ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56843— 2015/ ISO/IEEE 11073-10201: 2004

ИНФОРМАТИЗАЦИЯ ЗДОРОВЬЯ

Информационное взаимодействие с персональными медицинскими приборами

Часть 10201

Информационная модель предметной области

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

Издание официальное

Москва
Стандартинформ
2016

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным бюджетным учреждением «Центральный научно-исследовательский институт организации и информатизации здравоохранения Министерства здравоохранения Российской Федерации» (ЦНИИОИЗ Минздрава) и обществом с ограниченной ответственностью «Корпоративные электронные системы» на основе собственного аутентичного перевода на русский язык международного документа, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 468 «Информатизация здоровья» при ЦНИИОИЗ Минздрава — постоянным представителем в ISO ТС 215

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 28 декабря № 2231-ст

4 Настоящий стандарт идентичен международному стандарту ISO/ИИЭР 11073-10201:2004 «Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами. Часть 10201. Информационная модель предметной области» (ISO/IEEE 11073-10201:2004 «Health informatics — Point-of-care medical device communication — Part 10201: Domain information model)»

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов и документов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, 2016

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 4
4 Сокращения 7
5 Общие требования 8
6 Информационная модель предметной области (DIM) 9
6.1 Общие положения 9
6.2 Диаграмма пакетов. Краткий обзор 12
6.3 Модель Медицинского пакета 12
6.4 Модель Пакета тревоги 16
6.5 Модель Пакета системы 18
6.6 Модель Пакета управления 20
6.7 Модель Пакета дополнительных сервисов 23
6.8 Модель Пакета коммуникаций 26
6.9 Модель Архивного пакета 28
6.10 Модель Пакета пациента 30
6.11 Информационная модель предметной области (DIM). Динамическая модель 31
7 Определения объектов DIM 36
7.1 Общие положения 36
7.2 Главный объект (базовый класс) 45
7.3 Объекты в Медицинском пакете 46
7.4 Объекты в Пакете тревоги 83
7.5 Объекты в Пакете система 91
7.6 Объекты в Пакете управления 112
7.7 Объекты в Расширенном пакете сервисов 128
7.8 Объекты в Пакете коммуникаций 140
7.9 Объекты в Архивном пакете 148
7.10 Объекты в Пакете пациента 154
8 Модель сервисов коммуникационных систем 157
8.1 Общие положения 157
8.2 Коммуникационные системы 157
8.3 Общая модель сервисов 158
8.4 Общее определение сервисов для работы с управляемыми медицинскими объектами . . 160
9 Номенклатура MDIB 166
10 Модель соответствия 166
10.1 Применимость 166
10.2 Спецификация соответствия 167
10.3 Декларации соответствия реализации (ICS) 167
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов и документов национальным стандартам Российской Федерации 173

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАТИЗАЦИЯ ЗДОРОВЬЯ

Информационное взаимодействие с персональными медицинскими приборами

Часть 10201

Информационная модель предметной области

Health informatics. Point-of-care medical device communication.
Part 10201. Domain information model

Дата введения — 2016—11—01

1 Область применения

В общем контексте комплекса стандартов ИСО/ИИЭР 11073 задачей настоящего стандарта является определение и структуризация информации, которая непосредственно используется или хотя бы упоминается в процессе коммуникации между сущностями, относящимися к данной предметной области (предметными сущностями, прикладными сущностями).

Настоящий стандарт обеспечивает общее, независящее от синтаксиса, представление всех сущностей, относящихся к данной предметной области и участвующих в процессах, протекающих внутри различных приборов и относящихся к данной предметной области.

Определение порядка соединения приборов между собой, как и определение порядка низкоуровневой коммуникации между ними, лежат вне области действия настоящего стандарта.

2 Нормативные ссылки

Для применения настоящего стандарта необходимы следующие ссылочные документы. Для датированных ссылок применяют только указанное издание ссылочного документа, для недатированных ссылок применяют последнее издание ссылочного документа (включая все его изменения).

ЕКС ЕС 1064, Информатика в медицине. Стандартный протокол коммуникаций. Электрокардиография при помощи компьютера (CEN EN 1064, Medical informatics — Standard communication protocol — computer-assisted electrocardiography.)

ЕКС ЕПС 12052, Информатика в медицине. Обмен медицинскими изображениями (CEN ENV 12052, Medical informatics — Medical imaging communication (MEDICOM).)

ИИЭР Std 1073, Стандарт ИИЭР для коммуникаций медицинских приборов. Обзор и основа (IEEE Std 1073, IEEE Standard for Medical Device Communications — Overview and Framework.)

IETF RFC 1155, Структура и идентификация управляющей информации для управляющей информации для сетей интернет, основанных на TCP/IP (IETF RFC 1155, Structure and Identification of Management Information for TCP/IP-Based Internets.)

Издание официальное

ИСО 639-1, Коды для представления имен и языков. Часть 1: Код Альфа-2 (ISO 639-1, Code for the representation of names of languages — Part 1: Alpha-2 code.)

ИСО 639-2, Коды для представления имен и языков. Часть 2: Код Альфа-3 (ISO 639-2, Codes for the representation of names of languages — Part 2: Alpha-3 code.)

ИСО 3166-1, Коды для представления названий стран и подчиненных территорий. Часть 1: Коды стран (ISO 3166-1, Codes for the representation of names of countries and their subdivisions — Part 1: Country codes.)

ИСО 3166-2, Коды для представления названий стран и подчиненных территорий. Часть 2: Коды подчиненных территорий (ISO 3166-2, Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code.)

ИСО 3166-3, Коды для представления названий стран и подчиненных территорий. Часть 3: Код для старых названий стран (ISO 3166-3, Codes for the representation of names of countries and their subdivisions — Part 3: Code for formerly used names of countries.)

ИСО 8601, Элементы данных и форматы обмена данными. Обмен информацией. Представление даты и времени (ISO 8601, Data elements and interchange formats — Information interchange — Representation of dates and times.)

ИСО 15225, Номенклатура — Спецификация для системы номенклатуры медицинских приборов, используемых для целей обмена нормативными данными (ISO 15225, Nomenclature — Specification for a nomenclature system for medical devices for the purpose of regulatory data exchange.)

ИСО/МЭК 646, Информационные технологии. ИСО 7-битный набор кодированных символов для обмена данными (ISO/IEC 646, Information technology — ISO 7-bit coded character set for information interchange.)

ИСО/МЭК 2022, Информационные технологии. Структура кода символов и методы расширения (ISO/IEC 2022, Information technology — Character code structure and extension techniques.)

ИСО/МЭК 5218, Информационные технологии. Коды для представления человеческих полов (ISO/IEC 5218, Information technology — Codes for the representation of human sexes.)

ИСО/МЭК 7498-1, Информационные технологии. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель (ISO/IEC 7498-1, Information technology — Open systems interconnection — Basic reference model — Part 1: The basic model.)

ИСО/МЭК 7498-2, Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура защиты информации (ISO/IEC 7498-2, Information processing systems — Open systems interconnection — Basic reference model — Part 2: Security architecture.)

ИСО/МЭК 7498-3, Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 3. Именование и адрессация (ISO/IEC 7498-3, Information processing systems — Open systems interconnection — Basic reference model — Part 3: Naming and addressing.)

ИСО/МЭК 7498-4, Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 4. Основы административного управления (ISO/IEC 7498-4, Information processing systems — Open systems interconnection — Basic reference model — Part 4: Management framework.)

ИСО/МЭК 8649, Системы обработки информации. Взаимосвязь открытых систем. Определение сервиса для сервисного элемента управления ассоциацией (ISO/IEC 8649, Information processing systems — Open systems interconnection — Service definition for the Association Control Service Element.)

ИСО/МЭК 8650-1, Информационные технологии. Взаимосвязь открытых систем. Спецификация протокола для сервисного элемента управления ассоциацией. Часть 1. Протокол (ISO/IEC 8650-1, Information technology — Open systems interconnection — Connection-Oriented Protocol for the Association Control Service Element — Part 1: Protocol.)

ИСО/МЭК 8650-2, Информационные технологии. Взаимосвязь открытых систем. Спецификация протокола для сервисного элемента управления ассоциацией. Часть 2. Проформа свидетельства о конформности протокольной реализации (PICS) (ISO/IEC 8650-2, Information technology — Open systems interconnection — Protocol Specification for Association Control Service Element — Part 2: Protocol Implementation Conformance Statements (PICS) proforma.)

ИСО/МЭК 8824-1, Информационная технология. Абстрактная синтаксическая нотация 1 (ASN.1). Часть 1. Спецификация основной нотации (ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation.)

ИСО/МЭК 8824-2, Информационная технология. Абстрактная синтаксическая нотация 1 (ASN.1). Часть 2. Спецификация информационного объекта (ISO/IEC 8824-2, Information technology — Abstract Syntax Notation One (ASN.1) — Part 2: Information object specification.)

ИСО/МЭК 8859-n, Обработка информации. 8-битные однобайтовые наборы закодированных графических символов. Части 1—15. Различные алфавиты (ISO/IEC 8859-n, Information processing — 8-bit single-byte coded graphic character sets — Part 1 to Part 15: Various alphabets.)

ИСО/МЭК 9545, Информационные технологии. Взаимосвязь открытых систем. Структура прикладного уровня (ISO/IEC 9545, Information technology — Open systems interconnection — Application layer structure.)

ИСО/МЭК 9595, Информационные технологии. Взаимосвязь открытых систем. Определение сервиса общей управляющей информации (ISO/IEC 9595, Information technology — Open systems interconnection — Common management information service definition.)

ИСО/МЭК 9596-1, Информационные технологии. Взаимосвязь открытых систем. Протокол общей управляющей информации. Часть 1. Спецификация (ISO/IEC 9596-1, Information technology — Open systems interconnection — Common Management Information Protocol — Part 1: Specification.)

ИСО/МЭК 10040, Информационные технологии. Взаимосвязь открытых систем. Общее описание административного управления систем (ISO/IEC 10040, Information technology — Open systems interconnection — Systems management overview.)

ИСО/МЭК 10164-1, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 1. Функция административного управления объектами (ISO/IEC 10164-1, Information technology — Open systems interconnection — Systems management — Part 1: Object management function.)

ИСО/МЭК 10164-2, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 2. Функция административного управления состоянием (ISO/IEC 10164-2, Information technology — Open systems interconnection — Systems management — Part 2: State management function.)

ИСО/МЭК 10164-3, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 3. Атрибуты для представления связей. (ISO/IEC 10164-3, Information technology — Open systems interconnection — System management — Part 3: Attributes for representing relationships.)

ИСО/МЭК 10164-4, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 4. Функция аварийного уведомления (ISO/IEC 10164-4, Information technology — Open systems interconnection — Systems management — Part 4: Alarm reporting function.)

ИСО/МЭК 10164-5, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 5. Функция административного управления событиями (ISO/IEC 10164-5, Information technology — Open systems interconnection — Systems management — Part 5: Event management function.)

ИСО/МЭК 10164-6, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 6. Функция управления журналом (ISO/IEC 10164-6, Information technology — Open systems interconnection — Systems management — Part 6: Log control function.)

ИСО/МЭК 10164-7, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 7. Функция аварийного уведомления о защите (ISO/IEC 10164-7, Information technology — Open systems interconnection — Systems management — Part 7: Security alarm reporting function.)

ИСО/МЭК 10164-8, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 8. Функция журнала контроля системы защиты (ISO/IEC 10164-8, Information technology — Open systems interconnection — Systems management — Part 8: Security audit trail function.)

ИСО/МЭК 10164-9, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 9. Объекты и атрибуты для управления доступом (ISO/IEC 10164-9, Information technology — Open systems interconnection — Systems management — Part 9: Objects and attributes for access control.)

ИСО/МЭК 10164-10, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 10. Функция измерения частоты использования ресурса в целях учета (ISO/IEC 10164-10, Information technology — Open systems interconnection — Systems management — Part 10: Usage metering function for accounting purposes.)

ИСО/МЭК 10164-11, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 11. Метрические объекты и атрибуты (ISO/IEC 10164-11, Information technology — Open systems interconnection — Systems management — Part 11: Metric objects and attributes.)

ИСО/МЭК 10164-12, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 12. Функция управления тестированием (ISO/IEC 10164-12, Information technology — Open systems interconnection — Systems management — Part 12: Test management function.)

ИСО/МЭК 10164-13, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 13. Функция управления тестированием. Функция подведения итога (ISO/IEC 10164-13, Information technology — Open systems interconnection — Systems management — Part 13: Summarization function.)

ИСО/МЭК 10164-14, Информационные технологии. Взаимосвязь открытых систем. Административное управление системами. Часть 14. Категории полного и диагностического тестирования (ISO/IEC 10164-14, Information technology — Open systems interconnection — Systems management — Part 14: Confidence and diagnostic test categories.)

ИСО/МЭК 10165-1, Информационные технологии. Взаимосвязь открытых систем. Структура управляющей информации. Часть 1. Модель информации административного управления (ISO/IEC 10165-1, Information technology — Open systems interconnection — Structure of management information — Part 1: Management information model.)

ИСО/МЭК 10165-2, Информационные технологии. Взаимосвязь открытых систем. Структура управляющей информации. Часть 2. Определение информации административного управления (ISO/IEC 10165-2, Information technology — Open systems interconnection — Structure of management information — Part 2: Definition of management information.)

ИСО/МЭК 10646-1, Информационные технологии. Универсальный многооктетный набор кодированных символов (UCS). Часть 1. Архитектура многоязычная матрица (ISO/IEC 10646-1, Information technology — Universal multiple-octet coded character set (UCS) — Part 1: Architecture and basic multilingual plane.)

ИСО/ИИЭР 11073-10101, Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами. Часть 10101. Номенклатура (ISO/IEEE 11073-10101, Health informatics — Point-of-care medical device communication — Part 10101: Nomenclature.)

ИСО/ИИЭР 11073-20101, Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами. Часть 20101. Прикладные профили. Базовый стандарт (ISO/IEEE 11073-20101, Health informatics — Point-of-care medical device communication — Part 20101: Application profiles — Base standard.)

NEMA PS 3, Формирование цифровых изображений и обмен ими в медицине [NEMA PS 3, Digital imaging and communications in medicine (DICOM).]

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями. В случае отсутствия определения каких-либо терминов в данном разделе, обратиться к Официальному словарю терминов стандартов ИИЭР, Седьмая редакция.

3.1 **агент** (agent): Прибор, который поставляет данные в систему, реализующую взаимодействие по модели менеджер — агент.

3.2 **аварийный сигнал** (alarm): Сигнал о возникновении ненормальных изменений (аварийной ситуации) в состоянии пациента или прибора.

3.3 **сигнал тревоги, тревога** (alert): Общий синоним:

- аварийных ситуаций, связанных со здоровьем пациента;
- аварийных ситуаций, связанных с функционированием техники;
- ситуаций, требующих вмешательства пользователя в процесс функционирования оборудования;
- любой комбинации вышеуказанных ситуаций.

3.4 **отслеживание сигналов тревоги, аварийный монитор** (alert monitor): Объект, отражающий результаты работы процессора обработки аварий (аварийного процессора) прибора или системы, то есть, по существу, полное описание аварийной ситуации.

3.5 **уровень тревоги** (alert status): Объект, отражающий выходные данные обработки аварийных сигналов процессом, содержащим описания всех аварийных ситуаций в совокупности для одного или более объектов.

3.6 **архивный** (archival): Термин, относящийся к данным, предназначенным для длительного хранения.

3.7 **сервисный элемент управления связью** (association control service element, ACSE): Метод, применяемый для установления логической связи между системами медицинских приборов (системами MDS).

3.8 **канал** (channel): Вышестоящий объект в объектной модели, группирующий вместе данные — результаты физиологических измерений и данные, полученные из этих результатов.

3.9 **класс** (class): Типовое абстрактное описание одного или нескольких объектов — представителей класса, в форме единого набора атрибутов этого объекта (объектов) и предоставляемых им (ими) сервисов, включая описание процедуры создания нового объекта — представителя класса.

3.10 **коммуникационный контроллер** (communication controller): Часть системы медицинских приборов (MDS), отвечающая за обеспечение коммуникации.

3.11 **участник (сторона) процесса коммуникации** (communication party): Некто (или нечто), играющий определенную роль в предметной области и принимающий участие в процессах коммуникации в рассматриваемой предметной области.

3.12 **роль в процессе коммуникации** (communication role): Роль участника коммуникации в данной конкретной коммуникации, определяющая поведение этого участника в процессе коммуникации. С ролью связан набор сервисов, которые данный участник может предоставить другим участникам данной коммуникации.

3.13 **агент данных** (data agent): Система сбора данных о пациенте, представляющая собой медицинский прибор и предоставляющая необходимые данные другим приборам (устройствам).

3.14 **формат данных** (data format): Упорядоченное расположение данных в файле или потоке данных.

3.15 **регистратор данных** (data logger): Медицинский прибор, функционирующий в качестве оперативного хранилища данных и архива.

3.16 **структура данных** (data structure): Способ, которым прикладные сущности создают информацию о наборе данных в результате использования этими сущностями некоторого информационного объекта.

3.17 **словарь** (dictionary): Описание элементов, содержащихся в Базе данных медицинской информации (MDIB): информации о жизненно важных показателях, информации о приборе, демографические данные и т. д.

3.18 **дискретный параметр** (discrete parameter): Результат измерения жизненно важного показателя, который может быть представлен отдельным символьным или цифровым значением.

3.19 **информационная модель предметной области** (domain information model, DIM): Модель, описывающая общие понятия и связи между ними для данной предметной области.

3.20 **событие** (event): Изменение состояния прибора, о котором сообщила служба отчетов и уведомлений.

3.21 **отчет о событии** (event report): Сервис, предоставляемый элементом службы общего доступа к данным медицинских приборов для генерации отчета о событии, связанном с конкретным управляемым объектом.

3.22 **общий подход** (framework): Структурированная система процессов и спецификаций, разработанная для поддержки успешного решения некоторой определенной задачи в предметной области.

3.23 **параметр, представляемый графиком** (graphic parameter): Для того чтобы корректно представить результат измерения данного жизненно важного показателя, необходимо сформировать последовательность отдельных значений по результатам регулярных многократных измерений.

3.24 **ведущая система** (host system): Условное обозначение медицинской системы, к которой подключены измерительные приборы.

3.25 **информационный объект** (information object): Абстрактная модель данных, применяемая для передачи значений жизненно важных показателей и прочих данных о пациенте. Атрибуты в определении информационного объекта описывают его свойства. Каждое конкретное определение информационного объекта не несет конкретную информацию о реальном мире, но представляет собой описание класса объектов с одинаковыми свойствами.

3.26 **элемент службы обработки данных** (information service element): Экземпляры в базе данных медицинской информации (Medical data information base, MDIB).

3.27 **экземпляр** (instance): Конкретная реализация абстрактного понятия или спецификации, как например экземпляр объекта, экземпляр приложения, экземпляр элемента информационной службы, экземпляр виртуального медицинского прибора, экземпляр класса, операционный экземпляр.

3.28 **отделение интенсивной терапии** (intensive care unit, ICU): Отделение в структуре медицинского учреждения, в котором пациентов подвергают интенсивной терапии.

3.29 **формат обмена данными** (interchange format): Формат представления элементов данных и структуры сообщений, содержащих эти элементы данных в процессе обмена данными между системами. Формат обмена данными содержит данные о конструктивных элементах сообщений и синтаксисе. Конкретное представление формата обмена данными зависит от конкретной технологии обмена.

3.30 **интероперабельность** (interoperability): Идеальный случай совместной работы медицинских приборов различных типов, моделей и производителей и в том случае, если они подключены непосредственно друг к другу, и в том случае, если они взаимодействуют через коммуникационную систему.

3.31 **латентность, задержка, время ожидания** (latency): Время задержки между отправкой сигнала одним прибором и получением этого сигнала другим прибором в процессе коммуникации.

3.32 **нижние уровни** (lower layers): Уровни с 1-го по 4-й семиуровневой модели взаимодействия открытых систем (ВОИ) Международной организации по стандартизации (ИСО). Эти уровни охватывают спецификации механического, электрического и базового коммуникационных протоколов.

3.33 **менеджер** (manager): Прибор, который получает данные в системе, реализующей взаимодействие по модели менеджер — агент.

3.34 **модель взаимодействия менеджер — агент** (manager — agent model): Модель коммуникации, где устройство-агент предоставляет данные устройству-менеджеру, получающему эти данные.

3.35 **база данных медицинской информации** (medical data information base, MDIB): Это понятие относится к объектно-ориентированной базе данных, позволяющей, как минимум, хранить информацию о жизненно важных показателях организма.

3.36 **медицинский прибор** (medical device): Прибор (аппаратура или система приборов), используемый для мониторинга состояния, выполнения процедур или терапии пациента, и обычно не оказывающий влияние на процессы метаболизма. В данном стандарте под понятием медицинского прибора подразумеваются только приборы, подключенные к пациенту и предоставляющие возможность электронной коммуникации.

3.37 **система медицинского прибора** (medical device system, MDS): Абстракция системы, способной выполнять одну или несколько медицинских функций. В контексте настоящего стандарта этот термин используется, в частности, как объектно-ориентированная абстракция прибора, предоставляющего медицинскую информацию в форме информационных объектов, определяемых настоящим стандартом.

3.38 **монитор** (monitor): Медицинский прибор, разработанный для получения, отображения, фиксации и/или анализа данных, поступающих от пациента, и для оповещения лиц, осуществляющих наблюдение за пациентом, о событиях, требующих их внимания.

3.39 **объект** (object): Понятие, абстракция или предмет, с отчетливыми границами и значимостью для решения рассматриваемой проблемы.

3.40 **атрибуты/свойства объекта** (object attributes): Данные, которые вместе с методами определяют объект.

3.41 **класс объекта** (object class): Общее определение группы объектов с одинаковыми атрибутами, общим поведением, общими взаимоотношениями с другими объектами и общей семантикой.

3.42 **диаграмма объектов** (object diagram): Диаграмма, раскрывающая связи между объектами в системе.

3.43 **метод объекта** (object method): Процедура или процесс, влияющий на значения атрибутов объекта и состояния класса объекта.

3.44 **объектно-ориентированный анализ** (object-oriented analysis): Метод анализа предметной области или задачи, основанный на рассмотрении реального мира как совокупности неким образом взаимодействующих объектов и соответствующем моделировании предметной области или задачи.

3.45 **открытая система** (open system): Набор протоколов взаимодействия, позволяющих компьютерам с различной архитектурой связываться друг с другом.

3.46 **операция** (operation): Функция или преобразование, которая может быть применена объектом или к объекту в классе (иногда называется сервисом).

3.47 **проблемная область** (problem domain): Область здравоохранения, подвергаемая моделированию.

3.48 **протокол** (protocol): Стандартный набор правил, определяющих порядок передачи данных между приборами. Определяет формат передачи данных, а также сигналы для начала передачи, завершения передачи и контроля передачи данных.

3.49 **сканер** (scanner): Обозреватель и сумматор значений атрибутов объекта.

3.50 **сценарий** (scenario). Формальное описание некоторого класса предметной деятельности, включающее описание семантики предметных установлений и соглашений, а также и предметной информации.

3.51 **служба, сервис** (service): Определенное поведение одной из сторон коммуникации, выполняющей определенную роль в процессе коммуникации, которое указанная сторона обязана демонстрировать.

3.52 **синтаксис** (syntax) (например, формата обмена данными): Правила сочетания конструктивных элементов сообщения, предусмотренных форматом обмена данными.

3.53 **система** (system): Ограниченная часть постижимого мира, существующая во времени и пространстве, которая может рассматриваться как множество элементов и связей между ними.

3.54 **отметка времени** (timestamp): Атрибут или поле в наборе данных, который содержит указание на время создания этого набора данных.

3.55 **главный объект** (top object): Безусловный базовый класс для всех других объектов в той же модели.

3.56 **верхние уровни** (upper layers): Уровни с 5 по 7 семиуровневой модели взаимодействия открытых систем (ВОС) Международной организации по стандартизации (ИСО). Обеспечивают функционирование и специфицирование сессий обмена данными, представления данных в процессе обмена и взаимодействия приложений, обменивающихся данными.

3.57 **виртуальный медицинский прибор** (virtual medical device, VMD): Абстрактное представление подсистемы медицинского назначения в системе медицинских приборов (MDS).

3.58 **виртуальный медицинский объект** (virtual medical object, VMO): Абстрактное представление объекта в Медицинском пакете, являющемся частью информационной модели предметной области (DIM).

3.59 **показатель жизненно важных функций** (vital sign): Клиническая информация об одном или более пациентах, которая измеряется с помощью подключенной к пациенту аппаратуры или может быть получена от пациента иным способом.

3.60 **осциллограмма** (waveform): Представление для врача данных о здоровье пациента в форме графика, обычно — показателей жизненно важных функций, отражающего изменение значения показателя во времени.

4 Сокращения

ACSE — сервисный элемент управления связью;

ASN.1 — язык ASN.1 для описания абстрактного синтаксиса,

BCC — прикроватный контроллер связи;

BER — основные правила кодирования;

BMP — основная многоязыковая плоскость;

CMDIP — общий протокол обмена информацией между медицинскими приборами;

CMDISE — служба (сервис) единого протокола доступа к данным медицинских приборов;

CMIP — протокол общей управляющей информации;

CMISE — сервисный элемент общей управляющей информации;

DCC — коммуникационный контроллер медицинского прибора;

DICOM — формирование цифровых изображений и обмен ими в медицине;

DIM — информационная модель предметной области;

ЭКГ — электрокардиограмма;

ЭЭГ — электроэнцефалограмма;

EBWW — глазное яблоко и наручные часы;

FSM — конечный автомат;

GMDN — Всемирная номенклатура медицинских изделий;

GMT — среднее время по Гринвичу;

IANA — Комитет по цифровым адресам в Интернет (уполномоченный орган Общества Internet (ISOC) и Федерального Совета сети (FNC), наблюдающий и координирующий назначение каждого уникального идентификатора протокола, используемого в Интернет);

ICS — заявление о том, что результат реализации чего-либо соответствует требованиям;

ICU — отделение интенсивной терапии;

ID — персональный (индивидуальный) идентификатор или идентификация личности;

LAN — локальная вычислительная сеть;

LSB — младший бит;

MDIB — база данных медицинской информации;

MDS — программно-аппаратная система медицинского прибора;

MEDICOM — система обмена медицинскими данными в форме изображений;

MIB или Mib — база данных управляющей информации;

MOC — класс управляемых объектов;

OID — идентификатор объекта;

OR — операционная комната (операционная);

OSI — модель взаимодействия открытых систем (ВОС);

PC — персональный компьютер;

PDU — часть сообщения, содержащая служебную информацию протокола передачи данных;

PM — постоянная метрика;

SCADA — система оперативного сбора данных и диспетчерского контроля за состоянием какого-либо объекта, основанная на использовании системы датчиков;

SCP ECG — стандартный протокол обмена данными для компьютеризованной электрокардиографии;

SNMP — простой протокол управления сетью, входит в стек протоколов TCP/IP;

SNTP — простой сетевой протокол синхронизации времени;

UML — объединенный (унифицированный) язык моделирования;

UTC — поясное (местное) среднее время по Гринвичу;

VMD — виртуальный медицинский прибор;

VMO — виртуальный медицинский объект;

VMS — виртуальная медицинская система.

5 Общие требования

Семейство стандартов ИСО/ИИЭР 11073 предназначено для того, чтобы обеспечить для медицинских приборов связываться и взаимодействовать друг с другом и с компьютеризированными информационными системами в здравоохранении наиболее подходящим для данной клинической среды образом.

Идеология семейства стандартов ИСО/ИИЭР 11073 основана на объектно-ориентированной парадигме управления системами. Данные (например, результат измерения, состояние) смоделированы в форме информационных объектов, к которым можно получить доступ и которыми можно управлять посредством использования служебного протокола обеспечения доступа к объекту.

Информационная модель предметной области (DIM) определяет полный набор информационных объектов (классов) и их свойств, методов и функций доступа, необходимых для обеспечения коммуникации медицинских приборов друг с другом.

Требования уровня пользователя к организации коммуникации медицинских приборов определены в ИИЭР Стд 1073, который также определяет и пользовательские сценарии организации такой коммуникации, охваченные семейством стандартов ИСО/ИИЭР 11073.

В рамках семейства стандартов ИСО/ИИЭР 11073 к информационной модели предметной области предъявляются следующие основные требования.

- определить объектно-ориентированную модель, содержащую соответствующую информацию (то есть данные) и функции (например, управления прибором), встречающиеся в предметной области коммуникации медицинских приборов, включая информацию об измерениях, контекстные данные, способы управления приборами, и прочие соответствующие аспекты;
- предоставить детальную спецификацию информационных объектов (классов), определенных в объектно-ориентированной модели, включая их свойства и методы;
- определить модель обслуживания коммуникации медицинских приборов, позволяющую предоставить доступ к информационным объектам, их свойствам и их методам;
- использовать номенклатуру, определенную в стандарте ИСО/ИИЭР 11073-10101, для идентификации всех элементов данных в модели;
- быть пригодной для определения протоколов передачи данных, а также и для определения форматов хранения файлов данных;
- определять требования соответствия;
- быть гибкой и расширяемой, чтобы по мере необходимости в будущем позволять вносить изменения и дополнения в существующий набор инструментов моделирования.

6 Информационная модель предметной области (DIM)

6.1 Общие положения

6.1.1 Моделирование

Данная модель предметной области является объектно-ориентированной и включает в себя, объекты, их атрибуты и их методы, которые являются результатом применения абстрагирования к сущностям реального мира в области использования медицинских приборов, обеспечивающих получение и обмен данными о ключевых показателях жизнедеятельности человека.

Информационная модель и модель обслуживания (сервисная модель) для коммуникационных систем, определенные и используемые в настоящем стандарте, концептуально основаны на модели взаимодействия открытых систем (OSI — open systems interconnection) Международной организации по стандартизации (ИСО). Объекты, определенные в информационной модели, считаются управляемыми (в данном случае — медицинскими) объектами. В основном они непосредственно доступны для функций управления (то есть, к ним есть доступ), что обеспечивается общим протоколом доступа к данным медицинских приборов (CMDISE), как это определено в настоящем стандарте.

Для коммуникационных систем множество экземпляров объектов, доступных на любом медицинском приборе и соответствующих требованиям настоящего стандарта, формирует базу данных медицинской информации (MDIB). MDIB — структурированное множество управляемых медицинских объектов, отражающих информацию о показателях жизненно важных функций, предоставленную конкретным медицинским прибором. В настоящем стандарте определяются типы значений свойств объектов, иерархии и поведение объектов в MDIB.

Большинство объектов, определенных в настоящем стандарте, отражают обобщенные данные показателей жизненно важных функций и вспомогательную информацию. Специализация этих объектов устанавливается путем определения соответствующих свойств-атрибутов. Иерархии объектов и отношения между объектами используются, чтобы отразить конфигурацию прибора и возможности прибора.

Пример — Некий обобщенный объект определен, чтобы отразить показатели жизненно важных функций в форме осциллограммы в реальном времени. Набор свойств объекта используется, чтобы определить конкретную осциллограмму, как отражающую инвазивное артериальное кровяное давление. Положение данного объекта в общей иерархии всех объектов позволяет определить, какая именно подсистема отвечает за вывод этой осциллограммы.

На рисунке 6.1 представлена связь между управляемыми медицинскими объектами, MDIB, CMDISE, прикладными процессами и коммуникационной системой в медицинском приборе.



Рисунок 6.1 — База данных MDIB в коммуницируемых системах

В коммуникационных системах управляемые медицинские объекты доступны только с помощью функций, предоставляемых CMDISE. Способ сохранения этих объектов в MDIB в любой конкретной системе, и способ доступа к этим объектам CMDISE и приложений, являются особенностями реализации конкретной системы и в качестве таковых не регламентируются.

Формат представления архивированных данных о показателях жизненно важных функций, соответствующий требованиям настоящего стандарта, предусматривает, что экземпляр объекта хранится на одном носителе с данными об изменениях значений его свойств за некоторый установленный промежуток времени. В настоящем стандарте определяются типы значений свойств объектов и иерархии объектов для формата представления архивированных данных.

На рисунке 6.2 показана связь между управляемыми медицинскими объектами, MDIB, CMDISE, прикладными процессами и коммуникационными системами.

Для формата представления архивированных данных способ сохранения управляемых медицинских объектов на носитель является предметом стандартизации. Но сервисы доступа к архивам являются частными особенностями конкретной реализации и в качестве таковых не подлежат регламентации настоящим стандартом.

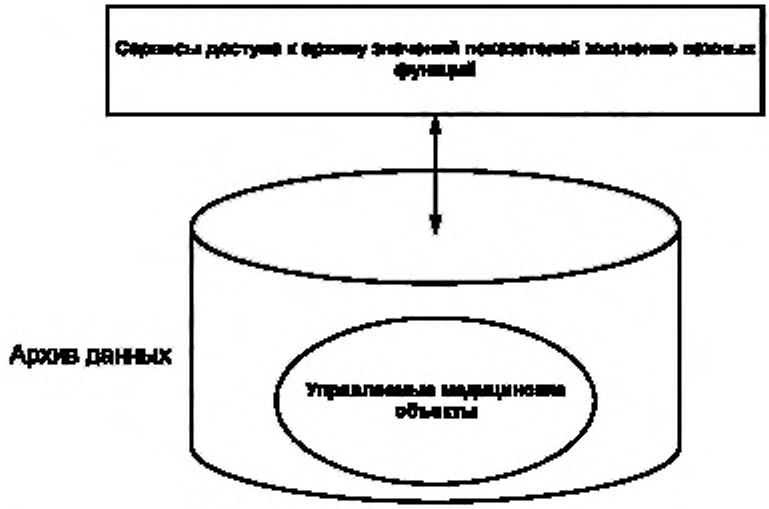


Рисунок 6.2 — Управляемые медицинские объекты в архиве значений показателей жизненно-важных функций

6.1.2 Область применения DIM

6.1.2.1 Общие положения

Информационные объекты показателей жизненно важных функций, которые определены в настоящем стандарте, охватывают оцифрованные биомедицинские сигналы, которые получены в процессе использования медицинских измерительных приборов, например, в анестезии, хирургии, терапии вливания, интенсивной терапии и акушерской заботе.

Данные биомедицинских сигналов с точки зрения настоящего стандарта включают полученные непосредственно или опосредовано количественные и качественные результаты медицинских измерений, технические и медицинские сигналы тревоги и настройки управления. Информация о пациенте, необходимая для интерпретации этих биомедицинских сигналов, также определяется в DIM.

6.1.2.2 Коммуникационные системы

Коммуникационные системы в рамках области применения настоящего стандарта включают физиологические измерители и анализаторы, в особенности это касается систем, обеспечивающих мониторинг в режиме реального времени или в течение длительного периода времени. Таким системам требуются соответствующие возможности обработки данных.

Объекты управления обменом данными, содержащие возможности и понятия для установления рентабельной коммуникации (особенно объекты, способные к резюмированию данных) и объекты, необходимые для обеспечения коммуникаций в реальном времени, также включаются в информационную модель, рассматриваемую в настоящем стандарте.

Рассмотрение конкретных вопросов обеспечения интероперабельности приборов, особенно таких, как низкоуровневое общение, временная синхронизация множества приборов и т. п., выходит за рамки настоящего стандарта.

6.1.2.3 Заархивированные показатели жизненно важных функций

Контекст информационных объектов, содержащий описания процесса получения данных, и необходимый для организации архива показателей жизненно важных функций, также относится к сфере настоящего стандарта.

6.1.3 Подход к моделированию

Для объектно-ориентированного моделирования применяется технология, основанная на использовании Унифицированного языка моделирования (UML). Для начала вся предметная область распределена по нескольким относительно небольшим пакетам моделирования. Каждый пакет определяется своими диаграммами объектов, содержащими краткие описания объектов и определяющими их иерархии и связи между ними.

Каждый объект необходимо снабдить текстовым определением. Свойства объектов определяются в таблице определения свойств. Для определения типов значений свойств используется язык ASN.1. Поведение объекта и уведомления, генерируемые объектами, также определяются в таблице определения свойств. Эти определения непосредственно касаются модели обслуживания, определенной в разделе 8 настоящего стандарта.

6.1.4 Расширение модели

Ожидается, что в долгосрочной перспективе могут потребоваться расширения данной модели для учета новых разработок в области медицинских приборов. Кроме того, в каких-то частных случаях может возникнуть необходимость в моделировании данных, специфичных для конкретного прибора или приложения и поэтому не отраженных в общей модели.

В некоторых случаях может быть применено такое понятие, как внешние связи объекта. Большинство объектов, определенных в этом стандарте, содержат группу атрибутов (например, группа атрибутов Взаимоотношения), которая может использоваться, чтобы представить информацию об объектах, связанных с данным объектом, которые не определены в информационной модели. Эта информация может быть отражена посредством определения соответствующей связи данного объекта с внешним объектом и назначения этой связи соответствующих атрибутов (см. 7.1.2.20).

В других случаях может оказаться необходимо определить совершенно новые объекты или добавить новые атрибуты, новые методы, или новые события в уже определенные объекты (классы). С помощью таких расширений модели может быть учтена информация личного характера или касающаяся производителя конкретного оборудования. Что делать с такими расширениями — это, главным образом, вопрос стандарта на взаимодействие систем и приборов, который основан на настоящем стандарте на представление показателей жизненно важных функций.

В общем случае, в формате взаимодействия объекты, атрибуты, и методы идентифицируются номенклатурными кодами. В пространстве номенклатурных кодов (например, в значениях кодов) оставляют место для частных расширений. Как правило, стандарт на взаимодействие систем и приборов, который основан на этой Информационной модели, должен быть в состоянии обрабатывать расширения, касающиеся личной информации или производителя оборудования, игнорируя объекты, атрибуты и т. д., с неизвестными идентификаторами (то есть номенклатурными кодами).

6.2 Диаграмма пакетов. Краткий обзор

Диаграмма пакетов организует классы проблемной области в несколько групп — пакетов. Диаграмма отражает основные объекты внутри каждого пакета и определяет взаимосвязи между пакетами.

Диаграмма пакетов, представленная на рисунке 6.3, содержит лишь небольшое подмножество множества объектов, определенных в информационной модели. Базовые общие объекты, за исключением главного объекта модели, не отражаются в данной диаграмме. Кроме того, отражаются не все взаимосвязи между пакетами. Более полную информацию можно найти в детализированных диаграммах пакетов.

Числа в пакетах указывают на соответствующие подразделы данного раздела, посвященного моделям, и раздела 7, посвященного определениям объектов.

Главный объект модели — абстрактный базовый класс и в то же время общий предок всех объектов (классов), определенных в информационной модели. Для удобства редактирования в моделирующих диаграммах в данном стандарте не показана вся иерархия классов.

Более детальные модели для этих пакетов содержатся в подразделах 6.3—6.10.

6.3 Модель Медицинского пакета

В Медицинском пакете мы имеем дело с происхождением и представлением биомедицинских сигналов и контекстной информацией, важной для правильной интерпретации результатов измерений.

На рисунке 6.4 представлена информационная модель (диаграмма классов) Medical Package.

Примечание — Представители классов Channel и PM-Store могут быть включены только в единственного представителя агрегирующего класса. Информацию о классах, связанных с сигналом тревоги см. в Пакете тревоги.

Модель Медицинского пакета содержит классы, описанные в 6.3.1—6.3.13.

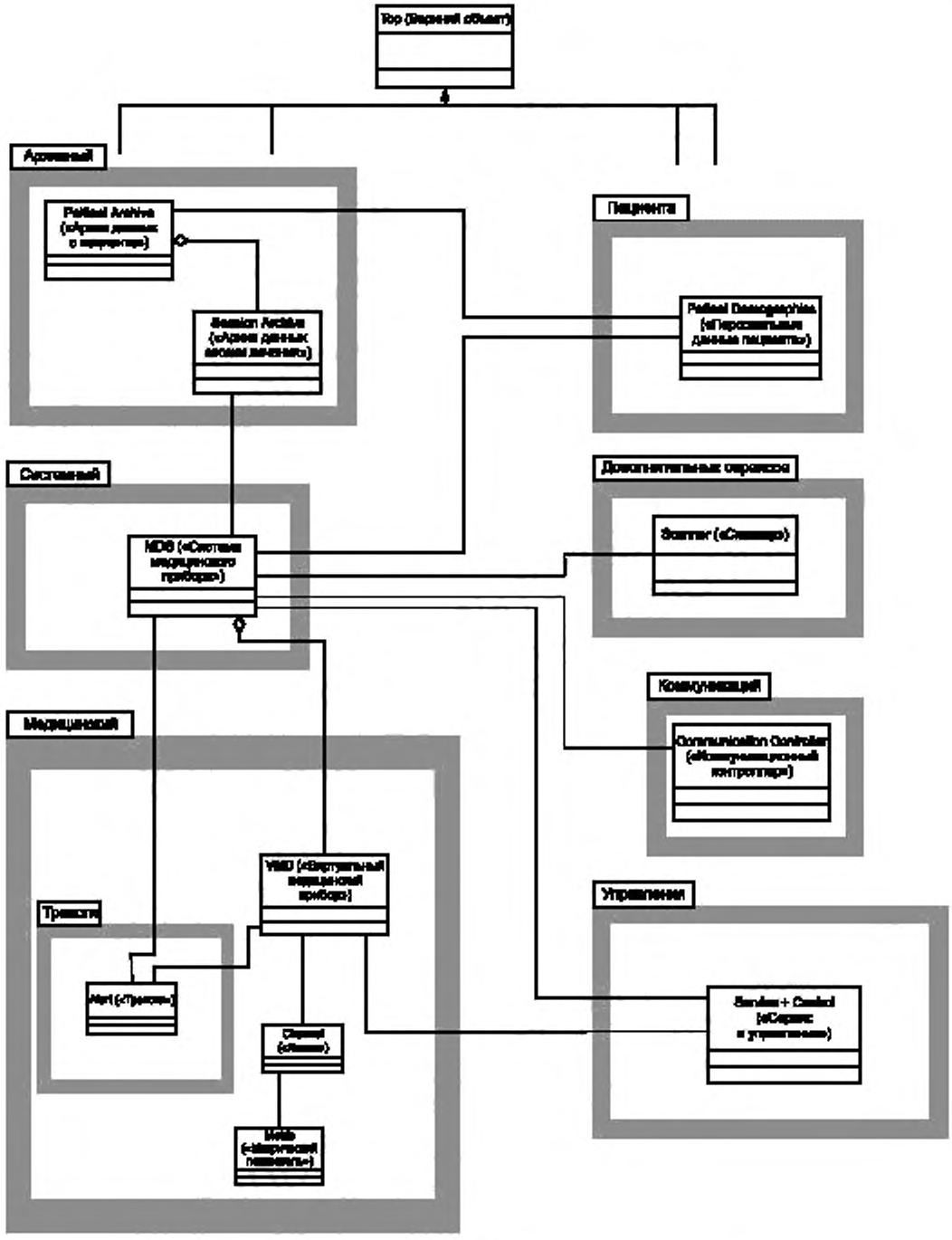


Рисунок 6.3 — Пакеты информационной модели предметной области

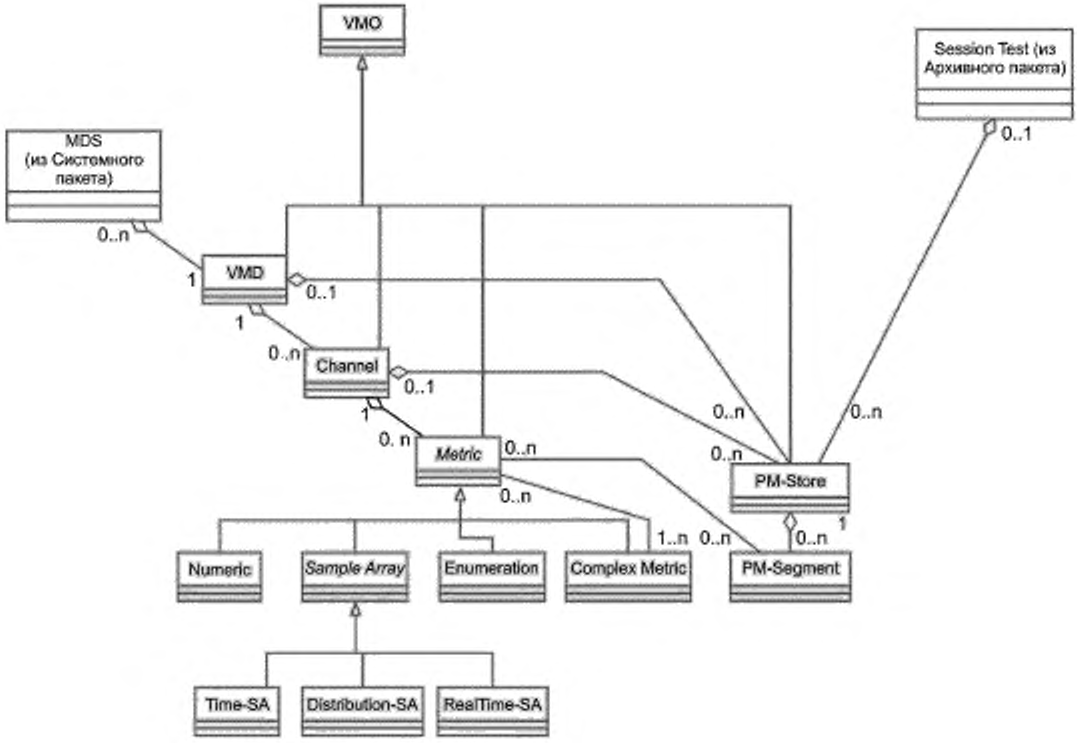


Рисунок 6.4 — Модель Медицинского пакета

Примечание — Экземпляры объекта Channel и объекта PM-Store должны содержаться только в одном экземпляре высшего объекта. См. Пакет тревоги для информации об объектах, связанных с сигналами тревоги.

6.3.1 Класс Virtual Medical Object (VMO — Виртуальный медицинский объект)

Класс VMO — базовый класс для всех классов медицинского назначения в модели. Это обстоятельство позволяет установить непротиворечивую систему наименования и обозначения классов в модели Медицинского пакета.

Будучи базовым абстрактным классом, класс VMO не может иметь конкретных представителей — экземпляров.

6.3.2 Класс Virtual Medical Device (VMD — Виртуальный медицинский прибор)

Класс VMD является абстракцией подсистемы медицинского назначения (например, программно-аппаратной или даже чисто программной) в составе медицинского прибора. В данном классе собраны характеристики такой подсистемы (например, режимы работы или версии). В то же время, объект класса VMD является контейнером для объектов классов, отражающих данных о результатах медицинских измерений и состоянии самого прибора.

Пример — Модульный монитор состояния пациента предоставляет возможности выполнения различных измерений с помощью подключаемых модулей. Каждый такой модуль описывается объектом — представителем класса VMD.

6.3.3 Класс Channel (Канал)

Объект класса Channel используется для группировки объектов класса Metric и, следовательно, предоставляет возможность иерархической организации информации. Но объект класса Channel не обязателен для представления объектов класса Metric в составе объекта класса VMD.

Пример — Для объекта класса VMD Измеритель кровяного давления можно определить объект класса Channel для группирования вместе всех метрик, связанных с кровяным давлением (значение показателя давления, график изменения давления и т. д.). Второй объект класса Channel может быть определен для группирования вместе метрик, связанных с пульсом.

6.3.4 Класс Metric (Метрический показатель)

Класс Metric — базовый класс для всех классов, отражающих непосредственные или опосредованные качественные или количественные результаты измерения биомедицинских сигналов, информацию о состоянии устройства и необходимую контекстную информацию.

Предусмотрена специализация подклассов (производных классов) класса Metric для работы с общепринятыми способами представления результатов измерений (например, отдельные значения, массив данных, индикация текущего состояния) и с отображением (например, на дисплее) результатов измерений.

Будучи абстрактным базовым классом, класс Metric не может иметь конкретных представителей — экземпляров.

6.3.5 Класс Numeric (Числовой показатель)

Класс Numeric отражает те значения биомедицинских измерений и ту информацию о состоянии прибора, которые представлены в числовом формате, как, например, измерения амплитуды колебаний или показания счетчиков.

Пример — Результат измерения пульса отражается в объекте класса Numeric.

Примечание — Объект класса Numeric может быть составным для эффективного моделирования, например, кровяного давления, с которым связаны значения сразу трех показателей: (систолическое давление, диастолическое давление, среднее по результатам измерений). Наличие сразу нескольких значений у одного объекта класса Numeric (или любого представителя подклассов класса Metric) может быть отражено в специальном структурированном атрибуте, унаследованном от класса Metric.

6.3.6 Класс Sample Array (Массив выборок)

Класс Sample Array — базовый класс для представления таких метрических показателей, значения которых представлены графиком, некоторой кривой, и, поэтому, результаты отслеживания их значений, поступающие от коммуницированных систем, представлены массивом данных об отдельных точках графика.

Будучи абстрактным базовым классом, класс Sample Array не может иметь конкретных представителей — объектов.

6.3.7 Класс Real Time Sample Array (Массив выборок в режиме реального времени)

Объект класса Real Time Sample Array — такая выборка значений, которая отражает график, изменяющийся в режиме реального времени. К такой информации в коммуницированных системах предъявляются специальные требования, как, например, вычислительная мощность, малое время ожидания, большая ширина полосы пропускания. Поэтому, потребовалось определение специализированного класса.

Пример — График электрокардиограммы (ЭКГ), формируемый в режиме реального времени представляется объектом класса Real Time Sample Array.

6.3.8 Класс Time Sample Array (Временной массив выборок)

Объект класса Time Sample Array — такая выборка значений, которая отражает последовательность отдельных фрагментов графика, ограниченных некоторым интервалом времени. В пределах одного наблюдения (т. е. одного фрагмента графика), точки графика фиксируются через равные промежутки времени.

Пример — Программное обеспечение для анализа сегмента ST может использовать объект класса Time Sample Array, чтобы представить фрагменты графика кардиограммы, фиксируемой в режиме реального времени, которые содержат только единственный комплекс QRS. В пределах этого фрагмента графика программное обеспечение может само расположить точки измерения ST. Например, можно получать новый фрагмент графика каждые 15 секунд.

6.3.9 Класс Distribution Sample Array (Распределенный массив выборок)

Объект класса Distribution Sample Array — выборка значений, представляющая линейные распределения значений в форме массивов, содержащих масштабированную выборку. Индекс значения в пределах множества, определяемого одним наблюдением, показывает положение значения в пространстве, но не во времени.

Пример — Приложение для снятия электроэнцефалограммы (ЭЭГ) может использовать преобразование Фурье, чтобы получить распределение частоты (то есть, спектр) сигнала ЭЭГ. В этом случае для представления спектра в базе данных MDIB используется как раз объект класса Distribution Sample Array.

6.3.10 Класс Enumeration (Перечисление)

Класс Enumeration отражает информацию о текущем состоянии и/или хронологическую информацию. Результаты отслеживания могут быть представлены в форме нормативных кодов (которые включены в номенклатуру, определенную в настоящем стандарте или в какой-либо другой номенклатуре), или в форме произвольного текста.

Пример — Квалификация сердечного ритма в электрокардиограмме может быть представлена как объект класса Enumeration. Вентилятор может предоставить информацию о текущем режиме вентиляции также объектом класса Enumeration.

6.3.11 Класс Complex Metric (Комплексный метрический показатель)

Объект класса Complex Metric может использоваться в некоторых случаях, чтобы сгруппировать большое количество тесно связанных объектов подклассов класса Metric в одном контейнере, для повышения производительности или удобства моделирования. Объект класса Complex Metric представляет собой некоторую композицию объектов подклассов класса Metric, возможно, рекурсивную.

Пример — Вентилятор может обеспечить широкие возможности анализа дыхания пациента. Для каждого вздоха он может вычислять различные числовые значения (например, объем вздоха, соотношение I:E, длительность вздоха) так же хорошо, как и перечисляемые значения (например, классификация типа дыхания или хронологические данные). Для повышения эффективности обработки вся эта информация группируется в одном экземпляре объекта Complex Metric, который обновляется после каждого вздоха.

