting integer numbers, the representation (e.g. little endian vs. big endian) has to be considered. Communicating systems negotiate this representation at association (*transfer syntax*). Archival data formats have to provide a mechanism to uniquely identify integer representation (e.g. a field in a specification header).

[Ed. note: In the object definitions, integer and bit string data types with named constants or named bits also use the above notation for simplicity. This is illegal ASN.1 syntax, but it is possible to be easily transform it to the correct syntax.]

8.1.2.2 Identification

All elements (classes, objects, measurement types etc.) that need unique identification are assigned an object identifier. The set of valid object identifiers for this standard is defined in Normative Annex A of this document. The nomenclature is split into a set of partitions, each partition has its own range of 16bit wide codes. This means the 16 bit wide code is context sensitive.

The 16 bit wide data type for identification is defined as follows:

```
--
-- Object Identifier Type as defined in nomenclature
-- (do not confuse with ASN.1 OBJECT IDENTIFIER)
--
OID-Type ::= INT-U16          -- 16 bit integer type
```