ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56844— 2015 /ISO/IEEE 11073-20101: 2004

Информатизация здоровья

ИНФОРМАЦИОННОЕ ВЗАИМОДЕЙСТВИЕ С ПЕРСОНАЛЬНЫМИ МЕДИЦИНСКИМИ ПРИБОРАМИ

Часть 20101

Прикладные профили. Базовый стандарт

(ISO/IEEE 11073-20101:2004, 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 англоязычной версии международного документа, указанного в пункте 4

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

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 28 декабря 2015 г. № 2232-ст

4 Настоящий стандарт идентичен международному стандарту ISO/IEEE 11073-20101:2004 «Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами. Часть 20101. Прикладные профили. Базовый стандарт» (ISO/IEEE 11073-20101:2004 «Health informatics — Point-of-care medical device communication — Part 20101: Application profiles — Base standard», IDT).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов и документов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в справочном 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 (*gost.e-ecolog.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 Обзор ... 1
1.1 Область применения ... 1
1.2 Назначение ... 2
1.3 Цели ... 2
1.4 Пользователи стандарта ... 2
2 Нормативные ссылки ... 2
3 Термины, определения и сокращения ... 3
3.1 Термины и определения ... 3
3.2 Сокращения ... 4
4 Условные обозначения ... 6
5 Обоснование ... 6
5.1 Коммуникационная модель ... 6
5.2 Информационная модель ... 7
6 Коммуникационная модель ... 7
6.1 Общие положения ... 7
6.2 Сервисный элемент управления ассоциацией (ACSE) ... 10
6.3 Протокол сеансового уровня ... 12
6.4 Протокол уровня представления ... 13
6.5 Протокол ROSE ... 13
6.6 Протокол CMDISE (CMDIP) ... 14
7 Информационная модель ... 14
7.1 Модель объекта ... 14
7.2 Модель формата ... 15
8 Соответствие ... 15
8.1 Область применения ... 15
8.2 Администрирование идентификатора объекта ... 15
8.3 Соответствие подмножества MDAP ... 15
8.4 Соответствие реализации ... 16
Приложение А (обязательное) Правила кодирования медицинских приборов (MDER) ... 17
Приложение В (обязательное) Распределение идентификаторов ... 26
Приложение С (справочное) Временная синхронизация ... 29
Приложение D (справочное) Динамическая модель ... 30
Приложение Е (обязательное) Абстрактный синтаксис ... 36
Приложение F (справочное) Примеры блоков PDU ... 53
Приложение G (справочное) Специализация ASN.1 ... 67
Приложение Н (справочное) Вопросы совместимости ... 70
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов и документов национальным стандартам Российской Федерации ... 72
Библиография ... 73

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информатизация здоровья

ИНФОРМАЦИОННОЕ ВЗАИМОДЕЙСТВИЕ С ПЕРСОНАЛЬНЫМИ МЕДИЦИНСКИМИ ПРИБОРАМИ

Часть 20101
Прикладные профили. Базовый стандарт

Health informatics. Point-of-care medical device communication.
Part 20101. Application profiles. Base standard

Дата введения — 2016—11—01

1 Обзор

Настоящий стандарт содержит восемь разделов:

- в разделе 1 описана область применения настоящего стандарта;
- в разделе 2 приведены ссылки на другие стандарты, используемые при применении настоящего стандарта;
- в разделе 3 приведены определения и сокращения;
- в разделе 4 представлены условные обозначения;
- в разделе 5 дано обоснование актуальности настоящего стандарта;
- в разделе 6 описана модель коммуникаций (т. е. протокол и сервис передачи данных);
- в разделе 7 описана информационная модель (модель объекта);
- в разделе 8 представлены требования соответствия.

Настоящий стандарт также содержит девять приложений:

- в приложении A (обязательное) определены специализированные правила кодирования медицинских приборов (MDER);
- в приложении B (обязательное) описано как распределены идентификаторы объектов;
- в приложении C представлены ссылки на протоколы временной синхронизации, используемые в настоящем стандарте;
- в приложении D представлены диаграммы переходов состояний для динамической модели;
- в приложении E (обязательное) описан абстрактный синтаксис, который предлагает расширения к стандартам, на которые опирается настоящий стандарт, такие как минимальные взаимодействия открытых систем (mOSI), которые ориентированы на настоящий стандарт.
- в приложении F представлены примеры некоторых PDU.
- в приложении G рассмотрены спецификации языка абстрактного синтаксиса 1 (ASN.1).
- в приложении H представлена информация о совместимости версий ASN.1 1988/90 г. и 1994 г.

1.1 Область применения

Область применения настоящего стандарта распространяется на сервисы и протоколы верхнего уровня (т. е. уровень приложения, уровень представления и сеансовый уровень взаимодействия открытых систем по ИСО), используемые для обмена информацией в соответствии со стандартами ИСО/ИИЭР 11073, регламентирующими коммуникации медицинских приборов (MDC).

Настоящий стандарт является базовым стандартом ИСО/ИИЭР 11073-20000 для прикладных профилей медицинских приборов (MDAP), что было согласовано с Европейским комитетом по стандартизации (ЕКС) и ИСО.

Издание официальное

1.2 Назначение

Назначением настоящего стандарта является определение приложения верхнего уровня коммуникаций медицинских приборов (MDC), т. е. профилей А-типа ИСО для обмена данными, которые определяются форматом языка данных медицинских приборов (MDDL), или же профилей F-типа ИСО (серия ИСО/ИИЭР 11073-10000).

1.3 Цели

Основная цель стандартов по прикладным профилям медицинских приборов (MDAP) — поддержать верхний уровень обмена данными между медицинскими приборами (MDC), осуществляемого на языке данных MDDL, между различными по типу и масштабу, будущими и применяемыми приборами, предназначенными для использования в местах оказания медицинской помощи (POC) в медицинских отделениях интенсивной терапии.

1.4 Пользователи стандарта

Стандарты MDAP в первую очередь предназначены для разработчиков программного обеспечения, создающих системы коммуникаций медицинских приборов (MDC) или пытающихся создать интерфейс для взаимодействия с ними.

Так как настоящая серия стандартов в основном основывается на профилях международной стандартизации, то ознакомление с соответствующими стандартами является желательным, если не обязательным условием. Рекомендуется ознакомиться, как минимум, со следующими стандартами:

a) комплекс ИСО/ИИЭР 11073, особенно стандарт ИИЭР Стд. 1073[1)], ИСО/ИИЭР 11073-10201 и стандарты нижнего уровня (например, ИСО/ИИЭР 11073-30200);

b) по архитектуре уровней взаимодействия открытых систем ИСО, прежде всего для верхних уровней, т. е. уровней приложения, представления и сеансового;

c) по управлению системами;

d) по объектно-ориентированному анализу и проектированию;

e) по теории машинных языков.

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).

ИИЭР 1073 Стандарт для обмена данными между медицинскими приборами. Обзор и основы (IEEE Std 1073, IEEE Standard for Medical Device Communications — Overview and Framework[2)])

ИСО/МЭК 8327-1 Информационные технологии. Взаимодействие открытых систем. Протокол сеансового уровня в режиме с установлением соединения. Часть 1. Спецификация протокола (то же, что и Рекомендации сектора электросвязи МСЭ X.225) (ISO/IEC 8327-1, Information technology — Open systems interconnection — Connection-oriented session protocol — Part 1: Protocol specification[3)] (same as ITU-T Recommendation X.225))

ИСО/МЭК 8650-1 Информационная технология. Взаимосвязь открытых систем. Сетевой протокол передачи с установлением соединения для протокола прикладного уровня, используемого в OSI для организации связи между двумя приложениями. Часть 1. Протокол (то же, что и Рекомендации сектора электросвязи МСЭ X.227) (ISO/IEC 8650-1, Information technology — Open systems interconnection — Connection-oriented protocol for the association control service element — Part 1: Protocol. (same as ITU-T Recommendation X.227))

ИСО/МЭК 8824-1 Информационные технологии. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации (то же, что и Рекомендации сектора электро-

1) Информацию по ссылкам можно найти в разделе 2.

2) Публикации ИИЭР доступны в Институте инженеров по электротехнике и радиоэлектронике, Хос Лэйн, Пискатавэй, NJ 08854, США (gost.e-ecolog.ru .

3) Публикации ИСО/МЭК доступны в Центральном Секретариате ИСО (Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (gost.e-ecolog.ru ке публикации ИСО/МЭК доступны в США в Global Engineering Documents (Всемирная инженерная документация),15 Inverness Way East, Englewood, CO 80112, USA (gost.e-ecolog.ru Электронные копии доступны в США в Американском национальном институте стандартов, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (gost.e-ecolog.ru

связи МСЭ X.680) (ISO/IEC 8824-1, Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation (same as ITU-T Recommendation X.680))

ИСО/МЭК 8824-2 Информационные технологии. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 2: Спецификация информационных объектов (то же, что и Рекомендации сектора электросвязи МСЭ X.681) (ISO/IEC 8824-2, Information technology — Abstract Syntax Notation One (ASN.1) — Part 2: Information object specification. (same as ITU-T Recommendation X.681))

ИСО/МЭК 8825-1 Информационные технологии. Правила кодирования языка ASN.1. Часть 1. Спецификация основных правил кодирования (BER), канонических правил кодирования (CER) и особых правил кодирования (DER) (то же, что и Рекомендации сектора электросвязи МСЭ X.690) (ISO/IEC 8825-1, Information technology — ASN.1 encoding rules — Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). (same as ITU-T Recommendation X.690))

ИСО/МЭК) 9072-2 Системы обработки информации. Передача текста. Удаленные операции. Часть 2. Спецификация протокола (ISO/IEC 9072-2, Information processing systems — Text communication — Remote operations — Part 2: Protocol specification)

ИСО/МЭК 9595 Информационная технология. Взаимосвязь открытых систем. Определение общих услуг информации административного управления (ISO/IEC 9595, Information technology — Open systems interconnection — Common management information service definition)

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

ИСО/МЭК 9899 Языки программирования — С (ISO/IEC 9899, Programming languages — C)

ИСО/МЭК 11188-3 Информационные технологии. Профиль международной стандартизации. Распространенные требования для верхнего уровня. Часть 3. Минимальные возможности верхнего уровня взаимодействия открытых систем (OSI) (ISO/IEC ISP 11188-3, Information technology — International standardization profile — Common upper layer requirements — Part 3: Minimal OSI upper layer facilities)

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[1])

ИСО/ИИЭР 11073-10201 Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами. Часть 10201: Информационная модель предметной области (DIM) (Health informatics — Point-of-care medical device communication — Part 10201: Domain information model (DIM))

ИСО/ИИЭР 11073-30200 Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами. Часть 30200. Транспортный профиль. Приборы, подключенные кабелем (ISO/IEEE 11073-30200, Health informatics — Point-of-care medical device communication — Part 30200: Transport profile — Cable connected)

ИСО/ИИЭР 11073-30300 Информатизация здоровья. Информационное взаимодействие с персональными медицинскими приборами 30300. Транспортный профиль. Инфракрасная беспроводная связь (ISO/IEEE 11073-30300, Health informatics — Point-of-care medical device communication — Part 30300: Transport profile — Infrared Wireless)

Рекомендации сектора электросвязи МСЭ X.681. Информационные технологии. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 2: Спецификация информационных объектов (то же, что и ИСО/МЭК 8824-2) (ITU-T Recommendation X.681, Information Technology — Abstract Syntax Notation One (ASN.1) — Information Object Specification (same as ISO/IEC 8824-2)[2])

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте применяются следующие термины и определения. Для получения информации о терминах, не указанных в данном разделе, см. [3].

[1] Публикации ИСО/МЭК доступны в Центральном секретариате ИСО (Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (gost.e-ecolog.ru ША в отделе продаж Американского национального института стандартов (25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/)) и в Институте инженеров по электротехнике и радиоэлектронике (445 Hoes Lane, Piscataway, NJ 08854, USA (gost.e-ecolog.ru)).

[2] Публикации ITU-T доступны в Международном союзе электросвязи (Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (gost.e-ecolog.ru

3.1.1 **абстрактный синтаксис** (abstract syntax): Спецификация структуры элемента данных, которая не ссылается или не содержит требования к конкретной технологии реализации.

3.1.2 **обратный порядок байтов** (big endian): Последовательность передачи байтов, при которой наиболее старший байт передается первым. Например, при передаче 32-битового целого числа первым передается наиболее старший байт (24-31 бит), а последним — самый младший байт (0-7 бит).

3.1.3 **порядок следования байтов** (byte order): Последовательность, в которой многобайтные простейшие элементы данных передаются в блоке данных протокола (PDU). Например, 32-битовое целое число содержит 4 байта. См. также 3.1.2.

3.1.4 **объединение** (coalescing): Функция объединения нескольких блоков данных протокола на уровне представлений (PPDU блоки) в один блок данных протокола на уровне сеанса (SPDU), который затем передается по коммуникационной сети.

3.1.5 **правила кодирования** (encoding rules): Спецификация преобразования простейших элементов данных, используемых в абстрактном синтаксисе, в формат реализации. В основном повторяет синтаксис передаваемых данных.

3.1.6 **связанный ответ** (linked reply): Ответ на команду, для передачи данных которого требуется более одного блока данных протокола (PDU). Например, отдельная команда выборки всего системного журнала может привести к получению нескольких, связанных между собой ответов от блоков данных протоколов, которые предоставляют запрашиваемую информацию[1)].

3.1.7 **значения данных уровня представления** (presentation data value — PDV): Объединение множеств значений во всех возможных абстрактных синтаксисах.

3.1.8 **синтаксис передачи** (transfer syntax): Спецификация структуры данных, во время их передачи в коммуникационной или физической среде.

3.2 Сокращения

В дополнении к сокращениям стандарта ИИЭР 1073 в настоящем стандарте используются следующие сокращения:

AA	— преждевременное прекращение (сеанса) принято (SPDU);
AARE	— сообщение ответа на ассоциацию;
AARQ	— сообщение запроса ассоциации;
AB	— преждевременное прекращение (сеанса) (SPDU);
ABRT	— преждевременное прекращение (APDU);
AC	— (сеанс) принят (SPDU);
ACSE	— сервисный элемент управления ассоциацией (связью);
AE	— прикладная сущность;
AP	— прикладной процесс;
APDU	— блок данных протокола прикладного уровня;
API	— прикладной программный интерфейс;
ARP	— аварийное разъединение по инициативе поставщика услуг (PPDU);
ARU	— аварийное разъединение по инициативе пользователя (PPDU);
ASN.1	— абстрактная синтаксическая нотация версии 1;
BCC	— прикроватный коммуникационный контроллер;
BER	— основные правила кодирования;
CC	— коммуникационный контроллер;
CMDIP	— общий протокол обмена информацией между медицинскими приборами;
CMIP	— протокол общей управляющей информации;
CMISE	— сервисный элемент общей управляющей информации;

[1)] Определение «связанной команды» (linked command) и описание ее поддержки в настоящем стандарте не рассматриваются. Данная команда требует, чтобы несколько блоков данных протокола передавали необходимую информацию. Например, команда передачи перечня наименований лекарственных средств в медицинский прибор может потребовать передачи нескольких связанных блоков данных протокола. Данная команда может быть добавлена в следующие редакции стандарта, если это потребуется другому профилю и стандартам на вспомогательный пакет услуг.

CMDISE — сервисный элемент общей информации о медицинских приборах;
CN — соединение на уровне (сеанса) (SPDU);
CP — соединение на уровне представления (PPDU),
CPA — принятие соединения на уровне представления (PPDU);
CPR — отказ соединения на уровне представления (PPDU);
DCC — коммуникационный контроллер прибора;
DICOM — формирование цифровых изображений и обмен ими в медицине;
DIF — интерфейс прибора;
DIM — информационная модель предметной области (см. ИСО/ИИЭР 11073-10201);
DN — отключение (сеанса) (SPDU);
DT — передача данных (сеанса) (SPDU);
FN — завершение (сеанса) (SPDU);
FSM — модель или машина конечного автомата;
LI — указатель длины;
LSB — наименее значимый бит;
MDAP — прикладные профили медицинских приборов (сокращение MDAP может быть заменено другим сокращением из серии стандартов ИСО/ИИЭР 11073-20000);
MMDAP-DT — нормальная передача данных прикладных профилей медицинских приборов (SPDU);
MMDAP-TD — передаваемые данные прикладных профилей медицинских приборов (PPDU);
MDAP-XT — передача сервисных данных прикладных профилей медицинских приборов (SPDU);
MDC — коммуникации медицинских приборов или номенклатура для такого рода коммуникаций (ИСО/ИИЭР 11073-10101);
MDCC — коммуникационный контроллер медицинского прибора;
MDDL — язык данных медицинских приборов (сокращение MDDL может быть заменено другим сокращением из серии стандартов ИСО/ИИЭР 11073-10000);
MDER — правила кодирования медицинских приборов;
MDIB — база данных медицинской информации;
MDNF — цифровой формат медицинских приборов;
MDS — система медицинского прибора;
MDSE — сервисный элемент медицинского прибора;
MIB — база управляющей информации;
mOSI — минимальное взаимодействие открытых систем;
MSB — наиболее значимый бит;
MTU — максимальный передаваемый блок данных;
NBO — порядок передачи байтов в сети;
OSI — взаимодействие открытых систем;
PDU — блок данных протокола (также именуется как сообщение, однако PDU означает передачу через транспортный профиль, ИСО/ИИЭР 11073-30200);
PDV — значение данных уровня представления;
PER — правила компактного кодирования;
PGI — идентификатор группы параметров;
PI — идентификатор параметра;
POC — место оказания медицинской помощи;
PPDU — блок данных протокола уровня представления;
QoS — качество обслуживания;
RF — отказ (сеанса) (SPDU);

ROER	— ошибка удаленной операции (APDU);
ROIV	— вызов удаленной операции (APDU);
ROLIV	— вызов линии передачи удаленной операции (APDU);
RORS	— результат удаленной операции (APDU);
ROSE	— сервисный элемент удаленной операции (APDU);
SI	— идентификатор SPDU;
SNTP	— простой сетевой протокол синхронизации времени;
SPDU	— блок данных протокола сеансового уровня,
SS	— служба сеансов;
TD	— данные представления (PPDU);
UML	— унифицированный язык моделирования.

4 Условные обозначения

Для обозначения специальных или оптимизированных свойств в различных определениях используются префиксные или суффиксные символы. Для указания специализованности понятия в некоторых определениях ставится звездочка (*). Например, BER* означает версию основных правил кодирования (BER), которые были специально оптимизированы для эффективности обработки данных.

5 Обоснование

Основная задача настоящего стандарта — обеспечить набор абстрактных синтаксических структур и синтаксических структур передачи данных (т. е. правил кодирования), оптимизированных для использования прикладными профилями и реализациями информационной модели предметной области (DIM).

5.1 Коммуникационная модель

Для базового стандарта и рассматриваемых в нем профилях существуют следующие положения и требования к определениям коммуникационного стека и соответствующих протоколов.

- хотя коммуникационный стек должен основываться, насколько это возможно, на уже существующих стандартах, основное внимание при определении протокола уделяется общей эффективности реализаций (например, сложность, требования к ресурсам, требования по пропускной способности). Важно чтобы даже устройства с ограниченными возможностями могли реализовать коммуникационный стек на основе настоящего стандарта;

- с целью сокращения затрат на вычислительные ресурсы при пересылке данных, заголовки, добавляемые каждым уровнем, должны быть короткими и иметь фиксированную структуру данных, а также не должны содержать необязательные элементы или элементы переменного размера.

Использование обязательных элементов или элементов постоянного формата в определениях типа данных блока данных протокола (блока PDU) позволит устройствам передачи использовать сообщения с фиксированным форматом (то есть в памяти можно заполнять шаблонное сообщение, в котором необходимо будет менять только обновляемые значения). Это также значительно снизит сложность анализа сообщения получающим устройством.

Данное требование также строго связано с требованием к оптимизированным правилам кодирования;

- коммуникационный стек должен быть достаточно гибким для того, чтобы другие профили передачи сообщений могли быть приспособлены к такому общему подходу.

Стек протокола определяется только определениями типа (данных) PDU и динамическим поведением. Нормативное определение интерфейсов прикладных программ (API) не входит в область применения настоящего стандарта, однако для упрощения реализации и повторного использования могут быть приведены ненормативные примеры.

Желательно определить транспортно-независимый интерфейс, хотя конкретные отображения блоков PDU верхнего уровня на сервисы транспортного профиля ИСО/ИИЭР 11073-30000, при необходимости, адресуются через подуровни, зависящие от транспортного протокола.

Могут быть рассмотрены основные механизмы уровня транспортного интерфейса для обеспечения информации и поведения, связанных с качеством обслуживания (QoS).

Подразумевается, что сложные протоколы сеансового уровня не требуются для коммуникации между медицинскими приборами, тем более что некоторые механизмы восстановления работоспособности системы после ошибки уже определены прикладными объектами, указанными в языке данных медицинских приборов (MDDL) (например, объектами scanner, т. е. «сканер»).

Однако, для поддержания стандартного сервисного элемента управления ассоциацией (ACSE) требуется минимальный набор стандартных служб сеансов (SS) ИСО/ВОС, как минимум на время ассоциации.

Кроме того, требуется определить конкретные расширения сеансового уровня для оптимизации его процессов во время нормальной передачи данных (после ассоциации), которые будут совместно выполняться с протоколом сеансового уровня, определенным в ИСО/МЭК 8327-1.

Данная оптимизация касается, например:

- упрощения нормальных PDU уровня сеанса;
- объединения прикладных данных в единые PDU сеансового уровня для сокращения скорости передачи сообщений на уровне транспортного интерфейса и ниже.

5.2 Информационная модель

Существуют следующие предположения и требования связанные с объектной моделью коммуникационного контроллера (CC):

- как и объекты, определенные в языке данных медицинских приборов (MDDL), объекты, определенные в данной модели, представляют информацию, которой приборы обмениваются через коммуникационный канал. Определенные здесь объекты являются частью информационной базы медицинских данных (MDIB) (доступные прямо или косвенно), по сути, являющиеся информационными контейнерами;
- объектная модель фокусируется на обобщенных концептах для представления возможностей коммуникационного интерфейса (варьирующихся от, например, скорости связи до общих параметров качества обслуживания (QoS)), вопросах конфигурации интерфейса и статистических данных (например, для выявления и устранения ошибок). Модель должна быть независимой от конкретных реализаций нижнего уровня, но может содержать адаптации, характерные для более низких уровней ИИЭР;
- коммуникационный контроллер медицинского прибора (MDCC) наследует от коммуникационного контроллера (CC) язык данных медицинских приборов (MDDL), как определено в MDDL.1 (IEEE Std 1073.3.1™ [5]); для внесения ясности и удобства использования ссылок настоящий стандарт может повторять определения из этого стандарта;
- в качестве нотации используется унифицированный язык моделирования (UML).

Атрибуты объектов и поведение будут определены в нотации, которая согласуется со стандартом языка данных медицинских приборов (MDDL), в частности в вопросах:

- форм статического представления: наследования, отношения включения, присваивания атрибутов,
- форм динамического представления в виде:
  - конечного автомата соединения приборов со всеми обменами сообщениями;
  - динамического поведения конкретных объектов, например, объектов scanner, определенных в языке данных MDDL.

Динамическое моделирование необходимо для определения фактических взаимодействий между коммуницирующими медицинскими приборами.

Кроме того, для упрощения поддержки необходимо определить информационные объекты, относящиеся к управляющей информации, например, объекты конфигурации, доступа, характеристик работы и информации, связанной с отказоустойчивостью, как указано в ИСО/ИИЭР 11073-30200.

6 Коммуникационная модель

Настоящий раздел предназначен для определения служб (сервисов) и протоколов.

6.1 Общие положения

На рисунке 1 показан верхний уровень коммуникационного стека, т. е. многоуровневый набор компонентов протокола и сервиса.