6.3.12 Класс Persistent Metric Store, PM-Store (Хранилище постоянной метрики)

Объект класса PM-Store предоставляет возможность долговременного хранения метрических данных. Он может содержать различное количество объектов класса PM-Segment, к которым можно получить доступ только через данный объект класса PM-Store. По умолчанию объект класса PM-Store предназначен, чтобы хранить данные единственного объекта подклассов класса Metric.

Пример — Прибор сохраняет числовое значение инвазивного кровяного давления на диске. В этом случае для представления неизменяемой информации используется объект класса PM-Store. Атрибуты объекта класса PM-Store описывают период осуществления выборки, алгоритм осуществления выборки, и формат хранящихся данных. Когда этикетка текущего измерения давления меняется (например, во время wedge procedure), в процессе хранения создается новый объект класса PM-Segment для хранения уже обновленных контекстных данных (в данном случае — этикетки).

6.3.13 Класс Persistent Metric Segment, PM-Segment (Сегмент постоянной метрики)

Объект класса PM-Segment отражает непрерывный промежуток времени, в течение которого метрические данные хранятся без каких-либо изменений значений соответствующих контекстных атрибутов (например, масштабы или этикетки).

Объект класса PM-Segment доступен только через соответствующий объект класса PM-Store (например, для того, чтобы извлечь сохраняемые данные).

6.4 Модель Пакета тревоги

В Пакете тревоги мы имеем дело с классами, отражающими информацию о текущем состоянии пациента и/или состоянии прибора, влияющем на измерение показателей или функционирование прибора. Информация, связанная с тревожными ситуациями, часто является предметом нормативного регулирования и, поэтому, требует специальной обработки.

В данной модели все объектно-ориентированные элементы, связанные с аварийными сигнали, определяются общим термином тревога. Термин тревога используется в настоящем стандарте в качестве синонима комбинации аварийных ситуаций, связанных со здоровьем пациента, аварийных ситуаций, связанных с функционированием техники, и сигналов, требующих вмешательства пользователя в процесс функционирования оборудования.

Аварийный сигнал представляет собой сигнал о наступлении ненормального события в состоянии здоровья пациента или состоянии прибора. Сигнал физиологической тревоги — сигнал либо о том, что значение отслеживаемого физиологического показателя вышло за пределы области допустимых значе-

ний, либо о том, что состояние пациента является патологически ненормальным. Технический аварийный сигнал — сигнал либо о том, что прибор не способен правильно отслеживать состояние пациента, либо о том, что отслеживание состояния пациента прекращено.

В данной модели предусмотрены три различных уровня аварийной сигнализации. Каждому уровню соответствует определенный пошаговый процесс обработки аварийной ситуации в диапазоне от обнаружения простого, не зависящего от контекста, аварийного сигнала до интеллектуальной системы обработки аварийного сигнала в приборе. Указанный процесс требует установления порядка приоритетности всех аварийных сигналов, которые может сгенерировать прибор, чтобы зафиксировать аварийный сигнал до момента привлечения внимания пользователя и осуществить аудиовизуальное представление сигнала пользователю для привлечения его внимания.

Для согласованности всех процессов обработки аварийных сигналов в системе, обычный медицинский прибор либо вообще должен быть не способен к подаче аварийных сигналов, либо должен быть способен подавать сигналы только одного уровня, который зависит от текущей функциональности прибора. Каждый уровень представлен в модели одним специальным классом объектов. Другими словами, в дереве состава прибора либо могут вообще отсутствовать представители классов объекта аварийного сигнала, либо могут присутствовать представители только одного из этих классов (Тревога (Alert), или Уровень тревоги (Alert Status), или Монитор тревожных состояний (Alert Monitor)), но в произвольном количестве. Допустимо присутствие множества экземпляров объекта (класса).

Примечание — Аварийная сигнализация медицинского прибора является предметом регламентации различными национальными и международными стандартами в области обеспечения безопасности (как, например, семейства стандартов МЭК 60601 и ИСО 9703). С учетом требований существующих стандартов в области обеспечения безопасности, соответствующие классы в настоящем стандарте определяют только содержание соответствующей информации. Любая реализация тревожной сигнализации должна, поэтому, соответствовать требованиям соответствующих стандартов и учитывать соответствующие изменения.

На рисунке 6.5 показана модель классов Пакета тревоги.

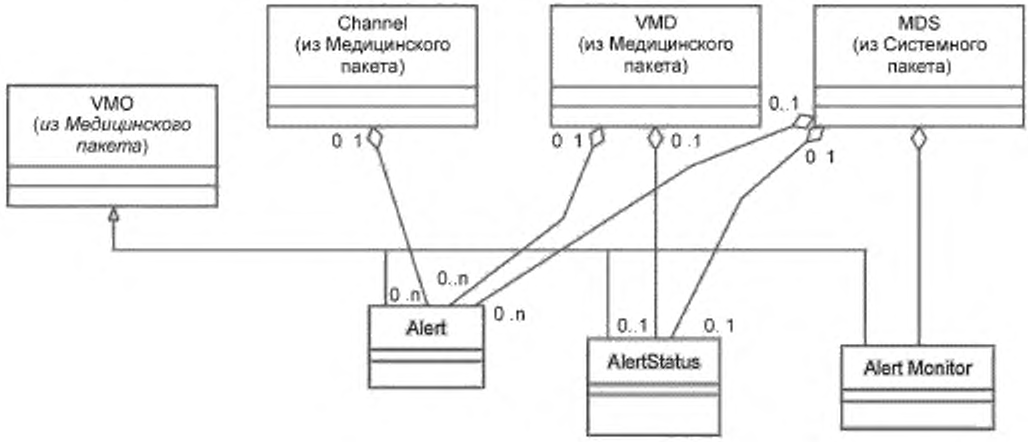


Рисунок 6.5 — Модель Пакета тревоги

Примечание — Экземпляры объекта Пакета тревоги должны содержаться только в одном объекте-контейнере (объекте одного из верхних (белых) классов в модели).

Модель Пакета тревоги включает в себя классы, описанные в 6.4.1—6.4.3.

6.4.1 Класс Alert (Тревога)

Класс Alert предназначен для отражения результата выявления простого сигнала тревоги. Поэтому, он отражает только одну единственную аварийную ситуацию физиологического или технического характера, связанную с состоянием соответствующего объекта (медицинский прибор (MDS), виртуаль-

ный медицинский прибор (VMD) или метрический показатель (Metric)). Если прибор порождает объект класса Alert, то он уже не может породить объект класса Alert Status или класса Alert Monitor. Для каждой тревоги, которую может выявить прибор, необходим отдельный объект класса Alert.

Объект Alert содержит ссылку на тот конкретный экземпляр объекта из Медицинского пакета, с которым связана соответствующая тревога.

Примечание — Экземпляр объекта Alert не создается и не уничтожается динамически в случаях, когда соответствующая тревога начинается или прекращается. Скорее, в этих случаях изменяются значения атрибутов некоторого существующего экземпляра объекта Alert.

Пример — Объект класса Alert может отражать состояние процесса контроля наступления физиологической тревоги, связанной с выходом значения пульса за пределы допустимого. В случае выхода значения пульса за пределы допустимого, объект класса Alert генерирует событие (например, изменение значения соответствующего атрибута), которое отражает данную тревогу в форме изменения значений атрибутов данного объекта.

6.4.2 Класс Alert Status (Уровень тревоги)

Объект класса Alert Status отражает результат процесса обработки сигнала тревоги, который включает всю текущую совокупность аварий для одного или нескольких медицинских объектов. В отличие от объекта класса Alert, объект класса Alert Status собирает данные по всем авариям, относящимся либо к иерархии объектов класса VMD, либо к объекту класса MDS, и представляет эту информацию в виде структурированного списка атрибутов. Сбор в одном объекте данных обо всех авариях позволяет реализовать процесс начального уровня по обработке аварии, где знания о конкретных объектах классов VMD и MDS может быть использовано для назначения приоритетов авариям и подавления ложных сигналов тревоги.

Для крупномасштабных приборов, для которых затруднительно осуществить полную обработку аварии, использование объектов класса Alert Status позволяет радикально уменьшить издержки на обработку большого количества экземпляров объекта Alert.

Если прибор порождает объект Alert Status, он уже не может породить объекты Alert или Alert Monitor. Каждый объект классов VMD или MDS в базе данных MDIB способен содержать не более одного экземпляра объекта Alert Status.

Пример — Виртуальный прибор (объект класса VMD) для выполнения ЭКГ получает значения сердечного ритма (пульса) пациента. Если виртуальный прибор может определить, что контакты аппарата отсоединены от пациента, порождаемый им объект класса Alert Status сообщает только о технической тревоге и подавляет появление сигнала о выходе значения сердечного ритма за пределы допустимого.

6.4.3 Класс Alert Monitor (Монитор тревожных ситуаций)

Объект класса Alert Monitor отражает результаты работы системного процессора сигналов тревоги. В качестве такового, он отражает общее описание аварийного состояния для прибора или системы в целом и представляет эту информацию в форме совокупного списка всех аварийных ситуаций в системе. Этот список включает информацию об общем состоянии системы и информацию о конкретных аварийных ситуациях, что позволяет реализовать соответствующее существующим стандартам безопасности отображение информации о тревоге в удаленной системе (на удаленном устройстве).

Если прибор порождает объект класса Alert Monitor, то он уже не может порождать объекты классов Alert или Alert Status. MDS должен содержать не более одного экземпляра объекта Alert Monitor.

Пример — Система мониторинга состояния пациента предоставляет информацию об аварийной ситуации в форме объекта класса Alert Monitor на центральный пост. Информация об аварийной ситуации включает в себя как информацию о сигналах тревоги высшего приоритета, выводимую в аудиовизуальной форме на дисплее монитора, так и общий список всех наступивших аварийных ситуациях, технических и физиологических. Системный процессор обработки сигналов тревоги работает в закрытом режиме, где данные о физиологических тревогах накапливаются в буфере до тех пор, пока они не будут однозначно распознаны пользователем.

6.5 Модель Пакета системы

В Пакете системы имеют дело с представлением приборов и систем, которые получают или обрабатывают информацию о жизненно важных показателях здоровья и соответствуют требованиям настоящего стандарта.

На рисунке 6.6 показана модель классов Пакета системы.

Модель Пакета системы содержит классы, описанные в 6.5.1—6.5.10.

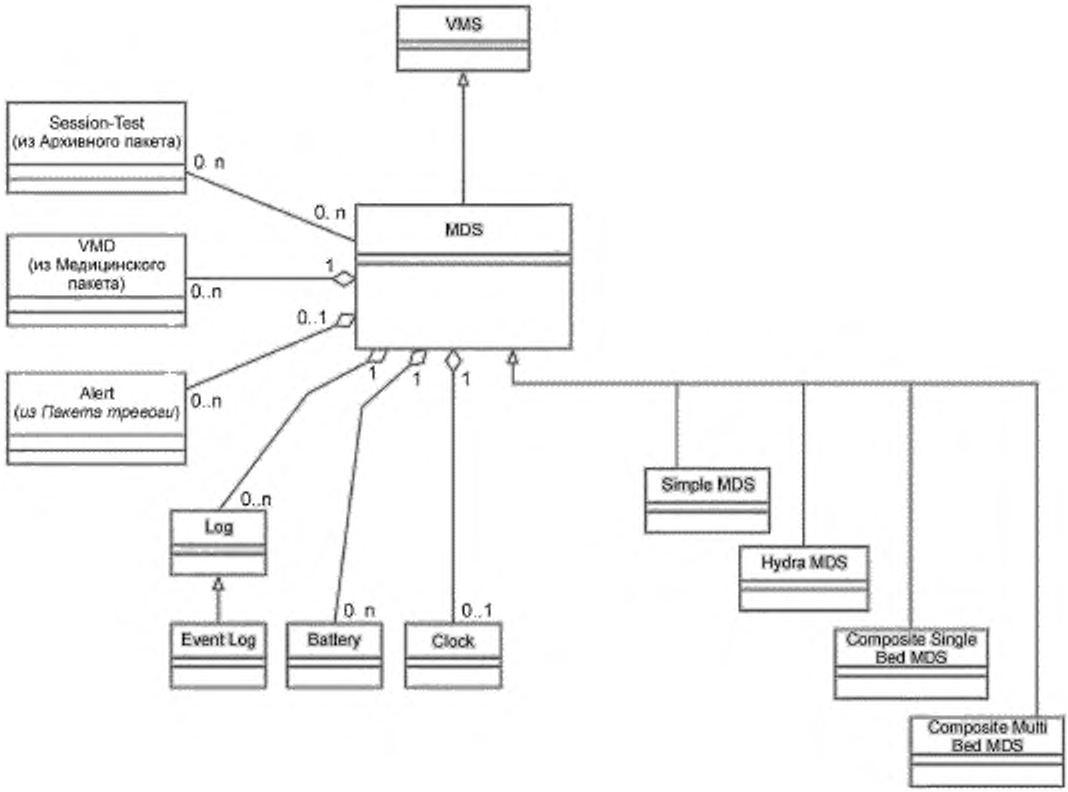


Рисунок 6.6 — Модель Пакета системы

6.5.1 Класс Virtual Medical System (Виртуальная медицинская система, VMS)

Класс VMS — абстрактный базовый класс для всех классов Пакета системы в данной модели. Это обстоятельство позволяет установить непротиворечивую систему наименования и обозначения классов в пакете System.

Будучи базовым абстрактным классом, класс VMS не может иметь конкретных экземпляров-представителей.

6.5.2 Класс Medical Device System (MDS)

Класс MDS — абстракция медицинского прибора, который предоставляет медицинские данные в форме классов, определенных в Медицинском пакете рассматриваемой информационной модели предметной области.

Объект класса MDS является объектом самого главного уровня в базе данных MDIB соответствующего прибора и сам его представляет. Составные приборы могут содержать в своей базе данных MDIB дополнительные объекты класса MDS.

Специализированные подклассы данного класса используются для отражения различий между системами по степени сложности и области применения.

Будучи базовым абстрактным классом, класс MDS не может иметь конкретных экземпляров-представителей.

6.5.3 Класс Simple MDS (Простая MDS)

Класс Simple MDS описывает медицинский прибор, который содержит единственный экземпляр объекта VMD (монофункциональный прибор).

6.5.4 Класс Hydra MDS (Модульная MDS)

Класс Hydra MDS описывает медицинский прибор, который содержит несколько разных экземпляров объекта VMD (многофункциональный прибор).

6.5.5 Класс Composite Single Bed MDS (Составная простая прикроватная MDS)

Класс Composite Single Bed MDS описывает медицинский прибор, который включает в себя (или связывается посредством некоторого интерфейса с) один или несколько объектов классов Simple MDS или Hydra MDS, сосредоточенных в одном месте (то есть, на одной или у одной кровати).

6.5.6 Класс Composite Multiple Bed MDS (Составная модульная прикроватная MDS)

Класс Composite Multiple Bed MDS описывает медицинский прибор, который содержит (или связывается посредством некоторого интерфейса с) несколько объектов классов Simple MDS или Hydra MDS, распределенных по нескольким местоположениям (то есть, на или у нескольких кроватей).

6.5.7 Класс Log (Журнал (лог))

Класс log — абстрактный базовый класс, который является контейнером для хранения важных локальных системных уведомлений и событий. Можно определить специализированные классы-потомки для различных типов событий.

Будучи базовым абстрактным классом, класс Log не может иметь конкретных экземпляров-представителей.

6.5.8 Класс Event Log (Журнал событий)

Класс Event Log — основная разновидность (класс-потомок) класса Log, предназначен для хранения системных событий в свободном текстовом формате (неформатированный текст).

Пример — При использовании прибора для инъекций может возникнуть необходимость отследить изменения способа инъекции и количества вводимого вещества на удаленном устройстве. При вызове удаленной операции, создается новая запись в журнале событий прибора (объект класса Event Log).

6.5.9 Класс Battery (Батарея)

Для приборов с батарейным питанием, некоторая часть информации о батарее включается в состав объекта класса MDS в форме соответствующих атрибутов. Для случаев, когда батарейная подсистема способна предоставлять дополнительную информацию (смарт-батарея) или некоторым способом управляться, определен специальный класс Battery.

6.5.10 Класс Clock (Часы)

Класс Clock предоставляет дополнительные возможности для обработки информации, связанной с датой и временем по сравнению с базовыми возможностями, предоставляемыми классом MDS. Он нужен для моделирования способности объектов класса MDS работать с данными в режиме реального времени.

Объект класса Clock применяется в случае, когда необходима точнейшая синхронизация работы медицинских приборов. Объект обеспечивает должное разрешение и точность информации, так что приложения могут синхронизировать потоки данных между приборами в режиме реального времени.

6.6 Модель Пакета управления

Пакет управления содержит классы, которые позволяют осуществлять дистанционное управление измерениями и прибором.

Модель для дистанционного управления, определенная в настоящем стандарте, обеспечивает следующие преимущества:

- система, которая позволяет дистанционное управление, в состоянии четко зафиксировать, к каким атрибутам или функциям обращалась удаленная система, и что именно она изменила;
- для атрибутов, допускающих дистанционное изменение, в системе управления предусмотрен список допустимых значений;
- дистанционно управляемый элемент не обязательно соответствует атрибуту медицинского объекта;
- возможно моделирование зависимости управляемого элемента от внутреннего состояния системы;

- простая схема блокировки транзакций позволяет осуществлять обработку переходных состояний в процессе дистанционного управления.

В настоящем стандарте рассматриваются, по крайней мере, два разных способа использования дистанционного управления:

- автоматическое управление может быть реализовано некоторыми процессами, выполняющимися на управляемом устройстве. Такой процесс должен быть в состоянии автоматически обнаружить, как он может получить доступ к управляемым элементам или изменить их, чтобы обеспечить их правильное функционирование;

- также возможно использовать дистанционное управление, чтобы предоставить некий интерфейс для управления человеку-оператору. Для этого варианта использования, необходимо предоставить оператору описания функций управляемого прибора и прочую возможную вспомогательную информацию.

В основе представленного подхода лежит использование классов, производных от класса Operation. Класс Operation позволяет изменять значение виртуального атрибута. Этот виртуальный атрибут может, например, быть измерительной меткой, состоянием фильтра (вкл/выкл) или коэффициентом усиления. Атрибут называется виртуальным, потому что он не обязан соответствовать никакому атрибуту в других классах, представленных объектами в системе.

Различные специализации потомков класса Operation определяют, как именно изменяется значение виртуального атрибута. Класс Select Item Operation, например, позволяет выбрать конкретное значение из данного списка возможных значений для атрибута. Класс Set Value Operation позволяет установить для атрибута значение из определенного диапазона с определенной шириной шага (то есть с определенным разрешением).

Идея состоит в том, что класс Operation предоставляет всю необходимую информацию о допустимых значениях атрибута. Кроме того, класс Operation определяет различные формы текстовой строки для поддержки человека, пользующегося соответствующей операцией. Он также содержит группирующую информацию, что позволяет логическую группировку нескольких представителей класса Operation (или классов-потомков) вместе, когда они являются частью пользовательского интерфейса.

К объектам класса Operation не могут непосредственно получить доступ сервисы, определенные в модели обслуживания в разделе 8 настоящего стандарта. Вместо этого все попытки управления должны направляться через объект класса SCO (Service-and-Control). Этот объект поддерживает простой механизм блокировки транзакций, чтобы предотвратить побочные эффекты, вызванные одновременными запросами.

Объект класса SCO группирует вместе все объекты класса Operation (и классов-потомков), которые принадлежат одной определенной сущности (то есть MDS и VMD). Объект класса SCO также позволяет обратную связь управляемому прибору, например, для визуального отображения того обстоятельства, что прибор в настоящее время находится под дистанционным управлением.

На рисунке 6.7 показана модель Пакета управления.

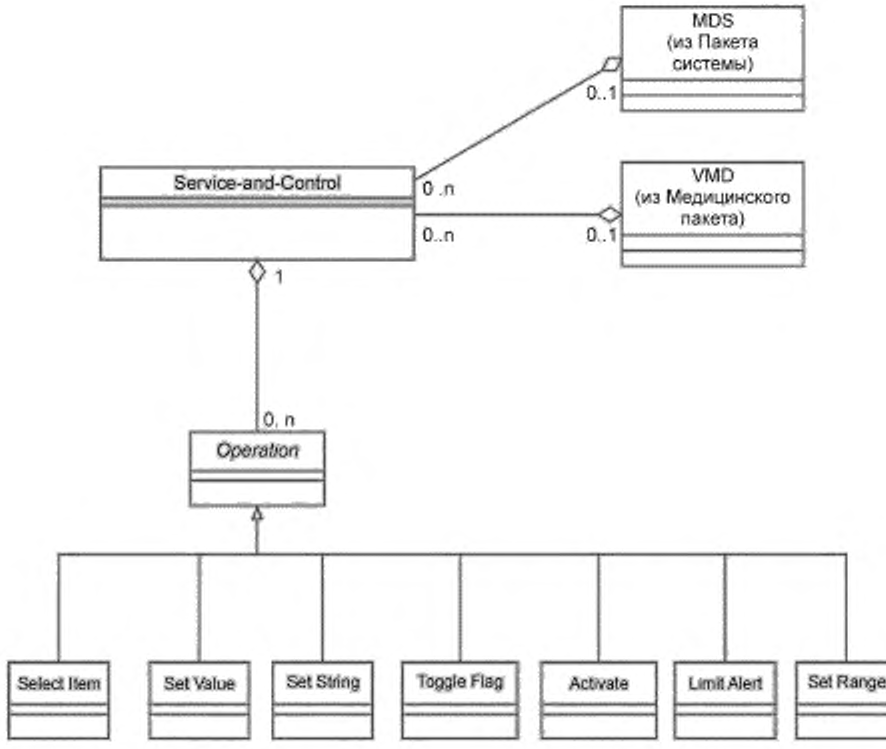


Рисунок 6.7 — Модель Пакета управления

Модель Пакета управления содержит классы, описанные в 6.6.1—6.6.9.

6.6.1 Класс Service-and-Control Object (Объект, предоставляющий сервис и управление, SCO)

Класс SCO отвечает за использование всех возможностей дистанционного управления, которые поддерживаются медицинским прибором.

Дистанционное управление в коммуницировании медицинских приборов чувствительно к проблемам обеспечения сохранности и безопасности. Класс SCO предоставляет средства для следующего:

a) простой процесс обработки транзакций, который предотвращает несогласованности, когда прибором пытаются управлять одновременно с нескольких точек (локальных и удаленных), и во время обработки управляющих команд;

b) индикация состояний, которая позволяет осуществлять локальную и удаленную индикацию текущих процессов управления.

6.6.2 Класс Operation (Операция)

Класс Operation — абстрактный базовый класс для классов, которые описывают удаленно-управляемые элементы. Каждый объект класса Operation (или класса-потомка) позволяет системе изменять некий конкретный элемент (то есть, виртуальный атрибут) конкретным способом, определенным объектом класса Operation (или класса-потомка). К объектам класса Operation не могут непосредственно получить доступ сервисы, определенные в модели обслуживания в разделе 8 настоящего стандарта. Вместо этого все попытки управления должны направляться через объект класса SCO (Service-and-Control Object) для упрощения обработки транзакций.

Набор объектов классов-потомков класса Operation, порожденных конкретным медицинским прибором, полностью определяет интерфейс дистанционного управления прибора. Пользуясь этим, цен-

тральная система в состоянии обнаружить способности прибора к дистанционному управлению в фазе конфигурирования.

6.6.3 Класс Select Item Operation (Операция выбора элемента из списка)

Класс Select Item Operation позволяет осуществить выбор одного элемента из данного списка.

Пример — VMD для измерения инвазивного давления может позволить модификацию своей этикетки. Для этого используется объект класса Select Item Operation. Список допустимых значений этикетки, предоставляемых операцией, может выглядеть, например, как {ABP, PAP, CVP, LAP}. Вызывая операцию, пользователь получает возможность выбрать одно значение из этого списка.

6.6.4 Класс Set Value Operation (Операция установления значения)

Класс Set Value Operation позволяет осуществлять подстройку значения атрибута в пределах заданного диапазона с заданным разрешением.

Пример — Объект VMD, осуществляющий измерение показателя, может позволить подстройку коэффициента усиления сигнала. Для этого используется объект класса Set Value Operation. Операция предоставляет поддерживаемый диапазон значений и ширину шага в пределах этого диапазона.

6.6.5 Класс Set String Operation (Операция задания строки)

Класс Set String Operation позволяет системе устанавливать содержание закрытой строковой переменной заданной максимальной длины и формата.

Пример — Устройство для инъекций может позволить удаленной системе определять имя влитого лекарственного средства в свободно-текстовой форме, для удобства показа на собственном (локальном) дисплее. Для этого используется экземпляр объекта Set String Operation. Операция определяет максимальную длину последовательности и формат символов так, чтобы устройство было в состоянии показать название лекарственного средства на своем дисплее.

6.6.6 Класс Toggle Flag Operation (Операция переключения)

Класс Toggle Flag Operation позволяет осуществлять операцию переключения между двумя альтернативными состояниями (например, вкл/выкл).

Пример — VMD для снятия кардиограммы может обеспечить линейную частотную фильтрацию. Для включения и выключения фильтра используется объект класса Toggle Flag Operation.

6.6.7 Класс Activate Operation (Операция активации)

Класс Activate Operation позволяет начать определенную деятельность (например, обнуление показателя давления).

Пример — Процедура обнуления показателя инвазивного давления для соответствующего VMD может начаться при наличии соответствующего объекта класса Activate Operation.

6.6.8 Класс Limit Alert (Operation Операция настройки границы тревожного состояния)

Класс Limit Alert Operation позволяет осуществлять подстройку границ рабочего диапазона датчика тревоги и переключения между включением и выключением границ установленного диапазона.

6.6.9 Класс Set Range Operation (Операция установления границ диапазона значений)

Класс Set Range Operation позволяет осуществить выбор диапазона значений параметра путем одновременной настройки верхнего и нижнего пределов диапазона значений в рамках определенных границ.

Пример — Объект VMD для измерения показателя может обеспечить прием аналогового сигнала, для которого диапазон значений может быть настроен при наличии объекта класса Set Range Operation.

6.7 Модель Пакета дополнительных сервисов

Пакет дополнительных сервисов (Extended Services) содержит классы, которые предоставляют дополнительные сервисы для управления медицинскими объектами, которые, в свою очередь, обеспечивают эффективный доступ к медицинской информации в коммуницирующих системах. Эффективность доступа достигается посредством использования ряда классов, которые помещают данные о значениях атрибутов нескольких объектов в одно сообщение о событии.

Классы, предоставляющие дополнительные сервисы, концептуально основаны на сервисах управления системой ИСО/ВОС, определенных в семействе стандартов ИСО/МЭК 10164 (Часть 5 и Часть 13). Определения были адаптированы и оптимизированы для удовлетворения специфических

потребностей именно в области обмена данными о показателях жизненно важных функций между медицинскими приборами.

На рисунке 6.8 показана модель классов Пакета дополнительных сервисов.

Модель Пакета дополнительных сервисов содержит объекты, описанные в 6.7.1—6.7.9.

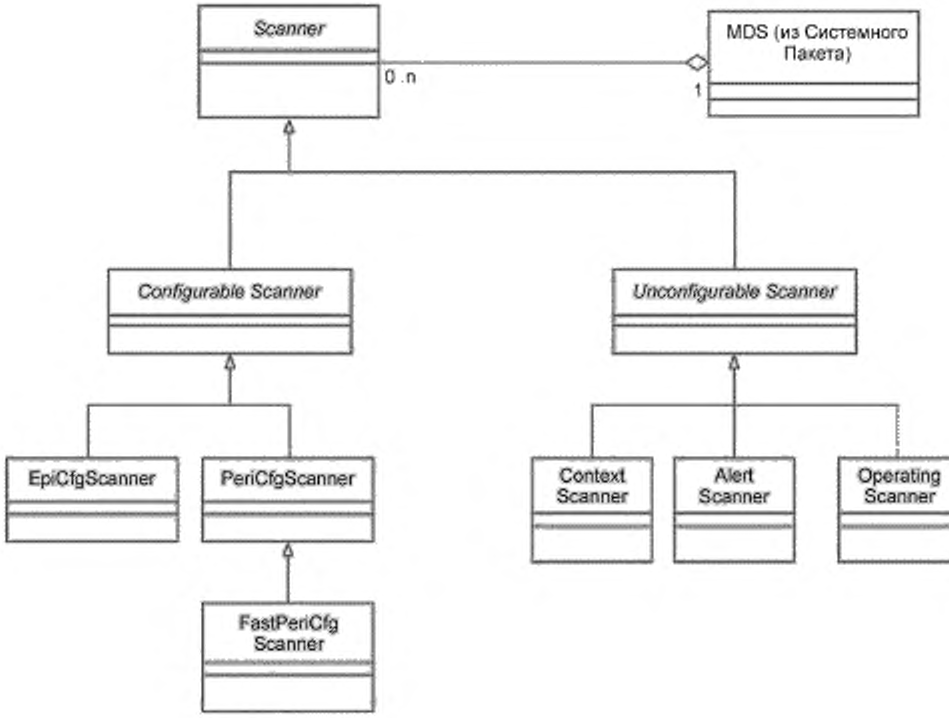


Рисунок 6.8 — Модель Пакета дополнительных сервисов.

6.7.1 Класс Scanner (Сканер)

Класс Scanner — абстрактный базовый класс, который является обозревателем и объединителем значений атрибутов объектов. Объект класса Scanner обозревает значения атрибутов управляемых медицинских объектов и генерирует сводные отчеты в форме уведомлений — отчетов о событиях. Эти отчеты о событиях содержат данные от нескольких объектов, которые обеспечивают лучшую коммуникацию, по сравнению с раздельными опрашивающими командами опроса (например, сервис GET), или многократными отдельными отчетами о событиях для всех экземпляров объекта.

Объекты классов, производных от класса Scanner, могут порождаться либо самой системой-агентом либо системой-менеджером (например, динамическое создание сканера посредством использования сервиса CREAT).

Будучи абстрактным базовым классом, класс Scanner не может иметь конкретных представителей-экземпляров.

6.7.2 Класс CfgScanner (Конфигурируемый сканер)

Класс CfgScanner — абстрактный базовый класс, у которого есть специальный атрибут ScanList, который позволяет системе устанавливать, какие атрибуты объектов следует сканировать. Значение атрибута ScanList может быть изменено или системой-агентом (автоконфигурация или преконфигурация) или системой-менеджером (полная динамически изменяемая конфигурация посредством использования сервиса SET).

Класс CfgScanner может выполнять сканирование с разной степенью детализации:

a) группа атрибутов (то есть, предопределенный набор атрибутов). Атрибут ScanList содержит идентификаторы (ID) групп атрибутов, и все атрибуты в такой группе сканируются;

b) отдельный атрибут. Атрибут ScanList содержит идентификаторы (ID) всех атрибутов, которые сканируются.

Чтобы эффективно работать с необязательными атрибутами объектов, рекомендуется использовать объекты классов, производных от класса CfgScanner в режиме работы с группами атрибутов.

Будучи абстрактным базовым классом, класс CfgScanner не может иметь конкретных представителей-экземпляров.

6.7.3 Класс EpiCfgScanner (Эпизодически конфигурируемый сканер)

Класс EpiCfgScanner предназначен для отслеживания значений атрибутов управляемых медицинских объектов и оповещения об изменениях значений атрибутов в форме небуферизуемых отчетов о событиях.

Небуферизуемый отчет о событии инициируется только изменением значения атрибута объекта. Если объект класса EpiCfgScanner использует режим работы с группами атрибутов, отчет о событии содержит все атрибуты просканированного объекта, которые принадлежат той группе атрибутов, в которой один или несколько атрибутов изменили свои значения.

Пример — Медицинский прибор формирует отчет о сердцебиении пациента в форме объекта класса Enumeration. Управляющая программа дисплея создает экземпляр объекта EpiCfgScanner и добавляет в его атрибут ScanList отслеживаемое значение объекта класса Enumeration. Экземпляр объекта Scanner впоследствии посылает уведомление каждый раз, когда соответствующий объект класса Enumeration сообщает о сердцебиении.

6.7.4 Класс PeriCfgScanner (Периодически конфигурируемый сканер)

Класс PeriCfgScanner предназначен для отслеживания значений атрибутов управляемых медицинских объектов и периодического оповещения о текущих значениях атрибутов в форме буферизуемых отчетов о событиях. Буферизуемый отчет о событии содержит значения атрибутов для всех доступных атрибутов, которые определены в списке сканирования, независимо от того, изменялись ли они.

Если сканер работает в специальном режиме суперпозиции, буферизуемый отчет о событии содержит все изменения значений атрибутов, имевшие место в отчетный период; в прочих случаях отчет содержит только последние по времени значения атрибутов.

Пример — Приложение, ведущее журнал изменения данных, порождает экземпляр объекта PeriCfgScanner и конфигурирует сканер так, чтобы он посылал обновление информации о значениях сканируемых атрибутов всех объектов класса Numeric в базу данных MDIB каждые 15 с.

6.7.5 Класс FastPeriCfgScanner (Быстрый периодически конфигурируемый сканер)

Класс FastPeriCfgScanner — специализированный класс объектов, предназначенный для того, чтобы отслеживать значения подлежащих просмотру значения атрибутов объекта класса Real Time Sample Array. Этот специальный сканер глубоко оптимизирован для оперативной отправки отчетов с минимальными задержками и эффективного использования пропускной способности канала коммуникации, что необходимо для нормального доступа к значениям данных в форме осцилограмм, получаемым в реальном масштабе времени.

Пример — Программе управления дисплеем в режиме реального времени (например, система-менеджер) нужно отразить графики кардиограммы. Она создает объект класса FastPeriCfgScanner в системе-агенте (например, на сервере) и запрашивает периодические обновления значений со всех датчиков ЭКГ.

6.7.6 Класс UcfgScanner (Неконфигурируемый сканер)

Класс UcfgScanner — абстрактный базовый класс, который предназначен для сканирования предопределенного набора управляемых медицинских объектов, которые не могут быть изменены. Другими словами, объект класса UcfgScanner — типичный генератор отчетов, предназначенный для решения одной конкретной задачи.

Будучи абстрактным базовым классом, класс UcfgScanner не может иметь конкретных представителей-экземпляров.

6.7.7 Класс Context Scanner (Контекстный сканер)

Класс Context Scanner предназначен для отслеживания изменений конфигурации устройства. Порожденный объект класса Context Scanner отвечает за оповещение о появлении новых экземпляров объектов в базе данных MDIB прибора. Сканер предоставляет данные об иерархии вложенности экземпляров объекта и значениях статических атрибутов объекта.

В случае динамически изменяемой конфигурации объект класса Context Scanner посылает уведомления о появлении новых экземпляров объекта или удалении существующих.

Пример — Приложение, ведущее журнал изменения данных, порождает новый экземпляр объекта Context Scanner в базе данных MDIB агента, чтобы получить уведомления об изменениях конфигурации MDS, когда подключаются новые модули измерения показателей (новый экземпляр объекта VMD) или когда такой модуль выключен (удаление конкретного экземпляра объекта VMD).

6.7.8 Класс Alert Scanner (Сканер тревожных состояний)

Класс Alert Scanner предназначен для отслеживания групп атрибутов, связанных с сигналами тревоги, у объектов в Пакете тревога. По соображениям обеспечения безопасности данный сканер — не конфигурируем (отслеживаются или все или никакие объекты класса Alert).

Объект класса Alert Scanner периодически посылает отчеты о событиях для проверки условий времени ожидания (тайм-аута).

6.7.9 Класс Operating Scanner (Сканер функционирования)

Класс Operating Scanner предназначен для предоставления полной информации о функционировании и системе управления медицинского прибора.

Другими словами, сканер поддерживает конфигурацию объектов классов, производных от класса Operation, содержащихся в объектах класса SCO (посредством отправки уведомлений CREAT для объектов классов, производных от Operation). Он отслеживает атрибуты объектов класса SCO, связанные с обработкой транзакций и атрибуты объектов классов, производных от Operation. Поскольку объекты класса SCO и объекты классов, производных от Operation, могут быть взаимозависимы, данный сканер — не конфигурируем.

6.8 Модель Пакета коммуникаций

В Пакете коммуникаций (Communication Package) мы имеем дело с классами, которые позволяют осуществлять и поддерживают базовую коммуникацию медицинских приборов.

На рисунке 6.9 показана модель Пакета коммуникаций.

Модель Пакета коммуникаций содержит объекты, описанные в 6.8.1—6.8.6.

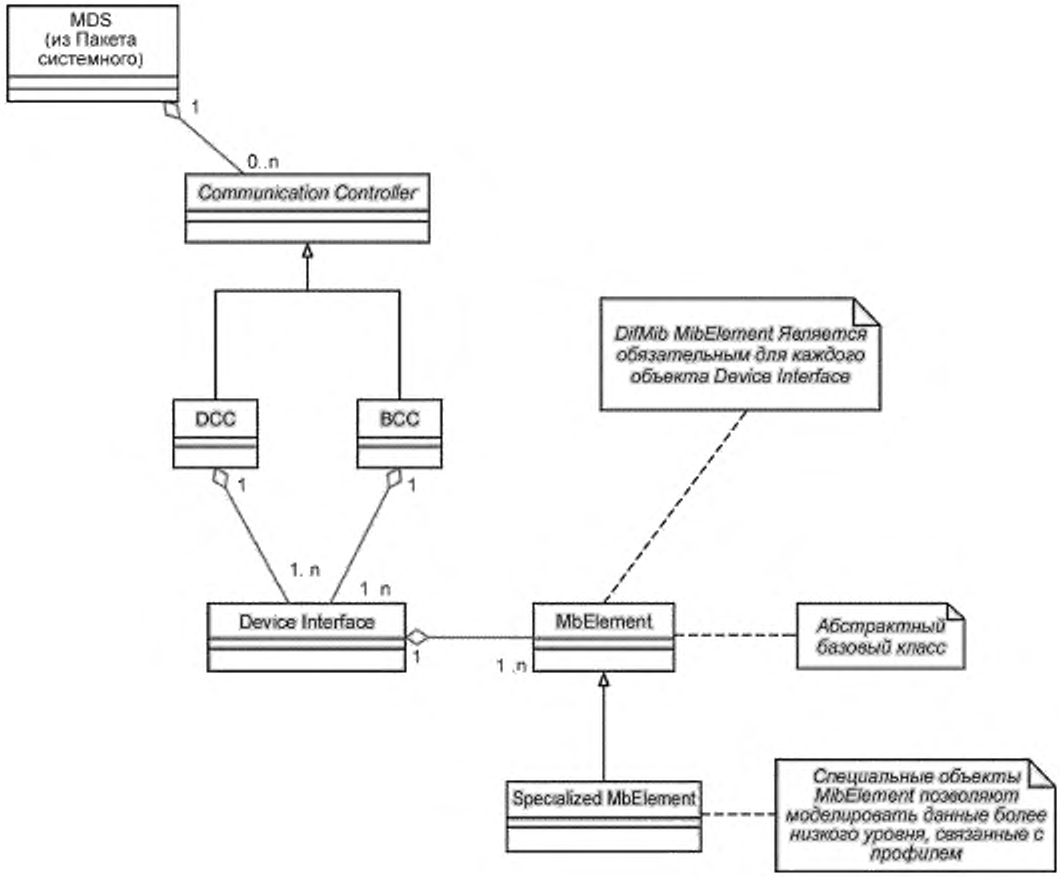


Рисунок 6.9 — Модель Пакета коммуникаций

6.8.1 Класс Communication Controller (Коммуникационный контроллер)

Класс Communication Controller представляет высоко- и низкоуровневый коммуникационный профиль медицинского прибора.

Объект класса Communication Controller — точка доступа для извлечения значений атрибутов объектов класса Device Interface и атрибутов объектов классов, производных от класса MibElement для того, чтобы получить управляющую информацию, связанную с коммуникациями для передачи данных.

Будучи абстрактным базовым классом, класс Communication Controller не может иметь конкретных представителей-экземпляров. Для него определены, однако, два специализированных производных класса: Bedside Communication Controller (BCC) и Device Communication Controller (DCC), которые могут иметь объекты-представителей. База данных MDIB медицинского прибора или не содержит ни одного объекта классов, производных от Communication Controller, или один объект класса BCC, или один объект класса DCC (в зависимости от назначения прибора).

6.8.2 Класс Device Communication Controller (Коммуникационный контроллер прибора, DCC)

Класс DCC — производный от класса Communication Controller. Его объекты используются в процессе работы медицинских приборов в качестве систем-агентов (отвечающих при ассоциировании приборов).

Объект класса DCC должен содержать один или несколько объектов класса Device Interface.

6.8.3 Класс Bedside Communication Controller (Прикроватный коммуникационный контроллер, BCC)

Класс Bedside Communication Controller — производный от класса Communication Controller. Его объекты используются в процессе работы медицинских приборов в качестве систем-менеджеров (запрашивающих ассоциирование приборов).

Объект класса BCC должен содержать один или несколько объектов класса Device Interface.

6.8.4 Класс Device Interface (Интерфейс прибора)

Объект класса Device Interface отражает частный интерфейс прибора, то, что называется порт. Порт представляет собой логический или физический конечный пункт ассоциации, для которого (например, статистически) могут быть независимо собраны данные, извлеченные из объектов классов, производных от класса MibElement.

И система-агент и система-менеджер могут иметь несколько логических и физических портов, в зависимости от выбранного способа реализации нижних уровней коммуникационной системы.

Объект класса Device Interface не доступен для функций сервиса CMDISE. Этот объект содержит, по крайней мере, один объект класса, производного от класса MibElement (то есть, объект подкласса Device Interface MibElement класса MibElement, который отражает свойства интерфейса прибора, включается в объект класса Device Interface, соответствующий этому интерфейсу), который может быть доступен посредством специального метода, определенного соответствующим объектом класса, производного от класса Communication Controller.

6.8.5 Класс MibElement (Объект базы данных управляющей информации)

Объект класса MibElement содержит статистику функционирования и данные о производительности для одного объекта класса Device Interface. Класс MibElement — всего лишь абстрактный базовый класс для производных специализированных классов, и поэтому не может иметь конкретных экземпляров-представителей.

Различные подклассы класса MibElement предназначены для группирования информации по определенным пакетам, которые могут быть универсальными или зависеть от конкретных транспортных профилей.

Объект класса, производного от класса MibElement, непосредственно не доступен. К его атрибутам можно получить доступ только через объект класса, производного от класса Communication Controller. Объект класса, производного от класса MibElement, не является частью базы данных MDIB прибора.

6.8.6 Класс Specialized MibElement (Специализированный производный класс (подкласс) от класса MibElement)

Управляющая информация для коммуникационной связи зависит от нижних уровней стека протоколов коммуникации (то есть, низкоуровневого профиля).

Настоящий стандарт, однако, определяет только два обобщенных подкласса класса MibElement:

- класс обязательно существующих объектов Device Interface MibElement, который описывает свойства некоторого интерфейса прибора и включается в объект класса Device Interface, соответствующий этому интерфейсу;
- класс необязательных объектов MibElement общей статистики коммуникации, который моделирует набор типовых статистических данных коммуникации, который применим в общем случае.

Специализированные подклассы класса MibElement определены в стандарте ИИЭР Р 1073.2.1.2.

6.9 Модель Архивного пакета

В Архивном пакете (Archival Package) имеют дело с классами, обеспечивающими хранение и представление данных о биомедицинских сигналах и различных состояниях и контекстной информации в архиве, доступном либо в неавтономном режиме, либо в автономном режиме.

На рисунке 6.10 показана модель классов Архивного пакета.

Модель Архивного пакета содержит классы, описанные в 6.9.1—6.9.7.

6.9.1 Класс Multipatient Archive (Общий архив данных о пациентах)

Объект класса Multipatient Archive группирует вместе несколько объектов класса Patient Archive, относящихся к разным пациентам.

Пример — Исследование лекарственного средства может быть задокументировано в форме объекта класса Multipatient Archive, содержащего несколько объектов класса Patient Archive, которые отражают влияние лекарства на отслеживаемые показатели жизненно важных функций конкретных пациентов.

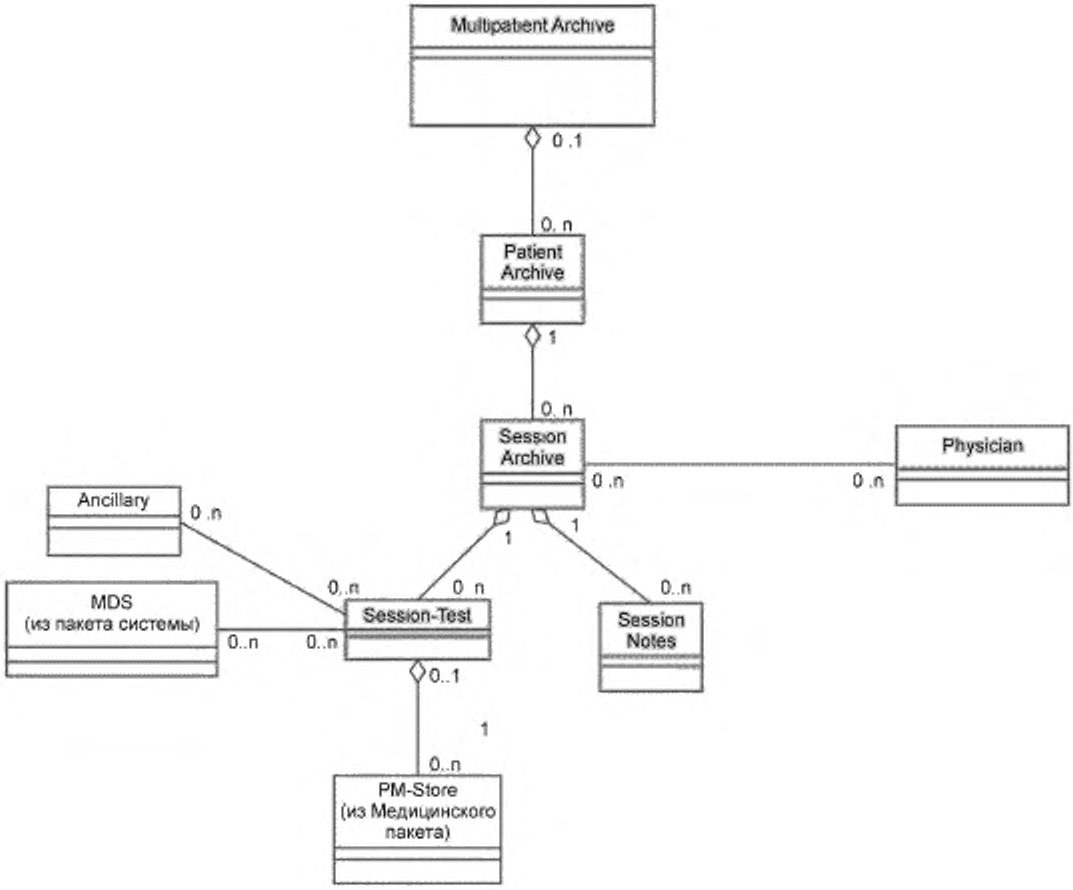


Рисунок 6.10 — Модель Архивного пакета

6.9.2 Класс Patient Archive (Архив данных о пациенте)

Объект класса Patient Archive группирует информацию, относящуюся к конкретному пациенту (например, данные показателей жизненно важных функций, данные о процессе лечения и демографические данные пациента), вместе в единственном объекте-архиве. Этот объект связан со статичными (то есть инвариантными) данными о конкретном пациенте, содержащимися только в соответствующем объекте класса Patient Demographics.

Пример — Больница может хранить данные о многократных посещениях единственного пациента в объекте класса Patient Archive, который содержит множество объектов класса Session Archive, в каждом из которых задокументирована информация о значениях показателей жизненно важных функций, полученная во время конкретного посещения определенного отделения больницы.

6.9.3 Класс Session Archive (Архив данных сессии лечения)

Объект класса Session Archive отражает данные о разовом визите пациента или периоде непрерывного пребывания в больнице или конкретном отделении больницы. Диагностические процедуры, проведенные за этот период, отражаются объектами класса Session Test, содержащимися в объекте класса Session Archive. Этот объект ссылается на динамически изменяющиеся (то есть вариативные) данные о конкретном пациенте, содержащиеся в объекте класса Patient Demographics.

6.9.4 Класс Physician (Врач)

Объект класса Physician отражает данные врача, ответственного за набор диагностических и терапевтических мероприятий за период времени, отраженный в соответствующем объекте класса Session Archive.

6.9.5 Класс Session Test (Проверка состояния здоровья в период сессии лечения)

Объект класса Session Test содержит информацию о показателях жизненно важных функций единичного пациента, зафиксированную во время единичного освидетельствования или диагностического мероприятия. Этот объект содержит метрические значения показателей жизненно важных функций в форме объектов класса PM-Store. Он также может содержать информацию об оборудовании, которое использовалось для того, чтобы сделать запись (в форме связей с объектами подклассов класса MDS и объектами класса Ancillary).

Пример — Информация о показателях жизненно важных функций, зарегистрированная во время снятия ЭКГ под дозированной физической нагрузкой, организованно сведена в объект класса Session Test.

6.9.6 Класс Session Notes (Примечания к сессии лечения)

Объект класса Session Notes играет роль контейнера для диагностических данных, деталей лечения и технологической информации в форме текстовых данных.

6.9.7 Класс Ancillary (Вспомогательный)

В настоящем стандарте класс Ancillary подробно не определяется. Объект этого класса присутствует в модели для отражения информации из источников, не относящихся к числу приборов, рассматриваемых в рамках настоящего стандарта, и может включаться в объект класса Session Test (или объект класса Session Test может на него ссылаться).

Пример — Поступающие графические данные, соответствующие требованиям стандартов MEDICOM (CEN ENV 12052) или DICOM (NEMA PS 3), разрешается включать в объект класса Session Test в качестве вспомогательных данных.

6.10 Модель Пакета пациента

В Пакете пациента (Patient Package) имеют дело со всей связанной с пациентом информацией, которая относится к сфере настоящего стандарта, но не является данными показателей жизненно важных функций, моделируемыми в Медицинском пакете.

На рисунке 6.11 показана модель классов Пакета пациента:

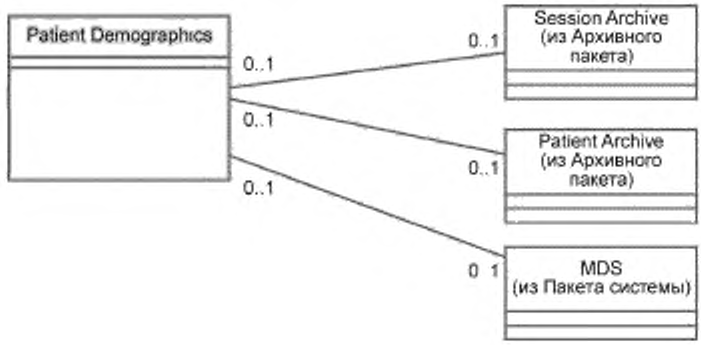


Рисунок 6.11 — Модель Пакета пациента

Модель Пакета пациента содержит только один класс (см. 6.10.1).

6.10.1 Класс Patient Demographics (Персональные данные пациента)

Объект класса Patient Demographics предназначен для хранения типового набора персональных данных пациента.

Настоящим стандартом предусматривается только минимально необходимый для работы с медицинскими приборами набор данных о пациенте. Сбор полной информации о пациенте выходит за рамки настоящего стандарта.

6.11 Информационная модель предметной области (DIM). Динамическая модель

6.11.1 Общие положения

Подраздел 6.11 описывает общую динамику поведения системы.

Необходимо отметить, что динамическое поведение объекта, вытекающее из обращения к сервису управления объектом, является частью описания объектов в разделе 7.

6.11.2 Конечный автомат комуникаций MDS (Finite state machine, FSM)

На рисунке 6.12 показан конечный автомат MDS для коммуницирующего медицинского прибора, соответствующего требованиям настоящего стандарта. Данный конечный автомат (FSM) используется, чтобы синхронизировать режимы работы системы-менеджера (то есть клиента) и системы-агента (то есть сервера).

После включения питания, в приборе выполняются все необходимые процедуры инициализации (загрузка прибора), по окончании которых прибор остается в состоянии разъединения и ожидает попытки соединения.

Обнаружив попытку соединения, прибор пытается установить логическое соединение (ассоциацию) с другим прибором. Система-менеджер (клиент) запрашивает ассоциацию (association requester), а система-агент (сервер) система — отвечает на запрос, т. е. выступает в качестве респондента (association responder). В процессе установления ассоциации выполняются базовые проверки на совместимость ассоциирующихся приборов.

После успешной ассоциации происходит обмен данными о конфигурации MDS (то есть актуальной структурой баз данных MDIB приборов) посредством использования базовых и дополнительных сервисов (в частности, объекта класса Context Scanner), как это описывается в настоящем стандарте. Предоставление дополнительной информации (например, значений атрибутов MDS) позволяет осуществить более глубокие проверки состояний и совместимости ассоциированных приборов.

После успешной конфигурации, происходит обмен медицинскими данными посредством использования базовых и дополнительных сервисов, как это описано в настоящем стандарте. Далее в процессе работы допустимо динамическое реконфигурирование. Если данный прибор или данный тип реконфигурирования не позволяет выполнить динамическое реконфигурирование в процессе работы, то должно быть предусмотрено специальное состояние реконфигурирование.

При наступлении события разъединения, прибор переходит в состояние разъединения.

Диаграмма, представленная на рисунке 6.12, не показывает, что происходит при возникновении ошибок. Фатальные ошибки выводят конечный автомат из рабочего состояния.

Примечание — Данный конечный автомат описывает только поведение коммуникационной системы MDS. Обычно прибор должен выполнять свою медицинскую функцию независимо от работы системы коммуникации.

Конечный автомат считается частью объекта класса MDS. Атрибут Status класса MDS отражает состояние автомата. Таким образом, MDS может объявлять об изменениях в состоянии автомата в форме отчетов о событии изменения значения этого атрибута.

Конкретные профили приложений должны использовать этот конечный автомат в качестве общего руководства, но в них могут быть определены некоторые отклонения, чтобы удовлетворять специфическим для данного профиля требованиям или допущениям.

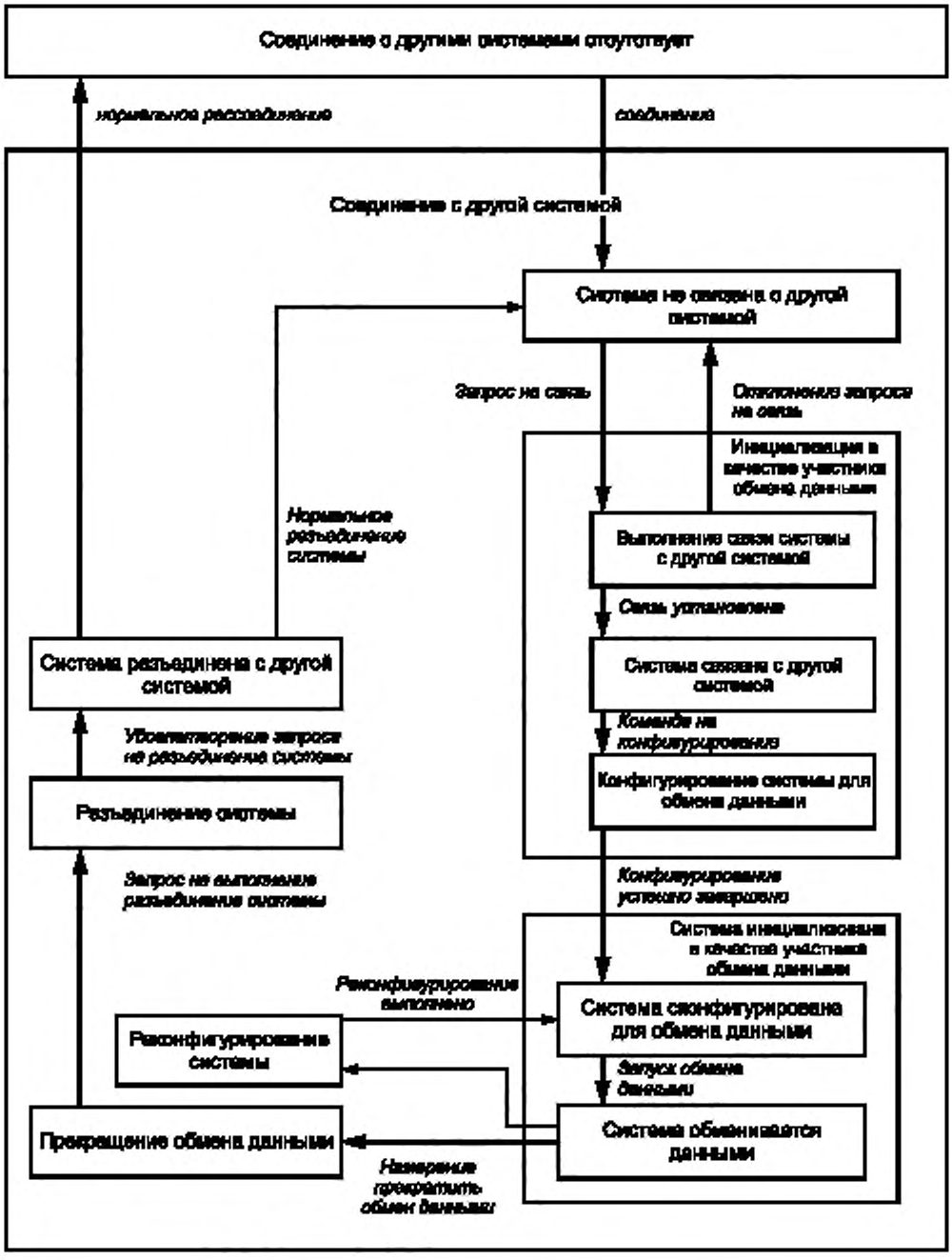


Рисунок 6.12 — Модель конечного автомата MDS

6.11.3 Коммуникационные системы. Диаграмма взаимодействия объектов при запуске

На рисунке 6.13 представлена диаграмма взаимодействия объектов, которая визуализирует фазу запуска процесса коммуникации после соединения двух приборов.

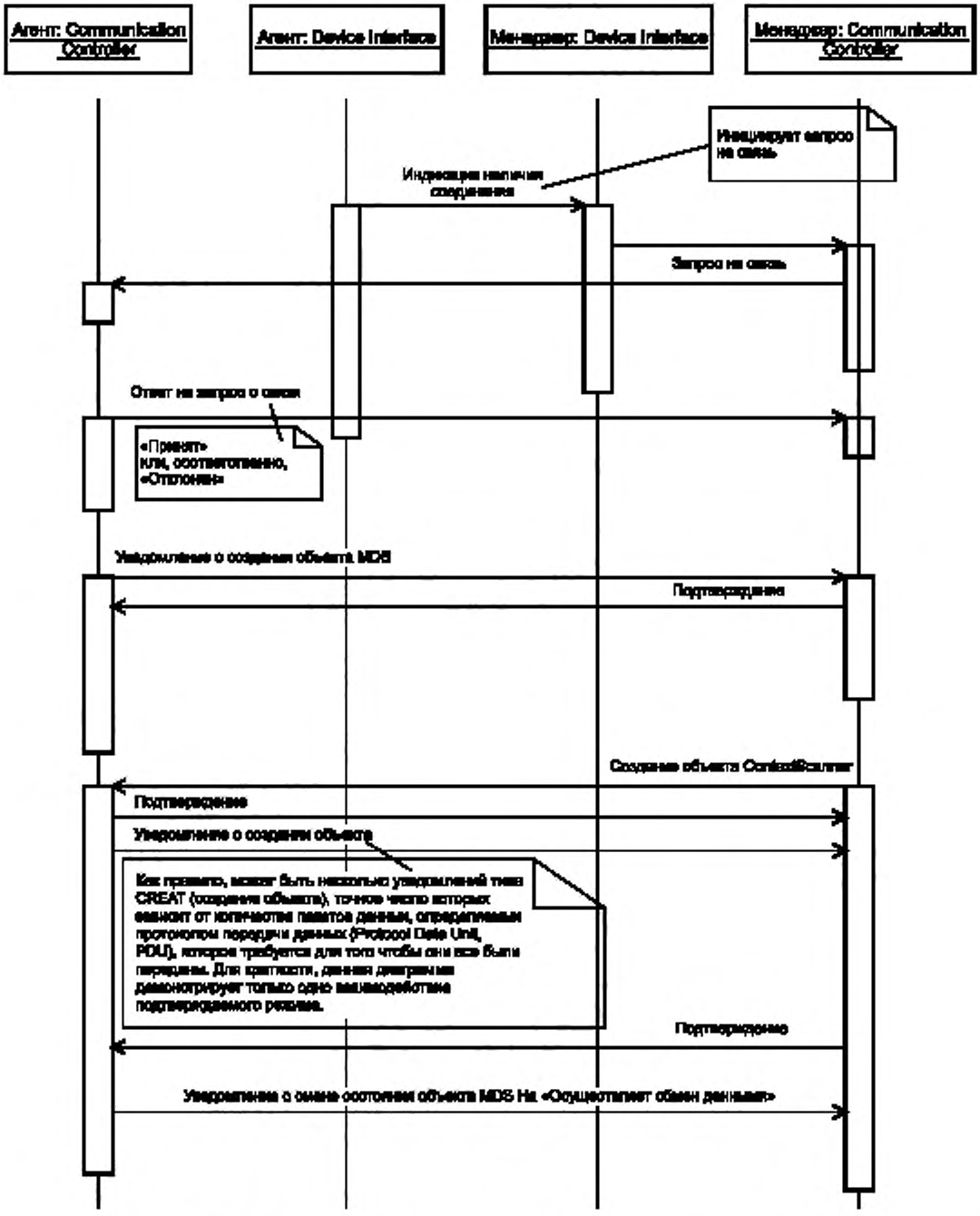


Рисунок 6.13 — Запуск процесса обмена данными после установления соединения между системами

В данном случае предполагается, что, концептуально, сообщениями обмениваются представители класса Communication Controller (используя интерфейс прибора).

Необходима какая-то форма индикации состояния соединения, чтобы система-менеджер могла узнать о новом агенте в сети. Конкретный механизм осуществления индикации зависит от конкретики низкоуровневой реализации коммуникационной системы; поэтому, он не описывается подробно в настоящем стандарте.

Конкретные профили приложений должны использовать эту диаграмму взаимодействия как общее руководство, но в них могут быть определены некоторые отклонения, чтобы удовлетворять специфическим для данного профиля требованиям или допущениям.

6.11.4 Пакет коммуникаций. Организация доступа к данным представителей подклассов класса MibElement

На рисунке 6.14 представлена диаграмма взаимодействия объектов, которая показывает, как система-менеджер получает доступ к данным объектов подклассов MibElement, используя классы, определенные в Пакете коммуникаций (см. 6.8).

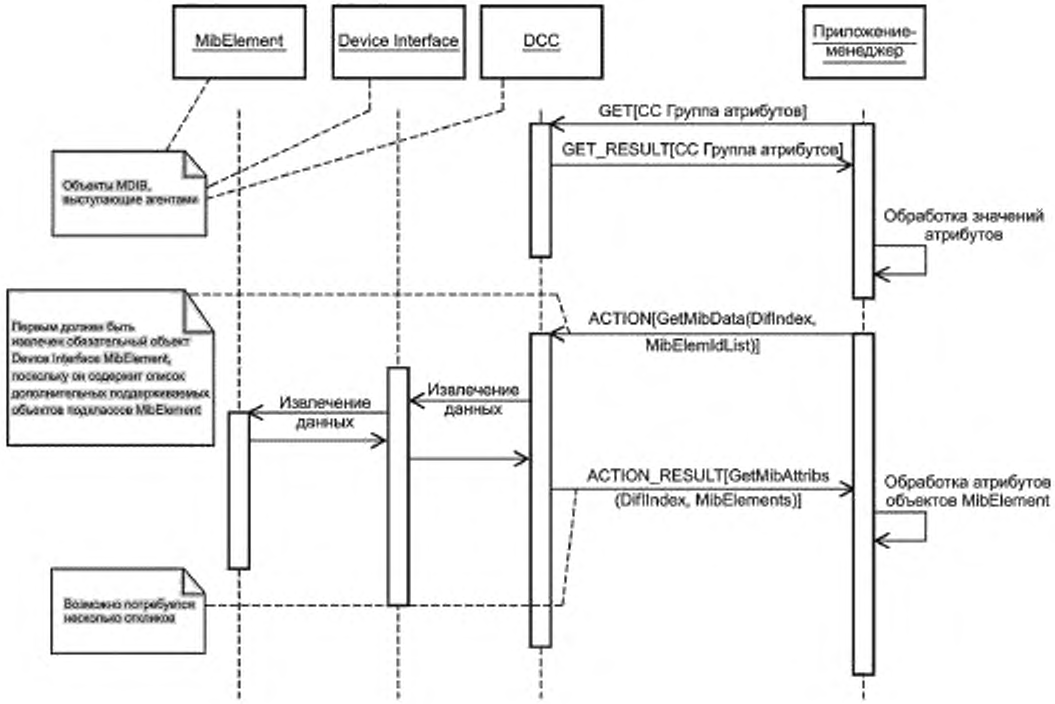


Рисунок 6.14 — Организация доступа к данным объектов подклассов класса MibElement

Данная диаграмма построена, исходя из следующих предположений:

- связь уже установлена между агентом и менеджером;
- фаза конфигурации уже закончена, и у менеджера есть полное представление о содержимом базы данных MDIB агента;
- объект класса DCC (Device Communication Controller) содержится в базе данных MDIB агента.

Сначала менеджер использует сервис GET, чтобы получить доступ к атрибутам всех объектов класса DCC и их значениям. Атрибуты определяют, сколько в данное время существует объектов класса Device Interface.

Затем Менеджер использует сервис ACTION с соответствующим методом класса Communication Controller, чтобы получить доступ к атрибутам обязательно присутствующего объекта класса Device

Interface MibElement. Количество атрибутов будет варьироваться, в случае, если у данного интерфейса присутствуют дополнительные объекты подклассов MibElement.

В последнем случае, менеджер может использовать ту же самую команду сервиса ACTION, чтобы дополнительную управляющую информацию, представленную в объектах подклассов MibElement.

6.11.5 Динамически изменяющиеся связи между объектами

Настоящий пункт посвящен связям между управляемыми медицинскими объектами (то есть объектами, которые определены в настоящем стандарте как управляемые).

В общем случае, связи между экземплярами объектов, определенными в моделях пакетов, являются динамически изменяющимися.

Пример — Модульный монитор состояния пациента моделируется объектом класса MDS. Модули измерений моделируются объектами класса VMD. Если к монитору подключается новый модуль, тут же образуется новая связь между объектом класса MDS и новым экземпляром объекта VMD.

Системы — коммуникационные агенты — использует определенные в настоящем стандарте сервисы, чтобы объявить о событиях изменения конфигурации прочим связанным системам. У этих систем-менеджеров изменяется представление о содержании базы данных MDIB системы-агента.

При этом архив показателей жизненно важных функций должен не только обновлять данные о текущей конфигурации подключенных приборов, но и должен постоянно хранить данные о событиях их подключения и отключения.

Пример — Экземпляр объекта Session Archive отражает пребывание пациента в отделении интенсивной терапии (ICU). В течение этого периода новые приборы подключаются к пациенту, чтобы увеличить число регистрируемых показателей жизненно важных функций. Как только состояние пациента стабилизируется, данные приборы отключаются. Объект класса Session Archive не должен удалять зафиксированные данные, при отключении фиксирующего их прибора.

Таким образом, в определенных приложениях (например, архивных), присутствует информация о связях между экземплярами объектов, которая должна быть получена.

При необходимости, связи сами по себе могут считаться особыми управляемыми объектами, как это показано на рисунке 6.15.

Пример на рисунке 6.15 показывает связь между классом Session Archive и классом MDS. Связь представлена классом. У этого класса есть атрибуты, которые отражают информацию, например, о времени установления соединения и разъединения.

У моделирования связей в форме классов есть то преимущество, что информация может быть представлена в форме атрибутов. Это позволяет не добавлять указанные атрибуты одному или обоим связанным классам.

То, как динамически изменяющиеся связи объектов обрабатываются системами, которые соответствуют определениям в настоящем стандарте, определено в 6.11.5.1 и 6.11.5.2.

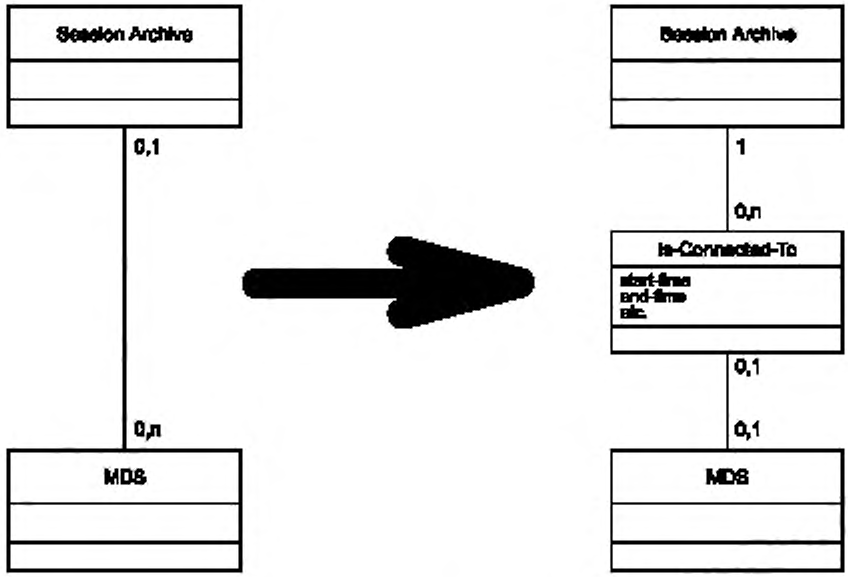


Рисунок 6.15 — Пример представления связи классом

6.11.5.1 Динамически изменяющиеся связи между объектами в системах связи

Связи между экземплярами объектов, которые определены в моделях пакетов, содержат информацию об их (связей) конфигурации. Объект Context Scanner предоставляет информацию о конфигурации в форме уведомлений о создании объекта и удалении объекта в связи. Для систем связи не определены никакие средства для постоянного хранения полной информации о старой конфигурации.

Другие связи между экземплярами объектов (например, чтобы сослаться в полученных данных на исходный сигнал) определены в форме атрибутов соответствующих объектов (например, атрибута VMO-Source-List объекта подкласса класса Metric). О динамических изменениях этих признаков объявляют сообщения о событиях изменения атрибута.

6.11.5.2 Динамически изменяющиеся связи между объектами в архивных системах

Для постоянного хранения информации о конфигурации может потребоваться архивная система. В этом случае, соответствующие связи предполагается представлять классами, как показано в 6.11.5.

Архивная система, которая использует формат данных, соответствующий требованиям настоящего стандарта, должна обеспечить средства хранения (то есть, архив) атрибутов динамически изменяющихся связей между объектами.

7 Определения объектов DIM

7.1 Общие положения

Настоящий раздел содержит определения для всех объектов в DIM. Пакеты, установленные в данной модели, используются для распределения объектов по категориям. Атрибуты, поведение и уведомления определяются для каждого класса объекта.

7.1.1 Условное обозначение

Каждый объект описывается в отдельном подразделе (см. 7.2—7.10). Остальные подразделы описывают атрибуты, поведение и уведомления для объектов.

В подразделе объект описывается следующим образом:

Объект:	Определяет название объекта
Описание:	Дается короткое информативно-текстуальное описание объекта.
Производный от:	Описываются потенциальные базовые классы объекта.
Связывание имен:	Описывается атрибут, который уникально идентифицирует экземпляр объекта в данном контексте. Для управляемых объектов данное определение является атрибутом Handle (описатель), а контекст — системой прибора (например, контекст отдельной MDS). См. также 7.1.2.5.
Зарегистрирован как:	Определяется термин, который описан в номенклатуре для уникальной идентификации [например, идентификатор объекта (OID), код] объекта.

Каждый атрибут объекта определен в подразделе атрибутов. В таблицах определены названия атрибутов, уникальные идентификаторы атрибутов, типы данных атрибутов и определенные квалификаторы. Квалификаторы имеют следующее значение:

M — атрибут обязателен;

O — атрибут необязателен (дополнительный атрибут);

C — атрибут условный; доступность атрибута зависит от предустановленного условия.

Если не указано иное, таблицы с определениями атрибутов не показывают унаследованные атрибуты снова. Другими словами списки атрибутов всех базовых классов необходимо проверять на наличие полного списка атрибутов объектов.

Атрибуты относятся к группам атрибутов или объединяются в группы так, чтобы их можно было классифицировать по способу их применения (например, статическая контекстная информация, динамичная контекстная информация, измерения значений). Объединение по группам позволяет эффективно работать с необязательными атрибутами: сервис GET позволяет легко найти все члены группы, так чтобы приложение могло определить, какой атрибут на данный момент присутствует в определенном экземпляре объекта.

Группы атрибутов могут расширяться. Другими словами производный класс объекта способен добавлять дополнительные члены к наследующей группе атрибутов.

Группы атрибутов также определены в таблицах, где указаны идентификация группы и список членов группы. Наследующие группы атрибутов не указаны в таблицах групп атрибутов, если данные группы не способны расширяться.

В подразделе Поведение описаны конкретные методы или функции, обеспечиваемые классом объекта. Данные методы можно активировать с помощью сервиса CMDISE ACTION.

События, создаваемые классом объекта (отличного от уведомления об изменении базового атрибута), описываются в подразделе Уведомления. Объект сообщает о данных событиях с помощью сервиса CMDISE EVENT REPORT (Отчет о событии элемента CMDISE).

7.1.2 Общие типы данных

Настоящий пункт описывает набор типов данных языка ASN.1, которые используются в определениях объектов.

7.1.2.1 Данные целого типа и данные типа строки бит

Для представления целых чисел определения объекта используют только типы данных фиксированного размера. Данные типа строки бит представляют собой битовое поле, в котором каждый отдельный бит имеет определенное значение. Применяются следующие целочисленные типы данных и типы данных строки бит:

```
--
-- 8-битное беззнаковое целое число
--
INT-U8 ::= INTEGER (0..255)
--
-- 8-битное целое число со знаком
--
```

```
INT-I8 ::= INTEGER (-128..127)
--
-- 16-битное беззнаковое целое число
--
INT-U16 ::= INTEGER (0..65535)
--
-- 16-битное целое число со знаком
--
INT-I16 ::= INTEGER (-32768..32767)
--
-- 32-битное беззнаковое целое число
--
INT-U32 ::= INTEGER (0..4294967295)
--
-- 32-битное целое число со знаком
--
INT-I32 ::= INTEGER (-2147483648..2147483647)
--
-- 16-битная битовая строка
--
BITS-16 ::= BIT STRING (SIZE(16))
--
-- 32-битная битовая строка
--
BITS-32 ::= BIT STRING (SIZE(32))
```

Примечания

1 При интерпретации целых чисел следует учитывать представление (например, прямой порядок байтов или обратный порядок байтов). Коммуникационные системы согласовывают это представление на этапе ассоциации (т. е. согласовывают синтаксис передачи). Архивные форматы данных должны обеспечивать механизм уникальной идентификации представления целых чисел (например, поле в заголовке спецификации).

2 В определениях объекта данные целого типа и данные типа битовой строки с именованными константами или битами также могут использовать вышеуказанное представление для удобства. Вышеуказанное представление не соответствует синтаксису языка ASN.1, но его легко можно преобразовать в правильный синтаксис.

7.1.2.2 Идентификационный тип данных

Всем элементам (например, классам, объектам, типам измерений), которые требуют использования уникальной идентификации, присваивают идентификаторы OID. Набор действующих идентификаторов OID для данного стандарта описан в ИСО/ИИЭЭ 11073-10101. Номенклатура разделена на ряд сегментов, и каждый сегмент имеет свой диапазон 16-ти битных кодов. Другими словами, 16-ти битный код — это код, зависящий от контекста.

Тип 16-ти битных идентификационных данных представлен в следующем виде:

```
--
-- тип OID согласно указанному в номенклатуре
-- (не путать с OID языка ASN.1)
--
OID-Type ::= INT-U16 -- 16-битный целый тип
```

Для идентификаторов, которые не являются частью стандартной номенклатуры (т. е. личный код или код, заданный производителем), специальный тип определяется следующим образом:

```
--
-- Личный OID
--
PrivateOid ::= INT-U16
```

7.1.2.3 Тип данных Описатель (Handle)

Тип данных Описатель используется для эффективной, локально уникальной идентификации всех управляемых медицинских экземпляров объекта. (*Локально уникальный* означает уникальный в рамках одного контекста системы MDS.) Данный тип данных определяется следующим образом:

```
--
-- описатель
--
HANDLE ::= INT-U16
```

7.1.2.4 Тип данных Номер экземпляра (Instance number)

Тип данных Номер экземпляра используется для установления различий между экземплярами класса или объекта одного типа или экземплярами объекта, управление которыми не осуществляется прямо (т. е. использованные, например, в качестве атрибута Связывания имен для объектов Операция). Данный тип данных определяется следующим образом:

```
--
-- Номер экземпляра
--
InstNumber ::= INT-U16
```

7.1.2.5 Идентификация глобального объекта

Типы данных Описатель и Номер экземпляра должны быть уникальными для одного определенного контекста именования (например, описатели уникальны в рамках минимум одного контекста системы MDS). Уникальность позволяет идентифицировать экземпляр объекта в рамках его контекста именования с помощью одного единственного небольшого идентификатора.

Для обращения к более крупным системам поле контекстного идентификатора на уровне системы MDS в рамках базы MDIB добавляется к типу данных описателя, так чтобы можно было различить многочисленные системы прибора. Этот тип данных глобального описателя определяется следующим образом.

```
--
-- Идентификатор контекста системы MDS
--
MdsContext ::= INT-U16

--
-- Глобальный описатель позволяет идентифицировать объект в более крупных
-- системах
--
GLB-HANDLE ::= SEQUENCE {
    context-id              MdsContext,
    handle                  HANDLE
}

--
-- Управляемый идентификатор OID в качестве типа для полной
-- идентификации глобального объекта
ManagedObjectId ::= SEQUENCE {
    m-obj-class             OID-Type,
    m-obj-inst              GLB-HANDLE
}
```

Пример — Медицинский прибор может быть связан с другими медицинскими приборами (т. е. суб-приборами). В информационной базе MDIB данный прибор может моделировать эти суб-приборы в качестве отдельных объектов системы MDS с их собственным контекстом именования. Таким образом, конфликтов пространств имен (например, дублированные значения описателя, дублированные коды номенклатуры) можно избежать без переназначения значений описателя. Системе менеджера необходимо интерпретировать идентификаторы контекста системы MDS вместе со значениями опи-

сателя, чтобы уникально идентифицировать экземпляры объектов в рамках данной составной информационной базы MDIB. Идентификаторы контекста присваиваются, когда уведомления о создании объекта Context Scanner (Сканер контекста) формируют базу MDIB.

Предположения и возможные ограничения касательно различных контекстов именования в рамках базы MDIB не зависят от профиля. Тем не менее, они описаны в серии стандартов ИСО/ИИЭЭ P11073-202xx.

7.1.2.6 Тип данных Идентификатор типа (Type ID)

Тип данных Идентификатор типа используется в объектах VMO и объектах системы VMS для обеспечения специальной статической информации о типе экземпляра объекта (например, артериальное давление может иметь тип объекта Numeric). Используются коды, указанные в номенклатуре. Номенклатура содержит ряд разделов, а значения кода являются уникальными только в рамках одного раздела. Так как тип данных Идентификатор типа не должен зависеть от контекста, то также предоставлен раздел кода номенклатуры. Настоящий тип данных представлен следующим образом:

```
--
-- Идентификатор типа
--
TYPE ::= SEQUENCE {
    partition               NomPartition,
    code                    OID-Type
}

--
-- Существуют указанные ниже разделы номенклатуры
--
NomPartition ::= INT-U16 {
    nom-part-unspec (0),
    nom-part-obj (1),               -- раздел объектно-ориентированных элементов
    nom-part-metric (2),            -- раздел метрик [система оперативного сбора данных
                                    -- и диспетчерского контроля SCADA]
    nom-part-alert (3),             -- раздел сигналов тревоги/событий
    nom-part-dim (4),               -- раздел размеров
    nom-part-vattr (5),             -- раздел виртуальных атрибутов для объектов Operation
    nom-part-pgrp (6),              -- раздел идентификатора группы параметров
    nom-part-sites (7),             -- измерения и локализация участков тела
    nom-part-infrastruct (8),       -- раздел элементов инфраструктуры
    nom-part-fef (9)                -- раздел формата обмена файлами
    nom-part-ecg-extn (10),         -- раздел расширений ЭКГ
    nom-part-ext-nom (256),         -- идентификаторы других номенклатур и словарей
    nom-part-priv (1024)            -- частный раздел
}
```

7.1.2.7 Тип данных Утверждение значения атрибута (Attribute value assertion)

Ряд сервисов, указанных в модели сервисов в разделе 8, обеспечивают доступ к атрибутам объекта (например, GET, SET). Обычно атрибут необходимо идентифицировать с помощью идентификатора объектов. Тип данных атрибута сам по себе зависит от этого идентификатора. Тип данных Утверждение значения атрибута представляет собой такую пару идентификатор-значение и представлен следующим образом:

```
--
AVA-Type ::= SEQUENCE {
    attribute-id                OID-Type,
    attribute-value                 ANY DEFINED BY attribute-id
}
```

7.1.2.8 Тип данных Список атрибутов (Attribute list)

Зачастую необходим список пар значения идентификатор-атрибут. Тип данных Список атрибутов — это особый тип данных, который служит для данной ситуации и представлен следующим образом:

```
--
AttributeList ::= SEQUENCE OF AVA-Type
```

7.1.2.9 Тип данных Список идентификаторов атрибута (Attribute ID list)

Зачастую используется список идентификаторов атрибута. Тип данных Список идентификаторов атрибута — это особый тип данных, который обеспечивается для удобства и представлен следующим образом:

```
--
AttributeIdList ::= SEQUENCE OF OID-Type
```

7.1.2.10 Тип данных Тип с плавающей точкой (Floating point)

Для эффективного определения объектов используют тип данных Тип с плавающей точкой, который является специальным типом данных для представления чисел с плавающей точкой. Предполагается, что данный тип данных имеет 32 бита. Настоящий тип данных представлен следующим образом:

```
--
-- 32-битный тип с плавающей точкой; целый тип числа является лишь
-- символом-заполнителем
--
FLOAT-Type ::= INT-U32
```

Конкретный формат числа с плавающей точкой либо явно согласуется во время соединения, либо косвено определяется с помощью контекста приложения при соединении.

7.1.2.11 Тип данных Относительное время (Relative time)

Тип данных Относительное время — это определение времени высокого разрешения, относящееся к какому-либо событию (например, событие синхронизации при автозагрузке). Настоящий тип данных используется для размещения событий относительно друг друга. Представлен в следующем виде:

```
--
-- Относительное время имеет разрешение 125 мкс [самый младший
-- разряд (LSB)], которое является достаточным при частотах выборки до 8
-- кГц и временных диапазонов до 6.2 дней
--
RelativeTime ::= INT-U32
```

Необходимо отметить, что погрешность времени указана в самой системе.

7.1.2.12 Тип данных Относительное время высокого разрешения (High-resolution relative time)

Если недостаточно разрешения или временного диапазона ранее установленного типа данных Относительное время, определяют тип данных Относительное время высокого разрешения. Тип данных длиной в 64 бита. Однако так как нет определенного типа данных целого 64-битного числа, используется непрозрачная (например, строка) структура данных. Тип представлен следующим образом:

```
--
-- 64-битное (8 байтов) время высокого разрешения, LSB представляет собой 1 мкс
--
HighResRelativeTime ::= OCTET STRING (SIZE(8))
```

Необходимо отметить, что погрешность времени указана в самой системе.

7.1.2.13 Тип данных Абсолютное время (Absolute time)

Тип данных Абсолютное время указывает на время дня с разрешением минимум 1 сек. В целях эффективности, значения в структуре закодированы в двоично-десятичный код (BCD) (т. е. 4-битные полубайты). Например, 1996 год представлен шестнадцатеричным значением 0x19 в поле век и шестнадцатеричным значением 0x96 в поле год. Данный формат можно легко перевести в представления, основанные на символах или целых числах. Тип данных абсолютного времени представлен следующим образом:

```
--
AbsoluteTime ::= SEQUENCE {
    century                 INT-U8,
    year                    INT-U8,
    month                   INT-U8,
```

```
    day                     INT-U8,
    hour                    INT-U8,
    minute                  INT-U8,
    second                  INT-U8,
    sec-fractions           INT-U8 -- 1/10 и 1/100 секунды, если доступно
}
```

7.1.2.14 Тип данных дата (Date)

Тип данных дата используется для определения определенного дня календаря. Чтобы облегчить трансформацию, тип данных имеет тот же код (например, BCD), что и тип данных абсолютное время. Тип данных дата представлен следующим образом:

```
--
Date ::= SEQUENCE {
    century                 INT-U8,
    year                    INT-U8,
    month                   INT-U8,
    day                     INT-U8
}
```

7.1.2.15 Тип данных Рабочее состояние (Operational state)

Тип данных Рабочее состояние определяет, активирован или деактивирован ли определенный объект или свойство. Определения берутся из ИСО/МЭК 10164-2 и представлены следующим образом:

```
--
OperationalState ::= INT-U16 {
    disabled (0),
    enabled (1),
    notAvailable (2)
}
```

7.1.2.16 Тип данных Административное состояние (Administrative state)

Тип данных Административное состояние определяет, разблокирован или заблокирован ли определенный объект. Определения берутся из ИСО/МЭК 10164-2 и представлены следующим образом:

```
--
AdministrativeState ::= INT-U16 {
    locked(0),
    unlocked(1),
    shuttingDown(2)
}
```

7.1.2.17 Тип данных Цвет (Colour)

Тип данных Цвет представляет собой основные RGB-цвета и представлен следующим образом:

```
--
SimpleColour ::= INT-U16 {      -- RGB
    col-black (0),              -- 000
    col-red (1),                -- 100
    col-green (2),              -- 010
    col-yellow (3),             -- 110
    col-blue (4),               -- 001
    col-magenta (5),            -- 101
    col-cyan (6),               -- 011
    col-white (7)               -- 111
}
```

7.1.2.18 Тип данных Локаль (Locale)

Тип данных Локаль должен использоваться для установления языка и информации по кодированию для типов данных, которые представляют собой строки удобночитаемого текста. Данный тип данных представлен следующим образом:

```
--
Locale ::= SEQUENCE {
    language        INT-U32,        -- из ИСО 639-1 или ИСО 629-2, кодирование приведено ниже
```

```
    country          INT-U32,          -- из ИСО 3166-1, ИСО 3166-2 или ИСО 3166-3,
                                       -- кодирование приведено ниже
    charset          CharSet,          -- формат кодирования символов
    str-spec         StringSpec
}
--
-- Названия кодировки соответствуют требованиям Агентства по
-- выделению имен и уникальных параметров протоколов Интернет
-- (IANA), числовые константы являются значениями IANA MIBenum для
-- зарегистрированных кодировок
--
CharSet ::= INT-U16 {
    charset-unspec(0),
    charset-iso-10646-ucs-2(1000),    -- схема кодирования двух-октетных
                                      -- символов согласно ISO 10646,
                                      -- обратный порядок байтов
    charset-iso-10646-ucs-4(1001),    -- схема кодирования четырех-октетных
                                      -- символов согласно ISO
                                      -- 10646,обратный порядок байтов
    charset-iso-8859-1(4),            -- кодирование согласно ИСО/МЭК 8859-1
    charset-iso-8859-2(5),            -- кодирование согласно ИСО/МЭК 8859-2
    charset-iso-8859-3(6),            -- кодирование согласно ИСО/МЭК 8859-3
    charset-iso-8859-4(7),            -- кодирование согласно ИСО/МЭК 8859-4
    charset-iso-8859-5(8),            -- кодирование согласно ИСО/МЭК 8859-5
    charset-iso-8859-6(9),            -- кодирование согласно ИСО/МЭК 8859-6
    charset-iso-8859-7(10),           -- кодирование согласно ИСО/МЭК 8859-7
    charset-iso-8859-8(11),           -- кодирование согласно ИСО/МЭК 8859-8
    charset-iso-8859-9(12),           -- кодирование согласно ИСО/МЭК 8859-9
    charset-iso-8859-10(13),          -- кодирование согласно ИСО/МЭК 8859-10
    charset-iso-8859-13(109),         -- кодирование согласно ИСО/МЭК 8859-13
    charset-iso-8859-14(110),         -- кодирование согласно ИСО/МЭК 8859-14
    charset-iso-8859-15(111),         -- кодирование согласно ИСО/МЭК 8859-15
    charset-iso-2022-kr(37),          -- кодирование согласно RFC 1557
                                      -- (Кодирование корейских символов)
    charset-ks-c-5601(36),            -- кодирование согласно Отраслевому
                                      -- стандарту Кореи,KSC 5601-1987
    charset-iso-2022-jp(39),          -- кодирование согласно RFC 1468
                                      -- (Кодирование японских символов)
    charset-iso-2022-jp-2(40),        -- кодирование согласно RFC 1554
                                      -- (Кодирование японских символов)
    charset-jis-x0208(63),            -- кодирование согласно JIS
                                      -- X0208:1983,1990
    charset-iso-2022-cn(104),         -- кодирование согласно RFC 1922
                                      -- (Кодирование японских символов)
    charset-gb-2312(2025)             -- кодирование согласно Набору
                                      -- графических китайских символов, GB
                                      -- 2312:1980
}
StringSpec ::= SEQUENCE {
    str-max-len INT-U16,              -- максимальная длина строки
    str-flags StringFlags             -- конкретные флажки слов для представления
                                      -- и кодирования строк
}
    StringFlags ::= BITS-16 {
    str-flag-nt(0)                    -- строки завершаются нулевым байтом
}
```

Поле Locale::language должно отображать представление по ИСО/МЭК 646 в нижнем регистре для двухсимвольного идентификационного кода языка из ISO 639-1 или ISO 639-2. Для удобства обработки идентификатор языка хранится в целочисленном 32-битном поле. Первый октет кода хранится в наиболее значимом байте данного поля. Неиспользуемые октеты в данном поле заполняются НУЛЕВЫМИ (NULL) байтами.

Пример —

Язык:	***"Английский"***
Идентификатор языка:	***"en"***
Кодирование:	***65 6E 00 00$_h$***

Поле Locale::country должно отображать представление по ИСО/МЭК 646 в верхнем регистре двухсимвольного идентификационного кода языка из ISO 3166-1, ISO 3166-2 или ISO 3166-3. Для удобства обработки идентификатор языка хранится в целочисленном 32-битном поле. Первый октет кода хранится в наиболее значимом байте данного поля. Неиспользуемые октеты в данном поле заполняются НУЛЕВЫМИ (NULL) байтами.

Код страны можно использовать для установления различий между определенными аспектами одного и того же языка, используемого в разных языках, например, английский в США по отношению к английскому в Великобритании.

Если страна не определена, в данном поле указывают 0.

Пример —

Страна:	***"США"***
Идентификатор языка:	***"US"***
Кодирование:	***55 53 00 00$_h$***

Поле Locale::charset указывает на схему кодирования символов, используемую в типах данных строки, которые используются для представления удобочитаемого текста.

Для взаимодействия рекомендуется использовать схему кодирования символов iso-10646-ucs-2. Данная схема кодирования соответствует ИСО/МЭК 10646 с 2-октетным (т. е. 16 битов на символ) кодированием с обратным порядком байтов, которое представляет собой основную многоязыковую плоскость (BMP). Коды символов в рамках ИСО/МЭК 10646 прямо не соответствуют глифам, т. е. графическому представлению символа. ИСО/МЭК 10646 также не зависит от языка. Другие значения Locale::charset могут в значительной степени зависеть от языка, так как они также обозначают определенный набор символов.

7.1.2.19 Тип данных Ссылка на внешнюю номенклатуру (External nomenclature reference)

В определенных случаях требуется ссылка на стандартные системы кодирования (т. е. номенклатуры), на которые не распространяется действие настоящего стандарта.

Пример — Номенклатура, описанная в настоящем стандарте, не описывает диагностические коды и коды процедур. Тем не менее, можно ссылаться на другую систему кодирования и обеспечивать информацию в виде внешнего кода.

Тип данных Ссылка на внешнюю номенклатуру — это особый тип данных, который указан для этой цели в следующем виде:

```
--
ExtNomenRef ::= SEQUENCE {
   nomenclature-id          OID-Type,      -- идентификатор внешней номенклатуры
                                           -- из раздела внешней номенклатуры
   nomenclature-code        ANY DEFINED BY nomenclature-id
}
```

7.1.2.20 Тип данных Список связей с внешними объектами (External object relation list data type)

В определенных случаях управляемые медицинские объекты, описанные в информационной модели предметной области (DIM), могут быть связаны с другими объектами, которые не описаны в настоящем стандарте (т. е. они являются внешними по отношению к определениям).

Тип данных Список связей с внешними объектами можно использовать для получения информации о подобных объектах и особой связи с ними. Этот тип данных представлен следующим образом:

```
--
-- ExtObjRelationList
--
ExtObjRelationList ::= SEQUENCE OF ExtObjRelationEntry

ExtObjRelationEntry ::= SEQUENCE {
    relation-type               OID-Type,
    related-object              OID-Type,
    relation-attributes         AttributeList
}
```

Примеры

1 В определенных ситуациях необходимо указать определенную производственную информацию (например, серийный номер) передатчика, который используется для проведения измерений. Передатчик не определен в настоящем стандарте как управляемый медицинский объект. Поэтому экземпляры объекта VMD используют запись о связи (отношении) для предоставления информации, например, {relation-type = is-connected; related-object = Transducer; relation-attributes = {model, "A-Model," serial-number = "12345"}}.

2 Определенное числовое значение измерения подтверждается на соответствие медсестрой вручную. Таким образом, система построения графиков хранит в себе информацию о ручном подтверждении информации. Медсестра не смоделирована в настоящем стандарте как объект. Поэтому система построения графиков использует запись об отношении в качестве дополнительного атрибута объекта Numeric, например, {relation-type = validated-by; related-object = Nurse; relation-attributes = {name, "C. Smith," date, "041295"}}.

Тип данных Список связей с внешними объектами очень сильная концепция для расширения информационной модели без определения дополнительных объектов.

7.2 Главный объект (базовый класс)

Объект: Главный

Описание: Главный объект — это общепринятая база наследования свойств для всех объектов в модели DIM.

Производное от: —

Связывание имен: —

Зарегистрировано как: MDC_MOC_TOP

7.2.1 Атрибуты

Класс Главного объекта определяет атрибуты, указанные в таблице 7.1.

Таблица 7.1 — Атрибуты класса главного объекта

Название атрибута	ID (Идентификатор) атрибута	Тип атрибута	Примечание	Квалификатор
Class	MDC_ATTR_CLASS	OID-Type	Обозначает идентификатор класса; идентификаторы берутся из раздела номенклатуры, ориентированного на объект	н/д
Name-Binding	MDC_ATTR_NAME_BINDING	OID-Type	Обозначает идентификатор атрибута Связывания имен, например, HANDLE; идентификаторы берутся из сегмента номенклатуры, ориентированной на объект	н/д

Класс главного объекта не определяет никакие группы атрибутов.

7.2.2 Поведение

Класс главного объекта не определяет никакие специальные методы.

7.2.3 Уведомления

Класс главного объекта определяет события, указанные в таблице 7.2.

Таблица 7.2 — События класс главного объекта

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Attribute-Update	Подтвержден/ неподтвержден	MDC_NOTI_ATTR_UPDATE	AttributeList	—

Уведомление об обновлении атрибута позволяет всем объектам сообщать значения своих атрибутов с помощью базового отчета о событии. Однако не рекомендуется использовать данное уведомление для системы с множественными экземплярами объектов. Вместо этого следует использовать объекты Scanner (Сканнер).

7.3 Объекты в Медицинском пакете

Определения объектов в Медицинском пакете указаны в пунктах 7.3.1—7.3.13.

7.3.1 VMO (Виртуальный медицинский объект)

Объект: VMO

Описание: Объект VMO — базовый класс для всех объектов медицинского назначения в модели. Это обстоятельство позволяет установить непротиворечивую систему наименования и обозначения в модели Медицинского пакета. Будучи базовым абстрактным классом, VMO не может иметь конкретных представителей — экземпляров.

Производное от: Главный объект

Связывание имен: Описатель (значения атрибута Описатель (Handle) достаточно для уникальной идентификации в системе прибора экземпляра объекта, производного от объекта VMO)

Зарегистрировано как: MDC_MOC_VMO

7.3.1.1 Атрибуты

Класс объекта VMO определяет атрибуты, представленные в таблице 7.3.

Таблица 7.3 — Атрибуты класса объекта VMO

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Type	MDC_ATTR_ID_ TYPE	TYPE	Обозначает особый статический объект данного объекта, как указано в метрическом сегменте номенклатуры или сегменте, ориентированном на объект	M
Handle	MDC_ATTR_ID_ HANDLE	HANDLE	Локальная идентификация уникального сокращения слова	M
Label-String	MDC_ATTR_ID_ LABEL_STRING	OCTET STRING	Текстовое представление идентификатора типа	O
Ext-Obj-Relations	MDC_ATTR_EXT_ OBJ_RELATION	ExtObjRelation- List	Отношения к объектам, которые не указаны в DIM	O

В таблице 7.4 класс VMO определяет группы атрибутов или расширения для унаследованных групп атрибутов.

Таблица 7.4 — Группы атрибутов класса VMO

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_ STATIC	из VMO: Type, Handle
Динамическая контекстная группа VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_ DYN	из VMO: Label-String
Группа атрибутов отношения	MDC_ATTR_GRP_ RELATION	из VMO: Ext-Obj-Relations

Необходимо отметить, что группа атрибутов отношения не показана снова в определениях производных классов.

7.3.1.2 Поведение

Объект VMO не определяет никакие конкретные методы.

7.3.1.3 Уведомления

Объект VMO не формирует никакие специальные уведомления.

7.3.2 Объект VMD (Виртуальный медицинский прибор)

Объект: VMD

Описание: Класс VMD является абстракцией подсистемы медицинского назначения (например, программно-аппаратной или даже чисто программной) в составе медицинского прибора.

Производное от: VMO

Связывание имен: Handle (Описатель) (унаследованный объект VMO)

Зарегистрировано как: MDC_MOC_VMO_VMD

7.3.2.1 Атрибуты

Класс объекта VMD определяет атрибуты, представленные в таблице 7.5.

Таблица 7.5 — Атрибуты класса объекта VMD

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
VMD-Status	MDC_ATTR_VMD_ STAT	VMDStatus	***Пример — вкл.***	M
VMD-Model	MDC_ATTR_ID_ MODEL	SystemModel	Производитель и номер модели	O
Instance-Number	MDC_ATTR_ID_ INSTNO	InstNumber	Если существуют несколько экземпляров одного и того же VMD, данный атрибут помогает упорядочить последовательность	O
Production-Specification	MDC_ATTR_ID_ PROD_SPECN	ProductionSpec	Серийные номера и изменения; имеются только, если прибор VMD представляет собой отдельную подсистему	O
Compatibility-Id	MDC_ATTR_ID_ COMPAT	INT-U32	Статичен для использования производителем	O
Parame-ter-Group	MDC_ATTR_ID_ PARAM_GRP	OID-Type	***Пример — кардиоваскулярный***	O
Position	MDC_ATTR_ID_ POSN	INT-U16	***Пример — Номер слота 0xffff обозначает недействительное или неизвестное положение***	O

Окончание таблицы 7.5

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Operating-Hours	MDC_ATTR_TIME_PD_OP_HRS	INT-U32		O
Operation-Cycles	MDC_ATTR_CYC_OP	INT-U32	***Пример — Число проведенных измерений***	O
Measurement-Principle	MDC_ATTR_MSMT_PRINCIPLE	MsmtPrinciple	Описывает физический принцип измерения	O
Locale	MDC_ATTR_LOCALE	Locale	Определяет кодировку символов и язык атрибутов печатаемой строки в настоящем VMD и вложенных объектах	O

Примечание — Идентификация и изменение атрибутов не требуются, если VMD не представляет компонент аппаратного обеспечения.

В таблице 7.6 класс объекта VMD описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.6 — Группы атрибутов класса объекта VMD

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из VMD: Parameter-Group, Instance-Number, Compatibility-Id, Measurement-Principle, Locale
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из VMD: Vmd-Status
Группа атрибутов применения VMD	MDC_ATTR_GRP_VMD_APPL	из VMD: Position, Operating-Hours, Operation-Cycles
Группа атрибутов производства VMD	MDC_ATTR_GRP_VMD_PROD	из VMD: Vmd-Model, Production-Specification

Примечание — Отдельная группа атрибутов определена для статических атрибутов VMD, которые необходимы только в определенных случаях.

Применяются следующие определения типов:

```
--
-- Биты индикации состояния VMD; все биты 0 указывают, что VMD
-- работает
--
VMDStatus ::= BITS-16 {
    vmd-off (0),
    vmd-not-ready (1),          -- например, для инфузионной помпы, которая еще не готова к работе
    vmd-standby (2),            -- например, для питаемого прибора, но не работающего
    vmd-transduc-discon (8),    -- датчик отключен
    vmd-hw-discon (9)           -- измерительные приборы отключены
}
```

```
--
-- Физический принцип измерения (может быть установлено множество битов)
--
MsmtPrinciple ::= BITS-16 {
    msp-other (0),
    msp-chemical (1),
    msp-electrical (2),
    msp-impedance (3),
    msp-nuclear (4),
    msp-optical (5),
    msp-thermal (6),
    msp-biological (7),
    msp-mechanical (8),
    msp-acoustical (9),
    msp-manual (15)
}
```

7.3.2.2 Поведение

Объект VMD не определяет никакие конкретные методы.

7.3.2.3 Уведомления

Объект VMD не формирует никакие конкретные уведомления.

7.3.3 Объект Channel (Канал)

Объект:	Channel
Описание:	Объект класса Channel используется для группировки объектов класса Metric и, следовательно, предоставляет возможность иерархической организации информации.
Производное от:	VMO
Связывание имен:	Handle (Описатель) (унаследованный объект VMO)
Зарегистрировано как:	MDC_MOC_VMO_CHAN

7.3.3.1 Атрибуты

Объект Channel определяет атрибуты, представленные в таблице 7.7.

Таблица 7.7 — Атрибуты класса объекта Channel

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Channel-Id	MDC_ATTR_CHAN_ID	OID-Type	Динамическая идентификация	O
Channel-Status	MDC_ATTR_CHAN_STAT	ChannelStatus	Пример: передатчик отключен	O
Physical-Channel-No	MDC_ATTR_CHAN_NUM_PHYS	INT-U16	Дает ссылку на определенный канал аппаратного обеспечения, например, A/D (аналогово-цифровой)	O
Logical-Channel-No	MDC_ATTR_CHAN_NUM_LOGICAL	INT-U16	Динамическая нумерация каналов	O
Parameter-Group	MDC_ATTR_ID_PARAM_GRP	OID-Type	Статическая группа метрики, например, кардиоваскулярной	O
Measurement-Principle	MDC_ATTR_MSMT_PRINCIPLE	MsmtPrinciple	Описывает физический принцип измерения	O
Color	MDC_ATTR_COLOR	SimpleColour	Используется для присвоения общего цвета объектам в одном канале	O

В таблице 7.8 класс объекта Канал описывает группу атрибутов или расширения унаследованных групп атрибутов.

Таблица 7.8 — Группы атрибутов класса объекта Канал

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_ STATIC	из VMO: Type, Handle из Канала: Parameter-Group, Physical-Channel-No, Measurement-Principle
Динамическая контекстная группа VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_ DYN	из VMO: Label-String из Канала: Channel-Id, Channel-Status, Color, Logical-Channel-No

Применяются следующие определения типа:

```
--
-- Биты индикация Channel Status (Статуса канала)
--
ChannelStatus ::= BITS-16 {
    chan-off (0),
    chan-not-ready (1),
    chan-standby (2),
    chan-transduc-discon (8),
    chan-hw-discon (9)
}
```

7.3.3.2 Поведение

Объект Канал не описывает никакой особый метод.