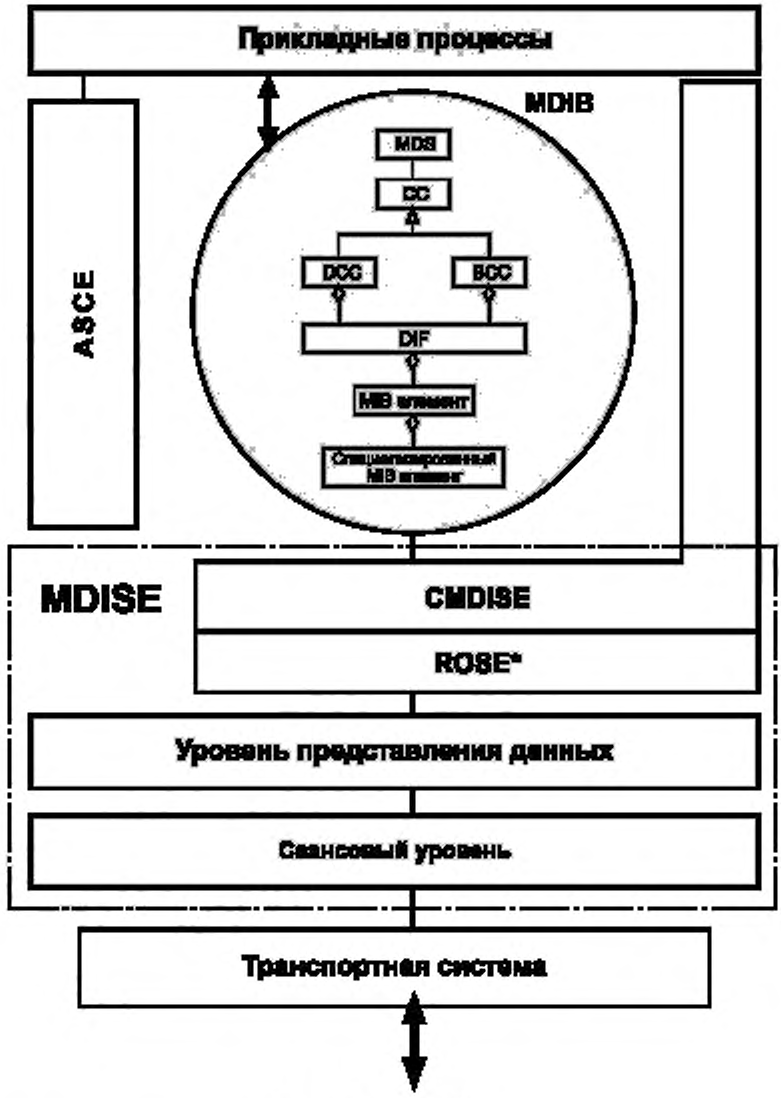


Рисунок 1 – Стек MDC

На рисунке показаны компоненты коммуникационного стека, а именно:

- ACSE (сервисный элемент управления ассоциацией) является стандартом ИСО/ВОС для управления ассоциациями.

Стандартный механизм ассоциаций обеспечивает гибкость адаптации для возможных будущих требований, например, посредством дополнительных профилей форматов на верхних уровнях (таких форматов, как форматы изображений и простой сетевой протокол синхронизации времени [SNTP]) и различных механизмов кодирования.

Он также обеспечивает безопасную и надежную проверку совместимости и некоторые ограниченные средства для согласования опций (например, чтобы убедиться, что оба устройства используют совместимые версии номенклатуры);

- сервисный элемент общей информации о медицинских приборах (CMDISE) – это служба управления объектами и по сути является упрощенной версией ИСО/ВОС сервисного элемента общей управляющей информации (CMISE);

- сервисный элемент удаленной операции (ROSE) предоставляет основные услуги, используемые сервисным элементом общей информации о медицинских приборах (CMDISE) (вызов операции, возврат результата операции, возврат ошибки, отклонение операции). В соответствии с определением оптимизированных правил кодирования, модифицированная версия сервисного элемента удаленной операции (ROSE) необходима для работы с сервисным элементом общей информации о медицинских приборах (CMDISE);

- уровни сеанса и представления несут только минимизированные издержки;

- сервисный элемент медицинского прибора (MDSE) – это общий набор всех таких элементов. Инкапсуляция обеспечивает прозрачность и гибкость реализации в применениях, которые не затрагивают внутреннюю структуру MDSE, а разработчики могут выбирать различные способы для интеграции отдельных элементов, если интерфейсы прикладных процессов и система передачи данных образуют согласованную реализацию.

Компоненты информационной базы медицинских приборов (MDIB) нормативно определяются в документах элементов информационной модели предметной области (DIM) и базы управляющей информации (MIB) и кратко описываются следующим образом:

- система медицинского прибора (MDS) – объект включения наивысшего уровня, представляющий устройство в целом;

- коммуникационный контроллер (CC) – общий объект, на основе которого определяются его специализации, как это показано ниже:

- коммуникационный контроллер прибора (DCC) – специализация, представляющая из себя агента коммуникационного контроллера прибора;

- прикроватный коммуникационный контроллер (BCC) – специализация, представляющая из себя менеджера коммуникационного контроллера прибора (DCC) (т. е. хост контроллера связи);

- интерфейс прибора (DIF) – абстрактное представление точки доступа к транспортному сервису;

- элемент базы управляющей информации (MIB) – абстрактное представление статуса или другой соответствующей информации. Специальные элементы базы управляющей информации (MIB) являются индивидуальными для данной конфигурации или реализации интерфейса прибора (DIF).

Сообщение приложения, например, сообщение с отчетом о событии объекта scanner, как определено в языке данных медицинских приборов (MDDL), проходит через этот коммуникационный стек, как показано на рисунке 2.

Объект scanner извлекает данные из базы данных медицинской информации (MDIB) и преобразует их в поле информации о событии объекта scanner.

После этого каждый уровень (или элемент стека) копирует часть данных и «помещает» их в свое собственное сообщение протокольного блока данных (PDU), обычно, посредством добавления некоторых данных заголовка, зависящих от конкретного уровня.

Принимающая система выполняет обратный процесс. Каждый уровень «извлекает» блок данных, снимая дополнительную информацию и заголовок, и далее передает результат на следующий уровень.

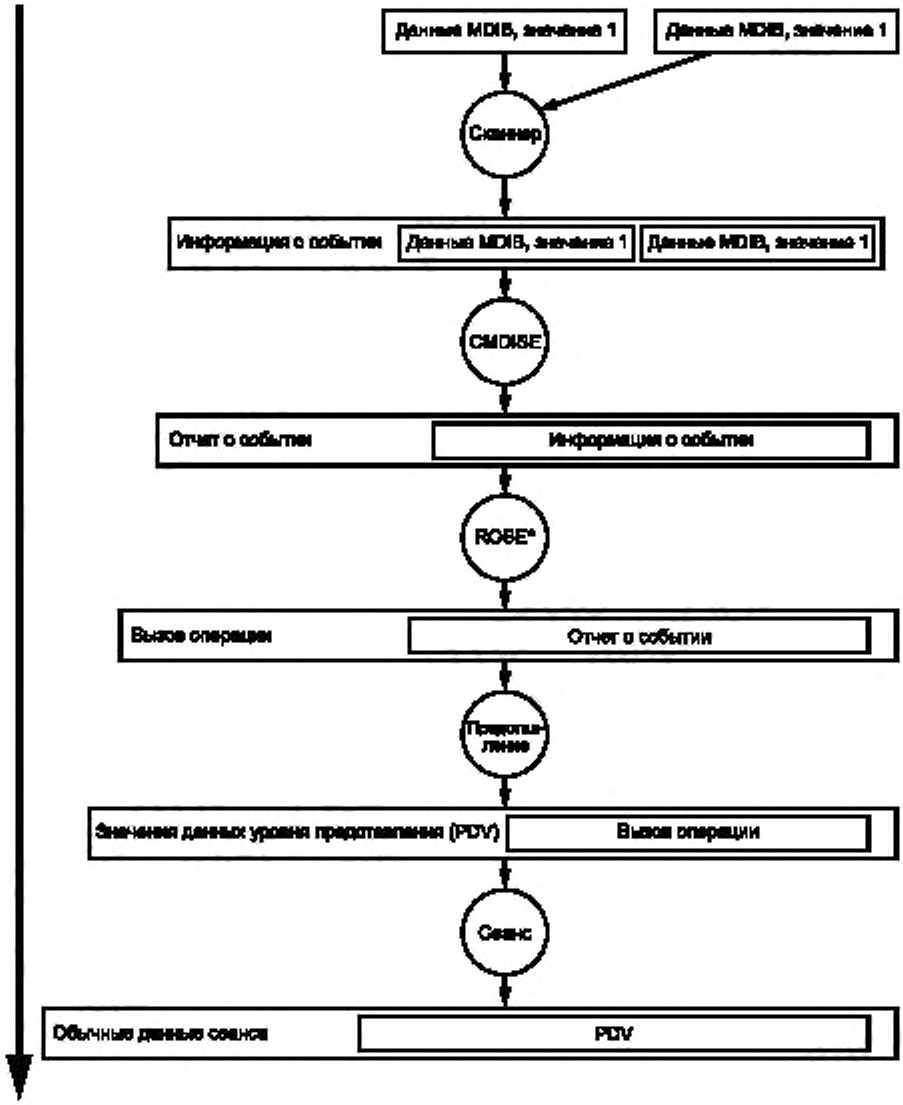


Рисунок 2 — Поток данных, проходящий через коммуникационный стек

6.2 Сервисный элемент управления ассоциацией (ACSE)

6.2.1 Общие положения

Для управления ассоциациями предполагается использовать стандартные сервисные элементы управления ассоциацией (ACSE), определенные в ИСО/МЭК 8650-1.

В дополнение к стандартным элементам ACSE необходимо определить набор полей информации пользователя, зависящих от приложения, а также минимальный (и обязательный) набор дополнительных элементов в блоках PDU ACSE.

6.2.2 Службы ACSE

В таблице 1 приведены службы (сервисы), предоставляемые сервисным элементом управления ассоциацией (ACSE).

Таблица 1 – Сводка сервисов ACSE

Служба	Тип
A-ASSOCIATE	Подтвержденный
A-RELEASE	Подтвержденный
A-ABORT	Не подтвержденный
A-P-ABORT	Инициируемый поставщиком услуг

Службы отображаются в сообщения, то есть в протокольные блоки данных (PDU) приложения. Для служб A-ASSOCIATE, например, имеются два сообщения: сообщение запроса на ассоциацию (AARQ) и сообщение ответа на ассоциацию (AARE).

Каждая служба (и таким образом полученный APDU) имеет определенное число полей данных или параметров. Таблицы 2 и 3 содержат фактические параметры сообщения запроса ассоциации (AARQ) и ответного сообщения ассоциации (AARE) для вызовов сервисов, которые определены в ACSE. Обозначения полей: M – обязательное, O – дополнительное, U – на усмотрение пользователя.

Таблица 2 – Поля AARQ блока данных APDU

Имя поля	Наличие
Версия протокола	O
Имя прикладного контекста	M
Наименование вызывающего прикладного процесса (AP)	U
Классификатор вызывающего прикладного компонента (AE)	U
Идентификатор вызова вызывающего прикладного процесса	U
Идентификатор вызова вызывающего прикладного компонента	U
Наименование вызываемого прикладного процесса	U
Классификатор вызываемого прикладного компонента	U
Идентификатор вызова вызываемого прикладного процесса	U
Идентификатор вызова вызываемого прикладного компонента	U
Информация о реализации	O
Информация о пользователе	U

Таблица 3 – Поля AARQ блока APDU

Имя поля	Наличие
Версия протокола	O
Имя прикладного контекста	M
Наименование отвечающего прикладного процесса (AP)	U
Классификатор отвечающего прикладного компонента (AE)	U
Идентификатор вызова отвечающего прикладного процесса	U
Идентификатор вызова отвечающего прикладного компонента	U
Результат	M
Источник результата – диагностика	M
Информация о реализации	O
Информация о пользователе	U

Как можно видеть, большинство полей являются дополнительными. Лишь небольшая часть из них является обязательными полями.

ACSE в профиле функциональной совместимости (интероперабельности) является лишь средством для стандартизированной установки соединения. Дополнительная информация, представляемая в поле «информация о пользователе», определяется в настоящем стандарте, чтобы облегчить взаимодействие медицинских приборов.

6.2.3 Определение сообщения ACSE ASN.1

Для получения более подробной информации см. приложение E.

Для обеспечения полной интероперабельности сообщения ACSE должны быть закодированы с помощью основных правил кодирования (BER). Кроме того, они должны быть преобразованы в соответствующий уровень представления протокольного блока данных PDU (Блок данных протокола уровня представления (PPDU)) (CP: Подключения уровня представления, CPA: Принятие подключения уровня представления) и блок данных протокола сеансового уровня (SDPU) (CN: сеанс подключения, AC: сеанс принятия), как определено в приложении E.

6.2.4 Поля информации о пользователе в ACSE

Информационные поля конкретного пользователя (то есть зависящие от определенного приложения) в определении сообщения ACSE для использования в коммуникационном стеке интероперабельности определены в приложении E.

Для запуска блоки ACSE должны быть обеспечены только минимальной информацией. После стадии ассоциации все другие данные, необходимые для приложения и проверки совместимости, могут быть предоставлены во встроенных службах CMDISE с использованием определений языка MDDL.

6.3 Протокол сеансового уровня

6.3.1 Общие положения

6.3.2 Службы сеансового уровня

Протокол сеансового уровня определяет набор служб, используемых для связи и передачи данных. Важные службы, которые необходимо учитывать:

- подключение сеанса (Session connect);
- принятие сеанса (Session accept);
- передача данных сеанса (Session data transfer).

Может быть использовано несколько дополнительных услуг и протокольных блоков данных. Этот вопрос требует дальнейшего рассмотрения, особенно в отношении отображений элементов между ACSE, PPDU и SPDU.

6.3.3 Определения сообщений сеансового уровня

Для всех необходимых стандартных служб сеанса (SSs), а также для оптимизированного расширения сеансового уровня будет определена структура PDU (заголовок сеанса).

SPDU блоки построены из простых элементов в форме, указанной на рисунке 3.

Блоки SPDU	**SI**	**LI**	**Поле параметров**	**Поле информации о пользователе**
Модули PGI	**PGI**	**LI**	**Поле параметров**	
Модули PI	**PI**	**LI**	**Поле параметров**	

Условные обозначения:

LI — индикатор длины (длина 0-254: один октет; иначе 3 октета, начиная с 255); PGI — идентификатор группы параметров (определяет группу параметров сеансового уровня); PI — идентификатор параметров (определяет отдельный параметр сеансового уровня); SI — идентификатор SPDU (уникальный идентификатор, определяющий тип сообщения сеансового уровня)

Рисунок 3 — Формат протокольного блока данных сеансового уровня (SPDU)

Поля параметров составлены из блоков PGI и PI по определенным правилам (в соответствии с ИСО/МЭК 8327-1).

Пример сообщения сеансового уровня представлен в п. 6.3.3.1.

6.3.3.1 Подключение сеанса (CN) SPDU

Формат и содержание блока SPDU представлен следующим образом:

```
OI  XX              SI - 13 (CN), LI - длина в октетах
     05  06             PGI - 05 (элемент соединить/принять), LI - 06
13   01  00         PI - 19 (опции), LI - 1, значение - 0
     16  01  03         PI - 22 (версия), LI - 1, значение - 3
     14  02  00  02     PI - 20 (требования пользователя), LI - 2, значение:
                        выберите полнодуплексный функциональный блок
CI   YY             PGI - 193 (данные пользователя), LI - длина данных пользо-
                    вателя
```

6.4 Протокол уровня представления

6.4.1 Общие положения

Протокол уровня представления позволяет согласовать абстрактный синтаксис (например, выбрать между MDDL и CMDISE ASN.1) и синтаксис передаваемых данных (т. е. оптимизированные правила кодирования, например, MDER) между системами.

Что касается протокола сеансового уровня, для поддержки ACSE необходимы некоторые ограниченные стандартные службы.

Основной дополнительной возможностью уровня представления является согласование синтаксиса во время ассоциации. Также это позволяет определить точки многоканальной передачи (идентификаторы контекста представления) в приложении, которые в свою очередь позволяют осуществлять передачу данных в различных форматах в рамках одной ассоциации [например, для формирования изображений и обмена ими в медицине (стандарт DICOM)]. В подобной ассоциации можно поддерживать несколько контекстов представления одновременно.

При нормальной связи медицинских приборов (после ассоциации), требований к обработке данных и обмену сообщениями для уровня представления должно быть как можно меньше.

6.4.2 Службы уровня представления

Уровень представления предоставляет, например, следующие службы:

- Подключение уровня представления (Connect presentation);
- Принятие подключения уровня представления (Connect presentation accept);
- Предоставление данных (Presentation data).

6.4.3 Сообщения уровня представления

Определения следующих блоков PDU см. в приложении Е:

- Подключение уровня представления (CP) PPDU;
- Принятие подключения уровня представления (CPA) PPDU;
- Отклонение подключения уровня представления (CPR) PPDU;
- Протокол аварийного разъединения по инициативе провайдера (ARP) PPDU;
- Протокол аварийного разъединения по инициативе пользователя (ARU) PPDU;
- Предоставление данных (TD) PPDU.

6.5 Протокол ROSE

6.5.1 Общие положения

Служебный элемент дистанционной операции (ROSE) использует общий протокол обмена информацией между медицинскими приборами (CMDIP). Он обеспечивает связь между сообщениями вызова и сообщениями результата (т. е. между запросами и ответами) с помощью полей идентификаторов вызова. Он также содержит поле, необходимое для отличия различных удаленных операций (в данном случае — сервисов CMDISE).

ROSE использует тот же абстрактный синтаксис, который рассматривается для применения в CMDIP и структур данных из ИСО/ИИЭР 11073-10201. Поэтому для соблюдения ограничений оптимизированных правил кодирования ASN.1 необходимо внести некоторые модификации в ROSE ИСО/ВОС.

6.5.2 Службы ROSE

Протокол ROSE является относительно простым протоколом, определяющим следующие службы (сервисы):

- Вызов удаленной операции (Remote operation invoke);
- Результат удаленной операции (Remote operation result);
- Ошибка удаленной операции (Remote operation error);
- Отклонение удаленной операции (Remote operation reject).

6.5.3 Определения сообщений протокола ROSE

Определения ROSE PDU см. в приложении Е.

6.6 Протокол CMDISE (CMDIP)

6.6.1 Общие положения

6.6.2 Сервисы CMDISE

В зависимости от масштабируемости (профили минимальной, базовой и расширенной масштабируемости имеют разные требования) прикладными профилями медицинских приборов (MDAP) могут предоставляться следующие основные службы:

- Возвращение значения атрибута объекта (Retrieve object attribute value);
- Модификация значения атрибута объекта (Modify object attribute value);
- Вызов заданных функций объекта (Invoke object defined functions);
- Создание и удаление экземпляров объекта (Create and delete object instances);
- Создание отчетов о событиях, произошедших внутри объекта (Report events that occurred within an object).

Параметры каждой из служб и соответствующие разультаты определены в языке данных медицинских приборов (MDDL). В таблице 4 приведен пример службы event report (отчет о событии).

Таблица 4 — Параметры службы event report

Параметр	Описание
Invoke identifier	Идентификатор вызова. Уникальный идентификатор (например, номер последовательности) назначается для конкретного экземпляра службы так, чтобы его можно было отличить от вызовов других служб, которые может разрабатывать поставщик услуг
Mode	Режим. Подтвержденный или неподтвержденный; неподтвержденный режим требует ответа
Object class	Класс объекта. Определяет класс объекта, генерирующего событие (значения определены в номенклатуре или глоссарии)
Object instance	Экземпляр объекта. Определяет экземпляр объекта, который генерирует событие
Event time	Время генерации события
Event type	Определяет тип события (значения определены в номенклатуре или глоссарии)
Event information	Дополнительная информация о событии согласно типу параметра события; информация о событии определяется объектом, генерирующем событие (опционально)

6.6.3 Определения сообщений CMDIP

Определения типов данных ASN.1 для всех служб CMDIP определены в приложении Е.

Необходимо отметить, что некоторые параметры служб CMDISE, определенные в языке MDDL, могут быть отображены на ROSE. В частности, идентификатор вызова и параметры режима в действительности определены в ROSE, как описано в 6.5.

Примеры блоков PDU приведены в приложении Е.

6.6.4 Простой сетевой протокол времени SNTP

См. приложение С.

7 Информационная модель

7.1 Модель объекта

Настоящий подраздел содержит определения классов объектов, которые в основном содержат следующие классы:

- коммуникационный контроллер (CC), например коммуникационный контроллер прибора (DCC), прикроватный коммуникационный контроллер (BCC);

- база управляющей информации (MIB).

- Для получения подробной информации об определении см. информационную модель предметной области (DIM).

7.2 Модель формата

7.2.1 Синтаксис

7.2.1.1 Синтаксис передаваемых данных

Синтаксис передаваемых данных определен в приложении A.

Отображения на ИСО АСН.1 приведены в приложении G.

7.2.1.2 Абстрактный синтаксис

Абстрактный синтаксис определен в приложении E.

7.2.2 Совместимость

Как правило, желательно поддерживать обратную синтаксическую совместимость (совместимость с предыдущими версиями), особенно для синтаксиса передачи данных, хотя такая совместимость не всегда возможна. Настоящий подраздел предназначен для выявления существенных вопросов совместимости и их разрешения в целях содействия реализации. См. таблицу 5.

Таблица 5 – Проблемы совместимости

Случай	Вопрос	Решение
1	Совместимость версий 1988/90 и 1994 ИСО АСН.1 и смежных правил кодирования (серия ИСО/МЭК 8824, серия ИСО/МЭК 8825)	
1.1	Тип ANY DEFINED BY изменен; см. приложение H для получения дополнительной информации	1 АБСТРАКТНЫЙ СИНТАКСИС. Модули должны соответствовать требованиям ИСО/МЭК 8824-1; см. приложение H 2 СИНТАКСИС ПЕРЕДАЧИ ДАННЫХ. MDER для ANY DEFINED BY или типа instance-of должны быть идентичными (см. приложение A)

8 Соответствие

Настоящий раздел предназначен для определения критериев соответствия.

8.1 Область применения

Чтобы в наибольшей степени обеспечить гибкость и интероперабельность реализации, область соответствия реализации в настоящем базовом стандарте ограничивается протоколами. Тем не менее профили приложений, в случае необходимости, могут устанавливать определения служб.

8.2 Администрирование идентификатора объекта

Настоящий стандарт выделяет административные полномочия идентификатора объекта для других стандартов, особенно для языка MDDL. Управление полномочиями требует обращения сначала к набору MDAP, а затем к спецификациям расширений, определенных более широким стандартом.

8.3 Соответствие подмножества MDAP

Профили MDAP должны определить степень соответствия с настоящим стандартом с помощью следующих категорий исключения:

а) *отсутствие исключений*. Подмножество не принимает никаких исключений к настоящему стандарту;

б) *некоторые исключения существуют*. Подмножество принимает некоторые исключения к настоящему стандарту. Но только в этом случае конкретные исключения требуют определения. Если в настоящем стандарте задается таблица для определения соответствия, то такой стандарт на подмножество должен повторить эту таблицу и включить конкретные случаи соответствия для всех случаев, вне зависимости от выполнения или невыполнения соответствия.

8.4 Соответствие реализации

Разработчики устройства могут не указывать соответствие настоящему стандарту, но необходимо указать соответствие профилю, который ссылается на настоящий стандарт. Как было отмечено в 8.1, соответствие реализации должно быть ограничено протоколами, и, хотя определения интерфейсов прикладных программ (API) несут справочный характер, они подходят для задачи облегчения повторного использования компонентов реализации.

Приложение А
(обязательное)

Правила кодирования медицинских приборов (MDER)

А.1 Общие положения

Настоящее приложение определяет специализированные правила MDER, связанные с представлением последовательных двоичных строк, таким образом, каким они должны быть представлены в сети в сравнении с соответствующей организацией в памяти компьютера, с представлением в абстрактном синтаксисе, то есть в языке программирования или диаграмм, которые используются в спецификациях. Предполагается, что данная спецификация должна быть согласована с любой и каждой из альтернатив нижнего уровня ИСО/ИИЭР 11073. Таким образом, реализация на верхних уровнях может обеспечивать прозрачность на основе конкретного профиля нижнего уровня.

Основные цели MDER включают в себя возможность оптимизировать выполнение форматирования и синтаксического анализа, а также снижает нагрузку на пропускную способность сети. Оптимизация форматирования основана на возможности процессора передачи данных определять так называемые сообщения с фиксированным форматом (*canned*), в которые только динамически изменяемые данные должны быть включены для относительно высокочастотных сообщений, в особенности это касается данных осциллограмм.

А.2 Поддерживаемый синтаксис ASN.1

ASN.1 является стандартным обозначением, которое используется для определения типов данных, значений и ограничений значений. Данное обозначение широко используется в стандартах ВОС и в серии стандартов ИСО/ИИЭР 11073 (например, в ИСО/ИИЭР 11073-10201, где все определения данных формируются с помощью ASN.1).

В целях выполнения требований эффективности кодирования и декодирования и поддержки сообщений с фиксированным форматом, MDER определяет методы для преобразования синтаксиса ASN.1 в поток байтов, пригодный для передачи.

В отличие от других стандартов ИСО/ВОС для правил кодирования ASN.1 (например, основные правила кодирования или BER, правила компактного кодирования или PER) правила MDER оптимизированы только для подмножества ASN.1. Правила MDER не поддерживают полный набор типов данных ASN.1, а лишь определенный ограниченный набор конструкций ASN.1.

Стандарты ИСО/ИИЭР 11073 используют этот ограниченный набор ASN.1 для определения типов данных, применяемых только в управляемых медицинских объектах, таким образом, правила MDER подходят и являются достаточными для кодирования структур данных в рамках этих стандартов.

Ограниченный набор ASN.1, используемый для компонентов PDU стандартов ИСО/ИИЭР 11073, является строгим подмножеством допустимых типов данных ASN.1, поэтому другие общие стандартные правила кодирования (например, BER, PER) можно так же использовать, как согласованные для конкретного профиля коммуникаций на более высоких уровнях.

Таблица А.1 определяет специализацию ASN.1, подходящую для кодирования MDER. Все компоненты ASN.1 PDU, предназначенные для кодирования MDER, являются предметами этой специализации.

Для каждого типа данных ASN.1 эта специализация сопровождается символом «I» для включенного с ограничением, «R» — для ограничений по использованию и «E» — для исключения.

Более подробную информацию о специализации типов ASN.1 в MDER см. в приложении Н.

Таблица А.1 — Типы данных, поддерживаемые ASN.1

Тип ASN.1	Статус	Комментарии
INTEGER	R	Целочисленный тип. Размерные ограничения должны быть использованы для всех типов данных INTEGER, для определения диапазона значений целого числа. Краткие имена для поддерживаемых типов ограничений определяются следующим образом: INT-U8 ::= INTEGER(0..255) INT-I8 ::= INTEGER (-127..128) INT-U16 ::= INTEGER (0..65535) INT-I16 ::= INTEGER (-32768..32767) INT-U32 ::=INTEGER (0..4294967295) INT-I32 ::= INTEGER (-2147483648..2147483647) Только сокращенные, ограниченные по размеру типы данных INTEGER следует использовать с определениями типов данных для кодирования в MDER

Окончание таблицы А.1

Тип ASN.1	Статус	Комментарии
BIT STRING	R	Битовая строка. Размерные ограничения должны быть использованы для всех типов данных BIT STRING, для определения диапазона значений битовой строки. Краткие имена для поддерживаемых типов ограничений определяются следующим образом: BITS-8 :: = BIT STRING (SIZE(8)) BITS-16 :: = BIT STRING (SIZE(16)) BITS-32 :: = BIT STRING (SIZE(32)) Только сокращенные, ограниченные по размеру типы данных BIT STRING следует использовать с определениями типов данных для кодирования в MDER
OCTET STRING	I	Строка октет
SEQUENCE	R	Может не использовать следующие типы тегирования: OPTIONAL (опциональный), DEFAULT (по умолчанию), автоматический
SEQUENCE OF	I	Последовательность
CHOICE	R	Выбор. Может использоваться явное и неявное тегирование
ANY DEFINED BY	I	ANY DEFINED BY должен определять компонент в структуре данных (в основном в SEQUENCE), который определяет структуру этих данных для преобразователя кода/синтаксического анализатора (парсера)

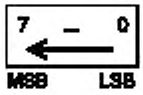
А.3 Порядок передачи байтов

На рисунке А.1 показано, как различные двоичные строки сети отображаются в строках памяти. На диаграммах представлен порядок передачи байтов в сети (Network byte order, NBO). Следующие правила пронумерованы для удобства использования ссылок:

- представление в диаграммах использует формат NBO, показанный на рисунке А.1;
- в MDER не используется выравнивание. То есть в строки байтов дополнительные байты не добавляются, например для получения длин, которые делятся на два или четыре. Тем не менее переменная длина элементов данных, то есть строк, должна содержать четное число байтов из соображений эффективности. Например, поскольку большинство элементов данных 16-битные, они не будут неправильно выровненными, если строки имеют четную длину;
- передачи данных в MDAP ограничены использованием соглашения NBO (обратный порядок передачи);
- для обеспечения общей интероперабельности протокол ассоциации должен использовать ИСО BER при согласовании условных обозначений MDER. Все остальные блоки PDU, которыми обменивается в период своей работы хост-устройство, будут основаны на MDER, например PDU CMIP* и ROSE*. Суффикс звездочка (*) означает, что MDER используется для оптимизации протокола ИСО, который, как правило, базируется на BER.

Многобайтовые структуры отображаются между сетью и компьютерной памятью и упорядочиваются в памяти компьютера двумя основными способами, называемыми *big endian* (формат с порядком следования байтов, начиная со старшего) и *little endian* (формат с порядком следования байтов, начиная с младшего). Формат big endian согласуется с NBO, а little endian — не согласуется. Например, в последнем примере на рисунке А.1 структура ABCD была бы упорядочена как DCBA. В этом случае если big endian является согласованным протоколом, то компьютер с little endian должен был бы переставлять компоненты этой структуры при получении их из памяти и передаче их в память, в случае необходимости. Макросы языка программирования и команды компьютера, выполняющие байтовый свопинг, которые, как правило, способствуют нормализации, являются проблемами реализации и могут быть упрощены ненормативными определениями, взятыми из настоящего стандарта или стандартов, связанных с ним.

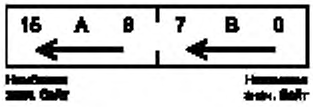
- NBO:
  - Однобайтовая строка битов, т.е. октета:
    - Последовательность битов: в порядке от наименее значащего бита (LSB) к наиболее значимому биту (MSB), т.е. 0, ...,7 или 24, ..., 31; порядок следования битов представлен в диаграммах с помощью нотации ←, в которой конец стрелки обозначает последний переданный бит:



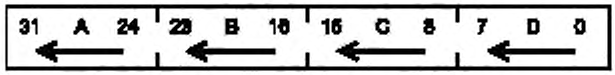
- Многобайтовая строка:
  - Неструктурированная: массив октет (т. е. строка октет):
    - Последовательность битов: для каждого типа так, как определено для октеты;
    - Последовательность байтов: в основном нумеруется от [0] до [n-1], например A[0] – A[n-1], где <n>= длина в октетах



- Структурированная: многобайтовая последовательность битов, в основном кратных двум байтам (например, короткое целое число — 16 битов, длинное целое число — 32 бита); числа с плавающей точкой в основном кратны 16 битам, хотя в настоящем стандарте определен только 32-битовый формат FLOAT. Приведены два общих примера (ABCD относится к порядку передачи байтов):
  - 16-битовая структура, например короткое (целое число):
    - Последовательность битов: каждый байт пересылается согласно определению пересылки для октета;
    - Последовательность байтов: пересылается от наиболее значащего байта к наименее значащему байту;
    - Для целых чисел со знаком обычно MSB наиболее значащего байта – бит знака(ов)



- 32-битовая структура, например длинное (целое число)



- Согласно условному обозначению мультиструктурные композиции показаны в порядке появления в последовательно передаваемой строке

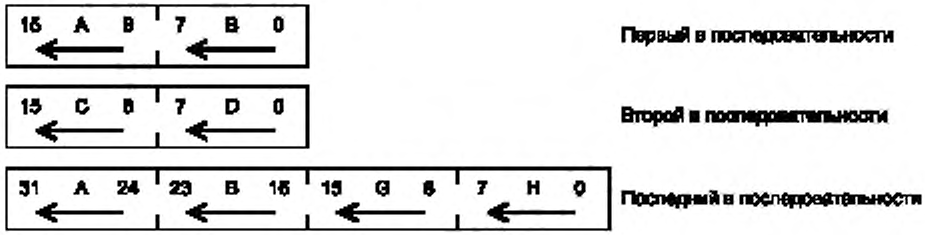


Рисунок А.1 — NBO. Условные обозначения представления двоичной строки

А.4 Кодирование

А.4.1 Общие положения

В MDER отсутствует тегирование для простых типов. Теги используются только там, где дешифратору (декодеру) необходимо различать типы (например, для типа CHOICE). Поля длины используются только для элементов с переменной длиной и ограничены 64 КБайтами (16 битами), которых должно быть достаточно для передачи данных.

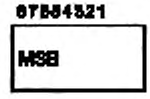
Простые типы определены из-за ограничений по размеру и имеют фиксированную длину. Типы SEQUENCE, имеющие фиксированную длину, поддерживаются при условии, если отсутствуют компоненты синтаксиса типа OPTIONAL. Если это неприемлемо, то должны быть определены стандартные правила кодирования для использования в стандартном профиле.

A.4.2 Тип INTEGER

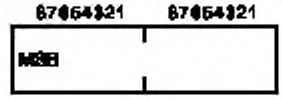
Кодирование целочисленного значения является примитивным кодированием, а содержимое октет представляет значение в дополнительном двоичном коде.

На рисунке А.2 представлено поддерживаемое в MDER кодирование октет[1)] для ограниченных по размеру целочисленных значений.

• 8-битовые типы INT-U8, INT-I8



• 16-битовые типы INT-U16, INT-I16



• 32-битовые типы INT-U32, INT-I32

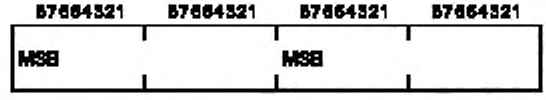


Рисунок А.2 – Кодирование целых чисел

Октеты содержат представление закодированного целочисленного значения в дополнительном двоичном коде.

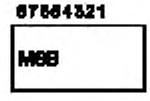
A.4.3 Тип BIT STRING

Кодирование значения битовой строки, относящейся к базовому типу, является простым. Содержимое октеты представляет множество битов в битовой строке. Битовая строка может содержать 8, 16, или 32 бита.

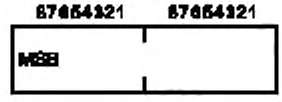
Бит 0 в кодировке представлен наиболее значимым битом (MSB), бит 1 представлен следующим битом в октете и т.д.

На рисунке А.3 представлено поддерживаемое в MDER кодирование октет для ограниченных по размеру битовых строк.

• 8-битовые типы BITS 8



• 16-битовые типы BITS 16



• 32-битовые типы BITS 32

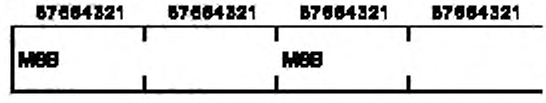


Рисунок А.3 – Кодирование битовой строки

[1)] Для стандартизации языка программирования С для целочисленных типов данных необходимо использовать определения ИСО/МЭК 9899.

Пример — Определение

```
state ::= BITS-16 {open(0), locked(1) }
```

может быть отображено на представление типа языка С следующим образом:

```
short unsigned int state;
#define locked 0x4000
#define open 0x8000
```

(по аналогии с именованными битами в битовых строках).

A.4.4 Тип OCTET STRING

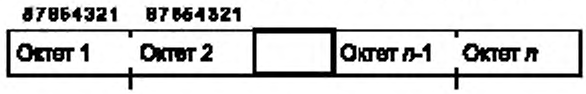
Кодирование значения OCTET STRING, относящегося к базовому типу, является простым. Содержимое октетов представляет собой строку элементов. Сами октеты используют кодирование, унаследованное от определения типа строки.

Будучи зависимы от этого типа октеты могут содержать печатаемые символы ASCII (в случае 16-битовых наборов символов символ использует 2 октета в кодировке) или строка может содержать больший объем инкапсулированных двоичных данных.

На рисунке А.4 представлено поддерживаемое в MDER кодирование октет для ограниченных по размеру значений битовых строк.

Как показано на рисунке А.4, MDER различают тип OCTET STRING с переменной длиной строки и тип OCTET STRING ограниченного размера.

• Фиксированный (ограниченный по размеру) тип: OCTET STRING (SIZE(n))



• Типы OCTET STRING переменной длины

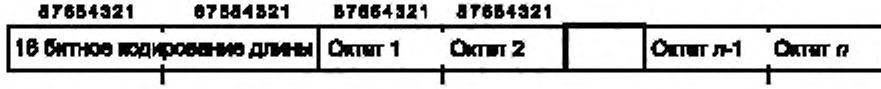


Рисунок А.4 – Кодирование типов OCTET STRING

Тип OCTET STRING с фиксированной (т. е. ограниченной по размеру) длиной строки кодируется только соответствующим набором октет содержания.

Типы OCTET STRING переменной длины кодируются с полем длиной 16 бит (целое число без знака в дополнительном двоичном коде), за которым следует определенное число октет содержания.

Пример – Следующие определения

```
fixed-sized-label ::= OCTET STRING (SIZE(12))
variable-label::= OCTET STRING
```

могут быть отображены на представление типа языка С следующим образом:

```
typedef unsigned char fixed_size_label[12];

typedef struct {
   unsigned short length;
   unsigned chardata [ ];/* здесь нужно вставить массив подходящего размера
   */
   } variable_label;
```

A.4.5 Тип SEQUENCE

Кодирование значения последовательности конструируется, а октеты содержания представляют закодированные значения элементов типа SEQUENCE, без каких-либо дополнительных закодированных данных. Пробелы (например, для выравнивания) не добавляются.

Значения компонентов должны появляться в порядке их определения в типе SEQUENCE.

Пример – Следующие определения

```
IdentType ::= SEQUENCE {
   id INT-U16,
```

```
    instanceINT-U16
    }
```

могут быть отображены на представление типа языка С следующим образом:

```
typedef struct {
    unsigned short id,
    unsigned short instance
    } IdentType;
```

и кодирование по MDER будет иметь вид, представленный на рисунке А.5.

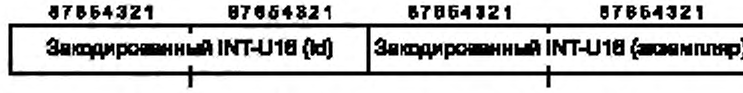


Рисунок А.5 – Образец кодирования типа SEQUENCE

А.4.6 Тип SEQUENCE OF

Кодирование значения SEQUENCE OF конструируется, а октеты содержания представляют закодированные значения элементов типа SEQUENCE OF таким образом, чтобы ему предшествовало поле счетчика, указывающее на число элементов, и поле длины, указывающее полную длину структуры данных (в которой не учитываются сами счетчик и длина).

Кодирование должно сохранить порядок значений компонентов. См. рисунок А.6.

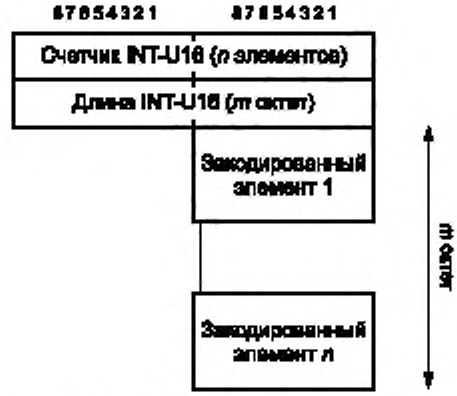


Рисунок А.6 – Кодирование типа SEQUENCE OF

Поле счетчика и поле длины с содержанием «0» указывают на структуру данных пустого списка. Такая комбинация значений допускается.

Пример – Следующие описания:

```
Array1 ::= SEQUENCE OF Entry
```

могут быть отображены на представление типа языка С следующим образом:

```
typedef struct {
    unsigned short count;
    unsigned short length;
    Entry data[1]; /* здесь нужно вставить достаточное число записей */
} Array1;
```

А.4.7 Тип CHOICE

Кодирование значения выбора конструируется, а октеты содержания представляют закодированные значения выбранной альтернативы таким образом, чтобы ему предшествовало поле тега, указывающее на выбранную альтернативу, и поле длины, указывающее длину кодирования выбранной альтернативы. См. рисунок А.7.

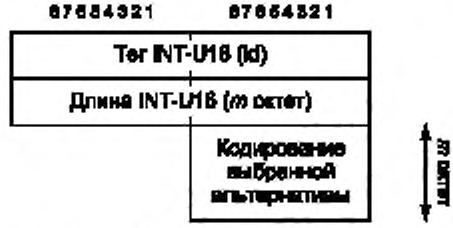


Рисунок А.7 – Кодирование типа CHOICE

Пример – Следующие описания

```
ChoiceType ::= CHOICE {
   one OneType,
   two TwoType
}
```

могут быть отображены на представление типа языка С следующим образом:

```
typedef struct {
   unsigned short choice_id;
   unsigned short length;
   union {
     OneType one;
     TwoType two;
     } data;
} ChoiceType;
#define one_type_chosen 1
#define two_type_chosen 2
```

Правила для значений тегов определяются следующим образом:

- теги могут быть заданы явно или неявно;

- абстрактный синтаксис для неявно заданных тегов не содержит явно заданного номера выбора и, следовательно, требуется правило для присвоения значений полю choice_id. Для неявно заданных тегов значения поля choice_id начинаются со значения 1 и последовательно размещаются в порядке абстрактных синтаксических выборов. В приведенном выше примере значения поля choice_id для полей one_type_chosen и two_type_chosen будут равны 1 и 2 соответственно;

- абстрактный синтаксис для явно заданных тегов содержит явно заданный номер выбора, который отражается непосредственно в поле choice_id только что определенного правила кодирования. В этом случае процедуры выбора должны выполняться последовательно и в зависимости от применения могут быть несвязанными, как показано в следующем примере:

```
choice-type ::= CHOICE {
   one  [1] OneType,-- defines tag value 1 in MDER
   four [4] FourType-- defines tag value 4 in MDER
   }
```

А.4.8 Тип ANY DEFINED BY и Instance-of

ANY DEFINED BY (ASN.1 1988/90) или тип instance-of (ASN.1 1994) кодируется заголовком поля длины, чтобы задать число октет в кодировании выбранного значения, как представлено ниже. См. рисунок А.8.

Данные типы, как правило, представляют встроенные синтаксисы, посредством зарегистрированного идентификатора объекта. См. приложение Н для случаев совместимости.

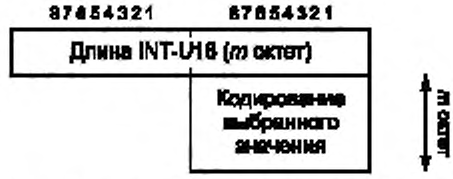


Рисунок А.8 – Кодирование типа ANY DEFINED BY (Instance-of)

Пример — Следующие описания

```
TestType ::= SEQUENCE {
    type-idOIDType,
    value ANY DEFINED BY type-id
}
```

могут быть отображены на представление типа языка С следующим образом:

```
typedef struct {
    OIDTypetype-id,
    unsigned shortany_length;
    char any_data;/* здесь нужно вставить закодированный тип данных*/
} TestType;
```

Данный пример показывает кодирование байтов последовательности SEQUENCE, содержащей идентификатор объекта, зависящий от контекста, и значение ANY DEFINED BY.

В предыдущем преобразовании поле type-id является идентификатором объекта, не зависящим от контекста. Приложение должно использовать поле идентификатора, чтобы привести поле any_data к правильному типу данных. Символьный тип данных для поля any_data, по существу, не несет значения и предоставляет только адрес поля. Следует отметить, что длина может быть 0, что означает, что поле any_data не существует.

Тип instance-of кодирует конструкцию TYPE-IDENTIFIER из ASN.1 и идентичен кодированию ANY DEFINED BY для обеспечения обратной совместимости (совместимости с предыдущими версиями).

А.5 Структура данных с плавающей точкой

Ограниченное подмножество ASN.1, которое может быть отображено с MDER, не содержит данных типа FLOAT.

Вместо этого в ИСО/ИИЭР 11073-10201 для чисел с плавающей точкой определяется универсальный тип данных FLOAT.

Тип FLOAT отображается как 32-битная структура, форматированная в соответствии с цифровым форматом медицинских приборов (MDNF).

MDNF — это 32-битное слово, содержащее 8-битную целочисленную экспоненту (показатель степени) со знаком, за которой следует 24-битная целочисленная величина со знаком. См. рисунок А.9.



Рисунок А.9 – Кодирование MDNF

Представленное число будет (величина)×($10^{\text{экспонента}}$). И показатель степени, и величина представлены в двоичном дополнительном коде. Нормализация величины не обязательна.

Существует четыре специальных значения, которые могут быть представлены, как показано в таблице А.2.

Таблица А.2 — Специальные значения MDNF

Специальное значение	Перевод	Порядок величины
NaN (not a number)	Не число	$+(2^{23}-1)$
NRes (not at this resolution)	Не при таком разрешении	$-(2^{23})$
+ INFINITY	+ Бесконечность	$+(2^{23}-2)$
– INFINITY	– Бесконечность	$-(2^{23}-2)$

В данных случаях показатель степени не важен. Это предоставляет для представления нормальных чисел следующие диапазоны:

- $-128 \le$ показатель степени ≤ 127;
- $-2(2^{23}-3) \le$ величина $\le +(2^{23}-3)$;
- NaN = $+(2^{23}-1)$;
- NRes = $-2(2^{23})$;
- ± INIFNITY = ± $(2^{23}-2)$.

Ниже приведены определения числа значащих цифр для представления на дисплее:

- если показатель степени <0, то целочисленное значение показателя степени отображает число значащих цифр после запятой. См. примеры в таблице А.3;

Таблица А.3 — Примеры при показателе <0

Показатель степени	Величина	Значение
–3	32 000	32.000
–1	320	32.0

- если показатель ≥ 0, то число значащих цифр после запятой равно нулю. См. примеры в таблице А.4.

Таблица А.4 — Примеры при показателе ≥0

Показатель степени	Величина	Значение
1	320	3200
2	32	3200

Приложение В
(обязательное)

Распределение идентификаторов

В.1 Введение

Настоящее приложение предусмотрено для составителей и разработчиков группы стандартов ИСО/ИИЭР 11073-20000 в качестве руководства для формирования идентификаторов объектов для стандартов ИСО/ИИЭР 11073 (определение и использование идентификаторов объектов см. в ИСО/МЭК 8824-2).

В.2 Основа распределения идентификаторов

Для простоты идентификаторы объектов, присвоенные в настоящем приложении, используют структурированную табличную форму, в которой каждый отступ соответствует ответвлению. См. рисунок В.1.