7.3.3.3 Уведомления

Объект Канал не формирует никакие специальные уведомления.

7.3.4 Объект Metric (Метрика)

Объект: Metric

Описание: Класс Metric — базовый класс для всех объектов, отражающих непосредственные или опосредованные качественные или количественные результаты измерения биомедицинских сигналов, информацию о состоянии устройства и необходимую контекстную информацию. Он является базовым классом, который используется только для наследования.

Производное от: VMO

Связывание имен: Handle (Описатель) (унаследованный объект VMO)

Зарегистрировано как: MDC_MOC_VMO_METRIC

7.3.4.1 Атрибуты

В таблице 7.9 класс объекта Metric описывает атрибуты.

В таблице 7.10 класс объекта Metric описывает группу атрибутов или расширения унаследованных групп атрибутов.

Таблица 7.9 — Атрибуты класса объекта Metric

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Metric-Specification	MDC_ATTR_METRIC_SPECN	MetricSpec	Статичен; обязательные базовые свойства	M
Max-Delay-Time	MDC_ATTR_DELAY_TIME_MAX	RelativeTime	Статичен; максимальная задержка по отношению к реальному времени	O
Metric-Status	MDC_ATTR_METRIC_STAT	MetricStatus		O
Measurement-Status	MDC_ATTR_MSMT_STAT	Measurement-Status	Обычно это часть наблюдаемого значения	O
Metric-Id-Partition	MDC_ATTR_METRIC_ID_PART	NomPartition	Идентифицирует раздел номенклатуры, связанный с MetricId, если он отличается от раздела, установленного в атрибуте VMO::Type объекта	O
Metric-Id	MDC_ATTR_ID_PHYSIO	OID-Type	Содержит динамическую идентификацию (например, специальную этикетку артериального давления), в отличие от статического, базового ID в объекте Metric-Specification (Спецификация метрики). Идентификатор OID происходит из VMO::Type или раздела Metric-Id-Partition. Обычно данный атрибут является частью наблюдаемого значения, не является отдельным атрибутом	O
Metric-Id-Ext	MDC_ATTR_ID_MSMT_EXT	ExtNomenRef	Динамическая идентификация метрики в другой номенклатуре или словаре. Использование данного атрибута в значительной степени ограничивает способность к взаимодействию приложений.	O
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	Пример: мм рт.ст.; обычно это часть наблюдаемого значения.	O
Unit-Label-String	MDC_ATTR_UNIT_LABEL_STRING	OCTET STRING	Текстовое представление размера.	O
Vmo-Source-List	MDC_ATTR_VMO_LIST_SRC	VmoSourceList	Указывает на источник данной метрики в виде ссылки на другие метрики.	O
Metric-Source-List	MDC_ATTR_METRIC_LIST_SRC	MetricSource List	Указывает на источники данной метрики в виде списка идентификаторов метрик.	O
Msmt-Site-List	MDC_ATTR_SITE_LIST_MSMT MDC_ATTR_SITE_LIST_MSMT_EXT	SiteList SiteListExt	Участки измерений, указанные во внутренней или внешней номенклатуре	O
Body-Site-List	MDC_ATTR_SITE_LIST_BODY MDC_ATTR_SITE_LIST_BODY_EXT	SiteList SiteListExt	Локализации на теле, указанные во внутренней или внешней номенклатуре	O
Metric-Calibration	MDC_ATTR_METRIC_CALIB	MetricCalibration	Указывает тип и последний раз проведения калибровки	O

Окончание таблицы 7.9

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Color	MDC_ATTR_COLOR	SimpleColour	Цвет представления	O
Measure-Mode	MDC_ATTR_MODE_MSMT	PrivateOid	Информация об измерениях, указанная производителем	O
Measure-Period	MDC_ATTR_TIME_PD_MSMT	MetricMeasure	Время повторения измерения; не обязательно такое же, что и период обновления	O
Averaging-Period	MDC_ATTR_TIME_PD_AVG	MetricMeasure	Период времени, используемый для усреднения значений, например, метрика для среднего расхода последнего часа	O
Start-Time	MDC_ATTR_TIME_START	AbsoluteTime	Время, когда началась измерительная деятельность, например, когда начался процесс инфузии	O
Stop-Time	MDC_ATTR_TIME_STOP	AbsoluteTime	Время, когда деятельность по измерению была прекращена	O
Metric-Info-LabelString	MDC_ATTR_METRIC_INFO_LABEL_	STRING OCTET STRING	Текстовый атрибут. Например, для указания перемещения электрода или другой особой информации об измерениях	O
Substance	MDC_ATTR_ID_SUBSTANCE	ExtNomenRef	Вещество, к которому относится метрика; выражено в номенклатуре, которая не указана в настоящем стандарте	O
Substance-Label-String	MDC_ATTR_ID_SUBSTANCE_LABEL_STRING	OCTET STRING	Текстовой атрибут, который определяет вещество	O

Таблица 7.10 — Группы атрибутов класса объекта Metric

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-LabelString, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-LabelString
Группа измеряемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_METRIC_VAL_OBS	Из Metric: Metric-Id-Partition

Применяются следующие определения типа:

```
--
-- Атрибут Metric-Status (Статус метрики)
--
MetricStatus ::= BITS-16 {
    metric-off (0),
    metric-not-ready (1),
    metric-standby (2),
    metric-transduc-discon (8),
    metric-hw-discon (9)
}
--
-- Атрибут Metric-Specification (Спецификация метрики) описывает
-- все обязательные статические свойства объекта Metric
--
MetricSpec ::= SEQUENCE {
    update-period           RelativeTime,       -- минимальное время между изменениями
                                                -- наблюдаемого значения
    category                MetricCategory,
    access                  MetricAccess,
    structure               MetricStructure,
    relevance               MetricRelevance
}
--
-- Структура описывает, представляет ли объект единичное значение или
-- множество связанных значений (например, инвазивное артериальное давление
-- может быть смешанным, когда оно представляет собой пульсирующее давление
-- и получает систолическое, диастолическое и среднее значения)
--
MetricStructure ::= SEQUENCE {
    ms-struct               INT-U8 {
    simple (0),
    compound (1),                               -- множественные наблюдаемые значения,
                                                -- одинаковый динамический контекст
    complex (2)                                 -- множественные наблюдаемые значения,
                                                -- множество динамических контекстов
},
    ms-comp-no              INT-U8              -- максимальное количество компонентов в
                                                -- структуре/комплексе
}
--
-- Битовое поле MetricAccess дает информацию о том, как можно получить
-- доступ к значению метрики и когда будет доступно новое значение
--
-- Примечания
-- 1 — Флаг avail-intermittent должен быть установлен в случае, если
-- наблюдаемое значение не будет всегда доступно
-- 2 — Минимум один бит режима обновления (upd-) должен быть установлен
-- 3 — Минимум один бит режима доступа (acc-) должен быть установлен
-- 4 — Установить биты опции сканирования (sc-) возможно только если
-- установлен бит acc-scan
-- 5 — Если установлен бит acc-scan, минимум один бит sc-opt должен быть
-- установлен
--
MetricAccess ::= BITS-16 {
```

```
    avail-intermittent (0),          -- значение периодически доступно
    upd-periodic (1),                -- значение переодически обновляется
                                     -- (фиксированный период)
    upd-episodic (2),                -- значение обновляется эпизодически
    msmt-noncontinuous (3),          -- измерение проводится с перерывами
                                     -- (например, NBP)
    acc-evrep (4),                   -- метрика отправляет отчет о событии для
                                     -- наблюдаемого значения
    acc-get (5),                     -- метрика поддерживает сервис GET
    acc-scan (6),                    -- доступ к наблюдаемому значению метрики
                                     -- можно получить через объект Scanner
    gen-opt-sync (8),                -- наблюдаемое значение должно
                                     -- обрабатываться синхронно
    sc-opt-normal (10),              -- опция сканирования: значение можно
                                     -- просканировать с периодом обновления
    sc-opt-extensive (11),           -- опция сканирования: в период обновления
                                     -- может возникнуть множество значений
    sc-opt-long-pd-avail (12),       -- опция сканирования: значение можно
                                     -- просканировать на низкой скорости
                                     -- (бит сканирования superpositive-avg)
    sc-opt-confirm (13),             -- опция сканирования: сканер должен
                                     -- работать в подтвержденном режиме
    sc-opt-refresh (14)              -- опция сканирования: разрешено
                                     -- возобновление работы сканера
}

--
-- Данная категория метрики позволяет различать между измерениями,
-- настройками и вычислениями
--
MetricCategory ::= INT-U16 {
    mcat-unspec (0),
    auto-measurement (1),
    manual-measurement (2),
    auto-setting (3),
    manual-setting (4),
    auto-calculation (5),
    manual-calculation (6)
}
--
-- Уместность метрики обуславливает то, каким образом следует использовать
-- метрику (т.е. значение 0 означает нормальное использование)
--
MetricRelevance ::= BITS-16 {
    rv-unspec (0),                   -- уместность не указана; обычно не должна
                                     -- использоваться
    rv-internal (1),                 -- только значение для внутреннего
                                     -- использования
    rv-store-only (2),               -- подходит только для хранения
    rv-no-recording (3),             -- не подходит для записи
    rv-phys-ev-ind (4),              -- метрика представляет собой
                                     -- физиологический активатор (не значение)
    rv-btb-metric (5),               -- метрика рассчитывается для каждого удара или вздоха,
                                     -- не усредненная по времени
```

```
    rv-app-specific (8),                -- требуется специальное приложение для
                                        -- интерпретации метрики
    rv-service (9)                      -- метрика предназначена для обслуживания
                                        -- или диагностики
}
--
-- Атрибут Metric-Calibration (Калибровка метрики) определяет методы
-- и продолжительность калибровки
-- Примечание — Допускаются множественные записи
--
MetricCalibration ::= SEQUENCE OF MetricCalEntry

MetricCalEntry ::= SEQUENCE {
    cal-type                MetricCalType,
    cal-state               MetricCalState,
    cal-time                AbsoluteTime
}

MetricCalType ::= INT-U16 {
    cal-unspec (0),
    cal-offset (1),         -- калибровка смещения
    cal-gain (2),           -- калибровка усиления
    cal-two-point (3)       -- калибровка по двум точкам
}

MetricCalState ::= INT-U16 {
    not-calibrated (0),
    cal-required (1),
    calibrated (2)
}

--
-- Упорядоченный список участков измерения, например, места
-- расположения электродов ЭЭГ
--
SiteList ::= SEQUENCE OF OID-Type        -- записи из раздела номенклатуры
                                         -- локализаций на теле
--
-- Список участков может также ссылаться на внешние номенклатуры
-- для уточнения участков измерений
--
SiteListExt ::= SEQUENCE OF ExtNomenRef
--
-- Атрибут Metric-Source-List (Список источников метрики) — это
-- упорядоченный список метрических идентификаторов OID
--
MetricSourceList ::= SEQUENCE OF OID-Type   -- идентификаторы OID из раздела
                                            -- VMO::Type или Metric-Id-
                                            -- Partition

--
-- Атрибут Vmo-Source-List (Список источников VMO) определяет ссылки на
-- другие объекты, производные объекта VMO, которые используются в качестве
-- источников данной метрики (это упорядоченный список)
--
```

```
VmoSourceList ::= SEQUENCE OF VmoSourceEntry

VmoSourceEntry ::= SEQUENCE {
    vmo-type            OID-Type,       -- из раздела номенклатуры,
                                        -- объекто-ориентированной
    glb-handle          GLB-HANDLE
}

--
-- Атрибут Measurement-Status (Статус измерения) определяет состояние
-- измерения; используется производными классами
--
MeasurementStatus ::= BITS-16 {
    invalid (0),
    questionable (1),
    not-available (2),
    calibration-ongoing (3),
    test-data (4),
    demo-data (5),
    validated-data (8),             -- соответствующий, например, в архиве
    early-indication (9),           -- начальная предположительная оценка
                                    -- значения
    msmt-ongoing (10),              -- указывает на то, что только что провели
                                    -- новое измерение(эпизодический)
    msmt-state-in-alarm (14),       -- указывает на то, что метрика имеет
                                    -- условие активных аварийных сигналов
    msmt-state-al-inhibited (15)    -- метрика поддерживает сигнализирование
                                    -- и аварийные сигналы отключены
                                    -- (дополнительная опция)
}

--
-- В ряде производных метрик необходимо указать диапазоны
-- Требующийся для этого тип определен здесь, в базовом классе
--
AbsoluteRange ::= SEQUENCE {
    lower-value                     FLOAT-Type,
    upper-value                     FLOAT-Type
}

--
-- Метрическое средство применяется для атрибутов, которые имеют значение и
-- размер
--
MetricMeasure ::= SEQUENCE {
    value               FLOAT-Type,
    unit-code           OID-Type    -- из раздела номенклатуры размеров
}
```

7.3.4.2 Поведение

Объект Metric не описывает никакие конкретные методы.

7.3.4.3 Уведомления

Объект Metric не формирует никакие конкретные уведомления.

7.3.5 Объект Numeric (Числовой показатель)

Объект: Numeric

Описание: Класс Numeric отражает те значения биомедицинских измерений и ту информацию о состоянии прибора, которые представлены в числовом формате, как, например, измерения амплитуды колебаний или показания счетчиков.

Производное от: Metrics (Метрика)

Связывание имен: Handle (Описатель) (унаследованный объект VMO)

Зарегистрирован как: MDC_MOC_VMO_METRIC_NU

7.3.5.1 Атрибуты

Объект Numeric определяет атрибуты, представленные в таблице 7.11.

Таблица 7.11 — Атрибуты класса объекта Numeric

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS	NuObsValue	***Пример — Значение измерения должно также содержать в себе информацию о его подтверждении соответствия***	C[a)]
Compound-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS	NuObsValueCmp	Используется, когда множество значений представлены в одном объекте Numeric. (Структура — составная)	C
Absolute-Time-Stamp	MDC_ATTR_TIME_STAMP_ABS	AbsoluteTime	Время наблюдения (временная метка)	O
Relative-Time-Stamp	MDC_ATTR_TIME_STAMP_REL	RelativeTime		O
HiRes-Time-Stamp	MDC_ATTR_TIME_STAMP_REL_HI_RES	HighResRelativeTime	Временная метка высокого разрешения	O
Nu-Measure-Range	MDC_ATTR_NU_RANGE_MSMT	AbsoluteRange	Потенциальный диапазон измерений	O
Nu-Physiological-Range	MDC_ATTR_NU_RANGE_PHYSIO	AbsoluteRange	Физиологически обусловленный диапазон (но это не диапазон аварийной сигнализации)	O
Nu-Measure-Resolution	MDC_ATTR_NU_MSMT_RES	FLOAT-Type	Разрешающая способность измерения (минимальная разница между двумя наблюдаемыми значениями)	O
Display-Resolution	MDC_ATTR_DISP_RES	DispResolution	Используется, когда требуется другое разрешение при отображении значения	O
Accuracy	MDC_ATTR_NU_ACCUR_MSMT	FLOAT-Type	Максимальное отклонение действительного значения от указанного наблюдаемого значения (если его можно определить)	O

[a)] Минимум один тип наблюдаемого значения должен присутствовать, как определено атрибутом Metric-Specification.

В таблице 7.12 класс объекта Numeric описывает группу атрибутов или расширения унаследованных групп атрибутов.

Таблица 7.12 — Группы атрибутов класса объекта Numeric

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_ STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time из Numeric:
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_ DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-LabelString, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-Label-String из Numeric: Nu-Measure-Range, Nu-Physiological-Range, Accuracy, Nu-Measure-Resolution, Display-Resolution
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_ METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Numeric: Nu-Observed-Value, Compound-Nu-Observed-Value, Absolute-Time-Stamp, Relative-Time-Stamp, HiRes-Time-Stamp

Применяют следующие определения типа:

```
--
-- Атрибут Nu-Observed-Value всегда включает в себя идентификацию,
-- состояние и размеры для обеспечения последовательности при минимальных
-- издержках
--
NuObsValue ::= SEQUENCE {
    metric-id       OID-Type                -- из раздела VMO::Type или
                                            -- Metric-Id-Partition
    state           MeasurementStatus,
                                            -- определен в базовом классе Metric
    unit-code       OID-Type,               -- из раздела номенклатуры размеров
    value           FLOAT-Type
}

--
-- Наблюдаемое значение для составных чисел
--
NuObsValueCmp ::= SEQUENCE OF NuObsValue

--
-- Атрибут Display-Resolution (Разрешение дисплея) является
```

```
-- представлением значения на дисплее (может иметь низкое разрешение)
--
DispResolution ::= SEQUENCE {
    pre-point           INT-U8,         -- количество знаков до запятой
    post-point          INT-U8          -- количество знаков после запятой
```

7.3.5.2 Поведение

Объект Numeric не описывает никакой конкретный метод.

7.3.5.3 Уведомления

Объект Numeric не формирует никакие конкретные уведомления.

7.3.6 Объект Sample Array (Массив выборок)

Объект: Sample Array

Описание: Класс Sample Array — базовый класс для представления таких метрик, значения которых представлены некоторой кривой на графике, и поэтому результаты их наблюдаемых значений, поступающих из коммуникационных систем, представлены массивами данных об отдельных точках графика.

Производное от: Metric (Метрика)

Связывание имен: Handle (Описатель) (унаследованный объект VMO)

Зарегистрирован как: MDC_MOC_VMO_METRIC_SA

7.3.6.1 Атрибуты

Класс Объекта Sample Array определяет атрибуты, представленные в таблице 7.13.

Таблица 7.13 — Атрибуты класса Объекта Sample Array

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Observed-Value	MDC_ATTR_SA_VAL_OBS	SaObsValue	Пример: массив значений измерения	C[a)]
Compound-Sa-Observed-Value	MDC_ATTR_SA_CMPD_VAL_OBS	SaObsValueCmp		C
Sa-Specification	MDC_ATTR_SA_SPECN	SaSpec	Статическое описание массива выборок и типов их значений	M
Compression	MDC_ATTR_COMPRES	PrivateOid	Описывает потенциальный алгоритм сжатия	O
Scale-and-Range-Specification	MDC_ATTR_SCALE_SPECN_I8 MDC_ATTR_SCALE_SPECN_I16 MDC_ATTR_SCALE_SPECN_I32	ScaleRangeSpec8 ScaleRangeSpec16 ScaleRangeSpec32	Определяет отображение данных между выборками и действительными значениями, а также диапазон измерений; тип зависит от размера выборки	M
Sa-Physiological-Range	MDC_ATTR_SA_RANGE_PHYS_I8 MDC_ATTR_SA_RANGE_PHYS_I16 MDC_ATTR_SA_RANGE_PHYS_I32	ScaledRange8 ScaledRange16 ScaledRange32	Для оптимального масштабирования дисплея указывается физиологически значимый диапазон	O

Окончание таблицы 7.13

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Visual-Grid	MDC_ATTR_GRID_VIS_I8 MDC_ATTR_GRID_VIS_I16 MDC_ATTR_GRID_VIS_I32	SaVisualGrid8 SaVisualGrid16 SaVisualGrid32	Определяет положения линии сетки на дисплеях и записывающих устройствах; тип зависит от размера выборки	O
Data	MDC_ATTR_SA_CALIB_I8 MDC_ATTR_SA_CALIB_I16 MDC_ATTR_SA_CALIB_I32	SaCalData8 SaCalData16 SaCalData32	Определяет положения калибровочных меток на дисплее и записывающих устройствах; тип зависит от размера выборки	O
Filter-Specification	MDC_ATTR_FILTER_SPECN	SaFilterSpec		O
Filter-Label-String	MDC_ATTR_FILTER_LABEL_STRING	OCTET STRING	Текстовая этикетка активного фильтра, например, Butterworth или Linear-Phase	O
Sa-Signal-Frequency	MDC_ATTR_SA_FREQ_SIG	SaSignal-Frequency	Максимальная частота сигнала	O
Sa-Measure-Resolution	MDC_ATTR_SA_MSMT_RES	FLOAT-Type		O
Sa-Marker-List	MDC_ATTR_SA_MARKER_LIST_I8 MDC_ATTR_SA_MARKER_LIST_I16 MDC_ATTR_SA_MARKER_LIST_I32	MarkerListSaVal8 MarkerListSa-Val16 MarkerListSa-Val32		
[a] Минимум один тип наблюдаемого значения должен присутствовать, как определено атрибутом Metric-Specification.				

В таблице 7.14 класс объекта Sample Array описывает группу атрибутов или расширения унаследованных групп атрибутов.

Таблица 7.14 — Группы атрибутов класса объекта Sample Array

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time из Sample Array: Sa-Specification, Compression, Sa-Marker-List

Окончание таблицы 7.14

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-LabelString, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-LabelString из Sample Array: Scale-and-Range-Specification, Sa-Physiological-Range, Visual-Grid, Sa-Calibration-Data, Filter-Specification, Filter-Label-String, Sa-Signal-Frequency, Sa-Measure-Resolution
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_ METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Sample Array: Sa-Observed-Value, Compound-Sa-Observed-Value

Применяют следующие определения типа:

```
--
-- Атрибут Sa-Observed-Value (Наблюдаемое значение SA)
--
SaObsValue ::= SEQUENCE {
    metric-id                   OID-Type,               -- из раздела VMO::Type или
                                                        -- Metric-Id-Partition
    state                       MeasurementStatus,
                                                        -- указан в объекте Metric
    array                       OCTET STRING
}
--
-- Атрибут Compound-Sa-Observed-Value — это составное наблюдаемое значение
--
SaObsValueCmp ::= SEQUENCE OF SaObsValue

--
-- Атрибут Sa-Specification
--
SaSpec ::= SEQUENCE {
    array-size              INT-U16,                -- количество выборок за один
                                                    -- период обновления метрики
    sample-type             SampleType,
    flags                   SaFlags
}

--
-- Тип выборки описывает одну выборку в массиве наблюдаемого значения
--
SampleType ::= SEQUENCE {
```

```
    array-size              INT-U8,         -- например, 8 для 8-битных выборок, 16
                                            -- для 16-битных выборок должны
                                            -- делиться на 8
    significant-bits        INT-U8          -- определяет значимые биты в одной
                                            -- выборке
      { signed-samples(255)}                -- если значение 255, то все выборки
                                            -- имеют знак; все биты значимы;
                                            -- выборки интерпретируются в
                                            -- дополнительном двоичном коде
}

--
-- Тип данных SaFlags описывает дополнительные свойства осциллограммы
--
SaFlags ::= BITS-16 {
    smooth-curve(0),                        -- для оптимального отображения используйте
                                            -- алгоритм сглаживания
    delayed-curve(1),                       -- задержка отображения кривой (не в
                                            -- реальном времени)
    static-scale(2),                        -- ScaleRangeSpec не меняется
    sa-ext-val-range(3)                     -- незначимые биты в выборке не являются 0,
                                            -- например, когда они используются для
                                            -- комментариев или меток; получатель
                                            -- должен применить битовую маску для
                                            -- извлечения значимых битов из выборки
}

--
-- Спецификация применяемого фильтра сигналов
--
SaFilterSpec ::= SEQUENCE OF SaFilterEntry

SaFilterEntry ::= SEQUENCE {
    filter-type             INT-U16 {other(0), low-pass(1), high-pass(2), notch(3) },
    filter-frequency        FLOAT-Type,
    filter-order                                    -- например, -1:6 дБ/октет
}

--
-- Атрибут Scale-and-Range-Specification описывает отношение между
-- масштабированными и абсолютными значениями; в зависимости от размера
-- выборки имеется множество типов атрибутов
--
-- Примечание — Если колебание не представляет собой абсолютные значения,
-- поля абсолютного значения должны содержать в себе конкретное значение;
-- если атрибут Sa-Specification указывает на выборки со знаком,
-- масштабированные значения должны интерпретироваться как значения со
-- знаком
--
ScaleRangeSpec8 ::= SEQUENCE {
    lower-absolute-value            FLOAT-Type,
    upper-absolute-value            FLOAT-Type,
    lower-scaled-value              INT-U8,
    upper-scaled-value              INT-U8
}
```

```
ScaleRangeSpec16 ::= SEQUENCE {
    lower-absolute-value        FLOAT-Type,
    upper-absolute-value        FLOAT-Type,
    lower-scaled-value          INT-U16,
    upper-scaled-value          INT-U16
}

ScaleRangeSpec32 ::= SEQUENCE {
    lower-absolute-value        FLOAT-Type,
    upper-absolute-value        FLOAT-Type,
    lower-scaled-value          INT-U32,
    upper-scaled-value          INT-U32
}

--
-- Атрибут Visual-Grid (Визуальная решетка) определяет линии координатной сетки
-- на различных уровнях; если Атрибут Sa-Specification указывает на выборки со знаками,
-- масштабированные значения должны интерпретироваться как значения со знаками
--
SaVisualGrid8 ::= SEQUENCE OF SaGridEntry8

SaGridEntry8 ::= SEQUENCE {
    absolute-value              FLOAT-Type,
    scaled-value                INT-U8,
    level                       INT-U8
}

SaVisualGrid16 ::= SEQUENCE OF SaGridEntry16

SaGridEntry16 ::= SEQUENCE {
    absolute-value              FLOAT-Type,
    scaled-value                INT-U16,
    level                       INT-U16
}

SaVisualGrid32 ::= SEQUENCE OF SaGridEntry32

SaGridEntry32 ::= SEQUENCE {
    absolute-value              FLOAT-Type,
    scaled-value                INT-U32,
    level                       INT-U16
}

--
-- Атрибут Sa-Calibration-Data задает калибровочные метки на дисплее
-- или на ленте для самопишущего прибора; если Атрибут Sa-Specification
-- указывает на выборки со знаками, измеренные значения должны
-- интерпретироваться как значения со знаками
--
SaCalData8 ::= SEQUENCE {
    lower-absolute-value        FLOAT-Type,
    upper-absolute-value        FLOAT-Type,
    lower-scaled-value          INT-U8,
    upper-scaled-value          INT-U8,
    increment                   INT-U16,        -- масштабированное значение для
                                                -- каждой цены деления
    cal-type                    SaCalDataType
```

```
}

SaCalData16 ::= SEQUENCE {
    lower-absolute-value        FLOAT-Type,
    upper-absolute-value        FLOAT-Type,
    lower-scaled-value          INT-U16,
    upper-scaled-value          INT-U16,
    increment                   INT-U16,        -- масштабированное значение для
                                                -- каждой цены деления
    cal-type                    SaCalDataType
    }
SaCalData32 ::= SEQUENCE {
    lower-absolute-value        FLOAT-Type,
    upper-absolute-value        FLOAT-Type,
    lower-scaled-value          INT-U32,
    upper-scaled-value          INT-U32,
    increment                   INT-U32,        -- масштабированное значение для
                                                -- каждой цены деления
    cal-type                    SaCalDataType
}

SaCalDataType ::= INT-U16 {
    bar(0),                                     -- отображение калибровочной шкалы
    stair(1)                                    -- отображение калибровочной каждой цены
                                                -- деления
}

--
-- Атрибут Sa-Signal-Frequency описывает частоту сигнала
--
SaSignalFrequency ::= {
    low-edge-freq               FLOAT-Type,
    high-edge-freq              FLOAT-Type      -- оба в герцах
}

--
-- Типы данных атрибута Sa-Physiological-Range
-- Если Атрибут Sa-Specification указывает выборки со знаками, то
-- масштабированные значения должны интерпретироваться как значения со
-- знаками

ScaledRange8 ::= SEQUENCE {
    lower-scaled-value          INT-U8,
    upper-scaled-value          INT-U8
}

ScaledRange16 ::= SEQUENCE {
    lower-scaled-value          INT-U16,
    upper-scaled-value          INT-U16
}

    ScaledRange32 ::= SEQUENCE {
    lower-scaled-value          INT-U32,
    upper-scaled-value          INT-U32
}
--
```

```
-- Атрибут Sa-Marker-List позволяет определить конкретные значения выборок
-- для маркировки или комментирования определенных условий непосредственно
-- в значении выборки; конкретное значение выборки может выступать в
-- качестве истинного значения или битовой маски, в зависимости от ID
-- метки; в любом случае значение выборки может использовать биты,
-- выходящие за пределы нормального диапазона (как указано в SampleType::
-- significant-bits), только если установлена метка SaFlags::sa-ext-val-
-- range
MarkerListSaVal8 ::= SEQUENCE OF MarkerEntrySaVal8

MarkerEntrySaVal8 ::= SEQUENCE {
    marker-id           OID-Type,       -- из раздела VMO::Type или
                                        -- Metric-Id-Partition
    marker-val          INT-U8,         -- значение или битовая маска, в
                                        -- зависимости от ID маркера
    unused              INT-U8          -- для выравнивания
}

MarkerListSaVal16 ::= SEQUENCE OF MarkerEntrySaVal16

MarkerEntrySaVal16 ::= SEQUENCE {
    marker-id           OID-Type,       -- из раздела VMO::Type или
                                        -- Metric-Id-Partition
    marker-val          INT-U16,        -- значение или битовая маска, в
                                        -- зависимости от ID маркера
}

MarkerListSaVal32 ::= SEQUENCE OF MarkerEntrySaVal32

MarkerEntrySaVal32 ::= SEQUENCE {
    marker-id           OID-Type,       -- из раздела VMO::Type или
                                        -- Metric-Id-Partition
    marker-val          INT-U32,        -- значение или битовая маска, в
                                        -- зависимости от ID маркера
}
```

7.3.6.2 Поведение

Объект Sample Array не описывает никакие конкретные методы.

7.3.6.3 Уведомления

Объект Sample Array не формирует никакие конкретные уведомления.

7.3.7 Объект Real Time Sample Array (Массив выборок в режиме реального времени)

Объект:	Real Time Sample Array
Описание:	Объект класса Real Time Sample Array — такая выборка значений, которая отражает график (осцилограмму), изменяющийся в режиме реального времени.
Производное от:	Sample Array (Массив выборок)
Связывание имен:	Handle (Описатель) (унаследованный объект VMO)
Зарегистрирован как:	MDC_MOC_VMO_METRIC_SA_RT

7.3.7.1 Атрибуты

Класс объект Real Time Sample Array определяет атрибуты, представленные в таблице 7.15.

Таблица 7.15 — Атрибуты класса объекта Real Time Sample Array

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Sample-Period	MDC_ATTR_TIME_PD_SAMP	RelativeTime	***Пример — В (долях) миллисекунд***	M
Sweep-Speed	MDC_ATTR_SPD_SWEEP_DEFAULT	MetricMeasure	***Пример — Миллиметров в секунду***	O
Average-Reporting-Delay	MDC_ATTR_REPORTING_DELAY_AVG	RelativeTime	Указывает среднее время между тем, когда первый элемент в обновлении массива (array update) попал в выборку и когда был сформирован отчет о событии FastPeriCfg-Scanner (т. е. метка времени отчета о событии)	O
Sample-Time-Sync	MDC_ATTR_SAMPLE_TIME_SYNC	RelativeTime	Указывает точное время выборки первого элемента в обновлении массива. Является дополнительным, если присутствует атрибут Average-Reporting-Delay; иначе не входит в область применения настоящего стандарта	C
HiRes-Sample-Time-Sync	MDC_ATTR_SAMPLE_TIME_SYNC_HIRES	HighRes-RelativeTime	Указывает точное время выборки первого элемента в обновлении массива. Является дополнительным, если присутствует атрибут Average-Reporting-Delay; иначе не входит в область применения настоящего стандарта	C

Примечание — Вместе с атрибутом Average-Reporting-Delay (Средняя задержка создания отчетов) можно использовать Атрибут Sample-Time-Sync (Синхронизация времени выборки) или HiRes-Sample-Time-Sync (Синхронизация времени выборки высокого разрешения), чтобы точно указать определенное время выборки. Отчеты об атрибутах Sample-Time-Sync и HiRes-Sample-Time-Sync должны поступать от сканнера эпизодически:

- при первом запуске процесса создания отчетов;

- время от времени после запуска этого процесса с частотой, которая обеспечивает, чтобы дрейф времени / расфазировка тактовых сигналов негативно не сказалась на кореляцию точного времени с выборкой для отдельной осцилограммы.

См. также 6.7.5 для получения информации по определению класса объекта FastPeriCfgScanner.

В таблице 7.16 класс объекта Real Time Sample Array описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.16 — Группы атрибутов класса объекта Real Time Sample Array

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time из Real Time Sample Array: Sample-Period, Sweep-Speed, Average-Reporting-Delay

Окончание таблицы 7.16

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-LabelString, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-Label-String из Sample Array: Scale-and-Range-Specification, Sa-Physiological-Range, Visual-Grid, Sa-Calibration-Data, Filter-Specification, Filter-Label-String, Sa-Signal-Frequency, Sa-Measure-Resolution из Real Time Sample Array: Sample-Time-Sync, HiRes-Sample-Time-Sync
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_ METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Sample Array: Sa-Observed-Value, Compound-Sa-Observed-Value

Не применимы никакие дополнительные определения типа.

7.3.7.2 Поведение

Объект Массив выборок реального времени не описывает никакие конкретные методы.

7.3.7.3 Уведомления

Объект Массив выборок реального времени не формирует никакие конкретные уведомления.

7.3.8 Объект Time Sample Array (Временный массив выборок)

Объект:	Time Sample Array
Описание:	Объект класса Time Sample Array — такая выборка значений, которая отражает последовательность отдельных фрагментов графика (осциллограммы), ограниченных некоторым интервалом времени.
Производное от:	Sample Array (Массив выборок)
Связывание имен:	Handle (Описатель) (унаследованный объект VMO)
Зарегистрирован как:	MDC_MOC_VMO_METRIC_SA_T

7.3.8.1 Атрибуты

Класс объекта Time Sample Array определяет атрибуты, представленные в таблице 7.17.

Таблица 7.17 — Атрибуты класса объекта Time Sample Array

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Absolute-Time-Stamp	MDC_ATTR_TIME_ STAMP_ABS	AbsoluteTime	Время наблюдения (временная метка)	O
Relative-Time-Stamp	MDC_ATTR_TIME_ STAMP_REL	RelativeTime		O

Окончание таблицы 7.17

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
HiRes-Time-Stamp	MDC_ATTR_TIME_STAMP_REL_HI_RES	HighRes-RelativeTime	Временная метка высокого разрешения	O
Sample-Period	MDC_ATTR_TIME_PD_SAMP	RelativeTime	***Пример — В (долях) миллисекунд***	M
Sweep-Speed	MDC_ATTR_SPD_SWEEP_DEFAULT	MetricMeasure	***Пример — Миллиметров в секунду***	O
Tsa-Marker-List	MDC_ATTR_TSA_MARKER_LIST	MarkerListRelTim	Отмечает положения отдельных фрагментов графика	O

В таблице 7.18 класс объекта Time Sample Array описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.18 — Группы атрибутов класса объекта Массив выборок времени

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time из Sample Array: Sa-Specification, Compression, Sa-Marker-List из Time Sample Array: Sample-Period, Sweep-Speed
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-Label-String, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-LabelString_ из Sample Array: Scale-and-Range-Specification, Sa-Physiological-Range, Visual-Grid, Sa-Calibration-Data, Filter-Specification, Filter-Label-String, Sa-Signal-Frequency, Sa-Measure-Resolution
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Sample Array: Sa-Observed-Value, Compound-Sa-Observed-Value из Time Sample Array: Absolute-Time-Stamp, Relative-Time-Stamp, HiRes-Time-Stamp, Tsa-Marker-List

Применяют следующие определения типов:

```
-- Атрибут Tsa-Marker-List можно также использовать, чтобы отмечать
-- определенные временные точки на фрагменте осциллограммы; первая выборка
-- осуществляется в относительный момент времени, равный 0
--
MarkerListRelTim ::= SEQUENCE OF MarkerEntryRelTim

MarkerEntryRelTim ::= SEQUENCE {
    marker-id           OID-Type,           -- из раздела VMO::Type или
                                            -- Metric-Id-Partition
    marker-time         RelativeTime
}
```

7.3.8.2 Поведение

Объект Массив выборок времени не описывает никакие конкретные методы.

7.3.8.3 Уведомления

Объект Массив выборок времени не формирует никакие конкретные уведомления.

7.3.9 Объект Distribution Sample Array (Распределенный массив выборок)

Объект: Distribution Sample Array

Описание: Объект класса Distribution Sample Array — выборка значений, представляющая линейные распределения значений в форме массивов, содержащих масштабированную выборку. Индекс значения в пределах множества, определяемого одним наблюдением, показывает положение значения в пространстве, но не во времени. Поэтому массив измеренных значений можно рассматривать как систему координат *x-y*, где ось *y* обозначается атрибутами, унаследованными от объекта Metric, а ось *x* — атрибутами, определенными в объекте Distribution Sample Array.

Производное от: Sample Array (Массив выборок)

Связывание имен: Handle (Описатель) (унаследованный объект VMO)

Зарегистрирован как: MDC_MOC_VMO_METRIC_SA_D

7.3.9.1 Атрибуты

Класс объекта Distribution Sample Array определяет атрибуты, представленные в таблице 7.19.

Таблица 7.19 — Атрибуты класса объекта Distribution Sample Array

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Absolute-Time-Stamp	MDC_ATTR_TIME_STAMP_ABS	AbsoluteTime	Время наблюдения (временная метка)	O
Relative-Time-Stamp	MDC_ATTR_TIME_STAMP_REL	RelativeTime		O
HiRes-Time-Stamp	MDC_ATTR_TIME_STAMP_REL_HI_RES	HighRes-RelativeTime	Временная метка высокого разрешения	O
Distribution-Range-Specification	MDC_ATTR_RANGE_DISTRIB	DsaRangeSpec	Преобразование индекса массива в абсолютное значение	M
x-Unit-Code	MDC_ATTR_UNIT_CODE_X	OID-Type	Применим к оси *x*	O

Класс объекта Distribution Sample Array описывает в таблице 7.20 группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.20 — Группы атрибутов класса объекта Distribution Sample Array

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time из Sample Array: Sa-Specification, Compression, Sa-Marker-List
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-Label-String, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-LabelString из Sample Array: Scale-and-Range-Specification, Sa-Physiological-Range, Visual-Grid, Sa-Calibration-Data, Filter-Specification, Filter-Label-String, Sa-Signal-Frequency, Sa-Measure-Resolution из Distribution Sample Array : Distribution-Range-Specification, x-Unit-Code, x-Unit-Label-String
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_ METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Sample Array: Sa-Observed-Value, Compound-Sa-Observed-Value из Distribution Sample Array : Absolute-Time-Stamp, Relative-Time-Stamp, HiRes-Time-Stamp, Dsa-Marker-List

Применяют следующие определения типов:

```
--
-- Атрибут Distribution-Range-Specification(Спецификация диапазона
-- распределения) определяет абсолютное значение для первого и последнего
-- элемента массива; здесь предполагается линейная шкала, если не
-- определена конкретная схема сжатия (последнее значение — первое
-- значение)/кол-во элементов массива = ширина шага
--
DsaRangeSpec ::= SEQUENCE {
    first-element-value             FLOAT-Type,
    last-element-value              FLOAT-Type
}
--
-- Атрибут DSA-Marker-List (Список маркеров DSA) позволяет комментировать
-- выборки путем указания ссылки на выборку с индексом
--
MarkerListIndex ::= SEQUENCE OF MarkerEntryIndex
```

```
MarkerEntryIndex ::= SEQUENCE {
    marker-id               OID-Type,          -- из раздела VMO::Type или Metric
                                               -- Id-Partition
    marker-index            INT-U16
}
```

7.3.9.2 Поведение

Объект Distribution Sample Array не описывает никакие конкретные методы.

7.3.9.3 Уведомления

Объект Distribution Sample Array не формирует никакие конкретные уведомления.

7.3.10 Объект Enumeration (Перечисление)

Объект:	Enumeration
Описание:	Класс Enumeration отражает информацию о текущем состоянии и/или хронологическую информацию. Результаты отслеживания могут быть представлены в форме нормативных кодов (которые включены в номенклатуру, определенную в настоящем стандарте или в какой-либо другой номенклатуре), или в форме произвольного текста.
Производное от:	Metric (Метрика)
Связывание имен:	Handle (Описатель) (унаследованный объект VMO)
Зарегистрирован как:	MDC_MOC_VMO_METRIC_ENUM

7.3.10.1 Атрибуты

Класс объекта Enumeration определяет атрибуты, представленные в таблице 7.21.

Таблица 7.21 — Атрибуты класса объекта Enumeration

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Value	MDC_ATTR_VAL_ENUM_OBS	EnumObsValue	Либо Enum-Observed-Value либо Compound-Enum-Observed-Value должны поддерживаться в одном экземпляре объекта	C
Compound-Enum-Observed-Value	MDC_ATTR_VAL_ENUM_OBS_CMPD	EnumObsValueCmp	Либо Enum-Observed-Value либо Compound-Enum-Observed-Value должны поддерживаться в одном экземпляре объекта	C
Absolute-Time-Stamp	MDC_ATTR_TIME_STAMP_ABS	AbsoluteTime		O
Relative-Time-Stamp	MDC_ATTR_TIME_STAMP_REL	RelativeTime		O
HiRes-Time-Stamp	MDC_ATTR_TIME_STAMP_REL_HI_RES	HighRes-RelativeTime	Временная метка высокого разрешения	C
Enum-Measure-Range	MDC_ATTR_ENUM_RANGE_MSMT	EnumMsmtRange	Список поддерживаемых идентификаторов OID наблюдаемых значений. Является необязательным (дополнительным), если в наблюдаемом значении используется тип OID (EnumVal::enumobj-id); иначе не входит в область применения настоящего стандарта	C

Окончание таблицы 7.21

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Enum-Measure-Range-Bit-String	MDC_ATTR_ENUM_RANGE_MSMT_BIT_STRING	BITS-32	Список поддерживаемых битов наблюдаемого значения в типе данных Битовая строка. Дополнительно если в наблюдаемом значении используется тип Битовая строка (EnumVal::enum-bit-str); иначе не входит в область применения настоящего стандарта	C
Enum-Measure-Range-Labels	MDC_ATTR_ENUM_RANGE_MSMT_LABELS	EnumMsmtRange	Ассоциирует текстовые строки с определенными значениями перечисления	O

В таблице 7.22 класс объекта Enumeration описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.22 — Группы атрибутов класса объекта Enumeration

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time, Enum-Measure-Range-Labels
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-Label-String, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-LabelString, Substance, Substance-LabelString из Enumeration: Enum-Measure-Range, Enum-Measure-Range-Bits
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Sample Array: Sa-Observed-Value, Compound-Sa-Observed-Value из Enumeration: Enum-Observed-Value, Compound-Enum-Observed-Value, Absolute-Time-Stamp, Relative-Time-Stamp, HiRes-Time-Stamp

Применяют следующие определения типов:

```
--
-- Атрибут Enum-Observed-Value
--
EnumObsValue ::= SEQUENCE {
```

```
    metric-id           OID-Type,               -- из раздела VMO::Type или
                                                -- Metric-Id-Partition
    state               MeasurementStatus,
    value               EnumVal                 -- поддерживает различные типы
                                                -- данных значений
}
-- Тип данных значения перечисления используется для обеспечения различных
-- конкретных типов данных наблюдения, как указано ниже (Следует отметить,
-- что тип измерения закодирован в структуре главного уровня
-- EnumObsVal::metric-id)
--
--      enum-obj-id:                    используется для сообщения OID метрики, например,
--                                      в виде комментирования или другого события,
--                                      определенного в разделе VMO::Type или Metric-Id-
--                                      Partition;
--      enum-text-string:               используется для сообщения строки произвольного
--                                      текста (например, сообщение о статусе);
--      enum-external-code:             используется для предоставления кода внешней
--                                      номенклатуры (например, можно использовать для
--                                      кодов процедур, которые не указаны в стандартной
--                                      номенклатуре;
--      enum-bit-str:                   для кодирования значений строки битов; тип данных
--                                      строка битов должен быть задан отдельно, например,
--                                      в номенклатуре или в стандарте, ориентированном
--                                      на данный прибор;
--      enum-record-metric/oo:          позволяет идентифицировать дополнительные типы
--                                      данных по коду номенклатуры из раздела VMO::Type
--                                      или Metric-Id-Partition; присоединенный тип данных
--                                      должен быть задан отдельно, например, в стандарте,
--                                      ориентированном на данный прибор;
--      enum-numeral:                   используется для обеспечения числовых
--                                      перечисляемых значений, которые должны
--                                      быть заданы отдельно, например, в стандарте,
--                                      ориентированном на данный прибор; данный тип не
--                                      должен использоваться для численных измерений
--
EnumVal ::= CHOICE {
    enum-obj-id             [1]OID-Type,        -- из раздела VMO::Type или
                                                -- Metric-Id-Partition
    enum-text-string        [2]OCTET STRING,
                                                -- произвольный текст
    enum-external-code [8]  ExtNomenRef,
                                                -- код, определенный в другой
                                                -- системе кодирования
    enum-bit-str [16]       BITS-32,            -- строка битов
    enum-record-metric [33] EnumRecordMetric,   -- тип записи, определенный
                                                -- посредством ID из раздела
                                                -- из раздела VMO::Type или
                                                -- Metric-Id-Partition
```

```
    enum-record-oo [34]         EnumRecordOo,
                                                    -- тип записи, определенный
                                                    -- посредством ID из раздела
                                                    -- объектно-ориентированной
                                                    -- номенклатуры
    enum-numeral [64]           INT-U32             -- перечисляемые целые
                                                    -- значения
}

--
-- Структурный тип со структурой и содержанием, определенными
-- идентификатором номенклатуры из раздела VMO::Type или Metric-Id-
-- Partition
--
EnumRecordMetric ::= SEQUENCE {
record-type-code                OID-Type,       -- из раздела VMO::Type или Metric-
                                                -- Id-Partition
record-data                     ANY DEFINED BY record-type-code
}

--
-- Структурный тип со структурой и содержанием, определенными
-- идентификатором номенклатуры из объектно-ориентированного раздела
-- номенклатуры,
--
EnumRecordOo ::= SEQUENCE {
record-type-code                OID-Type,       -- должен быть из раздела объекто-
                                                -- ориентированной номенклатуры
record-data                     ANY DEFINED BY record-type-code
}

--
-- Атрибут Compound-Enum-Observed-Value это составное наблюдаемое значение
--
EnumObsValueCmp ::= SEQUENCE OF EnumObsValue

-- Атрибут Enum-Measure-Range определяет множество величин потенциальных
-- (т. е. допустимых) значений атрибута Enum-Observed-Value (разрешен только
-- когда используется тип EnumVal::enum-obj-id)
--
EnumMsmtRange ::= SEQUENCE OF
                OID-Type        -- из раздела VMO::Type или Metric- Id-Partition
--
-- Атрибут Enum-Measure-Range-Labels определяет как множество величин потенциальных (т. е. допу-
стимых) значений атрибута Enum-Observed-Value, так и текстовую этикетку, которая может быть ассо-
циирована с каждым значением перечисления
--
EnumMsmtRangeLabels::= SEQUENCE OF EnumMsmtRangeLabel

EnumMsmtRangeLabel ::= SEQUENCE {
    value                       EnumVal,        -- определенная настройка
                                                -- перечисления
    label                       OCTET STRING    -- текстовая маркировка,
                                                -- связанная со значением
}
```

7.3.10.2 Поведение

Объект Перечисление не описывает никакие конкретные методы.

7.3.10.3 Уведомления

Объект Перечисление не формирует никакие конкретные уведомления.

7.3.11 Объект Complex Metric (Комплексный метрический показатель)

Объект:	Complex Metric
Описание:	Объект Complex Metric выступает в качестве объекта-контейнера для объектов Metric, что позволяет создавать отчеты о группе объектов, как об одной семантической сущности.
Производное от:	Metric (Метрика)
Связывание имен:	Handle (Описатель) (унаследованный объект VMO)
Зарегистрирован как:	MDC_MOC_VMO_METRIC_CMPLX

7.3.11.1 Атрибуты

Класс Объекта Complex Metric определяет атрибуты, представленные в таблице 7.23.

Таблица 7.23 — Атрибуты класса Объекта Complex Metric

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Cmplx-Metric-Info	MDC_ATTR_CMPLX_INFO	CmplxMetricInfo	Статический атрибут, обозначающий типы объектов, используемые в контейнере	M
Cmplx-Observed-Value	MDC_ATTR_CMPLX_VAL_OBS	CmplxObsValue		M
Cmplx-Dyn-Attr	MDC_ATTR_CMPLX_DYN_ATTR	CmplxDynAttr	Динамические атрибуты отдельных объектов в рамках объекта Complex Metric	O
Cmplx-Static-Attr	MDC_ATTR_CMPLX_STATIC_ATTR	CmplxStaticAttr	Статические атрибуты отдельных объектов в рамках объекта Complex Metric	O
Cmplx-Recursion-Depth	MDC_ATTR_CMPLX_RECURSION_DEPTH	INT-U16	Является обязательным если объект Complex Metric содержит дополнительные объекты Complex Metric (например, в слчуае рекурсии). Если это так, то атрибут определяет максимальную глубину рекурсии	C

Класс объекта Complex Metric должен устанавливать флаг Metric::MetricSpec::structure::ms-struct::complex.

В таблице 7.24 класс объекта Complex Metric описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.24 — Группы атрибутов класса объекта Enumeration

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_STATIC	из VMO: Type, Handle из Metric: Metric-Specification, Max-Delay-Time из Complex Metric: Cmplx-Metric-Info, Cmplx-Static-Attr, Cmplx-Recursion-Depth
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_DYN	из VMO: Label-String из Metric: Vmo-Source-List, Metric-Source-List, Unit-Code, Unit-LabelString, Msmt-Site-List, Body-Site-List, Metric-Status, Measure-Period, Averaging-Period, Start-Time, Stop-Time, Measure-Mode, Metric-Calibration, Color, Measurement-Status, Metric-Id, Metric-Id-Ext, Metric-Info-Label-String, Substance, Substance-LabelString из Complex Metric: Cmplx-Dyn-Attr
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_ METRIC_VAL_OBS	из Metric: Metric-Id-Partition из Complex Metric: Cmplx-Observed-Value

Применяют следующие определения типов:

```
--
-- Определения для атрибута Cmplx-Metric-Info
--

CmplxMetricInfo ::= SEQUENCE {
    max-mplex-obs           INT-U8,             -- максимальное количество сообщений
                                                -- до тех пор, пока все объединенные
                                                -- элементы не будут переданы в
                                                -- группу наблюдаемого значения
                                                -- метрики (Metric Observed Value
                                                -- Group)
    max-mplex-dyn           INT-U8,             -- максимальное количество сообщений
                                                -- до тех пор, пока все объединенные
                                                -- элементы не будут переданы в
                                                -- Динамическую контекстную
                                                -- группу VMO
    cm-elem-info-list       CmplxElemInfoList
}

CmplxElemInfoList ::= SEQUENCE OF CmplxElemInfo

CmplxElemInfo ::= SEQUENCE {
    class-id                OID-Type,
    max-inst                INT-U8,             -- количество объектов из типа
                                                -- class-id
```

```
    max-inst-comp       INT-U8,             -- количество составных объектов
                                            -- из типа class-id
    max-comp-no         INT-U8,             -- максимальное количество
                                            -- элементов в составном объекте
    max-inst-mplex      INT-U8,             -- количество объединенных объектов
                                            -- в диапазоне max-inst + max-inst-
                                            -- comp
    max-str-size        INT-U16             -- максимальный размер строки
}
--
-- Атрибут Cmplx-Observed-Value, представляющий собой иерархию содержащихся (вложенных) объектов класса Metric
--
CmplxObsValue ::= SEQUENCE {
    cm-obs-cnt          INT-U8,             -- счетчик последовательности
                                            -- начинает с 0, когда
                                            -- начинается мультиплексный
                                            -- период
    cm-obs-flags        CmplxFlags,
    cm-obs-elem-list    CmplxObsElemList
}

CmplxFlags ::= BITS-U8 {
    cmplx-flag-reserved(0)                  -- для будущих расширений
}

CmplxObsElemList ::= SEQUENCE OF CmplxObsElem

CmplxObsElem ::= SEQUENCE {
    cm-elem-idx             INT-U8,
    cm-obs-elem-flgs        CmplxObsElemFlags,
    attributes              AttributeList
}

CmplxObsElemFlags ::= BITS-8 {
    cm-obs-elem-flg-mplex (0),              -- элемент будет объединенным
    cm-obs-elem-flg-is-setting (2),
    cm-obs-elem-flg-updt-episodic (4),
    cm-obs-elem-flg-msmt-noncontinuous (5)
}

--
-- Атрибут Cmplx-Dyn-Attr с динамическими контекстуальными данными
-- об иерархии содержащихся объектов Metric
--
CmplxDynAttr ::= SEQUENCE {
    cm-obs-cnt              INT-U8,         -- счетчик последовательности
                                            -- начинает с 0, когда
                                            -- начинается мультиплексный
                                            -- период
    unused                  INT-U8,
    cm-dyn-elem-list
}

CmplxDynAttrElemList ::= SEQUENCE OF CmplxDynAttrElem
```

```
CmplxDynAttrElem ::= SEQUENCE {
    cm-elem-idx-1           INT-U8,         -- позволяет определить с
                                            -- помощью cm-elem-idx-2
                                            -- диапазон элементов, к
                                            -- которым применяются
                                            -- динамические атрибуты
    cm-elem-idx-2           INT-U8,
    attributes              AttributeList
}

--
-- Атрибут Cmplx-Static-Attr со статическими контекстуальными данными
-- об иерархии содержащихся объектов Metric
--
CmplxStaticAttr ::= SEQUENCE {
    cm-static-elem-list     CmplxStaticAttrElemList
}

CmplxStaticAttrElemList ::= SEQUENCE OF CmplxStaticAttrElem

CmplxStaticAttrElem ::= SEQUENCE {
    cm-elem-idx-1           INT-U8,         -- позволяет определить с
                                            -- помощью cm-elem-idx-2
                                            -- диапазон элементов, к
                                            -- которым применяются
                                            -- статические атрибуты
    cm-elem-idx-2           INT-U8,
    attributes              AttributeList   -- только статические атрибуты,
                                            -- в соответствии со
                                            -- специализацией метрики,
                                            -- разрешены (т. е.
                                            -- никаких атрибутов объектов
                                            -- VMO или Metric не
                                            -- допускается)
}
```

7.3.11.2 Поведение

Объект Complex Metric не описывает никакие конкретные методы.