Корневой объект для идентификаторов объектов в настоящем документе:

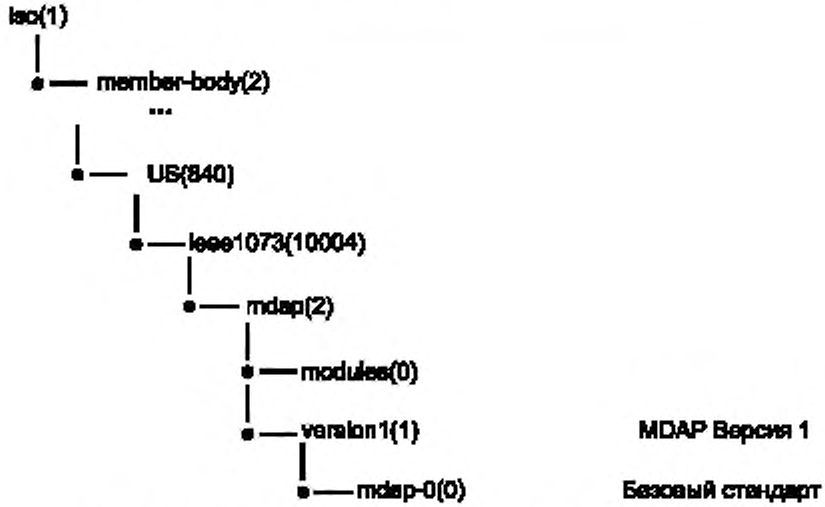
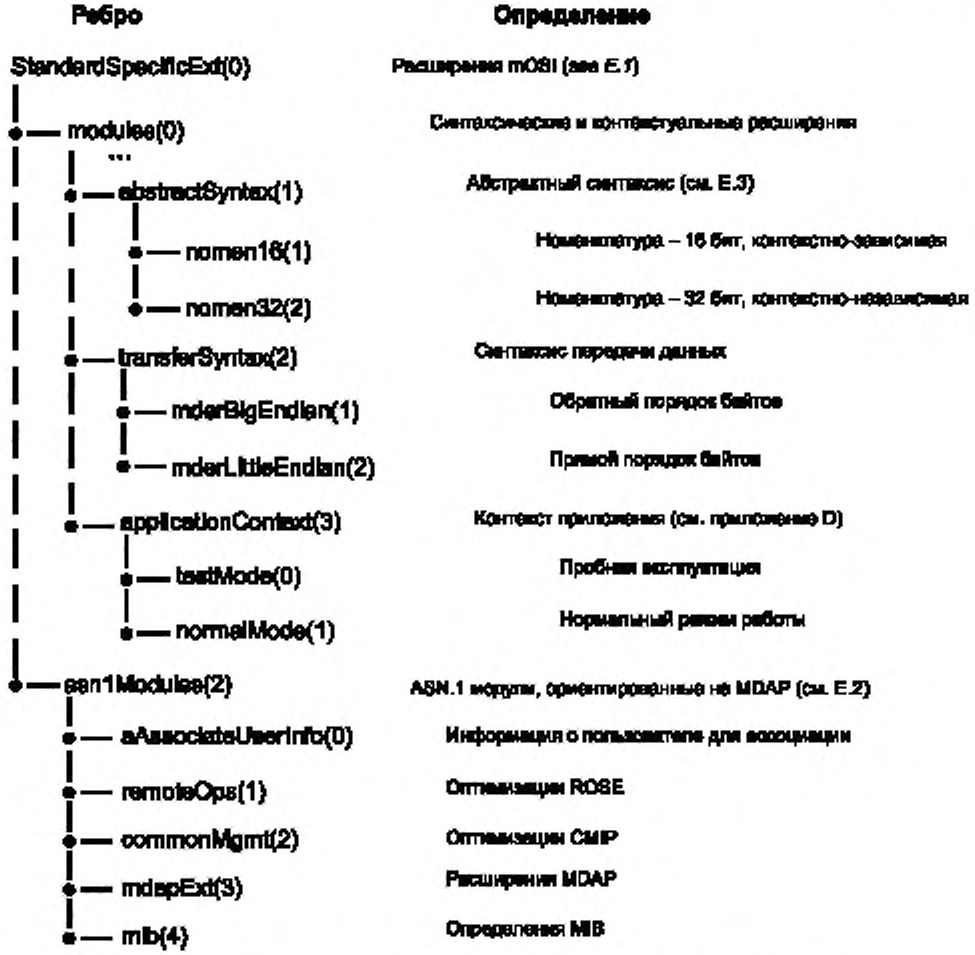


Рисунок В.1 – Присвоение идентификаторов объектов. Корневой путь

Дуги ниже корневого объекта mdap-0 показаны на рисунке В.2[1)].

[1)] Примечания в тексте, таблицах и рисунках носят исключительно справочный характер и не содержат требований, необходимых для применения настоящего стандарта.



Примечание – Дополнительные ответвления, определенные в стандарте ИИЭР 1073, опущены для краткости.

Рисунок В.2 – Распределение идентификаторов объекта. Конкретные стандартные расширения

В.3 Примеры получения идентификаторов

Настоящий подраздел демонстрирует получение нескольких идентификаторов объектов, используя определение, представленное на рисунке В.1.

Примеры

1 Контекст представления определяется как пара абстрактный синтаксис – синтаксис передачи, например:

- Абстрактный синтаксис:

```
nomemle (device not otherwise specified [MOS])
iso(1) member-body(2) US(840) ieee1073(10004) mdap(2) version1(1)
mdap-0(0) standardSpecificExt(0) modules(0) abstractSyntax(1) 1
```

- Синтаксис передачи: mderBigEndian

```
iso(1) member-body(2) US(840) ieee1073(10004) mdap(2) version1(1)
mdap-0(0) standardSpecificExt(0) modules(0) transferSyntax(2) 1
```

2 Контекст приложения, например,

- Контекст приложения: нормальный режим работы контроллера DCC

```
iso(1) member-body(2) US(840) ieee1073(10004) mdap(2) version1(1)
mdap-0(0) standardSpecificExt(0) modules(0) applicationContext(1) dcc(2) 1
```

Приложение С
(справочное)

Временная синхронизация

С.1 Цель

Настоящее приложение зарезервировано для подробной спецификации временной синхронизации пакетов, заданных прикладными профилями, ссылающимся на настоящий стандарт.

С.2 Область применения

Настоящий стандарт в текущий период не определяет метод для временной синхронизации между медицинскими приборами, что позволило бы обеспечить реализации на нижних уровнях (например, ИСО/ИИЭР 11073-30200, ИСО/ИИЭР 11073-30300) и объектно-ориентированную реализацию в профилях форматов, основанных на прикладном уровне (например, DIM).

С.3 Спецификация

Чтобы облегчить реализацию интероперабельности, профили приложений, применяющие SNTP, должны быть согласованы с соответствующими определениями в ИСО/ИИЭР 11073-30200, связанными с протоколами, и определениями в DIM, связанными с объектом clock (часы).

Приложение D
(справочное)

Динамическая модель

Общие определения можно найти в DIM. Настоящее приложение включает в себя дополнительную информацию.

В таблицах настоящего приложения представлена модель конечного автомата (FSM) в табличной форме. Таблица D.1 представляет FSM только для DCC агента, а таблица D.2 только для BCC менеджера. В данных таблицах «только» означает, что ни одна из сторон не обеспечивает симметричные функциональные возможности взаимодействия клиент-сервер.

Профили, использующие настоящий стандарт, должны рассматривать таблицы D.1 и D.2 как подходящие эквивалентные визуализации (т. е., как диаграммы состояний), разъяснения или расширения, а также эти таблицы должны учитываться при реализации заявлений о соответствии.

Таблица D.1 представляет таблицу переходов состояний для системы агента прибора (например, для инфузионной помпы). События, выделенные *курсивом*, являются внешними событиями (например, сгенерированными системой-менеджером). Другие события являются внутренними событиями. Например, если агент в состоянии **Disconnected** (Отсоединен) получает *connect event* (*событие подключения*), то затем он отправляет уведомление о начальной загрузке и меняет свое состояние на **Unassociated** (Неассоциированный).

Пустые поля в таблице D.1 означают, что событие не привело ни к каким действиям или изменениям состояния.

Таблица D.2 представляет таблицу перехода состояний хоста DCC, т. е. менеджера BCC, (например, хоста инфузионной помпы).

Пустые поля в таблице D.2 означают, что событие не привело ни к каким действиям или изменениям состояния. Некоторые состояния, которые указаны в диаграмме состояния, убраны из таблицы для редактирования (переходы должны быть очевидными).

Примечание – Специальное состояние горячего запуска (*hot start*), которое позволяет пропустить состояние **Configuring** (Настройка), еще не было определено и возможно будет добавлено в данные таблицы позже[1)].

[1)] Примечания в тексте, таблицах или на рисунках даются только в справочных целях и не содержат в себе требования, необходимые для реализации настоящего стандарта.

Таблица D.1 — Состояние только DCC агента

Событие	Состояние						
	Отсоединен	Неассоциированный	Ассоциирующий	Ассоциированный	Настройка	Настроенный	Функционирование
Событие подключения (например, обнаруживаемый физический канал)	Неассоциированный						
Уведомление о начальной загрузке сервера		Если ассоциация разрешена {отправка запроса на переход в состояние Ассоциирующий}					
Принятие ассоциации			Отправить принятие ассоциации, переход в состояние Ассоциированный				
Отклонение запроса на ассоциацию			Отправить отклонение ассоциации, переход в состояние Неассоциированный				
Создание события MDS				Если (серверу это необходимо) отправлено уведомление о созданном MDS, то создать сканер CTXT. Переход в состояние Настройка			
Реакция на создание сканера контекста					rx obj. create ERs создает другие сканеры, отправляет изменение состояния MDS на Настроенный		

Продолжение таблицы D.1

Событие	Состояние						
	Отсоединен	Неассоциированный	Ассоциирующий	Ассоциированный	Настройка	Настроенный	Функционирование
Изменение состояния MDS на состояние функционирования [a)]						Функционирование	
Регофигурирование (например, создать/удалить событие)							Обновить конфигурацию. Функционирование
Намерение завершения ассоциации				Завершение	Завершение	Завершение	Завершение
Запрос на выполнение разъединения ассоциации				Отправка ответа о разъединении, сброс конфигурации, переход в состояние Неассоциированный	Отправка ответа о разъединении, сброс конфигурации, переход в состояние Неассоциированный	Отправка ответа о разъединении, сброс конфигурации, переход в состояние Неассоциированный	Отправка ответа о разъединении, сброс конфигурации, переход в состояние Неассоциированный
Прекращение ассоциации			Сброс конфигурации, переход в состояние Неассоциированный	Сброс конфигурации, переход в состояние Неассоциированный	Сброс конфигурации, переход в состояние Неассоциированный	Сброс конфигурации, переход в состояние Неассоциированный	Сброс конфигурации, переход в состояние Неассоциированный
Событие рассоединения		Сброс конфигурации, переход в состояние Отсоединен	Сброс конфигурации, переход в состояние Отсоединен	Сброс конфигурации, переход в состояние Отсоединен	Сброс конфигурации, переход в состояние Отсоединен	Сброс конфигурации, переход в состояние Отсоединен	Сброс конфигурации, переход в состояние Отсоединен
Восстанавливаемая ошибка					Отправка ошибки MDS. Если состояние изменяется {отправка в MDS установить новое состояние}, иначе переход в состояние Настройка	Отправка ошибки MDS. Если состояние изменяется {отправка в MDS установить новое состояние}, иначе переход в состояние Настройка	Отправка ошибки MDS. Если состояние изменяется {отправка в MDS установить новое состояние}, иначе переход в состояние Настройка

Окончание таблицы D.1

Событие	Состояние						
	Отсоединен	Неассоциированный	Ассоциирующий	Ассоциированный	Настройка	Настроенный	Функционирование
Изменение состояния MDS					Проверить: если переход допустим, то подтвердить изменение состояния на новое состояние	Проверить: если переход допустим, то подтвердить изменение состояния на новое состояние	Проверить: если переход допустим, то подтвердить изменение состояния на новое состояние
Невосстанавливаемая ошибка			Сброс конфигурации, отправка указания о прекращении ассоциации (если возможно), переход в состояние Неассоциированный	Сброс конфигурации, отправка указания о прекращении ассоциации (если возможно), переход в состояние Неассоциированный	Сброс конфигурации, отправка указания о прекращении ассоциации (если возможно), переход в состояние Неассоциированный	Сброс конфигурации, отправка указания о прекращении ассоциации (если возможно), переход в состояние Неассоциированный	Сброс конфигурации, отправка указания о прекращении ассоциации (если возможно), переход в состояние Неассоциированный

[a] В случае некоторых сбоев завершение состояния конфигурирования может быть не обнаружено.

Таблица D.2 — Состояние только BCC менеджера

Событие	Состояние						
	Отсоединен	Неассоциированный	Ассоциирующий	Ассоциированный	Настройка	Настроенный	Функционирование
Событие подключения (например, обнаруживаемый физический канал)	Неассоциированный						
Запрос на ассоциацию		Если ассоциация разрешена {отправка запроса на переход в состояние Ассоциирующий}					
Запрос на ассоциацию принят			Отправить принятие ассоциации, переход в состояние Ассоциированный				

Продолжение таблицы D.2

Событие	Состояние						
	Отсоединен	Неассоциированный	Ассоциирующий	Ассоциированный	Настройка	Настроенный	Функционирование
Запрос на ассоциацию отклонен			Отправить отклонение ассоциации, переход в состояние Неассоциированный				
Настройка				Отправить уведомление, созданное MDS, переход в состояние Настройка			
Команда создания сканера контекста					Запуск сканера CTXT, отправить созданный ERs объект, отправить изменение состояния MDS, переход в состояние Настроенный		
Функционирование						Запуск всех эпизодических данных, переход в состояние Функционирование	
Изменение настройки (например, новые/измененные сканеры)							Обновить настройку, функционирование
Намерение завершения ассоциации				Завершение	Завершение	Завершение	Завершение
Запрос на выполнение разъединения ассоциации				Отправка ответа о разъединении, сброс настройки, переход в состояние Неассоциированный	Отправка ответа о разъединении, сброс настройки, переход в состояние Неассоциированный	Отправка ответа о разъединении, сброс настройки, переход в состояние Неассоциированный	Отправка ответа о разъединении, сброс настройки, переход в состояние Неассоциированный

Окончание таблицы D.2

Событие	Состояние						
	Отсоединен	Неассоциированный	Ассоциирующий	Ассоциированный	Настройка	Настроенный	Функционирование
Прекращение ассоциации			Сброс настройки, переход в состояние Неассоциированный	Сброс настройки, переход в состояние Неассоциированный	Сброс настройки, переход в состояние Неассоциированный	Сброс настройки, переход в состояние Неассоциированный	Сброс настройки, переход в состояние Неассоциированный
Событие рассоединения		Сброс настройки, переход в состояние Отсоединен	Сброс настройки, переход в состояние Отсоединен	Сброс настройки, переход в состояние Отсоединен	Сброс настройки, переход в состояние Отсоединен	Сброс настройки, переход в состояние Отсоединен	Сброс настройки, переход в состояние Отсоединен
Восстанавливаемая ошибка					Отправка ошибки MDS. Если состояние изменяется {отправка в MDS установить новое состояние}, иначе переход в состояние Настройка	Отправка ошибки MDS. Если состояние изменяется {отправка в MDS установить новое состояние}, иначе переход в состояние Настройка	Отправка ошибки MDS. Если состояние изменяется {отправка в MDS установить новое состояние}, иначе переход в состояние Настройка
Изменение состояния MDS					Проверить: если переход допустим, то подтвердить изменение состояния на новое состояние	Проверить: если переход допустим, то подтвердить изменение состояния на новое состояние	Проверить: если переход допустим, то подтвердить изменение состояния на новое состояние
Невосстанавливаемая ошибка			Сброс настройки, отправка отмены ассоциации (если возможно), переход к состоянию Неассоциированный	Сброс настройки, отправка отмены ассоциации (если возможно), переход к состоянию Неассоциированный	Сброс настройки, отправка отмены ассоциации (если возможно), переход к состоянию Неассоциированный	Сброс настройки, отправка отмены ассоциации (если возможно), переход к состоянию Неассоциированный	Сброс настройки, отправка отмены ассоциации (если возможно), переход к состоянию Неассоциированный

Приложение Е
(обязательное)

Абстрактный синтаксис

Настоящее приложение определяет несколько специализаций абстрактного синтаксиса, а именно:

- расширения mOSI, относящиеся к сеансовому уровню и уровню представления (см. Е.1);
- модули языка ASN.1, относящиеся к удаленному функционированию приложения, а также к сервисам и протоколам общей управляющей информации (например, ROSE*, CMIP*) (см. Е.2);
- расширения абстрактного синтаксиса и синтаксиса передачи, относящиеся к языку MDDL и правилам MDER (см. Е.3);
- определения прокси MIB (см. Е.4).

Предположения для спецификаций и примеры, используемые в настоящем приложении, включают в себя следующее:

a) воспроизведены синтаксисы ИСО и приведены шестандцатеричные аннотации (дампы) строк октет. Несмотря на то, что данные определения слишком сложные для детального разъяснения, они упрощают реализацию, предоставляя более точную компиляцию отображений между абстрактным синтаксисом и синтаксисом передачи данных. Кодирование правил ИСО BER и правил MDER профиля MDAP рассматривается, основываясь на контексте представления данных, в частности ИСО ACSE/BER и ИСО/ИЭР 11073 MDDL/MDER;

b) для соответствия определениям прикладного уровня расширения сеансового уровня и уровня представления могут быть определены в раздельных модулях ASN.1 (в Е.1.1 и Е.1.2).

Е.1 Расширения mOSI

Расширения профиля MDAP, ориентированные на стандарт, принадлежат к модификациям сеансового уровня и уровня представления mOSI, которые были оптимизированы для использования в медицинских приборах.

Е.1.1 Сеансовый уровень

Полную спецификацию требований mOSI к устройствам сеансового уровня можно найти в ИСО/МЭК ISP 11188-3.

Сеансовый уровень поддерживает только базовый и дуплексный функциональные блоки. Поддерживается механизм протокола базовой конкатенации. Максимально допустимый размер данных пользователя SS должен быть больше 512 байт (например, не поддерживаются срочные данные, сегментирование).

Базовая конкатенация означает, что сеансовый уровень соединяет только один блок SPDU Категории 0 с одним блоком SPDU Класса 2 (в отличие от расширенных конкатенаций, которые могут содержать в себе множественные блоки данных SPDU Категории 2).

В настоящем стандарте перечень поддерживаемых блоков SPDU определяется с помощью текстовой нотации, которая описывает содержание блока данных SPDU. Описанные элементы блоков данных SPDU являются обязательными. Необходимо отметить, что обычно блоки данных SPDU содержат данные пользователя сеансового уровня, которые входят в состав сообщений перечисленных ниже, но не представлены в данном пункте.

Используются следующие сокращения (в соответствии с ИСО/МЭК 8327-1):

- LI: индикатор длины (длина 0–254: один октет, иначе 3 октета, начиная с 255);
- PGI: идентификатор группы параметров (определяет группу параметров сеансового уровня);
- PI: идентификатор параметров (определяет один параметр сеансового уровня);
- SI: идентификатор блока SPDU (уникальный идентификатор, который определяет тип сообщения сеансового уровня).

E.1.1.1 Подключение сеансового уровня (Connect — CN) SPDU

Ниже представлен формат и содержание блока CN SPDU:

```
0D XX                    SI=13 (CN), LI - длина в октетах
   05 08                 PGI=05(элемент connect/accept), LI=08
   13 01 00              PI=19 (опции), LI=1, значение = 0
   16 01 02              PI=22 (версия), LI=1, значение = 2
   80 00                 PI=128 (расширения сеанса MDAP), LI=0
   14 02 00 02           PI=20 (требования пользователя), LI=2, значение: выбран полный дуплексный функциональный блок
C1 YY                    PGI=193 (данные пользователя), LI= длина данных пользователя
```

Обычно данный блок SPDU содержит сообщение о представлении соединения (см. Е.1.2.1), которое, в свою очередь, содержит запрос ассоциации элемента ACSE (см. Е.1.3.1).

E.1.1.2 Приемка сеансового уровня (Accept — AC) SPDU

Ниже представлен формат и содержание блока AC SPDU:

```
0E XX                          SI=14 (AC) LI = длина в октетах
    05 08                      PGI=05(элемен. connect/accept), LI=06
    13 01 00                   PI=19 (опции), LI=1, значение = 0
    16 01 02                   PI=22 (версия), LI=1, значение = 2
    80 00                      PI=128 (расширения сеанса MDAP), LI=0
    14 02 00 02                PI=20 (требования пользователя), LI=2,  значение: выбрать
                               полный дуплексный функциональный блок
C1 YY                          PGI=193 (данные пользователя), LI= длина данных пользова-
                               теля
```

Обычно данный блок AC SPDU содержит сообщение о представлении соединения (см. E.1.2.2), которое, в свою очередь, содержит запрос ассоциации элемента ACSE (см. E.1.3.1).

E.1.1.3 Отказа сеансового уровня (Refuse — RF) SPDU

Ниже представлен формат и содержание блока RF SPDU:

```
0C 03                          SI=12 (RF), LI = длина в октетах (фиксированная: 3)
    32 01 00                   PI=50 (причина), длина 1, причина не указана
```

Блок данных RF SPDU используется в качестве ответа в случае использования не поддерживаемых опций сеанса или поврежденных данных заголовка сеанса (в профиле MDPA используется только такой одиночный блок данных SPDU отказа).

E.1.1.4 Окончание сеансового уровня (Finish — FN) SPDU

Ниже представлен формат и содержание блока FN SPDU:

```
09 XX                          SI=9 (FN), LI = длина в октетах
    C1 YY                      PGI=193 (данные пользователя), LI = длина даты
```

Как правило, данный блок FN SPDU содержит обычное значение данных уровня представления (PDV), которое, в свою очередь, содержит запрос на разъединение связи ACSE (см. E.1.3.1). См. пример на рисунке F.3.

E.1.1.5 Отключение сеансового уровня (Disconnect — DN) SPDU

Ниже представлен формат и содержание блока DN SPDU:

```
0A XX                          SI=10 (DN), LI = длина в октетах
    C1 YY                      PGI=193 (данные пользователя), LI = длина даты
```

Обычно данный блок DN SPDU содержит обычное значение PDV, которое в свою очередь содержит запрос освобождения ассоциации элемента ACSE (см. E.1.3.1). См. пример на рисунке F.4.

E.1.1.6 Передача данных сеансового уровня (Data Transfer – DT) SPDU

Ниже представлен формат и содержание блока DT SPDU:

```
01 00                          SI=1 LI=0 МАРКЕР ПЕРЕДАЧА (GT) SPDU
01 00                          SI=1 LI=0 ПЕРЕДАЧА ДАННЫХ (DT) SPDU
```

Передача данных — это блок SPDU Категории 2, которому должен предшествовать блок SPDU Категории 0 (token *give* (маркер *передача)*, который имеет такое же значение поля SI). Поля LI указывают на то, что нет никаких параметров (например, поля PGI, PI). Информация пользователя просто добавляется к настоящему сообщению.

Блок данных DT SPDU содержит

a) TD PPDU или

b) MDAP-TD PPDU.

E.1.1.7 Прекращение сеансового уровня (Abort – AB) SPDU

Существуют две основные формы блока AB SPDU. Первая – в сообщении нет данных пользователя (см. 8.3.9.3 в ИСО/МЭК 8327-1):

```
19 03                          SI=25 LI = длина в октетах
    11 01 09                   PI=17 (отключение передачи),LI=1, значение=3(бит 1 =
                               = разъединение передачи; бит 4 = нет причины(т.е.,
                               нет ARP/ARU))
```

Вторая форма используется, когда предоставляется информация пользователя:

```
19 XX                          SI=25 LI = длина в октетах
    11 01 03                   PI=17 (отключение передачи), LI=1, значение = 3 (би. 2 =
                               = user abort; бит 1 = разъединение передачи)
    C1 YY                      PGI=193 (данные пользователя), LI= длина данных
```

Блок данных AB SPDU второй формы может содержать одно из следующего:

a) аварийное разъединение связи по инициативе поставщика услуг (ARP) PPDU без данных пользователя;

b) аварийное разъединение по инициативе пользователя (ARU) PPDU, которое содержит аварийное прекращение работы (ABRT) APDU;

c) пустые данные пользователя, в данном случае длина будет равна нулю (например, 0xC1 0x00).

Так как настоящий стандарт не определяет данные о преждевременном прекращении по инициативе пользователя в языке MDDL, информация, содержащаяся PPDU ARP или PPDU ARU, несет минимальную значимость для получателя AB SPDU. Поэтому первая форма только для сессии предпочитает формат AB SPDU.

Независимо от того какая форма используется (т. е. с использованием данных пользователя или без них), получения AB SPDU (SI = 25) будет достаточно для прекращения сеанса.