7.3.11.3 Уведомления

Объект Complex Metric не формирует никакие конкретные уведомления.

7.3.12 Объект PM-Store (Хранилище постоянной метрики)

Объект:	PM-Store
Описание:	Объект класса PM-Store предоставляет возможность долговременного хранения метрических данных. Он может содержать различное количество объектов класса PM-Segment, к которым можно получить доступ только через данный объект класса PM-Store.
Производное от:	Metric (Метрика)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMO_PMSTORE

7.3.12.1 Атрибуты

Класс объекта PM-Store определяет атрибуты, представленные в таблице 7.25.

Таблица 7.25 — Атрибуты класса объекта PM-Store

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Metric-Class	MDC_ATTR_METRIC _CLASS	OID-Type	Класс объекта хранимой метрики (метрик)	M
Store-Sample-Algorithm	MDC_ATTR_METRIC _STORE_SAMPLE_ ALG	StoSampleAlg	Метод, используемый для извлечения хранимых значений из метрических наблюдаемых значений	O
Storage-Format	MDC_ATTR_METRIC _STORE_FORMAT	StorageFormat	Раскладка хранимых данных в объектах PM-Segment	M
Store-Capacity-Count	MDC_ATTR_METRIC_ STORE_CAPAC_CNT	INT-U32	Максимальное число хранимых значений	O
Store-Usage-Count	MDC_ATTR_METRIC _STORE_USAGE_ CNT	INT-U32	Действительное число хранимых значений	O
Operational-State	MDC_ATTR_OP_ STAT	OperationalState	Указывает, хранятся ли на данный момент новые выборки	O
Sample-Period	MDC_ATTR_TIME_ PD_SAMP	RelativeTime	Используется, если выборки значений осуществляются переодически	C
Number-Of-Segments	MDC_ATTR_NUM_ SEG	INT-U16	Объекты PM-Segment, экземпляры которых создаются в настоящий момент, содержащиеся в объекте PM-Store	M

Класс объекта PM-Store описывает в таблице 7.26 Группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.26 — Группы атрибутов класса объекта PM-Store

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_STATIC	из VMO: Type, Handle из PM-Store:
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_ VMO_DYN	из VMO: Label-String из PM-Store
Группа наблюдаемого значения метрики (расширяемая группа атрибутов)	MDC_ATTR_GRP_ METRIC_VAL_OBS	из PM-Store: Metric-Class, Store-Sample-Algorithm, Storage-Format, Store-Capacity-Count, Store-Usage-Count, Operational-State, Sample-Period, Number-Of-Segments

Применяют следующие определения типов:

```
--
-- Тип хранилища определяет структуру атрибута Segment-Data во всех
-- содержащихся объектах PM-Segment
--        1..255:          диапазон нормативных форматов
--        32768..65535:    диапазон частных стандартов
```

```
--        другой:                  зарезервировано
--
StorageFormat ::= INT-U16 {
    sto-t-nos (0),
    sto-t-gen (1),                 -- предполагает общий формат (т. е. объект
                                   -- PM-Segment; см. 7.3.13)
    sto-t-nu-opt (2),              -- предполагает оптимизированный формат
                                   -- объекта Numeric
    sto-t-rtsa-opt (3)             -- предполагает оптимизированный формат
                                   -- объекта Real Time Sample Array
}

--
-- Атрибут Store-Sample-Algorithm описывает процесс осуществления выборки
--
StoSampleAlg ::= INT-U16 {
    st-alg-nos (0),
    st-alg-moving-average (1),
    st-alg-recursive (2),
    st-alg-min-pick (3),
    st-alg-max-pick (4),
    st-alg-median (5)
}
```

7.3.12.2 Поведение

Объект PM-Store задает методы, представленные в таблице 7.27.

Таблица 7.27 — Методы объекта PM-Store

Действие	Режим	Идентификатор действия	Характеристика действия	Результат действия
Clear-Segments	Подтверждено	MDC_ACT_SEG_CLEAR	SegmSelection	(пусто)
Get-Segments	Подтверждено	MDC_ACT_SEG_GET	SegmSelection	SegmentAttrList
Get-Segment-Info	Подтверждено	MDC_ACT_SEG_GET_INFO	SegmSelection	Segment-InfoList

Применяют следующие определения типов:

```
--
-- SegmSelection выбирает объекты PM-Segment, на которых распространяется
-- действие метода
--
SegmSelection ::= CHOICE {
    all-segments [1]               INT-U16,        -- если, для выделения всех
                                                   -- сегментов, выбран данный тип,
                                                   -- то фактическое содержание поля
                                                   -- не имеет значения и должно
                                                   -- иметь значение 0
    segm-id-list [2]               SegmIdList,
    abs-time-range [3]             AbsTimeRange
}

--
-- SegmIdList выбирает объекты PM-Segment по их ID
--
SegmIdList ::= SEQUENCE OF InstNumber
```

```
--
-- Временной диапозон позволяет выбирать объекты PM-Segment за период
-- времени
--
AbsTimeRange ::= SEQUENCE {
    from-time                   AbsoluteTime,
    to-time                     AbsoluteTime
}

--
-- Метод Get-Segments возвращает список списков атрибута PM-Segment,
-- который включает в себя атрибут Segment-Data; номер экземпляра
-- используется для идентификации каждого сегмента
--
SegmentAttrList ::= SEQUENCE OF SegmentAttr

SegmentAttr ::= SEQUENCE {
    seg-inst-no                 InstNumber,
    seg-attr                    AttributeList
}

--
-- Информация, содержащаяся в сегменте, в результате действия Get-Segment-
-- Info, возвращает все атрибуты объектов PM-Segment, кроме атрибута
-- Segment-Data; это удобно использовать, если необходимоо получить только
-- информацию о содержании
--
SegmentInfoList ::= SEQUENCE OF SegmentInfo

SegmentInfo ::= SEQUENCE {
    seg-inst-no                 InstNumber,
    seg-info                    AttributeList
}
```

7.3.12.3 Уведомления

Объект PM-Store не формирует никакие конкретные уведомления.

7.3.13 Объект PM-Segment (Сегмент постоянной метрики)

Объект:	PM-Segment
Описание:	Объект класса PM-Segment отражает непрерывный промежуток времени, в течение которого метрические данные хранятся без каких-либо изменений значений соответствующих контекстных атрибутов (например, масштабы или этикетки). Объект класса PM-Segment доступен только через соответствующий объект класса PM-Store (например, для того, чтобы извлечь сохраняемые данные).
Производное от:	Top (Главный объект)
Связывание имен:	Instance Number (Номер экземпляра) (объектом нельзя управлять непосредственно; номер экземпляра уникален в рамках одного экземпляра PM-Store)
Зарегистрирован как:	MDC_MOC_PM_SEGMENT

7.3.13.1 Атрибуты

Класс объекта PM-Segment определяет атрибуты, представленные в таблице 7.28.

Таблица 7.28 — Атрибуты класса объекта PM-Segment

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Instance-Number	MDC_ATTR_ID_INSTNO	InstNumber		M
Metric-Id	MDC_ATTR_ID_PHYSIO	OID-Type	Идентификатор хранимой метрики (из раздела VMO::Type или Metric-Id-Partition)	M
Metric-Id-Ext	MDC_ATTR_ID_MSMT_EXT	ExtNomenRef	Динамическая идентификация метрики в другой номенклатуре или словаре. Использование данного атрибута в значительной степени ограничивает интероперабельность приложений	O
Vmo-Global-Reference	MDC_ATTR_VMO_REF_GLB	GLB-HANDLE	Ссылка на хранимый объект Metric	O
Segment-Start-Abs-Time	MDC_ATTR_TIME_START_SEG	AbsoluteTime	Начальное время сегмента	O
Segment-End-Abs-Time	MDC_ATTR_TIME_END_SEG	AbsoluteTime	Конечное время сегмента	O
Segment-Usage-Count	MDC_ATTR_SEG_USAGE_CNT	INT-U32	Действительное (т. е. текущее) количество хранимых значений	O
Segment-Data	MDC_ATTR_SEG_DATA_GEN MDC_ATTR_SEG_DATA_NU_OPT MDC_ATTR_SEG_DATA_RTSA_OPT	SegDataGen SegDataNuOpt SegDataRtsaOpt	Данные сегмента, хранимые в формате, как это установлено в объекте PMStore	M
Context Attributes	As defined for Metricderived objects	Любой атрибут из объектов производных Метрики, который является членом статической контекстной группы объекта VMO или динамической контекстной группой объекта VMO	Разрешено использование контекстных атрибутов метрики в объекте без контейнера. Атрибуты идентифицируются с помощью их OID идентификаторов. Данная ссылка на атрибуты используется для удобства редактирования. Нет необходимости копировать все атрибуты из различных объектов в объект PM-Segment. Копирование атрибутов не является скрытой формой наследования	

Класс объекта PM-Segment не обозначает никакую группу атрибутов.

Применяют следующие определения типов:

```
--
-- Формат данных общего сегмента; каждое хранимое значение является списком атрибутов
-- Примечание — Данный формат может занимать много места в хранилище
--
SegDataGen ::= SEQUENCE OF AttributeList

--
-- Оптимизированный формат объекта Numeric для периодически приобретаемых
```

```
-- численных данных (numerics); хранится только действительное значение
--
SegDataNuOpt ::=SEQUENCE OF FLOAT-Type

--
-- Оптимизированный формат объекта Real Time Sample Array; последовательный
-- массив выборок
--
SegDataRtsaOpt ::= OCTET STRING
```

7.3.13.2 Поведение

Объект PM-Segment не описывает никакие конкретные методы.

7.3.13.3 Уведомления

Объект PM-Segment не формирует никакие конкретные уведомления.

7.4 Объекты в Пакете тревоги

Определения объектов в Пакете тревоги указаны в пунктах 7.4.1—7.4.3.

7.4.1 Объект Alert (Тревога)

Объект:	Alert
Описание:	Класс Alert предназначен для отражения результата выявления простого сигнала тревоги. Поэтому, он отражает только одну единственную аварийную ситуацию физиологического или технического характера, связанную с состоянием соответствующего объекта (медицинский прибор (MDS), виртуальный медицинский прибор (VMD) или метрический показатель (Metric)).
Производное от:	VMO
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMO_AL

7.4.1.1 Атрибуты

Класс объекта Alert определяет атрибуты, представленные в таблице 7.29.

Таблица 7.29 — Атрибуты класса объекта Тревога

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Alert-Condition	MDC_ATTR_AL_COND	AlertCondition		M
Limit-Specification	MDC_ATTR_AL_LIMIT	LimitSpecEntry	Соответствует только аварийным сигналам о достижении пределельного значения	O
Vmo-Reference	MDC_ATTR_VMO_REF	HANDLE		O

Примечание — Поле производного типа объекта VMO определяет является ли Тревога сигналом технического или физиологического характера.

В таблице 7.30 класс объекта Alert описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.30 — Группы атрибутов класса объекта Тревога

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Тревога: Vmo-Reference
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из Alert: Limit-Specification
Группа тревожных сигналов	MDC_ATTR_GRP_AL	из Alert: Alert-Condition

Применяют следующие определения типов:

```
--
-- Атрибут Alert-Condition (Опасная ситуация) — это сообщение о статусе
-- процесса, который обнаруживает тревогу
--
AlertCondition ::= SEQUENCE {
    obj-reference       HANDLE,
    controls            AlertControls,
    alert-flags         AlertFlags,             -- поддерживающие флажки
    alert-source        OID-Type,               -- из раздела номенклатуры метрики
                                                -- или раздела объектно-
                                                -- ориентированной номенклатуры
    alert-code          OID-Type,               -- из раздела номенклатуры событий
    alert-type          AlertType,              -- определяет тип и степень
                                                -- тяжести состояния (ситуации)
    alert-info-id       PrivateOid,             -- конкретная информация может
                                                -- быть добавлена; 0, если не
                                                -- используется
    alert-info          ANY DEFINED BY
                        alert-info-id
}
```

Примечание — Код alert-code берется из раздела номенклатуры событий. Записи (т.е. коды) в данном разделе — это четные числа. Последний бит кода используется для определения раздела номенклатуры, откуда берется al-source (в объекте Alert Monitor, см. 7.4.3.1). Если последний бит — 0, al-source берется из раздела номенклатуры метрики. Если последний бит — 1 (1 добавляется к основному коду в номенклатуре события), источник al-source — из раздела объектно-ориентированной номенклатуры.

```
--
-- Средства управления объекта Alert задают флажки для передачи информации
-- о статусе, имеющей значение для процессора аварийных сигналов; данная
-- структура повторно используется в объекте Alert Status
--
AlertControls ::= BITS-16 {
    ac-obj-off (0),                 -- объект, контролируемый сигналом
                                    -- тревоги, выключен
    ac-chan-off (1),                -- канал выключен
```

```
    ac-all-obj-al-off (3),          -- все сигналы тревоги,
                                    -- контролирующие
                                    -- объекты, на которые дается
                                    -- ссылка, выключены
    ac-alert-off (4),               -- данный процесс контроля тревоги
                                    -- выключен
    ac-alert-muted (5)              -- данный сигнал тревоги временно
                                    -- приглушен пользователем
                                    -- (например, на аппаратах ИВЛ,
                                    -- чтобы провести физиотерапию или
                                    -- отсос жидкости)
}

--
-- Флажки сигнала тревоги дают дополнительную информацию о том, как
-- обрабатывать создавшуюся ситуации; данная структура также используется
-- объектом Alert Status
--
AlertFlags ::= BITS-16 {
    local-audible (1),              -- указывает, что сигнал слышен в
                                    -- местной системе
    remote-audible (2),             -- сигнал можно услышать на
                                    -- расстоянии (т.е. не подавлен)
    visual-latching (3),            -- имеется зрительная блокировка
                                    -- сигнала
    audible-latching (4),           -- имеется аудио блокировка сигнала
    derived (6),
    record-inhibit (8)              -- не запускает запись тревожного
                                    -- сигнала
}
--
-- Тип Alert используется для установления различий между уровнем
-- серьезности технического и физиологического аварийного сигнала
--
AlertType ::= INT-U16 {
    no-alert (0),
    low-pri-t-al (1),               -- техническая тревога с низким уровнем приоритета
    med-pri-t-al (2),               -- техническая тревога со средним уровнем приоритета
    hi-pri-t-al (4),                -- техническая тревога с высоким уровнем приоритета
    low-pri-p-al (256),             -- для информации
    med-pri-p-al (512),             -- требуется быстрое реагирование (т. е. анормальное
                                       состояние)
    hi-pri-p-al (1024)              -- требуется немедленное реагирование
                                       (т. е. чрезвычайное состояние)
}

--
-- Атрибут Limit-Specification указывает на контролируемый предельный
-- диапазон
--
LimitSpecEntry ::= SEQUENCE {
    object-handle       HANDLE,
    al-source-id        OID-Type,   -- как правило, это метрический
                                    -- идентификатор измерения
```

```
    unit-code          OID-Type,          -- из раздела DIM
    lim-al-stat        CurLimAlStat,      -- определение см. в 7.6.8.1
    lim-al-val         CurLimAlVal        -- определение см. в 7.6.8.1
}
```

7.4.1.2 Поведение

Объект Тревога не описывает никакие конкретные методы.

7.4.1.3 Уведомления

Объект Тревога не формирует никакие конкретные уведомления.

7.4.2 Объект Alert Status (Уровень тревоги)

Объект: Alert Status

Описание: Объект класса Alert Status отражает результат процесса обработки сигнала тревоги, который включает всю совокупность аварийных состояний для одного или нескольких медицинских объектов. В отличие от объекта класса Alert, объект класса Alert Status собирает данные по всем авариям, относящимся либо к иерархии объектов класса VMD, либо к объекту класса MDS, и представляет эту информацию в виде структурированного списка атрибутов.

Производное от: VMO

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_VMO_AL_STAT

7.4.2.1 Атрибуты

Класс объекта Alert Status определяет атрибуты, представленные в таблице 7.31.

Таблица 7.31 — Атрибуты класса объекта Alert Status

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Alert-Capab-List	MDC_ATTR_AL_STAT_AL_C_LIST	AlertCapabList	Функциональные возможности объекта Alert Status	M
Tech-Alert-List	MDC_ATTR_AL_STAT_AL_T_LIST	AlertList	Список информации о тревоге технического характера	O
Physio-Alert-List	MDC_ATTR_AL_STAT_AL_P_LIST	AlertList	Список информации о тревоге физиологического характера	O
Limit-Spec-List	MDC_ATTR_AL_LIMIT_SPEC_LIST	LimitSpecList	Список предельных диапазонов для аварийных сигналов	O

В таблице 7.32 класс объекта Alert Status описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.32 — Группы атрибутов класса объекта Alert Status

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Alert Status: Alert-Capab-List

Окончание таблицы 7.32

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из Alert Status: Limit-Spec-List
Группа тревожных сигналов	MDC_ATTR_GRP_AL	из Alert Status: Tech-Alert-List, Physio-Alert-List

Применяют следующие определения типов:

```
--
-- Список тревог используется для сообщения об аварийных ситуациях,
-- полученных из объекта Alert Status
--
AlertList ::= SEQUENCE OF AlertEntry

AlertEntry ::= SEQUENCE {
    obj-reference       HANDLE,
    instance            InstNumber,             -- для поддержки множественных
                                                -- аварийных сигналов одного
                                                -- объекта
    controls            AlertControls,
    alert-source        OID-Type,               -- из метрического или объектно-
                                                -- ориентированного раздела
                                                -- номенклатуры
    alert-code          OID-Type,               -- из раздела номенклатуры сигналов тревоги
    alert-type          AlertType,
    alert-info-id       PrivateOid,
    alert-info          ANY DEFINED BY          alert-info-id
}
```

Примечание — Код alert-code берется из раздела номенклатуры событий. Записи (т. е. коды) в данном разделе — это четные числа. Последний бит кода используется для определения раздела номенклатуры, откуда берется al-source (в объекте Alert Monitor, см. 7.4.3.1). Если последний бит — 0, al-source берется из раздела номенклатуры метрики. Если последний бит — 1 (1 добавляется к основному коду в номенклатуре события), источник al-source — раздела объектно-ориентированной номенклатуры.

```
--
-- Объект Alert Status дает список функциональных возможностей с записями
-- для каждого контролируемого объекта в области его действия
--
AlertCapabList ::= SEQUENCE OF AlertCapabEntry
AlertCapabEntry ::= SEQUENCE {
    obj-reference       HANDLE,
    obj-class           OID-Type,
    alert-group         OID-Type,               -- позволяет объединять объекты
                                                -- Alert в группы так, чтобы
                                                -- процессор мог выбрать только один
                                                -- объект из данных групп для
                                                -- отображения(метрический
                                                -- идентификатор)
    al-rep-flags        BITS-16                 -- определяет, как коммуницируются
                                                -- множественные аварийные сигналы
```

```
    { dyn-inst-contents(1), rep-all-inst(2) },
    max-t-severity      AlertType,          -- технический аварийный сигнал
                                            -- наиболее высокого уровня
                                            -- серьезности
    max-t-obj-al        INT-U16,            -- максимальное число технических
                                            -- аварийных сигналов, запущенных
                                            -- параллельно, для данного объекта
    max-p-severity      AlertType,          -- физиологический аварийный сигнал
                                            -- наиболее высокого уровня
                                            -- серьезности
    max-p-obj-al        INT-U16             -- максимальное число
                                            -- физиологических аварийных
                                            -- сигналов, запущенных параллельно,
                                            -- для данного объекта
}

--
-- Атрибут Limit-Spec-List обозначает контролируемые предельные диапазоны
--
LimitSpecList ::= SEQUENCE OF LimitSpecEntry
```

7.4.2.2 Поведение

Объект Статус тревоги не описывает никакие конкретные методы.

7.4.2.3 Уведомления

Объект Статус тревоги не формирует никакие конкретные уведомления.

7.4.3 Объект Alert Monitor (Монитор тревожных ситуаций)

Объект:	Alert Monitor
Описание:	Объект класса Alert Monitor отражает результаты работы системного процессора сигналов тревоги. В качестве такового, он отражает общее описание аварийного состояния для прибора или системы в целом и представляет эту информацию в форме совокупного списка всех аварийных ситуаций в системе. Этот список включает информацию об общем состоянии системы и информацию о конкретных аварийных ситуациях, что позволяет реализовать соответствующее существующим стандартам безопасности отображение информации о тревоге в удаленной системе (на удаленном устройстве).
Производное от:	VMO
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMO_AL_MON

7.4.3.1 Атрибуты

Класс объекта Alert Monitor определяет атрибуты, представленные в таблице 7.33.

Таблица 7.33 — Атрибуты класса объекта Alert Monitor

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Condition	MDC_ATTR_DEV_AL_COND	DevAlert-Condition	Общий статус тревоги прибора	M
Device-P-Alarm-List	MDC_ATTR_AL_MON_P_AL_LIST	DevAlarmList	Список активных физиологических аварийных сигналов	M

Окончание таблицы 7.33

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Device-T-Alarm-List	MDC_ATTR_AL_MON_T_AL_LIST	DevAlarmList	Список активных аварийных сигналов технического характера	M
Device-Sup-Alarm-List	MDC_ATTR_AL_MON_S_AL_LIST	DevAlarmList	Список подавленных аварийных сигналов физиологического характера	O
Limit-Spec-List	MDC_ATTR_AL_LIMIT_SPEC_LIST	LimitSpecList	Список предельных диапазонов аварийного сигнала	O
Suspension-Period	MDC_ATTR_TIME_PD_AL_SUSP	RelativeTime	Оставшееся время удержания сигнала	O

Класс объекта Alert Monitor описывает в таблице 7.34 группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.34 — Группы атрибутов класса объекта Alert Monitor

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из Alert Monitor:
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из Alert Monitor: Limit-Spec-List
Группа отслеживания тревог	MDC_ATTR_GRP_AL_MON	из Alert Monitor: Device-Alert-Condition, Device-P-Alarm-List, Device-T-Alarm-List, Device-Sup-Alarm-List, Suspension-Period

Применимы следующие определения типов:

```
--
-- Атрибут Device-Alert-Condition описывает общий статус аварийного сигнала MDS
--
DevAlertCondition ::= SEQUENCE {
    device-alert-state          AlertState,
    al-stat-chg-cnt             AlStatChgCnt,       -- счетчик изменений отмечает
                                                    -- изменение состояния или активных
                                                    -- тревог
    max-p-alarm                 AlertType,
    max-t-alarm                 AlertType,
    max-aud-alarm               AlertType           -- максимальный уровень серьезности
                                                    -- воспринимаемого на слух
                                                    -- аварийного сигнала
}

AlertState ::= BITS-16 {
    al-inhibited (0),                               -- выключено
```

```
    al-suspended (1),                   -- тревога (сигнализация) временно
                                        -- отключена; опасная ситуация
                                        -- принята во внимание
    al-latched (2),                     -- фиксируется кокретное
                                        -- предупреждение (или AlMon
                                        -- фиксирует аварийные ситуации)
    al-silenced-reset (3),              -- (только для перехода);
                                        -- индикация тревоги прекращена,
                                        -- но аварийная сигнализация
                                        -- возобновляется
    al-dev-in-test-mode (5),            -- прибор в тестовом режиме;
                                        -- аварийные сигналы не являются
                                        -- сигналами, подаваемыми от
                                        -- реальных пациентов
    al-dev-in-standby-mode (6),         -- прибор в режиме ожидания
    al-dev-in-demo-mode (7)             -- прибор в демонстрационном
                                        -- режиме, аварийные сигналы не
                                        -- являются сигналами, подаваемыми
                                        -- от реальных пациентов
}

AlStatChgCnt ::= SEQUENCE {
    al-new-chg-cnt          INT-U8,     -- Атрибут Device-Alert-Condition
                                        -- изменен
    al-stack-chg-cnt        •INT-U8     -- стек тревог (атрибуты списка
                                        -- активных аварийных сигналов)
                                        -- изменен
}

--
-- Список сигналов прибора
--
DevAlarmList ::= SEQUENCE OF DevAlarmEntry

DevAlarmEntry ::= SEQUENCE {
    al-source           OID-Type,       -- из метрического или объектно-
                                        -- ориентированного раздела
                                        -- номенклатуры
    al-code             OID-Type,       -- из раздела номенклатуры событий
    al-type             AlertType,
    al-state            AlertState,
    object              ManagedObjectId,
    alert-info-id       PrivateOid,
    alert-info          ANY DEFINED BY alert-info-id
}
```

Примечание — Код alert-code берется из раздела номенклатуры событий. Записи (т. е. коды) в данном разделе — это четные числа. Последний бит кода используется для определения раздела номенклатуры, откуда берется al-source (в объекте Alert Monitor, см. 7.4.3.1). Если последний бит — 0, al-source берется из раздела номенклатуры метрики. Если последний бит — 1 (1 добавляется к основному коду в номенклатуре события), источник al-source — раздела объектно-ориентированной номенклатуры.

7.4.3.2 Поведение

Объект Отслеживание сигналов не определяет никакие конкретные методы.

7.4.3.3 Уведомления

Объект Отслеживание сигналов не формирует никакие конкретные уведомления.

7.5 Объекты в Пакете система

Определения объектов Пакета система указаны в пунктах 7.5.1—7.5.10.

7.5.1 Объект Virtual Medical System (Виртуальная медицинская система, VMS)

Объект: VMS

Описание: Класс VMS — абстрактный базовый класс для всех классов Пакета системы в данной модели. Это обстоятельство позволяет установить непротиворечивую систему наименования и обозначения классов в пакете System.

Производное от: Top (Главный объект)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_VMS

7.5.1.1 Атрибуты

Класс объекта VMS определяет атрибуты, представленные в таблице 7.35.

Таблица 7.35 — Атрибуты класса объекта VMS

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_ HANDLE	HANDLE	Атрибут связывания имен	M
System-Type	MDC_ATTR_SYS_ TYPE	TYPE	***Примеры — Аппарат ИВЛ, монитор, как указано в номенклатуре***	M
System-Model	MDC_ATTR_ID_ MODEL	SystemModel	Модель описывает производителя и номер модели	C
System-Id	MDC_ATTR_SYS_ID	OCTET STRING	Уникальный идентификатор системы, например, серийный номер	C
Compatibility-Id	MDC_ATTR_ID_ COMPAT	INT-U32	Для использования производителем	O
Nomenclature-Version	MDC_ATTR_NOM_ VERS	Nomenclature-Version	Версия номенклатуры, применяемой системой	C
System-Capability	MDC_ATTR_SYS_ CAPAB	SystemCapability	Набор поддерживаемых функций; зависит от системы	O
System-Specification	MDC_ATTR_SYS_ SPECN	SystemSpec	Определяет функциональные компоненты	O
Production-Specification	MDC_ATTR_ID_ PROD_SPECN	ProductionSpec	Новые версии компонентов, серийные номера и т. д	O
Ext-Obj-Relations	MDC_ATTR_EXT_ OBJ_RELATION	ExtObjRelation-List	Связь с объектами, которые не определены в DIM	O

Примечание — Условные (C) атрибуты системы VMS обязательны для высокоуровневого экземпляра объекта VMS (т. е. экземпляра корневого объекта дерева состава). В других обстоятельствах они используются по желанию.

В таблице 7.36 класс объекта системы VMS описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.36 — Группы атрибутов класса объекта VMS

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов идентификации системы (расширяемая группа атрибутов)	MDC_ATTR_GRP_ SYS_ID	из VMS: Type, Handle из Metric: System-Type, System-Model, System-Id, Compatibility-Id, Nomenclature-Version
Группа атрибутов применения системы (расширяемая группа атрибутов)	MDC_ATTR_GRP_ SYS_APPL	из VMS: System-Capability, System-Specification
Группа атрибутов системного производства (расширяемая группа атрибутов)	MDC_ATTR_GRP_ SYS_PROD	из VMS: Production-Specification
Группа атрибутов отношений	MDC_ATTR_GRP_ RELATION	из VMS: Ext-Obj-Relations

Необходимо отметить, что группа атрибутов Отношения не будет снова указана в определениях производных классов.

Применяют следующие определения типов:

```
--
-- Атрибут System-Model (Модель системы) определяется производителем и
-- номером модели, зависящим от производителя
--
SystemModel ::= SEQUENCE {
    manufacturer        OCTET STRING,
    model-number        OCTET STRING
}

--
-- Атрибут System-Capability (Функциональные возможности системы) — это
-- высокоуровневая спецификация реализованных функций; (описанное ниже,
-- приведено исключительно в качестве примера)
--
SystemCapability ::= BITS-32 {
    sc-multiple-context (0),            -- указывает на то, что система
                                        -- использует множественные
                                        -- контексты именования
    sc-dyn-configuration (1),           -- дерево состава подлежит
                                        -- динамическим изменениям
    sc-dyn-scanner-create (2),          -- система позволяет основному
                                        -- устройству (host) динамически
                                        -- создавать объекты Scanner
    sc-auto-init-scan-list (3),         -- объект CfgScanner
                                        -- поддерживает автоматическую
                                        -- инициализацию списка сканирования
    sc-auto-updt-scan-list (4)          -- объект CfgScanner поддерживает
                                        -- автоматическое обновление списка сканирования
}

--
-- Атрибут System-Specification (Спецификация системы) позволяет создавать
-- специальные записи для функциональных компонентов системы
--
```

```
SystemSpec ::= SEQUENCE OF SystemSpecEntry

SystemSpecEntry ::= SEQUENCE {
    component-capab-id          PrivateOid,
    component-spec              ANY DEFINED BY component-capab-id
}

--
-- Атрибут Production-Specification (Спецификация производства) имеет дело
-- с серийными номерами, номерами частей, обновлениями версий и т. д.;
-- Необходимо отметить, что прибор может иметь множество компонентов,
-- поэтому Атрибут Production-Specification должен быть представлен
-- печатной строкой, определяющей компонент и номер
--
ProductionSpec ::= SEQUENCE OF ProdSpecEntry

ProdSpecEntry ::= SEQUENCE {
    spec-type                   INT-U16 {
    unspecified (0),
    serial-number (1),
    part-number (2),
    hw-revision (3),
    sw-revision (4),
    fw-revision (5),
    protocol-revision (6),
    prod-spec-gmdn (7)                          -- Всемирная номенклатура
                                                -- медицинских изделий[1)]
    },
    component-id                PrivateOid,
    prod-spec                   OCTET STRING
}

--
-- Атрибут Nomenclature-Version (Версия номенклатуры) содержит часть поля
-- основной версии (т.е. основную совместимость) и вспомогательной версии
-- (используемой для идентификации последнего примененного обновления);
-- часть основной версии закодирована в виде битового поля так, чтобы
-- системы, поддерживающие разные версии, могли согласовать версию,
-- используемую в рамках одного соединения
--
NomenclatureVersion ::= SEQUENCE {
    nom-major-versio            BITS-16 {       -- идентификатор основного
                                                -- номера версии
    majorVersion1 (0),
    majorVersion2 (1),
    majorVersion3 (2),
    majorVersion4 (3)
    },
    nom-minor-version           INT-U16         -- счетчик для идентификации
                                                -- незначительных обновлений
}
```

[1)] Всемирная номенклатура медицинских изделий (GMDN) основана на стандарте ИСО 15225 и была разработана при содействии CEN TC257 SC1.

7.5.1.2 Поведение
Объект VMS не описывает никакие конкретные методы.
7.5.1.3 Уведомления
Объект VMS не формирует никакие конкретные уведомления.

7.5.2 Объект Medical Device System (MDS)

Объект:	MMS
Описание:	Класс MDS — абстракция медицинского прибора, которая предоставляет медицинские данные в форме объектов, определенных в Медицинском пакете рассматриваемой информационной модели предметной области. Специализированные подклассы данного класса используется для отражения различий между системами по степени сложности и области применения. Будучи базовым абстрактным классом, класс MDS не может иметь конкретных экземпляров-представителей.
Производное от:	VMS
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMS_MDS

7.5.2.1 Атрибуты
Класс объекта системы MDS определяет атрибуты, представленные в таблице 7.37.

Таблица 7.37 — Атрибуты класса объекта системы MDS

Название атрибута	Идентификатор атрибута[a)]	Тип атрибута	Примечание	Квалификатор[b)]
Mds-Status	MDC_ATTR_VMS_MDS_STAT	MDSStatus	Состояние прибора согласно конечному автомату FSM системы MDS	C
Bed-Label	MDC_ATTR_ID_BED_LABEL	OCTET STRING	Печатная строка, идентифицирующая местонахождение системы	O
Soft-Id	MDC_ATTR_ID_SOFT	OCTET STRING	Настраиваемый атрибут, например, больничный инвентарный номер	O
Operating-Mode	MDC_ATTR_MODE_OP	PrivateOid		O
Application-Area	MDC_ATTR_AREA_APPL	ApplicationArea		O
Patient-Type	MDC_ATTR_PT_TYPE	PatientType	Может управлять алгоритмами, см. 7.10.1.1 для определения типа	O[c)]
Date-and-Time	MDC_ATTR_TIME_ABS	AbsoluteTime	MDS поддерживает время прибора	O
Relative-Time	MDC_ATTR_TIME_REL	RelativeTime		O
HiRes-Relative-Time	MDC_ATTR_TIME_REL_HI_RES	HighResRelativeTime		O
Power-Status	MDC_ATTR_POWER_STAT	PowerStatus	Питание onBattery (от аккумулятора) или onMains (от сети)	O[d)]
Altitude	MDC_ATTR_ALTITUDE	INT-I16	Измеряет абсолютные отметки (выше/ниже уровня моря)	O

Окончание таблицы 7.37

Название атрибута	Идентификатор атрибута[a)]	Тип атрибута	Примечание	Квалификатор[b)]
Battery-Level	MDC_ATTR_VAL_BAT_CHARGE	INT-U16	В % от емкости; не определяется, если значение > 100	O
Remaining-Battery-Time	MDC_ATTR_TIME_BATT_REMAIN	BatMeasure	См. 7.5.9.1 по определениям типа; минуты являются рекомендуемой единицей измерения	O
Line-Frequency	MDC_ATTR_LINE_FREQ	LineFrequency	Частота сети; выражена в герцах (обычно 50 Гц или 60 Гц)	O
Association-Invoke-Id	MDC_ATTR_ID_ASSOC_NO	INT-U16	Счетчик числа ассоциаций в данном коммуникационном порте увеличивается с каждой ассоциацией сервисного элемента управления ассоциацией (ACSE)	O
Locale	MDC_ATTR_LOCALE	Locale	Определяет кодировку символов и язык атрибутов печатной строки в данной системе MDS и содержащихся объектах. Объекты, содержащиеся в системе MDS или приборе VMD могут задавать разные атрибуты Locale для своих областей применения	C

[a)] Должен быть произведен обмен некоторыми атрибутами VMS и MDS на поля информации пользователя в протоколе ACSE. Поля информации пользователя ACSE должны содержать только атрибуты VMS или MDS.

[b)] Условные (C) атрибуты системы MDS обязательны для высокоуровневого экземпляра объекта VMS (т. е. экземпляра корневого объекта дерева состава); в других обстоятельствах они используются по желанию.

[c)] Если система MDS поддерживает объект Patient Demographics (Индивидуальные данные пациента), то объект MDS не должен содержать этот атрибут, чтобы исключить конфликты данных.

[d)] Если требуется больше информации об источнике питания для приборов с химическим источником питания (в частности, если источник питания управляем), то следует использовать специальный объект Battery (Батарея).

В таблице 7.38 класс объекта MDS описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.38 — Группы атрибутов класса объекта MDS

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов идентификации системы (расширяемая группа атрибутов)	MDC_ATTR_GRP_SYS_ID	из VMS: System-Type, System-Model, System-Id, Compatibility-Id, Nomenclature-Version из MDS: Soft-Id, Association-Invoke-Id, Locale
Группа атрибутов применения системы (расширяемая группа атрибутов)	MDC_ATTR_GRP_SYS_APPL	из VMS: System-Capability, System-Specification из MDS: Mds-Status, Operating-Mode, Patient-Type, Date-and-Time, Power-Status, Battery-Level, Remaining-Battery-Time, Application-Area, Bed-Label, Relative-Time, HiRes-Relative-Time, Altitude, Line-Frequency
Группа атрибутов системного производства (расширяемая группа атрибутов)	MDC_ATTR_GRP_SYS_PROD	из VMS: Production-Specification

Применяют следующие определения типов:

```
--
-- MDS состояние одной ассоциации/соединения согласно FSM (конечному
-- автомату)
--
MDSStatus ::= INT-U16 {
    disconnected (0),       unassociated (1),       associating (2),        associated (3),
    configuring (4),        configured (5),         operating (6),          re-initializing (7),
    terminating (8),        disassociating (9),     disassociated (10),     re-configuring (11)
}

--
-- Атрибут Application-Area (Прикладная область)
--
ApplicationArea ::= INT-U16 {
    area-unspec (0),
    area-operating-room (1),
    area-intensive-care (2)
}

--
-- Атрибут Power-Status (Состояние питания) определяет, подключено ли
-- устройство к источнику питания или к сети; верхние биты определяют
-- уровень заряда
--
PowerStatus ::= BITS-16 {
    onMains (0),
    onBattery (1),
    chargingFull (8),
    chargingTrickle (9),
    chargingOff (10)
}

--
-- Атрибут Line-Frequency (Частота сканирования)
--
LineFrequency ::= INT-U16 {
    line-f-unspec (0),
    line-f-50hz (1),
    line-f-60hz (2)
}
```

7.5.2.2 Поведение

Объект системы MDS определяет методы, описанные в таблице 7.39.

Таблица 7.39 — Методы объекта MDS

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Mds-Set-Status	Подтверждено	MDC_ACT_SET_MDS_ STATE	MdsSetStateInvoke	MdsSetState-Result

Применяют следующие определения типов:

```
--
-- Метод MDS-Set-State (Установка состояния MDS) позволяет изменить
-- состояние конечного автомата системы MDS, например, для активации сброса
```

```
-- настроек (если поддерживается прибором)
-- Примечание — Использование типа зависит от реализации, в особенности,
-- учитывая проблемы защиты и координации, связанные с этим использованием
--
MdsSetStateInvoke ::= SEQUENCE {
    new-state MDSStatus,
    authorization INT-U32
}

MdsSetStateResult ::= MDSStatus
```

7.5.2.3 Уведомления

Объект системы MDS определяет события, описанные в таблице 7.40.

Таблица 7.40 — События объекта системы MDS

Событие	Режим	Идентификатор события	Параметр события	Результат события
System-Error	Неподтверждено	MDC_NOTI_SYS_ERR	MdsErrorInfo	—
Mds-Create-Notification	Подтверждено	MDC_NOTI_MDS_CREAT	MdsCreateInfo	—
Mds-Attribute-Update	Подтверждено	MDC_NOTI_MDS_ ATTR_UPDT	Mds-Attribu-teChange-Info	—

Применяют следующие определения типов:

```
--
-- Уведомление System-Error в случае системных ошибок
--
MdsErrorInfo ::= SEQUENCE {
    error-type          PrivateOid,
    error-info          ANY DEFINED BY error-type
}

--
-- Событие Mds-Create-Notification (Создание уведомления MDS) отправляется
-- после того, как установится ассоциация
--
MdsCreateInfo ::= SEQUENCE {
    class-id                ManagedObjectId,
    attribute-list          AttributeList           -- атрибуты из Группы атрибутов
                                                    -- идентификации системы и
                                                    -- Группы атрибутов применения
                                                    -- системы
}
--
-- Система MDS может отправлять отчеты об изменении значений атрибутов
--
MdsAttributeChangeInfo ::= AttributeList
```

7.5.3 Объект Simple MDS (Простая MDS)

Объект:	Simple MDS
Описание:	Класс Simple MDS описывает медицинский прибор, который содержит единственный экземпляр объекта VMD (монофункциональный прибор).

Производное от:	MDS
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMS_MDS_SIMP

Данная специализация объекта MDS не определяет никакие специализированные атрибуты, методы и уведомления.

7.5.4 Объект MDS (Множественная MDS)

Объект:	Simple MDS
Описание:	Класс Hydra MDS описывает медицинский прибор, который содержит несколько разных экземпляров объекта VMD (многофункциональный прибор).
Производное от:	MDS
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMS_MDS_HYD

Данная специализация объекта MDS не определяет никакие специализированные атрибуты, методы и уведомления.

7.5.5 Объект Composite Single Bed MDS (Составная простая прикроватная MDS)

Объект:	Composite Single Bed MDS
Описание:	Класс Composite Single Bed MDS описывает медицинский прибор, который включает в себя (или связывается посредством некоторого интерфейса с) один или несколько объектов классов Simple MDS или Hydra MDS, сосредоточенных в одном месте (то есть, на или у одной кровати).
Производное от:	MDS
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMS_MDS_COMPOS_ SINGLE_BED

Данная специализация объекта MDS не определяет никакие специализированные атрибуты, методы и уведомления.

7.5.6 Объект Composite Multiple Bed MDS (Составная множественная прикроватная MDS)

Объект:	Composite Multiple Bed MDS
Описание:	Класс Composite Multiple Bed MDS описывает медицинский прибор, который содержит (или связывается посредством некоторого интерфейса) несколько объектов классов Simple MDS или Hydra MDS, распределенных по нескольким местоположениям (то есть, на или у нескольких кроватей).
Производное от:	MDS
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_VMS_MDS_COMPOS_MULTI_BED

Данная специализация объекта MDS не определяет никакие специальные атрибуты, методы и уведомления.

7.5.7 Объект Log (Журнал)

Объект:	Log
Описание:	Класс log — абстрактный базовый класс, который является контейнером для хранения важных локальных системных уведомлений и событий. Можно определить специализированные классы-потомки для различных типов событий. Будучи базовым абстрактным классом, класс Log не может иметь конкретных экземпляров-представителей.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_LOG

7.5.7.1 Атрибуты

Класс объекта Log определяет атрибуты, представленные в таблице 7.41.

Таблица 7.41 — Атрибуты класса объекта Log

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
Max-Log-Entries	MDC_ATTR_LOG_ENTRIES_MAX	INT-U32	Максимальная вместимость объекта Log; сервис GET используется для получения данного атрибута	M
Current-Log-Entries	MDC_ATTR_LOG_ENTRIES_CURR	INT-U32	Вместимость объекта Log, используемая на данный момент; сервис GET используется для получения данного атрибута	M
Log-Change-Count	MDC_ATTR_LOG_CHANGE_COUNT	INT-U16	Увеличивается, когда происходит изменение содержимого журнала	O

Примечание — Предполагается, что записи объекта Log индексируются от 0 до значения атрибута Current-Log-Entries (текущие записи в журнале).

Класс объекта Log не определяет никакие атрибутивные группы, а также ему не требуются никакие дополнительные определения типа.

7.5.7.2 Поведение

Объект Log определяет методы, описанные в таблице 7.42.

Таблица 7.42 — Методы объекта Log

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Clear-Log	Подтверждено	MDC_ACT_CLEAR_LOG	ClearLogRange-Invoke (не обязательный)	ClearLog-RangeResult (не обязательный)

Применяют следующие определения типов:

```
--
-- Диапазон записей в журнале, предназначенных для удаления; если параметр
-- не добавлен к методу Clear-Log (очистка журнала), журнал будет очишен
-- полностью в обязательном порядке
--
ClearLogRangeInvoke ::= SEQUENCE {
    clear-log-option                ClearLogOption,
```

```
    log-change-count            INT-U16,        -- 0 безвозвратная очистка
    from-log-entry-index        INT-U32,
    to-log-entry-index          INT-U32
}
ClearLogRangeResult ::= SEQUENCE {
    clear-log-result            ClearLogResult,
    log-change-count            INT-U16,        -- подсчет текущих изменений
                                                -- после очистки
    from-log-entry-index        INT-U32,        -- не важно, если очистка не
                                                -- пройдет успешно
    to-log-entry-index          INT-U32         -- не важно, если очистка не
                                                -- пройдет успешно
    current-log-entries         INT-U32         -- обновленное число записей в
                                                -- журнале
}

--
-- Опции, которые управляют командой очистки
--

ClearLogOptions ::= BITS-16 {
    log-clear-if-unchanged (1)                  -- выполнять только данное действие,
                                                -- если журнал не был изменен;
                                                -- другими словами, в запросе все
                                                -- еще присутствует evlog-change-
                                                -- count
}

--
-- Результат функции очистки журнала
--
ClearLogResult ::= INT-U16 {
    log-range-cleared (0),                      -- успешно выполненная операция
    log-changed-clear-error (1),                -- неправильный подсчет изменений
                                                -- (т.е журнал был модифицирован)
    log-change-counter-not-supported (2)        -- журнал не поддерживает счетчик
                                                -- изменений
}
```

Примечание — Обработка счетчика изменений при команде очистка предотвращает соперничество, при котором записи системного журнала, которые еще не были загружены пользователем, неумышленно удаляются.

7.5.7.3 Уведомления

Объект Log не формирует никакие конкретные уведомления.

7.5.8 Объект Event Log (Журнал событий)

Объект:	Event Log
Описание:	Класс Event Log — основная разновидность (класс-потомок) класса Log, предназначен для хранения системных событий в свободном текстовом или в бинарном представлении.
Производное от:	Log (Журнал)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_LOG_EVENT

7.5.8.1 Атрибуты

Класс объекта Event Log определяет атрибуты, представленные в таблице 7.43.

Таблица 7.43 — Атрибуты класса объекта Event Log

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Type	MDC_ATTR_ID_TYPE	TYPE	Дополнительная спецификация формата записи в журнале	O
Event-Log-Entry-List	MDC_ATTR_EVENT_LOG_ENTRY_LIST	EventLogEntry-List	Записи о событиях; могут извлекаться с помощью сервис GET	M
Event-Log-Info	MDC_ATTR_EVENT_LOG_INFO	EventLogInfo	Статические и динамические спецификации	O

Класс объекта Event Log не определяет никакие атрибутивные группы.

Применяют следующие определения типов:

```
--
-- Атрибут Event-Log-Entry-List (Список записей журнала событий)
--
EventLogEntryList ::= SEQUENCE OF EventLogEntry

EventLogEntry ::= SEQUENCE {
    entry-number        INT-U32,        -- счетчик записей не зависит от
                                        -- номера индекса, который
                                        -- используется для получения доступа
    abs-time            AbsoluteTime,   -- время события
    event-entry         OCTET STRING    -- информация о событии,
                                        -- указанного в произвольной или в
                                        -- двоичной форме; структура
                                        -- определяется атрибутом Type
}

--
-- Атрибут Event-Log-Info (Информация о журнале событий)
-- Биты с 0 по 15 зарезервированы для статической информации; биты с 16 по
-- 31 динамически обновляются для отображения изменений статуса журнала.
-- Если данный атрибут отсутствует, все биты условно принимаются за 0
--
EventLogInfo ::= BITS-32 {
    ev-log-clear-range-sup (0),     -- поддерживает очистку указанных
                                    -- диапазонов (не только всего
                                    -- журнала)
    ev-log-get-act-sup (1),         -- поддерживает загрузку отдельных
                                    -- записей с помощью метода Get-
                                    -- Event-Log (получение журнала
                                    -- событий) (а не только простого
                                    -- сервиса GET)
    ev-log-binary-entries (8),      -- записи журнала представлены в
                                    -- двоичной форме, не в произвольной
```

```
    ev-log-full (16),                   -- журнал заполнен; очистка
                                        -- проводится, как только журнал
                                        -- будет содержать минимум 1
                                        -- произвольную запись в результате
                                        -- выполнения очистки
    ev-log-wrap-detect (17)             -- устанавливается, когда журнал
                                        -- заполнен и перезаписана первая
                                        -- старая запись; очистка проводится,
                                        -- как только журнал будет содержать
                                        -- минимум 1 произвольную запись в
                                        -- результате выполнения очистки
}
```

7.5.8.2 Поведение

Объект Event Log описывает методы в таблице 7.44.

Таблица 7.44 — Методы объекта Event Log

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Get-Event-Log-Entries	Подтверждено	MDC_ACT_GET_EVENT_LOG_ENTRIES	GetEventLogEntry-Invoke	GetEventLog-EntryResult

Применяют следующие определения типов:

```
--
-- Диапазон записей журнала, предназначенных для загрузки
--
GetEventLogEntryInvoke ::= SEQUENCE {
    from-log-entry-index            INT-U32,
    to-log-entry-index              INT-U32
}

--
-- Ответ, содержащий запрашиваемые записи; в зависимости от установленных
-- запретов агента, ответ может содержать только часть запрашиваемых
-- записей; данная ситуация должна проверяться администратором
--
GetEventLogEntryResult ::= SEQUENCE {
    log-change-count                INT-U16,        -- счетчик текущего количества
                                                    -- изменений журнала (0 если
                                                    -- не поддерживается)
    from-log-entry-index            INT-U32,
    to-log-entry-index              INT-U32,
    entry-list                      EventLogEntryList
}
```

7.5.8.3 Уведомления

Объект Event Log не формирует никакие конкретные уведомления.

7.5.9 Объект Battery (Батарея)

Объект: Battery

Описание: Для приборов с батарейным питанием, некоторая часть информации о батарее включается в состав объекта класса MDS в форме соответствующих атрибутов. Для случаев, когда батарейная подсистема способна предоставлять дополнительную информацию (смарт-батарея) или некоторым способом управляться, определен специальный класс Battery.

Производное от: Top (Главный объект)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_BATT

7.5.9.1 Атрибуты

Класс объекта Battery определяет атрибуты, представленные в таблице 7.45.