E.1.1.8 Приемка прекращения сеансового уровня (Abort Accept – AA) SPDU

Блок AA SPDU не используется.

E.1.1.9 Расширения сеансового уровня MDAP

MDAP устанавливает дополнительные службы сеансового уровня. Цель использования расширения сеансового уровня заключается в обеспечении простого и эффективного демультипексирования (распределения каналов) стандартных (mOSI) и нестандартных блоков данных (MDAP) PPDU. В данном случае демультиплексирование возможно посредством единичного бита, и что еще более важно, в заголовке представления можно не указывать правила кодирования BER, которые потребовали бы наличия поля переменной длины.

Расширение сеансового уровня профиля MDAP также обеспечивает консолидирование ресурсов. Подобно расширяемой конкатенации в стандартном сеансовом уровне, множество PPDU может быть соединено в один SPDU. Данная функция полезна для уменьшения количества сообщений, проходящих через нижние уровни (требующие ресурсы для обработки информации).

E.1.1.9.1 Элементы соединения/приемки сеанса профиля MDAP

Использование расширений сеансового уровня профиля MDAP согласовывается во время установления соединения сеанса[1]. В элементе соединения/приемки определяются два дополнительных поля PI:

```
1: B0 00                    PI-128 (запустить расширения сеанса MDAP), LI=0
2: B1 01 xx                 PI-129 (запустить объединение MDAP), LI-1, значение - xx
                                0000 0001b - с периодом 32 ms
                                0000 0010b - с периодом 64 ms
                                0000 0100b - с периодом 128 ms
                                0000 1000b - с периодом 256 ms
                                и т.д.
```

Ниже приведен пример CN SPDU:

```
0D XX                       SI-13 (CN), LI - длина в октетах
05 06                       PGI-05(элемент connect-accept), LI-08
13 01 00                    PI-19 (опции), LI-1, значение - 0
16 01 03                    PI-22 (версия), LI-1, значение - 3
B0 00                       PI-128 (расширения сеанса MDAP), LI-0
14 02 00 02                 PI-20 (требования пользователя), LI-2,  значение: выбрать
                            полный дуплексный функциональный блок
C1 YY                       PGI-193 (данные пользователя), LI- len
```

E.1.1.9.2 SPDU передачи данных профиля MDAP

Расширение сеансового уровня профиля MDPA определяет дополнительные поля SI для передачи обычных и срочных данных. Ниже приведены варианты формата этих полей:

- SPDU передачи нормальных данных MDPA (MDAP-DT)

```
E1 00                       SI-E1h LI - 0
```

- блок SPDU передачи срочных данных MDPA (MDAP- XT)

```
E2 00                       SI-E2h LI - 0
```

Данные сообщения содержат сообщение с представлением профиля MDAP и передаются в расширение уровня представления профиля MDAP вместо нормального уровня представления. Поля PGI и PI не задаются и, следовательно, поле LI равно 0. Эти сообщения определяются как блоки данных SPDU Категории 1, не требующие маркера *передача* (подобно типизированным данным при обычном сеансе).

Данные пользователя просто прикрепляются к этим сообщениям и содержат PDV профиля MDAP (MDAP-PPDU), которое рассматривается как значения PDU протокола CMIP*/элемента ROSE*.

---

[1] Метод согласования определяется для каждого используемого прикладного профиля. Однако в большинстве случаев система менеджера может указывать на наличие у нее поддержки объединения. Если система агента также указывает на то, что она поддерживает объединение, сообщая об этом в своем ответе на запрос ассоциации, то объединение запускается. Менеджер и агент указывают период сброса на диск (flush period), который будет применяться, независимо друг от друга. См. E.1.1.9.3.

E.1.1.9.3 Объединение сеанса профиля MDAP

Чтобы уменьшить общее количество сообщений профиля MDAP, сеансовый уровень профиля MDAP поддерживает объединение, при котором множество PPDU MDAP-DT объединяется и отправляется в одном SPDU. Если размер буфера превысит MTU (максимальный передаваемый блок данных), то буфер SPDU отправляется (передается на нижние уровни) со следующей частью MDAP-DT или спустя определенное количество времени, соответствующее периоду сбрасывания на диск, которое можно установить во время выполнения подключения сеанса.

Если множество PPDU MDAP-DT объединяются, то к компонентам необходимо добавить поля длины такие, чтобы сеансовый уровень мог разделить их на отдельные части (демультиплексировать). Для этого используется поле LI, которое указывает на общую длину блока данных SPDU. В соответствии с ИСО/МЭК 8327-1, используется поле LI длиной в 3 октета с 0xFF в качестве первого октета для кодирования длины в диапазоне 255–65535. Расширение сеансового уровня профиля MDAP использует эту форму представления длины для кодирования длины в диапазоне 0–65535. С учетом этого, начальный октет идентификатора длины LI блока данных SPDU можно использовать для обозначения того, имеет ли блок SPDU один или множество блоков PPDU (обычно это 0x00; 0xFF указывает на объединенные PPDU).

Каждый PPDU, входящий в объединение, также включает в себя длину, что позволяет разделить множество блоков PPDU. В результате блоки PPDU должны представлять собой массив с множеством записей следующей формы (когда объединение включено[1)]):

```
LL LL       Длина PPDU
CC CC       INT-U16 с Presentation Context ID  (ID Контекста представления)
DD DD       MDAP-User-data
```

Поэтому SPDU с тремя встроенными PPDU будет иметь следующую структуру:

```
FF FF       SI-MDAP-DT SPDU; LI-255 указывает на множество PPDU
XX XX       Длина всего блока SPDU
LL LL       Длина PPDU #1
CC CC       INT-U16 с Presentation Context ID
DD DD       MDAP-User-data
LL LL       Длина PPDU #2
CC CC       INT-U16 с Presentation Context ID
DD DD       MDAP-User-data
LL LL       Длина PPDU #3
CC CC       INT-U16 с Presentation Context ID
DD DD       MDAP-User-data
```

Использование объединения увеличивает задержку сообщения, но благодаря этому уменьшается расход пропускной способности (меньшее количество блоков PDU).

Для некоторых функций ответа увеличенное время задержки нежелательно. Поэтому реализация коммуникационного стека MDAP должна обеспечивать функцию размещения в стеке или сброса на диск, способную вызвать передачу в буфере сеанса.

E.1.2 Уровень представления

Как и в случае с протоколом сеансового уровня, у уровня представления профиля MDAP имеется два разных элемента или части. Элемент mOSI обеспечивает стандартное представление базовых и дуплексных функциональных блоков.

Элемент представления профиля MDAP, который должен рассматриваться как расширение обычного протокола уровня представления, можно использовать только для нормальных данных пользователя. Он принимает данные от сеансового элемента профиля MDAP и отправляет данные в сеансовый элемент MDAP. Он не работает со стандартным сеансовым элементом mOSI.

Полную спецификацию требований mOSI для обеспечения уровня представления можно найти в приложении B ИСО/МЭК ISP 11188-3. Как было уже отмечено, уровень представления поддерживает только базовые и дуплексные функциональные блоки.

Поддерживаемые блоки данных PPDU определяются в E.1.2.1–E.1.2.7. Описания блоков PPDU, указанные здесь, являются не полными. Они предназначены лишь для того, чтобы дать представление об основном содержании блоков PDU. Полные определения можно найти в ИСО/МЭК 8327-1. Поддерживаемые опции соответствуют ИСО/МЭК МФС 11188-3. Блоки PPDU кодированы согласно BER, даже если само поле данных пользователя не закодировано согласно BER. Примеры закодированных блоков PPDU можно найти в приложении F.

E.1.2.1 Подключение уровня представления (Connection Presentation – CP) PPDU

Блок CP PPDU определяется следующим образом:

[1)] См. E.1.1.9.

```
CP-type ::= SET {
    mode-selector          0  IMPLICIT Mode-selector,              должен быть «нормаль-
                                                                  ный режим»
    normal-mode-           2  IMPLICIT SEQUENCE {
    parameters
    protocol-version       0  IMPLICIT Protocol-version DEFAULT
                              {version-1},
    presentation-context-definition-list
                              4   IMPLICIT  Presentation-context-
                              definition-list,
    user-data                 User-data OPTIONAL
    } OPTIONAL
    -- должен использоваться только для нормального режима.
    -- должен иметь параметры блока CP PPDU
}
```

Данный блок PPDU содержит сообщение запроса ассоциации в виде данных пользователя (см. E.1.3.1). Оно содержится в блоке CN SPDU (см. E.1.1.1).

Список определения контекста уровня представления (см. E.1.2.8) определяет кортежи абстрактного синтаксиса (например, MDAP) и синтаксис передачи (например, синтаксис передачи MDAP с обратным порядком байтов), идентифицированные как объектные идентификаторы, и назначает идентификатор контекста представления (целое число) для каждой из данных комбинаций. Данный идентификатор состоит максимум из 16 бит.

Список контекстов уровня представления будет содержать одну запись для протокола ACSE (абстрактного синтаксиса и синтаксиса передачи ACSE). Одна дополнительная запись содержит абстрактный синтаксис профиля MDAP со списком возможных синтаксисов передачи (например, один для прямого порядка байтов, один для обратного порядка байтов, если оба поддерживаются).

Дополнительные (необязательные) данные пользователя контекста уровня представления (см. E.1.2.8) определяют контекст приложения с сообщением запроса ACSE (см. AARQ-apdu в E.1.3.1). В свою очередь, AARQ-apdu содержит ассоциативные сведения о пользователе (см. MDSEUserInfo в E.2.3).

E.1.2.2 Принятие подключения уровня представления (Connect Presentation Accept – CPA) PPDU

Блок CPA PPDU определяется следующим образом:

```
CPA-PPDU ::= SET
    mode-selector          [0] IMPLICIT Mode-selector,          - должен быть «нормаль-
                                                                ный режим»
    normal-mode-parameters [2] IMPLICIT SEQUENCE {
    protocol-version       [0] IMPLICIT Protocol-version DEFAULT
                               [version-1],
    presentation-context-definition-list
                               [5] IMPLICIT Presentation,
                                                                context-definition-
                                                                result-list
    user-data              User-data OPTIONAL
       -- должен использоваться только для нормального режима.
}
```

Настоящий блок данных PPDU содержит (положительное) сообщение association response в виде данных пользователя (см. E.1.2.8). Он содержится в AC SPDU (см. E.1.1.2).

Список результатов дает представление о том, какие контексты представления данных приняты или отклонены респондентом. Список результатов будет содержать запись для каждого предложенного контекста представления данных в одной последовательности вместе с соответствующим указанием о принятии или отклонении.

Список результатов контекста представления данных должен принимать контекст ACSE и MDAP. Для контекста MDAP должен быть выбрать один синтаксис передачи.

После этого в ассоциации определяются два контекста представления данных. Другими словами существуют только два действительных идентификатора p-context (контекста представления данных). (Эти два контекста определяют контекстный набор, определенный представлением).

E.1.2.3 Отклонение подключения уровня представления (Connect Presentation Reject – CPR) PPDU

Блок CPR PPDU определяется следующим образом:

```
CPR-PPDU ::= CHOICE {                          -- здесь определяется только нормальный режим
normal-mode-parameters SEQUENCE {
presentation-context-definition-result-list[5]    IMPLICIT Presentation-context
                                                  definition-result-list,
provider-reason                                [10] IMPLICIT Provider-reason,
user-data                                      User-data,
    }
```

Настоящий блок данных PPDU содержит ответное сообщение с отклонением ассоциации в виде данных пользователя, а именно AARE с полем результатов, установленным как rejected-permanent (окончательное отклонение) или rejected-transient (временное отклонение). Он содержится в блоке AC SPDU (несмотря на то, что он не участвует в принятии подключения).

E.1.2.4 Протокол аварийного разъединения по инициативе провайдера (ARP PPDU)

ARP PPDU не содержит данные пользователя. Они содержатся в блоке AB SPDU.

ARP PPDU посылается в случае получения неверно сформированного PPDU по одной из следующих причин:

a) PPDU содержит недопустимое значение параметра PPDU;

b) PPDU содержит непредусмотренный параметр PPDU;

c) PPDU включает в себя непредусмотренный идентификатор контекста представления данных;

d) значение PDV не допустимо.

E.1.2.5 Протокол аварийного разъединения по инициативе пользователя (ARU PPDU)

ARU PPDU содержит сообщение аварийного прекращения ассоциации в виде данных пользователя. Он содержится в SPDU AB.

ARU PPDU посылается, когда приложение требует аварийного разъединения контекста представления данных.

E.1.2.6 Представление данных (TD) PPDU

TD PPDU определяется следующим образом:

```
TD-PPDU ::= User-data
```

Настоящий блок PPDU содержит сообщение user data PDV. Он содержится в DT SPDU. Для определения типов данных пользователя см. E.1.2.8.

Примечание — TD PPDU определяется как часть данного профиля, так как он требуется для минимального уровня представления OSI (ИСО/МЭК ISP 11188-3). Тем не менее, предполагается, что все данные медицинских приборов будут использовать службу расширения уровня представления MDAP-TD. На данный момент не были установлены определенные области применения для TD PPDU.

E.1.2.7 MDAP-TD (представление данных для расширения уровня представления MDAP)

Расширение уровня представления определяет один дополнительный тип блока PPDU следующим образом:

```
MDAP-PPDU ::= SEQUENCE {
    presentation-context-id          INT-U16,
    user-data                        MDAP-User-data
}
```

MDAP-PPDU кодируется с помощью MDER, но не BER.

Идентификатор контекста представления данных является 16-битным (тип «big endian») целым числом. Само значение идентификатора определяется в CP PPDU.

Поле данных пользователя является непрозрачным. Нет конкретной структуры данных (например, нет поля длины). Тем не менее, содержанием обычно является блок APDU ROSE*, который в свою очередь размещает в себе блок APDU протокола CMIP* (см. E.2.1).

Например, начальная часть неподтвержденного отчета о событии будет представлена в MDER следующим образом:

Абстрактный синтаксис	Шестнадцатеричное кодирование
MDAP Normal Data Transfer SPDU (MDAP-DT)	E1 00
presentation-context-id (e.g., #1)	00 01
ROSE* Invoke (ROIVapdu), length	00 01 xx xx
Invoke ID (например, #1, Unconfirmed mode, arg length)	00 01 00 00 xx xx

MDAP-PPDU может содержать только одно значение PDV. Оно содержится в блоке SPDU MDAP-DT и включает приложение MDAP (ROSE* PDU) в качестве данных пользователя.

Расширение уровня представления MDAP согласовывается во время выполнения presentation connect. Версия протокола должна быть явно указана как version-mdap (15).

E.1.2.8 Определения типа данных пользователя уровня представления.
Данные пользователя уровня представления состоят из следующих структур данных:

```
User-data ::= CHOICE {
      APPLICATION 1, IMPLICIT Fully-encoded-data }            -- иной выбор не
                                                              -- учтен

Fully-encoded-data ::= SEQUENCE OF PDV-list
PDV-list ::= SEQUENCE {
     transfer-syntax-name                    Transfer-syntax-name OPTIONAL,
     presentation-context-identifier         Presentation-context-identifier,
     presentation-data-values                CHOICE {
        single-ASN1-type [0] ABSTRACT-SYNTAX.&Type (CONSTRAINED BY {
        -- Тип, соответствующий идентификатору контекста представления -- данных -- }),
        octet-aligned                        [1] IMPLICIT OCTET STRING,
        arbitrary                            [2] IMPLICIT BIT STRING }
        -- Содержит один или несколько PDV того же контекста
        -- представления.
       }
```

Определения PDU требуют дополнительные типы данных, указанные ниже:

```
Mode-selector ::= SET {
     mode-value      [0] IMPLICIT INTEGER {
     x410-1984-mode(0),
     normal-mode (1)         -- это единственный режим, поддерживающий MOAP
                             }
}

     Abstract-syntax-name ::= OBJECT IDENTIFIER

Context list ::= SEQUENCE OF SEQUENCE {
     presentation-context-identifier       Presentation-context-identifier,
     abstract-syntax-name                  Abstract-syntax-name,
     transfer-syntax-name-list             SEQUENCE OF
                                               Transfer-syntax-name
}

Presentation-context-definition-list ::= Context list
Presentation-context-definition-result-list ::= Result-list
Presentation-context-identifier ::= INTEGER   -- MOAP поддерживает
                                                  только 16-битные целые числа
Protocol-version ::= BIT STRING {version-1(0)}

Provider reason ::= INTEGER {
     reason-not-specified(0)                  -- другие причины отброшены для целей насто-
                                              ящего документа
      }

Result ::= INTEGER{
     acceptance (0),
     user-rejection (1),
     provider-rejection(2)
      }
Result-list ::= SEQUENCE OF SEQUENCE {
     result                                [0] IMPLICIT Result,
     transfer-syntax-name                  [1] IMPLICIT Transfer-syntax-name OPTIONAL,
     provider-reason                       [2] IMPLICIT INTEGER {
```

```
        reason-not-specified (0),
        abstract-syntax-not-supported (1),
        proposed-transfer-syntaxes-not-supported (2),
        local-limit-on-DCS-exceeded (3)
                                        }OPTIONAL
    }
Transfer-syntax-name ::= OBJECT IDENTIFIER
```

E.1.3 Приложение

E.1.3.1 Ассоциация

Ассоциация использует только обязательные поля и данные пользователя, полученные из определений ASN.1 ITU-T Рекомендации X.227.

```
ASSOC-apdu ::= CHOICE {
    aarq                AARQ-apdu,      -- Association Request
    aare                AARE-apdu,      -- Association Response
    rlrq                RLRQ-apdu,      -- Assoc. Release Request
    rlre                RLRE-apdu,      -- Assoc. Release Response
    abrt                ABRT-apdu       -- Assoc. Abort
    }

AARQ-apdu ::= [APPLICATION 0] IMPLICIT SEQUENCE {
    protocol-version                [0] IMPLICIT BIT STRING {version1(0)}
                                        DEFAULT {version1},
    application-context-name        [1] Application-context-name,
    user-information                [30] IMPLICIT Association-information
    }

AARE-apdu ::= [APPLICATION 1] IMPLICIT SEQUENCE {
    protocol-version                [0] IMPLICIT BIT STRING {version1(0)}
                                    DEFAULT {version1},
    application-context-name        [1] Application-context-name,
    result                          [2] Associate-result,
    result-source-diagnostic        [3] Associate-source-diagnostic,
    user-information                [30] IMPLICIT Association-information
    }

RLRQ-apdu ::= [APPLICATION 2] IMPLICIT SEQUENCE {
    reason                          [0] IMPLICIT Release-request-reason
    }

RLRE-apdu ::= [APPLICATION 3] IMPLICIT SEQUENCE {
    reason                          [0] IMPLICIT Release-request-reason
    }

ABRT-apdu ::= [APPLICATION 4] IMPLICIT SEQUENCE {
    abort-source                    [0] IMPLICIT ABRT-source
    }
ABRT-source ::= INTEGER {
    acse-service-user(0),
    acse-service-provider(1)
    }

Application-context-name ::= OBJECT IDENTIFIER
Associate-result ::= INTEGER {
```

```
      accepted(0),
      rejected-permanent(1),
      rejected-transient(2)
      }

Associate-source-diagnostic ::= CHOICE {
      acse-service-user          [1] INTEGER {
         null(0),
         no-reason-given(1),
         application-context-name-not-supported (2),
         },
      acse-service-provider      [2] INTEGER {
         null(0),
         no-reason-given(1),
         no-common-acse-version(2)
         }
      }

Association-information ::= SEQUENCE OF EXTERNAL

EXTERNAL ::= [UNIVERSAL 8]   IMPLICIT SEQUENCE {
      direct-reference       OBJECT IDENTIFIER OPTIONAL,
      indirect-reference     INTEGER OPTIONAL,
      data-value-descriptor  ObjectDescriptor OPTIONAL,
      encoding               CHOICE {
          single-ASN1-type       [0] ANY,
          octet-aligned          [1] IMPLICIT OCTET STRING,
          arbitrary              [2] IMPLICIT BIT STRING
          }
}

Release-request-reason ::= INTEGER { normal(0),
                urgent(1),
                user-defined(30)
}
Release-response-reason ::= INTEGER {normal(0),
                not-finished(1),
                user-defined(30)
}
```

E.2 Модули ASN.1

Модули ASN.1, ориентированные на MDAP, относятся к данным пользователя ассоциации и протоколу удаленной операции, а также протоколу CMIP, а именно к ROSE* и CMIP*.

E.2.1 ROSE*

```
MDAP-ROSE DEFINITIONS ::= BEGIN

IMPORTS;

ROSEapdus ::= CHOICE {
      roiv-apdu       [1] IMPLICIT ROIVapdu,       - Remote Operation Invoke
      rors-apdu       [2] IMPLICIT RORSapdu,       - Remote Operation Result
      roer-apdu       [3] IMPLICIT ROERapdu,       - Remote Operation Error
      rorj-apdu       [4] IMPLICIT RORJapdu,       - Remote Operation Reject
      roliv-apdu      [5] IMPLICIT ROLIVapdu     - Linked Invoke
      }
ROIVapdu ::= SEQUENCE {
      invokeID          InvokeIDType,
      operation-value   OPERATION,
      argument          ANY DEFINED BY operation-value
      }

RORSapdu ::= SEQUENCE {
```

```
    invokeID        InvokeIDType,
    SEQUENCE {
        operation-value         OPERATION,
        result                  ANY DEFINED BY operation-value
        }
}

ROERapdu ::= SEQUENCE {
    invokeID            InvokeIDType,
    error-value         ERROR,
    parameter           ANY DEFINED BY error-value
}

RORJapdu ::= SEQUENCE {
    invokeID            InvokeIDType,
    problem             Problem
}

ROLIVapdu ::= SEQUENCE {
    invokeID            InvokeIDType,                   -- использует специальную семантику!!
    linkedID            InvokeIDType,
    operation-value     OPERATION,
    argument            ANY DEFINED BY operation-value
}

Problem ::= INTEGER {
    unrecognizedAPDU(0),
    mistypedAPDU(1),
    badlyStructuredAPDU(2),
    duplicateInvocation(100),
    unrecognizedOperation(101),
    mistypedArgument(102),
    resourceLimitation(103),
    initiatorReleasing(104),
    unrecognizedResultInvocation(200),
    resultResponseUnexpected(201),
    mistypedResult(202),
    unrecognizedErrorInvocation(300),
    errorResponseUnexpected(301),
    unrecognizedError(302),
    unexpectedError(303),
    mistypedErrorParameter(304)
    } (0..65535)
InvokeIDType ::= INTEGER (0..65535) -- как правило INTEGER
--
-- определение OPERATION содержит значения для CMIP*
--
OPERATION ::= INTEGER {
    cmipEventReport(0),
    cmipConfirmedEventReport(1),
    cmipGet(3),
    cmipSet(4),
    cmipConfirmedSet(5),
    cmipAction(6),
    cmipConfirmedAction(7),
    cmipCreate(8),
    cmipDelete(9)
    } (0..65535) -- как правило CHOICE (INT/OID)

--
-- определение ERROR содержит значения для CMIP*
--
```

```
ERROR          ::=    INTEGER {
     noSuchObjectClass(0),
     noSuchObjectInstance(1),
     accessDenied(2),
     noSuchAttribute(5),
     invalidAttributeValue(6),
     getListError(7),
     setListError(8),
     noSuchAction(9),
     processingFailure(10),
     duplicateManagedObjectInstance(11),
     noSuchEventType(13),
     noSuchArgument(14),
     invalidArgumentValue(15),
     invalidScope(16),                    - см. сноску1)
     invalidObjectInstance(17),
     missingAttributeValue(18),
     classInstanceConflict(19),
     unrecognizedOperation(21),
     noSuchInvokeId(22),
     } 0..65535;
END
```

Указанные ниже определения используются для различных полей-идентификаторов в сообщении:

```
InvokeIDType ::= INT-U16          -- обычно INTEGER
OPERATION ::=INT-U16              - обычно CHOICE (INT;OID)
ERROR::= INT-U16
```

Е.2.1.1 Отличия от ROSE ИСО/МЭК

Данные определения в настоящем стандарте имеют следующие отличия от ИСО/МЭК 9072-2:

- Invoke APDU: поле Invoke APDU в ИСО/МЭК 9072-2 содержит дополнительное поле связанного идентификатора, которое используется для механизма связанного ответа. Для элемента CMISE (и CMISE*) данный механизм используется только, если во время ассоциации выбирается функциональный блок множественного ответа (не обязательная опция) (см. 7.2.3, а определения полей ассоциации пользователя MDSE в ИСО/МЭК 9595).

- Правила кодирования языка MDDL не допускают применение дополнительных элементов. Поэтому был определен дополнительный (нестандартный) связанный вызов блока APDU;

- Result APDU: SEQUNECE (последовательность), которая содержит значение операции и поле результата, не являются дополнительными в настоящем стандарте. CMIP требует наличия этих полей в своих сообщениях-ответах;

- Reject APDU: по сравнению со стандартным элементом ROSE Reject APDU в настоящем стандарте более простой. Его структура такая же, как и у других блоков APDU.

Значения задачи отображаются в виде одиночного целочисленного значения вместо выбора, который позволяет установить разделение между проблемными областями;

- Linked invoke APDU: Linked invoke APDU используется, когда ответ на вызов операции приводит к образованию множественных сообщений-ответов. ИСО/МЭК ROSE использует нормальный invoke APDU со связанным полем идентификатора, установленным вместо этого специального блока APDU. Другими словами ИСО/МЭК ROSE не требует Linked invoke APDU.

Однако правила кодирования ASN.1, применяемые в языке MDDL, не поддерживают дополнительные структурные элементы, так как они требуют дополнительной информации о тегировании, что увеличивает затраты на лексический разбор и размер сообщения. Поэтому в случае необходимости добавляют данный специальный блок APDU, который содержит связанный идентификатор;

- InvokeIDType, OPERATION, ERROR: данные типы отображаются на ограниченные целые типы (вместо отображения на неограниченный целый тип для идентификатора вызова или на выбор между идентификатором объекта и неограниченным целым типом для других типов).

Таким образом, основным отличием от стандартного протокола ROSE является наличие необязательных элементов, которые либо присутствуют либо отсутствуют в определении ROSE*, в зависимости от того, используются или не используются они протоколом CMIP*. Определение операции (operation) и ошибки (error) упрощены согласно способу их использования протоколом CMIP*. Определен новый блок данных APDU для взаимодействия со связанными ответами.

---

1) Scope (область применения) — это диапазон объектов, к которым применяется операция, например, единичный объект или набор объектов, содержащийся в единичном объекте. О дополнительных технических характеристиках см. Е.2.2.

В настоящем стандарте отсутствуют нестандартные поля, что является результатом адоптации ROSE для конкретных целей POC MDC.

В результате этого заголовок ROSE имеет постоянный размер и фиксированную структуру, которая обеспечивает эффективный лексический разбор (когда не используется связанный вызов блока APDU).

E.2.1.2 Поля сообщения ROSE*

Объяснение полей в блоках данных APDU элемента ROSE дано в E.2.1.2.1—E.2.1.2.5.

E.2.1.2.1 Поле Invoke identifier (идентификатор вызова)

Поле идентификатора вызова — это числовое поле, которое идентифицирует сообщение со стороны отправителя. Если, к примеру, клиент отправляет сообщение-вызов на сервер, то сервер помещает данный идентификатор вызова в получившееся сообщение (например, в результат GET или подтверждение Event Report). Данный механизм позволяет клиенту соотнести ответ с его запросом. (Идентификатор вызова дублируется обратно клиенту в сообщении-ответе.)

Вызыватель операций должен убедиться, что идентификаторы вызова (как минимум те, что обрабатываются в данный момент в системе) уникальны.

Другими словами для реализации процесс получения или процедура получения может быть зарегистрирована в компоненте AE языка MDDL на вызываемом клиенте, как только придет сообщение-ответ.

E.2.1.2.2. Поле Linked identifier (связанный идентификатор)

В случае отправки множества ответов (множество сообщений-результатов) на вызов операции поле связанного идентификатора будет содержать значение идентификатора вызова операции.

Как и идентификатор вызова в сообщении-результате, связанный идентификатор позволяет получателю результатов определить, какой процесс вызвал операцию.

E.2.1.2.3 Поле Operation value (значение операции)

Пользователь или предоставитель услуги удаленной операции определяет операции, которые могут быть использованы. Каждая операция имеет соответствующее числовое значение в целях идентификации (определения см. в E.2.2).

E.2.1.2.4 Поле Error value (значение ошибки)

Как и значение операции, числовые значения устанавливаются для определенных условий ошибок. Допустимые значения ошибок для языка MDDL определены в CMIP*. Необходимо отметить, что значение ошибки не является значением операции.

Примечание — Требуется некоторое разъяснение. Рассмотрим команду GET протокола CMIP* в качестве примера. Предположим, что поле класса управляемого объекта в данном сообщении неправильное, тогда APDU ошибки удаленной операции (remote operation error, ROER) отправляется обратно, а значение ошибки это noSuchObjectClass. Другими словами блок APDU не передает информацию, которая бы указывала на то, что он поступил в ответ на запрос GET. Данную информацию можно извлечь при изучении идентификатора вызова, который является ссылкой на исходное сообщение-запрос GET: отсюда следует требование, чтобы (активные) идентификаторы вызова в системе были уникальными.

E.2.1.2.5 Поле Argument (аргумент)

Удаленный вызов операции можно рассматривать как (удаленный) вызов процедуры. Аргументы, которые передаются процедуре, просто прикрепляются к сообщению элемента ROSE*. В языке MDDL блоки PDU протокола CMIP* являются аргументами сообщений ROSE*.

E.2.1.3 Обработка связанных ответов

В некоторых случаях вызов операции может привести к образованию множества сообщений-ответов. В языке MDDL это может произойти в случае операций, входящих в область действия, или в случае операций (например, GET), которые возвращают большие количества данных. Данные множественные ответы все связаны с одним идентификатором вызова (отсюда связанные ответы).

В отношении связанных ответов применяется следующее:

- для всех сообщений-ответов, кроме самого последнего:
- используется вызов линии передачи удаленной операции (ROLIV) APDU;
- поле идентификатора вызова устанавливается респондентом и имеет особую семантику, которая позволяет обнаруживать отсутствующие части;
- поле связанного идентификатора имеет значение поля идентификатора вызова запроса;
- для самого последнего сообщения:
- используется APDU remote operation result (RORS);
- поле идентификатора вызова в данном ответе имеет значение поля идентификатора вызова запроса.

Примечание — Если необходимы два сообщения в ответе, первое – это APDU ROLIV с последним битом вызова с установленным значением 1 и со счетчиком, установленным на 1;

- для ROLIV APDU поле идентификатора вызова имеет следующую семантику:

```
ROLIV-Invoke-ID ::= SEQUENCE {
    state INT-U8 {
       roliv-first(1),                -- это первый apdu ROLIV
       roliv-not-first not-last(2),
       roliv-last(3)                  -- это последний apdu ROLIV, далее - следует один apdu RORS
```

```
    },
    countINT-U8                    -- счетчик начинает со значения 2 -- (в первом состоянии)
}
```

С помощью этого определения можно обнаружить недостающие части полного набора сообщений-ответов.

E.2.2 CMIP*

Так же как и в случае с ROSE*, CMISE* является частью прикладного уровня. CMISE предоставляет услуги управления объектами, которые обеспечивают доступ к значениям атрибутов, функциям вызова объеков и т.д. Ассоциированный протокол (CMIP) определяет сообщения прикладного уровня (блоки APDU), которые инициируют работу данных служб. Для выполнения своих функций CMISE располагается на уровне, расположенном поверх ROSE.

Поэтому сообщения протокола CMIP обычно задаются посредством ASN.1 макросов ROSE (ИСО/МЭК 9596-1). Для облегчения понимания в настоящем подразделе не используется нотация макрокоманд (макросов) ROSE. Используемые в настоящем пункте определения ASN.1 можно рассматривать как макрорасширения. Однако, как было уже разъяснено, определения сообщений здесь не полностью соответствуют стандартному протоколу CMIP. Несмотря на то, что все поля данных, отправляемые в блоках APDU протокола CMIP*, существуют также в стандартном протоколе CMIP, для упрощения определений типы данных были изменены.

Сообщение протокола CMIP* (более точно: структура данных аргумента операции протокола CMIP*) просто присоединяют в качестве поля аргумента к блоку данных APDU элемента ROSE*. Поле значения операции ROSE* должно быть для определения типа присоединенного аргумента.

В таблице Е.1 показаны, какие типы аргументов (типы сообщений) определяются или используются в протоколе CMIP*.

- Тип данных вызова добавляется к APDU вызова удаленной операции (ROIV).
- Тип данных ответа прилагается к APDU RORS.
- Сообщения об ошибке в APDUS ROER обрабатываются по-разному.

Таблица Е.1 — Типы аргумента протокола CMIP*

Тип сообщения	Значение операции	Тип данных вызова	Тип данных ответа
Event Report (Отчет о событии)	0	EventReportArgument	—
Confirmed Event Report (Подтвержденный отчет о событии)	1	EventReportArgument	EventReportResult
Get (Получить, всегда подтвержденный)	3	GetArgument	GetResult
Set (Установить)	4	SetArgument	—
Confirmed Set (Подтвержденная установка)	5	SetArgument	SetResult
Action (Действие)	6	ActionArgument	—
Confirmed Action (Подтвержденное действие)	7	ActionArgument	ActionResult
Create (Создание, всегда подтвержденный)	8	CreateArgument	CreateResult
Delete (Удаление, всегда подтвержденный)	9	DeleteArgument	DeleteResult

Значения операции соответствуют ИСО/МЭК 9596-1.

Значение операции в таблице Е.1 является значением, которое присваивается полю значения операции в блоках данных APDU вызова и ответа ROSE*. (В данном случае не требуется операция связанного ответа.)

Если, к примеру, хост-система требует значение атрибута какого-либо объекта из сервера, то она отправляет APDU ROIV элемента ROSE* со значением операции 3 и присоединенный к нему GetArgument. Сервер отвечает блоком RORS APDU со значением операции 3 и присоединенным GetResult. Поэтому и тип PDU элемента ROSE* и значение операции требуются для определения типа присоединенных данных (т.е. типа сообщения CMIP*).

В случае возникновения ошибки в качестве ответа высылается PDU ROER элемента ROSE*, а поле значения ошибки содержит код ошибки (см. определения ниже). Дополнительно, если длина поля ANY DEFINED BY > 0, то предоставляется дополнительная информация.

Все сообщения-ответы и сообщения-вызовы (сообщения об ошибке) протокола CMIP* определены ниже в данном пункте. Определения взяты из ИСО/МЭК 9596-1 с объяснением различий.

Типы данных, которые присоединяются к блоку данных APDU элемента ROSE* (т.е. блоку вызова, результата или ошибки), в следующих определениях ASN.1 выделены жирным шрифтом:

```
MDAP-CMIP DEFINITIONS ::= BEGIN
-- Нижеуказанное - это повсеместно используемые типы синтаксиса;
  - может быть полезным определить их в отдельном модуле и
-- импортировать их (IMPORT), как например, MDDL-TYPES Begin MDDL-TYPES
```

```
RelativeTime ::= INT-U32 -- 32-битный целочисленный  тип

OID-Type ::= INT-U16 -- 16-битный целочисленный  тип

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

AttributeList ::= SEQUENCE OF AVA-Type

AttributeIdList ::= SEQUENCE OF OID-Type

ManagedObjectId ::= SEQUENCE {
   m-obj-class          OID-Type,
   m-obj-inst           GLB-HANDLE
}

-- End MDDL-TYPES

EventReportArgument ::= SEQUENCE {
   managedObject          ManagedObjectId,
   eventTime              RelativeTime,
   eventType              OID-Type,
   eventInfo              ANY DEFINED BY eventType
   }

EventReportResult ::= SEQUENCE {
   managedObject          ManagedObjectId,
   currentTime            RelativeTime,
   eventType              OID-Type,
   eventReplyInfo         ANY DEFINED BY eventType
-- Примечание - Каждый класс объекта должен определять особый формат
-- eventType, в качестве результата.
-- Это не указано в настоящем документе. В отношении
-- DIM, см. определение «ReplyInfo» для индивидуального класса объекта.
}

GetArgument ::= SEQUENCE {
   managedObject             ManagedObjectId,
   scope                     Scope,
   attributeIdList           AttributeIdList
   }

GetResult ::= SEQUENCE {
   managedObject             ManagedObjectId,
   attributeList             AttributeList
   }

GetError ::= SEQUENCE {
   errorStatus               ErrorStatus,
   attributeId               OID-Type
   }

GetListError ::= SEQUENCE {
   managedObject              ManagedObjectId,
   getInfoList                SEQUENCE OF GetError
   }

ModifyOperator ::= INTEGER {
   replace(0),
   addValues(1),
```

```
    removeValues(2),
    setToDefault(3)
    } (0..65535)

AttributeModEntry ::= SEQUENCE {
    modifyOperator          ModifyOperator,
    attribute               AVA-Type
    }

ModificationList ::= SEQUENCE OF AttributeModEntry

SetArgument ::= SEQUENCE {
    managedObject           ManagedObjectId,
    scope                   Scope,
    modificationList        ModificationList
    }

SetResult ::= SEQUENCE {
    managedObject           ManagedObjectId,
    attributeList           AttributeList
    }

SetError ::= SEQUENCE {
    errorStatus             ErrorStatus,
    modifyOperator          ModifyOperator,
    attributeId             OID-Type
    }
SetListError ::= SEQUENCE {
    managedObject           ManagedObjectId,
    setInfoList             SEQUENCE OF SetError
    }

ActionArgument ::= SEQUENCE {
    managedObject           ManagedObjectId,
    scope                   Scope,
    actionInfo              ActionInfo
    }

ActionInfo ::= SEQUENCE {
    actionType              OID-Type,
    actionInfoArgs          ANY DEFINED BY actionType
    }

ActionResult ::= SEQUENCE {
    managedObject           ManagedObjectId,
    actionReply             ActionReply
    }

ActionReply ::= SEQUENCE {
    actionType              OID-Type,
    actionInfoArgs          ANY DEFINED BY actionType
    }

CreateArgument ::= SEQUENCE {
    managedObjectClass      OID-Type,
    superiorManagedObject   ManagedObjectId,
    attributeList           AttributeList
    }

CreateResult ::= SEQUENCE {
managedObject               ManagedObjectId,
attributeList               AttributeList
}
```

```
DeleteArgument ::= SEQUENCE {
   managedObject                ManagedObjectId,
   scope                        Scope
   }

DeleteResult ::= SEQUENCE {
   managedObject                ManagedObjectId
   }

Scope ::= INTEGER { base-object(0) } (0.. 4294967295)- см. сноску [1]

NoSuchAction ::= SEQUENCE {
   managedObjectClass           OID-Type,
   actionType                   OID-Type
   }
NoSuchArgument ::= SEQUENCE {
   managedObjectClass           OID-Type,
   eventType                    OID-Type
   }

NoSuchEventType ::= SEQUENCE {
   managedObjectClass           OID-Type,
   eventType                    OID-Type
   }

ErrorStatus ::= INTEGER {
   attr-accessDenied(2),            -- GET, SET
   attr-noSuchAttribute(5),         -- GET, SET
   attr-invalidAttributeValue(6),   -- SET
   attr-invalidOperation(21),       -- SET
   attr-invalidOperator(25)         -- SET
   } (0..65535)

   ProcessingFailure ::= SEQUENCE {
   error id                     OID-Type,     - use MDC
   error info                   ANY DEFINED BY error-id
   }
END
```

E.2.3 Ассоциированные сведения о пользователе

Информация о пользователе присоединяется к сообщениям с запросами и ответами на ассоциации. В случае MDSE определение этих полей пользователя имеет минимальный размер по причине сложности кодирования и обработки сообщений ACSE. Необходимо отметить, что поля пользователя в MDSE закодированы в согласованном синтаксисе передачи (MDER), а не в BER.

Информация о пользователе для MDSE определяются следующим образом (в виде модуля ASN.1):

```
MDSEUserInfo ::= SEQUENCE {
protocolVersion                 ProtocolVersion,
nomenclatureVersion             NomenclatureVersion,
functionalUnits                 FunctionalUnits,
systemType                      SystemType,
startupMode                     StartupMode,
optionList                      AttributeList,          -- данное поле
                                                        -- зарезервировано для
                                                        -- будущих расширений
supportedAProfiles              AttributeList           -- содержит атрибуты
                                                        -- Profile Support

                                                        -- (поддержка профиля)
}
```

[1] Как минимум должна поддерживаться область применения объекта base-object (базовый объект), но прикладные профили, применяющие данный стандарт, могут устанавливать дополнительные области применения для соответствующих случаев.

```
ProtocolVersion ::=BITS-32          -- значение ссылается на определенный
                                    -- стандарт MDAP
NomenclatureVersion ::= BITS-32     -- значение ссылается на определенный
                                    -- стандарт номенклатуры
FunctionalUnits ::= BITS-32 {
   extendedGetSelection (0),        -- поддерживает области действия
                                    -- отличные от области базового объекта
                                    -- Бит 1 зарезервирован [filter(1)]
   multipleReply(2)                 -- поддерживает множественные связанные
                                    -- ответы
                                    -- Бит 3 зарезервирован
                                    -- [extendedService(3)]
                                    -- Бит 4 зарезервирован [cancelGet(4)]
   }
SystemType ::= BITS-32 {
   sys-type-manager(0),
   sys-type-agent(8)
   }

StartupMode ::= BITS-32 {
   hot-start(0),
   warm-start(1),
   cold-start(2)
   }
END
```

Применяются обычные правила расширяемости: неизвестные теги игнорируются, неизвестные биты в строке битов игнорируются. Если дополнительные атрибуты или дополнительные поля приводят к несовместимости, версию протокола следует изменить.

Примечания

1 Если в поле версии протокола или в поле версии номенклатуры не установлены никакие биты, то подразумевается, что сами используемые версии закодированы в поле optionList. Подобное использование поля optionList оставляет возможность для будущих расширений, когда все биты версий будут присвоены.

2 Версия номенклатуры здесь требует изменения только в том случае, если какие-либо номенклатурные коды, которые используются в поле информации о пользователе или в исходном сообщении MDS create event, были модифицированы. Другие версии (незначительные изменения версии) кодируются в соответствующем атрибуте объекта MDS.

3 Данная информация о пользователе сервисного элемента ACSE ссылается на контекст представления MDER, заданный в определении контекста представления данных и на список результатов сообщений ACSE. Поле информации о пользователе не использует правила BER.

4 Идентификаторы атрибутов в структурах списков optionList и supportedAProfiles определены в таблице номенклатуры элементов инфраструктуры.

E.3 Расширения абстрактного синтаксиса

Большинство абстрактных синтаксисов, специфичных для медицинских приборов, определено в MDDL объектно-ориентированной модели DIM. По традиции, классы объекта, атрибуты, группы атрибутов, уведомления, действия и другие номенклатуры определяются в общей номенклатуре языка MDDL (ИСО/ИИЭР 11073-10101).

Тем не менее, идентификация контекста представления управления ассоциацией ИСО требует одного идентификатора объекта для отображения абстрактного синтаксиса. В результате в языке MDDL выделяются расширения для идентификатора объекта MDAP для абстрактного синтаксиса, в частности 16-битные зависящие от контекста и 32-битные контекстно-независимые разделы.

E.4 Определения базы MIB

В состав определений базы MIB входят расширения идентификаторов объектов и абстрактный синтаксис, представляющий содержание информации.

Идентификаторы объектов для расширений MIB выделяются из asn1Modules(2) mib(4), как указано на рисунке B.2. Затем номенклатура присоединяется к ребру mib(4), согласно определению в Таблицах коммуникационной инфраструктуры номенклатуры стандарта ИСО/ИИЭР 11073-10101.

Приложение F
(справочное)

Примеры блоков PDU

Настоящее приложение дает представление о блоках PDU с учетом абстрактного и конкретного синтаксиса (кодирования) для облегчения понимания и реализации.

Первые примеры представляют блоки PDU стадии ассоциации ИСО, которые основаны на языке ASN.1 и правилах BER (см. F.1). Дополнительные примеры показывают блоки PDU стадии конфигурации и функционирования (см. F.2 и F.3, соответственно). Подробные данные по абстрактному синтаксису содержатся в модели DIM, а информация о правилах кодирования (MDER) в приложении А.

В настоящем приложении используются два типа форматов блоков PDU:

- первый формат обеспечивает декомпозицию абстрактного синтаксиса, связанное с ней кодирование (шестнадцатеричный формат) и соответствующие примечания. Пример данного формата см. на рисунках F.1—F.5;
- второй формат для краткости не сопровождается абстрактным синтаксисом, но просто предоставляет кодирование с примечаниями. В подобных случаях для получения подробной информации по абстрактному синтаксису см. модель DIM. Пример данного формата см. на рисунках F.6 – F.7, на которых показаны примеры блоков PDU object create notification event report и create notification event report response соответственно.

Примеры были переняты из проектов демонстрации медицинских вентиляционных и инфузионных приборов. Насколько это практически возможно, значения полей являются точными, хотя некоторые погрешности и необходимы для представления информации в данном формате. В частности, показаны ответвления ИСО/ИИЭР 11073, основыванные на списке в приложении В, но шестнадцатеричные списки не показаны.

F.1 Ассоциация

В число примеров ассоциаций входят запрос и ответ ассоциации (см. рисунки F.1 и F.2, соответственно), запрос на разъединение и ответ на разъединение ассоциации (см. рисунки F.3 и F.4, соответственно) и прекращение ассоциации (см. рисунок F.5).

F.2 Настройка

Примеры включают в себя отчет о событии уведомления создания объекта системы MDS и ответ на данный отчет (см. рисунки F.6 и F.7, соответственно).

F.3 Операция

Примеры включают в себя периодический отчет о сканере событий и буферизированный отчет о сканере событий (см. рисунки F.8 и F.9 соответственно).

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ

Структура		Шестнадцатеричные коды	
-- Session Connect PDU			
Session Header (CN SPDU)		0d de	-- SI='0d'==13 (CN), LI = длина в октетах ='de'==222
		05 08	-- PGI='05'==5 (элемент connect/accept), LI='08'==8
		13 01 00	-- PI='13'==19 (опции), LI=1, значение = 0
		16 01 02	-- PI='16'==22 (версия), LI=1, значение = 2
		80 00	-- PI='80'==128 (включение Сеансовых Расширений MDAP), LI=0
			-- Примечание – По умолчанию объединения нет Чтобы подтвердить включение объединения
			-- добавить + 81 01 xx, где xx - период объединения (см. определение pdu).
		14 02 00 02	-- PI=20 (требования пользователя).
			-- LI=2, значение выбрать полный дуплексный функциональный блок
		c1 ce	-- PGI='c1'==193 (данные пользователя). LI= длина данных пользователя ='ce' == 206
-- Connect Presentation – Presentation PDU			
CP-PPDU ::= SET {	(1)	31 80	-- [CONSTRUCTED]+Tag #17
-- mode-selector			
[0] IMPLICIT Mode-selector	(2)	a0 80	-- [context- specific +CONSTRUCTED]+ Tag #0 "[0]"
::= SET {mode-value [0]			
IMPLICIT INTEGER { normal-mode(1)		80 01 01	-- должен быть "normal mode" (т.е. единственный режим, поддерживаемый MDDL)
}	(2)	00 00	-- end Mode-selector
-- normal-mode-parameters			
[2] IMPLICIT SEQUENCE{	(3)	a2 80	-- [context- specific +CONSTRUCTED] + Tag #2 "[2]"
-- protocol-version			
[0] IMPLICIT Protocol-version ::= BITSTRING{ version-mdap(15)}		a0 03 00 00 01	-- Должно быть version-mdap, см. E.1.2.7
			-- [context-specific+CONSTRUCTED]+Tag #0 "[0]" + len + unused bits + BITS-16
-- presentation-context-definition-list			
[4] IMPLICIT Presentation-context-definition-list ::= Context-list	(4)	a4 80	-- [context-specific+CONSTRUCTED] + Tag #4 "[4]"
::= SEQUENCE OF			
--Определение контекста представления данных #1			
SEQUENCE	(5)	30 80	-- Контекст представления данных #1 - ISO ACSE ASN 1 / ISO BER
{ Presentation-context-identifier ::= INTEGER		02 01 01	
Abstract-syntax-name ::= OBJECT IDENTIFIER		06 04	-- ИСО ACSE версия1
{ joint-iso-ccitt association-control(2)		52 01 00 01	
abstract-syntax(1) apdus(0) version1(1) }			
SEQUENCE OF	(6)	30 80	
Transfer-syntax-name ::= OBJECT IDENTIFIER		06 02 51 01	-- ИСО BER
}	(6)	00 00	
}	(5)	00 00	-- end Определения контекста представления данных #1

Примечание — Данный пример основан на запросе ассоциации (агента) медицинского прибора искусственной вентиляции легких.

Рисунок F.1 – Пример форматирования запроса ассоциации (Association request)

```
-- Определение контекста представления данных #2
    SEQUENCE                                    (7)   30 80      -- Контекст представления данных  #2: ИИЭР 1073 MDDL/ ИИЭР 1073 MDER
    { Presentation-context-identifier  = INTEGER                  02 01 02
    Abstract-syntax-name  ::= OBJECT IDENTIFIER                   06 0c     -- Абстрактный синтаксис: MDDL 16-битная номенклатура iso(1) member- body(2) US(840)
                                                                            -- ieee1073(10004) mdap(2) version1(1) mdap-0(0) standardSpecificExt(0) modules(0)
                                                                            -- abstractSyntax(1) nomen16(1)
                                                                            -- iso(1) X 40 + member-body(2) = 40 = 0x2A, US(840) = 0x348 = 11 0100 1000 => 0x80 +
                                                                            -- 110, 100 1000 = 0x86 0x48
                                                                  2A 86 48 CE 14 02 01 00 00 00 01 01
    SEQUENCE OF                                 (8)   30 80
    Transfer-syntax-name  = OBJECT IDENTIFIER                     06 0c     -- Синтаксис передачи: MDER iso(1) member-body(2) US(840) ieee1073(10004) mdap(2)
                                                                            -- version1(1) mdap-0(0) standardSpecificExt(0) modules(0) transferSyntax(2)
                                                                            -- mderBigEndian(1)
                                                                  2A 86 48 CE 14 02 01 00 00 00 02 01
    }                                           (8)   00 00
  }                                             (7)   00 00     -- конец определения контекста представления данных #2
}                                               (4)   00 00     -- конец списка определения контекста представления данных
-- (нормальный режим) Данные пользователя
User-data  = CHOICE { [APPLICATION 1] IMPLICIT Fully-encoded-data}  (9)   61 80     -- данные пользователя уровня представления
 = SEQUENCE OF PDV-list
  = SEQUENCE {                                (10)  30 80
    Presentation-context-identifier  = INTEGER                    02 01 01  -- Используйте контекст представления данных #1 для кодирования User-data (т е
                                                                            кодирования BER)
    CHOICE { single-ASN.1-type[0] ANY           (11)  a0 80
      { AARQ-apdu
       = [APPLICATION 0] IMPLICIT SEQUENCE {    (12)  60 80
      protocol-version [0] IMPLICIT BIT STRING { version-1(0) }                 -- PPDV User-data == AARQ (Запрос ACSE I)
      application-context-name [1] Application-context-name
                                                (13)  a1 80     -- DEFAULT {версия-1}, опущен
      ::= OBJECT IDENTIFIER                             06 0c
                                                                            -- Application Context Normal Operation Mode iso(1) member-body(2) US(840)
                                                                            -- ieee1073(10004) mdap(2) version1(1) mdap-0(0) standardSpecificExt(0) modules(0)
                                                                            -- applicationContext(3) normalMode(1)
                                                                  2A 86 48 CE 14 02 01 00 00 00 03 01
                                                (13)  00 00
      user-information [30] IMPLICIT Association-information
                                                (14)  be 80     -- context-specific + CONSTRUCTED + Tag #30 "[30]"
       = SEQUENCE OF EXTERNAL
      EXTERNAL                                  (15)  28 80     -- CONSTRUCTED + Tag #8 "EXTERNAL"[1)]
      ::= IMPLICIT SEQUENCE {                                   -- MDDL a-Associate User Info - Закодировано с помощью контекста представления
                                                                  данных #2 (MDER)
```

[1)] Тип EXTERNAL (внешний) обеспечивает получателя сообщения информацией, которая необходима ему, чтобы должным образом переключать контексты представления данных и должным образом обработать последующую информацию, которая в данном случае закодирована с помощью правил MDER

Рисунок F.1, лист 2

direct-reference [0] = OBJECT IDENTIFIER,	06 0c	-- Идентификатор объекта для Контекста представления MDAP *iso(1) member-body(2) -- US(840) -- ieee1073(10004) mdap(2) version1(1) mdap-0(0) standardSpecificExt(0) modules(0) -- transferSyntax(2) mderBigEndian(1)*
indirect-reference [1] = INTEGER	2A 86 48 CE 14 02 01 00 00 00 02 01	
encoding [2] = CHOICE { [1] IMPLICIT OCTET STRING	02 01 02	-- Идентификатор контекста представления данных (ID‡) для MDAP (согласно примечанию 2 в X.209 34)
	81 3A	
-- Информация о пользователе MDDL		-- Примечание – Остальная часть находится в выбранных правилах кодирования, например MDER.
MDSEUserInfo [1] = SEQUENCE {		
protocolVersion [0] IMPLICIT ProtocolVersion	80 00 00 00	-- Протокол версии 1 [некоторые старшие биты используются для CtI младшей версии]
[0] = BITS-32 { mddl-version1 (0), }		
nomenclatureVersion [1] IMPLICIT NomenclatureVersion		
[1] = BITS-32 { nomen-version1(1), }	40 00 00 00	-- Номенклатура версии 1
functionalUnits [2] IMPLICIT FunctionalUnits		
[2] = BITS-32 {extendedObjectSelection(0) }	00 00 00 00	-- Расширенная функция выбора объектов не используется
systemType [3] IMPLICIT SystemType		
[3] = BITS-32 { sys-type-manager(0), }	80 00 00 00	-- [Медицинские данные] сервер
startupMode [4] IMPLICIT StartupMode		
[4] = BITS-32 { cold-start(2), }	20 00 00 00	-- Зависит от реализации прибора (для Demo . . фиксирован при холодном запуске)
-- расширения INTEROP		
-- optionList AttributeList,	00 00 00 00	-- ZERO AttributeList
-- supportedAProfiles AttributeList		
[1] = SEQUENCE OF AVA-Type	00 01 00 1E	-- Идентифицирован один профиль, длина = 30
	00 02 00 1A	-- OID-Type == NOM_BASELINE_PROFILE_SUPPORT (Раздел инфраструктуры)
	80 00 00 00	-- BaselineRevision :baseline-rev-0(0)
	00 00 00 00	-- max-mtu-rx ~ макс. размер получаемого сообщения в байтах
	00 00 00 00	-- max-mtu-tx ~ макс. размер передаваемого сообщения в байтах
	FF FF FF FF	-- max-bw-tx ~ макс. передаваемый диапазон частот (0xFFFFFFFF=не указано)
	00 01	-- max-mds-hierarchy ~ макс. глубина иерархии объекта MDS, -- из корня MDIB до самого глубокого MDS
	00 00 00 00	-- BaselineOptions ~ Отсутствие динамического создания/удаления объектов -- FUTURE: будет добавлен Patient Demographics Opt Pkg
	00 00 00 00	-- optional-packages ~ ZERO AttributeList -- Примечание – Позднее необходимо добавить Patient Demographics opt pkg
(15)	00 00 (14) 00 00	
(12)	00 00 (11) 00 00	
(10)	00 00 (9) 00 00	
(3)	00 00 (1) 00 00	

‡ Данное значение включено в поле идентификатора контекста представления данных всех сообщений языка MDDL.

Рисунок F.1, лист 3

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ

Структура		Шестнадцатеричные коды	
-- Session Accept PDU			
Session Header (AC SPDU)		0e c0	-- SI='0e'==13.(CN), LI = длина в октетах='c0'== 192.
		05 08	-- PGI='05'==5 (элемент connect/accept), LI='08'==8
		13 01 00	-- PI='13'==19 (опции), LI=1. value = 0
		16 01 02	-- PI='16'==22 (версия). LI=1, value = 2
		80 00	-- PI='80'==128 (Расширения сеанса MDAP), LI=0
			-- П р и м е ч а н и е — По умолчанию объединения нет Чтобы подтвердить включение объединения
			-- добавить → 81 01 xx, где xx → период объединения (см определение pdu)
		14 02 00 02	-- PI=20 (требования пользователя) LI=2, значение: выбрать полный дуплексный функциональный
			-- блок
		c1 b0	-- PGI='c1'==193 (данные пользователя), LI= длина данных пользователя ='b0' == 176
[Session user data == Presentation Connect Accept PDU]			
CPA-PPDU ::= SET {	(1)	31 80	-- [CONSTRUCTED]+Tag #17
-- mode-selector			
[0] IMPLICIT Mode-selector	(2)	a0 80	-- [context- specific +CONSTRUCTED]+ Tag #0 "[0]"
::= SET {mode-value [0]			
IMPLICIT INTEGER { normal-mode(1) }		80 01 01	-- должен быть «нормальный режим» (т е единственный режим, поддерживаемый языком MDDL)
}	(2)	00 00	-- окончание выбора режима
-- normal-mode-parameters			
[2] IMPLICIT SEQUENCE{	(3)	a2 80	-- [context- specific +CONSTRUCTED] + Tag #2 "[2]"
-- protocol-version			
[0] IMPLICIT Protocol-version ::= BIT STRING { version-mdap(15) }		A0 03 00 00 01	-- Версия должна быть version-mdap; см. пункт E 1 2.7
-- presentation-context-definition-result-list			
[5] IMPLICIT Presentation-context-definition-result-list Result-list	(4)	a5 80	-- [context-specific+CONSTRUCTED] + Tag #5 "[5]"
::= SEQUENCE OF			
--Результат определения контекста представления данных #1			
SEQUENCE {	(5)	30 80	
[0] IMPLICIT Result ::= INTEGER { acceptance(0) }		80 01 00	-- Контекст представления данных #1 ПРИНЯТЬ!
[1] IMPLICIT Transfer-syntax-name OPTIONAL,			
::= OBJECT IDENTIFIER		81 02 51 01	-- ИСО BER
	(5)	00 00	-- конец результата определения контекста представления данных #1
--Результат определения контекста представления данных #2			
SEQUENCE {	(6)	30 80	
[0] IMPLICIT Result ::= INTEGER { acceptance(0) }		80 01 00	-- Контекст представления данных #2 ПРИНЯТЬ!
[1] IMPLICIT Transfer-syntax-name OPTIONAL,			
::= OBJECT IDENTIFIER		81 0c	-- ИИЭР 1073 MDER Big Endian *iso(1) member-body(2) US(840) ieee1073(10004)*
			-- *mdap(2) version1(1) mdap-0(0) standardSpecificExt(0) modules(0) transferSyntax(2)*
			-- *mderBigEndian(1)*
		2A 86 48 CE 14 02 01 00 00 00 02 01	

Рисунок F.2 — Пример форматирования ответа ассоциации (Association response)

```
        }                                                   (6)   00 00          -- конец результата определения контекста представления данных #1
    }                                                       (4)   00 00          -- конец списка результатов определения контекста представления данных
-- (нормальный режим) данные пользователя
User-data ::= CHOICE {[APPLICATION 1] IMPLICIT Fully-encoded-data}  (7)   61 80          -- данные пользователя уровня представления данных
    ::= SEQUENCE OF PDV-list
    ::= SEQUENCE {                                          (8)   30 80
        presentation-context-identifier ::= INTEGER                02 01 01
        CHOICE {single-ASN.1-type[0] ANY                    (9)   a0 80          -- Контекст  представления данных #1 ACSE
AARE-apdu ::=
    [APPLICATION 1] IMPLICIT SEQUENCE {                     (10)  61 80          -- PPDV User-data == AARE (ACSE Response)
        protocol-version [0] IMPLICIT BIT STRING {version-1(0)}
        application-context-name [1] Application-context-name   (11)  a1 80          -- DEFAULT {версия-1}, опущено
            ::= OBJECT IDENTIFIER                               06 0c
                                                                       -- Нормальный режим работы прикладного контекста iso(1) member-body(2) US(840)
                                                                       -- ieee1073(10004) mdap(2) version1(1) mdap-0(0) standardSpecificExt(0) modules(0)
                                                                       -- applicationContext(3) normalMode(1)
                                                                2A 86 48 CE 14 02 01 00 00 00 03 01
                                                            (11)  00 00
result [2] Associate-result ::= INTEGER {accepted(0)            a2 03 02 01 00  -- context-specific+CONSTRUCTED+ Tag #2, INTEGER TLV
    result-source-diagnostic [3] Associate-source-diagnostic ::= CHOICE {
                                                                a3 05          -- context-specific+CONSTRUCTED+Tag #3, len=5
    acse-service-user [1] INTEGER {null(0)}                     a1 03 02 01 00  -- context-specific+CONSTRUCTED+
    user-information [30] IMPLICIT Association-information      (12) be 80     -- context-specific + CONSTRUCTED + Tag #30 "[30]"
     ::= SEQUENCE OF EXTERNAL
        EXTERNAL                                                (13) 28 80     -- CONSTRUCTED + Tag #8 "EXTERNAL"
          ::=IMPLICIT SEQUENCE {
            direct-reference ::= OBJECT IDENTIFIER                             -- OPTIONAL, опущено
            indirect-reference ::= INTEGER                      02 01 02       -- Идентификатор  контекста представления данных† MDER обратный порядок байтов
            encoding ::= CHOICE {[1] IMPLICIT OCTET STRING
                                                                81 3A
                                                                               -- Примечание – Контекст представления данных и его формат были
                                                                               -- «приняты», за этим следует «MDDL/MDER»
            MDSEUserInfo ::= SEQUENCE {
                protocolVersion [0] IMPLICIT ProtocolVersion
                 ::= BIT STRING {mddl-version1 (0)}             80 00 00 00    -- Протокол версии 1
                nomenclatureVersion [1] IMPLICIT NomenclatureVersion
                ::= BIT STRING {nomen-version1(1)}              40 00 00 00    -- Номенклатура версии 1
                functionalUnits [2] IMPLICIT FunctionalUnits
                ::= BIT STRING {extendedObjectSelection(0)}     00 00 00 00    -- Расширенная функция выбора объектов не используется
```

† Данное значение включено в поле идентификатора контекста представления данных всех сообщений языка MDDL.

Рисунок F.2, лист 2

```
        systemType [3] IMPLICIT SystemType
          = BIT STRING { sys-type-agent(8), }          00 80 00 00   -- [Медицинские данные] Агент
        startupMode [4] IMPLICIT StartupMode
        ::= BIT STRING { cold-start(2) }               20 00 00 00   -- Зависит от реализации прибора (для Demo ... фиксирован при холодном запуске)

-- расширения INTEROP
-- optionList AttributeList.                         00 00 00 00   -- ZERO AttributeList
-- supportedAProfiles AttributeList
        ::= SEQUENCE OF AVA-Type                       00 01 00 1E   -- Идентифицирован один профиль, длина = 30
                                                       00 02 00 1A   -- OID-Type == NOM_BASELINE_PROFILE_SUPPORT (Раздел инфраструктуры)
                                                       80 00 00 00   -- BaselineRevision 'baseline-rev-0(0)
                                                       00 00 00 00   -- max-mtu-rx - макс. размер получаемого сообщения в байтах
                                                       00 00 00 00   -- max-mtu-tx - макс. размер передаваемого сообщения в байтах
                                                       FF FF FF FF   -- max-bw-tx - макс. передаваемый диапазон частот (0xFFFFFFFF= не указано)
                                                       00 01         -- max-mds-hierarchy - макс. глубина иерархии объекта MDS, – из корня MDIB в самый глубокий MDS
                                                       00 00 00 00   -- BaselineOptions - Отсутствие динамического создания/удаления объектов
                                                                     -- Будущее – Будет добавлен Patient Demographics Opt Pkg
                                                       00 00 00 00   -- optional-packages - ZERO AttributeList
                                                                     -- Примечание – Позднее необходимо добавить Patient Demographics package
(13)  00 00   (12)  00 00
(10)  00 00   (9)   00 00
(8)   00 00   (3)   00 00
(2)   00 00   (1)   00 00
```

Рисунок F.2, лист 3

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ

```
Структура                                                Шестнадцатеричные коды
-- Session Finish PDU
Session Header (FINISH SPDU)                                 09 18     -- SI='09'=9.(FN), LI = длина в октетах='18'==24.
                                                             c1 16     -- PGI='c1'==193. (данные пользователя), LI= длина данных пользователя ='16' == 22.
-- Presentation PDU
User-data . = CHOICE [APPLICATION 0] IMPLICIT Fully-encoded-data  (1)  61 80
    SEQUENCE OF PDV-list                                     (2)  30 80
        Presentation-context-identifier                           02 01 01
        presentation-data-values .= CHOICE single-ASN.1-type [0] ANY  (3)  A0 80
        ACSE-apdu .= CHOICE RLRQ-apdu = [APPLICATION 2 ] IMPLSEQ  (4)  62 80
        Reason [0] IMPLICIT Release-request-reason normal [0]     80 01 00
                                                             (4)  00 00
                                                             (3)  00 00
                                                             (2)  00 00
                                                             (1)  00 00
```

Рисунок F.3 — Пример форматирования запроса на разъединение ассоциации (Association release request)

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ

Структура	Шестнадцатеричные коды	
-- Session Disconnect PDU		
Session Header (DISCONNECT SPDU)	0a 18	-- SI='0a'==10.(DISCONNECT), LI = длина в октетах ='18'==24
	c1 16	-- PGI='c1'==193 (данные пользователя), LI= длина данных пользователя ='16' == 22
-- Presentation PDU		
User-data ::= CHOICE [APPLICATION 0] IMPLICIT Fully-encoded-data	(1) 61 80	
SEQUENCE OF PDV-list	(2) 30 80	
Presentation-context-identifier	02 01 01	
presentation-data-values ::= CHOICE single-ASN 1-type [0] ANY	(3) A0 80	
ACSE-apdu = CHOICE RLRE-apdu ::= [APPLICATION 3] IMPL SEQ(4)	63 80	
Reason [0] IMPLICIT Release-request-reason normal [0]	80 01 00	
	(4) 00 00	
	(3) 00 00	
	(2) 00 00	
	(1) 00 00	

Рисунок F.4 — Пример форматирования ответа на разъединение ассоциации (Association release response)

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ

Структура	Шестнадцатеричные коды	
-- Session Abort PDU		
Session Header (ABORT SPDU)	19 03	-- SI='19'==25.(DISCONNECT), LI = длина в октетах='03'==3.
Transport Disconnect	11 01 09	-- бит 4: нет причины, бит 1: транспортное соединение сброшено

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ

Структура		Шестнадцатеричные коды	
-- Session Abort PDU			
Session Header (ABORT SPDU)		19 2E	-- SI='19'==25 (DISCONNECT), LI = длина в октетах= 2E'==46
Transport Disconnect		11 01 03	-- бит 2. отмена пользователем, бит 1. транспортное соединение сброшено
		c1 29	-- PGI='c1'==193 (данные пользователя), LI= длина данных пользователя ='29' == 41
Abort-type ::= CHOICE {			
ARU-PPDU ::= CHOICE [0] IMPLICIT SEQUENCE	(1)	A0 80	
[0] IMPLICIT Presentation-context-identifier-list	(2)	A0 80	
Presentation-context-identifier-list ::= SEQUENCE OF SEQUENCE	(3)	30 80	-- ACSE PCID
Presentation-context-identifier		02 01 01	-- ACSE Transfer Syntax OID
Transfer-syntax-name		06 02 5101	
	(3)	00 00	
	(2)	00 00	
User-data = CHOICE [APPLICATION 0] IMPLICIT Fully-encoded-data	(4)	61 80	
SEQUENCE OF PDV-list	(5)	30 80	
Presentation- context-identifier		02 01 01	-- ACSE PCID
presentation-data-values ::= CHOICE single-ASN 1-type [0] ANY	(6)	A0 80	
ACSE-apdu ::= CHOICE ABRT-apdu ::= [APPLICATION 4] IMPL SEQ	(7)	64 80	
abort-source [0] IMPLICIT ABRT-source acse-service-provider	(1)	80 01 01	
	(7)	00 00	
	(6)	00 00	
	(5)	00 00	
	(4)	00 00	
	(1)	00 00	

Рисунок F.5 — Пример форматирования прекращения ассоциации (Association abort)

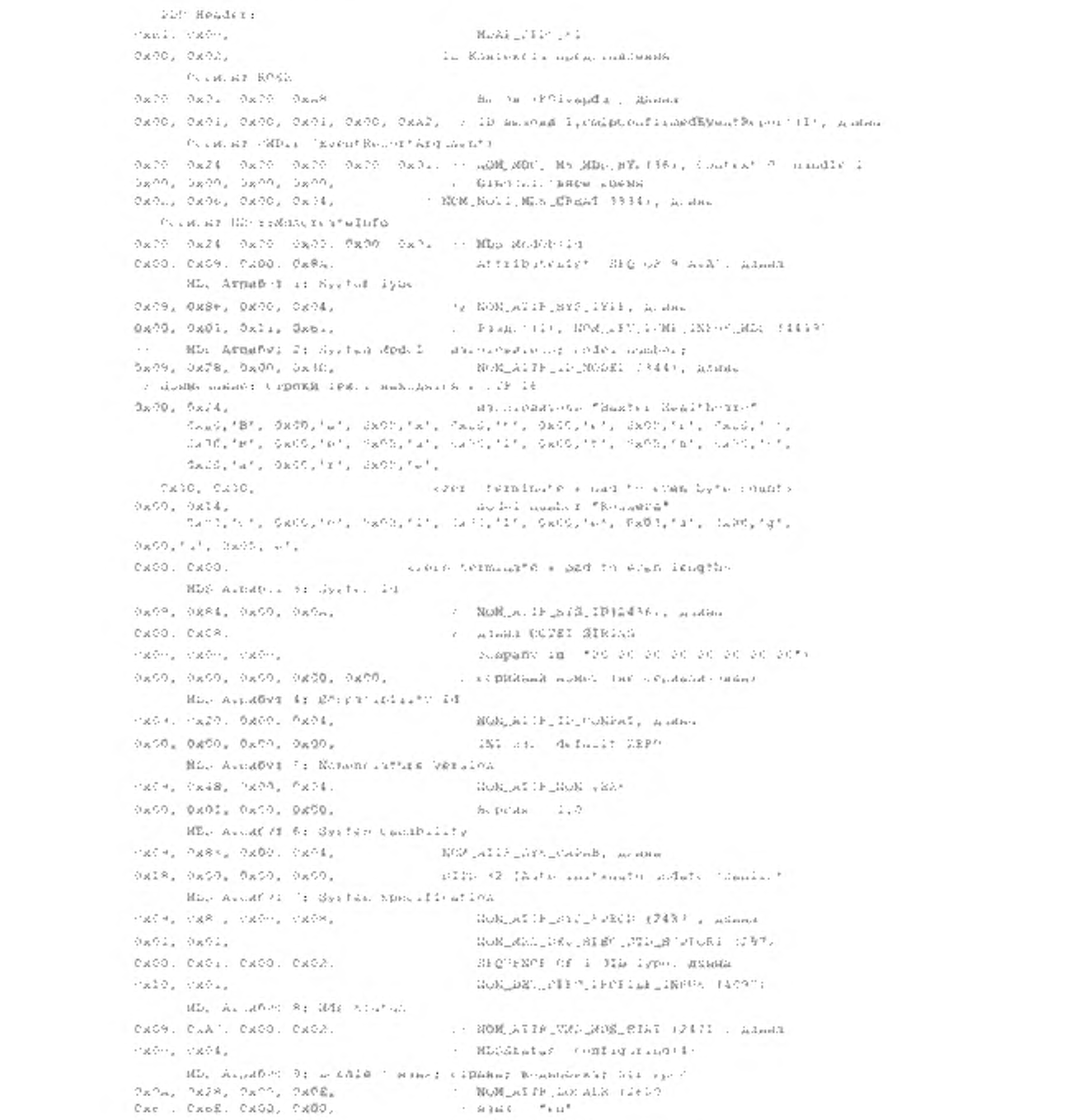


Рисунок F.6 – Пример форматирования отчета о событии уведомления о создании объекта системы MDS (MDS object create notification event report)

```
// Заголовок ID :
0xE1, 0x00,                          // MDAP_SPDU_SI
0x00, 0x02,                            ID Контекста представления
// -- Сегмент ROSE
0x00, 0x07, 0x00, 0x14,                Результат (ROSApdu), длина
0x00, 0x01, 0x00, 0x01, 0x00, 0x0E,// ID вызова=1,cmipConfirmedEventReport(1),длина
      Сегмент CMDIP (EventReportResult):
0x00, 0x24, 0x00, 0x00, 0x00, 0x01,// NOM_MOC_VMS_MDS_HYD(36), Контекст=0, Handle=1
 0x00, 0x00, 0x00, 0x00,                Относительное время
0x0D, 0x06, 0x00, 0x00              // NOM_NOTI_MDS_CREAT(3334), result info len=ZERO
```

Рисунок F.7 – Пример форматирования ответа на отчет о событии уведомления о создании объекта системы MDS (MDS object create notification event report response)

Структура	Номинальное значение	Шестнадцатеричное кодирование
Session		
Id :	MDAP_SPDU_SI	E1 00
Presentation		
Pres. context id:	0x0001	00 01
Application		
ROSEapdu		
choice :	ROIVapdu	00 01
length :	64	00 40
ROIVapdu		
invoke identifier :		xx xx
operation value :	cmipEventReport	00 00
argument length :	58	00 3A
CMDIP EventReportArg.		
managed object class:	NOM_MOC_SCAN_CFG_PERI	00 1B
man. obj. context id :		xx xx
managed obj. handle :		xx xx
event time :		xx xx xx xx
event type (Any defined by)		
nom oid :	NOM_NOTI_MDS_SCAN_RT	09 03
event info length :	44	00 2C
ScanReportInfo		
scan-report-no	1	00 01
glb-scan-info.count		00 01
SingleCtxtScan		
context-id		00 00
scan-info.cnt		00 02
ObservationScan		
obj-handle		xx xx
attributes.cnt		00 01
AVA-Type		
attribute-id		09 79
attribute-val-len		00 0A
.data		
physio-id(...INF_RATE)		22 03
state (Test data)		00 20
units		xx xx
value (FLOAT-Type		xx xx xx xx
ObservationScan		
obj-handle		xx xx
attributes.cnt		00 01
AVA-Type		
attribute-id		09 79
attribute-val-len		00 0A
.data		
physio-id (...INF_VOL)		22 04
state (Test data)		00 20
units		xx xx
value (FLOAT-Type)		xx xx xx xx

Рисунок F.8 – Пример форматирования отчета о конфигурируемом периодическом сканере событий (Configured periodic scanner event report)

```
// Заголовок PDU:
0xE1, 0x0C,                         // MDAP_SPDU_SI
0x00, 0x02,                         // ID Контекста представления
// -- Сегмент ROSE
0x00, 0x01, 0x00, 0xAC,             // Вызов (ROIVapdu), длина
0x00, 0x06, 0x00, 0x00, 0x00, 0xA6, // ID вызова-6, cmipEventReport(0),длина
// -- Сегмент CMDIP (EventReportArgument)
0x00, 0x13, 0x00, 0x00, 0x00, 0x0C, // NOM_MOC_SCAN_CFG_PERI(19)-Handle-12
0x00, 0x0C, 0x00, 0x0C,             // Метка относительного времени для Отчета
0x0D, 0x03, 0x00, 0x98,             // NOM_NOTI_BUT_SCAN_RPT(3331), длина
// Сегмент Periodic Scanner::ScanReportInfo
0x00, 0x01,                         // scan-report-no - 1
0x00, 0x01, 0x00, 0x92,               // glb-scan-info - SEQ OF 1 SingleCtxtScan
0x0C, 0x0d,                         // context-id - 0x0CC0
0x00, 0x07, 0x0C, 0x8C,             // scan-info - SEQ OF 7 Observation Scan
// Object Observation Scan 1: Primary VTBI (Metric Observed Value Group)
0x00, 0x7C,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
// -- Атрибут 1: Nu-Observed-Value
0x09, 0x5C, 0x00, 0x0A,             // NOM_ATTR_NU_VAL_OBS(2384), длина
0x68, 0xBd, 0x08, 0x0C,             // NOM_VOL_FLUID_TBI_REMAIN(26800),test-data(4)
0x00, 0x52,                         // NOM_DIM_MILLI_L(1618)
0x0C, 0x0C, 0x0C, 0x0C,             // значение-FLOAT-Type
// Object Observation Scan 2: Primary Duration (Metric Observed Value Group)
0x00, 0x71,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
// -- Атрибут 1: Nu-Observed-Value
0x09, 0x5C, 0x00, 0x0A,             // NOM_ATTR_NU_VAL_OBS(2384), длина
0x68, 0xDC, 0x08, 0x0C,             // NOM_TIME_PD_REMAIN(26844),test-data(4)
0x08, 0xAC,                         // NOM_DIM_MIN(2220)
0x0C, 0x0C, 0x0C, 0x0C,             // значение-FLOAT-Type
// Object Observation Scan 3: Primary VI (Metric Observed Value Group)
0x00, 0x73,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
// -- Атрибут 1: Nu-Observed-Value
0x09, 0x50, 0x00, 0x0A,             // NOM_ATTR_NU_VAL_OBS(2384), длина
0x68, 0xA8, 0x08, 0x00,             // NOM_VOL_FLUID_DELIV(26792),test-data(4)
0x00, 0x52,                         // NOM_DIM_MILLI_L(1618)
0x00, 0x00, 0x00, 0x00,             // значение-FLOAT-Type
// Object Observation Scan 4: Secondary VTBI (Metric Observed Value Group)
0x00, 0x84,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
// -- Атрибут 1: Nu-Observed-Value
0x09, 0x5C, 0x00, 0x0A,             // NOM_ATTR_NU_VAL_OBS(2384), length
0x68, 0xB0, 0x08, 0x00,             // NOM_VOL_FLUID_TBI_REMAIN(26800),test-data(4)
0x00, 0x52,                         // NOM_DIM_MILLI_L(1618)
0x00, 0x00, 0x00, 0x00,             // значение-FLOAT-Type
// Object Observation Scan 5: Secondary Duration (Metric Observed Value Group)
0x00, 0x85,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
// -- Атрибут 1: Nu-Observed-Value
0x09, 0x50, 0x00, 0x0A,             // NOM_ATTR_NU_VAL_OBS(2384), длина
0x68, 0xDC, 0x08, 0x00,             // NOM_TIME_PD_REMAIN(26844),test-data(4)
0x08, 0xA0,                         // NOM_DIM_MIN(2220)
0x00, 0x00, 0x00, 0x00,             // значение-FLOAT-Type

// Object Observation Scan 6: Secondary VI (Metric Observed Value Group)
0x00, 0x87,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
```

```
// -- Атрибут 1: Nu-Observed-Value
0x09, 0x50, 0x00, 0x0A,              // NOM_ATTR_NU_VAL_OBS(2384), длина
0x66, 0xA8, 0x06, 0x00,              // NOM_VOL_FLUID_DELIV(26792),test-data(4)
0x06, 0x52,                          // NOM_DIM_MILLI_L(1618)
0x00, 0x00, 0x00, 0x00               // значение-FLOAT-Type
// Object Observation Scan /: Total VI (Metric Observed Value Group)
0x00, 0x98,0x00, 0x01, 0x00, 0x0E,  // Handle + AttributeList (SEQ OF 1 AVA)
// -- Атрибут 1: Nu-Observed Value
0x09, 0x50, 0x00, 0x0A,             // NOM_ATTR_NU_VAL_OBS(2384), длина
0x66, 0xFC, 0x06, 0x00,             // NOM_VOL_INFUS_ACTUAL_TOTAL(26876),test-data(4)
0x06, 0x52,                         // NOM_DIM_MILLI_L(1618)
0x00, 0x00, 0x00, 0x00              // значение-FLOAT-Type
```

Рисунок F.9 – Пример форматирования отчета о периодическом сканере, буферизирующего просматриваемые события (Periodic scanner buffered scan event report)

Приложение G
(справочное)

Специализация ASN.1

G.1 Введение

ASN.1 – это стандартная нотация, которая используется для определения типов данных, значений и ограничений на значения. Данная нотация широко используется в стандартах OSI. Нотация также является ключевым компонентом DIM и семейства ИСО/ИИЭР 11073-10300 стандартов по специализации приборов.

Правила MDER, описанные в приложении А, определяют методы для преобразования синтаксиса ASN.1 в поток байтов, пригодный для коммуникаций. Следует отметить, что правила MDER функционируют только на подмножестве языка ASN.1.

Настоящее приложение описывает специализацию языка ASN.1 для кодирования с помощью правил MDER. Все компоненты блоков данных PDU языка ASN.1, предназначенные для кодирования с помощью правил MDER являются предметом данной специализации.

G.2 Специализация ASN.1

Для каждого типа данных языка ASN.1 данная специализация сопровождается символом «I» для включения с ограничениями, «R» для ограничений по использованию или «E» для исключения.

Специализация типов данных языка ASN.1 дана в таблице G.1. См. приложение А для связей с кодированием MDER.

Таблица G.1 — Специализация типов данных языка ASN.1

Тип ASN.1	Статус	Комментарии
BOOLEAN	E	—
INTEGER	E	См. таблицу G.2 для получения списка поддерживаемых альтернативных типов
ENUMERATED	E	В таблице G.2 использовать NamedNumberedList с типами INTEGER
REAL	E	Использовать FLOAT
BITSTRING	E	См. таблицу G.2 для получения списка поддерживаемых альтернативных типов
OCTETSTRING	I	—
NULL	R	Нулевой базисный элемент обычно исключают из MDER, но включают с ограничениями в примитивы CHOICE и ANY DEFINED BY в MDER
SEQUENCE	R	Можно не использовать ни ключевые слова OPTIONAL, DEFAULT или COMPONENTS OF, ни автоматическое тегирование
SEQUENCE OF	I	—
SET E	E	Использовать тип SEQUENCE
SET OF	E	Использовать тип SEQUENCE OF
CHOICE	R	Альтернативы должны сопровождаться тегами. Автоматическое тегирование не поддерживается
SELECTION	E	—
TAGGED	R	Только для использования в качестве альтернативы в типе CHOICE
OBJECT IDENTIFIER	E	—
EMBEDDED PDV	E	—
EXTERNAL	E	—
CHARACTER STRING	E	—
ANY DEFINED BY	R	ANY DEFINED BY должен определять компонент, содержащего его SEQUENCE. Этот компонент должен быть OBJECT IDENTIFIER (номенклатура). OBJECT IDENTIFIER может быть контекстно-независимым или контекстно-зависимым

Таблица G.2 — Поддерживаемые типы integer, bitstring и float

Типы	Нотация	Определение	Описание
INTEGER (целочисленная переменная)	INT-U8	INTEGER (0..255)	8-битное целое число без знака
	INT-I8	INTEGER (–128..127)	8-битное целое число со знаком
	INT-U16	INTEGER (0..65535)	16-битное целое число без знака
	INT-I16	INTEGER (–32768..32767)	16-битное целое число со знаком
	INT-U32	INTEGER (0..4294967295)	32-битное целое число без знака
	INT-I32	INTEGER (–2147483648 ..2147483647)	32-битное целое число со знаком
BITSTRING (битовая строка)	BITS-16	BIT STRING (SIZE(16))	16-битная битная строка
	BITS-32	BIT STRING (SIZE(32))	32-битная битная строка
FLOAT (данные с плавающей запятой)	FLOAT-Type	REAL (WITH COMPONENTS {мантисса (–8388605..8388605), основание (10), экспонента (–128..127)})	32-битное десятичное с плавающей точкой

Ниже кратко описаны элементы таблицы G.1:

- BOOLEAN (логическое значение): тип BOOLEAN не является частью специализации языка ASN.1;
- INTEGER (целочисленная переменная): тип ASN.1 INTEGER не является частью специализации ASN.1. Вместо этого в MDER предлагается кодирование для типов INTEGER, представленное в таблице G.2. Эти типы следуют синтаксису, показанному в таблице;
- ENUMERATED (перечисление): тип ASN.1 ENUMERATED не является частью специализации языка ASN.1, в MDER предлагается кодирование для типов INTEGER, представленное в таблице G.2. Эти типы INTEGER можно использовать с NamedNumberList, который является аналогичным типу ENUMERATED;
- REAL (реальный): тип ASN.1 REAL не является частью специализации языка ASN.1. Вместо этого, в MDER предлагается кодирование для FLOAT-Type, которое является 32-битным типом с плавающей точкой. Данный тип есть в таблице G.2;
- BITSTRING (битовая строка): тип BIT STRING не является частью специализации языка ASN.1. Вместо этого, в MDER предлагается кодирование для типов BIT STRING, которое является 32-битным типом с плавающей запятой. Данный тип представлен в таблице G.2;
- OCTETSTRING (строка октет): тип OCTET STRING является частью специализации языка ASN.1. Нет ограничений по его использованию;
- NULL (ноль): нулевой базисный элемент обычно исключают из правил MDER, но включают с ограничениями в примитивы CHOICE и ANY DEFINED BY в MDER;
- SEQUENCE (последовательность): тип SEQUENCE является частью специализации языка ASN.1 с определенными ограничениями. Компонент SEQUENCE может не определяться вместе с ключевыми словами OPTIONAL, DEFAULT или COMPONENTS OF. Автоматическое тегирование не поддерживается;
- SEQUENCE OF (последовательность ч-л.): тип SEQUENCE OF является частью специализации языка ASN.1. Нет ограничений по его использованию;
- SET (набор инструкций): тип SET не является частью специализации языка ASN.1. Тип SEQUENCE является его рекомендуемой заменой;
- SET OF (набор инструкций ч-л.): тип SET OF не является частью специализации языка ASN.1. Тип SEQUENCE OF является его рекомендуемой заменой;
- CHOICE (выбор): тип CHOICE является частью специализации языка ASN.1 с определенными ограничениями. Каждая альтернатива в CHOICE должна быть типом TAGGED. Автоматически теги не поддерживаются;
- SELECTION (выбор): оператор типа SELECTION не является частью специализации языка ASN.1;
- TAGGED (тегированный): в общем, типы TAGGED не являются частью специализации языка ASN.1. Однако каждый альтернативный тип CHOICE должен быть типом TAGGED;
- OBJECT IDENTIFIER (идентификатор объекта): тип OBJECT IDENTIFIER не является частью специализации языка ASN.1;
- EMBEDDED PDV (встроенное PDV): тип EMBEDDED PDV не является частью специализации языка ASN.1;
- EXTERNAL (внешний объект): тип EXTERNAL не является частью специализации языка ASN.1;
- CHARACTER STRING (строка символов): типы CHARACTER STRING не являются частью специализации языка ASN.1. Вместо этого рекомендуется использовать тип OCTETSTRING;

- ANY DEFINED BY (любое значение, определенное с помощью): тип ANY DEFINED BY является частью специализации языка ASN.1. ANY DEFINED BY должен определять компонент, содержащей его SEQUENCE. Этот компонент должен быть OBJECT IDENTIFIER (номенклатура). OBJECT IDENTIFIER может быть контекстно-независимым или контекстно-зависимым. В приложении H дано больше информации об использовании ANY DEFINED BY.

Приложение Н
(справочное)

Вопросы совместимости

Настоящее приложение представляет информацию для логического обоснования вопросов и решений совместимости.

Н.1 Тип ANY DEFINED BY

Изменения в языке ИСО ASN.1 и связанных с ним правилах кодирования (например, правила BER), произошедшие между версиями 1988/90 и 1994, привели к определенным изменениям в синтаксисе определения ANY DEFINED BY (версия 1988/90). Указанные ниже пункты являются выдержками из соответствующих документов ИСО, затрагивающих эти изменения. См. основной текст стандарта для получения информации по нормативным спецификациям, связанным с влиянием на MDAP.

Н.1.1 Переход к текущей нотации языка ASN.1

Ниже дана выдержка из приложения А стандарта ИСО/МЭК 8824-1:1994:

«А.3 Переход к текущей нотации ASN.1

При модификации модуля (изначально написанного согласно нотации ASN.1-88/90) с целью приведения его в соответствие с текущей нотацией необходимо учесть следующее:

...

b) Все применения ANY и ANY DEFINED BY должны поддерживаться соответствующим определением класса информационного объекта (information object class definition), в котором ANY и ANY DEFINED BY (и компонент, на который делается ссылка) заменяются на соответствующие ссылки на поля этого класса объектов. В большинстве случаев спецификацию можно в значительной степени улучшить, если уделять должное внимание включению таблиц и ограничений на отношения компонента. Во многих случаях спецификацию можно еще улучшить, если таблицы или ограничение на отношения компонента выполнены в виде параметра данного типа.»

Н.1.2 Класс информационного объекта TYPE-IDENTIFER в языке ASN.1

Ниже дана выдержка из приложения А ИСО/МЭК 8824-2:1994:

«А.1 Настоящее приложение описывает класс объектов полезной информации со ссылочным описанием класса TYPE-IDENTIFIER (Идентификатор типа).

Примечание — Данный класс информационного объекта является простейшим полезным классом, который имеет всего лишь два поля, поле идентификатора типа OBJECT IDENTIFIER (идентификатор объекта), и поле типа, которое определяет тип ASN.1 для переноса всей информации, касающейся любого определенного объекта в данном классе. Все это описано в настоящей рекомендации или международном стандарте по причине широкого использования информационных объектов данной формы.

А.2 Класс информационного объекта TYPE-IDENTIFIER определяется следующим образом:

```
TYPE-IDENTIFIER ::= CLASS
{
   &id OBJECT IDENTIFIER UNIQUE,
   &Type
}
WITH SYNTAX {&Type IDENTIFIED BY &id}
```

А.3 Данный класс определяется как класс «полезного» информационного объекта и присутствует в каждом модуле, без необходимости его импортировать.

А.4 Пример

Тело коммуникации MHS можно определить следующим образом:

```
MHS-BODY-CLASS ::= TYPE-IDENTIFIER

g4FaxBody MHS-BODY-CLASS ::=
   {BIT STRING IDENTIFIED BY {mhsbody 3}}
```

Проектировщик протокола, как правило, определяет компонент для переноса MHS-BODY-CLASS путем указания типа «INSTANCE OF MHS-BODY-CLASS», определенного в С.9.»

Н.1.3 Кодирование типа instance-of в BER

Ниже дана выдержка из стандарта ИСО/МЭК 8825-1:1994:

«8.16 Кодирование значения instance-of

8.16.1 Кодирование значения instance-of должно быть кодированием BER следующего типа последовательности со значением, указанным в 8.16.2:

```
[UNIVERSAL 8] IMPLICIT SEQUENCE
{
```

```
   type-id .DefinedObjectClass.&id,
   value [0] EXPLICIT .DefinedObjectClass>.&Type
},
```

где «<DefinedObjectClass>» заменен определенным «DefinedObjectClass», используемым в нотации «InstanceOfType».

Примечание — Когда значение является значением одного типа ASN.1, и для этого значения применяется кодирование BER, то кодирование данного типа идентично кодированию соответствующего значения внешнего типа, где для представления абстрактного значения используется альтернатива «синтаксиса».

8.16.2 Значение компонентов типа последовательности в 8.16.1 должно быть таким же, как и значения соответствующих компонентов ассоциированного типа в МСЭ-Т Рек. X.681 или ИСО/МЭК 8824-2, С.7.»

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
ИИЭР 1073	–	*
ИСО/МЭК 8327-1	IDT	ГОСТ Р ИСО/МЭК 8327-1—96. «Информационная технология. Взаимосвязь открытых систем. Протокол сеансового уровня в режиме с установлением соединения. Ч. 1. Спецификация протокола»
ИСО/МЭК 8650-1	–	*
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 « Информационные технологии. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 2. Спецификация информационного объекта»
ИСО/МЭК 8825-1	IDT	ГОСТ Р ИСО/МЭК 8825-1—2003 «Информационные технологии. Правила кодирования языка ASN.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования»
ИСО/МЭК 9072-2	IDT	ГОСТ Р ИСО/МЭК 9072-2—93 «Системы обработки информации. Передача текста. Удаленные операции. Часть 2. Спецификация протокола»
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 9899	–	*
ИСО/МЭК 11188-3	–	*
ИСО/ИИЭР 11073-10101	–	*
ИСО/ИИЭР 11073-10201	–	*
ИСО/ИИЭР 11073-30200	–	*
ИСО/ИИЭР 11073-30300	IDT	ГОСТ Р 54481-2011/ISO/IEEE 11073-30300:2004 «Информатизация здоровья. Взаимодействие медицинских приборов на месте лечения. Часть 30300. Транспортный профиль. Инфракрасный канал связи»

* 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.

Библиография

[1] Black, Uyless, Network Management Standards — SNMP, CMIP, TMN, MIBs, and Object Libraries, New York: McGraw-Hill, 1994

[2] Dubuisson, Oliver, ASN.1 — Communication Between Heterogeneous Systems, Morgan Kaufmann, 2000

[3] IEEE 100, The Authoritative Dictionary of IEEE Standards Terms and Definitions, Seventh Edition.16

[4] IEEE Std 1003.1g™, Information Technology — Portable Operating System Interface (POSIX®) — Protocol Independent Interfaces (PII)

[5] IEEE Std 1073.3.1, IEEE Standard for Medical Device Communications — Transport Profile — Connection Mode

[6] IETF RFC 2030, Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI.17

[7] ISO/IEEE P11073-20102, Health informatics — Point-of-care medical device communication — Part 20101: Application profiles — MIB elements.18

[8] ISO/IEEE P11073-20201, Health informatics — Point-of-care medical device communication — Part 20201: Application profile — Polling mode

[9] ISO/IEEE P11073-20202, Health informatics — Point-of-care medical device communication — Part 20202: Application profile — Baseline

[10] ITU-T Recommendation X.225, Information Technology — Open Systems Interconnection — Connection-oriented session protocol: Protocol Specification.19 (same as ISO/IEC 8327-1)

[11] ITU-T Recommendation X.226, Information Technology — Open Systems Interconnection — Connection-oriented presentation protocol: Protocol Specification

[12] ITU-T Recommendation X.227, Information Technology — Open Systems Interconnection—Connection-oriented protocol for the association control service element (same as ISO/IEC 8650-1)

[13] ITU-T Recommendation X.680, Information Technology — Abstract Syntax Notation One (ASN.1) — Specification of Basic Notation (same as ISO/IEC 8824-1)

[14] ITU-T Recommendation X.690, Information Technology — ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) (same as ISO/IEC 8825-1)

[15] Piscitello, David M., and Chapin, A. Lyman, Open Systems Networking: TCP/IP and OSI, Pearson Addison Wesley, 1993

[16] Stallings, William, SNMP, SNMPv2, and CMIP — The Practical Guide to Network-Management Standards, Pearson Addison Wesley, 1993

[17] Stevens, W. Richard, UNIX Network Programming, Prentice Hall, 1990

УДК 004:61:006.354 ОКС 35.240.80

Ключевые слова: здравоохранение, информатизация здоровья, структуры данных, персональные медицинские приборы, передача данных, информационное взаимодействие, прикладные профили, базовый стандарт, коммуникационная модель, информационная модель

Редактор *А.Ф. Колчин*
Технический редактор *В.Ю. Фотиева*
Корректор *Ю.М. Прокофьева*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 29.04.2016. Подписано в печать 11.05.2016. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 8,84. Уч.-изд. л. 8,20. Тираж 33 экз. Зак. 1254.

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123995 Москва. Гранатный пер., 4.
www.gostinfo.ru gost.e-ecolog.ru

gost.e-ecolog.ru