Таблица 7.45 — Атрибуты класса объекта Battery

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_ HANDLE	HANDLE	Атрибут связывания имен	M
Battery-Status	MDC_ATTR_BATT_ STAT	BatteryStatus		M
Production-Specification	MDC_ATTR_ID_ PROD_SPECN	ProductionSpec	Система смарт-батареи может иметь серийный номер или версию	O
Capacity-Remaining	MDC_ATTR_CAPAC_ BATT_REMAIN	BatMeasure	Остаточная емкость при текущей нагрузке (например, в milliAmperehours)	O
Capacity-Full-Charge	MDC_ATTR_CAPAC_ BATT_FULL	BatMeasure	Емкость батареи после полной зарядки	O
Capacity-Specified	MDC_ATTR_CAPAC_ BATT_SPECN	BatMeasure	Заданная емкость новой батареи	O
Remaining-Battery-Time	MDC_ATTR_TIME_ BATT_REMAIN	BatMeasure		O
Voltage	MDC_ATTR_BATT_ VOLTAGE	BatMeasure	Настоящее напряжение батареи	O
Voltage-Specified	MDC_ATTR_BATT_ VOLTAGE_SPECN	BatMeasure	Заданное напряжение батареи	O
Current	MDC_ATTR_BATT_ CURR	BatMeasure	Текущий ток, подаваемый батареей/к батареи; отрицательный, если батарея заряжена	O
Battery-Temperature	MDC_ATTR_TEMP_ BATT	BatMeasure		O
Charge-Cycles	MDC_ATTR_BATT_ CHARGE_CYCLES	INT-U32	Число циклов зарядки/разрядки	O

В таблице 7.46 класс объекта Battery описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.46 — Группы атрибутов класса объекта Battery

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Батареи	MDC_ATTR_GRP_BATT	из Battery: (все)

Применяют следующие определения типов::

--

```
-- Битовое поле Battery Status (Статус батареи)
--
BatteryStatus ::= BITS-16 {
    batt-discharged (0),
    batt-full (1),                          -- > 95% емкости
    batt-discharging (2),
    batt-chargingFull (8),
    batt-chargingTrickle (9),
    batt-malfunction (12),
    batt-needs-conditioning (13)            -- батарее требуется тренировка
}
--
-- Все измерения, касающиеся батареи, являются значениями с их
-- размерностями
--
BatMeasure ::= SEQUENCE {
    value           FLOAT-Type,
    unit        OID-Type                    -- из раздела номенклатуры размерностей
```

7.5.9.2 Поведение

Объект Battery не определяет никакие конкретные методы.

7.5.9.3 Уведомления

Объект Battery не формирует никакие конкретные уведомления.

7.5.10 Объект Clock (Часы)

Объект: Clock

Описание: Класс Clock предоставляет дополнительные возможности для обработки информации, связанной с датой и временем по сравнению с базовыми возможностями, предоставляемыми классом MDS. Объект Clock не требует никакое специальное аппаратное или программное обеспечение.

Производное от: Top (Главный объект)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_CLOCK

7.5.10.1 Атрибуты

Класс объекта Clock определяет атрибуты, представленные в таблице 7.47.

Таблица 7.47 — Атрибуты класса объекта Clock

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
Time-Support	MDC_ATTR_TIME_SUPPORT	TimeSupport	Указывает сервисы времени, обеспечиваемые прибором	M
Date-Time-Status	MDC_ATTR_DATE_TIME_STATUS	DateTimeStatus	Общая информация о работе сервисов поддержки времени. Обязательно, если прибор поддерживает удаленные сервисы синхронизации [например, простой сетевой протокол синхронизации времени (SNTP)]; в других случаях не обязательно	C

Окончание таблицы 7.47

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Date-and-Time	MDC_ATTR_TIME_ABS	AbsoluteTime	Текущие настройки даты/времени	O
ISO-Date-and-Time	MDC_ATTR_TIME_ABS_ISO	AbsoluteTimeISO	Строка даты и времени отформатирована согласно ИСО 8601; предусмотрена для синхронизации скоординированного всемирного времени (UTC). Атрибут широко используется вычислительными системами; однако он основан на ASCII и поэтому менее эффективен, чем абсолютное время	O
Relative-Time	MDC_ATTR_TIME_REL	RelativeTime	Относительное время (метка в 8 кГц)	O
HiRes-Relative-Time	MDC_ATTR_TIME_REL_HI_RES	HighRes-RelativeTime	Относительное время высокого разрешения (отметка в 1 МГц)	O
Ext-Time-Stamp-List	MDC_ATTR_TIME_STAMP_LIST_EXT	ExtTimeStampList	Расширенные метки времени (которые можно использовать отдельно в любом другом месте в структуре данных)	O
Absolute-Relative-Sync	MDC_ATTR_TIME_ABS_REL_SYNC	Absolute-RelativeTimeSync	Обеспечивает способ корреляции между значениями абсолютного времени и относительного времени[a)]	O
Time-Zone	MDC_ATTR_TIME_ZONE	UTCTimeZone	Определяет смещение относительного местного часового пояса UTC [относительно времени по Гринвичу (GMT)] и этикетку (маркировку)	O
Daylight-Savings-Transition	MDC_ATTR_TIME_DAYLIGHT_SAVINGS_TRANS	Daylight-SavingsTransition	Обеспечивает настройки для перехода на следующее летнее/зимнее время	O
Cumulative-Leap-Seconds	MDC_ATTR_CUM_LEAP_SECONDS	INT-U32	Общее количество потерянных секунд по отношению к 1 января 1900 г., 00:00:00.00. Формат — +nn. За весь 2011 год это значение составит +32[b)]	O
Next-Leap-Seconds	MDC_ATTR_NEXT_LEAP_SECOND	LeapSeconds-Transition	Указывает настройки для следующего перехода на следующие потерянные секунды и следующее значение	O

[a)] Данный атрибут время от времени обновляется внутри программы (например, один раз за минуту) и поэтому не отображает текущее время при считывании (например, используя сервис GET). Погрешность между относительным и абсолютным временем должна быть наименьшей, устанавливая системные ограничения (например, по возможности должна использоваться элементарная операция). Атрибут должен постоянно обновляться, чтобы уменьшить погрешность между сообщаемыми преобразованиями и должен обновляться минимум раз в 6 дней, а именно, когда относительное время переваливает за 0.

[b)] При вычитании из секунд SNTP выдает секунды UTC.

Класс объекта Clock описывает в таблице 7.48 группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.48 — Группы атрибутов класса объекта Clock

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Часы	MDC_ATTR_GRP_CLOCK	из Clock: (все)

Применяют следующие определения типов:

```
--
-- Атрибут Time-Support (Поддержка времени) дает общую информацию о
-- сервисах, связанных со временем, обеспечиваемых прибором. Некоторую
-- часть этой информации можно определить путем определения
-- наличия/отсутствия различных атрибутов в дереве состава; однако его
-- присутствие здесь упрощает управление временем для менеджмеров прибора
--
-- Примечания
-- 1 — Если поддерживается удаленная синхронизация даты/времени (например, -- SNTP), то должен
также поддерживаться либо Атрибут Date-And-Time (дата и -- время) либо ISO-Date-And-Time (дата и
время согласно ISO).
-- 2 — Если прибор также является сервером информации времени (например,
-- сервер SNTP), это должно быть указано в идентификаторах протокола времени.
--
TimeSupport ::= SEQUENCE {
    time-capability         TimeCapability,         -- флажки, указывающие
                                                    -- поддержку общего времени
    relative-resolution     INT-U32,                -- время между фактическими
                                                    -- тактами в микросекундах;
                                                    -- устанавливают на
                                                    -- 0xFFFFFFFF, если не указано
                                                    -- иное
    time-protocols          SEQUENCE OF             -- перечень поддерживаемых
                            TimeProtocolId          -- внешних протоколов времени
                                                    -- (например, SNTP)
}
```

Примечание — Тип relative-resolution (относительное расширение) связывает частоту в 8кГц, сообщенную посредством значения относительного времени, с источником времени прибора, из которого его получают. Например, если таймер прибора обновляется при 100 Гц или 18.2 Гц [так было в устаревших моделях персональных компьютеров (ПК)], то разрешение и погрешность относительного времени будет отражать разрешение и погрешность данного источника времени.

```
--
-- Возможности поддержки времени
--
TimeCapability ::= BITS-32 {
    time-capab-real-time-clock (0),     -- прибор предоставляет аппаратную
                                        -- поддержку времени (в том числе
                                        -- питание от батареи)
    time-capab-ebww (1),                -- время можно установить
                                        -- локально/вручную (глазное яблоко
                                        -- или наручные часы или EBWW)
    time-capab-leap-second-aware (2),   -- поддерживает регулировку времени
                                        -- для потерянных секунд (связано с SNTP)
    time-capab-time-zone-aware (3),     -- поддерживает атрибуты, связанные
                                        -- с временными поясами
    time-capab-internal-only (4),       -- дата/время используется только
                                        -- внутри прибора и не отображается
                                        -- оператору
```

```
    time-capab-time-displayed (5),          -- дата/время может беспрерывно
                                            -- отображаться на приборе в отличии
                                            -- от меню
    time-capab-patient-care (6),            -- дата/время используется в
                                            -- алгоритмах/протоколах ухода за
                                            -- критическим больным
    time-capab-rtsa-time-sync-annotations (7),  -- комментарии к временным меткам
                                            -- поддерживаются для данных
                                            -- осциллограммы реального времени
                                            -- (объекты Real Time Sample Array)
    time-capab-rtsa-time-sync-high-precision (8),  -- Объекты Real Time Sample Array
                                            -- поддерживают атрибуты для
                                            -- высокоточных временных меток
                                            -- выборок
    time-capab-set-time-action-sup (16),    -- Объект Clock поддерживает
                                            -- действие установки времени (set
                                            -- time)
    time-capab-set-time-zone-action-sup (17),  -- Объект Clock поддерживает
                                            -- действие задания временного пояса
                                            -- (set time zone)
    time-capab-set-leap-sec-action-sup (18),  -- Объект Clock поддерживает
                                            -- действие задания потерянных
                                            -- секунд (set leap seconds)
    time-capab-set-time-iso-sup (19)        -- Объект Clock поддерживает
                                            -- действие ISO для задания времени
}

--
-- Идентификатор протокола времени обозначает протоколы времени, которые
-- поддерживаются/используются прибором
--
TimeProtocolId ::= OID-Type     -- из раздела номенклатуры инфраструктуры

--
-- Идентификатор временных отметок (например, для временных меток SNTP)
--
TimeStampId ::= OID-Type        -- из раздела номенклатуры инфраструктуры

--
-- Расширенная временная метка (например, значение временной метки SNTP)
--
ExtTimeStamp ::= SEQUENCE {
    time-stamp-id               TimeStampId,
    time-stamp                  ANY DEFINED BY time-stamp-id
}

ExtTimeStampList ::= SEQUENCE OF ExtTimeStamp

--
-- Атрибут Date-Time-Status (статус даты/времени) определяет статус
-- текущего/активного использования даты и времени в приборе
--
DateTimeStatus ::= SEQUENCE {
```

```
    usage-status            DateTimeUsage,      -- флажки, указывающие на
                                                -- использование динамического
                                                -- времени
    clock-last-set          AbsoluteTime,       -- время, когда было последний
                                                -- раз установлено абсолютное
                                                -- время
    clock-accuracy          FLOAT-Type,         -- десятичное число, указывающее
                                                -- точность или максимальную
                                                -- погрешность абсолютного
                                                -- времени по отношению к
                                                -- источнику опорных первичных
                                                -- тактовых сигналов (в секундах)
    active-sync-protocol    TimeProtocolId      -- протокол, который активно
                                                -- используется для синхронизации
                                                -- времени
}
```

Примечания

1 Если используется протокол синхронизации времени, который меняет время и дату с большой частотой, значение типа clock-last-set должно обновляться не так часто (например, каждые 10 минут или раз в час), так чтобы пропускная способность канала связи не использовалась напрасно.

2 В системах, где используется синхронизация времени (например, EBWW — источник), тип clock-accuracy должен запускаться за 2—3 минуты, при условии, что время уже установлено, и должен время от времени увеличиваться, чтобы отражать отклонение от абсолютного внешнего источника. Если используется протокол NTP (сетевой протокол службы времени), то значение запуска типа clock-accuracy эквивалентно: Корневая дисперсия + ½ Корневая задержка.

```
--
-- Флажки использования даты/времени обозначают динамический статус
-- использования даты и времени в приборе; если не установлены никакие
-- биты, то статус неизвестный/неопределенный
--
DateTimeUsage ::= BITS-16 {
    dt-use-remote-sync (0),             -- дата/время синхронизируется с
                                        -- внешним источником
    dt-use-operator-set (1),            -- настройку даты/времени
                                        -- производит оператор
                                        -- (т. е. EBWW)
    dt-use-rtc-synced (2),              -- дата/время в RTC были
                                        -- синхронизированы с удаленным
                                        -- источником времени
    dt-use-critical-use (3),            -- дата/время активно
                                        -- используются в
                                        -- алгоритмах/протоколах
                                        -- проведения лечения
    dt-use-displayed (4)                -- дата/время отображается на
                                        -- экране для оператора

--
-- Атрибут ISO-Date-and-Time — это строка ASCII, которая может дать
-- дополнительную информацию по настройкам даты/времени (например, смещение
-- относительно времени по Гринвичу или указание часового пояса, где
-- находится прибор); данный атрибут можно настроить с помощью сервиса SET
-- (как и с помощью атрибута Date-And-Time)
```

```
-- Обращаем Ваше внимание на то, что если параллельно поддерживаются оба
-- типа AbsoluteTime и AbsoluteTimeIS, они должны отражать то же время
-- (по отношению к их погрешности и ограничению по разрешающей способности)
-- Несмотря на то, что это не обязательно, очень рекомендуется включить все
-- необязательные поля в строку Для упрощения процесса обработки должны
-- применяться следующие условия
-- (a) Должны использоваться только полные представления
-- (b) используются только расширенные форматы
-- (c) Представление Дата недели и обычное представление дата в году не
-- используются; только календарные даты
-- (d) Десятичные дроби должны использоваться только для секундных долей
-- (например, не для долей часа)
-- (e) Согласно ISO 8601:2000(E), представление десятичных дробей должно
-- быть в соответствии с разделом 5.3.1.3
-- (f) Если известно, время UTC должно передаваться с помощью формата zulu
-- (z или время по Гринвичу) или с помощью смещения между местным временем
-- и временем GMT/UTC; указание смещения по времени используется только,
-- если оно известно
-- (g) Определение интервалов времени и периодов повторения не
-- распространяется на данный тип данных и требует определения нового типа
-- данных, в случае его использования (например, ISOTimeInterval ::= OCTET
-- STRING); например: 24 ноября 2001, 15:45:32.65 в Сан-Диего,
-- Калифорния, США, должно представляться в следующем виде: 2001-11-
-- 24T15:45:32,65-08:00
--
AbsoluteTimeISO ::= OCTET STRING                -- текстовая строка кода ASCII,
                                                -- которая связана с форматом
                                                -- ISO 8601

--
-- SNTPTimeStamp, значение 64-битной временной метки, которое
-- обеспечивается сервисом синхронизации времени SNTP
--
SNTPTimeStamp ::= SEQUENCE {
    seconds                 INT-U32,            -- секунды с 1 января 1900
                                                -- 00:00
    fraction                INT-U32             -- двоичная дробь секунды
}

--
-- Атрибут Absolute-Relative-Sync обеспечивает средства для корреляции
-- относительных временных меток с настройкой даты/времени прибора
-- Примечание — Данный атрибут необходимо время от времени обновлять для
-- учета отклонения между различными источниками времени (например, раз в
-- минуту)
--
AbsoluteRelativeTimeSync ::= SEQUENCE {
    absolute-time-mark      AbsoluteTime,       -- использование этого типа
                                                -- данных ограничивает
                                                -- разрешение до 1/100 секунды
    relative-time-mark      RelativeTime,       -- разрешение ограничено
                                                -- отметкой в 125 мкс и
                                                -- настройками
                                                -- разрешения/погрешности для
                                                -- сервиса относительного
                                                -- времени
```

```
    relative-rollovers          INT-U16,            -- количество раз относительное
                                                    -- время переваливало со
                                                    -- своего максимального
                                                    -- значения на 0
                                                    -- Примечание — Относительное
                                                    -- время будет переваливать
                                                    -- каждые 6.2 дня
    hires-time-mark             HighResRelativeTime, -- сброс настроек на
                                                    -- 0x00000000, если не
                                                    -- поддерживается
    ext-time-marks              ExtTimeStampList    -- список пуст, если не
                                                    -- поддерживается ни одна
                                                    -- расширенная временная метка
}

--
-- Атрибут Time-Zone (часовой пояс) поддерживает информацию о поясе для
-- UTC
--
UTCTimeZone ::= SEQUENCE {
    time-zone-offset-hours      INT-I8,             -- местный часовой пояс прибора
                                                    -- (т. е. в точке оказания
                                                    -- лечения) по отношению к UTC
                                                    -- формату: +чч для часовых
                                                    -- поясов к востоку от
                                                    -- GMT и -чч для поясов к
                                                    -- западу от GMT
    time-zone-offset-minutes    INT-U8,             -- смещение минут относительно
                                                    -- GMT (если указано); условное
                                                    -- обозначение формата такое же
                                                    -- как и для часов, только они
                                                    -- не имеют знака (должны
                                                    -- всегда иметь положительное
                                                    -- значение); по умолчанию —
                                                    -- НОЛЬ (NULL)
    time-zone-label             OCTET STRING        -- этикетка (маркировка)
                                                    -- местного часового пояса
                                                    -- прибора, например, PST или
                                                    -- PDT; смотрите атрибут
                                                    -- прибора Region для
                                                    -- кодирования строки
}

--
-- Атрибут Daylight-Savings-Transition (переход на зимнее/летнее время)
-- обозначает настройки для следующего перехода на зимнее/летнее время
--
DaylightSavingsTransition::= SEQUENCE {
    transition-date             AbsoluteTime,       -- местная дата/время на
                                                    -- приборе при переходе на
                                                    -- зимнее/летнее время
```

```
    next-offset                 UTCTimeZone         -- смещение нового местного
                                                    -- часового пояса и этикетка
                                                    -- (маркировка) после даты
                                                    -- перехода. Примечание —
                                                    -- Может быть таким же как и
                                                    -- предыдущее значение.
}

--
-- Атрибут Next-Leap-Seconds (Следующие потерянные секунды) определяет настройки для перехода
на следующие потерянные секунды
--
LeapSecondsTransition::= SEQUENCE {
    transition-date             Date,               -- местная дата прибора, когда
                                                    -- происходит переход;
                                                    -- настройка происходит в конце
                                                    -- (т. е. 23:59:59Z) указанной
                                                    -- даты
    next-cum-leap-seconds       INT-U32             -- следующее суммарное
                                                    -- значение потерянных секунд
                                                    -- (смотрите Cumulative-Leap-
                                                    -- Seconds в таблице 7.47)
                                                    -- Примечание — может быть
                                                    -- таким же, что и предыдущее
                                                    -- значение
}
```

7.5.10.2 Поведение

Объект Clock определяет методы, описанные в таблице 7.49.

Таблица 7.49 — Методы объекта Часы

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Set-Time	Подтверждено	MDC_ACT_SET_TIME	SetTimeInvoke	Нет
Set-Time-Zone	Подтверждено	MDC_ACT_SET_TIME_ZONE	SetTimeZoneInvoke	Нет
Set-Leap-Seconds	Подтверждено	MDC_ACT_SET_LEAP_SECONDS	SetLeapSeconds-Invoke	Нет
Set-Time-ISO	Подтверждено	MDC_ACT_SET_TIME_ISO	AbsoluteTimeISO	Нет

Примечание — При настройке времени с помощью Set-Time или Set-Time-ISO, все поддерживаемые атрибуты абсолютных временный меток (т. е. Date-and-Time, ISO-Date-and-Time и возможно Ext-Time-Stamp-List) должны обновляться в обязательном порядке.

Применяют следующие типы данных:

```
--
-- Настройка даты/времени
--
SetTimeInvoke ::= SEQUENCE {
    date-time                   AbsoluteTime,
    accuracy                    FLOAT-Type          -- рассчитывается для времени,
                                                    -- настроенного вручную
                                                    -- (например, погрешность 2
                                                    -- минуты); значение указано в
                                                    -- секундах
```

```
}

--
-- Настройка информации о часовом поясе
--
SetTimeZoneInvoke ::= SEQUENCE {
    time-zone           UTCTimeZone,            -- текущий часовой пояс,
                                                -- используемый прибором
    next-time-zone      DaylightSavingsTransition -- информация для
                                                -- следующего перехода на
                                                -- зимнее/летнее время
}

--
-- Сводная информация об потерянных секундах
--
SetLeapSecondsInvoke ::= SEQUENCE {
    leap-seconds-cum    INT-I32,                -- потерянные секунды в
                                                -- совокупности при вычитании
                                                -- секунд SNTP из секунд UTC
    next-leap-seconds   LeapSecondsTransition   -- дата перехода с
                                                -- предыдущего на новое
                                                -- суммарное значение
                                                -- истекших секунд + новое
                                                -- значение
}
```

7.5.10.3 Уведомления

Объект Журнал событий описывает события в таблице 7.50.

Таблица 7.50 — События объекта Часы

Событие	Режим	Идентификатор события	Параметр события	Результат события
Clock-Date-Time-Status-Changed	Неподтверждено	MDC_NOTI_DATE_TIME_CHANGED	ClockStatus-UpdateInfo	—

Применяют следующие типы данных:

```
--
-- Информация об обновление статуса часов отправляется, когда, например,
-- настройка относительного времени переваливает за 0 или когда оператор
-- прибора изменяет время
--
ClockStatusUpdateInfo ::= SEQUENCE {
    date-time-status    DateTimeStatus,         -- текущий статус
                                                -- использования
                                                -- часов/времени
    time-sync           AbsoluteRelativeTimeSync -- текущие значения
                                                -- синхронизации времени
}
```

7.6 Объекты в Пакете управления

Определения объектов в Пакете управления указаны в пунктах 7.6.1—7.6.9.

7.6.1 Service-and-Control Object (Объект, предоставляющий сервис и управление, SCO)

Объект:	SCO
Описание:	Класс SCO отвечает за использование всех возможностей дистанционного управления, которые поддерживаются медицинским прибором. Объект SCO является точкой первичного доступа для активации функций дистанционного контроля. Он включает в себя все объекты Operation и обеспечивает обработку транзакций. Все команды вызова объекта Operation выполняются с помощью объекта SCO.
Производное от:	VMO
Связывание имен:	Handle (Описатель) (унаследованный объект VMO)
Зарегистрирован как:	MDC_MOC_CNTRL_SCO

7.6.1.1 Атрибуты

Класс объекта SCO определяет атрибуты, представленные в таблице 7.51.

Таблица 7.51 — Атрибуты класса объекта SCO

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Sco-Capability	MDC_ATTR_SCO_CAPAB	ScoCapability	Поле флажка статической опции	M
Sco-Help-Text-String	MDC_ATTR_SCO_HELP_TEXT_STRING	OCTET STRING	Текст справки	O
Vmo-Reference	MDC_ATTR_VMO_REF	HANDLE	Ссылка на управляемый объект, если это не VMD	O
Activity-Indicator	MDC_ATTR_INDIC_ACTIV	ScoActivity-Indicator	Может быть установлен удаленной системой, чтобы показать, что система управляется дистанционно	O
Lock-State	MDC_ATTR_STAT_LOCK	Administrative-State	Если заблокированное (locked) состояние, нельзя вызвать никакую операцию	M
Invoke-Cookie	MDC_ATTR_ID_INVOK_COOKIE	INT-U32	Идентификатор входного сообщения, назначаемый командой вызова	M

В таблице 7.52 класс объекта SCO описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.52 — Группы атрибутов класса объекта SCO

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_STATIC	из VMO: Type, Handle из SCO: Sco-Help-Text-String, Sco-Capability
Динамическая контекстная группа объекта VMO (расширяемая группа атрибутов)	MDC_ATTR_GRP_VMO_DYN	из VMO: Label-String из SCO: Activity-Indicator, Vmo-Reference

Окончание таблицы 7.52

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа входных сообщений объекта SCO	MDC_ATTR_GRP_SCO_ TRANSACTION	из SCO: Lock-State, Invoke-Cookie

Применяют следующие определения типов:

```
--
-- Атрибут Activity-Indicator (Индикатор активности) можно установить с
-- помощью удаленной системы, чтобы показать, что дистанционное управление
-- активно
--
ScoActivityIndicator ::= INT-U16 {
    act-ind-off(0),
    act-ind-on(1),
    act-ind-blinking(2)
}

--
-- Биты Sco-Capability (Функциональные возможности объекта SCO)
--
ScoCapability ::= BITS-16 {
    act-indicator (0),              -- поддерживает индикатор активности
    sco-locks (1),                  -- минимум одна операция устанавливает
                                    -- флажок блокировки SCO
    sco-ctxt-help (8)               -- SCO поддерживает контекстно-зависимую
                                    -- динамическую справку
}
```

7.6.1.2 Поведение

В дополнении к сервису SET, который можно использовать для изменения атрибута Activity-Indicator (Индикатор активности), в таблице 7.53 определены методы объекта SCO.

Таблица 7.53 — Методы объекта SCO

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Operation-Invoke	Подтверждено	MDC_ACT_SCO_OP_ INVOKE	OperationInvoke	Operation-InvokeResult
Get-Ctxt-Help	Подтверждено	MDC_ACT_GET_CTXT_HELP	CtxtHelpRequest	CtxtHelpResult

Применяют следующие типы данных:

```
--
-- Метод Operation-Invoke (Вызов операции) имеет дополнительный механизм
-- обеспечения защиты
--
OperationInvoke ::= SEQUENCE {
    checksum                INT-I16,        -- 16-битное дополнение до двух
    invoke-cookie           INT-U32,        -- произвольно выбранный
                                            -- идентификатор, дублируемый в
                                            -- получившихся обновлениях
    op-elem-list            OpInvokeList

}
```

Примечание — Если не используется контрольное суммирование, то поле контрольной суммы будет иметь значение 0. Если подсчитанная контрольная сумма равна 0, то поле контрольной суммы будет иметь значение 1. Расчет контрольной суммы — это 16-тибитная сумма, полученная в дополнительном двоичном коде для 16-тибитных слов в сообщении, начинающемся в поле адреса после поля контрольной суммы.

```
--
OpInvokeList ::= SEQUENCE OF OpInvokeElement
OpInvokeElement ::= SEQUENCE {
    op-class-id                 OID-Type,           -- из раздела объекто-
                                                    -- ориентированной номенклатуры
    op-instance-no              InstNumber,
    op-mod-type                 OpModType,
    attributes                  AttributeList
}

OpModType ::= INT-U16 {
    op-replace (0),                                 -- обычно заменяет значение
                                                    -- виртуального атрибута
    op-setToDefault (3),                            -- установлено на значение по
                                                    -- умолчанию, если
                                                    -- поддерживается
    op-invokeAction (10),                           -- необходимо для отдельного типа действия
    op-invokeActionWithArgs (15)                    -- работа с параметрами
}

--
-- Результат подтверждает получение (и выполнение) операций
-- Обновленные атрибуты передаются обычным методом обновления (например,
-- сканер) во избежании несоответствия
--
OperationInvokeResult ::= SEQUENCE {
    invoke-cookie               INT-U32,
    result                      OpInvResult
}

OpInvResult ::= INT-U16 {
    op-successful (0),
    op-failure (1)
}

--
-- Нижеуказанные типы позволяют осуществить возврат динамической справочной
-- информации, которая зависит от контекста объекта SCO или объекта
-- Operation (т. е. зависит от состояния)
--
CtxtHelpRequest ::= SEQUENCE {
    type                        OID-Type,           -- ID класса объекта Operation
                                                    -- или ID класса объекта SCO
    op-instance-no              InstNumber          -- номер экземпляра операции (0,
                                                    -- если адресуется объект SCO)
}

CtxtHelpResult ::= SEQUENCE {
```

```
    type                    OID-Type,           -- ID класса объекта Operation
                                                -- или ID класса объекта SCO
    op-instance-no          InstNumber
    hold-time               RelativeTime,       -- время отображения справки; 0, если не
                                                   применимо
    help                    CtxtHelp
}

CtxtHelp ::= CHOICE {
    text-string [1]         OCTET STRING,
    oid [8]                 OID-Type
}
```

7.6.1.3 Уведомления

Объект SCO определяет события, описанные в таблице 7.54.

Таблица 7.54 — События объекта SCO

Событие	Режим	Идентификатор события	Характеристика события	Результат события
SCO-Operating-Request	Подтвержден/ неподтвержден	MDC_NOTI_SCO_OP_ REQ	ScoOperReqSpec (дополнительно)	—
SCO-Operation-Invoke-Error	Подтвержден/ неподтвержден	MDC_NOTI_SCO_OP_ INVOK_ERR	ScoOperInvoke-Error	—

Применяют следующие типы данных:

```
--
-- Функциональный запрос может присоединять в конце дополнительную информацию
--
ScoOperReqSpec ::= SEQUENCE {
    op-req-id               PrivateOid,         -- определяется прибором или
                                                -- производителем
    op-req-info             ANY DEFINED BY op-req-id
}

--
-- Уведомление SCO-Operation-Invoke-Error (Ошибка вызова операции объекта
-- SCO)
--
ScoOperInvokeError ::= SEQUENCE {
    invoke-cookie           INT-U32,
    op-error                INT-U16 {
        op-err-unspec (0),
        checksum-error (1),
        sco-lock-violation (2),
        unknown-operation (3),
        invalid-value (4),
        invalid-mod-type (5)
    },
    failed-operation-list   SEQUENCE OF InstNumber
}
```

7.6.2 Объект Operation (Операция)

Объект:	Operation
Описание:	Класс Operation — абстрактный базовый класс для классов, которые описывают удаленно-управляемые элементы.
Производное от:	Top (Главный объект)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа; уникален в рамках одного экземпляра объекта SCO)
Зарегистрирован как:	MDC_MOC_CNTRL_OP

7.6.2.1 Атрибуты

Класс объекта Operation определяет атрибуты, представленные в таблице 7.55.

Таблица 7.55 — Атрибуты класса объекта Operation

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Instance-Number	MDC_ATTR_ID_INSTNO	InstNumber	Уникален в рамках объекта SCO для идентификации операции	M
Operation-Spec	MDC_ATTR_OP_SPEC	OperSpec	Структура, определяющая типы операции и свойства	M
Operation-Text-Strings	MDC_ATTR_OP_TEXT_STRING\	OperTextStrings	Статическое описание операции	O
Operation-Text-Strings-Dyn	MDC_ATTR_OP_TEXT_STRING_DYN	OperTextStrings	Динамическое описание операции	O
Vmo-Reference	MDC_ATTR_VMO_REF	HANDLE	Ссылка на объект	O
Operational-State	MDC_ATTR_OP_STAT	OperationalState	Определяет, доступна ли операция	O

В таблице 7.56 класс объекта Operation определяет группы атрибутов или расширения для наследования групп атрибутов.

Таблица 7.56 — Группы атрибутов класса объекта Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_DYN_CTXT	из Operation: Operational-State, Vmo-Reference

Применяют следующие определения типов:

```
--
-- Атрибут Operation-Spec (Технические характеристики операции) указывают
-- на то, что действительно осуществляет данная операция
--
OperSpec ::= SEQUENCE {
```

```
    vattr-id                OID-Type,       -- ID виртуального атрибута, который
                                            -- изменяется операцией
    op-target               OID-Type,       -- из метрического или объектно-
                                            -- ориентированного раздела
                                            -- номенклатуры
    options                 OpOptions,      -- специальные опции
    level                   OpLevel,        -- диапазон разделения по степени
                                            -- важности
    grouping                OpGrouping      -- для описания отношений между
                                            -- операциями
}
```

Примечание — Код vattr-id берется из раздела номенклатуры виртуального атрибута. Вводимые данные (т. е. коды) в данном разделе имеют четное значение. Последний бит кода используется для определения того, из какого раздела номенклатуры берется код op-target. Если последний бит равен 0, код op-target берется из метрического раздела номенклатуры. Если последний бит равен 1 (1 добавляется к базовому коду в номенклатуре виртуального атрибута), то код op-target берется из раздела объектно-ориентированной номенклатуры.

```
--
-- Тестовые значения объекта Operation
--
OperTextStrings ::= SEQUENCE {
    label                   OCTET STRING,   -- строка метки (маркировки)
                                            -- указывает значение операции
    help                    OCTET STRING,   -- строка справки может содержать
                                            -- дополнительную справку для
                                            -- пользователя
    confirm                 OCTET STRING    -- строка подтверждения, которую
                                            -- менеджмер демонстрирует
                                            -- пользователю для повторного
                                            -- подтверждения операции
                                            -- (например, Вы действительно
                                            -- хотите завершить работу?)
}

--
-- Опции объекта Operation
--
OpOptions ::= BITS-16 {
    needs-confirmation (0),
    supports-default (1),                   -- для виртуальных атрибутов
                                            -- поддерживается значение по умолчанию
    sets-sco-lock (2),                      -- требует обработки транзакций,
                                            -- чтобы избежать побочных эффектов
    is-setting (3),                         -- значение, сохраненное при
                                            -- отсутствия питания в системе
    op-dependency (6),                      -- операция взаимозависит от
                                            -- остальных (всегда установлено,
                                            -- если установлен бит sets-sco lock)
    op-auto-repeat (7),                     -- поддержка автоповтора
    op-ctxt-help (8)                        -- обеспечивает справку в
                                            -- зависимости от контекста
                                            -- посредством действия SCO
}
--
-- Уровень
```

```
--
OpLevel ::= BITS-16 {
    op-level-basic (0),                    -- обычная операция
    op-level-advanced (1),                 -- усовершенствованная операция
    op-level-professional (2),
    op-item-normal (8),                    -- операция модифицирует обычного
                                           -- элемент пользователя
    op-item-config (9),                    -- операция модифицирует элемент
                                           -- конфигурации
    op-item-service (10)                   -- операция модифицирует сервисный
                                           -- элемент (не реализуется обычным
                                           -- оператором)
}

--
-- Поле для операций группирования (т.е. определяет логические связи); на
-- интерфейсе оператора (т. е. дисплее) можно использовать для организации
-- операций в необходимой последовательности
--
OpGrouping ::= SEQUENCE {
    group                  INT-U8,
    priority               INT-U8
}
```

7.6.2.2 Поведение

Объект Operation не определяет никакие конкретные методы.

7.6.2.3 Уведомления

Объект Operation не формирует никакие конкретные уведомления.

7.6.3 Объект Select Item Operation (Операция выбора элемента из списка)

Объект:	Select Item Operation
Описание:	Класс Select Item Operation позволяет осуществить выбор одного элемента из данного списка. Список может иметь различные типы.
Производное от:	Operation (Операция)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)
Зарегистрирован как:	MDC_MOC_CNTRL_OP_SEL_IT

7.6.3.1 Атрибуты

Класс объекта Select Item Operation определяет атрибуты, описанные в таблице 7.57.

Таблица 7.57 — Атрибуты класса объекта Select Item Operation

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Selected-Item-Index	MDC_ATTR_INDEX_SEL	INT-U16	Индекс текущего выбираемого элемента	M
Nom-Partition	MDC_ATTR_ID_NOM_PARTITION	NomPartition	Если записи в списке являются идентификаторами OID, то указывает используемый раздел номенклатуры	C
Select-List	MDC_ATTR_LIST_SEL	SelectList	Список возможных варинтов выбора	M

В таблице 7.58 класс объекта Select Item Operation определяет группы атрибутов или расширения для унаследованных групп атрибутов.

Таблица 7.58 — Группы атрибутов класса объекта Select Item Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts из Select Item Operation : Nom-Partition
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Select Item Operation: Selected-Item-Index, Select-List

Применяют следующие определения типов:

```
--
-- Атрибуты Select-List (Список выбранных элементов) определяют действующие
-- выбранные элементы
--
SelectList ::= CHOICE {
    oid-list                [1] SEQUENCE OF OID-Type,
    value-list              [3] SEQUENCE OF FLOAT-Type,
    value-u-list            [4] SEQUENCE OF SelectUValueEntry,
    string-list             [5] SEQUENCE OF OCTET STRING
}

--
-- Значение с кодом единицы измерения/размера
--
SelectUValueEntry ::= SEQUENCE {
    value                   FLOAT-Type,
    m-units                 OID-Type            -- из раздела номенклатуры
                                                -- размерностей
}
```

7.6.3.2 Поведение

Объект Select Item Operation не определяет никакие конкретные методы.

7.6.3.3 Уведомления

Объект Select Item Operation не формирует никакие конкретные уведомления.

7.6.4 Объект Set Value Operation (Операция установления значения)

Объект:	Set Value Operation
Описание:	Класс Set Value Operation позволяет осуществлять подстройку значения атрибута в пределах заданного диапазона с заданным разрешением.
Производное от:	Operation (Операция)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)
Зарегистрирован как:	MDC_MOC_CNTRL_OP_SEL_VAL

7.6.4.1 Атрибуты

Класс объекта Set Value Operation определяет атрибуты, описанные в таблице 7.59.

Таблица 7.59 — Атрибуты класса объекта Set Value Operation

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Current-Value	MDC_ATTR_VAL_CURR	FLOAT-Type	Текущее значение	M
Set-Value-Range	MDC_ATTR_VAL_RANGE	OpSetValueRange	Диапазон допустимых значений	M
Step-Width	MDC_ATTR_VAL_STEP_WIDTH	OpValStepWidth	Разрешенная ширина шага	O
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	Из раздела номенклатуры размеров	O

В таблице 7.60 класс объекта Set Value Operation определяет группы атрибутов или расширения для унаследованных групп атрибутов.

Таблица 7.60 — Группы атрибутов класса объекта Set Value Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Set Value Operation: Current-Value, Set-Value-Range, Unit-Code, Step-Width

Применяют следующие определения типов:

```
--
-- Атрибут Set-Value-Range (Установка диапазона значения) определяет
-- диапазон и минимальное разрешение
--
OpSetValueRange ::= SEQUENCE {
    minimum                 FLOAT-Type,
    maximum                 FLOAT-Type,
    resolution              FLOAT-Type
}

--
-- Атрибут Step-Width (Ширина шага) является упорядоченным (по возрастанию) массивом диапазонов
и соответствующих минимальных значений ширины шагов; нижней границей является минимальное
значение спецификации диапазона
--
OpValStepWidth ::= SEQUENCE OF StepWidthEntry
StepWidthEntry ::= SEQUENCE {
    upper-edge          FLOAT-Type,
    step-width          FLOAT-Type
}
```

7.6.4.2 Поведение

Объект Set Value Operation не определяет никакие конкретные методы.

7.6.4.3 Уведомления

Объект Set Value Operation не формирует никакие конкретные уведомления.

7.6.5 Объект Set String Operation (Операция задания строки)

Объект:	Set String Operation
Описание:	Класс Set String Operation позволяет системе устанавливать содержание виртуального атрибута строкового типа.
Производное от:	Operation (Операция)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)
Зарегистрирован как:	MDC_MOC_CNTRL_OP_SET_STRING

7.6.5.1 Атрибуты

Класс объекта Set String Operation определяет атрибуты, описанные в таблице 7.61.

Таблица 7.61 — Атрибуты класса объекта Set String Operation

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Current-String	MDC_ATTR_STRIG _CURR	OCTET STRING	Текущее значение виртуального атрибута строкового типа	C[a)]
Set-String-Spec	MDC_ATTR_SET_ STRING_SPEC	SetStringSpec	Свойства виртуального атрибута строкового типа	M
[a)] Атрибут Current-String (Текущая строка) не входит в сферу действия настоящего стандарта, если в атрибуте спецификации установлен флажок setstr-hidden-val; иначе это обязательное условие.				

В таблице 7.62 класс объекта Set String Operation определяет группы атрибутов или расширения для унаследованных групп атрибутов.

Таблица 7.62 — Группы атрибутов класса объекта Set String Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_ STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_ DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Set String Operation: Current-String, Set-String-Spec

Применяют следующие определения типов:

```
--
-- Атрибут Set-String-Spec (Установка строки спецификации)
--
SetStringSpec ::= SEQUENCE {
max-str-len          INT-U16,      -- максимальная поддерживаемая длина строки
char-size            INT-U16,      -- длина знака в битах, например, 7, 8, или 16
set-str-opt          SetStrOpt     -- специальный бит регистра опций
}
--
-- Опции для строки
--
SetStrOpt ::= BITS-16 {
```

```
    setstr-null-terminated (0),     -- строка оканчивается знаком НОЛЬ
    setstr-displayable (1),         -- строка отображается
    setstr-var-length (2),          -- строка имеет различную длину (вплоть до максимальной)
    setstr-hidden-val (3)           -- фактическое содержимое скрыто,
                                    -- например, для ввода пароля
}
```

7.6.5.2 Поведение

Объект Установка строки операции не определяет никакие конкретные методы.

7.6.5.3 Уведомления

Объект Установка строки операции не формирует никакие конкретные уведомления.

7.6.6 Объект Toggle Flag Operation (Операция переключения флажка)

Объект:	Toggle Flag Operation
Описание:	Класс Toggle Flag Operation позволяет осуществлять операцию переключения между двумя альтернативными состояниями (например, вкл/выкл).
Производное от:	Operation (Операция)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)
Зарегистрирован как:	MDC_MOC_CNTRL_OP_TOG

7.6.6.1 Атрибуты

Класс объекта Toggle Flag Operation определяет атрибуты, описанные в таблице 7.63.

Таблица 7.63 — Атрибуты класса объекта Toggle Flag Operation

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Toggle-State	MDC_ATTR_STA_OP_TOG	ToggleState	Текущее состояние выключателя	M
Toggle-Label-Strings	MDC_ATTR_TOG_ LABELS_STRING	ToggleLabel-Strings		M

В таблице 7.64 класс объекта Toggle Flag Operation определяет группы атрибутов или расширения для унаследованных групп атрибутов.

Таблица 7.64 — Группы атрибутов класса объекта Toggle Flag Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_ STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_ DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Toggle Flag Operation: Toggle-State, Toggle-Label-Strings

Применяют следующие определения типов:

```
--
-- Атрибут Toggle-State (состояние переключателя)
--
ToggleState ::= INT-U16 {
```

```
    tog-state0 (0),
    tog-state1 (1)
}

--
-- Каждое состояние имеет этикетку (маркировку)
--
ToggleLabelStrings ::= SEQUENCE {
    lbl-state0                  OCTET STRING,
    lbl-state1                  OCTET STRING
}
```

7.6.6.2 Поведение

Объект Операция по смене флажка не определяет никакие конкретные методы.

7.6.6.3 Уведомления

Объект Операция по смене флажка не формирует никакие конкретные уведомления.

7.6.7 Объект Activate Operation (Операция активации)

Объект:	Activate Operation
Описание:	Класс Activate Operation позволяет начать определенную деятельность (например, обнуление показателя давления).
Производное от:	Operation (Операция)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)
Зарегистрирован как:	MDC_MOC_CNTRL_OP_ACTIV

7.6.7.1 Атрибуты

Класс объекта Activate Operation не определяет никакие дополнительные атрибуты.

В таблице 7.65 данный класс объекта определяет группы атрибутов или расширения для наследуемых групп атрибутов.

Таблица 7.62 — Группы атрибутов класса объекта Activate Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Activate Operation: Current-String, Set-String-Spec

Нет необходимости в дополнительных определениях типа.

7.6.7.2 Поведение

Объект Activate Operation не определяет никакие конкретные методы.

7.6.7.3 Уведомления

Объект Activate Operation не формирует никакие конкретные уведомления.

7.6.8 Объекта Limit Alert Operation (Режим предельного предупреждения)

Объект:	Limit Alert Operation
Описание:	Класс Limit Alert Operation позволяет осуществлять подстройку границ рабочего диапазона датчика аварийных сигналов, а также его включение и выключение на границах установленного диапазона.

Производное от: Operation (Операция)

Связывание имен: Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)

Зарегистрирован как: MDC_MOC_CNTRL_OP_LIM

7.6.8.1 Атрибуты

Класс объекта Limit Alert Operation определяет атрибуты, описанные в таблице 7.66.

Таблица 7.66 — Атрибуты класса объекта Limit Alert Operation

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Alert-Op-Capability	MDC_ATTR_AL_OP_CAPAB	AlOpCapab	Указывает на то, что можно включить или выключить	M
Alert-Op-State	MDC_ATTR_AL_OP_STAT	CurLimAlStat	Текущее состояние: вкл./выкл.; можно настроить методом Operation-Invoke (Вызов операции)	M
Current-Limits	MDC_ATTR_LIMI_CURR	CurLimAlVal	Текущие границы сигнализации; можно настроить методом Operation-Invoke (Вызов операции)	M
Alert-Op-Text-String	MDC_ATTR_AL_OP_TEXT_STRING	AlOpTextString	Отдельный текст для верхнего и нижнего пределов	O
Set-Value-Range	MDC_ATTR_VAL_RANGE	OpSetValueRange	Допустимый диапазон границ	M
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	Размерность значений	M
Metric-Id	MDC_ATTR_ID_PHYSIO	OID-Type	Измерение (т. е. объект Numeric), к которому применима граница, из метрического раздела номенклатуры	M

В таблице 7.67 класс объекта Limit Alert Operation определяет группы атрибутов или расширения для наследования групп атрибутов.

Таблица 7.67 — Группы атрибутов класса объекта Limit Alert Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts из Операции сигнализатора уровня: Alert-Op-Capability, Alert-Op-Text-String
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Операции сигнализатора уровня: Alert-Op-State, Current-Limits, Set-Value-Range, Unit-Code, Metric-Id

Применяют следующие определения типов:

--

```
-- Статические флажки объекта Limit Alert Operation указывают, какие флажки
-- вкл./выкл. поддерживаются
--
AlOpCapab ::= BITS-16 {
    low-limit-sup (1),                  -- поддерживает нижнюю границу
    high-limit-sup (2),                 -- поддерживает верхнюю границу
    auto-limit-sup (5),                 -- поддерживает автоматические
                                        -- границы
    low-lim-on-off-sup (8),             -- поддерживает включение/выключение
                                        -- нижней границы
    high-lim-on-off-sup (9),            -- поддерживает включение/выключение
                                        -- верхней границы
    lim-on-off-sup (10)                 -- поддерживает включение/выключение
                                        -- всей сигнализации
}

--
-- Атрибут Alert-Op-State определяет текущее состояние Limit Alert
-- Примечание — Биты относятся только к объекту Limit Alert, а не к
-- общему аварийному состоянию в метрике
--
CurLimAlStat ::= BITS-16 {
    lim-alert-off (0),                  -- если установлен данный бит, все
                                        -- границы аварийного состояния
                                        -- (верхняя и нижняя) выключены
    lim-low-off (1),                    -- обнаружение нарушения нижней
                                        -- границы выключено
    lim-high-off (2)                    -- обнаружение нарушения верхней
                                        -- границы выключено
}

--
-- Атрибут Current-Limits (Текущие границы)
--
CurLimAlVal ::= SEQUENCE {
    lower                   FLOAT-Type,
    upper                   FLOAT-Type
}

--
-- Атрибут Alert-Op-Text-String (Текстовая строка тревоги) присваивает
-- отдельные этикетки (маркировки) верхней и нижней границы аварийного
-- состояния
--
AlertOpTextString ::= SEQUENCE {
    lower-text              OCTET STRING,
    upper-text              OCTET STRING
}
```

7.6.8.2 Поведение

Объект Limit Alert Operation не определяет никакие конкретные методы.

7.6.8.3 Уведомления

Объект Limit Alert Operation не формирует никакие конкретные уведомления.

7.6.9 Объект Set Range Operation (Операция установления границ диапазона значений)

Объект:	Set Range Operation
Описание:	Объект Set Range Operation позволяет системе настроить верхние и нижние значения (т. е. диапазон значений) в пределах установленных границ.
Производное от:	Operation (Операция)
Связывание имен:	Instance-Number (номер экземпляра) (сервис управления объектами не имеет к нему прямого доступа)
Зарегистрирован как:	MDC_MOC_CNTRL_OP_SET_RANGE

7.6.9.1 Атрибуты

Класс объекта Операция настройки диапазона определяет атрибуты, описанные в таблице 7.68.

Таблица 7.68 — Атрибуты класса объекта Операция по установки диапазона

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Current-Range	MDC_ATTR_RANGE_CURR	CurrentRange	Текущее значение	M
Range-Op-Text	MDC_ATTR_RANGE_OP_TEXT_STRING	RangeOpText	Статический атрибут для определения отдельных текстовых сообщений для верхней и нижней границ	O
Set-Value-Range	MDC_ATTR_VAL_RANGE	OpSetValueRange	Диапазон допустимых значений	M
Step-Width	MDC_ATTR_VAL_STEP_WIDTH	OpValStepWidth	Допустимая ширина шага	O
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	Из раздела номенклатуры размеров	O

В таблице 7.69 класс объекта Set Range Operation определяет группы атрибутов или расширения для унаследованных групп атрибутов.

Таблица 7.69 — Группы атрибутов класса объекта Set Range Operation

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Статическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_STATIC_CTXT	из Operation: Operation-Spec, Operation-Texts из Операции настройки диапазона: Range-Op-Text
Динамическая контекстная группа Операции (расширяемая группа атрибутов)	MDC_ATTR_GRP_OP_DYN_CTXT	из Operation: Operational-State, Vmo-Reference из Set Range Operation: Current-Range, Set-Value-Range, Unit-Code, Step-Width

Применяют следующие определения типов:

```
--
-- Атрибут Current-Range (Текущий диапазон) определяет текущий диапазон значений
```

```
--
CurrentRange ::= SEQUENCE {
    lower                   FLOAT-Type,
    upper                   FLOAT-Type
}

--
-- Атрибут Range-Op-Text присваивает этикетки (маркировки) верхней и нижней границе
--
RangeOpText ::= SEQUENCE {
    low-text                OCTET STRING,    -- печатный текст этикетки
                                             -- (маркировки) для нижнего
                                             -- значения
    high-text               OCTET STRING     -- печатный текст этикетки
                                             -- (маркировки) для верхнего
                                             -- значения
}
```

7.6.9.2 Поведение

Объект Set Range Operation не определяет никакие конкретные методы.

7.6.9.3 Уведомления

Объект Set Range Operation не формирует никакие конкретные уведомления.

7.7 Объекты в Расширенном пакете сервисов

Определения объектов в Расширенном пакете сервисов даны в пунктах 7.7.1—7.7.9.

7.7.1 Объект Scanner (Сканер)

Объект:	Scanner
Описание:	Объект Scanner — это объект, который наблюдает и суммирует значения атрибутов объекта. Он следит за атрибутами управляемых медицинских объектов и выдает сводную информацию в виде отчетов уведомлений о событиях. Будучи абстрактным базовым классом, класс Scanner не может иметь конкретных представителей-экземпляров.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_SCAN

7.7.1.1 Атрибуты

Класс объекта Scanner определяет атрибуты, представленные в таблице 7.70.

Таблица 7.70 — Атрибуты класса объекта Scanner

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Сканеры идентифицируются описателями (объектами handle)	M
Instance-Number	MDC_ATTR_ID_INSTNO	InstNumber	Должен использоваться, когда разрешено динамическое создание экземпляров сканера	C
Operational-State	MDC_ATTR_OP_STAT	OperationalState	Определяет активен сканнер или нет; может быть установлен (настроен)	M

В таблице 7.71 класс объекта Scanner описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.71 — Группы атрибутов класса объекта Scanner

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Сканер (расширяемая группа атрибутов)	MDC_ATTR_GRP_SCAN	из Scanner: (все)

Атрибуты не требуют никаких новых типов определений.

7.7.1.2 Поведение

Объект Scanner не описывает никакие конкретные методы.

Полученные специализации сканера используют следующие общие типы данных:

```
--
-- Список объектов, для которого просканированные атрибуты подвергаются
-- обновлению
-- Если список пуст, то все объекты в списке сканирования обновляются
-- Если scanned-attribute равен 0 (NOS), то все просканированные атрибуты
-- объекта обновляются.
-- Если object-glb-handle равен 0 (во всех компонентах), то указанный
-- идентификатор атрибута обновляется для всех объектов в списке
-- сканирования
--
RefreshObjList ::= SEQUENCE OF RefreshObjEntry
RefreshObjEntry ::= SEQUENCE {
    object-glb-handle           GLB-HANDLE,
    scanned-attribute           OID-Type        -- ID атрибута из объектно-
                                                -- ориентированного раздела
                                                -- номенклатуры
}
```

7.7.1.3 Уведомления

События определяются в полученных специализациях сканера.

Тем не менее, большинство специализаций сканера (специализированных классов-потомков сканера) имеют одинаковую структуру данных в отчете о событии, которая указана ниже:

```
--
-- Сканер может сканировать объекты из множества различных контекстов
-- прибора. В целях эффективности просканированные данные, которые
-- относятся к персональному контексту прибора, объединяют в одну группу
--
ScanReportInfo ::= SEQUENCE {
    scan-report-no              INT-U16,        -- счетчик для обнаружения
                                                -- пропущенных событий
    glb-scan-info               SEQUENCE OF SingleCtxtScan
}

SingleCtxtScan::= SEQUENCE {
    context-id                  MdsContext,
    scan-info                   SEQUENCE OF ObservationScan
}

ObservationScan ::= SEQUENCE {
    obj-handle                  HANDLE,
    attributes                  AttributeList
}
```

7.7.2 Объект CfgScanner (Конфигурируемый сканер)

Объект: CfgScanner

Описание: У объекта CfgScanner есть специальный атрибут ScanList, который позволяет системе устанавливать, какие атрибуты объектов следует сканировать. Объект CfgScanner имеет следующие свойства:

- он сканирует объекты, производные от VMO (в основном объекты Metric, Channel и объекты прибора VMD);
- он содержит список просканированных объектов/атрибутов, которые могут быть модифицированы.

Объект CfgScanner является абстрактным классом. Невозможно создать его экземпляр.

Производное от: Scanner (Сканер)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_SCAN_CFG

7.7.2.1 Атрибуты

Класс объекта CfgScanner определяет атрибуты, описанные в таблице 7.72.

Таблица 7.72 — Атрибуты класса объекта CfgScanner

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Scan-List	MDC_ATTR_SCAN_LIST	ScanList	Список просканированных объектов и атрибутов; можно установить	M
Confirm-Mode	MDC_ATTR_CONFIRM_MODE	ConfirmMode	Определяет, используются ли подтвержденные отчеты о событии	M
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	Определяет, когда повторно отправить подтвержденный отчет о событии, если нет ответа	C
Transmit-Window	MDC_ATTR_TX_WIND	INT-U16	Максимальное количество до сих пор не подтвержденных отчетов о событии за одно и тоже время	C
Scan-Config-Limit	MDC_ATTR_SCAN_CFG_LIMIT	ScanConfigLimit	Даже конфигурируемый сканер может запрещать способ, с помощью которого он может быть сконфигурирован	O

Класс объекта CfgScanner описывает в таблице 7.73 Группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.73 — Группы атрибутов класса объекта CfgScanner

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Сканер (расширяемая группа атрибутов)	MDC_ATTR_GRP_SCAN	(все)

Применяют следующие определения типа:

```
--
-- Атрибут Scan-List (Список сканирования) определяет, какие атрибуты
-- объекта наблюдаются
--
```

```
-- Примечания
-- 1 Если список сканирования пуст, эпизодический сканер вынужден
-- отправлять пустые отчеты о событии.
-- 2 Список сканирования обычно содержит идентификаторы группы атрибутов
-- для определенных объектов.
--
ScanList ::= SEQUENCE OF ScanEntry

ScanEntry ::= SEQUENCE {
    object-glb-handle          GLB-HANDLE,    -- работает для всех объектов с
                                              -- описателем связывания имен
    scanned-attribute          OID-Type       -- могут быть также ID группы
                                              -- атрибутов
}

--
-- Атрибут Confirm-Mode (Режим подтверждения) определяет, какие отчеты о
-- событии используются: подтвержденные или неподтвержденные
--
ConfirmMode ::= INT-U16 {
    unconfirmed (0),
    confirmed (1)
}

--
-- Даже конфигурируемый сканер может запретить способ, которым его можно
-- сконфигурировать.
-- Если Атрибут Scan-Config-Limit (Предел конфигурирования сканера)
-- отсутствует, то сканер может быть полностью сконфигурирован
--
ScanConfigLimit ::= BITS-16 {
    no-scan-delete (0),                       -- сканер нельзя удалить
    no-scan-list-mod (1),                     -- список сканирования нельзя
                                              -- модифицировать в динамическом
                                              -- режиме
    auto-init-scan-list (3),                  -- список сканирования автоматически
                                              -- инициируется после создания
                                              -- сканера
    auto-updt-scan-list (4)                   -- список сканирования автоматически
                                              -- обновляется в случае изменения
                                              -- конфигурации
}
```

7.7.2.2 Поведение

Объект CfgScanner не определяет никакие конкретные методы.

7.7.2.3 Уведомления

События определяются в полученных специализациях сканера.

7.7.3 Объект EpiCfgScanner (Эпизодически конфигурируемый сканер)

Объект:	EpiCfgScanner
Описание:	Объект EpiCfgScanner отвечает за сканирование атрибутов или групп атрибутов объектов и за регистрацию данных атрибутов в эпизодических, не находящихся в буфере (т. е. инициируемых только изменением значения атрибута объекта) отчетах о событии.

Производное от: CfgScanner

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_SCAN_CFG_EPI

7.7.3.1 Атрибуты

Класс объекта EpiCfgScanner не определяет никакие атрибуты, кроме тех, которые являются производными от объекта Cfg-Scanner.

Класс объекта EpiCfgScanner использует группу атрибутов сканера, которая является производной от объекта Cfg-Scanner.

7.7.3.2 Поведение

Объект EpiCfgScanner определяет методы, описанные в таблице 7.74.

Таблица 7.74 — Методы объекта EpiCfgScanner

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Refresh-Episodic-Data	Подтвержден	MDC_ACT_REFR_EPI_DATA	RefreshObjList	нет

Метод Refresh-Episodic-Data (Обновление эпизодических данных) запускает обновление всех просканированных атрибутов.

7.7.3.3 Уведомления

Объект EpiCfgScanner определяет события, описанные в таблице 7.75.

Таблица 7.75 — События объекта EpiCfgScanner

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Unbuf-Scan-Report	Подтвержден/ неподтвержден	MDC_ACT_REFR_EPI_DATA	ScanReportInfo	—

Примечания

1 Если при сканировании групп атрибутов объекта EpiCfgScanner одно или более значения атрибутов в группе изменяются, то сканер отправляет отчет обо всех значениях атрибутов в группе, даже о тех, которые не изменили свое значение. Важно, чтобы атрибуты, которые подлежат удалению из экземпляра объекта в динамическом режиме, могли быть удалены без особого уведомления.

2 Если ни один атрибут объекта не меняет свое значение, то в отчете о сканировании не указываются никакие данные этого объекта (если не была запущена определенная фаза обновления).

3 Так как эпизодический сканер не буферизирует никакие изменения и не имеет атрибута спецификации периода обновления (который не требуется, так как обновления опираются на изменения значений), уведомления об изменении атрибута должны отправляться с такой частотой, которая обеспечит защиту от потери данных. Например, чтобы обеспечить, чтобы ни одно метрическое значение не менялось более одного раза в период между сканированиями динамических групп атрибутов, эпизодический сканер должен проверять изменения с минимум той же скоростью, что и наименьший период MetricSpec::update-period метрических экземпляров в списке сканирования сканера.

4 После проведения операции по созданию экземпляра сканера, все значения атрибута считаются измененными, для того, чтобы первый отчет сканирования содержал все значения атрибутов всех объектов.

7.7.4 Объект PeriCfgScanner (Периодически конфигурируемый сканер)

Объект: PeriCfgScanner

Описание: Объект PeriCfgScanner отвечает за сканирование атрибутов и групп атрибутов объектов и за регистрацию данных атрибутов в периодических отчетах о событии.

Производное от: CfgScanner

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_SCAN_CFG_PERI

7.7.4.1 Атрибуты

Класс объекта PeriCfgScanner определяет атрибуты, описанные в таблице 7.76.

Таблица 7.76 — Атрибуты класса объекта PeriCfgScanner

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Scan-Extensibility	MDC_ATTR_SCAN_EXTEND	ScanExtend	По умолчанию — расширенное значение	M
Reporting-Interval	MDC_ATTR_SCAN_REP_PD	RelativeTime	Период получения отчетов	M

В таблице 7.77 класс объекта PeriCfgScanner описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.77 — Группы атрибутов класса объекта PeriCfgScanner

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Сканер (расширяемая группа атрибутов)	MDC_ATTR_GRP_SCAN	(все)

Атрибуты требуют следующие определения новых типов:

```
--
-- Атрибут Scan-Extensibility (Расширенное сканирование) определяет,
-- включает ли сканер все наблюдения в параметр события ScanReportInfo или
-- только последние (т. е. суперположительный)
--
ScanExtend ::= INT-U16 {
    extensive (0),              -- включены все изменения атрибутов,
                                -- происходящие в период сканирования
    superpositive (1),          -- включено только последнее изменение атрибута
    superpositive-avg (2)       -- суперположительный, но все значения
                                -- в этот период усредняются
}
```

7.7.4.2 Поведение

Объект PeriCfgScanner не описывает никакие конкретные методы.

7.7.4.3 Уведомления

Объект PeriCfgScanner отправляет уведомления, указанные в таблице 7.78.

Таблица 7.78 — События объекта PeriCfgScanner

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Buf-Scan-Report	Подтвержден/неподтвержден	MDC_NOTI_BUF_SCAN_RPT	ScanReportInfo	—

7.7.5 Объект FastPeriCfgScanner

Объект: FastPeriCfgScanner

Описание:	Класс FastPeriCfgScanner — специализированный класс объектов, предназначенный для того, чтобы отслеживать значения подлежащих просмотру значения атрибутов объекта класса Real Time Sample Array. Этот специальный сканер глубоко оптимизирован для оперативной отправки отчетов с минимальными задержками и эффективного использования пропускной способности канала коммуникации, что необходимо для нормального доступа к значениям данных в форме осциллограмм, получаемым в реальном масштабе времени.
Производное от:	PeriCfgScanner
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_SCAN_CFG_PERI_FAST

7.7.5.1 Атрибуты

Класс объекта FastPeriCfgScanner не определяет никакие атрибуты, кроме тех, которые являются производными от объекта PeriCfgScanner.

Класс объекта FastPeriCfgScanner использует группу атрибутов сканера, которая является производной от объекта PeriCfgScanner.

7.7.5.2 Поведение

Объект FastPeriCfgScanner не определяет никакие конкретные методы.

7.7.5.3 Уведомления

Объект FastPeriCfgScanner отправляет уведомления, указанные в таблице 7.79.

Таблица 7.79 — События объекта FastPeriCfgScanner

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Fast-Buf-Scan-Report	Подтвержден/ неподтвержден	MDC_NOTI_BUF_SCAN_RPT	FastScanReportInfo	—

Применяют следующие определения типа:

```
--
-- Отчет о событии содержит в себе наблюдаемые значения просканированных
-- объектов Real Time Sample Array
--
FastScanReportInfo ::= SEQUENCE {
    scan-report-no          INT-U16,
    glb-scan-info           SEQUENCE OF SingleCtxtFastScan
}

SingleCtxtFastScan ::= SEQUENCE {
    context-id              MdsContext,
    scan-info               SEQUENCE OF RtsaObservationScan
}

RtsaObservationScan ::= SEQUENCE {
    handle                  HANDLE,
    observation             SaObsValue
}
```

Объект FastPeriCfgScanner — это сканер, выделенный специально для объектов Real Time Sample Array. В целях эффективной работы массивы выборок имеют одну и ту же временную метку в каждой структуре сканирования наблюдения. Для синхронизации времени и формирования временных меток определенных выборок могут поддерживаться два разных метода:

a) метод по умолчанию предполагает, что временная метка, обеспечиваемая сервисом EVENT REPORT, является значением времени первой выборки в каждой структуре данных RtsaObservationScan::SaObsValue;

b) для более точной синхронизации времени объекты Real Time Sample Array могут поддерживать атрибуты Average-Reporting-Delay (среднее время задержки из-за составления отчета) и Sample-Time-Sync (время синхронизации образца). Наличие флажка Time-Support::time-capability-time-capab-rtsa-time-sync-highprecision в объекте Clock сигнализирует о поддержке данного метода. В случае использования данного метода, эти атрибуты определяют отдельные значения времен выборки, и они становятся независимыми от временной метки, обеспечиваемой службой EVENT REPORT.

7.7.6 Объект UcfgScanner (Неконфигурируемый сканер)

Объект: UcfgScanner

Описание: Класс UcfgScanner — абстрактный базовый класс, который предназначен для сканирования предопределенного набора управляемых медицинских объектов, которые не могут быть изменены. Другими словами, объект класса UcfgScanner — типичный генератор отчетов, предназначенный для решения одной конкретной задачи. Он имеет следующие свойства:

a) отчеты о событии сканера обычно используются в подтверждаемом режиме, так как данные, которые они содержат, могут менять свое состояние;

b) список сканированных объектов/атрибутов фиксирован (т. е. их нельзя конкретизировать).

Будучи абстрактным базовым классом, класс UcfgScanner не может иметь конкретных представителей-экземпляров.

Производное от: Scanner (Сканер)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_SCAN_UCFG

7.7.6.1 Атрибуты

Класс объекта UcfgScanner определяет атрибуты, описанные в таблице 7.80.

Таблица 7.80 — Атрибуты класса объекта UcfgScanner

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Confirm-Mode	MDC_ATTR_CONFIRM_MODE	ConfirmMode	По умолчанию является подтвержденным режимом	O
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	Определяет, когда повторно отправлять подтвержденный отчет о событии в случае отсутствия ответа	O
Transmit-Window	MDC_ATTR_TX_WIND	INT-U16	Максимальное количество еще не подтвержденных отчетов о событии за один раз	O

В таблице 7.81 класс объекта UcfgScanner описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.81 — Группы атрибутов класса объекта UcfgScanner

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Сканер (расширяемая группа атрибутов)	MDC_ATTR_GRP_SCAN	(все)

7.7.6.2 Поведение

Объект UcfgScanner не определяет никакие конкретные методы.

7.7.6.3 Уведомления

События описаны в полученных специализациях сканера.

7.7.7 Объект Context Scanner (Контекстный сканер)

Объект:	UcfgScanner
Описание:	Класс Context Scanner предназначен для отслеживания изменений конфигурации устройства. Порожденный объект класса Context Scanner отвечает за оповещение о появлении новых экземпляров объектов в базе данных MDIB прибора. Сканер предоставляет данные об иерархии вложенности экземпляров объекта и значениях статических атрибутов объекта. В случае динамически изменяемой конфигурации объект класса Context Scanner посылает уведомления о появлении новых экземпляров объекта или удалении существующих.
Производное от:	UcfgScanner
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_SCAN_UCFG_CTXT

7.7.7.1 Атрибуты

Класс объекта Context Scanner определяет атрибуты, описанные в таблице 7.82.

Таблица 7.82 — Атрибуты класса объекта Context Scanner

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Context-Mode	MDC_ATTR_SCAN_CTXT_MODE	ContextMode	Значение по умолчанию динамическое	M

Класс объекта Context Scanner использует группу атрибутов сканера, которая определяется объектом Scanner.

Атрибуты требуют следующие определения новых типов:

```
--
-- Атрибут Context-Mode (Контекстный режим) определяет, отправил ли
-- контекстный сканер уведомления о создании для максимального набора
-- экземпляров объекта в MDIB (и не требует ли уведомлений об удалении) или
-- только для активных объектов
--
ContextMode ::= INT-U16 {
    static-mode (0),
    dynamic-mode (1)
}
```

7.7.7.2 Поведение

Объект Context Scanner определяет методы, описанные в таблице 7.83.

Таблица 7.83 — Методы объекта Context Scanner

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Refresh-Context	Подтвержден	MDC_ACT_REFR_CTXT	RefreshObjList	ObjCreateInfo (номер отчета о сканировании — 0)

Метод Refresh-Context (Обновление контекста) возвращает информацию о конфигурации для всех экземпляров объекта, находящихся в текущий момент в MDIB.

7.7.7.3 Уведомления

Объект Context Scanner определяет события, описанные в таблице 7.84.

Таблица 7.84 — События объекта Context Scanner

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Object-Create-Notification	Подтвержден/ неподтвержден	MDC_NOTI_OBJ_ CREAT	ObjCreateInfo	—
Object-Delete-Notification	Подтвержден	MDC_NOTI_OBJ_DEL	ObjDeleteInfo	—

Применяют следующие определения типа:

```
--
-- Событие Object-Create-Notification (Уведомление о создании объекта)
-- содержит тип, идентификатор и атрибутивную информацию о новых
-- экземплярах объекта в MDIB
--
ObjCreateInfo ::= SEQUENCE {
    scan-report-no              INT-U16,
    scan-report-info            SEQUENCE OF CreateEntry
}

--
-- Отдельная новая запись для одного родительского объекта, необходимого
-- для построения иерархии в базе MDIB
--
CreateEntry ::= SEQUENCE {
    superior-object             ManagedObjectId,
    created-object              SEQUENCE OF CreatedObject
}

--
-- Вот, наконец, и сам новый объект
--
CreatedObject ::= SEQUENCE {
    class-id                    ManagedObjectId,
    attributes                  AttributeList
}

--
-- Событие Object-Delete-Notification (Уведомление об удалении объекта)
-- косвенно удаляет также все дочерние объекты
--
ObjDeleteInfo ::= SEQUENCE {
    scan-report-no              INT-U16,
    object-list                 SEQUENCE OF ManagedObjectId
}
```

7.7.8 Объект Alert Scanner (Сканер аварийных состояний)

Объект: Alert Scanner

Описание: Класс Alert Scanner предназначен для отслеживания групп атрибутов, связанных с аварийными сигналами, у объектов в Пакете тревога. По соображениям обеспечения безопасности данный сканер не конфигурируем (отслеживаются или все или никакие объекты класса Alert). Объект класса Alert Scanner периодически посылает отчеты о событиях для проверки условий времени ожидания (тайм-аута).

Производное от: UcfgScanner

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_SCAN_UCFG_ALSTAT

7.7.8.1 Атрибуты

Класс объекта Alert Scanner определяет атрибуты, описанные в таблице 7.85.

Таблица 7.85 — Атрибуты класса объекта Alert Scanner

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квали-фикатор
Reporting-Interval	MDC_ATTR_SCAN_REP_PD	RelativeTime	Период отправления отчетов	M

Класс объекта Alert Scanner использует группу атрибутов сканера, которая определяется объектом Scanner.

Атрибут не требует определений новых типов.

7.7.8.2 Поведение

Объект Alert Scanner не определяет никакие конкретные методы.

7.7.8.3 Уведомления

Объект Alert Scanner определяет события, описанные в таблице 7.86.

Таблица 7.86 — События объекта Alert Scanner

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Alert-Scan-Report	Подтвержден/ неподтвержден	MDC_NOTI_AL_STAT_SCAN_ RPT	ScanReportInfo	—

7.7.9 Объект Operating Scanner (Сканер функционирования)

Объект: Operating Scanner

Описание: Класс Operating Scanner предназначен для предоставления полной информации о функционировании и системе управления медицинского прибора. Данная информация в основном включает в себя объекты Operation, содержащиеся в SCO, которые рассматриваются как свойства объекта SCO, а не отдельно управляемые медицинские объекты. Рабочий сканер:

- отправляет события CREATE для экземпляров объекта Operation;
- сканирует атрибуты объекта Operation вместе с атрибутами Группы операций SCO (см. 7.6.1.1);
- обеспечивает механизм обновления атрибутов объекта Operation.

Производное от: UcfgScanner

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_SCAN_UCFG_OP

7.7.9.1 Атрибуты

Класс объекта Operating Scanner не определяет никакие атрибуты, кроме тех, которые являются производными от объекта UcfgScanner.

Класс объекта Operating Scanner использует группу атрибутов сканера, которая определяется объектом Scanner.

7.7.9.2 Поведение

Объект Operating Scanner определяет методы, описанные в таблице 7.87.

Таблица 7.87 — Методы объекта Operating Scanner

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Refresh-Operation-Context	Подтвержден	MDC_ACT_REFR_OP_CTXT	RefreshObjList	OpCreateInfo (номер отчета о сканировании — 0)
Refresh-Operation-Attributes	Подтвержден	MDC_ACT_REFR_OP_ATTR	RefreshObjList	—

Примечание — Параметр действия RefreshObjList (Обновление списка объектов) для метода Refresh-Operation-Attributes (Обновление атрибутов Operation) может идентифицировать как атрибуты объекта SCO, так и атрибуты объекта Operation.

7.7.9.3 Уведомления

Объект Operating Scanner определяет события, описанные в таблице 7.88.

Таблица 7.88 — События объекта Operating Scanner

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Oper-Create-Notification	Подтвержден/неподтвержден	MDC_NOTI_OP_CREAT	OpCreateInfo	—
Oper-Delete-Notification	Подтвержден	MDC_NOTI_OP_DEL	OpDeleteInfo	—
Oper-Attribute-Update	Подтвержден/неподтвержден	MDC_NOTI_OP_ATTR_UPDT	OpAttributeInfo	—

Применяют следующие определения типа:

```
--
-- Данные типов поддержки
--
OpElemAttr ::= SEQUENCE {
    op-class-id                 OID-Type,
    op-instance-no              InstNumber,
    attributes                  AttributeList
}

OpElemAttrList ::= SEQUENCE OF OpElemAttr

OpElem ::= SEQUENCE {
    op-class-id                 OID-Type,
    op-instance-no              InstNumber
}

--
-- Операции по созданию и удалению
--
OpCreateInfo ::= SEQUENCE {
    scan-report-no              INT-U16,
    scan-info                   SEQUENCE OF OpCreateEntry
}

OpCreateEntry ::= SEQUENCE {
    sco-glb-handle              GLB-HANDLE,
    created-op-list             OpElemAttrList
}
```

```
OpDeleteInfo ::= SEQUENCE {
    scan-report-no          INT-U16,
    deleted-op-list         SEQUENCE OF OpDeleteEntry
}

OpDeleteEntry ::= SEQUENCE {
    sco-glb-handle          GLB-HANDLE,
    deleted-op-list         SEQUENCE OF OpElem
}

--
-- Отчет, содержащий атрибуты объекта Operation (при необходимости из
-- разных контекстов)
--
OpAttributeInfo ::= SEQUENCE {
    scan-report-no          INT-U16,
    glb-scan-info           SEQUENCE OF SingleCtxtOperScan
}

SingleCtxtOperScan ::= SEQUENCE {
    context-id              MdsContext,
    scan-info               SEQUENCE OF OpAttributeScan
}

--
-- Просканированная информация содержит атрибуты транзакции SCO и
-- атрибуты объекта Operation
--
OpAttributeScan ::= SEQUENCE {
    sco-handle              HANDLE,
    invoke-cookie           INT-U32,
    lock-state              AdministrativeState,
    op-elem-updt-list       OpElemAttrList
}
```

7.8 Объекты в Пакете коммуникаций

Определения объектов в Пакете коммуникаций указаны в пунктах 7.8.1—7.8.7.

7.8.1 Объект Communication Controller (Коммуникационный контроллер)

Объект:	Communication Controller
Описание:	Объект Communication Controller представляет собой коммуникационный профиль главного уровня и низкого уровня (т. е. прикладной профиль, профиль формата и транспортный профиль) и обеспечивает методы доступа для получения управляющей информации, связанной с коммуникациями для передачи данных.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_CC (из раздела номенклатуры объектно-ориентированных элементов)

7.8.1.1 Атрибуты

Класс объекта Communication Controller определяет атрибуты, описанные в таблице 7.89.

Таблица 7.89 — Атрибуты класса объекта Communication Controller

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ ID_HANDLE	Handle	Идентификатор для указания ссылки на объект	M
Capability	MDC_ATTR_ CC_CAPAB	CcCapability	Битовое поле, показывающее конкретные функциональные возможности реализации объекта Communication Controller	M
CC-Type	MDC_ATTR_ CC_TYPE	CC-Oid	Можно использовать для указания вариантов, например, ИСО/ИИЭР 11073, локальная вычислительная сеть (LAN), комбинации	O
Number-Of-Difs	MDC_ATTR_ CC_NUM_DIFS	INT-U16	Количество интерфейсов прибора; по умолчанию 1 в случае отсутствия. Объекты Device Interface идентифицируются по их индексу. Индекс — это 16-тибитное число между 1 и значением атрибута Number-Of-Difs. Список конфигурируется в статическом режиме во время конфигурации объекта Communication Controller	O
This-Connection-Dif-Index	MDC_ATTR_ CC_THIS_ DIF_INDEX	INT-U16	Интерфейс прибора, используемый для текущего соединения. 0 или отсутствует, если это нельзя определить/установить с помощью реализации	O
Cc-Ext-Mgmt-Proto-Id	MDC_ATTR_ CC_EXT_ MNG_PROT	CcExtMgmtProto	Указывает идентификатор для внешнего протокола управления, например, простой протокол управления сетью (SNMP) или протокол общей управляющей информации (CMIP)	O

Применяют следующие определения типа данных:

```
--
-- Атрибут Capability (Функциональные возможности) определяет объект Communication Controller
--
CcCapability ::= BITS-32 {
    cc-sup-ext-mgmt-protocol(0)          -- Объект Communication Controller
                                         -- поддерживает протокол внешнего
                                         -- управления (например, SNMP); если
                                         -- установлен этот бит, то требуется
                                         -- наличие атрибута CC-Ext-Mgmt-Proto-Id
}

--
-- Тип данных CC-OID — это обычный 16-тибитный идентификатор OID из раздела
-- номенклатуры инфраструктуры элементов
--
CC-Oid ::= OID-Type

--
-- Ниже представлен список протоколов управления сетями. Диапазон значений
-- от 32768 до 65535 зарезервирован для протоколов, ориентированных на
-- производителей
--
CcExtMgmtProto ::= INT-U16 {
      mgmt-proto-snmp-v1 (1),            -- Простой протокол сетевого
                                         -- управления Версия 1
```

```
        mgmt-proto-snmp-v2 (2),          -- Простой протокол сетевого
                                         -- управления Версия 2
        mgmt-proto-snmp-v3 (3),          -- Простой протокол сетевого
                                         -- управления Версия 3
        mgmt-proto-cmip (16)             -- Протокол общей управляющей
                                         -- информации
}
```

В таблице 7.90 класс объекта Communication Controller описывает атрибутивные группы.

Таблица 7.90 — Группы атрибутов класса объекта Communication Controller

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов объекта Communication Controller (расширяемая группа атрибутов)	MDC_ATTR_GRP_CC	из Communication Controller: (все)

7.8.1.2 Поведение

Объект Communication Controller определяет методы, описанные в таблице 7.91.

Таблица 7.91 — Методы объекта Communication Controller

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Get-Mib-Data	Подтвержден	MDC_ACT_GET_MIB_DATA	GetMibData-Request	GetMibData-Result

Необходимы следующие дополнительные определения типов данных:

```
--
-- Тип данных для сервиса ACTION
-- На один запрос можно получить данные только для одного интерфейса
-- прибора
--
-- Примечание — Если тип mib-id-list пуст, в ответ не будут получены никакие данные MibElement;
действительные записи в типе mib-id-list определяются в атрибуте Mib-Element-List объекта Device
Interface MibElement
--
GetMibDataRequest ::= SEQUENCE {
    dif-index                    INT-U16,
    mib-id-list                  MibIdList
}

MibIdList ::= SEQUENCE OF CC-Oid

--
-- Тип данных для результата сервиса ACTION
--
GetMibDataResult ::= SEQUENCE {
    dif-index                    INT-U16,
    mib-data-list                MibDataList
}

MibDataList ::= SEQUENCE OF MibDataEntry

MibDataEntry ::= SEQUENCE {
    mib-id                       CC-Oid,
    mib-attributes               AttributeList
}
```

7.8.1.3 Уведомления

В объекте Communication Controller не определены никакие конкретные события.

7.8.2 Объект Device Communication Controller (Коммуникационный контроллер прибора, DCC)

Объект:	DCC
Описание:	Класс DCC является производным от класса Communication Controller. Его объекты используются в процессе работы медицинских приборов в качестве систем-агентов (отвечающих при соединении приборов).
Производное от:	Communication Controller (Контроллер связи)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_DCC (из раздела номенклатуры объектно-ориентированных элементов)

7.8.2.1 Атрибуты

Объект DCC не определяет никакие дополнительные атрибуты

7.8.2.2 Поведение

Объект DCC не определяет никакие конкретные методы.

7.8.2.3 Уведомления

В объекте DCC не определены никакие конкретные события.

7.8.3 Объект Bedside Communication Controller (Прикроватный коммуникационный контроллер, BCC)

Объект:	BCC
Описание:	Класс Bedside Communication Controller — производный от класса Communication Controller. Его объекты используются в процессе работы медицинских приборов в качестве систем-менеджеров (запрашивающих соединение приборов).
Производное от:	Communication Controller (Контроллер связи)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_BCC (из раздела номенклатуры объектно-ориентированных элементов)

7.8.3.1 Атрибуты

Объект BCC не определяет никакие дополнительные атрибуты.

7.8.3.2 Поведение

Объект BCC не определяет никакие конкретные методы.

7.8.3.3 Уведомления

В объекте BCC не определены никакие конкретные события.

7.8.4 Объект Device Interface (Интерфейс прибора)

Объект:	Device Interface
Описание:	Объект Device Interface отражает частный интерфейс прибора, то, что называется порт. Порт представляет собой логический или физический конечный пункт соединения, для которого объект Communication Controller независимо собирает (например, статические) данные. Объект класса Device Interface не доступен для функций сервиса CMDISE.
Производное от:	—
Связывание имен:	—

Зарегистрирован как: MDC_CC_DIF (из раздела номенклатуры инфраструктуры)

7.8.4.1 Атрибуты

Класс объекта Device Interface не определяет никакие атрибуты. Все его свойства отражены в объекте Device Interface MibElement. Данный объект MibElement обязателен для каждого экземпляра объекта Device Interface, который поддерживается объектом Communication Controller.

7.8.4.2 Поведение

Объект Device Interface не определяет никакие конкретные методы.

7.8.4.3 Уведомления

В объекте Device Interface не определены никакие конкретные события.

7.8.5 Объект MibElement

Объект: MibElement

Описание: Объект MibElement представляет собой управляющую информацию об определенном физическом или логическом порте объекта Device Interface. Объект MibElement является только абстрактным базовым классом.

Производное от: —

Связывание имен: —

Зарегистрирован как: MDC_CC_MIB_ELEM (из раздела номенклатуры инфраструктуры)

Описание:

7.8.5.1 Атрибуты

Класс объекта MibElement определяет атрибуты, описанные в таблице 7.92.

Таблица 7.92 — Атрибуты класса объекта MibElement

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Mib-Ext-Oid	MDC_CC_MIB_DATA_EXT_OID	OCTET STRING	OCTET STRING (строка октетов) содержит зарегистрированный идентификатор объекта (OID) ИСО, который полностью закодирован с помощью базовых правил кодирования (BER), если MibElement является зарегистрированным концептом. Размер OCTET STRING должен быть целым числом и может требовать использования заполняющего байта. Данный атрибут позволяет включать определения базы данных управляющей информации (MIB) из других стандартов в настоящий стандарт	O

Атрибуты можно извлечь только с помощью специального метода объекта Communication Controller.

Нет необходимости в дополнительных определениях типа данных.

Класс объекта MibElement не определяет никакие атрибутивные группы.

7.8.5.2 Поведение

Объект MibElement не определяет никакие конкретные методы.

7.8.5.3 Уведомления

Для объекта MibElement не определены никакие конкретные события.

7.8.6 Объект The Device Interface MibElement (MibElement интерфейса Прибора)

Объект: The Device Interface MibElement

Описание: Объект The Device Interface MibElement описывает свойства объекта Device Interface. Объект MibElement обязателен для каждого объекта Device Interface, принадлежащего объекту Communication Controller.

Производное от: MibElement

Связывание имен: —

Зарегистрирован как: MDC_CC_MIB_ELEM_DIF (из раздела номенклатуры инфраструктуры)

Объект: MibElement интерфейса прибора

7.8.6.1 Атрибуты

Класс Объекта The Device Interface MibElement определяет атрибуты, описанные в таблице 7.93.

Таблица 7.93 — Атрибуты класса Объекта The Device Interface MibElement

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Dif-Id	MDC_CC_MIB_DATA_DIF_ID	INT-U16	Между значением атрибута равным 1 и Number-Of-Difs в объекте Communication Controller	M
Port-State	MDC_CC_MIB_DATA_DIF_PORT_ST	DifMibPortState	Указывает информацию о порте	M
Port-Number	MDC_CC_MIB_DATA_DIF_PORT_NO	INT-U16	Номер логического порта данного интерфейса прибора	O
Dif-Type	MDC_CC_MIB_DATA_DIF_TYPE	CC-Oid	Допускает записи в раздел номенклатуры инфраструктуры	O
Active-Profile	MDC_CC_MIB_DATA_PROFILE_ID	OID-Type	Данный ID должен содержать идентификатор атрибута поддержки профиля (Profile Support Attribute ID) (см. ИСО/ИИЭР 11073-20101), по аналогии с его использованием в структуре информации пользователя в ASCE, оговоренной для активного профиля во время фазы ассоциации. Если нет активного профиля, то в поле должно быть установлено значение 0	O
Supported-Pro-files	MDC_CC_MIB_DATA_SUPP_PROFILES	SupportedProfileList	См. ниже	O
MTU	MDC_CC_MIB_DATA_MTU	INT-U32	Максимальная длина передаваемых пакетов, в байтах	O
Link-Speed	MDC_CC_MIB_DATA_LINK_SPEED	INT-U32	В битах в секунду	O
Mib-Element-List	MDC_CC_MIB_DATA_MIB_ELEM_LIST	MibElementList	Список элементов MibElements, поддерживаемых объектом Device Interface (в дополнение к данному обязательному элементу MibElement). Предполагается, что элементы MibElements зарегистрированы в номенклатуре	M

Атрибуты можно извлечь только с помощью специального метода объекта Communication Controller.

Применяют следующие определения типа данных:

```
--
-- Атрибут Supported-Profiles (Поддерживаемые профили) определяет, какие
-- профили поддерживаются объектом Device Interface; записи в списке
-- являются идентификаторами Атрибута Поддержки Профиля, по аналогии с тем,
-- как они используются в структуре информации пользователя в ACSE,
-- применяемой для согласования активного профиля (см. определение
```

```
-- прикладных профилей) (записи в списке берутся из раздела номенклатуры
-- инфраструктуры)
--
SupportedProfileList ::= SEQUENCE OF CC-Oid

--
-- Атрибут Mib-Element-List (Список элементов MIB) определяет, какие
-- объекты MibElement поддерживаются объектом Device Interface (записи в
-- списке берутся из раздела номенклатуры инфраструктуры)
--
MibElementList ::= SEQUENCE OF CC-Oid

-- Состояние коммуникационного порта
--
DifMibPortState ::= BITS-16 {
    difmib-port-enabled (0),
    difmib-port-connected (1),          -- порт физически подключен к
                                        -- линии/сети
    difmib-port-associated (2),         -- логическое соединение активно на
                                        -- порте
    difmib-port-failure (15)            -- порт находится в состоянии отказа
                                        -- аппаратного обеспечения
}
```

Объект The Device Interface MibElement не определяет никакие атрибутивные группы.

7.8.6.2 Поведение

Объект The Device Interface MibElement не определяет никакие конкретные методы.

7.8.6.3 Уведомления

Для Объекта The Device Interface MibElement не определены никакие конкретные события.

7.8.7 Объект General Communication Statistics MibElement (MibElement общей статистики коммуникации)

Объект: General Communication Statistics MibElement

Описание: Объект General Communication Statistics MibElement представляет собой общую статистику связи для одного интерфейса прибора.

Производное от: MibElement

Связывание имен: —

Зарегистрирован как: MDC_CC_MIB_ELEM_GEN_COMM_STATS (из таблицы номенклатуры элементов инфраструктуры)

7.8.7.1 Атрибуты

Класс Объекта General Communication Statistics MibElement определяет атрибуты, описанные в таблице 7.94.

Таблица 7.94 — Атрибуты класса объекта General Communication Statistics MibElement

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Packets-In	MDC_CC_MIB_DATA_PACK_IN	MibCcCounter	Количество полученных пакетов	O
Packets-Out	MDC_CC_MIB_DATA_PACK_OUT	MibCcCounter	Количество выданных пакетов	O

Окончание таблицы 7.94

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Octets-In	MDC_CC_MIB_DATA_OCT_IN	MibCcCounter	Количество байтов полученной полезной информации на транспортном интерфейсе (например, без кадрирования)	O
Octets-Out	MDC_CC_MIB_DATA_OCT_OUT	MibCcCounter	Количество байтов отправленной полезной информации на транспортном интерфейсе (например, без кадрирования)	O
Discarded-Packets-In	MDC_CC_MIB_DATA_DISC_PACK_IN	MibCcCounter	Полученные пакеты, не доставленные на верхние уровни	O
Discarded-Packets-Out	MDC_CC_MIB_DATA_DISC_PACK_OUT	MibCcCounter	Пакеты с верхних уровней, не отправленные на сетевой интерфейс	O
Unknown-Protocol-Packets-In	MDC_CC_MIB_DATAUNK_PROT_PACK_IN	MibCcCounter	Полученные пакеты с неизвестным протоколом	O
Queue-Len-In	MDC_CC_MIB_DATA_QUEUE_LEN_IN	MibCcGauge	Размер очереди выходных пакетов в байтах	O
Queue-Len-Out	MDC_CC_MIB_DATA_QUEUE_LEN_OUT	MibCcGauge	Размер очереди входных пакетов в байтах	O
Dif-Admin-Status	MDC_CC_MIB_DATA_DIF_STATE	OperationalState	Желаемое состояние интерфейса прибора	O
Dif-Oper-Status	MDC_CC_MIB_DATA_CUR_DIF_STATE	OperationalState	Текущий статус интерфейса прибора	O
Dif-Last-Change	MDC_CC_MIB_DATA_TIME_DIF_LAST_CHANGE	AbsoluteTime	Время, когда интерфейс прибора последний раз сменил свое состояние	O
Errors-In	MDC_CC_MIB_DATA_ERRS_IN	MibCcCounter	Поврежденные полученные пакеты	O
Errors-Out	MDC_CC_MIB_DATA_ERRS_OUT	MibCcCounter	Поврежденые отправленные пакеты	O
Generic-Mode	MDC_CC_MIB_DATA_COMM_MODE	MibCc-CommMode	Режим коммуникаций	O
Average-Speed	MDC_CC_MIB_DATA_AVG_SPEED	INT-U32	В битах в секунду	O
Maximum-Speed	MDC_CC_MIB_DATA_MAX_SPEED	INT-U32	В битах в секунду	O

Атрибуты можно извлечь только с помощью специального метода объекта Communication Controller.

Применяют следующие определения типа данных:

```
--
-- Тип контрольно-измерительного прибора (gauge type) (из IETF RFC 1155) представляет собой натуральное число, которое может увеличиваться или уменьшаться, но которое удерживается на максимальном значении
--
--
MibCcGauge ::= INT-U32
```

```
--
-- Тип счетчика (из IETF RFC 1155) представляет собой натуральное число,
-- которое однообразно увеличивается, пока не достигнет максимального
-- значения, и затем оно обнуляется и начинает снова увеличиваться с 0
--
MibCcCounter ::= INT-U32

--
-- Тип коммуникационного режима представляет собой коммуникационные режимы,
-- которые поддерживаются интерфейсом прибора.
--
MibCcCommMode ::= BITS-32 {
    comm-mode-simplex (0),
    comm-mode-half-duplex (1),
    comm-mode-full-duplex (2)
}
```

Класс Объекта General Communication Statistics MibElement не определяет никакие атрибутивные группы.

7.8.7.2 Поведение

Объект General Communication Statistics MibElement не определяет никакие конкретные методы.

7.8.7.3 Уведомления

Для объекта General Communication Statistics MibElement не определены никакие конкретные события.

7.9 Объекты в Архивном пакете

Определения объектов в Архивном Пакете даны в пунктах 7.9.1—7.9.6.

7.9.1 Объект Multipatient Archive (Общий архив данных о пациентах)

Объект:	Multipatient Archive
Описание:	Объект класса Multipatient Archive группирует вместе несколько объектов класса Patient Archive.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_ARCHIVE_MULTI_PT

7.9.1.1 Атрибуты

Класс объекта Multipatient Archive определяет атрибуты, описанные в таблице 7.95.

Таблица 7.95 — Атрибуты класса объекта Multipatient Archive

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
System-Id	MDC_ATTR_SYS_ID	OCTET STRING		M
Location	MDC_ATTR_LOCATION	OCTET STRING	***Пример — Название больницы***	M
Study-Name	MDC_ATTR_STUDY_NAME	OCTET STRING		M
Version	MDC_ATTR_ARCHIVE_VERS	OCTET STRING	***Пример — ADS версия 1.0***	M

В таблице 7.96 класс объекта Multipatient Archive описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.96 — Группы атрибутов класса объекта Multipatient Archive

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Archive	MDC_ATTR_GRP_ ARCHIVE	из Multipatient Archive: (все)

7.9.1.2 Поведение

Объект Multipatient Archive не определяет никакие конкретные методы.

7.9.1.3 Уведомления

Объект Multipatient Archive не формирует никакие конкретные уведомления.

7.9.2 Объект Patient Archive (Архив данных о пациенте)

Объект:	Patient Archive
Описание:	Объект The Patient Archive объединяет вместе основные физиологические показатели и другую информацию об одном пациенте.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_ARCHIVE_PT

7.9.2.1 Атрибуты

Класс объекта The Patient Archive определяет атрибуты, описанные в таблице 7.97.

Таблица 7.97 — Атрибуты класса объекта The Patient Archive.

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
System-Id	MDC_ATTR_SYS_ID	OCTET STRING		M
System-Name	MDC_ATTR_NAME_SYS	OCTET STRING	***Пример — Имя файла***	M
Processing-History	MDC_ATTR_PROC_HIST	OCTET STRING	***Пример — Не обработан***	M
Protection	MDC_ATTR_PROTECTION	ArchiveProtection	***Пример — Оригинальная запись***	M

В таблице 7.98 класс объекта The Patient Archive определяет группы атрибутов или расширения для наследования групп атрибутов.

Таблица 7.98 — Группы атрибутов класса объекта The Patient Archive

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Архив	MDC_ATTR_GRP_ARCHIVE	из The Patient Archive: (все)

Применяют следующие определения типов:

```
--
-- Атрибут защиты определяет механизм контроля доступа: данный механизм,
-- определяется продавцом или сферой применения
--
ArchiveProtection ::= SEQUENCE {
    protection-type                 PrivateOid,
    protection-key                  ANY DEFINED BY protection-type
}
```

7.9.2.2 Поведение
Объект The Patient Archive не определяет никакие конкретные методы.
7.9.2.3 Уведомления
Объект The Patient Archive не формирует никакие конкретные уведомления.

7.9.3 Объект Session Archive (Архив данных сессии лечения)

Объект:	Session Archive
Описание:	Объект Session Archive содержит в себе информацию об одном пациенте, которую собирают во время его одного пребывания в больнице или за одно посещение больницы.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_ARCHIVE_SESSION

7.9.3.1 Атрибуты
Класс Объекта Session Archive определяет атрибуты, представленные в таблице 7.99.

Таблица 7.99 — Атрибуты класса Объекта Session Archive

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_ HANDLE	HANDLE	Атрибут связывания имен	M
S-Archive-Id	MDC_ATTR_ID_ SESS_ARCHIVE	OCTET STRING		M
S-Archive-Name	MDC_ATTR_NAME_ SESS_ARCHIVE	OCTET STRING	***Пример — Название исследования***	M
S-Archive-Comments	MDC ATTR SESS ARCHIVE_COMMENTS	OCTET STRING	***Пример — Первая часть теста MSLT (Определение периодов скрытой сонливости)***	O
Start-Time	MDC_ATTR_TIME_ START	AbsoluteTime		M
Stop-Time	MDC_ATTR_TIME_ STOP	AbsoluteTime		M
Protection	MDC_ATTR_ PROTECTION	ArchiveProtection	***Пример — Оригинальная запись***	C

В таблице 7.100 класс объекта Session Archive описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.100 — Группы атрибутов класса Объекта Session Archive

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Архив	MDC_ATTR_GRP_ ARCHIVE	из Session Archive: (все)

7.9.3.2 Поведение
Объект Session Archive не описывает никакие конкретные методы.
7.9.3.3 Уведомления
Объект Session Archive не формирует никакие конкретные уведомления.

7.9.4 Объект Physician (Врач)

Объект: Physician

Описание: Объект Physician представляет собой врача-терапевта.

Производное от: Top (Главный объект)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_PHYSICIAN

7.9.4.1 Атрибуты

Класс объекта Physician определяет атрибуты, представленные в таблице 7.101.

Таблица 7.101 — Атрибуты класса объекта Physician

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
Physician-Id	MDC_ATTR_ID_PHYSICIAN	OCTET STRING		M
Authorization-Level	MDC_ATTR_AUTH_LEVEL	Authorization		C
Name	MDC_ATTR_PHYSICIAN_NAME	OCTET STRING	Имя врача произвольной структуры	O
Given-Name	MDC_ATTR_PHYSICIAN_NAME_GIVEN	OCTET STRING		O
Family-Name	MDC_ATTR_PHYSICIAN_NAME_FAMILY	OCTET STRING		O
Middle-Name	MDC_ATTR_PHYSICIAN_NAME_MIDDLE	OCTET STRING		O
Title-Name	MDC_ATTR_PHYSICIAN_NAME_TITLE	OCTET STRING	***Пример — Профессор***	O

Класс объекта Physician описывает в таблице 7.102 группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.102 — Группы атрибутов класса объекта Physician

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Врач	MDC_ATTR_GRP_PHYSICIAN	из Physician: (все)

Применяют следующие определения типа:

```
--
-- Атрибут Authorization-Level (Уровень авторизации) устанавливает права
-- доступа, используемые для управления доступом; данный механизм зависит
-- от продавца или реализации
--
Authorization ::= SEQUENCE {
    authorization-type      PrivateOid,
    authorization-key       ANY DEFINED BY authorization-type
}
```

7.9.4.2 Поведение
Объект Physician не определяет никакие конкретные методы.
7.9.4.3 Уведомления
Объект Physician не формирует никакие конкретные уведомления.

7.9.5 Объект Session Test (Проверка состояния здоровья в период сессии лечения)

Объект:	Session Test
Описание:	Объект класса Session Test содержит информацию о показателях жизненно важных функций отдельного пациента, зафиксированную во время отдельного освидетельствования или диагностического мероприятия. Этот объект содержит метрические значения показателей жизненно важных функций в форме объектов класса PM-Store. Он также может содержать информацию об оборудовании, которое использовалось для того, чтобы сделать запись (в форме связей с объектами подклассов класса MDS и объектами класса Ancillary).
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_SESSION_TEST

7.9.5.1 Атрибуты
Класс объекта Session Test определяет атрибуты, описанные в таблице 7.103.

Таблица 7.103 — Атрибуты класса объекта Session Test

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
St-Archive-Id	MDC_ATTR_ID_SESS_TEST_ ARCHIVE	OCTET STRING		M
St-Archive-Name	MDC_ATTR_NAME_SESS_ TEST_ARCHIVE	OCTET STRING	***Пример — Название исследования***	M
St-Archive-Comments	MDC_ATTR_SESS_TEST_ ARCHIVE_COMMENTS	OCTET STRING		O
Start-Time	MDC_ATTR_TIME_START	AbsoluteTime		M
Stop-Time	MDC_ATTR_TIME_STOP	AbsoluteTime		M
Protection	MDC_ATTR_PROTECTION	ArchiveProtection		C

В таблице 7.104 класс объекта Session Test определяет группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.104 — Группы атрибутов класса объекта Session Test

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Архив	MDC_ATTR_GRP_ ARCHIVE	из Session Test: (все)

7.9.5.2 Поведение
Объект Session Test не описывает никакие конкретные методы.
7.9.5.3 Уведомления
Объект Session Test не формирует никакие конкретные уведомления.

7.9.6 Объект Session Notes (Примечания к сессии лечения)

Объект:	Session Notes
Описание:	Объект класса Session Notes играет роль контейнера для диагностических данных, деталей лечения и технологической информации в форме текстовых данных.
Производное от:	Top (Главный объект)
Связывание имен:	Handle (Описатель)
Зарегистрирован как:	MDC_MOC_SESSION_NOTES

7.9.6.1 Атрибуты

Класс объекта Session Notes определяет атрибуты, представленные в таблице 7.105.

Таблица 7.105 — Атрибуты класса объекта Session Notes

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
Sn-Id	MDC_ATTR_ID_SESS_NOTES_ ARCHIVE	OCTET STRING		M
Sn-Name	MDC_ATTR_NAME_SESS_ NOTES_ARCHIVE	OCTET STRING		M
Sn-Comments	MDC_ATTR_SESS_NOTES_ ARCHIVE_COMMENTS	OCTET STRING		O
Start-Time	MDC_ATTR_TIME_START	AbsoluteTime		M
Stop-Time	MDC_ATTR_TIME_STOP	AbsoluteTime		M
Findings	MDC_ATTR_FINDINGS	OCTET STRING		O
Diagnostic-Codes	MDC_ATTR_CODE_ DIAGNOSTIC	SEQUENCE OF ExtNomenRef	Диагностические коды, указанные в схеме номенклатуры, не определены в настоящем стандарте	M
Diagnosis-Description	MDC_ATTR_DESC_ DIAGNOSTIC	OCTET STRING		O
Procedure-Code	MDC_ATTR_CODE_ PROCEDURE	SEQUENCE OF ExtNomenRef	Коды процедуры, указанные в схеме номенклатуры, не определены в настоящем стандарте	M
Procedure-Description	MDC_ATTR_DESC_ PROCEDURE	OCTET STRING		O
Protection	MDC_ATTR_PROTECTION	ArchiveProtection		C

В таблице 7.106 класс объекта Session Notes описывает группы атрибутов и расширения к унаследованным группам атрибутов.

Таблица 7.106 — Группы атрибутов класса объекта Session Notes

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Архив	MDC_ATTR_GRP_ARCHIVE	из Session Notes: (все)

7.9.6.2 Поведение

Объект Session Notes не определяет никакие конкретные методы.

7.9.6.3 Уведомления

Объект Session Notes не формирует никакие конкретные уведомления.

7.10 Объекты в Пакете пациента

7.10.1 Объект Patient Demographics (Персональные данные пациента)

Объект: Patient Demographics

Описание: Объект Patient Demographics содержит минимальную информацию о пациентах, как того требует медицинские приборы.

Производное от: Top (Главный объект)

Связывание имен: Handle (Описатель)

Зарегистрирован как: MDC_MOC_PT_DEMOG

7.10.1.1 Атрибуты

Класс объекта Patient Demographics определяет атрибуты, описанные в таблице 7.107.

Таблица 7.107 — Атрибуты класса объекта Patient Demographics

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут связывания имен	M
Pat-Demo-State	MDC_ATTR_PT_DEMOG_ST	PatDemoState	Будучи контейнером, данный объект обладает состоянием	M
Patient-Id	MDC_ATTR_PT_ID	OCTET STRING		O
Name	MDC_ATTR_PT_NAME	OCTET STRING	Имя пациента произвольной структуры	O
Given-Name	MDC_ATTR_PT_NAME_GIVEN	OCTET STRING		O
Family-Name	MDC_ATTR_PT_NAME_FAMILY	OCTET STRING		O
Middle-Name	MDC_ATTR_PT_NAME_MIDDLE	OCTET STRING		O
Birth-Name	MDC_ATTR_PT_NAME_BIRTH	OCTET STRING	Имя медсестры	O
Title-Name	MDC_ATTR_PT_NAME_TITLE	OCTET STRING	***Пример — Профессор***	O
Sex	MDC_ATTR_PT_SEX	PatientSex		O
Race	MDC_ATTR_PT_RACE	PatientRace		O
Patient-Type	MDC_ATTR_PT_TYPE	PatientType		O
Date-Of-Birth	MDC_ATTR_PT_DOB	Date		O
Patient-Gen-Info	MDC_ATTR_PT_GEN_INFO	OCTET STRING	Текстовая информация, относящаяся к пациенту	O
Patient-Age	MDC_ATTR_PT_AGE	PatMeasure	Возраст новорожденных, например, в часах или неделях	O

Окончание таблицы 7.107

Название атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификатор
Gestational-Age	MDC_ATTR_PT_AGE_GEST	PatMeasure	Возраст новорожденных	O
Patient-Height	MDC_ATTR_PT_HEIGHT	PatMeasure		O
Patient-Weight	MDC_ATTR_PT_WEIGHT	PatMeasure		O
Patient-Birth-Length	MDC_ATTR_PT_BIRTH_ LENGTH	PatMeasure	Для новорожденных	O
Patient-Birth-Weight	MDC_ATTR_PT_BIRTH_ WEIGHT	PatMeasure	Для новорожденных	O
Mother-Patient-Id	MDC_ATTR_ID_P_MOTHER	OCTET STRING	Для новорожденных	O
Mother-Name	MDC_ATTR_PT_NAME_ MOTHER	PatientName	Для новорожденных	O
Patient-Head-Circumference	MDC_ATTR CIRCUM _HEAD	PatMeasure		O
Patient-Bsa	MDC_ATTR_PT_BSA	PatMeasure	Площадь поверхности тела; можно рассчитать	O
Patient-Lbm	MDC_ATTR_PT_LBM	PatMeasure	Безжировая масса тела; используемая для расчета дозировки лекарства	O
Bed-Id	MDC_ATTR_ID_BED	OCTET STRING		O
Diagnostic-Info	MDC_ATTR_DIAGNOSTIC_ INFO	OCTET STRING	Произвольный текст для диагноза	O
Diagnostic-Codes	MDC_ATTR_CODE_ DIAGNOSTIC	SEQUENCE OF ExtNomenRef	Диагностические коды указаны в схеме номенклатуры, которая не определена в настоящем стандарте	O
Admitting-Physician	MDC_ATTR_PHYSICIAN_ ADMIT	OCTET STRING	Для блока интенсивной терапии	O
Attending-Physician	MDC_ATTR_PHYSICIAN_ ATTEND	OCTET STRING	Для блока интенсивной терапии	O
Date-Of-Procedure	MDC_ATTR_PROCEDURE_ DATE	Date	Для операционной комнаты (OR)	O
Procedure-Description	MDC_ATTR_DESC_ PROCEDURE	OCTET STRING	Для OR	O
Procedure-Codes	MDC_ATTR_CODE_ PROCEDURE	SEQUENCE OF ExtNomenRef	Для OR; коды процедуры указаны в схеме номенклатуры, которая не определена в настоящем стандарте	O
Anaesthetist	MDC_ATTR_ANAESTHETIST	OCTET STRING	Для OR	O
Surgeon	MDC_ATTR_SURGEON	OCTET STRING	Для OR	O

Примечание — В практических целях, некоторые данные, относящиеся к пациенту, которые также можно смоделировать в виде объектов Metric (например, вес, рост) дублируются здесь в объекте Patient Demographics. Для наилучшего моделирования следует уделить внимание потребностям приложения с учетом сферы применения.

В таблице 7.108 класс объекта Patient Demographics определяет группы атрибутов или расширения для наследования групп атрибутов.

Таблица 7.108 — Группы атрибутов класса объекта Patient Demographics

Группа атрибутов	Идентификатор группы атрибутов	Элементы группы
Группа атрибутов Индивидуальные данные пациента	MDC_ATTR_GRP_ PT_DEMOG	из Patient Demographics: (все)

Применяют следующие определения типа:

```
--
-- Состояние объекта Patient Demographics
--
PatDemoState ::= INT-U16 {
    empty (0),
    pre-admitted (1),
    admitted (2),
    discharged (8)
}

--
-- Измеренное значение для персональных данных пациента
--
PatMeasure ::= SEQUENCE {
    value                   FLOAT-Type,
    m-unit                  OID-Type            -- код единицы измерения
}

--
-- Пол пациента согласно ИСО/МЭК 5218
--
PatientSex ::= INT-U16 {
    sex-unknown (0),
    male (1),
    female (2),
    sex-unspecified (9)
}

--
-- Атрибут Patient-Type (Тип пациента)
--
PatientType ::= INT-U16 {
    pt-unspecified (0),
    adult (1),
    pediatric (2),
    neonatal (3)
}

--
-- Раса пациента согласно стандартному протоколу связи [для компьютеризированной] электрокардиографии (SCP ECG) (смотрите CEN EN 1064)
--
```

```
PatientRace ::= INT-U16 {
    race-unspecified (0),
    race-caucasian (1),
    race-black (2),
    race-oriental (3)
}
```

7.10.1.2 Поведение

Объект Patient Demographics определяет методы, описанные в таблице 7.109.

Таблица 7.109 — Методы объекта Patient Demographics

Действие	Режим	Идентификатор действия	Параметр действия	Результат действия
Discharge-Patient	Подтвержден	MDC_ACT_DISCH_PT	—	PatDemoState
Admit-Patient	Подтвержден	MDC_ACT_ADMIT_PT	AdmitPatInfo	PatDemoState
Pre-Admit-Patient	Подтвержден	MDC_ACT_PRE_ADMIT_PT	AdmitPatInfo	PatDemoState

Применяют следующие определения типа:

```
--
-- Метод Admit-Patient (Администратор — Пациент)
--
AdmitPatInfo ::= AttributeList
```

7.10.1.3 Уведомления

Объект Patient Demographics определяет события, описанные в таблице 7.110.

Таблица 7.110 — События объекта Patient Demographics

Событие	Режим	Идентификатор события	Характеристика события	Результат события
Patient-Demographics-Modified	Подтвержден/ неподтвержден	MDC_NOTI_PT_DEMOG_MOD	AttributeList	—
Patient-Demographics-State-Change	Подтвержден/ неподтвержден	MDC_NOTI_PT_DEMOG_ST_MOD	AttributeList	—

Нет необходимости в дополнительных определениях типа.

8 Модель сервисов коммуникационных систем

8.1 Общие положения

В этом разделе определяются основные сервисы прикладного уровня, предоставляемые коммуникационными системами, соответствующими настоящему стандарту. Эти сервисы используются прикладными процессами для обмена данными о значениях жизненно важных показателей и командами, позволяющими контролировать состояние прибора и выполняемые с его помощью измерения.

8.2 Коммуникационные системы

Предлагаемая архитектура коммуникации основана на модели взаимодействия агент — менеджер, понятие которой было введено Международной организацией по стандартизации (ИСО) в области

управления системами. Можно выделить три типа систем, которые обрабатывают и передают данные о жизненно важных показателях организма:

1 Агент данных о жизненно важных показателях организма (т. е. система — агент, предоставляющая данные о значениях жизненно важных показателей в форме управляемых медицинских объектов).

2 Менеджер данных о жизненно важных показателях организма (т. е. система — менеджер, принимающая и обрабатывающая данные о значениях жизненно важных показателей в форме управляемых медицинских объектов).

3 Гибридная система, сочетающая функции и агента, и менеджера данных о жизненно важных показателях организма.

Примечание — Термин жизненно важные показатели относится к области применения настоящего стандарта, но не к типу или своевременности поступления этих данных. Например, архивированные данные о значениях жизненно важных показателей организма могут быть предоставлены агентом данных для обеспечения менеджера данных запрашиваемой информацией с использованием удаленного доступа к базе данных.

Конкретное приложение, участвующее в процессе коммуникации, может содержать в себе две и более из числа вышеупомянутых систем.

Пример — Полная система контроля аритмии может состоять из монитора кардиограммы (т. е. агента данных о значениях показателей жизненно важных функций), компьютера обработки данных об аритмии (то есть гибридной системы данных о значениях показателей жизненно важных функций) и центрального дисплея и устройства хранения данных (т. е. менеджера данных о значениях показателей жизненно важных функций).

8.3 Общая модель сервисов

Диапазон приборов, системных конфигураций и приложений, охватываемых областью применения настоящего стандарта (т. е. показатели жизненно важных функций), очень широк. Их сложность изменяется в диапазоне от простого прибора, обеспечивающего измерение единственного числового параметра, до системы, которая состоит из нескольких динамически переконфигурируемых приборов измерения и обработки данных о значениях жизненно важных показателей. В долгосрочной перспективе вполне вероятно, что новые информационные объекты или объекты, обеспечивающие определенные функциональные возможности, должны быть добавлены к информационной модели предметной области, чтобы отразить продолжающиеся усовершенствования в области медицинских приборов и измерений.

Поэтому, специализированные сообщения для всех возможных показателей жизненно важных функций и каждого возможного приложения не могут быть определены, не порождая издержек для небольших устройств и трудностей в дальнейшем применении настоящего стандарта и любых основанных на нем практических реализаций систем. Эти обстоятельства требуют определения обобщенной модели сервисов, которая в значительной степени будет независима от информационной модели предметной области и не потребует модификации при необходимости введения новых классов.

Информационная модель предметной области основана строго на объектно-ориентированной методологии. Этим определяется представление данных о значениях показателей жизненно важных функций в форме классов (объектов) и иерархий классов (т. е., управляемых медицинских объектов). Каждый класс имеет свои уникальные идентификаторы, атрибуты и методы.

Модель сервисов коммуникационных систем обеспечивает доступ к этим управляемым медицинским объектам посредством основных сервисов управления объектом, которые независимы от определений конкретных классов. Такие сервисы управления объектом позволяют расширять информационную модель, добавляя дополнительные классы в последующих стандартах и не затрагивая при этом модель сервисов, рассматриваемую в настоящем стандарте.

Основные сервисы управления объектом, определяемые в настоящем стандарте, в основном концептуально базируются на управлении взаимодействием открытых систем (т. е. на стандарте ИСО/МЭК 10040 и семействе стандартов ИСО/МЭК 10164) и в частности на сервисе единого протокола доступа к управляющей информации ИСО/МЭК (CMISE) (т. е., на стандарте ИСО/МЭК 9595).

Примечание — Объекты, которые предоставляют дополнительные сервисы управления как определено в семействе стандартов ИСО/МЭК 10164, определены в информационной модели предметной области, в частности в пакете Дополнительные сервисы. Дополнительные сервисы, определенные объектами, вызываются основными сервисами управления объектом, определенными в настоящем разделе. В настоящем разделе термин сервисы относится к определяемым здесь сервисам прикладного уровня, если не указано иное.

Сервисы позволяют осуществлять обмен информацией об управляемых медицинских объектах, определенных в информационной модели предметной области, между двумя равноправными сущностями. В коммуникационной системе сервисы обеспечиваются единым протоколом доступа к данным медицинских приборов (CMDISE). Это название (т. е. CMDISE) выбрано, чтобы явно показать его функциональное подобие единому протоколу доступа к управляющей информации (CMISE), но в то же время оставить определение усовершенствованной с точки зрения стоимости и производительности модели сервисов для отдельного вида работы по реализации интероперабельности, вместо требуемой ИСО/МЭК реализации единого протокола доступа к управляющей информации (CMIP).

Вызовы сервисов должны соответствовать общему протоколу обмена информацией между медицинскими приборами (CMDIP), определение которого не входит в область применения настоящего стандарта. Протокол CMDIP определен в ИСО/ИИЭР 11073-20101.

Единый протокол доступа к управляющей информации (CMIP, определенный в стандарте ИСО/МЭК 9596-1) рассматривается, как допустимый пример реализации в протоколе обобщенных сервисов управления объектом, предоставляемых протоколом CMDISE, определенным в настоящем стандарте. Дополнительные (например, оптимизированные по стоимости и производительности) реализации сервисов явно не исключаются и могут быть найдены в стандарте (стандартах), зависящих от настоящего.

8.3.1 Концептуальная архитектура коммуникационных систем

Концептуальная архитектура коммуникационных приборов, для которых настоящий стандарт предоставляет определения классов, представлена на рисунке 8.1. На этом рисунке показано, как сервисы, предоставляемые протоколом CMDISE, связаны с определениями классов, определенными в настоящем стандарте, и с приложениями, использующими эти определения классов.

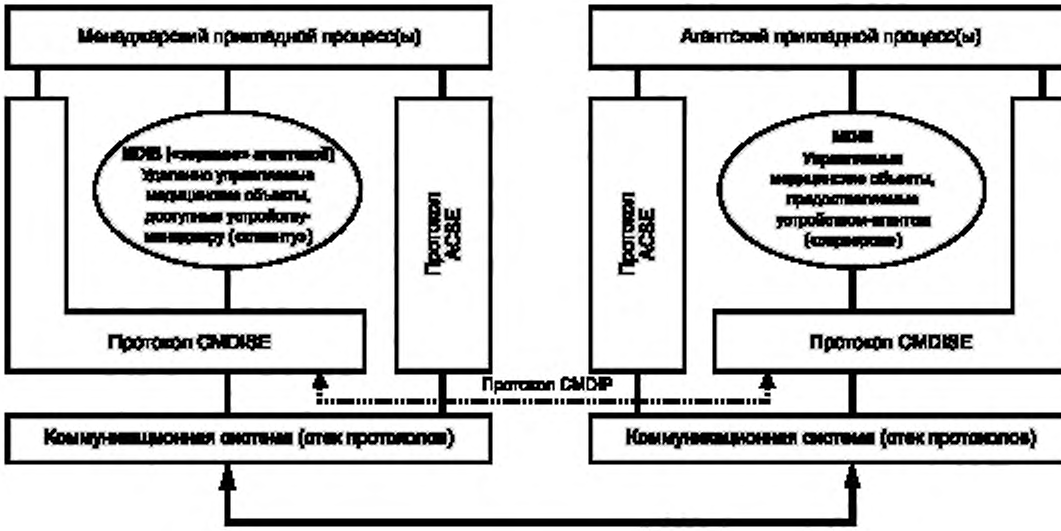


Рисунок 8.1 — Концептуальная архитектура коммуникационных систем

Прикладные процессы, представленные на рисунке 8.1, являются пользователями сервисов; протоколы CMDISE и ACSE — предоставляют сервисы.

Эта архитектурная модель показывает следующие компоненты, которые непосредственно используются или на которые имеются ссылки в настоящем стандарте:

- протокол CMDISE, который предоставляет определенные в разделе 8 сервисы прикладным процессам. Сервисы сопоставлены сообщениям, определенным протоколом CMDIP;
- протокол ACSE, который предоставляет сервисы для установления логического соединения между системами медицинских приборов (MDS). Модуль ACSE и соответствующий протокол определены в ИСО/МЭК 8649 и ИСО/МЭК 8650-1. Он предоставляет примитивные сервисы для:
  - запроса ассоциации и принятия запроса ассоциации;

- запроса прекращения ассоциации и принятие запроса прекращения ассоциации;
- аварийного прерывания ассоциации;
- стандартизированная база данных MDIB, которая содержит экземпляры управляемых медицинских объектов, как это определено в настоящем стандарте (т. е. в информационной модели предметной области);
- стандартизированная коммуникационная система (т. е., стек протоколов или профиль коммуникации на уровнях с 1-го по 6-й модели взаимодействия открытых систем (ВОС)). В дополнение к основным возможностям коммуникации эта система может также предоставлять сервисы для синхронизации между многими приборами, фрагментации больших сообщений, управления потоками, и т. д.

Примечание — Определение контроля ассоциации и низкоуровневых профилей коммуникации лежит вне области применения настоящего стандарта. На рисунке 8.1 они показаны только для полноты картины.

Система-менеджер может использовать сервисы протокола CMDISE для того, чтобы построить и поддерживать локальную копию базы данных MDIB системы-агента. На рисунке 8.1 концептуально представлено зеркальное отражение базы данных MDIB агента в системе-менеджере. Необходимо отметить, что манипулирование объектами всегда выполняется агентскими прикладными процессами.

Приложение-агент помещает и обновляет классы и значения их атрибутов в агентской базе данных MDIB. Приложения, выполняющиеся в системе-менеджере, могут получить доступ к локальной копии базы данных MDIB в системе-менеджере для повышения эффективности работы с данными (но при этом соответствующие сервисы всегда воздействуют на представителей классов и в исходной агентской базе данных MDIB).

К объектам в базе данных MDIB можно получить удаленный доступ только при использовании сервисов протокола CMDISE.

Примечание — Организация доступа приложений к базе данных MDIB в конкретной системе — задача конкретной реализации и не определяется настоящим стандартом. То же самое относится и к каждой конкретной реализации базы данных MDIB.

В настоящем стандарте определяются следующие группы сервисов для управления медицинской информацией.

a) сервисы работы с классами. Настоящий стандарт определяет следующие сервисы для работы с управляемыми медицинскими объектами:

1) извлечь значения атрибута класса,
2) изменить значения атрибута класса,
3) вызвать определенные для класса методы,
4) создать и удалить экземпляры класса;

b) сервисы оповещения. Настоящий стандарт определяет следующий сервис, который позволяет передавать уведомления о событиях между коммуникационными системами:

1) сообщать об асинхронных событиях, происходящих внутри объекта;

c) сервисы, используемые системой-менеджером. Система-менеджер (т. е. клиент) вызывает сервисы работы с классами, чтобы определить конфигурацию системы-агента (т. е. сервера), извлечь значения атрибутов медицинских классов (например, результаты измерений), и управлять агентом. Если требуется, система-менеджер выдает подтверждения получения сообщений сервисам оповещения;

d) сервисы, используемые системой-агентом. Система-агент вызывает сервисы оповещения, чтобы сообщить о возникновении определенных событий. Система-агент предоставляет результат выполнения сервисам работы с классами;

e) сервисы, используемые гибридной системой. Гибридная система вызывает и сервисы работы с классами, и сервисы оповещения в зависимости от потребности приложения. Гибридная система также порождает ответы и сервисам оповещения и сервисам работы с классами.

8.4 Общее определение сервисов для работы с управляемыми медицинскими объектами

Коммуникационные системы, соответствующие настоящему стандарту, предоставляют или используют сервисы для работы с управляемыми медицинскими объектами, определенные в настоящем подразделе. То, в какой степени эти сервисы используются конкретной коммуникационной системой, зависит от ее роли и назначения. Классы, определенные в информационной модели предметной области, определяют, в какой степени они могут использовать эти сервисы и в какой степени они могут управляться этими сервисами.

Параметры сервисов, определенные в настоящем подразделе, лишь минимально закрывают потребности коммуникационных систем. Другими словами, в конкретной реализации модели сервисов могут добавляться расширенные функциональные возможности (например, аутентификация, управление доступом, расширенные возможности выбора класса), которые потребуют введения дополнительных параметров в базовые классы, определяемые настоящим стандартом. Определение протокола должно определять фактические типы значений параметров и их использование в сообщениях протокола.

8.4.1 Сервис EVENT REPORT (Отчет о событии)

Сервис EVENT REPORT используется, чтобы сообщить о событии, относящемся к конкретному представителю управляемого класса. Сервис может использоваться в режимах с подтверждением и без подтверждения. В режиме с подтверждением вызов сервиса EVENT REPORT требует получения ответа.

Пример — Монитор SpO2 (т.е., агент в приложении, ведущем лог данных) может обнаружить отказ датчика. Прикладной процесс использует сервис EVENT REPORT, предоставляемый протоколом CMDISE, чтобы уведомить связанный с ним процесс-менеджер о выполнении условия возникновения технической тревоги.

В отличие от всех прочих сервисов, сервис EVENT REPORT, будучи сервисом оповещения, вызывается прикладным процессом-агентом; прикладной процесс-менеджер принимает сообщения сервиса и отвечает на них.

Параметры вызова сервиса EVENT REPORT представлены в таблице 8.1.

Таблица 8.1 — Параметры вызова сервиса EVENT REPORT

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Уникальный идентификатор (например, номер), связанный с конкретным экземпляром сервиса для того, чтобы его можно было отличить от других выполняемых в это же время экземпляров данного сервиса
Режим (Mode)	С подтверждением или без подтверждения. В случае режима с подтверждением требуется получения ответа на сообщение вызова сервиса
Класс (Object Class)	Идентифицирует класс, генерирующий событие (со значениями, определенными в номенклатуре/словаре классов)
Представитель класса (Object Instance)	Идентифицирует конкретный экземпляр класса, генерирующий событие
Момент наступления события (Event Time)	Момент времени, когда произошло событие
Тип события (Event Type)	Идентифицирует тип события (со значениями, определенными в номенклатуре/словаре классов)
Информация о событии (Event Information)	Необязательный параметр, содержащий дополнительную информацию о событии, определяемую типом события. Значение данного параметра и потребность в нем определяются классом, генерирующим событие

Параметры ответа на вызов сервиса EVENT REPORT в режиме с подтверждением представлены в таблице 8.2.

Таблица 8.2 — Параметры ответа на вызов сервиса EVENT REPORT

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Возвращает значение уникального идентификатора вызова сервиса EVENT REPORT, к которому относится ответ
Класс (Object Class)	Необязательный параметр. Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Представитель класса (Object Instance)	Необязательный параметр. Имеет то же значение, что и в соответствующем сообщении вызова сервиса

Окончание таблицы 8.2

Параметр	Описание
Текущее время (Current Time)	Необязательный параметр. Значение текущего момента времени
Тип события (Event Type)	Необязательный параметр. Имеет то же значение, что и в соответствующем сообщении вызова сервиса Момент времени, когда произошло событие
Ответная информация о событии (Event Reply Information)	Необязательный параметр, содержащий дополнительную ответную информацию о событии, определяемую типом события

Если вызов сервиса EVENT REPORT не может быть обработан, возвращается сообщение об ошибке с указанием типа отказа.

8.4.2 Сервис GET (Получить значение)

Сервис GET позволяет извлечение значений атрибутов экземпляров управляемого класса. Сервис GET всегда работает в режиме с подтверждением. Ответ на вызов сервиса GET содержит запрошенные данные (или сообщение об ошибке).

Пример — Приложение-менеджер хранения данных может использовать сервис GET, чтобы получить серийный номер и информацию о версии от связанного измерительного прибора.

Процесс-менеджер вызывает сервис GET (т. е., посылает сообщение с запросом сервиса GET), чтобы получить значения одного, нескольких или всех атрибутов выбранного экземпляра управляемого класса в агенте. В результате предоставления сервиса GET в ответ возвращается список, содержащий запрошенные значения атрибутов.

Параметры вызова сервиса GET представлены в таблице 8.3.

Таблица 8.3 — Параметры вызова сервиса GET

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Уникальный идентификатор (например, номер), относящийся к конкретному экземпляру сервиса
Класс (Object Class)	Идентифицирует класс, который содержит атрибуты, значения которых запрашиваются (со значениями, определенными в номенклатуре/словаре классов)
Представитель класса (Object Instance)	Идентифицирует представителя класса, который содержит атрибуты, значения которых запрашиваются
Список идентификаторов атрибутов (Attribute Identifier List)	Список уникальных идентификаторов атрибутов (со значениями, определенными в номенклатуре/словаре классов), значения которых запрашиваются

Параметры ответа на вызов сервиса GET представлены в таблице 8.4.

Таблица 8.4 — Параметры ответа на вызов сервиса GET

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Возвращает значение уникального идентификатора вызова сервиса GET, к которому относится ответ
Класс (Object Class)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Представитель класса (Object Instance)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Список атрибутов (Attribute List)	Список пар идентификатор — значение для атрибутов, значения которых запрашиваются

Если вызов сервиса GET не может быть обработан, возвращается сообщение об ошибке с указанием типа отказа.

8.4.3 Сервис SET (Установить значение)

Сервис SET позволяет изменение значений атрибутов представителей управляемых классов. Сервис SET может использоваться как в режиме с подтверждением, так и в режиме без подтверждения. В режиме с подтверждением вызов сервиса SET требует получения ответа.

Пример — Центральный компьютер может использовать сервис SET, чтобы назначить текущую дату и время в приборе, который был недавно подключен к сети.

Процесс-менеджер вызывает сервис SET (т. е. посылает сообщение с запросом на выполнение сервиса SET), чтобы изменить значение одного или нескольких атрибутов выбранных объектов в агенте. Для каждого атрибута, значение которого должно быть изменено, запрос содержит уникальный идентификатор атрибута, оператор изменения (чтобы выбрать, должно ли значение атрибута быть изменено, добавлено к списку, удалено из списка, или установлено в значение по умолчанию) и (опционально) значение атрибута.

Параметры вызова сервиса SET представлены в таблице 8.5.

Таблица 8.5 — Параметры вызова сервиса SET

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Уникальный идентификатор (например, номер), относящийся к конкретному экземпляру сервиса SET
Режим (Mode)	С подтверждением или без подтверждения
Класс (Object Class)	Идентифицирует класс, который содержит атрибуты, значения которых должны быть изменены (со значениями, определенными в номенклатуре/словаре классов)
Представитель класса (Object Instance)	Идентифицирует представителя класса, который содержит атрибуты, значения которых должны быть изменены
Список изменений (Modification List)	Список записей типа оператор изменения — идентификатор атрибута — значение атрибута, где оператор изменения может принимать значения: replace, addValues, removeValues, setToDefault

Параметры ответа на вызов сервиса SET в режиме с подтверждением представлены в таблице 8.6.

Таблица 8.6 — Параметры ответа на вызов сервиса SET

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Возвращает значение уникального идентификатора вызова сервиса GET, к которому относится ответ
Класс (Object Class)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Представитель класса (Object Instance)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Список атрибутов (Attribute List)	Необязательный атрибут, содержащий список пар идентификатор — значение для атрибутов, значения которых изменяются

Если вызов сервиса SET не может быть обработан, возвращается сообщение об ошибке с указанием типа отказа.

8.4.4 Сервис ACTION (Выполнить действие)

Сервис ACTION позволяет вызвать предопределенный метод (то есть процедуру) управляемого медицинского объекта. Сервис ACTION может использоваться как в режиме с подтверждением, так и в режиме без подтверждения. В режиме с подтверждением вызов сервиса ACTION требует получения ответа.

Пример — Система мониторинга может использовать сервис ACTION, чтобы начать процедуру калибровки измерительного прибора.

Определение методов и, в дальнейшем, поведения класса зависит от спецификации класса, но не от сервиса ACTION. Спецификация класса в информационной модели предметной области определяет все доступные методы класса, которые могут быть выполнены посредством вызова сервиса ACTION, наряду с их определенными типами значений параметров и возвращаемого результата.

Параметры вызова сервиса ACTION представлены в таблице 8.7.

Таблица 8.7 — Параметры вызова сервиса ACTION

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Уникальный идентификатор (например, номер), относящийся к конкретному экземпляру сервиса ACTION
Режим (Mode)	С подтверждением или без подтверждения
Класс (Object Class)	Идентифицирует класс, который должен выполнить действие (со значениями, определенными в номенклатуре/словаре классов)
Представитель класса (Object Instance)	Идентифицирует представителя класса, который должен выполнить действие
Тип действия (Action Type)	Идентифицирует тип действия (со значениями, определенными в номенклатуре/словаре классов)
Информация о действии (Action Information)	Дополнительные параметры действия, определяемые типом действия

Параметры ответа на вызов сервиса ACTION в режиме с подтверждением представлены в таблице 8.8.

Таблица 8.8 — Параметры ответа на вызов сервиса ACTION

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Возвращает значение уникального идентификатора вызова сервиса ACTION, к которому относится ответ
Класс (Object Class)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Представитель класса (Object Instance)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Тип действия (Action Type)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Возвращаемый результат действия (Action Reply)	Необязательный параметр, содержащий возвращаемый результат выполнения действия, определяемый типом действия

Если вызов сервиса ACTION не может быть обработан, возвращается сообщение об ошибке с указанием типа отказа.

8.4.5 Сервис CREATE (Создать новый экземпляр класса)

Сервис CREATE используется для создания нового экземпляра управляемого медицинского объекта. С использованием данного сервиса можно определить и атрибуты нового объекта. Сервис CREATE всегда используется в режиме с подтверждением и требует ответа на вызов сервиса.

Пример — Приложение, ведущее лог данных, может использовать сервис CREATE, чтобы создать экземпляр дополнительного сервиса (например, экземпляр класса Scanner) в агенте мониторинга измерений (т.е. в системе-агенте). Этот сканер обрабатывает все числовые данные измерения и ежеминутно посылает сообщение (то есть отчет о событии) приложению, строящему графическое отражение результатов измерений.

Сервис CREATE допускает создание представителей не всякого класса в базе данных MDIB системы-агента. Система, соответствующая требованиям настоящего стандарта, должна четко определять, экземпляры каких классов могут создаваться динамически.

Параметры вызова сервиса CREATE представлены в таблице 8.9.

Таблица 8.9 — Параметры вызова сервиса CREATE

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Уникальный идентификатор (например, номер), относящийся к конкретному экземпляру сервиса CREATE
Класс (Object Class)	Идентифицирует класс объекта, который должен быть создан (со значениями, определенными в номенклатуре/словаре классов)
Класс-агрегатор (Superior Object Class)	Идентифицирует класс, к которому принадлежит объект, агрегирующий создаваемый экземпляр (в соответствии с иерархией вхождения)
Представитель класса-агрегатора (Superior Object Instance)	Идентифицирует объект, агрегирующий создаваемый экземпляр (в соответствии с иерархией вхождения)
Список атрибутов (Attribute List)	Необязательный параметр, содержащий список пар идентификатор атрибута — значение атрибута для установления значений по умолчанию атрибутам создаваемого экземпляра

Параметры ответа на вызов сервиса CREATE представлены в таблице 8.10.

Таблица 8.10 — Параметры ответа на вызов сервиса CREATE

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Возвращает значение уникального идентификатора вызова сервиса CREATE, к которому относится ответ
Класс (Object Class)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Представитель класса (Object Instance)	Идентификатор объекта, назначаемый протоколом CMDISE в соответствии с правилом связывания имен

Если вызов сервиса CREATE не может быть обработан, возвращается сообщение об ошибке с указанием типа отказа.

8.4.6 Сервис DELETE (Удалить экземпляр класса)

Сервис DELETE используется для удаления экземпляров управляемых медицинских объектов. Сервис DELETE всегда используется в режиме с подтверждением и требует ответа на вызов сервиса.

Пример — Когда приложение, ведущее лог данных, из предыдущего примера больше не нуждается в данных измерения, предоставленных системой-агентом, оно использует сервис DELETE, чтобы удалить экземпляр дополнительного сервиса (т. е. экземпляр класса Scanner).

Сервис DELETE допускает удаление представителей не всякого класса в базе данных MDIB системы-агента. Система, соответствующая требованиям настоящего стандарта, должна четко определять, экземпляры каких классов могут удаляться динамически.

Параметры вызова сервиса DELETE представлены в таблице 8.11.

Таблица 8.11 — Параметры вызова сервиса DELETE

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Уникальный идентификатор (например, номер), относящийся к конкретному экземпляру сервиса DELETE
Класс (Object Class)	Идентифицирует класс, объект которого должен быть удален (со значениями, определенными в номенклатуре/словаре классов)
Представитель класса (Object Instance)	Идентификатор экземпляра, подлежащего удалению

Параметры ответа на вызов сервиса DELETE представлены в таблице 8.12.

Таблица 8.12 — Параметры ответа на вызов сервиса DELETE

Параметр	Описание
Идентификатор вызова (Invoke Identifier)	Возвращает значение уникального идентификатора вызова сервиса DELETE, к которому относится ответ
Класс (Object Class)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса
Представитель класса (Object Instance)	Имеет то же значение, что и в соответствующем сообщении вызова сервиса

Если вызов сервиса CREATE не может быть обработан, возвращается сообщение об ошибке с указанием типа отказа.

9 Номенклатура MDIB

Базу данных MDIB образует множество классов (объектов) и экземпляров классов, описанных в информационной модели предметной области и встречающихся в любом приборе коммуникационной системы. Каждый объект класса этой модели нуждается в уникальной идентификации. Полный набор терминов образует номенклатуру базы данных MDIB или словарь данных. Поскольку количество экземпляров обычно велико, необходима структурированная схема идентификации, то есть — номенклатура. Номенклатура базы данных MDIB включает несколько тысяч терминов, относящихся к элементам объектно-ориентированного моделирования, демографическим данным пациента, описаниям приборов, значениям измерений, методам измерений, местоположениям измерений, сигнальной информации, и т. д. Она открыта для дополнения по мере развития медицины и техники при сохранении структуры и терминов в пределах установленной системы терминов.

Номенклатура также поддерживает развитие словаря данных, не зависящего от языка, со схемой кодирования для обеспечения легкого и быстрого компьютерного доступа.

Словарь данных MDIB представлен в стандарте ИСО/ИИЭР 11073-10101. Он содержит термины (т. е. систематизированные наименования), описания и коды для следующих целевых категорий:

- элементы объектно-ориентированного моделирования, следующие из информационной модели предметной области;
- медицинские приборы и системы приборов;
- единицы измерений;
- метрики (измерения и перечисления);
- участки тела (т. е. спецификации местоположения измерений);
- предупреждения об опасности;
- внешние номенклатуры.

Каждый из соответствующих разделов начинается с детального описания того, как построить систематическое наименование для соответствующей целевой категории.

10 Модель соответствия

10.1 Применимость

Настоящий стандарт предполагается использовать совместно с другими базовыми стандартами или в качестве ссылки в других базовых стандартах семейства ИСО/ИИЭР 11073 для определения приложений (например, для обмена базами данных измерений показателей жизненно-важных функций) или для определения функциональных профилей коммуникаций (например, профилей интероперабельности медицинских приборов).

Подобные дополнительные спецификации и стандарты необходимы для создания всех требующихся условий для осуществления реализации или построения системы на основании настоящего стандарта.

Реализация или система может соответствовать следующему элементу настоящего стандарта, содержащему четкие определения:

- иерархии классов в DIM и определению объектов (т. е. определению атрибутов объектов, уведомлений, методов и типов данных).

Тем не менее, соответствие одному лишь этому элементу не обеспечивает интероперабельность приложений или медицинских приборов.

Предполагается, что в стандартах для определенных приложений или функциональных профилей коммуникаций должна быть определена надлежащая модель соответствия, включающая в себя конкретные требования соответствия настоящему стандарту, связанные с представлением показателей жизненно-важных функций. От данных стандартов также требуются определения дополнительных критериев соответствия семантического и динамического поведения реализации, что не входит в область применения настоящего стандарта.

Соответствие определениям настоящего стандарта указывается только для интерфейсов соответствующего приложения или системы. Только на таком интерфейсе рассматривается на соответствие поведение системы или приложения. Детали реализации, которые невозможно обнаружить внешне не являются предметом спецификации соответствия.

Пример — Медицинский прибор, учавствующий в коммуникациях, использует классы объектов и определения типов, заданные в настоящем стандарте, для передачи данных другим приборам. Он может соответствовать настоящему стандарту, даже если внутренне его реализация не является объектно-ориентированной.

10.2 Спецификация соответствия

Настоящий стандарт, посвященный представлению показателей жизненно-важных функций обладает высокой степенью гибкости в применении описанной в нем модели для определенного медицинского прибора, в особенности для следующих областей:

- построение информационной модели конкретного прибора;
- использование атрибутов, диапазонов значений и доступа;
- использование дополнительных (расширенных) сервисов коммуникаций (т. е. сканнеров), периодов сканирования и конфигурируемости сканнеров.

Для обеспечения интероперабельности приложений и систем, реализация, основанная на настоящем стандарте, должна сопровождаться специфическими подробностями того, каким образом применяются определения настоящего стандарта.

Подобные спецификации должны быть предоставлены в форме набора заявлений о соответствии реализации (ICS). ICS представляет из себя разновидность перечня данных, в котором раскрываются подробности определенной реализации и указывается, какие свойства она предоставляет. Определенные приложения или функциональные профили коммуникаций, основанные на настоящем стандарте, должны определять более конкретные требования к соответствию в дополнение к или в качестве замены заявлений ICS, определенных в настоящем стандарте.

Примечание — Заявления соответствия, определенные в 10.3, служат для ознакомления с деталями реализации. Тем не менее, их одних не достаточно для обеспечения интероперабельности приборов и приложений. Для подобной интероперабельности следует учитывать дополнительные спецификации (например, длительность, задержки, предположения о нагрузке системы). Данные спецификации не входят в область применения настоящего стандарта.

10.3 Декларации соответствии реализации (ICS)

10.3.1 Общий формат

Декларации ICS должны быть представлены в форме таблиц. Шаблоны для подобных таблиц ICS представлены в 10.3.2—10.3.7. Таблицы должны быть заполнены и предоставлены в качестве полного документа заявления о соответствии. Как правило, заголовки столбцов ICS таблиц содержат следующую информацию:

- индекс, являющийся идентификатором (например, номером) определенного свойства;
- свойство, т. е. краткое описание характеристики, для которой должно быть сделано заявление о соответствии;
- ссылка, т. е. ссылка на определение свойства (может быть не заполнено);
- статус, устанавливающий требования соответствия (т. е., требования для претендующей на соответствие реализации, связанные с рассматриваемым свойством). Для некоторых случаев настоящий стандарт не указывает требования соответствия, но, так или иначе, требует определение статуса конкретного свойства;

- поддержка, которая заполняется специалистом по внедрению (реализации) и содержит характеристики свойств реализации;
- комментарий, содержащий дополнительную информацию, предоставленную специалистом по внедрению.

Значения столбцов Статус и Поддержка могут охватывать диапазон от простых до сложных записей. Ниже приведены примеры простых значений:

m — обязательное;

o — не обязательное (дополнительное);

x — запрещенное;

c — условное;

n/a — не применимо.

Более сложные выражения или конкретные списки элементов определены в конкретной ICS таблице.

10.3.2 Общая декларация о соответствии реализации (ICS)

В общей ICS наивысшего уровня специалист по внедрению указывает версии/исполнения, поддерживаемые реализацией, а также некоторые определения высокого уровня для поведения системы.

В таблице 10.1 приведена общая ICS.

Таблица 10.1 — Общая ICS

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
GEN-1	Описание реализации	—	Идентификация прибора/приложения. Описание функционала		
GEN-2	Новая версия стандартного документа	(Стандартные документы)	(Набор существующих обновлений версий)	(Набор существующих обновлений версий)	
GEN-3	Отклонение от соответствия	—	Обеспечивает информацию о вероятных отклонениях от DIM (например, нестандартные атрибуты, объекты)	(Набор отклонений)	
GEN-4	Дерево состава объекта	6.2	Предоставляет диаграмму дерева состава объекта, демонстрирующую связи между экземплярами объектов, используемыми приложениями. Реализация, соответствующая требованиям, использует связи объектов только в соответствии с DIM		
GEN-5	Новая версия номенклатуры	(Стандартные документы)	(Набор существующих обновлений версий)	(Средства поддержки обновлений версий)	
GEN-6	Использование других схем номенклатуры	—	Используются ли в данной реализации номенклатурные коды из других стандартных схем кодирования?	Да/Нет (Если Да, то список других номенклатур)	Следует отметить, что использование других номенклатур значительно сказывается на интероперабельности

Окончание таблицы 10.1

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
GEN-7	Кодирование структуры данных	—	—	Описание метода кодирование для структур данных ASN.1	
GEN-8	Динамические экземпляры объектов	—	Набор экземпляров объектов является статическим или динамическим во время работы?	Статический/ Динамический	
GEN-9	Использование иных объектов	—	В реализации используются объекты, не определенные в DIM?	Да/Нет [Если Да, то привести пояснения в DIM MOCICS (см. 10.3.4)]	
GEN-10	Использование иных расширений номенклатуры	—	В реализации используются иные расширения номенклатуры? Иные раширения номенклатуры разрешены только в случае, если в стандартную номенклатуру не входят специфические термины необходимые для приложения	Да/Нет (Если Да, привести пояснения в соответствующей ICS)	
GEN-11	Коммуникационный профиль и аппаратное обеспечение	—	Описание коммуникационного профиля и требований аппаратного обеспечения для взаимодействия (применимо только к коммуницирующим приборам)		
GEN-12	Формат файлов и носители данных		Описание формата файлов, применяемого для архивации данных показателей жизненно важных функций; определение поддерживаемых носителей данных (применимо только к архивным приложениям)		
GEN-13	ACSE	ИСО/ МЭК 8649 ИСО/ МЭК 8650-1	Использование протокола ACSE (применимо только к коммуницирующим системам)	Уточните использование дополнительных полей в блоках данных протокола (блоках PDU) ACSE	

Для каждой реализации должна быть приведена одна общая ICS.

10.3.3 Декларация о соответствии реализации для поддержки сервисов

ICS для поддержки сервисов определяет какие из сервисов, заданных в моделе сервисов, реализованы. Подобная ICS необходима только для коммуницирующих приборов.

ICS для поддержки сервисов приведена в таблице 10.2.

Таблица 10.2 — ICS для поддержки сервисов

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
SRV-1	Сервис GET (Получить)	8.4.2	o		
SRV-2	Сервис SET (Установить)	8.4.3	o		
SRV-3	Подтвержденный сервис SET	8.4.3	o		
SRV-4	Сервис EVENT REPORT (Отчет о событии)	8.4.1	m		
SRV-5	Подтвержденный сервис EVENT REPORT	8.4.1	o		
SRV-6	Сервис ACTION (Действие)	8.4.4	o		
SRV-7	Подтвержденный сервис ACTION	8.4.4	o		
SRV-8	Сервис CREATE (Создать)	8.4.5	o		
SRV-9	Сервис DELETE (Удалить)	8.4.6	o		

Столбец Поддержка заполненной таблицы должен быть определен, если реализация вызывает сервис (например, отправляет GET PDU), предоставляет сервис (например, обрабатывает полученный GET PDU) или не реализует сервис вообще.

Помимо этого, должны быть перечислены специфические ограничения (например, если определенный сервис ограничен одним классом объекта).

10.3.4 Декларация о соответствии реализации класса управляемых объектов (MOC) DIM

Декларация о соответствии реализации MOC DIM определяет, какие из управляемых медицинских объектов (не являющихся базовыми классами) используются в реализации. Таблица 10.3 является всего лишь шаблоном. Для каждого поддерживаемого реализацией объекта должен быть заполнен один ряд.

Таблица 10.3 — Шаблон декларации о соответствии реализации MOC DIM

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
MOC-[1-n]	Имя объекта и его идентификатор (OID)	Ссылка на раздел настоящего стандарта, в котором определен данный объект	Реализован (внедрен)	Указать ограничения, например, максимальное число поддерживаемых экземпляров. Сервисы CREAT/DELETE поддерживаются	

Если в реализации применяются иные объекты, то их также следует указать в декларации о соответствии реализации MOC DIM. К декларации о соответствии должно быть добавлено отдельное определение, на которое можно разместить ссылку в столбце Ссылка.

Столбец Поддержка должен указывать на конкретные ограничения реализации объекта. В частности, в нем должно быть указано можно ли динамически создавать/удалять объекты посредством сервисов CREATE/DELETE.

В дополнении к декларации о соответствии реализации MOC DIM должна быть приведена диаграмма состава объекта (т. е. диаграмма экземпляров класса), с помощью которой можно изучить иерархию, использованную в реализации.

10.3.5 Декларация о соответствии реализации атрибутов класса управляемых объектов (MOC)

Для каждого поддерживаемого объекта, как это определено в декларации о соответствии реализации MOC DIM, в ней должны быть представлены атрибуты MOC и указано, какие атрибуты используются/поддерживаются в реализации, включая любые унаследованные атрибуты. Таблица 10.4 является всего лишь шаблоном.

Таблица 10.4 — Шаблон декларации о соответствии реализации атрибутов МОС

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
ATTR-*x-n*	Имя атрибута и идентификатор (ID) атрибута	Ссылка на раздел настоящего стандарта, в котором определен данный объект	m/o/c (см. 7.1.1 для объяснения этих сокращений)	Получение доступа (т. е. GET, GETGRP, SET, SCAN, SCANGRP, ER, CRER, см. третий абзац после этой таблицы). Диапазоны значений Дополнительные ограничения Статическое/динамическое значение	

Символ *x* в столбце Индекс является ID управляемого объекта, для которого формируется таблица (т. е. индексом управляемого объекта, как он указан в ICS для МОС в DIM). Для каждого поддерживаемого управляемого объекта предоставляется одна таблица.

Символ *n* в столбце Индекс является обычным серийным номером (1..*m*).

Если в реализации предусмотрены сервисы доступа для атрибутов, то в столбце Поддержка должны быть заполнены поля спецификации доступа к атрибутам. (Другими словами, эти поля не требуются для формата, предназначенного только для хранения.)

Поля имеют следующие значения:

GET — доступ к атрибуту можно получить индивидуально посредством сервиса GET;

GET-GRP — доступ к атрибуту, как к группе атрибутов, может быть получен посредством сервиса GET;

SET — атрибут может быть индивидуально модифицирован посредством сервиса SET;

SCAN — объект Scanner может получить индивидуальный доступ к атрибуту (запись в списке сканирования для индивидуального атрибута);

SCAN-GRP — объект Scanner может получить доступ к атрибуту (запись в списке сканирования для группы атрибутов);

ER — информация об изменениях в атрибутах сообщается посредством отчетов о событиях, поступающих от самого объекта-контейнера;

CR-ER — значение атрибута поступает в рамках уведомления о доступности объекта-контейнера (уведомление о создании объекта).

Столбец Поддержка должен также содержать диапазоны значений атрибутов (если применимо), советы о конкретных ограничениях для доступа к атрибутам или доступности атрибутов и информации, а также указание на то, является ли значение атрибута статическим или динамическим в данной реализации.

Примечание — Таблицы определений атрибутов, представленные в настоящем стандарте, устанавливают минимальный обязательный набор атрибутов для каждого объекта.

10.3.6 Декларация о соответствии реализации поведения класса управляемых объектов (МОС)

Декларация о соответствии реализации поведения МОС устанавливает все реализованные объектные методы, которые могут быть вызваны посредством сервиса ACTION. Таблица 10.5 является всего лишь шаблоном. Для каждого объекта, поддерживающего специальные методы, должна быть приведена одна таблица.

Таблица 10.5 — Шаблон для ICS поведения МОС

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
ACT-*x-n*	Имя метода и ID метода	Ссылка на раздел настоящего стандарта, в котором определен данный объект		Специфические ограничения	

Символ *x* в столбце Индекс является ID управляемого объекта, для которого предоставляется таблица (т. е. индексом управляемого объекта, как он указан в ICS для МОС в DIM). Для каждого управляемого объекта, поддерживающего определенные объектные методы (т. е. действия), предоставляется одна таблица.

Символ *n* в столбце Индекс является обычным серийным номером (1..*m*).

В столбце Поддержка должны быть указаны любые ограничения для метода.

10.3.7 Декларация о соответствии реализации уведомлений МОС

Декларация о соответствии реализации уведомлений МОС устанавливает все реализованные уведомления (как правило, в форме сервиса EVENT REPORT), генерируемые поддерживаемыми объектами. Таблица 10.6 является всего лишь шаблоном. Для каждого объекта, поддерживающего специальные объектные уведомления, должна быть приведена одна таблица.

Таблица 10.6 — Шаблон для декларации о соответствии реализации уведомлений МОС

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
NOTI-*x-n*	Имя уведомления и ID уведомления	Ссылка на раздел настоящего стандарта, в котором определен данный объект		Специфические ограничения	

Символ *x* в столбце Индекс является ID управляемого объекта, для которого предоставляется таблица (т. е. индексом управляемого объекта, как он указан в ICS для МОС в DIM). Для каждого управляемого объекта, поддерживающего определенные объектные уведомления (т. е. события), предоставляется одна таблица.

Символ *n* в столбце Индекс является обычным серийным номером (1..*m*).

В столбце Поддержка должны быть указаны любые ограничения для уведомления.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов и документов национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта, документа	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ЕКС ЕС 1064	—	*
ЕКС ЕПС 12052	—	*
ИИЭР Std 1073	—	*
IETF RFC 1155	—	*
ИСО 639-1	—	*
ИСО 639-2	—	*
ИСО 3166-1	—	*
ИСО 3166-2	—	*
ИСО 3166-3	—	*
ИСО 8601	IDT	ГОСТ ИСО 8601—2001 «Система стандартов по информации, библиотечному и издательскому делу. Представление дат и времени. Общие требования»
ИСО 15225	IDT	ГОСТ Р ИСО 15225—2003 «Номенклатура. Номенклатура данных по медицинским изделиям для информационного обмена»
ИСО/МЭК 646	—	*
ИСО/МЭК 2022	—	*
ИСО/МЭК 5218	—	*
ИСО/МЭК 7498-1	IDT	ГОСТ Р ИСО/МЭК 7498-1—99 «Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель»
ИСО/МЭК 7498-2	IDT	ГОСТ Р ИСО/МЭК 7498-2—99 «Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура защиты информации»
ИСО/МЭК 7498-3	IDT	ГОСТ Р ИСО/МЭК 7498-3—97 «Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 3. Присвоение имен и адресация»
ИСО/МЭК 7498-4	IDT	ГОСТ Р ИСО/МЭК 7498-4—99 «Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 4. Основы административного управления»
ИСО/МЭК 8649	—	*
ИСО/МЭК 8650-1	—	*
ИСО/МЭК 8650-2	—	*

Продолжение таблицы ДА.1

Обозначение ссылочного международного стандарта, документа	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8824-1	IDT	ГОСТ Р ИСО/МЭК 8824-1—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации»
ИСО/МЭК 8824-2	IDT	ГОСТ Р ИСО/МЭК 8824-2—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 2. Спецификация информационного объекта»
ИСО/МЭК 8859-n	—	*
ИСО/МЭК 9545		ГОСТ Р ИСО/МЭК 9545—98 «Информационная технология. Взаимосвязь открытых систем. Структура прикладного уровня»
ИСО/МЭК 9595	IDT	ГОСТ Р ИСО/МЭК 9595—99 «Информационная технология. Взаимосвязь открытых систем. Определение общих услуг информации административного управления»
ИСО/МЭК 9596-1	IDT	ГОСТ Р ИСО/МЭК 9596-1—99 «Информационная технология. Взаимосвязь открытых систем. Протокол информации административного управления. Часть 1. Спецификация протокола»
ИСО/МЭК 10040		ГОСТ Р ИСО/МЭК 10040—99 «Информационная технология. Взаимосвязь открытых систем. Общее описание административного управления систем»
ИСО/МЭК 10164-1		ГОСТ Р ИСО/МЭК 10164-1—99 «Информационная технология. Взаимосвязь открытых систем. Административное управление систем. Функция административного управления объектами»
ИСО/МЭК 10164-2		ГОСТ Р ИСО/МЭК 10164-2—99 «Информационная технология. Взаимосвязь открытых систем. Административное управление систем. Функция административного управления состояниями»
ИСО/МЭК 10164-3	—	*
ИСО/МЭК 10164-4	—	*
ИСО/МЭК 10164-5	—	*
ИСО/МЭК 10164-6	—	*
ИСО/МЭК 10164-7	—	*
ИСО/МЭК 10164-8	—	*
ИСО/МЭК 10164-9	—	*
ИСО/МЭК 10164-10	—	*
ИСО/МЭК 10164-11	—	*
ИСО/МЭК 10164-12	—	*
ИСО/МЭК 10164-13	—	*
ИСО/МЭК 10164-14	—	*

Окончание таблицы ДА.1

Обозначение ссылочного международного стандарта, документа	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 10165-1		ГОСТ Р ИСО/МЭК 10165-1—2001 «Информационная технология. Взаимосвязь открытых систем. Структура информации административного управления. Часть 1. Модель информации административного управления»
ИСО/МЭК 10165-2		ГОСТ Р ИСО/МЭК 10165-2—2001 «Информационная технология. Взаимосвязь открытых систем. Структура информации административного управления. Часть 2. Определение информации административного управления»
ИСО/МЭК 10646-1	—	*
ИСО/ИИЭР 11073-10101	—	*
ИСО/ИИЭР 11073-20101	—	*
NEMA PS 3	—	*

*Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:

- IDT — идентичные стандарты.

УДК 004:61:006.354 ОКС 35.240.80 П85 ОКСТУ 4002

Ключевые слова: здравоохранение, информатизация здоровья, структуры данных, персональные медицинские приборы, передача данных, информационное взаимодействие, информационная модель, предметная область

Редактор *А.Ф. Колчин*
Технический редактор *В.Ю. Фотиева*
Корректор *С.В. Смирнова*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 19.05.2016. Подписано в печать 14.06.2016. Формат 60×84 1/8. Гарнитура Ариал
Усл. печ. л. 20,93. Уч.-изд. л. 20,00. Тираж 29 экз. Зак. 1461.

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru