ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56845— 2019/ ISO/IEEE 11073-20601: 2016

ИНФОРМАТИЗАЦИЯ ЗДОРОВЬЯ

Обмен данными с персональными медицинскими приборами

Часть 20601

Прикладной профиль. Оптимизированный протокол обмена

(ISO/IEEE 11073-20601:2016, IDT)

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным унитарным предприятием «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ») на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 468 «Информатизация здоровья»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 августа 2019 г. № 578-ст

4 Настоящий стандарт идентичен международному стандарту ISO/IEEE 11073-20601:2016 «Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 20601. Прикладной профиль. Оптимизированный протокол обмена» (ISO/IEEE 11073-20601:2016 «Health informatics — Personal health device communication — Part 20601: Application profile — Optimized exchange protocol», IDT)

5 Настоящий стандарт рекомендуется применять совместно с ГОСТ Р 56845—2019/ISO/IEEE 11073-20601:2016/Cor.1:2016

6 ВЗАМЕН ГОСТ Р 56845—2015/ ISO/IEEE 11073-20601:2010

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». 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

© ISO, 2016 — Все права сохраняются
© Стандартинформ, оформление, 2019

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 Общие положения . . . 1
1.1 Область применения . . . 1
1.2 Цель . . . 1
1.3 Контекст . . . 2
2 Нормативные ссылки . . . 5
3 Термины, определения и сокращения . . . 5
3.1 Термины и определения . . . 5
3.2 Сокращения . . . 6
4 Руководящие принципы . . . 7
5 Начальные сведения о персональных медицинских приборах (ИИЭР 11073) . . . 8
5.1 Общие положения . . . 8
5.2 Информационная модель предметной области (DIM) . . . 9
5.3 Сервисная модель . . . 9
5.4 Коммуникационная модель . . . 9
5.5 Соответствие другим стандартам . . . 9
5.6 Безопасность . . . 9
6 Модель предметной области персонального медицинского прибора . . . 10
6.1 Общие положения . . . 10
6.2 Использование номенклатуры . . . 11
6.3 Определения классов персональных медицинских объектов . . . 12
6.3.1 Общие положения . . . 12
6.3.2 Класс MDS . . . 14
6.3.3 Класс Metric . . . 20
6.3.4 Класс Numeric . . . 25
6.3.5 Класс RT-SA . . . 27
6.3.6 Класс Enumeration . . . 28
6.3.7 Класс PM-store . . . 30
6.3.8 Класс PM-segment . . . 34
6.3.9 Классы Scanner . . . 38
6.4 Правила расширения информационной модели . . . 46
7 Сервисная модель персонального медицинского прибора . . . 46
7.1 Общие положения . . . 46
7.2 Служба ассоциации . . . 47
7.3 Службы доступа к объектам . . . 47
7.4 Специфичное применение служб доступа к объектам EVENT REPORT для персональных медицинских приборов . . . 48
7.4.1 Общие положения . . . 48
7.4.2 Подтверждаемые и неподтверждаемые отчеты о событиях . . . 48
7.4.3 Отчет о событиях конфигурации . . . 48
7.4.4 Передача результатов измерений, инициируемая агентом и менеджером . . . 51
7.4.5 Переменный, фиксированный и групповой форматы отчетов о событиях . . . 51
7.4.6 Отчеты о событиях с одним и несколькими лицами . . . 53
7.4.7 Временно хранящиеся результаты измерений . . . 53
8 Коммуникационная модель . . . 54
8.1 Общие положения . . . 54
8.2 Контекст системы . . . 54
8.3 Коммуникационные характеристики . . . 55
8.3.1 Общие положения . . . 55
8.3.2 Общие коммуникационные характеристики . . . 56
8.3.3 Характеристики надежных коммуникаций . . . 56
8.3.4 Характеристики наилучших коммуникаций . . . 57

8.4 Конечные автоматы . . . 57
8.4.1 Конечный автомат агента . . . 57
8.4.2 Конечный автомат менеджера . . . 61
8.4.3 Переменные тайм-аута . . . 62
8.5 Процедура соединения . . . 62
8.5.1 Общие сведения . . . 62
8.5.2 Условия входа . . . 62
8.5.3 Нормальные процедуры . . . 63
8.5.4 Условия выхода . . . 63
8.5.5 Условия ошибки . . . 63
8.6 Процедура в состоянии «Не ассоциирован» . . . 63
8.6.1 Общие положения . . . 63
8.6.2 Условия входа . . . 63
8.6.3 Нормальные процедуры . . . 63
8.6.4 Условия выхода . . . 63
8.6.5 Условия ошибки . . . 63
8.7 Процедура ассоциирования . . . 63
8.7.1 Общие положения . . . 63
8.7.2 Условия входа . . . 63
8.7.3 Нормальные процедуры . . . 63
8.7.4 Условия выхода . . . 67
8.7.5 Условия ошибки . . . 67
8.7.6 Тестовая ассоциация . . . 67
8.8 Процедура конфигурации . . . 68
8.8.1 Общие положения . . . 68
8.8.2 Условия входа . . . 68
8.8.3 Нормальные процедуры . . . 68
8.8.4 Условия выхода . . . 71
8.8.5 Условия ошибки . . . 71
8.9 Процедура выполнения . . . 72
8.9.1 Общие положения . . . 72
8.9.2 Условия входа . . . 72
8.9.3 Нормальные процедуры . . . 72
8.9.4 Условия выхода . . . 82
8.9.5 Условия ошибки . . . 82
8.10 Процедура завершения ассоциации . . . 84
8.10.1 Общие положения . . . 84
8.10.2 Условия входа . . . 84
8.10.3 Нормальные процедуры . . . 84
8.10.4 Условия выхода . . . 84
8.10.5 Условия ошибки . . . 84
8.11 Кодирование сообщений . . . 84
8.12 Координация времени . . . 85
8.12.1 Общие положения . . . 85
8.12.2 Абсолютное время . . . 85
8.12.3 Базовое время со смещением . . . 86
8.12.4 Относительное время . . . 87
8.12.5 Относительное время высокой точности . . . 87
9 Модель соответствия . . . 88
9.1 Применимость . . . 88
9.2 Спецификация соответствия . . . 88
9.3 Заявления о соответствии реализации (ЗСР) . . . 89

9.4 Общее соответствие . . . 89
9.4.1 Общее ЗСР . . . 89
9.4.2 Минимальные требования ЗСР . . . 90
9.4.3 ЗСР для поддержки служб . . . 91
9.5 Дополнения/расширения ЗСР для устройств . . . 92
9.5.1 Общие дополнения/расширения ЗСР . . . 92
9.5.2 ЗСР объекта и класса (ОК ПМП) информационной модели предметной области персонального медицинского прибора . . . 92
9.5.3 ЗСР атрибутов ОК ПМП . . . 93
9.5.4 ЗСР поведения ОК ПМП . . . 93
9.5.5 ЗСР уведомлений ОК ПМП . . . 94
9.5.6 ЗСР номенклатуры ОК ПМП . . . 94
Приложение A (обязательное) Определения ACH.1 . . . 95
Приложение B (справочное) Пример спецификации масштаба и диапазона . . . 125
Приложение C (справочное) Концепция PM-store . . . 127
Приложение D (справочное) Типы транспортных профилей . . . 131
Приложение E (обязательное) Таблицы состояний . . . 134
Приложение F (обязательное) Правила кодирования медицинских приборов (MDER) . . . 156
Приложение G (справочное) Определения типов кодированных данных . . . 165
Приложение H (справочное) Примеры . . . 183
Приложение I (обязательное) Номенклатурные коды . . . 195
Приложение J (справочное) История происхождений и изменений . . . 200
Приложение K (справочное) Библиография . . . 202

Предисловие к ИСО/ИИЭР 11073-20601

Международная организация по стандартизации (ИСО) является всемирной федерацией национальных организаций по стандартизации (членов ИСО). Работу по подготовке международных стандартов обычно выполняют технические комитеты ИСО. Любой член ИСО, заинтересованный в предмете, по которому создан технический комитет, имеет право на представительство в этом комитете. Международные правительственные и неправительственные организации, имеющие связи с ИСО, также принимают участие в работах. ИСО тесно сотрудничает с Международной электротехнической комиссией (МЭК) по всем вопросам стандартизации в области электротехники.

Стандарты ИИЭР разрабатываются научными обществами ИИЭР и Координационными комитетами по стандартизации, входящими в состав Бюро стандартов Ассоциации по стандартизации ИИЭР (IEEE-SA). ИИЭР разрабатывает свои стандарты на основе принципов всеобщего согласия. Стандарты ИИЭР разрабатываются на основе процесса достижения консенсуса, одобренного Американским национальным институтом стандартов (American National Standards Institute, ANSI), который для получения окончательного документа сводит вместе добровольных участников, представляющих разные точки зрения и интересы. Добровольные участники не обязаны быть членами ИИЭР и работают на безвозмездной основе. Хотя ИИЭР управляет этим процессом и устанавливает правила по обеспечению беспристрастности в процессе достижения консенсуса, тем не менее ИИЭР не производит независимую оценку, тестирование или проверку точности какой-либо информации, содержащейся в его стандарте.

Основной задачей технических комитетов является подготовка международных стандартов. Проекты международных стандартов, одобренные техническими комитетами, рассылаются членам ИСО на голосование. Публикация в качестве международного стандарта требует одобрения по меньшей мере 75 % членов ИСО, участвовавших в голосовании.

Необходимо отметить возможность того, что какие-либо элементы настоящего стандарта могут оказаться предметом патентных прав. Публикация настоящего стандарта не связана с существованием или действием каких-либо патентных прав. Ни ИСО, ни ИИЭР не несут ответственности ни за выявление любых патентов или патентных прав, по которым необходимо получение лицензии, ни за проведение исследований, являются ли разумными или недискриминационными любые лицензионные условия или положения, предусмотренные в связи с представлением гарантийного письма или патентного заявления и формы лицензионной декларации, если таковые имеются, или в любых лицензионных соглашениях. Пользователи настоящего стандарта несут ответственность за определение юридической силы любых патентных прав и за риск нарушения таких прав. Более подробная информация может быть получена в ИСО или в Ассоциации по стандартизации ИИЭР.

Стандарт ИСО/ИИЭР 11073-20601 подготовлен Комитетом по стандартизации 11073 Научного общества ИИЭР по техническим средствам медицины и биологии (как ИИЭР 11073-20601—2014). Он был одобрен Техническим комитетом ИСО/ТК 215 «Информатизация здоровья» и утвержден членами ИСО по ускоренной процедуре, которая определена в Соглашении о сотрудничестве между ИСО и ИИЭР. ИИЭР отвечает за поддержание настоящего стандарта при содействии со стороны стран — членов ИСО.

Аннотация: В контексте серии стандартов ИСО/ИИЭР 11073 по обмену данными между устройствами настоящий стандарт определяет общую основу для построения абстрактной модели персональной медицинской информации с помощью транспортно-независимого синтаксиса передачи, необходимого для формирования логических соединений между системами и предоставления возможностей и служб представления при решении коммуникационных задач. Протокол оптимизирован с учетом требований, предъявляемых к использованию персональной медицинской информации, и по возможности использует общеупотребительные методы и средства.

Ключевые слова: ИИЭР 11073-20601™, обмен данными с медицинскими приборами, персональные медицинские приборы

Важные уведомления и оговорки, касающиеся стандартизирующих документов ИИЭР

Документы ИИЭР доступны для использования в соответствии с важными уведомлениями и правовыми оговорками. Эти уведомления и оговорки или ссылка на данную страницу имеются во всех стандартах, и их можно отыскать под заголовком «Важное уведомление» или «Важные уведомления и правовые оговорки в отношении использования стандартизирующих документов ИИЭР».

Уведомление и правовая оговорка об ограничении ответственности в отношении использования стандартизирующих документов ИИЭР

Стандартизирующие документы ИИЭР (стандарты, рекомендованные практики и руководства), как утвержденные, так и для пробного использования, разрабатываются в научных обществах ИИЭР, а также в Координационных комитетах по стандартизации, относящихся к ведению Бюро стандартов Ассоциации по стандартизации ИИЭР (IEEE Standards Association, IEEE-SA). ИИЭР разрабатывает стандарты на основе процесса достижения консенсуса, одобренного Американским национальным институтом стандартов (American National Standards Institute, ANSI), который для получения окончательного документа сводит вместе добровольных участников, представляющих разные точки зрения и интересы. Добровольные участники не обязаны быть членами ИИЭР и работают на безвозмездной основе. Хотя ИИЭР управляет этим процессом и устанавливает правила по обеспечению беспристрастности в процессе достижения консенсуса, тем не менее ИИЭР не производит независимую оценку, тестирование или проверку точности какой-либо информации или обоснованность любых суждений, содержащихся в его стандартах.

ИИЭР не гарантирует и не подтверждает точность либо содержание материала, включенного в его стандарты, и явным образом отказывается от каких-либо гарантий (явных, неявных и предусмотренных законом), не включенных в этот или любой другой документ, относящийся к стандарту, включая, не ограничиваясь, такие гарантии, как: пригодность для продажи; пригодность для конкретной цели; отсутствие нарушения прав, а также качество, точность, эффективность, действительность или полнота материала. Кроме того, ИИЭР отказывается от каких-либо и всех условий, относящихся к результатам и качественному исполнению. Документы по стандартам ИИЭР предоставляются «КАК ЕСТЬ» и «БЕЗ ГАРАНТИИ».

Использование стандарта ИИЭР является абсолютно добровольным. Наличие стандарта ИИЭР не означает, что отсутствуют другие варианты изготовления, тестирования, измерения, покупки, рынка или предоставления других товаров и услуг, относящихся к области применения стандарта ИИЭР. Более того, точка зрения, выраженная в момент утверждения и выпуска стандарта, может измениться после изменений состояния дел, а также получения комментариев от пользователей стандарта.

Публикуя и делая стандарты доступными, ИИЭР тем самым не предлагает и не оказывает профессиональных либо других услуг от имени какого-либо лица или предприятия, а кроме того, ИИЭР не выполняет каких-то обязательств какого-либо другого лица или предприятия перед другими. Любое лицо, применяющее какой-либо стандарт ИИЭР, должно основываться на своем независимом суждении при соблюдении должной осторожности в любых указанных обстоятельствах или, в зависимости от конкретного случая, обратиться за советом к компетентному специалисту при определении правомерности указанного стандарта ИИЭР.

НИ ПРИ КАКИХ УСЛОВИЯХ ИИЭР НЕ БУДЕТ НЕСТИ ОТВЕТСТВЕННОСТЬ ЗА КАКИЕ-ЛИБО ПРЯМЫЕ, КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ТИПИЧНЫЕ УБЫТКИ ИЛИ ПОСЛЕДУЮЩИЙ УЩЕРБ (ВКЛЮЧАЯ, НЕ ОГРАНИЧИВАЯСЬ, ЗАКУПКУ ЗАМЕЩАЮЩИХ ТОВАРОВ ИЛИ УСЛУГ), А ТАКЖЕ ЗА НЕВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ ДАННЫХ ИЛИ ДОХОДОВ ЛИБО ЗА ОПЕРАЦИОННЫЙ ПРОСТОЙ, НЕЗАВИСИМО ОТ ПРИЧИН И ОСНОВАНИЙ ВОЗНИКНОВЕНИЯ ОТВЕТСТВЕННОСТИ, БУДЬ

ТО НАРУШЕНИЕ УСЛОВИЙ КОНТРАКТА, ПРЯМОЙ ОТВЕТСТВЕННОСТИ ИЛИ ВНЕДОГОВОРНОЙ ОТВЕТСТВЕННОСТИ, ВКЛЮЧАЯ ХАЛАТНОСТЬ И ДРУГИЕ ПРИЧИНЫ, ВОЗНИКАЮЩИЕ В РЕЗУЛЬТАТЕ ПУБЛИКАЦИИ, ИСПОЛЬЗОВАНИЯ ИЛИ ОПОРЫ НА ЛЮБОЙ СТАНДАРТ, ДАЖЕ ЕСЛИ БЫЛО СООБЩЕНО О ВОЗМОЖНОСТИ ТАКОГО УЩЕРБА, И ВНЕ ЗАВИСИМОСТИ ОТ ВОЗМОЖНОГО ПРОГНОЗИРОВАНИЯ ТАКОГО УЩЕРБА.

Переводы

Процесс разработки ИИЭР на основе достижения консенсуса включает в себя анализ документов только на английском языке. В случае перевода стандарта ИИЭР на другой язык только версия на английском языке, публикуемая ИИЭР, считается утвержденным стандартом ИИЭР.

Официальные заявления

Любое письменное или устное заявление, которое не прошло специальную процедуру отдела стандартов IEEE-SA, не должно рассматриваться или восприниматься в качестве официальной позиции ИИЭР или любого его комитета, а также не должно рассматриваться или восприниматься в качестве выраженной позиции ИИЭР. На лекциях, симпозиумах, семинарах или учебных курсах любое физическое лицо, предоставляющее информацию о стандартах ИИЭР, должно четко указать, что его взгляды необходимо рассматривать как личную точку зрения, а не официальную позицию ИИЭР.

Комментарии к стандартам

Комментарии по поводу изменения стандартизирующих документов ИИЭР принимаются от любой заинтересованной стороны независимо от ее принадлежности к ИИЭР. Однако ИИЭР не предоставляет консультации и рекомендации в отношении стандартизирующих документов ИИЭР. Предложения об изменении документов следует представлять в форме предлагаемого изменения текста вместе с подходящими сопроводительными комментариями. Поскольку стандарты ИИЭР представляют собой консенсус соответствующих интересов, необходимо, чтобы все ответы на комментарии и вопросы также обеспечивали баланс интересов. По этой причине ИИЭР, члены его научных обществ и координационных комитетов по стандартизации не могут предоставить незамедлительный ответ на комментарии или вопросы (за исключением ранее рассмотренных). По этой же причине ИИЭР не отвечает на просьбы о толковании. Любое лицо, которое хотело бы участвовать в изменении какого-либо стандарта ИИЭР, может присоединиться к соответствующей рабочей группе ИИЭР.

Комментарии к стандартам необходимо направлять по адресу:

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

Нормативные правовые акты

Пользователи стандартизирующих документов ИИЭР должны ознакомиться со всеми применимыми законами и нормами. Соблюдение положений любого стандартизирующего документа ИИЭР не означает соответствие каким-либо применимым нормативным требованиям. Разработчики стандарта несут ответственность за соблюдение или цитирование подходящих нормативных требований. Публикуя свои стандарты, ИИЭР не призывает к немедленным действиям, которые не согласуются с применимым законодательством. Кроме того, эти документы не могут толковаться как призыв к таким действиям.

Авторские права

Проекты и утвержденные версии стандартов ИИЭР охраняются авторским правом, принадлежащим ИИЭР в рамках национального (США) и международного законодательства об авторском праве. Они предоставляются ИИЭР для использования в различных общественных и личных целях. Например, они могут упоминаться в законах и нормативных актах, а также использоваться для частного саморегламентирования, стандартизации, продвижения способов и методов проектирования. Предоставляя эти документы для использования и применения уполномоченными органами и частными пользователями, ИИЭР не передает какие-либо авторские права на них.

Ксерокопии

При условии уплаты соответствующего сбора ИИЭР предоставит пользователям ограниченную, не исключительную лицензию на ксерокопирование частей любого отдельного стандарта только для некоммерческого внутреннего использования физическим лицом или компанией. По вопросам оплаты лицензионных сборов обращайтесь по адресу: Copyright Clearance Center, Customer Service,

222 Rosewood Drive, Danvers, MA 01923 USA, или по телефону: +1 978 750 8400. Кроме того, Copyright Clearance Center может предоставить разрешение на ксерокопирование частей любого отдельного стандарта для образовательных целей.

Обновление стандартизирующих документов ИИЭР

Пользователи стандартизирующих документов ИИЭР должны иметь в виду, что эти документы могут в любое время заменяться на их новые издания или время от времени корректироваться путем публикации изменений, поправок или списка опечаток. Официальный документ ИИЭР по состоянию на любой момент времени состоит из текущей редакции документа, дополненной изменениями, поправками или списками опечаток, вступившими в силу.

Каждый стандарт ИИЭР не реже одного раза в десять лет проходит процедуру пересмотра. Если документ не проходил процедуру пересмотра более десяти лет, разумно предположить, что такой документ по-прежнему имеет определенную ценность, однако не вполне точно характеризует фактическое положение дел. Пользователям настоятельно рекомендуется проверить наличие у них самого последнего издания необходимого стандарта ИИЭР.

Для определения степени актуальности данного документа и наличия дополнений к нему в виде опубликованных изменений, поправок или списков опечаток посетите веб-сайт IEEE-SA gost.e-ecolog.ru ieee.org/xpl/standards.jsp или обратитесь к ИИЭР по ранее указанному почтовому адресу. Дополнительные сведения о IEEE-SA и процессе разработки стандартов ИИЭР доступны на веб-сайте IEEE-SA по адресу: gost.e-ecolog.ru .

Список опечаток

Со списком опечаток (если имеется) в стандартах ИИЭР можно ознакомиться на веб-сайте ИИЭР-СА по следующему адресу: gost.e-ecolog.ru . Пользователям рекомендуется периодически посещать эту веб-страницу для ознакомления со списком опечаток.

Патенты

Необходимо учесть, что для внедрения настоящего стандарта может потребоваться использование предмета, на который распространяется действие патентных прав. Опубликование настоящего стандарта не означает, что ИИЭР проведена проверка существования или действительности каких-либо патентных прав в связи с вышеизложенным. Если владелец или заявитель патента зарегистрировал заявление с использованием принятого гарантийного письма, такое заявление публикуется на веб-сайте ИИЭР-СА по адресу: gost.e-ecolog.ru . Гарантийные письма могут содержать сведения о том, что отправитель готов или не готов предоставить лицензии в рамках патентных прав без компенсации или за разумное вознаграждение при разумных условиях и положениях, которые явно свободны от любой недобросовестной дискриминации заявителей, желающих получить такие лицензии.

Возможно наличие существенных пунктов формулы изобретения, для которых не получено гарантийное письмо. ИИЭР не несет ответственности за идентификацию существенных пунктов формулы изобретения, для которых может потребоваться лицензия, а также за выяснение законности или области применения пунктов формулы изобретения, или за определение разумности или недискриминационности каких-либо условий или положений лицензии, предоставленной в связи с отправкой гарантийного письма (при наличии) или в любых лицензионных соглашениях. Пользователи настоящего стандарта несут прямую ответственность в части определения законности любых патентных прав и риска нарушения таких прав. IEEE Standards Association может предоставить необходимую дополнительную информацию.

Участники

На момент отправки этого стандарта в Бюро стандартов IEEE-SA с целью согласования рабочая группа по персональным медицинским приборам имела следующий состав:

Дайди Джонг (Daidi Zhong), *сопредседатель*
Майкл Дж. Кирван (Michael J. Kirwan), *сопредседатель*
Дуглас П. Боджиа(Douglas P. Bogia), *сопредседатель*

Charles R. Abbruscato
Nabil Abujbara
Maher Abuzaid
Jinhan Chung
Malcolm Clarke
John A. Cogan
Raul Gonzalez Gomez
Chris Gough
Channa Gowda

Manfred Aigner
Jorge Alberola
Karsten Alders
Murtaza Ali
Rolf Ambuehl
David Aparisi
Lawrence Arne
Diego B. Arquillo
Serafin Arroyo
Muhammad Asim
Merat Bagha
Doug Baird
David Baker
Anindya Bakshi
Ananth Balasubramanian
Sunlee Bang
M. Jonathan Barkley
Gilberto Barron
David Bean
John Bell
Rudy Belliardi
Kathryn M. Bennett
Daniel Bernstein
George A. Bertos
Chris Biernacki
Ola Björsne
Thomas Blackadar
Marc Blanchet
Thomas Bluethner
Xavier Boniface
Shannon Boucousis
Julius Broma
Lyle G. Bullock
Bernard Burg
Chris Burns
Anthony Butt
Jeremy Byford-Rew
Satya Calloji
Carole C. Carey
Santiago Carot-Nemesio
Randy W. Carroll
Simon Carter
Seungchul Chae
Rahul Chauhan
James Cheng
Peggy Chien
Chia-Chin Chong
Saeed A. Choudhary
Ralph Kent
Laurie M. Kermes
Ikuo Keshi

John T. Collins
Cory Condek
Todd H. Cooper
David Cornejo
Douglas Coup
Nigel Cox
Hans Crommenacker
Tomio Crosley
David Culp
Allen Curtis
Ndifor Cyril Fru
Jesus Daniel Trigo
Eyal Dassau
David Davenport
Russell Davis
Ed Day
Sushil K. Deka
Pedro de-las-Heras-Quiros
Jim DelloStritto
Matthew d'Entremont
Lane Desborough
Kent Dicks
Hyoungho Do
Xiaolian Duan
Brian Dubreuil
Jakob Ehrensvard
Fredrik Einberg
Roger M. Ellingson
Michihiro Enokida
Javier Escayola Calvo
Leonardo Estevez
Roger Feeley
Bosco T. Fernandes
Christoph Fischer
Morten Flintrup
Joseph W. Forler
Russell Foster
Eric Freudenthal
Matthias Frohner
Ken Fuchs
Jing Gao
Marcus Garbe
John Garguilo
Rick Geimer
Igor Gejdos
Ferenc Gerbovics
Nicolae Goga
Julian Goldman
Jim Niswander
Hiroaki Niwamoto
Thomas Norgall

Charles M. Gropper
Amit Gupta
Jeff Guttmacher
Rasmus Haahr
Christian Habermann
Michael Hagerty
Jerry Hahn
Robert Hall
Nathaniel Hamming
Rickey L. Hampton
Sten Hanke
Jordan Hartmann
Kai Hassing
Marc Daniel Haunschild
Wolfgang Heck
Charles Henderson
Jun-Ho Her
Takashi Hibino
Timothy L. Hirou
Allen Hobbs
Alex Holland
Arto Holopainen
Robert Hoy
Frank Hsu
Anne Huang
Sen-Der Huang
Zhiqiang Huang
Ron Huby
Robert D. Hughes
David Hughes
Jiyoung Huh
Hugh Hunter
Hitoshi Ikeda
Yutaka Ikeda
Philip O. Isaacson
Atsushi Ito
Michael Jaffe
Praduman Jain
Danny Jochelson
Chris Johnson
Phaneeth Junga
Akiyoshi Kabe
Steve Kahle
Tomio Kamioka
Kei Kariya
Andy Kaschl
Junzo Kashihara
Kohichi Kashiwagi
Sternly K. Simon
Marjorie Skubic
Robert Smith

Junhyung Kim
Min-Joon Kim
Minho Kim
Taekon Kim
Tetsuya Kimura
Alfred Kloos
Jeongmee Koh
Jean-Marc Roller
John Koon
Patty Krantz
Alexander Kraus
Ramesh Krishna
Geoffrey Kruse
Falko Kuester
Rafael Lajara
Pierre Landau
Jaechul Lee
JongMuk Lee
Kyong Ho Lee
Rami Lee
Sungkee Lee
Woojae Lee
Yonghee Lee
Joe Lenart
Kathryn A. Lesh
Qiong Li
Ying Li
Patrick Lichter
Jisoon Lim
Joon-Ho Lim
John Lin
Jiajia Liu
Wei-Jung Lo
Charles Lowe
Don Ludolph
Christian Luszick
Bob MacWilliams
Srikkanth Madhurbootheswaran
Romain Marmot
Sandra Martinez
Miguel Martinez de Espronceda Camara
Peter Mayhew
Jim McCain
Laszlo Meleg
Alexander Mense
Ethan Metsger
Yu Miao
Jinsei Miyazaki
Erik Moll
Darr Moore
Anand Noubade
Yoshiteru Nozoe
Abraham Ofek
Brett Olive
Begonya Otal
Charles Palmer
Bud Panjwani
Carl Pantiskas
Harry P. Pappas
Mikey Paradis
Hanna Park
Jong-Tae Park
Myungeun Park
Soojun Park
Phillip E. Pash
TongBi Pei
Soren Petersen
James Petisce
Peter Piction
Michael Pliskin
Jeff Price
Harald Prinzhorn
John Quinlan
Arif Rahman
Tanzilur Rahman
Steve Ray
Phillip Raymond
Tim Reilly
Barry Reinhold
Brian Reinhold
Melvin I. Reynolds
John G. Rhoads
Jeffrey S. Robbins
Moskowitz Robert
Timothy Robertson
David Rosales
Bill Saltzstein
Benedikt Salzbrunn
Giovanna Sannino
Jose A. Santos-Cadenas
Stefan Sauermann
John Sawyer
Guillaume Schatz
Alois Schloegl
Paul S. Schluter
Lars Schmitt
Mark G. Schnell
Richard A. Schrenker
Antonio Scorpiniti
Kwang Seok Seo
Riccardo Serafin
Ivan Soh
Motoki Sone
Emily Sopensky
Rajagopalan Srinivasan
Andreas Staubert
Nicholas Steblay
Beth Stephen
Lars Steubesand
John (Ivo) Stivoric
Raymond A. Strickland
Hermanni Suominen
Lee Surprenant
Ravi Swami
Ray Sweidan
Jin Tan
Haruyuyki Tatsumi
John W. Thomas
Brad Tipler
Jonas Tirén
James Tomcik
Janet Traub
Gary Tschautscher
Masato Tsuchid
Ken Tubman
Yoshihiro Uchida
Sunil Unadkat
Fabio Urbani
Philipp Urbauer
Laura Vanzago
Alpo Värri
Ciro de la Vega
Dalimar Velez
Naveen Verma
Rudi Voon
Isobel Walker
David Wang
Jerry P. Wang
Yao Wang
Yi Wang
Steve Warren
Fujio Watanabe
Tom Watsuji
Mike Weng
Kathleen Wible
Paul Williamson
Jan Wittenber
Jia-Rong Wu
Will Wykeham
Ariton Xhafa
Junjie Yang
Ricky Yang

Piotr Murawski	Sid Shaw	Melanie Yeung
Soundharya Nagasubramanian	Frank Shen	Done-Sik Yoo
Jae-Wook Nah	Liqun Shen	Jason Zhang
Alex Neefus	Bozhi Shi	Zhiqiang Zhang
Trong-Nghia Nguyen-Dobinsky	Min Shih	Thomas Zhao
Michael E. Nidd	Mazen Shihabi	Miha Zoubek
Tetsu Nishimura	Redmond Shouldice	Szymon Zysko

Голосование, посвященное утверждению этого стандарта, проходило с привлечением нижеуказанных участников соответствующего комитета. Участники голосования могут высказаться за утверждение или отклонение стандарта, а также воздержаться при голосовании.

Hector Barron Gonzalez	David Friscia	Charles Ngethe
Pieter Botman	David Fuschi	Melvin I. Reynolds
Lyle G. Bullock	Randall Groves	Terence Rout
Juan Carreon	Kai Hassing	Bartien Sayogo
Randy W. Carroll	Werner Hoelzl	Lars Schmitt
Lawrence Catchpole	Ruimin Hu	Carl Singer
Jianwen Chen	Noriyuki Ikeuchi	Kapil Sood
Keith Chow	Akio Iso	Raymond A. Strickland
Donald Cragun	Atsushi Ito	Walter Struppler
Paul Croll	Raj Jain	Jiande Sun
Russell Davis	Junghoon Jee	Hung-Yu Wei
Douglas Dorr	Piotr Karocki	Jan Wittenber
Sourav Dutta	Stuart Kerry	Oren Yuen
Christoph Fischer	Geoff Ladwig Richard Lancaster	Daidi Zhong

Настоящий стандарт утвержден IEEE-SA 21 августа 2014 года в следующем составе:

Джон Кулик (John Kulick), *председатель*
Йон Уолтер Родел (Jon Walter Rosdahl), *заместитель председателя*
Ричард Х. Халетт (Richard H. Hulett), *предыдущий председатель*
Константинос Карачалиос (Konstantinos Karachalios), *секретарь*

Peter Balma	Michael Janezic	Ron Peterson
Farooq Bari	Jeffrey Katz	Adrian Stephens
Ted Burse	Joseph L. Koepfinger*	Peter Sutherland
Clint Chaplain	David Law	Yatin Trivedi
Stephen Dukes	Hung Ling	Phil Winston
Jean-Phillippe Faure	Oleg Logvinov	Don Wright
Gary Hoffman	T. W. Olsen	Yu Yuan
	Glenn Parsons	

В состав отдела стандартов IEEE-SA также входят следующие участники, которые не голосовали:

Ричард Дэблэсио (Richard DeBlasio), *представитель DOE*
Майкл Янежич (Michael Janezic), *представитель NIST*
Дон Мессина (Don Messina)
Издательство IEEE-SA
Кэтрин Беннетт (Kathryn Bennett)
Программы технического сообщества IEEE-SA

* Заслуженный участник

Введение

Данное введение не является частью стандарта ИИЭР 11073-20601-2014 «Информатизация здоровья — Обмен данными с персональными медицинскими приборами — Часть 20601. Прикладной профиль — Оптимизированный протокол обмена».

Стандарты ИСО и ИИЭР 11073 регламентируют обмен данными между медицинскими устройствами и внешними компьютерными системами. Настоящий стандарт и соответствующие стандарты ИИЭР 11073-104zz ориентированы на необходимость упрощенного и оптимизированного подхода к обмену данными для персональных регистрируемых или нерегистрируемых медицинских приборов. Такие стандарты согласуются с имеющимися клинически ориентированными стандартами и разработаны на их основе, чтобы обеспечить простое управление данными, полученными от клинических или персональных медицинских приборов.

Настоящий стандарт представляет собой открытый независимый стандарт, регламентирующий преобразование собранной информации в интероперабельный формат передачи информации между агентами и менеджерами.

Существуют следующие тесно связанные между собой стандарты:

- ИИЭР 11073-00103-2012 [B5]* содержит основные сведения о предметной области личного здоровья, а также определяет базовые варианты и модели использования;
- ИСО/ ИИЭР 11073-10101 [B16] документирует номенклатурные термины, доступные для использования;
- ИСО/ИИЭР 11073-10201:2004 [B17] документирует расширенную информационную модель предметной области (DIM), используемую в настоящем стандарте;
- стандарты ИСО/ИИЭР 11073-104zz определяют конкретные специализации приборов. Например, ИСО/ИИЭР 11073-10404 [B18] указывает, каким образом работают пульсоксиметры, обеспечивающие взаимодействие;
- ИСО/ИИЭР 11073-20101:2004 [B21] определяет правила кодирования медицинских приборов (MDER), используемые в настоящем стандарте.

* Числа в скобках соответствуют порядковой нумерации библиографических источников в приложении К.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАТИЗАЦИЯ ЗДОРОВЬЯ

Обмен данными с персональными медицинскими приборами

Часть 20601

Прикладной профиль. Оптимизированный протокол обмена

Health informatics. Personal health device communication.
Part 20601. Application profile. Optimized exchange protocol

Дата введения — 2020—05—01

ВНИМАНИЕ! Стандартизирующие документы ИИЭР не предназначены для обеспечения безопасности, защищенности, охраны здоровья или защиты окружающей среды либо защиты от помех со стороны других устройств или сетей. Исполнители, занимающиеся практической реализацией стандартизирующих документов ИИЭР, несут ответственность за определение и обеспечение соответствия всем подходящим методикам в области физической и информационной безопасности, защиты окружающей среды и здоровья, защиты от помех, а также за соблюдение всех требований действующего законодательства и нормативных документов.

Данный документ ИИЭР доступен для использования в соответствии с важными уведомлениями и правовыми оговорками. Такие уведомления и оговорки содержатся во всех публикациях, содержащих настоящий документ, и выделяются заголовком «Важное уведомление» или «Важные уведомления и оговорки, касающиеся документов ИИЭР». Их также можно получить, обратившись с запросом к ИИЭР, либо просмотреть на сайте gost.e-ecolog.ru ***org/IPR/disclaimers.html.***

1 Общие положения

1.1 Область применения

В контексте серии стандартов ИСО/ИИЭР 11073, посвященных персональным медицинским приборам, настоящий стандарт определяет оптимизированный протокол обмена и методы моделирования, которые будут использоваться разработчиками персональных медицинских приборов для обеспечения интероперабельности между типами приборов разных изготовителей, построения абстрактной модели персональной медицинской информации с помощью транспортно-независимого синтаксиса передачи, необходимого для формирования логических соединений между системами и предоставления возможностей и служб представления, необходимых для выполнения задач коммуникации. Протокол оптимизирован с учетом требований, предъявляемых к использованию персональной медицинской информации и, по возможности, использует общеупотребительные методы и средства.

1.2 Цель

Настоящий стандарт отвечает потребности в открытом независимом стандарте управления обменами информацией между персональными медицинскими приборами и менеджерами (например, мобильными телефонами, персональными компьютерами, персональными медицинскими устройствами и телевизионными приставками). Интероперабельность имеет ключевое значение для потенциального роста рынка таких устройств и улучшения информированности людей об управлении своим здоровьем.

Издание официальное

1.3 Контекст

На рисунке 1 показаны категории и типичные приборы, предназначенные для персонального медицинского применения. Агенты (например, мониторы артериального давления, весы и шагомеры) собирают информацию о лице (или лицах и передают эту информацию менеджеру (например, мобильному телефону, медицинскому устройству или персональному компьютеру) для хранения, отображения и, возможно, последующей передачи. Менеджер может также пересылать данные дистанционным службам поддержки с целью дальнейшего анализа. Информация доступна в определенном диапазоне предметных областей, среди которых управление течением заболевания, здоровье, фитнес и одинокое старение.

Канал связи между агентом и менеджером предполагается логическим двухточечным соединением. Обычно в любой момент времени агент взаимодействует с одним менеджером. Менеджер может обмениваться данными одновременно с несколькими агентами, используя отдельные двухточечные соединения.

Пунктирное наложение показывает главное направление деятельности Рабочей группы по персональным медицинским приборам ИИЭР 11073™. Первостепенный интерес представляют интерфейс и обмен данными между агентами и менеджером. Однако данный интерфейс невозможно создать изолированно, игнорируя остальную часть пространства решений. Восприятие всей системы в целом помогает при необходимости обеспечить целесообразную передачу данных от агентов к дистанционным службам поддержки. Она может включать в себя преобразования формата данных, протоколы обмена и транспортные протоколы через различные интерфейсы. Большая часть усилий по стандартизации находится вне области деятельности Рабочей группы по персональным медицинским приборам, однако объединение усилий по стандартизации позволяет осуществлять бесперебойную передачу данных через все множество систем.

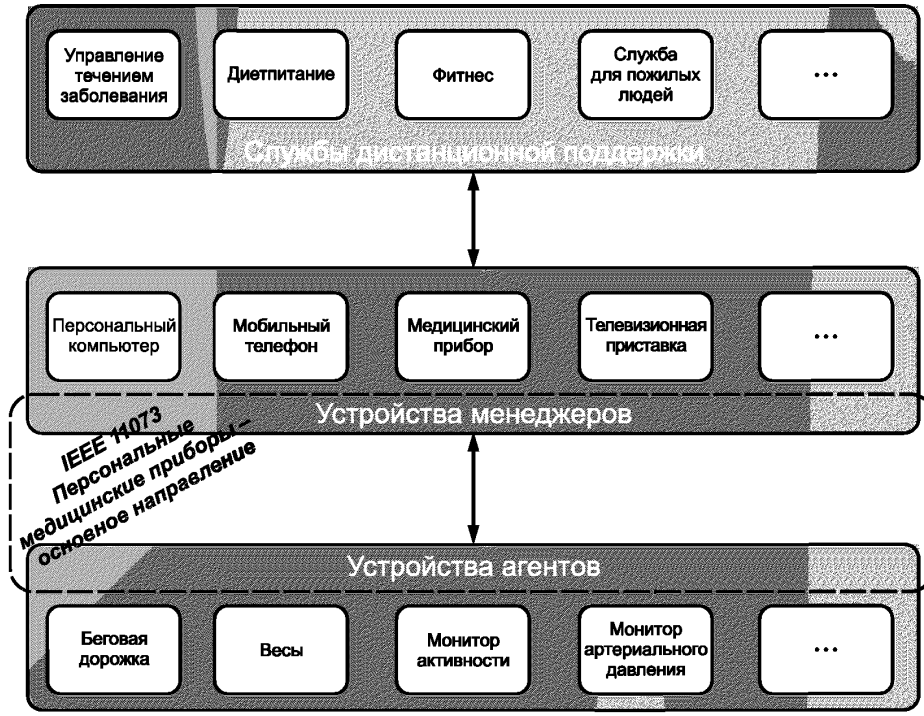


Рисунок 1 — Общий контекст работы

На рисунке 2 показано иерархическое представление архитектуры агента или менеджера с указанием соответствующих стандартов. Прикладные уровни в основном не специфичны для какого-либо конкретного транспортного протокола. Там, где это необходимо, настоящий стандарт определяет допущения, которые требуют прямой поддержки на транспортном уровне или некотором «промежуточном» уровне над ним. Такой подход позволяет поддерживать разные транспортные протоколы. Определение транспортных протоколов находится вне области применения настоящего стандарта и деятельности рабочей группы.

Над транспортным уровнем расположен оптимизированный протокол обмена (описанный в настоящем стандарте). Данный протокол регламентирует два аспекта: службы прикладного уровня и определение протокола обмена данными между агентами и менеджерами. Службы прикладного уровня предоставляют протокол управления соединением и надежной передачи действий и данных между агентом и менеджером. Протокол обмена данными определяет команды, информацию о конфигурациях агента, формат данных и общий протокол. Оптимизированный протокол обмена обеспечивает основу для поддержки агента любого типа. Для приборов конкретного типа читателю необходимо ознакомиться со специализацией прибора агента, чтобы понять его возможности и условия реализации согласно требованиям настоящего стандарта. Специализация прибора указывает, какие аспекты настоящего стандарта следует охватить и где найти дополнительную информацию по реализации соответствующего прибора.

Над протоколом обмена находятся специализации приборов, описывающие специфичные сведения об определенном агенте (например, мониторе артериального давления, весах или шагомере). Специализации предоставляют подробную информацию о том, как работают такие агенты, и служат детальным описанием, полезным для создания агента специфичного типа. Кроме того, специализации ссылаются на соответствующие стандарты, содержащие дополнительную информацию. Номера стандартов, посвященных специализациям приборов, варьируются в диапазоне от ИИЭР 11073-10401 до ИИЭР 11073-10499 включительно. При ссылке на стандарты используется указание *ИИЭР 11073-104zz*, где zz — любое число в диапазоне от 01 до 99 включительно.

Некоторые специализации медицинских приборов охватывают широкие категории типов приборов (например, ИИЭР 11073-10441™ моделирует типы оборудования, способствующего укреплению сердечно-сосудистой деятельности, например шагомеры или тренажеры). Другие специализации приборов имеют узкий охват и концентрируются на приборах одного типа (например, ИИЭР 11073-10408™ моделирует термометры). Специализации, предназначенные для одного или нескольких типов приборов, могут также определять *профили*. Профиль дополнительно ограничивает модель, определенную в специализации, чтобы повысить интероперабельность (например, профиль шагомеров использует ограниченную часть модели из ИИЭР 11073-10441).

Технический отчет ИИЭР 11073-00103-2012 [B5][1] описывает предметную область персонального здоровья с последующим пояснением базовых вариантов и моделей использования.

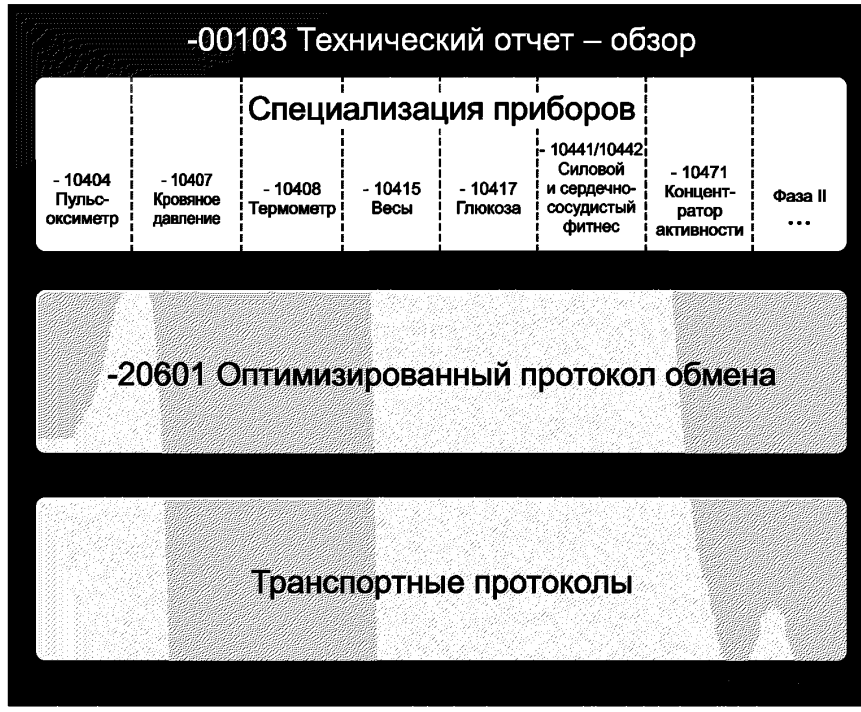


Рисунок 2 — Схема документа

[1] Числа в скобках соответствуют порядковой нумерации библиографических источников в приложении К.

Специализации персональных медицинских приборов не создаются независимо от всех других стандартов. Существует целый ряд действующих стандартов, предназначенных для применения в условиях медицинских учреждений, из которых производятся эти специализации. На рисунке 3 показана связь с остальными документами ИИЭР 11073. Существуют два типа связей:

- использование базовых идей и/или фрагментов из других документов (пунктирные линии);
- эффективное использование информации из другого документа и добавление нового текста в документ для поддержки настоящего стандарта (сплошные линии).

Настоящий стандарт заимствует информацию из ИСО/ИИЭР 11073-10201:2004 [B17] и ИСО/ИИЭР 11073-20101:2004 [B21] в качестве обязательных приложений. При наличии несоответствий между этими стандартами приоритет имеет настоящий стандарт. В связи с заимствованием структур из этих стандартов некоторые наименования могут оказаться более ориентированными на медицинские учреждения (например, система медицинского прибора (MDS) вместо системы персонального медицинского прибора), однако для сохранения согласованности используются традиционные наименования.

Настоящий стандарт копирует релевантные части стандарта ИСО/ИИЭР 11073-10101 [B16] и добавляет новые номенклатурные коды.

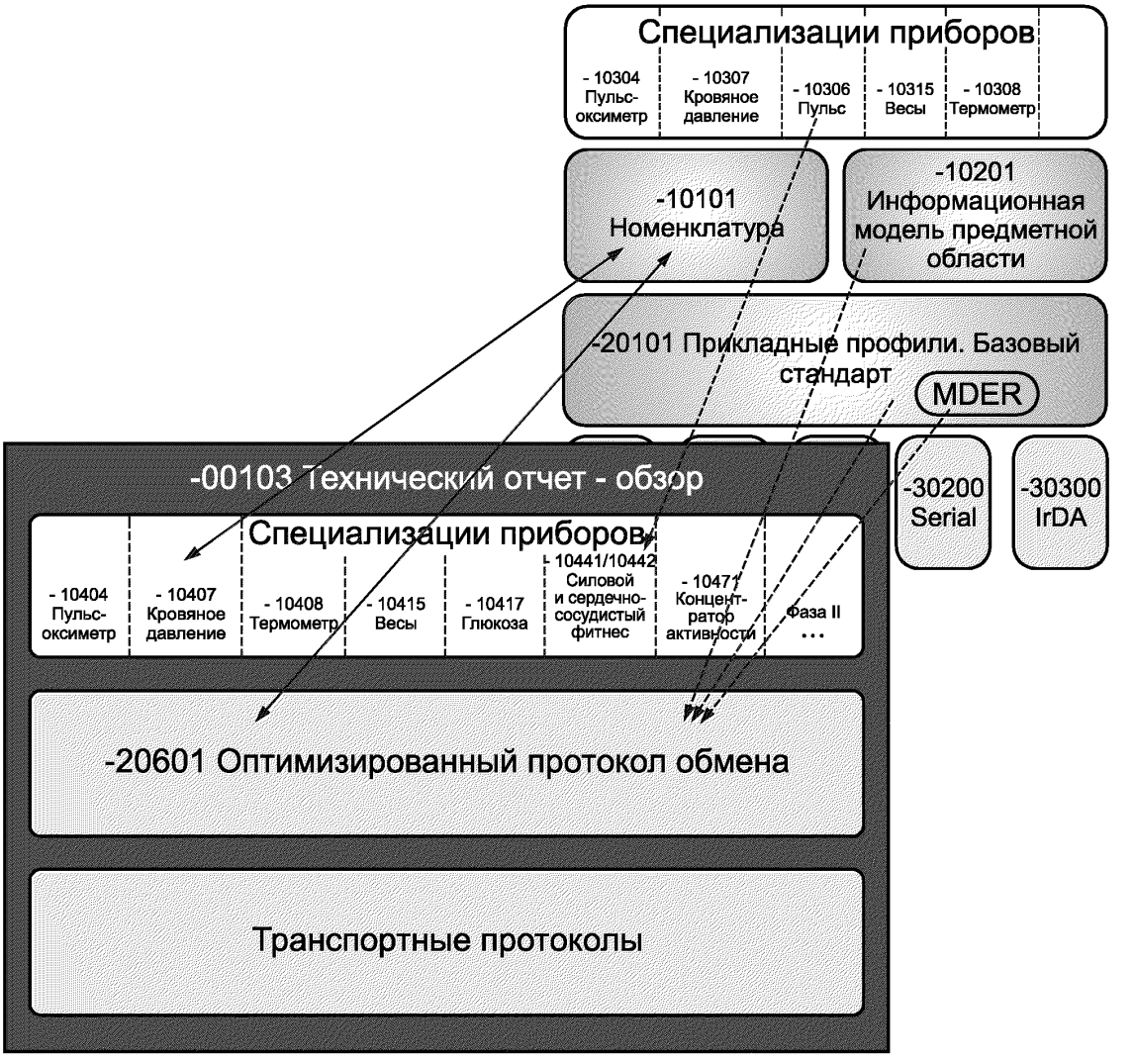


Рисунок 3 — Взаимосвязь с другими документами ИИЭР 11073

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты. Для датированных ссылок применяют только указанные издания. Для недатированных ссылок применяют последние издания (включая любые изменения к стандартам).

ИИЭР 802®-2014, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture (Стандарт ИИЭР для локальных и городских вычислительных сетей: общие положения и архитектура)[2,3]

ИИЭР 1541™-2002 (подтв. 2008), IEEE Standard for Prefixes for Binary Multiples (Стандарт ИИЭР для префиксов двоичных кратных величин)

ИСО/МЭК 80000-13:2008, Quantities and units — Part 13: Information science and technology (Физические величины и единицы измерения. Часть 13. Информатика и информационные технологии)[4,5]

ITU-T Rec. X.667 (сентябрь 2004), Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities: Generation and registration of universally unique identifiers (UUIDs) and their use as ASN.1 object identifier (Информационная технология. Взаимосвязь открытых систем. Процедуры работы уполномоченных по регистрации ВОС. Создание, регистрация универсально уникальных идентификаторов (УУИд) и их использование в качестве компонентов идентификатора объекта АСН.1)[6]

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте применены термины с соответствующими определениями. Определения терминов, не указанных в данном разделе, см. в Онлайновом словаре по стандартам ИИЭР (*IEEE Standards Dictionary Online*)[7].

3.1.1 **агент** (agent): Узел, собирающий и передающий персональные медицинские данные ассоциированному менеджеру.

3.1.2 **атрибут** (attribute): Данные, представляющие свойство объекта. Атрибуты совместно с действиями определяют объект.

3.1.3 **AttributeChangeSet:** Набор изменений значений атрибутов, представляющий атомарное изменение объекта.

Примечание — Система медицинского прибора (MDS) или объект Scanner уведомляются о завершении AttributeChangeSet. Группы этих наборов AttributeChangeSet отображаются системой медицинского прибора или объектом Scanner в одну из структур ObservationScan отчета о событиях сканирования, передаваемого менеджеру. Перед извлечением любого семантического поведения менеджер обновляет свой объект с помощью набора изменений значений атрибутов, содержащихся в ObservationScan.

3.1.4 **вычислительная система** (compute engine): См. **менеджер**.

3.1.5 **подтверждение** (confirmed): Механизм службы прикладного уровня, обеспечивающий уведомление о завершении.

Примечание — Для служб EVENT REPORT (т.е. на уровне данных) подтверждение позволяет агенту узнать, когда менеджер «принял ответственность» за часть данных, после чего агент может ее удалить. Для служб ACTION, GET и SET (т.е. на уровне управления) подтверждение позволяет менеджеру узнать, когда агент «завершил» запрошенную транзакцию.

---

[2] Публикации ИИЭР распространяются Институтом инженеров по электротехнике и электронике (gost.e-ecolog.ru).

[3] Стандарты или продукция ИИЭР, упомянутые в настоящем разделе, являются товарными знаками, принадлежащими Институту инженеров по электротехнике и электронике.

[4] Настоящий стандарт отменяет и замещает разделы 3.8 и 3.9 стандарта МЭК 60027-2 (2005).

[5] Публикации ИСО/МЭК распространяются Международной организацией по стандартизации (gost.e-ecolog.ru Кроме того, на территории США публикации ИСО/МЭК распространяются компанией Global Engineering Documents (gost.e-ecolog.ru). Электронные копии на территории США распространяются Американским национальным институтом стандартов (gost.e-ecolog.ru

[6] Публикации ITU-T распространяются Международным союзом электросвязи (gost.e-ecolog.ru

[7] Подписки на IEEE Standards Dictionary Online доступны по адресу: gost.e-ecolog.ru

3.1.6 **устройство** (device): Физический прибор, выполняющий роль агента или менеджера.

3.1.7 **динамический атрибут** (dynamic attribute): Атрибут, значение которого может изменяться во время ассоциации.

Примечание — Значение атрибута следует отправлять во время конфигурирования, а также во время или до того момента времени, когда такое значение потребуется для интерпретации сообщенных результатов наблюдения. Значение может обновляться позже (например, в отчете о событиях данных сегмента или сканирования). Значение атрибута остается действующим, пока не будет изменено в последующем отчете о событиях данных сегмента или сканирования или пока система не завершит ассоциацию.

3.1.8 **дескриптор** (handle): Локально уникальное беззнаковое 16-битовое число, идентифицирующее один из экземпляров объекта внутри агента.

3.1.9 **менеджер** (manager): Узел, получающий данные от одной или нескольких агентских систем.

Примечание — Примерами менеджеров могут служить мобильный телефон, медицинский прибор, телевизионная приставка или компьютерная система.

3.1.10 **объект Metric** (metric): Объект, используемый для моделирования различных видов измерений.

3.1.11 **наблюдаемый атрибут** (observational attribute): Атрибут, значение которого может меняться на протяжении срока существования ассоциации.

Примечание — Значение может передаваться в отчете о событиях данных сегмента или сканирования. Если получен набор значений наблюдаемого атрибута, такие значения объединяются с доступной контекстной информацией (например, значениями всех связанных динамических и статических атрибутов), чтобы представить результаты, полученные во время наблюдения. В отличие от значений динамических и статических атрибутов значения наблюдаемого атрибута объединяются с контекстной информацией только однократно (например, значения наблюдаемого атрибута не используются повторно при последующем получении любых новых значений атрибута).

3.1.12 **объект** (object): Единица, обладающая некоторой функциональностью, или элемент устройства, свойства которых описываются атрибутами.

Примечание — Объекты Metric представляют измерения (например, артериального давления, веса или температуры), система медицинского прибора (MDS) — устройство, объекты постоянного хранения данных Metric (PM-store) — механизмы постоянного хранения в памяти агента, а объекты Scanner — механизм управления и отчетности.

3.1.13 **персональный медицинский прибор** (personal health device): Прибор, используемый для личного медицинского применения.

3.1.14 **персональный телемедицинский прибор** (personal telehealth device): См. **персональный медицинский прибор**.

3.1.15 **статический атрибут** (static attribute): Атрибут, значение которого не изменяется (остается фиксированным) на протяжении срока существования ассоциации.

Примечание 1 — Значение передается в отчете о событиях конфигурации. Значение остается действующим до момента выхода системы из состояния ассоциации.

Примечание 2 — Необходимо отличать понятие «статический» (static) в рамках настоящего стандарта от конструкции static в языке программирования С[8].

3.2 Сокращения

В настоящем стандарте применены следующие сокращения:

APDU — блок данных прикладного протокола (Application protocol data unit);

API — прикладной программный интерфейс (Application Programming Interface);

ASCII — американский стандартный код для обмена информацией; код ASCII[9] (American Standard Code for Information Interchange);

ASN.1 — Абстрактная синтаксическая нотация версии 1, АСН.1 (Abstract Syntax Notation One);

AVA — объявление значения атрибута (Attribute Value Assertion);

BER — базовые правила кодирования (Binary Encoding Rules);

DIM — информационная модель предметной области (Domain Information Model);

---

[8] Примечания в тексте, таблицах и на рисунках стандарта предназначены исключительно для информационных целей и не содержат требования по применению настоящего стандарта.

[9] В настоящем стандарте термин «ASCII» подразумевает набор знаков согласно стандарту ИСО/МЭК 646:1991 [В14].

DST — летнее время (Daylight Savings Time);
ECG — Электрокардиограмма или электрокардиограф (ЭКГ);
EUI-64 — 64-битный расширенный уникальный идентификатор (Extended Unique Identifier);
FIFO — первым пришел — первым ушел (First-In, First-Out);
GMDN — глобальная номенклатура медицинских приборов (Global Medical Device Nomenclature);
ICS — заявление о соответствии реализации, ЗСР (Implementation Conformance Statement);
ID — идентификатор (IDentifier);
LSB — младший значащий бит (Least Significant Bit);
MDER — правила кодирования медицинских приборов (Medical Device Encoding Rules);
MDNF — цифровой формат медицинских приборов (Medical Device Numeric Format);
MDS — система медицинского прибора (Medical Device System);
MOC — класс медицинского объекта (Medical Object Class);
MSB — старший значащий бит (Most Significant Bit);
NaN — не число (Not A Number);
NBO — порядок передачи байтов в сети (Network Byte Order);
NRes — не при этом разрешении экрана (Not at this Resolution);
NTP — сетевой протокол службы времени (Network Time Protocol);
OID — объектный идентификатор, ОИД (Object IDentifier);
OUI — идентификатор, уникальный в пределах организации (Organizationally Unique Identifier);
PDU — блок данных протокола (Protocol Data Unit);
PER — правила уплотненного кодирования (Packed Encoding Rules);
PM — постоянная метрика (Persistent Metric);
POC — Объект и класс информационной модели предметной области персональных медицинских приборов, ОК ПМП (personal health device domain information model object and class)
RCassoc — Счетчик попыток: процедура ассоциации;
RTC — часы реального времени (Real-Time Clock);
RT-SA — массив считываний в реальном времени (Real-Time Sample Array);
SCADA — система диспетчерского управления и сбора данных (Supervisory Control And Data Acquisition);
SNTP — простой сетевой протокол службы времени (Simple Network Time Protocol);
TCP — протокол управления передачей данных (Transmission Control Protocol);
TOassoc — Тайм-аут: процедура ассоциации;
TOca — Тайм-аут: служба подтвержденного действия;
TOcer-mds — Тайм-аут: служба подтвержденного отчета о событиях для объекта системы медицинского прибора;
TOcer-pms — Тайм-аут: служба подтвержденного отчета о событиях для объекта PM-store;
TOcer-scan — Тайм-аут: служба подтвержденного отчета о событиях для объекта Scanner;
TOclr-pms — Тайм-аут: служба подтвержденного события для очистки объекта PM-store;
TOconfig — Тайм-аут: процедура конфигурации;
TOcs — Тайм-аут: служба подтвержденного набора;
TOget — Тайм-аут: служба GET;
TOrelease — Тайм-аут: процедура завершения ассоциации;
TOsp-mds — Тайм-аут: специальный период времени между службами для объекта системы медицинского прибора;
TOsp-pms — Тайм-аут: специальный период времени передачи сегмента для объекта PM-segment;
USB — универсальная последовательная шина (universal serial bus);
UTC — всемирное координированное время (Coordinated Universal Time);
UUID — универсальный уникальный идентификатор, УУИд (Universally Unique IDentifier);
XER — правила XML-кодирования (Extensible Markup Language (XML) Encoding Rules).

4 Руководящие принципы

Настоящий стандарт и другие стандарты, посвященные персональным медицинским приборам, связаны с более широким контекстом серии стандартов ИСО/ИИЭР 11073. Полный набор стандартов позволяет агентам соединяться и взаимодействовать с менеджерами и компьютеризованными медицинскими информационными системами.

Коммуникационный профиль, определенный в настоящем стандарте, учитывает специфичные требования медицинских агентов и менеджеров, которые обычно используются за пределами медицинского учреждения (например, в мобильных условиях или на дому), а именно:

- Агенты персональных медицинских приборов обычно имеют очень ограниченные вычислительные возможности.

- Агенты персональных медицинских приборов обычно имеют постоянную конфигурацию и используются с одним менеджером.

- Агенты персональных медицинских приборов зачастую представляют собой переносные приборы с батарейным питанием, использующие канал беспроводной связи. Следовательно, крайне важна энергоэффективность протокола.

- Агенты персональных медицинских приборов зачастую не активны постоянно. Например, весы могут предоставлять данные только один или два раза в сутки. Для минимизации эксплуатационных расходов на такие приборы необходима процедура эффективного соединения.

- Менеджеры персональных медицинских приборов обычно обладают большей вычислительной мощностью, оперативной памятью и пространством хранения данных, поэтому протокол намеренно возлагает большую нагрузку на менеджеры.

- Агенты и менеджеры персональных медицинских приборов передают информацию, которая может оказаться полезной медицинским специалистам. Поэтому качество данных может рассматриваться как полезное для клинических целей, даже если эти данные собраны дистанционно или в домашних условиях.

Основой серии стандартов ИСО/ИИЭР 11073 служит парадигма управления объектно-ориентированными системами. Данные (измерения, состояния и т. д.) моделируются в виде информационных объектов, которые вызываются и обрабатываются с помощью протокола службы доступа к объектам.

В настоящем стандарте дается определение специализированного прикладного профиля, позволяющего учесть уникальные требования, предъявляемые к персональным медицинскими приборам. В целях определения оптимизированного коммуникационного профиля для данной предметной области этот профиль учитывает идеи серии стандартов ИСО/ИИЭР 11073 и успешный отраслевой опыт, а именно:

- По возможности коммуникационный профиль не конкретизируется для какого-либо определенного транспортного протокола.

- Информационная модель коммуникационного профиля строится на информационной модели предметной области (DIM) ИСО/ИИЭР 11073 и подразумевает оптимизацию при наличии возможности.

- Оптимизированный протокол обмена данными определяется с целью уменьшения размера сообщений, времени создания пакета и издержек синтаксического разбора. Это возможно благодаря меньшей сложности приборов в области персонального здоровья.

- Определения, требуемые для реализации протокола, включены в текст настоящего стандарта без ссылок на другие документы. Данный подход способствует более простому применению настоящего стандарта. В случае расхождений между нормативными положениями и ссылочным документом настоящий стандарт имеет приоритет.

По возможности версии настоящего стандарта полностью обратно совместимы как минимум с двумя предыдущими основными версиями.

Примечание — Ожидается, что любые добавления в DIM или другие соответствующие части серии стандартов ИСО/ИИЭР 11073 будут приняты и учтены в последующих редакциях этих стандартов.

5 Начальные сведения о персональных медицинских приборах (ИИЭР 11073)

5.1 Общие положения

Согласно ISO/IEEE 11073 общая модель системы состоит из трех основных частей: информационная модель предметной области, сервисная модель и коммуникационная модель. Эти три модели используются совместно для представления данных, определения методологий команд и доступа к данным, а также для описания передачи данных от агента к менеджеру. Ввиду тесных связей между этими моделями они кратко описаны соответственно в 5.2, 5.3 и 5.4, так что при ознакомлении с разделами 6, 7 и 8, где они описаны более детально, читателю будут известны основные понятия.

5.2 Информационная модель предметной области (DIM)

Информационная модель предметной области (DIM), подробно описанная в разделе 6, характеризует информацию агента в виде набора объектов. Каждый объект обладает одним или несколькими атрибутами. Атрибуты описывают результаты измерений, передаваемые менеджеру, и элементы, управляющие поведением и сообщающие статус агента.

5.3 Сервисная модель

Сервисная модель, подробно описанная в разделе 7, предусматривает базовые операции доступа к данным, передаваемые между агентом и менеджером для извлечения данных, описанных в DIM. К базовым операциям относятся такие команды, как Get (получить), Set [задать], Action (выполнить действие) и Event Report (передать отчет о событии).

5.4 Коммуникационная модель

Коммуникационная модель, подробно описанная в разделе 8, поддерживает топологию, при которой один или несколько агентов взаимодействуют с одним менеджером с помощью двухточечных соединений. Для каждого двухточечного соединения поведение динамической системы определяется конечным автоматом соединения, определяющим переходы состояний и подсостояний пары «менеджер — агент», в том числе состояний, имеющих отношение к соединению, ассоциации и операциям. Коммуникационная модель также подробно определяет условия входа, выхода и ошибки для соответствующих состояний, а также различные рабочие процедуры передачи результатов измерений. Коммуникационная модель описывает также предположения о поведении нижележащих коммуникационных уровней.

Еще одной функцией коммуникационной модели является преобразование абстрактных данных моделирования (абстрактный синтаксис), используемых в модели DIM, в синтаксис передачи данных, например в двоичные сообщения при помощи правил кодирования медицинских приборов (MDER), посылаемых с использованием коммуникационной модели.

5.5 Соответствие другим стандартам

Может также потребоваться, чтобы устройства, соответствующие требованиям настоящего стандарта, отвечали требованиям других стандартов, специфичных для предметных областей и устройств, которые замещают требования настоящего стандарта в отношении ряда аспектов, среди которых безопасность, надежность и управление рисками. Предполагается, что пользователь настоящего стандарта знаком со всеми остальными подобными применяемыми стандартами и учитывает все спецификации, имеющие более высокий приоритет. Обычно медицинские приборы будут соответствовать требованиям базовых стандартов IEC 60601-1 (2005) + A1 (2012) [B1] в части электрической и механической безопасности, а также требованиям стандартов на конкретные устройства, например, определенным в серии стандартов МЭК 60601-2 [B2]. Аспекты программного обеспечения могут регламентироваться такими стандартами, как МЭК 62304 (2006)/EN 62304 (2006) [B3].

В устройствах, соответствующих требованиям настоящего стандарта, реализуются верхние уровни сетевого программного обеспечения и используются нижние уровни с учетом особенностей приложения. Требования, предъявляемые к характеристикам таких приложений и соответствию, формулируются в других документах и не входят в область применения настоящего стандарта. Кроме того, использование любого медицинского оборудования нуждается в оценке рисков и управлении ими с учетом применения. Релевантными примерами могут служить стандарты ИСО 14971:2007 [B12] и МЭК 80001-1 (2010) [B4]. Требования, связанные с оценкой рисков, управлением рисками и соответствием, не входят в область применения настоящего стандарта.

5.6 Безопасность

Настоящий стандарт не предоставляет описание каких-либо методов обеспечения безопасности обмена данными. Предполагается, что безопасность реализуется с помощью иных средств, например с использованием защищенного канала передачи данных.

6 Модель предметной области персонального медицинского прибора

6.1 Общие положения

В настоящем стандарте персональные медицинские приборы описаны с помощью объектно-ориентированной модели. Для моделирования агента в этой модели определены несколько классов. Модель описывает прибор агента как набор объектов, представляющих источники данных как элементы, которые менеджер может использовать для управления поведением агента, и как механизм, который агент использует для передачи отчетов об изменениях статуса представления агента. Объекты прибора агента обладают атрибутами, представляющими информацию и статус объекта.

Приборы-менеджеры обмениваются данными с объектами прибора агента с помощью хорошо определенных методов (например, GET и SET), определенных в каждом подразделе, описывающем объект. Информация, например измерения, передается от объектов данных агента прибору менеджера с помощью отчетов о событиях.

Информационная модель предметной области персональных медицинских приборов представляет собой объектно-ориентированную модель, определяющую объекты данных агентов, в том числе их атрибуты и методы. Использование объектно-ориентированной информационной модели поддерживает следующее:

- отделение спецификации от реализации с помощью принципа инкапсуляции;
- развитие на основе принципа наследования;
- обратную совместимость с помощью принципа полиморфизма.

Объекты, произведенные от классов, определенных в информационной модели, представляют все данные, которые система агента может передавать системе менеджера с помощью прикладного протокола, определенного настоящим стандартом. Такие данные моделируются в виде атрибутов объектов. Кроме того, информационная модель определяет конкретные службы доступа к данным в форме методов, используемых для обмена данными между системами агента и менеджера. Службы моделируют сообщения прикладного протокола (базовые процедуры доступа к данным), определенные в настоящем стандарте.

Объекты определяют структуру и возможности системы агента. Система менеджера получает доступ к этим объектам, чтобы извлечь данные из системы агента и/или управлять ею. Настоящий стандарт не определяет информационную модель системы менеджера.

Информационная модель представляет собой иерархическую модель, определяющую логическую структуру и возможности персонального медицинского прибора. На верхнем уровне объект системы медицинского прибора (MDS) представляет свойства и службы собственно прибора, независимо от возможностей передачи медицинских данных. Свойства объекта MDS включают в себя атрибуты для идентификации прибора и последующего технического описания и предоставления данных о его состоянии. Данные, специфичные для приложения (например, медицинские данные и результаты измерений), предоставляемые персональным медицинским прибором, моделируются в виде дополнительных информационных объектов, логически содержащихся в объекте MDS. Набор атрибутов объектов вместе с этим отношением вложения описывает конфигурацию и тем самым возможности персонального медицинского прибора.

Обратите внимание, что определения в настоящем стандарте используют объектную ориентацию, чтобы определить информационную модель, однако такая практика не подразумевает использование объектно-ориентированных технологий (например, объектно-ориентированные языки программирования) для реализации настоящего стандарта в конкретном приборе. Модель используется для определения структур данных и методов доступа (сообщений протокола) удобным и согласованным способом. Соответствие этим определениям требуется только на уровне сообщений коммуникационного протокола. В частности, определения, приведенные в настоящем стандарте, оптимизированы для обеспечения простой реализации агентов (например, с помощью предварительно заданных шаблонов передачи). Аналогично реализация прибора менеджера может свободно выбрать архитектуру, использующую информационные объекты или другие ее варианты.

Настоящий стандарт использует информационные классы и объекты, определенные в ИСО/ИИЭР 11073-10201:2004 [B17], однако они следующим образом адаптированы к предметной области коммуникации персональных медицинских приборов:

- определения обязательных, необязательных и условно обязательных атрибутов могут отличаться;
- могут быть определены дополнительные службы объектов;
- могут быть определены дополнительные атрибуты;
- некоторые свойства исходной модели могут не использоваться.

6.2 Использование номенклатуры

Ключевое свойство модели DIM состоит в том, что ссылки на классы объектов и атрибуты осуществляются с помощью номенклатурных кодов, приведенных в ИСО/ИИЭР 11073-10101 [16]. Применение согласованной номенклатуры улучшает интероперабельность, поскольку все реализации поддерживают одинаковое семантическое значение числовых кодов. Использование номенклатурных кодов способствует также международному применению, поскольку сокращено количество строковых значений.

Номенклатура ИСО/ИИЭР 11073 определена в виде набора контекстно-зависимых разделов. Номенклатурный код в каждом контекстно-зависимом разделе определяется 16-битным кодом, который поддерживает до 65 536 независимых терминов для одного раздела. Ссылка на разделы осуществляется с помощью 16-битного кода раздела. Если код раздела номенклатуры определяется контекстом, возможно использование только 16-битного кода. Если контекст не определен или требуется контекстно-независимый код термина, тогда используется 32-битный код, образованный 16-битным кодом раздела и 16-битным кодом термина. Разделы, которые определены в настоящем стандарте и/или ИСО/ИИЭР 11073-10101 [B16], приведены в таблице 1.

Коды терминов с 0xF000 по 0xFFFF в каждом разделе номенклатуры зарезервированы для местных (определяемых производителем) номенклатурных кодов.

Для каждого номенклатурного термина стандарт ИСО/ИИЭР 11073-10101 [B16] определяет систематическое имя, объясняющее этот термин, значение уникального кода и ссылочный идентификатор (ID). Ссылочный идентификатор указывается в форме MDC_XXX_YYY (MDC означает «коммуникацию с медицинским прибором»). В тексте настоящего стандарта номенклатурные термины и номенклатурные коды указываются с помощью ссылочного идентификатора.

Таблица 1 — Разделы номенклатуры

Номер раздела	Категория номенклатуры
0	Не определена
1	Объектно-ориентированная
2	Система диспетчерского управления и сбора данных (SCADA)
3	События
4	Размерности (единицы измерения)
5	Виртуальные атрибуты
6	Группы параметров
7	Участки тела
8	Инфраструктура
9	Формат обмена файлами
10	Расширения для электрокардиограммы (ЭКГ)
11	Расширения для имплантируемого устройства — сердечного — наблюдения (IDCO)
12-127	Зарезервирована
128	Управление течением заболевания с использованием персонального медицинского прибора
129	Здоровье и фитнес с использованием персонального медицинского прибора
130	Одинокое старение с использованием персонального медицинского прибора
131-254	Зарезервирована
255	Коды возврата
256	Внешние номенклатурные ссылки
257-1023	Зарезервирована
1024	Местная
1025-65 535	Зарезервирована

6.3 Определения классов персональных медицинских объектов

6.3.1 Общие положения

Представление информационных объектов персонального медицинского агента и отношений классов показано на рисунке 4 с помощью Унифицированного языка моделирования (UML). Объект, показанный на самом верхнем уровне, представляет информацию и статус объекта MDS (см. 6.3.2). С этим объектом ассоциированы ноль или более числовых массивов считываний в режиме реального времени (real-time sample array, RT-SA), перечислений Enumeration, объектов Scanner или объектов постоянного хранения измерений (persistent metric store, PM-store). С объектом PM-store ассоциированы ноль или более объектов PM-segment, содержащих постоянные измерения. Объекты Numeric, RT-SA и Enumeration произведены от общего базового класса Metric, который содержит общие и разделяемые атрибуты (см. 6.3.3). В общем случае объекты Numeric представляют эпизодические измерения (см. 6.3.4), объекты RT-SA — непрерывные считывания или биосигналы (см. 6.3.5), объекты Enumeration — аннотации событий (см. 6.3.6), а объекты PM-store (см. 6.3.7) в сочетании с объектами PM-segment (см. 6.3.8) — постоянные механизмы хранения измерений, доступ к которым позже получает менеджер. Кроме того, объект Scanner (подробно определенный в 6.3.9) упрощает информирование о передачах данных, инициированных агентом.

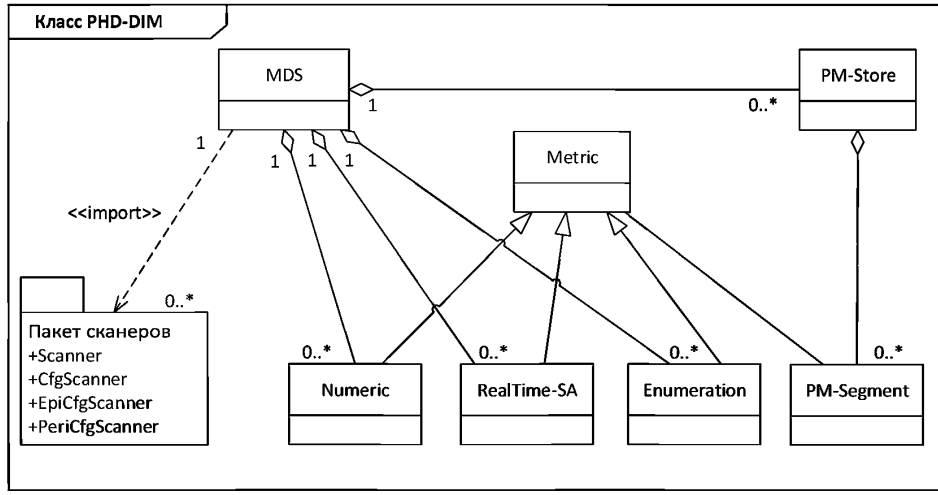


Рисунок 4 — Информационная модель предметной области персонального медицинского прибора

Классы информационной модели предметной области персонального медицинского прибора описаны в подразделах 6.3.2 — 6.3.9. Каждый подраздел использует следующий формат:

- номенклатурный код, используемый для идентификации класса. Данный код используется во время события конфигурирования для сообщения о классе каждого объекта и позволяет менеджеру узнать, является ли указанный объект экземпляром класса Numeric, RT-SA или любого другого класса;
- атрибуты, определяемые классом;
- доступные методы;
- потенциальные события, генерируемые объектами, являющимися экземплярами класса;
- доступные службы, например получение или установка атрибутов.

Каждый тип данных атрибута определяется с помощью Абстрактной синтаксической нотации версии 1 (ACH.1) [B22]. Определения ACH.1 для всех типов данных и форматов обмена представлены в приложении А.

Атрибуты каждого класса определены в таблицах, в которых указаны имя атрибута, ссылочный идентификатор его номенклатуры, тип, описание и квалификаторы. Квалификаторы «условно обязательный», «необязательный» и «обязательный» указывают необходимость реализации атрибута объекта. Условно обязательный атрибут реализуется при условии, указанном в графе «Примечание». Условно обязательные атрибуты должны быть реализованы, если это условие выполнено, и могут быть

реализованы в другом случае. Необязательные атрибуты могут быть реализованы агентом. Обязательные атрибуты должны быть реализованы агентом.

Атрибут дополнительно классифицируется как статический, динамический или наблюдаемый. Значения статических атрибутов не должны изменяться на протяжении срока существования ассоциации. Значение передается в отчете о событии конфигурирования. Значение остается действующим до момента выхода системы из состояния ассоциации. Динамические атрибуты имеют значения, которые могут меняться на протяжении срока существования ассоциации. Значение атрибута необходимо передавать при конфигурировании, а также во время или до того момента времени, когда такое значение потребуется для интерпретации сообщенных результатов наблюдения. Значение может изменяться позже (например, в отчете о событиях данных сегмента или сканера). Значение атрибута остается действующим, пока не будет изменено в следующем отчете о событиях данных сегмента или сканера или пока система не завершит ассоциацию. Наблюдаемые атрибуты имеют значения, которые могут меняться на протяжении срока существования ассоциации. Значение может передаваться в отчете о событиях данных сегмента или сканера. При получении набора значений наблюдаемого атрибута их объединяют с доступной контекстной информацией (например, значениями всех сопутствующих динамических и статических атрибутов), чтобы представить результаты, полученные во время наблюдения. В отличие от значений динамических и статических атрибутов значения наблюдаемого атрибута объединяются с контекстной информацией только однократно (например, значения наблюдаемого атрибута не используются повторно при последующем получении любых новых значений этого атрибута).

В таблицах объектов Metric атрибуты, помеченные как наблюдаемые, могут дополнительно маркироваться как (1) «задаваемые», (2) «вводимые вручную» или (3) «вычисляемые» полностью на основе вводимых вручную и/или задаваемых атрибутов. В этих трех случаях наблюдаемый атрибут должен считаться пригодным для интервала времени с момента первого присвоения значения атрибута до обновления этого или любого другого атрибута его объекта. Предполагается, что без этих настроек атрибут наблюдения действителен только во время его взятия. Агент устанавливает эти флаги в атрибуте Metric-Spec-Small. Типы атрибутов обобщены в таблице 2.

Таблица 2 — Типы атрибутов

Тип атрибута	Возможность предоставления значения	Момент времени, до наступления которого значение атрибута остается действительным
Динамический атрибут может изменяться на протяжении срока существования ассоциации	- По возможности должен отправляться в отчете о событиях конфигураций и должен отправляться в то время или до того момента времени, когда значение потребуется для интерпретации сообщенных результатов наблюдения. - Может передаваться в отчете о событиях данных сегмента или сканера	- Изменяется при последующем обновлении отчета о событиях данных сегмента или сканера. - Система выходит из состояния ассоциации
Наблюдаемый атрибут может измениться на протяжении срока существования ассоциации	- Должен передаваться в отчете о событиях данных сегмента или сканера	- Такие значения объединяются с доступной контекстной информацией (например, со значения всех связанных динамических и статических атрибутов), чтобы представить результаты, полученные во время наблюдения
Статический атрибут не должен изменяться на протяжении срока существования ассоциации	- Должен передаваться в отчете о событиях конфигураций	- Система выходит из состояния ассоциации

Номенклатурный код класса объектов (например, Numeric или RT-SA) передается менеджеру во время конфигурирования, чтобы создать зеркальное представление объекта. Каждый объект обладает атрибутом Handle (дескриптор), который используется для идентификации объекта при выполнении действия (от

и к объекту). Остальные атрибуты используются для представления и передачи информации о физическом приборе и его источниках данных. Доступ к атрибутам и их изменение осуществляются с помощью таких методов, как GET (получить) и SET (задать). Данные передают, используя события (EVENT).

6.3.2 Класс MDS

6.3.2.1 Общие положения

Каждый персональный медицинский прибор-агент определяется объектно-ориентированной моделью, показанной на рисунке 4. Экземпляр объекта верхнего уровня каждого агента создается на основе класса MDS. Каждый агент обладает одним объектом MDS. Этот объект представляет идентификационную информацию и статус агента в своих атрибутах.

6.3.2.2 Идентификация класса MDS

Для идентификации класса MDS используется номенклатурный код MDC_MOC_VMS_MDS_SIMP.

6.3.2.3 Атрибуты класса MDS

Набор атрибутов класса MDS, поддерживаемых для коммуникации с персональным медицинским агентом, описан в таблице 3. Объект MDS должен поддерживать все обязательные атрибуты, но может иметь поднаборы условно обязательных и необязательных атрибутов.

Таблица 3 — Атрибуты класса MDS

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Handle	MDC_ATTR_ID_ HANDLE	HANDLE	Атрибут Handle представляет собой ссылочный идентификатор этого объекта. Значение атрибута Handle объекта MDS должно равняться 0	Обязательный, статический
System-Type	MDC_ATTR_ SYS_TYPE	TYPE	Данный атрибут указывает тип агента согласно определению в номенклатуре (например, весы). Значения должны заимствоваться из раздела nom-part-object и подраздела MD-Gen (общий медицинский прибор), описанных в стандарте ISO/ IEEE 11073-10101 [B16]. Должен присутствовать один и только один атрибут System-Type или System-Type-Spec-List	Условно обязательный, статический
System-Model	MDC_ATTR_ID_ MODEL	SystemModel	Данный атрибут указывает производителя и номер модели прибора агента	Обязательный, статический
System-Id	MDC_ATTR_ SYS_ID	OCTET STRING	Данный атрибут имеет тип IEEE EUI-64, состоящий из 24-битного идентификатора, уникального в пределах организации (OUI), и 40-битного идентификатора, определяемого производителем. Значение OUI должно назначаться комитетом IEEE Registration Authority (http://standards.ieee.org/regauth/index.html) и использоваться согласно требованиям стандарта IEEE 802-2014[10]	Обязательный, статический
Dev-Configuration-Id	MDC_ATTR_ DEV_CONFIG_ID	ConfigId	Данный атрибут определяет идентификацию конфигурации прибора агента. Атрибут Dev-Configuration-Id статичен на протяжении срока существования ассоциации. Обычно он передается во время процедуры ассоциации. Менеджер может получить этот атрибут во время работы с помощью метода GET. Если этот атрибут запрашивается до того, как агент и менеджер согласуют конфигурацию, агент должен вернуть идентификатор конфигурации, предлагаемый в момент запроса. Дополнительную информацию об этом атрибуте см. в 8.8	Обязательный, статический

[10] Дополнительные сведения о ссылках см. в разделе 2.

Продолжение таблицы 3

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Attribute-Value-Map	MDC_ATTR_ ATTRIBUTE_ VAL_MAP	AttrValMap	Данный атрибут определяет атрибуты, передаваемые в сообщениях фиксированного формата об обновлении данных (дополнительные сведения см. в 7.4.5). Если при передаче отчета о динамических данных для объекта агент использует сообщения с фиксированным форматом значений, этот атрибут необходимо задать до отправки такого отчета. Значение этого атрибута может измениться в интервале времени между отправками таких отчетов о событиях, поэтому значение атрибута должно отправляться в отчете о событии, если изменение происходит в состоянии ассоциации	Условно обязательный, динамический
Production-Specification	MDC_ATTR_ID_ PROD_SPECN	Production-Spec	Данный атрибут указывает версии компонента, серийные номера и т. д. в формате, заданном производителем.	Необязательный, статический
Mds-Time-Info	MDC_ATTR_ MDS_TIME_INFO	MdsTimeInfo	Данный атрибут указывает возможности обработки времени и статус MDS. Использование этого атрибута требуется в случае поддержки синхронизации или настройки часов	Условно обязательный, динамический
Date-and-Time	MDC_ATTR_ TIME_ABS	AbsoluteTime	Данный атрибут определяет дату и время на часах агента с точностью 1/100 секунды (при наличии такой возможности). Дополнительную информацию об этом атрибуте см. в 8.12. Если в любом сообщении агента указано абсолютное время, этому атрибуту должно присваиваться текущее значение абсолютного времени. Если этот атрибут используется, то атрибут Base-Offset-Time не должен использоваться	Условно обязательный, наблюдаемый
Base-Offset-Time	MDC_ATTR_ TIME_BO	BaseOffset-Time	Данный атрибут определяет дату и время на часах агента в форме базового времени со смещением местного времени в минутах. Дополнительную информацию об этом атрибуте см. в 8.12. Если в любом сообщении агента указано базовое время со смещением (BaseOffsetTime), то должно быть указано текущее значение базового времени в этом атрибуте. Если этот атрибут используется, атрибут Date-and-Time не должен использоваться	Условно обязательный, наблюдаемый
Relative-Time	MDC_ATTR_ TIME_REL	RelativeTime	Агент, передающий относительное время (RelativeTime) в любом сообщении, должен указать его текущее значение в этом атрибуте	Условно обязательный, наблюдаемый
HiRes-Relative-Time	MDC_ATTR_ TIME_REL_HI_ RES	HighResRelativeTime	Агент, передающий относительное время с высоким разрешеним (HighResRelative-Time) в любом сообщении, должен указать текущее значение этого времени в данном атрибуте	Условно обязательный, наблюдаемый

Продолжение таблицы 3

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Date-and-Time-Adjustment	MDC_ATTR_TIME_ABS_ADJUST	AbsoluteTime-Adjust	В данном атрибуте сообщаются любые изменения даты и времени, которые вносятся при настройке часов пользователем или при возникновении таких событий, как переход на летнее время. Используется только в отчетах о событиях. Если запрашивается с помощью команды Get MDS Object, данное значение должно отсутствовать или равняться нулю. Если агент осуществляет настройку времени и даты, данный атрибут используется в отчете о событии, чтобы сообщить о таком изменении	Условно обязательный, наблюдаемый
Power-Status	MDC_ATTR_POWER_STAT	PowerStatus	В данном атрибуте сообщается, осуществляется ли электропитание от батареи или электросети, а также указывается состояние заряда батареи	Необязательный, динамический
Battery-Level	MDC_ATTR_VAL_BATT_CHARGE	INT-U16	Данный атрибут указывает процент оставшейся емкости батареи (не определен, если значение > 100)	Необязательный, наблюдаемый
Remaining-Battery-Time	MDC_ATTR_TIME_BATT_REMAIN	BatMeasure	Данный атрибут представляет прогнозируемый объем оставшегося времени работы от батареи. Единицы измерения должны содержать значение MDC_DIM_MIN, MDC_DIM_HR или MDC_DIM_DAY для указания минут, часов или дней соответственно	Необязательный, наблюдаемый
Reg-Cert-Data-List	MDC_REG_CERT_DATA_LIST	RegCertDataList	В данном атрибуте с информационными целями перечислены различные сведения о соответствии регуляторным и/или сертификационным нормам. Заявления о соответствии реализации (см. раздел 9) имеют приоритет над этим атрибутом и являются документами, имеющими юридическую силу	Необязательный, статический
System-Type-Spec-List	MDC_ATTR_SYS_TYPE_SPEC_LIST	TypeVerList	Данный атрибут указывает тип (типы) агента согласно номенклатуре (например, весы). Значения должны заимствоваться из подраздела DEVspec раздела nom-part-infrastruct, приведенного в ISO/IEEE 11073-10101 [B16] и ссылаться на специализации ISO/IEEE 11073-104zz. Если агент не соответствует ни одной из специализаций, список должен остаться пустым. Если агент соответствует профилю специализации, этот список должен содержать специализации и номенклатурные значения профиля. Данный список должен также содержать версии специализаций/профилей. Должен присутствовать только один из атрибутов, System-Type или System-Type-Spec-List. Данный атрибут должен присутствовать у многофункциональных агентов	Условно обязательный, статический

Окончание таблицы 3

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Confirm-Timeout	MDC_ATTR_ CONFIRM_TIME-OUT	RelativeTime	Данный информационный атрибут тайм-аута определяет минимальное время, в течение которого агент должен ожидать ответного сообщения от менеджера после передачи ему сообщения вызова подтверждаемого отчета о событиях, прежде чем перейти в состояние «Не ассоциирован». Атрибут является информационным и используется для улучшения работы менеджера. В случае предоставления этого атрибута необходимо обеспечить его соответствие фактическому значению времени ожидания, которое используется агентом для подтверждаемого отчета о событии, сгенерированного объектом MDS. Данный атрибут является информационным для менеджера в том смысле, что менеджер не использует этот атрибут при фактической реализации протокола (например, менеджер не инициирует тайм-аут для переданного агентом подтверждаемого отчета о событии). Однако менеджер может пожелать использовать данную информацию для назначения приоритета обработки сообщения от агента с «коротким» тайм-аутом по отношению к сообщению от агента с «длинным» тайм-аутом	Необязательный, динамический
Trans-port-Timeout	MDC_ATTR_ TRANSPORT_ TIMEOUT	RelativeTime	Данный тайм-аут определяет минимальное время, в течение которого менеджер должен ожидать ответного сообщения от агента после отправки сообщения через базовый транспортный уровень, прежде чем перейти в состояние «Разъединен». Для любого агента, использующего транспортный протокол, способный вызвать длительную задержку, настоятельно рекомендуется добавить этот атрибут в список параметров структуры PhdAsso-ciationInformation, чтобы менеджер мог подготовиться к длительной задержке в состоянии «Конфигурирующий»	Необязательный, динамический

Обратите внимание, что атрибут System-Type не зависит от аналогично названного поля System-Type структуры PhdAssociationInformation, описывающей запрос ассоциации.

Атрибуты MDS обеспечивают представление на аппаратном уровне и не зависят от конкретной предложенной или принятой конфигурации. Например, атрибут System-Type-Spec-List или Reg-Cert-Data-List предоставляет все возможности, которые способен предложить прибор. Текущая конфигурация может предоставить или не предоставить все перечисленное в этом атрибуте.

Типы данных атрибутов определены в приложении А.

6.3.2.4 Методы объектов MDS

Таблица 4 указывает методы (действия), доступные для объекта MDS. Данные методы вызываются с помощью службы ACTION. Графа «Метод/Действие» таблицы 4 содержит имя метода. Графа «Режим» указывает, может ли метод вызываться как неподтверждаемое действие (например, roiv-cmip-action из А.10.2) или подтверждаемое действие (например, roiv-cmip-confirmed-action). Столбец «Тип действия» содержит номенклатурный идентификатор для использования в поле action-type запроса действия и ответа на него (см. А.10.6). Графа «action-info-args» определяет ассоциированную структуру данных поля запроса action-info-args, используемую в сообщении действия. Графа «Поле action-info-args результата» указывает структуру, используемую в поле action-info-args ответа.

Таблица 4 — Методы объектов MDS

Метод/действие	Режим	Тип действия	Поле action-info-args	Поле action-info-args результата
MDS-Data-Request	Подтверждаемый	MDC_ACT_DATA_REQUEST	DataRequest	DataResponse
Set-Time	Подтверждаемый	MDC_ACT_SET_TIME	SetTimeInvoke	Нет
Set-Base-Offset-Time	Подтверждаемый	MDC_ACT_SET_BO_TIME	SetBOTimeInvoke	Нет

- MDS-Data-Request

Данный метод позволяет системе менеджера разрешить или запретить передачу результатов измерений со стороны агента (описание см. в 8.9.3.3.3).

- Set-Time

Данный метод позволяет системе менеджера настроить часы реального времени (RTC) с помощью абсолютного времени. Агент указывает, допустима ли команда Set-Time, используя бит mds-time-capab-set-clock в атрибуте Mds-Time-Info (см. таблицу 3). Агент, поддерживающий Set-Time, должен возвратить параметр rors-cmip-confirmed-action, однако в этом ответе поле action-info-args пустое. Агент, не поддерживающий Set-Time, должен возвратить ошибку no-such-action (roer).

- Set-Base-Offset-Time

Данный атрибут позволяет системе менеджера настроить часы реального времени с помощью базового времени и смещения в минутах по местному времени. Агент указывает, допустима ли команда Set-Base-Offset-Time, используя бит mds-time-capab-set-clock в атрибуте Mds-Time-Info (см. таблицу 3). Агент, поддерживающий Set-Base-Offset-Time, должен возвратить параметр rors-cmip-confirmed-action, однако в этом ответе поле action-info-args пустое. Агент, не поддерживающий Set-Base-Offset-Time, должен возвратить ошибку no-such-action (roer). Если поле секунд базового времени и поле дробной части секунд базового времени заданы равными 0x0 в аргументах действия Set-Base-Offset-Time (эти значения не определены в сетевом протоколе службы времени (NTP)), должно задаваться только смещение по местному времени. Если базовое время (поле секунд) согласуется с всемирным координированным временем (UTC) (с точностью, соответствующей применению), это должно быть обозначено, задав бит mds-time-state-bo-time-utc-aligned в атрибуте Mds-Time-Info.

Агент может поддерживать только абсолютное или базовое время, но не оба одновременно.

6.3.2.5 События объектов MDS

Таблица 5 содержит сведения о потенциальных событиях, передаваемых объектом MDS. Менеджер должен поддерживать все методы, указанные в таблице 5. Если объект MDS использует подтверждаемый отчет о событии, то в любой момент времени допустимо получение от этого объекта не более одного неквитированного подтверждаемого отчета о событии.

Таблица 5 — События объектов MDS

Событие	Режим	Тип события	Параметр информации о событии	Event-reply-info
MDS-Configuration-Event	Подтверждаемый	MDC_NOTI_CONFIG	ConfigReport	ConfigReportRsp
MDS-Dynamic-Data-Update-Var	Подтверждаемый или неподтверждаемый	MDC_NOTI_SCAN_REPORT_VAR	ScanReportInfoVar	—
MDS-Dynamic-Data-Update-Fixed	Подтверждаемый или неподтверждаемый	MDC_NOTI_SCAN_REPORT_FIXED	ScanReportInfoFixed	—
MDS-Dynamic-Data-Update-MP-Var	Подтверждаемый или неподтверждаемый	MDC_NOTI_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	—
MDS-Dynamic-Data-Update-MP-Fixed	Подтверждаемый или неподтверждаемый	MDC_NOTI_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	—

- MDS-Configuration-Event

Данное событие передается агентом в состоянии «Конфигурирующий», если менеджер еще не определил конфигурацию агента на основе предыдущих ассоциаций. Событие предоставляет статическую информацию о поддерживаемых измерительных возможностях агента.

- MDS-Dynamic-Data-Update-Var

Данное событие предоставляет динамические данные (обычно измерения) от агента для некоторых или всех объектов, поддерживаемых агентом. Данные о сообщаемых объектах передают, используя общий список атрибутов в переменном формате (дополнительные сведения о форматах отчетов о событиях см. в 7.4.5). Событие инициируется с помощью запроса MDS-Data-Request от системы менеджера или передается агентом как незапрашиваемое сообщение.

Информация об управлении активацией и/или периодом передачи данных у агентов, поддерживающих инициированную менеджером передачу результатов измерений, представлена в 8.9.3.3.3. Информация об ограниченном управлении со стороны менеджера агентами, не поддерживающими инициируемую менеджером передачу результатов измерений, представлена в 8.9.3.3.2.

- MDS-Dynamic-Data-Update-Fixed

Данное событие предоставляет динамические данные (обычно результаты измерений) от агента для некоторых или всех объектов результатов измерений либо объекта MDS, поддерживаемых агентом. Данные сообщаются в фиксированном формате, определяемом атрибутом Attribute-Value-Map для сообщаемых объектов результатов измерений или объекта MDS (дополнительные сведения о форматах отчетов о событиях см. в 7.4.5). Событие инициируется запросом MDS-Data-Request от системы менеджера (т.е. передача результатов измерений, инициируемая менеджером) или передается агентом как незапрашиваемое сообщение (т.е. передача результатов измерений, инициируемая агентом). Информация об управлении активацией и/или периодом передачи данных у агентов, поддерживающих инициированную менеджером передачу результатов измерений, представлена в 8.9.3.3.3. Информация об ограниченном управлении со стороны менеджера агентами, не поддерживающими инициируемую менеджером передачу результатов измерений, представлена в 8.9.3.3.2.

- MDS-Dynamic-Data-Update-MP-Var

Данное событие аналогично MDS-Dynamic-Data-Update-Var, но позволяет использовать информацию о нескольких лицах.

- MDS-Dynamic-Data-Update-MP-Fixed

Данное событие аналогично MDS-Dynamic-Data-Update-Fixed, но позволяет использовать информацию о нескольких лицах.

6.3.2.6 Прочие службы MDS

6.3.2.6.1 Служба GET

Для извлечения значений всех реализованных атрибутов объектов MDS любой агент, поддерживающий двустороннюю коммуникацию, должен поддерживать службу GET. Служба GET должна вызываться менеджером сразу после перехода в состояние «Конфигурирующий»/«Передающий GetMDS». Менеджер должен ожидать ответ на GET перед вызовом каких-либо действий. Ожидание ответа на GET позволяет менеджеру определить, нуждается ли агент в настройке времени. Агент, запросивший настройку своего времени, должен ожидать действие Set-Time перед переходом в состояние «Выполнение».

Служба GET может вызываться, когда агент находится в подсостоянии «Конфигурирующий»/«ожидающий MDS», «Конфигурирующий»/«ожидающий SetTime» или «Выполнение».

За исключением атрибута Date-and-Time-Adjustment, если менеджер не имеет текущее значение необходимого атрибута MDS, должна использоваться служба GET. Агент может также отправлять отчеты о событиях сканера, предоставляющие менеджеру обновления текущих значений атрибутов, однако такое поведение агента не обязательно, кроме исключений, описанных для attribute-value-map и date-and-time-adjustment в таблице 3.

Менеджер может запросить атрибуты объекта MDS агента. В этом случае менеджер должен отправить команду «Remote Operation Invoke | Get» (см. описание roiv-cmip-get в А.10.2) с зарезервированным нулевым значением дескриптора. Агент должен сообщить менеджеру свои атрибуты объекта MDS, используя команду ответа «Remote Operation Response | Get» (см. описание rors-cmip-get в А.10.2). В ответ на команду Get MDS Object возвращаются только атрибуты, реализованные агентом. Подробное описание операции GET см. в 8.9.3.2.

Примечание — Некоторые требования (например, агент должен запрашивать у менеджера настройку своего времени или передавать временно хранимые данные в фиксированном формате вместе с Date-and-Time-Adjustment) вынуждают менеджера запрашивать от агента объект MDS. Практика также показала, что запросы, которые уже переданы прибору агента перед отправкой запроса GET после AARE accepted-unknown-config (обязы-

вают агента асинхронно обрабатывать потенциально большие отчеты о конфигурации и ответы на GET), зачастую служили причиной тайм-аутов и прочих проблем. Текущие руководства по отношению к запросам GET сформулированы с целью устранения этих проблем.

6.3.2.6.2 Служба SET

На сегодняшний день служба MDS SET, определенная в настоящем стандарте, не используется, однако могут существовать специализации приборов, определенные в стандартах серии ISO/IEEE 11073-104zz, или местные определения, которые ее задействуют.

6.3.3 Класс Metric

6.3.3.1 Общие положения

Класс Metric представляет собой базовый класс для всех объектов, представляющих данные измерений, статуса и контекста. Класс Metric не имеет экземпляров и поэтому никогда не является частью конфигурации агента. В качестве базового этот класс определяет все атрибуты, методы, события и службы, которые являются общими для всех объектов, представляющих измерения.

6.3.3.2 Идентификация класса Metric

Для идентификации класса Metric используется номенклатурный код MDC_MOC_VMO_METRIC. Данный номенклатурный код не используется при реализации агента или менеджера, поскольку класс Metric представляет собой базовый класс для других классов.

6.3.3.3 Атрибуты класса Metric

Таблица 6 содержит описание набора атрибутов класса Metric, поддерживаемых для обмена данными с персональными медицинскими приборами и наследуемых всеми классами, произведенными от Metric.

Таблица 6 — Атрибуты класса Metric

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут Handle представляет собой ссылочный идентификатор этого объекта. Каждый объект должен иметь уникальный идентификатор, присвоенный агентом. Атрибут Handle идентифицирует объект в отчетах о событиях, передаваемых менеджеру	Обязательный, статический
Type	MDC_ATTR_ID_TYPE	TYPE	Данный атрибут определяет специфичный статический тип этого объекта в соответствии с номенклатурой (например, частота пульса для специфичного экземпляра объекта Numeric). Атрибут Type содержит идентификаторы раздела номенклатуры и кода термина для контекстно-свободной расширяемой идентификации	Обязательный, статический
Supplemental-Types	MDC_ATTR_SUPPLEMENTAL_TYPES	Supplemental-TypeList	Данный атрибут может использоваться для передачи дополнительной информации об объекте помимо атрибутов Type и Metric-Id. Дополнительная информация охватывает различные условия, например местоположение датчика или скорость реагирования объекта на изменения. Специализации приборов определяют предполагаемое применение данного атрибута. Например, IEEE 11073-10471™ [B11] определяет номенклатуру расположения, указывая местоположение датчика в доме, а ISO/IEEE 11073-10404 [B18] указывает три дополнительных типа для быстрого ответа, медленного ответа и выборочной проверки частоты пульса или насыщения крови кислородом	Необязательный, динамический

Продолжение таблицы 6

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Metric-Spec-Small	MDC_ATTR_ METRIC_SPEC_ SMALL	MetricSpecSmall	Атрибут «Metric-Spec-Small» описывает характеристики измерений. Данный динамический атрибут позволяет агенту изменить значения битов до отправки первых результатов наблюдений (например, настроить стандартную конфигурацию). После отправки результатов наблюдения агент не должен изменять значение этого атрибута	Обязательный, динамический
Metric-Structure-Small	MDC_ATTR_ METRIC_ STRUCT_SMALL	MetricStructureSmall	Этот атрибут описывает структуру результатов измерений. В случае отсутствия этого атрибута менеджер должен предполагать, что MetricStructureSmall := {ms-struct-simple, 0}. Данный динамический атрибут позволяет агенту изменить значения параметров до отправки первых результатов наблюдений (например, настроить стандартную конфигурацию). После отправки результатов наблюдения агент не должен изменять значение этого атрибута	Необязательный, динамический
Measurement-Status	MDC_ATTR_ MSMT_STAT	MeasurementStatus	Данный атрибут указывает допустимость конкретного значения или набора считываний. Если объект Scanner предоставляет RTSA и существует вероятность того, что от Scanner может потребоваться отчет об отсутствующих результатах измерений (см. 6.3.9.2), этот атрибут должен присутствовать	Условно обязательный, наблюдаемый
Metric-Id	MDC_ATTR_ID_ PHYSIO	OID-Type	Данный атрибут может использоваться для хранения идентификационной информации, которая более специфична по сравнению с общим идентификатором ID в атрибуте Type. Если атрибуту Metric-Id-Partition присвоено значение, он определяет номенклатурный раздел для атрибута Metric-Id. Иначе OID-Type заимствуется из номенклатурного раздела, указанного в поле раздела атрибута Type. Данный атрибут необходим только в случае изменения идентификационной информации во время работы, когда атрибут Type не обеспечивает полную идентификацию. Например, если атрибут Type содержит общий код температуры, код (MDC_TEMP), данный атрибут может сообщить специфичную, но изменяющую идентификацию оральной температуры MDC_TEMP_ ORAL или ректальной температуры MDC_TEMP_RECT. Должен присутствовать только один из атрибутов, Metric-Id или Metric-Id-List	Необязательный, динамический

Продолжение таблицы 6

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Metric-Id-List	MDC_ATTR_ID_PHYSIO_LIST	MetricIdList	Данный атрибут должен использоваться для составного наблюдаемого значения и непосредственно не включает в себя Metric-Id (например, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value), чтобы обеспечить возможность индивидуальной идентификации элементов в списке наблюдаемых значений. Порядок элементов Metric-Id-List должен соответствовать порядку элементов в составном наблюдаемом значении. Должен присутствовать только один из атрибутов, Metric-Id или Metric-Id-List	Условно обязательный, динамический
Metric-Id-Partition	MDC_ATTR_METRIC_ID_PART	NomPartition	Данный атрибут может использоваться для определения раздела, из которого взяты номенклатурные термины для Metric-Id или Metric-Id-List. Если атрибут отсутствует, раздел совпадает с номенклатурным разделом, указанным в поле раздела атрибута Type	Необязательный, динамический
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	Данный атрибут определяет номенклатурный код единиц измерения, взятый из раздела nom-part-dim (например, MDC_DIM_KILO_G). Префиксы единиц измерения должны генерироваться согласно стандартам IEEE 1541-2002 и ISO/IEC 80000-13:2008	Необязательный, динамический
Attribute-Value-Map	MDC_ATTR_ATTRIBUTE_VAL_MAP	AttrValMap	Данный атрибут определяет атрибуты, передаваемые в сообщениях фиксированного формата об изменении данных. Если для отчета о динамических результатах измерений, хранящихся в объекте, агент использует сообщения с фиксированным форматом, то этот атрибут должен быть задан до отправки такого отчета. Значение этого атрибута может меняться между передачами отчетов о событиях	Условно обязательный, динамический
Source-Handle-Reference	MDC_ATTR_SOURCE_HANDLE_REF	HANDLE	Данный атрибут устанавливает отношение этого экземпляра объекта к объекту-источнику (например, пульс ссылается на источник SpO2). Данный атрибут используется в тех случаях, когда необходимо моделировать явное соотношение между экземплярами объектов для определения зависимостей. Использование этого атрибута определяется специализациями приборов. Объект Metric может содержать только один из атрибутов, Source-Handle-Reference или Source-Handle-Reference-List	Необязательный, динамический

Продолжение таблицы 6

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Source-Handle-Reference-List	MDC_ATTR_SOURCE_HANDLE_REF_LIST	HANDLEList	Данный атрибут устанавливает отношение этого экземпляра объекта к нескольким объектам-источникам (например, индекс массы тела (BMI) ссылается на источники рост и вес). Данный атрибут используется в тех случаях, когда необходимо моделировать явное соотношение между экземплярами объекта для определения зависимостей. Использование этого атрибута определяется специализациями приборов. Объект Metric может содержать только один из атрибутов, Source-Handle-Reference или Source-Handle-Reference-List	Необязательный, динамический
Label-String	MDC_ATTR_ID_LABEL_STRING	OCTET STRING	Данный атрибут определяет текстовое представление атрибута Type в печатаемых символах кодировки ASCII. Значение этого атрибута задается по полному усмотрению производителя агента. Потенциально может оказаться полезным для менеджера в качестве изображаемой строки или при принятии решения о выборе поведения, когда он не может понять значение MDC_ATTR_ID_TYPE, отправленное агентом	Необязательный, динамический
Unit-LabelString	MDC_ATTR_UNIT_LABEL_STRING	OCTET STRING	Данный атрибут определяет текстовое представление размерности Unit-Code в печатаемых символах кодировки ASCII. Значение этого атрибута задается по полному усмотрению производителя агента. Потенциально может оказаться полезным для менеджера в качестве изображаемой строки или при принятии решения о выборе поведения, когда он не может понять значение MDC_ATTR_UNIT_CODE, отправленное агентом	Необязательный, динамический
Absolute-Time-Stamp	MDC_ATTR_TIME_STAMP_ABS	AbsoluteTime	Данный атрибут определяет дату и время наблюдения с точностью до 1/100 секунды (если возможно). Дополнительную информацию об этом атрибуте см. в 8.12. Агент, хранящий данные (в объекте PM-store или как «временно хранимые измерения»), должен сопоставлять с этими данными одну и только одну метку времени (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp или HiRes-Time-Stamp). Если этот атрибут используется, атрибут Base-Offset-Time-Stamp не должен использоваться	Условно обязательный, наблюдаемый

Окончание таблицы 6

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Base-Offset-Time-Stamp	MDC_ATTR_TIME_STAMP_BO	BaseOffsetTime	Данный атрибут определяет базовые дату и время наблюдения, а также смещение в минутах по местному времени. Дополнительную информацию об этом атрибуте см. в 8.12. Агент, хранящий данные (в объекте PM-store или как «временно хранимые измерения»), должен сопоставлять с данными одну и только одну метку времени (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp или HiRes-Time-Stamp). Если этот атрибут используется, атрибут Absolute-Time-Stamp не должен использоваться	Условно обязательный, наблюдаемый
Relative-Time-Stamp	MDC_ATTR_TIME_STAMP_REL	RelativeTime	Данный атрибут определяет время наблюдения (метку времени в формате относительного времени/числа тактов часов согласно типу данных RelativeTime). Агент, хранящий данные, должен сопоставлять с данными одну и только одну метку времени (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp или HiRes-Time-Stamp)	Условно обязательный, наблюдаемый
HiRes-Time-Stamp	MDC_ATTR_TIME_STAMP_REL_HI_RES	HighResRelative-Time	Данный атрибут определяет время наблюдения (метку времени в формате высокоточного относительного времени/числа тактов часов согласно типу данных HighResRelativeTime). Агент, хранящий данные, должен сопоставлять с данными одну и только одну метку времени (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp или HiRes-Time-Stamp)	Условно обязательный, наблюдаемый
Measure-Active-Period	MDC_ATTR_TIME_PD_MSMT_ACTIVE	FLOAT-Type	Данный атрибут определяет длительность периода наблюдения в секундах. По умолчанию этот атрибут считается динамическим, однако стандарты ISO/IEEE 11073-104zz могут определять его в качестве наблюдаемого с учетом потребностей варианта использования	Необязательный, динамический или наблюдаемый

6.3.3.4 Методы объектов Metric

В настоящем стандарте отсутствуют определения методов объектов Metric, однако они могут существовать в специализациях приборов ISO/IEEE 11073-104zz или в местных документах.

6.3.3.5 События объектов Metric

Объекты, которые произведены от класса Metric, не сообщают напрямую о результатах наблюдений; эти результаты передаются с помощью другого объекта, например объекта системы MDS, объекта Scanner или объекта PM-store.

6.3.3.6 Прочие службы класса Metric

На сегодняшний день службы SET или GET класса Metric, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ISO/IEEE 11073-104zz, или местные определения, которые их задействуют.

6.3.4 Класс Numeric

6.3.4.1 Общие положения

Экземпляр класса Numeric представляет числовой результат измерения. Значения объекта Numeric передаются от агента к менеджеру с помощью службы EVENT REPORT (см. 7.3). Данный класс произведен от базового класса Metric.

6.3.4.2 Идентификация класса Numeric

Для идентификации класса Numeric используется номенклатурный код MDC_MOC_VMO_METRIC_NU.

6.3.4.3 Атрибуты класса Numeric

Таблица 7 содержит описание набора атрибутов класса Numeric, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 7 — Атрибуты класса Numeric

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Simple-Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS_SIMP	SimpleNuObsValue	Данный атрибут определяет числовое наблюдаемое значение объекта без какой-либо вложенной информации о статусе, присутствующей в арибуте Nu-Observed-Value. Должен присутствовать один и только один из следующих атрибутов: Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value	Условно обязательный, наблюдаемый
Compound-Simple-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS_SIMP	SimpleNuObsValueCmp	Данный атрибут представляет массив значений атрибута Simple-Nu-Observed-Values. Должен присутствовать один и только один из следующих атрибутов: Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value	Условно обязательный, наблюдаемый
Basic-Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS_BASIC	BasicNuObsValue	Данный атрибут определяет числовое наблюдаемое значение объекта без какой-либо вложенной информации о статусе, но с меньшим числовым представлением по сравнению с Simple-Nu-Observed-Value. Должен присутствовать один и только один из следующих атрибутов: Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value	Условно обязательный, наблюдаемый

Окончание таблицы 7

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Compound-Basic-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS_BASIC	BasicNuObsValueCmp	Данный атрибут представляет массив значений атрибута Basic-Nu-Observed-Values. Должен присутствовать один и только один из следующих атрибутов: Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value	Условно обязательный, наблюдаемый
Nu-Observed-Value	MDC_ATTR_NU_VAL_OBS	NuObsValue	Данный атрибут определяет числовое наблюдаемое значение объекта и объединяет его с информацией о состоянии и единице измерения. Используется, когда статус/единица измерения непостоянны и всегда представляются вместе со значением. Должен присутствовать один и только один из следующих атрибутов: Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value	Условно обязательный, наблюдаемый
Compound-Nu-Observed-Value	MDC_ATTR_NU_CMPD_VAL_OBS	NuObsValueCmp	Данный атрибут представляет собой массив сочетаний значения, статуса и единицы измерения. Данный атрибут доступен для использования только в отчетах о событиях, имеющих переменный формат. Должен присутствовать один и только один из следующих атрибутов: Simple-Nu-Observed-Value, Basic-Nu-Observed-Value, Nu-Observed-Value, Compond-Nu-Observed-Value, Compound-Simple-Nu-Observed-Value или Compound-Basic-Nu-Observed-Value	Условно обязательный, наблюдаемый
Accuracy	MDC_ATTR_NU_ACCUR_MSMT	FLOAT-Type	Данный атрибут имеет абсолютное значение и определяет максимальное отклонение фактического значения от сообщенного наблюдаемого значения (если оно может быть указано) во всем диапазоне измерений. Выражается в тех же единицах измерения, что и результаты наблюдений. При передаче результатов измерений с определенной точностью сообщаемое значение должно обладать определенным числом разрядов (см. F.8), достаточным для представления такой точности	Необязательный, статический

Атрибуты Compound-Simple-Nu-Obs-Value, Compound-Basic-Nu-Obs-Value и Compound-Nu-Observed-Value представляют концепцию списка для наблюдаемых значений. Данную концепцию необходимо использовать в тех случаях, когда между отдельными наблюдаемыми значениями задана сильная взаимосвязь, которая может зависеть от номенклатуры и/или применения. Составные наблюдаемые значения используют общий контекст статической атрибуции, кроме идентификации элемен-

тов. Примером этого может служить применение кровяного давления, когда номенклатурный базовый термин обозначает «кровяное давление», а более специфические термины соответствуют «систолическому кровяному давлению», «диастолическому кровяному давлению» и «среднему кровяному давлению». Соответствующий DIM должен содержать только один экземпляр объекта Numeric, который будет использовать один из форматов составных числовых наблюдаемых значений для представления «систолической», «диастолической» и «средней» частей «кровяного давления».

6.3.4.4 Методы объектов Numeric

В настоящем стандарте отсутствуют определения методов объектов Numeric, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

6.3.4.5 События объектов Numeric

В настоящем стандарте события объектов Numeric не определены.

6.3.4.6 Прочие службы класса Numeric

На сегодняшний день службы SET или GET класса Numeric, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые их задействуют.

6.3.5 Класс RT-SA

6.3.5.1 Общие положения

Экземпляр класса RT-SA представляет измерение формы сигнала. Значения объекта RT-SA передаются от агента к менеджеру с помощью службы EVENT REPORT (см. 7.3). Данный класс произведен от базового класса Metric.

6.3.5.2 Идентификация класса RT-SA

Для идентификации класса RT-SA используется номенклатурный код MDC_MOC_VMO_METRIC_SA_RT.

6.3.5.3 Атрибуты класса RT-SA

Таблица 8 содержит описание набора атрибутов класса RT-SA, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 8 — Атрибуты класса RT-SA

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Sample-Period	MDC_ATTR_ TIME_PD_SAMP	RelativeTime	Данный атрибут определяет интервал времени между последовательными считываниями, измеряемый в 1/8 мс. Следовательно, 8000 = 1 с	Обязательный, статический
Simple-Sa-Observed-Value	MDC_ATTR_ SIMP_SA_OBS_ VAL	OCTET STRING	Данный байтовый массив содержит считывания, которые сообщаются агентом в формате, описанном атрибутами Sa-Specification и Scale-and-Range-Specification. Длина должна быть четной (при необходимости с дополняющими байтами в конце). Атрибут Sa-Specification определяет фактическое количество используемых байтов	Обязательный, наблюдаемый
Scale-and-Range-Specification	MDC_ATTR_ SCALE_SPECN_ I8 MDC_ATTR_ SCALE_SPECN_ I16 MDC_ATTR_ SCALE_SPECN_ I32	ScaleRangeSpec8 ScaleRangeSpec16 ScaleRangeSpec32	Данный атрибут определяет отображение считываний на фактические значения, а также диапазон измерений. Тип зависит от разрешающей способности считываний (поле sample-size в поле sample-type атрибута Sa-Specification). Необходимо указать только одну из трех спецификаций	Обязательный, динамический
Sa-Specification	MDC_ATTR_SA_ SPECN	SaSpec	Данный атрибут описывает массив и типы считываний	Обязательный, статический

Характеристики объекта RT-SA можно получить, анализируя атрибут Sa-Specification. Данный атрибут определяет число элементов в массиве и их размер (подробнее см. в А.3.4).

Информация об управлении активацией и/или периодом передачи данных у агентов, поддерживающих инициированную менеджером передачу результатов измерений, представлена в 8.9.3.3.3. Информация об ограниченном управлении со стороны менеджера агентами, не поддерживающими инициируемую менеджером передачу результатов измерений, представлена в 8.9.3.3.2.

- Scale-and-Range-Specification

Атрибут Scale-and-Range-Specification определяет коэффициенты для алгоритма отображения масштабированных величин на их абсолютные значения. Менеджер должен применять следующий алгоритм:

$Y = M \times X + B$,

где

Y = преобразованная абсолютная величина;

M = (верхнее абсолютное значение — нижнее абсолютное значение) / (верхнее масштабированное значение — нижнее масштабированное значение);

B = верхнее абсолютное значение — (M × верхнее масштабированное значение);

X = масштабированное значение.

Пример использования этого алгоритма приведен в приложении В.

Необходимо учесть, что термин «абсолютное значение» не обозначает математическое абсолютное значение, которое всегда положительно, а относится к фактическому измеренному значению.

6.3.5.4 Методы объектов RT-SA

В настоящем стандарте отсутствуют определения методов объектов RT-SA, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

6.3.5.5 События объектов RT-SA

В настоящем стандарте события объектов RT-SA не определены.

6.3.5.6 Прочие службы класса RT-SA

На сегодняшний день службы SET или GET класса Numeric, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые их задействуют.

6.3.6 Класс Enumeration

6.3.6.1 Общие положения

Экземпляр класса Enumeration представляет информацию о статусе и/или аннотации. Значения объекта Enumeration кодируются в форме нормативных кодов (согласно ИСО/ИИЭР 11073-10101 [B16]) или произвольного текста. Значения объекта Enumeration передаются от агента к менеджеру с помощью службы EVENT REPORT (см. 7.3). Данный класс произведен от базового класса Metric.

6.3.6.2 Идентификация класса Enumeration

Для идентификации класса Enumeration используется номенклатурный код MDC_MOC_VMO_METRIC_ENUM.

6.3.6.3 Атрибуты класса Enumeration

Таблица 9 содержит описание набора атрибутов класса Enumeration, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 9 — Атрибуты класса Enumeration

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Enum-Observed-Value-Simple-OID	MDC_ATTR_ENUM_OBS_VAL_SIMP_OID	OID-Type	Данное значение сообщается в виде номенклатурного кода. Если атрибуту Enum-Observed-Value-Partition присвоено значение, то оно определяет раздел номенклатуры для данного атрибута. В противном случае OID-Type заимствуется из номенклатурного раздела, указанного в поле раздела атрибута Type. Должен присутствовать один и только один из следующих атрибутов: Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str или Enum-Observed-Value	Условно обязательный, наблюдаемый

Окончание таблицы 9

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Enum-Observed-Value-Simple-Bit-Str	MDC_ATTR_ENUM_OBS_VAL_SIMP_BIT_STR	BITS-32	Данное значение сообщается в виде 32-битовой строки. Должен присутствовать один и только один из следующих атрибутов: Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str или Enum-Observed-Value	Условно обязательный, наблюдаемый
Enum-Observed-Value-Basic-Bit-Str	MDC_ATTR_ENUM_OBS_VAL_BASIC_BIT_STR	BITS-16	Данное значение сообщается в виде 16-битовой строки. Должен присутствовать один и только один из следующих атрибутов: Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str или Enum-Observed-Value	Условно обязательный, наблюдаемый
Enum-Observed-Value-Simple-Str	MDC_ATTR_ENUM_OBS_VAL_SIMP_STR	EnumPrintableString	Данное значение передается в виде печатаемой строки в кодировке ASCII. Должен присутствовать один и только один из следующих атрибутов: Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str или Enum-Observed-Value	Условно обязательный, наблюдаемый
Enum-Observed-Value	MDC_ATTR_VAL_ENUM_OBS	EnumObsValue	Данный атрибут определяет структурированное наблюдаемое значение, обеспечивающее дополнительную гибкость типа данных сообщаемого значения. Должен присутствовать один и только один из следующих атрибутов: Enum-Observed-Value-Simple-OID, Enum-Observed-Value-Simple-Bit-Str, Enum-Observed-Value-Basic-Bit-Str, Enum-Observed-Value-Simple-Str или Enum-Observed-Value	Условно обязательный, наблюдаемый
Enum-Observed-Value-Partition	MDC_ATTR_ENUM_OBS_VAL_PART	NomPartition	Данный атрибут может использоваться для определения раздела, из которого взят наблюдаемый номенклатурный термин OID для Enum-Observed-Value-Simple-OID или Enum-Observed-Value. Если этот атрибут отсутствует, то раздел совпадает с разделом номенклатуры, указанным в поле раздела атрибута Type	Необязательный, динамический

6.3.6.4 Методы объектов Enumeration

В настоящем стандарте отсутствуют определения методов объектов Enumeration, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

6.3.6.5 События объектов Enumeration

В настоящем стандарте события объектов Enumeration не определены.

6.3.6.6 Прочие службы объектов Enumeration

На сегодняшний день службы SET или GET класса Enumeration, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые их задействуют.

6.3.7 Класс PM-store

6.3.7.1 Общие положения

Экземпляр класса PM-store предоставляет возможности длительного хранения данных класса Metric. Данные хранятся в переменном числе объектов (сегментов) PM-segment (см. 6.3.8). Хранимые данные объекта PM-store запрашиваются менеджером у агента с помощью служб доступа к объектам (см. 7.3). Если понятие класса PM-store не известно, то перед тем, как ознакомиться со следующими подпунктами, можно ознакомиться с концептуальным обзором, приведенным в приложении С.

Для описания измерений может потребоваться дополнение значений атрибутов, хранящихся в сегменте PM-segment, дополнительными атрибутами этого объекта. Типичным примером могут служить единицы измерения. Если значение атрибута сегмента PM-segment зависит от значения атрибута, не хранящегося в этом сегменте, то такой зависимый атрибут не должен изменять значение в течение срока существования сегмента PM-segment. Иначе агент должен хранить значение зависимого атрибута в сегменте PM-segment.

6.3.7.2 Идентификация класса PM-store

Для идентификации класса PM-store используется номенклатурный код MDC_MOC_VMO_PMSTORE.

6.3.7.3 Атрибуты класса PM-store

Таблица 10 содержит описание набора атрибутов класса PM-store, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 10 — Атрибуты класса PM-store

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Атрибут Handle представляет собой ссылочный идентификатор этого объекта. Каждый объект должен иметь уникальный идентификатор, присвоенный агентом. Атрибут Handle идентифицирует объект в отчетах о событиях, передаваемых менеджеру и в адрес экземпляра объекта в сообщениях, вызывающих методы объекта	Обязательный, статический
PM-Store-Capab	MDC_ATTR_PM_STORE_CAPAB	PmStoreCapab	Данный атрибут определяет базовые возможности экземпляра объекта PM-store	Обязательный, статический
Store-Sample-Algorithm	MDC_ATTR_METRIC_STORE_SAMPLE_ALG	StoSampleAlg	Данный атрибут описывает, как обрабатываются значения считываний, хранящихся в сегментах PM-segment. Структура StoSampleAlg описывает доступные алгоритмы обработки считываний. Если алгоритм обработки считываний не применяется (другими словами, значения считываний являются необработанными данными), данный атрибут должен иметь значение st-alg-no-downsampling	Обязательный, статический
Store-Capacity-Count	MDC_ATTR_METRIC_STORE_CAPAC_CNT	INT-U32	Данный атрибут указывает максимальное число записей, которые могут храниться в сегментах PM-segment (записей во всех содержащихся сегментах PM-segment)	Необязательный, статический
Store-Usage-Count	MDC_ATTR_METRIC_STORE_USAGE_CNT	INT-U32	Данный атрибут указывает фактическое число записей, хранящихся в сегментах PM-segment (записей во всех содержащихся сегментах PM-segment)	Необязательный, динамический

Окончание таблицы 10

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Operation-al-State	MDC_ATTR_ OP_STAT	Operation-alState	Данный атрибут указывает, вносятся ли в данный момент времени новые записи в какой-либо сегмент PM-segment. Если в данный момент активно добавляются новые данные в какой-либо сегмент PM-segment, содержащийся в данном объекте PM-store, то данный атрибут должен быть активирован. Иначе он дожен быть задан неактивным	Обязательный, динамический
PM-Store-La-bel	MDC_ATTR_ PM_STORE_LA-BEL_STRING	OCTET STRING	Данный атрибут представляет собой зависящую от приложения метку для объекта PM-store в печатаемых символах кодировки ASCII, позволяющую указать предполагаемое назначение, которая может использоваться для целей изображения	Необязатель-ный, статический
Sample-Period	MDC_ATTR_ TIME_PD_SAMP	RelativeTime	Данный атрибут определяет частоту добавления записей в сегмент PM-segment. Если значения считываются периодически, этот атрибут должен присутствовать в объекте PM-store (применяется ко всем сегментам PM-segment, периодически сохраняемым в объекте PM-store) или в каждом сегменте PM-segment таким образом, чтобы разность времени помещения двух записей в атрибут Fixed-Segment-Data оставалась постоянной (т.е. в атрибуте Pm-Store-Capab задан бит pmsc-peri-seg-entries)	Условно обязательный, статический
Num-ber-Of-Seg-ments	MDC_ATTR_NUM_ SEG	INT-U16	Данный атрибут указывает текущее число сегментов PM-segment, содержащихся в объекте PM-store. Необходимо отметить, что атрибут Instance-Number сегмента PM-segment НЕ связан с этим числом (т. е. не обязан находиться в диапазоне от 0 до значения Number-Of-Segments), но должен извлекаться с помощью метода Get-Seg-ment-Info или Get-Segment-Id-List	Обязательный, динамический
Clear-Timeout	MDC_ATTR_ CLEAR_TIMEOUT	RelativeTime	Данный атрибут тайм-аута определяет мини-мальное время, в течение которого менеджер должен ожидать завершения выполнения команды очистки объекта PM-store. Если после отправки менеджером команды вызова подтверждаемого действия (Clear-Segments) тайм-аут истек до того, как менеджер получил соответствующее сообщение ответа подтверждаемого действия, менеджер должен перейти в состояние «Не ассоциирован» согласно 8.9.5.6. Данный атрибут необходим в тех случаях, когда агент поддерживает действие очистки сегментов	Условно обязательный, динамический

Атрибуты Handle и PM-Store-Capab являются частью конфигурации агента; следовательно, менеджер знает значения соответствующих атрибутов после процедуры конфигурирования.

6.3.7.4 Методы объектов PM-store

Таблица 11 содержит информацию о методах (действиях) объекта PM-store. Данные методы могут вызываться с помощью службы ACTION.

Таблица 11 — Методы объектов PM-store

Метод/действие	Режим	Тип действия	Поле action-info-args	Поле action-info-args результата
Clear-Segments	Подтверждаемый	MDC_ACT_SEG_CLR	SegmSelection	(пусто)
Get-Segment-Info	Подтверждаемый	MDC_ACT_SEG_GET_INFO	SegmSelection	SegmentInfoList
Get-Segment-Id-List	Подтверждаемый	MDC_ACT_SEG_GET_ID_LIST	(пусто)	SegmIdList
Trig-Segment-Data-Xfer	Подтверждаемый	MDC_ACT_SEG_TRIG_XFER	TrigSegmDataXferReq	TrigSegmDataXferRsp

Если агент поддерживает класс PM-store, поддержка метода Get-Segment-Info или Get-Segment-Id-List обязательна, и поддержка метода Trig-Segment-Data-Xfer также обязательна. Поддержка метода Clear-Segments не обязательна и указывается в атрибуте PM-Store-Capab.

Если менеджер поддерживает класс PM-store, поддержка вызова методов Get-Segment-Info, Get-Segment-Id-List и Trig-Segment-Data-Xfer обязательна. Поддержка вызова метода Clear-Segments не обязательна.

- Clear-Segments

Данный метод позволяет менеджеру удалить данные, хранимые в одном или нескольких выбранных сегментах PM-segment. Все записи в этих объектах удаляются. Если агент поддерживает переменное количество сегментов PM-segment, агент может удалить пустые объекты PM-segment. Кроме того, агент может очистить сегменты PM-segment без команды от менеджера (например, пользователь агента может удалить данные, хранимые агентом), однако, если это действие выполняется в состоянии «Ассоциирован», атрибут Instance-Number должен оставаться действительным и ссылаться на пустой объект PM-segment в течение срока существования ассоциации. Атрибут Instance-Number всех остальных сегментов PM-segment должен оставаться неизменным при очистке сегмента.

Удаление всех выбранных сегментов PM-segment не гарантируется этим методом. Если у сегмента PM-segment атрибут Operational-State активен, то запрошенное удаление не будет выполнено. Кроме того, агент может принять решение о защите определенных сегментов от удаления, присвоив им статус «только для чтения» (например, пользователь агента решил «заблокировать» определенные данные). Защищенные и незащищенные сегменты останутся неизменными при выполнении операции clear-segments.

Если хотя бы один из выбранных сегментов очищен, то должно передаваться сообщение об успешном завершении операции (rors). Однако получение сообщения об успешном завершении операции не всегда означает, что все целевые сегменты действительно очищены (и потенциально удалены), поскольку возможно наличие защищенного или активного подмножества.

Если все выбранные сегменты оказались неочищенными (вследствие защиты или активного использования), агент должен сообщить об ошибке not-allowed-by-object (roer). Код возврата должен задаваться равным MDC_RET_CODE_OBJ_BUSY, если какие-либо сегменты не удается очистить вследствие активного использования. Иначе код возврата должен равняться MDC_RET_CODE_UNKNOWN, указывая, что во время выполнения операции обнаружены только сегменты, защищенные агентом.

При очистке объектов PM-segment с помощью метода time с использованием абсолютного времени очищаются только те объекты PM-segment, поля Segment-Start-Abs-Time и Segment-End-Abs-Time которых находятся полностью внутри указанного интервала времени.

При очистке объектов PM-segment с помощью метода time с использованием базового времени и смещения очищаются только те объекты PM-segment, поля Segment-Start-BO-Time и Segment-End-BO-Time которых находятся полностью внутри указанного интервала времени. При использовании Segment-Start-BO-Time и Segment-End-BO-Time базовое время должно иметь действительное значение (например, ненулевое значение). Если поле смещения имеет значение 0x7FFF (32767), очищаются только те объекты PM-segment, базовое время которых находится полностью внутри указанного интервала. При любом другом значении поля смещения очищаются только те объекты PM-segment, местное время (базовое время с добавленным смещением) которых находится полностью внутри указанного интервала.

Необходимо отметить, что поведение метода Clear-Segments зависит от конкретного приложения. Метод может удалить все записи из определенного сегмента PM-segment, оставив его пустым, или

может полностью удалить указанный сегмент PM-segment. Данное поведение определяется атрибутом PM-Store-Capab. Для специфичных приложений рекомендации формулируются в соответствующих специализациях приборов, применяющих объекты PM-store.

Агент, поддерживающий метод Clear-Segments, должен поддерживать для поля action-info-args типа SegmSelection, передаваемого при вызове этого метода, как минимум вариант all-segments (все сегменты).

Если агент поддерживает вариант all-segments для поля action-info-args типа SegmSelection, передаваемого при вызове метода Clear-Segments, то агент должен установить в атрибуте PM-Store-Capab флаг pmsc-clear-segm-all-sup. Если агент поддерживает вариант segm-id-list (список идентификаторов сегментов) для поля action-info-args типа SegmSelection, передаваемого при вызове метода Clear-Segments, то агент должен установить в атрибуте PM-Store-Capab флаг pmsc-clear-segm-by-list-sup. Если агент поддерживает вариант abs-time-range (диапазон абсолютного времени) или bo-time-range (диапазон базового времени и смещения) для поля action-info-args типа SegmSelection, передаваемого при вызове метода Clear-Segments, то агент должен установить флаг pmsc-clear-segm-by-time-sup в атрибуте PM-Store-Capab.

Если агент не поддерживает действие Clear-Segments, инициированное менеджером, то он должен возвратить ошибку no-such-action (roer).

Если менеджер поддерживает отправку метода Clear-Segments, то он должен поддерживать как минимум вариант all-segments для поля action-info-args типа SegmSelection, передаваемого при вызове этого метода. Менеджер может поддерживать дополнительные варианты выбора.

Если ни один объект PM-segment не соответствует критериям, указанным в поле action-info-args типа SegmSelection, и никакой объект PM-segment не очищается действием, то это не считается ошибкой и передается нормальный ответ.

- Get-Segment-Info

Данный метод позволяет менеджеру извлечь атрибуты объекта PM-segment из одного или нескольких объектов PM-segment, за исключением атрибута Fixed-Segment-Data, который содержит фактические сохраненные данные и извлекается с помощью метода Trig-Segment-Data-Xfer. В частности, метод Get-Segment-Info позволяет менеджеру извлечь атрибуты и их данные из экземпляров объектов PM-segment, идентифицируемых параметром типа SegmSelection.

Агент, поддерживающий метод Get-Segment-Info, должен поддерживать для поля action-info-args типа SegmSelection, передаваемого при вызове этого метода, вариант all-segments. Агент может поддерживать варианты segm-id-list, abs-time-range и/или bo-time-range для поля action-info-args типа SegmSelection, передаваемого при вызове метода Get-Segment-Info. В этом случае агент должен установить флаг pmsc-segm-id-list-select и/или pmsc-abs-time-select атрибута PM-Store-Capab. Если менеджер отправляет метод Get-Segment-Info с вариантом, не поддерживаемым агентом, то агент должен сообщить об ошибке неподдерживаемого варианта (unsupported-choice, roer).

Для информации об объектах PM-segment, возвращаемой по заданному диапазону времени, сегменты выбираются с использованием алгоритма, описанного для метода Clear-Segments.

Если менеджер поддерживает отправку метода Get-Segment-Info, то он должен поддерживать как минимум вариант all-segments для поля action-info-args типа SegmSelection, передаваемого при вызове этого метода. Менеджер может поддерживать дополнительные варианты выбора.

Если стандартная конфигурация содержит какой-либо объект PM-store, менеджер должен отправить метод Get-Segment-Info или Get-Segment-Id-List в начале доступа к любому объекту PM-store.

Если ни один объект PM-segment не соответствует критериям, указанным в поле action-info-args типа SegmSelection, и никакой объект PM-segment не обнаружен, то это не считается ошибкой, передается нормальный ответ, и список информации о сегментах будет просто пустым.

Если для поля action-info-args типа SegmSelection использован вариант segm-id-list, имеющий пустое значение, то ответом должен быть пустой результат segment-info-list.

Если агент поддерживает метод Get-Segment-Info, то он должен установить в атрибуте PM-Store-Capab флаг pmsc-get-segm-info-sup.

- Get-Segment-Id-List

Данный метод позволяет менеджеру извлечь список номеров экземпляров всех PM-segment PM-store. В частности, метод Get-Segm-Id-List позволяет менеджеру затем извлечь атрибуты выбранных экземпляров объектов PM-segment и их данные без необходимости извлечения информации всех PM-segment. Кроме того, менеджер может извлечь несколько PM-segment с помощью последовательности запросов.

Если стандартная конфигурация содержит какой-либо объект PM-store, менеджер должен отправить Get-Segment-Info или Get-Segment-Id-List в начале доступа к любому объекту PM-store.

Если агент поддерживает метод Get-Segment-Id-List, то он должен установить флаг pmsc-get-segm-id-list-sup в атрибуте PM-Store-Capab.

- Trig-Segment-Data-Xfer

Данный метод позволяет менеджеру начать передачу атрибута Fixed-Segment-Data заданного сегмента PM-segment. Агент указывает в ответе, принимает ли он или отклоняет этот запрос. Ели агент принимает запрос, то он посылает сообщения Segment-Data-Event в соответствии с 6.3.7.5. Если сегмент PM-segment, указанный в данном методе, имеет активный атрибут Operational-State, то агент должен возвратить ошибку not-allowed-by-object (roer) с кодом возврата MDC_RET_CODE_OBJ_BUSY.

6.3.7.5 События объектов PM-store

Таблица 12 определяет потенциальные события, передаваемые объектом PM-store.

Таблица 12 — События объектов PM-store

Событие	Режим	Тип события	Параметр информации о событии	Event-reply-info
Segment-Data-Event	Подтверждено	MDC_NOTI_SEGMENT_DATA	SegmentDataEvent	SegmentDataResult

- Segment-Data-Event

По этому событию данные, хранящиеся в атрибуте Fixed-Segment-Data сегмента PM-segment, передаются от агента к менеджеру. Данное событие инициируется менеджером с помощью метода Trig-Segment-Data-Xfer. После инициирования передачи данных агент посылает сообщения о событии Segment-Data-Event до момента полной передачи содержания атрибута Fixed-Segment-Data или прерывания передачи со стороны менеджера или агента. Полное описание передачи содержания сегмента PM-segment см. в 8.9.3.4.2.

Рекомендуется помещать в одно сообщение о событии Segment-Data-Event максимально возможное число записей, хранящихся в сегменте PM-segment, чтобы уменьшить количество сообщений, необходимых для передачи содержания сегмента. Агент должен передавать все записи, хранящиеся в сегменте, в порядке «первым пришел — первым ушел» (FIFO: first-in, first-out).

Поддержка события агентом обязательна, если агент поддерживает объекты PM-store.

Если подтверждаемый отчет о событии используется объектом PM-store, то в любой момент времени допустимо получение от этого объекта не более одного неквитированного подтверждаемого отчета о событии.

6.3.7.6 Прочие службы PM-store

6.3.7.6.1 Служба GET

Поддержка службы GET должна обеспечиваться любым агентом, поддерживающим один и более объектов PM-store, только пока находится в состоянии «Выполнение». Менеджер использует службу GET в целях извлечения значений всех атрибутов объекта PM-store. Если менеджер не имеет текущее значение необходимого атрибута PM-store, то должна использоваться служба GET. Агент может также отправлять отчеты о событиях сканера, предоставляющие менеджеру обновления текущих значений атрибутов, однако такое поведение агента не обязательно.

Менеджер может запросить у агента атрибуты объекта PM-store, для чего должен послать команду «Remote Operation Invoke | Get» (см. roiv-cmip-get в А.10.2) со значением дескриптора объекта PM-store, определенного в конфигурации агента. Агент должен возвратить менеджеру отчет с атрибутами объекта PM-store, используя ответ «Remote Operation Response | Get» (см. rors-cmip-get в А.10.2). Подробное описание операции GET см. в 8.9.3.4.2.

6.3.7.6.2 Служба SET

На сегодняшний день служба SET PM-store, определенная в настоящем стандарте, не используется, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые ее задействуют.

6.3.8 Класс PM-segment

6.3.8.1 Общие положения

Экземпляр класса PM-segment представляет собой постоянно хранимый эпизод измерений данных. Объект (сегмент) PM-segment не является частью статической конфигурации агента, поскольку количество созданных экземпляров сегментов PM-segment может динамично меняться. Менеджер осу-

ществляет опосредованный доступ к сегментам PM-segment с помощью методов и событий объекта PM-store.

6.3.8.2 Идентификация класса PM-segment

Для идентификации класса PM-segment используется номенклатурный код MDC_MOC_PM_SEGMENT.

6.3.8.3 Атрибуты класса PM-segment

Таблица 13 содержит описание набора атрибутов класса PM-segment, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 13 — Атрибуты класса PM-segment

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Instance-Number	MDC_ATTR_ID_INSTNO	InstNumber	Атрибут Instance-Number представляет собой идентификатор определенного экземпляра сегмента PM-segment. Каждый экземпляр должен иметь уникальный номер (в контексте объекта PM-store), назначенный агентом. Такой номер используется менеджером для идентификации сегмента PM-segment	Обязательный
PM-Segment-Entry-Map	MDC_ATTR_PM_SEG_MAP	PmSegment-EntryMap	Данный атрибут определяет формат и содержимое одной сохраненной записи. Запись обладает необязательным заголовком, содержащим информацию, применимую ко всем элементам записи. Запись содержит один или более элементов, определяемых классом, идентификатором метрики, дескриптором и картой значений атрибута, определяющей атрибуты объекта для каждого элемента в сегменте PM-segment	Обязательный
PM-Seg-Person-Id	MDC_ATTR_PM_SEG_PERSON_ID	PersonId	Настоящий стандарт поддерживает приборы, которые способны обрабатывать данные нескольких лиц. Идентификатор лица используется для их различия. В атрибуте PM-Store-Capab сегмента PM-store, способного хранить данные нескольких лиц, должен быть установлен бит pmsc-multi-person. Если он установлен, то все экземпляры сегментов PM-segment, содержащиеся в объекте PM-store, должны поддерживать атрибут PM-Seg-Person-Id. В противном случае данный атрибут не определен	Условно-обязательный
Operational-State	MDC_ATTR_OP_STAT	OperationalState	Этот атрибут указывает, вносятся ли в данный момент новые записи в этот сегмент PM-segment. Если в текущий момент времени в сегмент PM-segment добавляются новые данные, данный атрибут должен быть активирован. Иначе он должен быть сделан неактивным	Обязательный
Sample-Period	MDC_ATTR_TIME_PD_SAMP	RelativeTime	Данный атрибут определяет частоту добавления записей в сегмент PM-segment. Если значения считываются периодически, данный атрибут должен присутствовать в объекте PM-store (применяется ко всем сегментам PM-segment, периодически сохраняемым в объекте PM-store) или в каждом сегменте PM-segment. Если значения считываются периодически, то в атрибуте PM-Store-Capab необходимо установить бит pmsc-peri-seg-entries	Условно обязательный

Продолжение таблицы 13

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Segment-Label	MDC_ATTR_PM_SEG_LABEL_STRING	OCTET STRING	Данный атрибут представляет собой зависящую от приложения метку для сегмента PM-segment в печатаемых символах кодировки ASCII, позволяющую указать предполагаемое назначение, которая может использоваться для целей изображения	Необязательный
Segment-Start-Abs-Time	MDC_ATTR_TIME_START_SEG	AbsoluteTime	Данный атрибут определяет начальное время сегмента PM-segment. Данный атрибут требуется в тех случаях, когда сегмент поддерживает действия поиска по диапазону времени (например, заданы биты pmsc-abs-time-select и/или pmsc-clear-segm-by-time-sup). Если этот атрибут используется, то атрибут Segment-Start-BO-Time не должен использоваться	Условно обязательный
Segment-End-Abs-Time	MDC_ATTR_TIME_END_SEG	AbsoluteTime	Данный атрибут определяет конечное время сегмента. Данный атрибут требуется в тех случаях, когда сегмент поддерживает действия поиска по диапазону времени (например, заданы биты pmsc-abs-time-select и/или pmsc-clear-segm-by-time-sup). Если этот атрибут используется, то атрибут Segment-End-BO-Time не должен использоваться	Условно обязательный
Date-and-Time-Adjustment	MDC_ATTR_TIME_ABS_ADJUST	AbsoluteTime-Adjust	Данный атрибут сообщает о любых изменениях даты и времени, которые вносятся при настройке часов или возникновении таких событий, как переход на летнее время. Если агент когда-либо изменяет дату и время, данный атрибут сообщает о таком изменении	Условно обязательный
Segment-Start-BO-Time	MDC_ATTR_TIME_START_SEG_BO	BaseOffset-Time	Данный атрибут определяет начальное время сегмента в качестве базового времени со смещением в минутах по местному времени. Данный атрибут требуется в тех случаях, когда сегмент поддерживает действия поиска по диапазону времени (например, заданы биты pmsc-abs-time-select и/или pmsc-clear-segm-by-time-sup). Базовое время со смещением рекомендуется использовать в тех случаях, когда ожидаются корректировки времени. Если этот атрибут используется, то атрибут Segment-Start-Abs-Time не должен использоваться	Условно обязательный
Segment-End-BO-Time	MDC_ATTR_TIME_END_SEG_BO	BaseOffset-Time	Данный атрибут определяет конечное время сегмента в качестве базового времени со смещением в минутах по местному времени. Данный атрибут требуется в тех случаях, когда сегмент поддерживает действия поиска по диапазону времени (например, заданы биты pmsc-abs-time-select и/или pmsc-clear-segm-by-time-sup). Базовое время со смещением рекомендуется использовать в тех случаях, когда ожидаются корректировки времени. Если этот атрибут используется, атрибут Segment-End-Abs-Time не должен использоваться	Условно обязательный
Segment-Usage-Count	MDC_ATTR_SEG_USAGE_CNT	INT-U32	Данный атрибут указывает фактическое (текущее) количество хранящихся записей	Необязательный

Окончание таблицы 13

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Segment-Statistics	MDC_ATTR_SEG_STATS	SegmentStatistics	Данный атрибут определяет массив для передачи минимальной, средней и максимальной статистики, связанной с каждым маркируемым элементом	Необязательный
Fixed-Segment-Data	MDC_ATTR_SEG_FIXED_DATA	Данные хранятся внутри прибора, поэтому этот тип данных никогда не появляется непосредственно ни в каком определении протокола	Атрибут Fixed-Segment-Data определяет данные сегмента, передаваемые как массив записей в формате, указанном атрибутом PM-Segment-Entry-Map. В настоящем стандарте этот массив имеет непрозрачную структуру без определенного типа данных. Обратите внимание, что данный атрибут доступен опосредованно и извлекается только менеджером с помощью метода Trig-Segment-Data-Xfer PM-store	Обязательный
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	Данный информационный атрибут тайм-аута определяет минимальное время, на протяжении которого агент должен ожидать ответного сообщения от менеджера после генерирования сообщения вызова подтверждаемого отчета о событии, прежде чем перейти в состояние «Не ассоциирован». Атрибут является информационным и используется для улучшения работы менеджера. В случае предоставления этого атрибута необходимо обеспечить его соответствие фактическому значению времени ожидания, которое используется агентом для подтверждаемого отчета о событии, сгенерированного на основе объекта PM-store. Данный атрибут является информационным для менеджера в том смысле, что менеджер не использует этот атрибут при фактической реализации протокола (например, менеджер не инициирует тайм-аут для переданного агентом подтверждаемого отчета о событии). Однако менеджер может пожелать использовать данную информацию для назначения приоритета обработки сообщения от агента с «коротким» тайм-аутом по отношению к сообщению от агента с «длинным» тайм-аутом	Необязательный
Transfer-Timeout	MDC_ATTR_TRANSFER_TIMEOUT	RelativeTime	Данный атрибут тайм-аута определяет минимальное время, в течение которого менеджер должен ожидать полной передачи информации сегмента PM-segment. Если время ожидания истекает до получения полного содержания сегмента PM-segment, то менеджер должен перейти в состояние «Не ассоциирован» согласно 8.9.5.6	Обязательный

Примечание — Квалификаторы атрибутов «статический», «динамический» и «наблюдаемый» исключены из таблицы 13, поскольку сегменты PM-segment динамичны (сам по себе объект может изменяться на протяжении срока существования ассоциации).

Атрибут Fixed-Segment-Data хранит массив одинаково форматированных записей. Если результаты измерения не доступны в определенный момент времени, значение численного измерения, представленного типом данных (S)FLOAT-type, должно использовать специальное значение NaN (не число) для обозначения недоступности значения.

В зависимости от возможностей агента и приложения атрибут Fixed-Segment-Data может содержать очень большие объемы данных. Агент может ограничить максимальный размер атрибута Fixed-Segment-Data, чтобы таким образом согласовать его с максимальным размером блока передачи транспортной системы. Для реализации такого поведения менеджер, поддерживающий объект PM-store, должен быть способен принимать атрибуты Fixed-Segment-Data в нескольких прикладных сообщениях.

6.3.8.4 Методы объектов PM-segment

В настоящем стандарте методы объектов PM-segment не определены.

6.3.8.5 События объектов PM-segment

В настоящем стандарте события объектов PM-segment не определены.

6.3.8.6 Прочие службы объектов PM-segment

На сегодняшний день службы SET и GET объектов PM-segment, определенные в настоящем стандарте, не используются.

6.3.9 Классы Scanner

6.3.9.1 Общие положения

Класс Scanner используется для двух целей: (1) обеспечивает менеджеру возможность управления потоком данных, и (2) обеспечивает оптимизированный (по сравнению с использованием событий класса MDS) механизм компоновки и передачи отчетов. Он позволяет более эффективно добавлять в один отчет о событии различные наборы изменений значений атрибутов (AttributeChangeSet), полученные от одного или нескольких объектов Metric. У класса Scanner есть потомки EpCfgScanner и PeriCfgScanner, описывающие соответственно эпизодическое и периодическое сканирование. В отчетах о событиях сканирования оба этих объекта могут передаваться в переменном, фиксированном или групповом формате (см. 7.4.5). Иерархия класса Scanner и его потомков показана на рисунке 5. Каждый класс описан соответственно в 6.3.9.3 — 6.3.9.6.

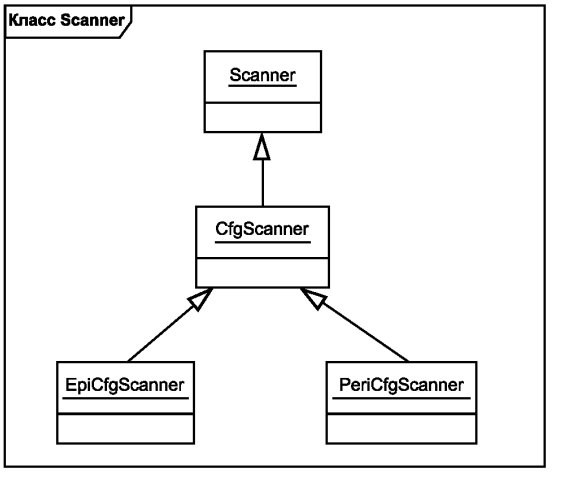


Рисунок 5 — Информационная модель предметной области персонального медицинского прибора: модель класса Scanner

6.3.9.2 Концептуальная модель

Объект Scanner не сканирует объекты (то есть не считывает состояние объекта и не сообщает о том, произошли или нет какие-либо изменения). Вместо этого объект Scanner собирает наборы AttributeChangeSet и отображает их в структуру ObservationScan отчета о событиях сканирования. Объекты эпизодического сканирования (EpiCfgScanner) передают отчеты о событиях сканирования после завершения эпизода. Что считать эпизодом, определяется приложением, однако обычно он соответствует одному или нескольким наборам AttributeChangeSet (один и тот же объект не генерирует два набо-

ра AttributeChangeSet), возникающим эпизодически (интервал времени между эпизодами неизвестен). Объекты периодического сканирования (PeriCfgScanner) передают отчеты о событиях сканирования по истечении периода времени, указанного в атрибуте Reporting-Interval, при этом от одного и того же объекта может быть получено несколько наборов AttributeChangeSet.

Ошибки функционирования датчика или прочие условия (например, отсутствие доступных данных во время первой активации объекта Scanner) могут привести к отсутствию наборов AttributeChangeSet, когда генерируется отчет о событиях сканирования. Отчет о событии сканирования, генерируемый в этих случаях, зависит от типа объекта Scanner и используемого формата.

Объекты EpiCfgScanner и PeriCfgScanner, использующие групповой формат, должны создавать отчеты о событиях сканирования, соответствующие следующим требованиям.

- Любому набору AttributeChangeSet, содержащему значения наблюдаемого атрибута, полученные от объекта Numeric (см. таблицу 7), присваивается значение NaN. Если набор AttributeChangeSet содержит атрибут Measurement-Status, то в зависимости от ситуации он должен указывать, что значение недействительное или недоступное.

- Любой набор AttributeChangeSet, содержащий значения Simple-Sa-Observed-Value объекта RT-SA, должен содержать атрибут Measurement-Status. В зависимости от ситуации атрибут Measurement-Status должен указывать, что значение недействительное или недоступное. В этом случае значения Simple-Sa-Observed-Value не определены и менеджер должен игнорировать сообщаемые значения.

- Любому набору AttributeChangeSet, содержащему значения наблюдаемого атрибута, полученные от объекта Enumeration (см. таблицу 9), должно быть присвоено подходящее перечислимое значение. При необходимости он должен содержать атрибут Measurement-Status. В зависимости от ситуации атрибут Measurement-Status должен указывать, что значение недействительное или недоступное. Если атрибут Measurement-Status используется, чтобы указать недействительность или недоступность, менеджер должен игнорировать сообщаемое значение.

- Если объектом EpiCfgScanner не собран ни один набор AttributeChangeSets, отчет о событиях сканирования не должен передаваться.

- Если объектом PeriCfgScanner не собран ни один набор AttributeChangeSets, должен передаваться пустой отчет о событиях сканирования.

Объекты EpiCfgScanner и PeriCfgScanner, использующие переменный или фиксированный формат, должны создавать отчеты о событиях сканирования, соответствующие следующим требованиям.

- В структуру ObservationScan отображаются только собранные наборы AttributeChangeSet.

- Если объектом EpiCfgScanner не собран ни один набор AttributeChangeSets, отчет о событиях сканирования не должен передаваться.

- Если объектом PeriCfgScanner не собран ни один набор AttributeChangeSets, должен передаваться пустой отчет о событиях сканирования.

Периодический сканер PeriCfgScanner отличается от эпизодического EpiCfgScanner способностью получать несколько наборов AttributeChangeSet от одного объекта перед отправкой. Периодическому Scanner также требуется, чтобы скорость генерирования всех собранных наборов AttributeChangeSet имела постоянную временную связь с периодом сканера. Если набор AttributeChangeSet не имеет явно заданной метки времени, соответствующая метка времени должна определяться на основе метки времени отчета о событии сканирования. Из этого следует, что любой набор AttributeChangeSet, полученный в момент времени, отличающийся от времени отчета о событии сканирования, должен передаваться со своей собственной меткой времени. Периодический сканер должен добавлять наборы AttributeChangeSet одного и того же объекта в отчет о событиях сканирования, соблюдая хронологический порядок, начиная с наиболее старого набора вверху сообщения.

Различные требования, предъявляемые к отчетам о результатах наблюдений, можно выполнить, используя группы периодических и эпизодических Scanner, по одному на каждый поток наблюдений для управления его характеристиками. Например, пульсоксиметр может использовать периодический конфигурируемый сканер со значением атрибута Reporting-Interval = 50 мс для объекта RT-SA, представляющего плетизмограмму; периодический конфигурируемый сканер со значением атрибута Reporting-Interval = 1 сек для объектов Numeric, представляющих уровень насыщения кислородом, и объектов Enumeration, сообщающих о событиях, связанных со значением; а также эпизодический конфигурируемый сканер EpiCfgScanner для пульсовых объектов Metric (числовых и перечисляемых).

6.3.9.3 Класс Scanner

6.3.9.3.1 Общие положения

Класс Scanner (сканер) представляет собой абстрактный класс, определяющий атрибуты, методы, события и службы, являющиеся общими для его подклассов. Поэтому он не имеет экземпляров.

Понятие сканера предусматривает три формата представления отчетов о событиях: переменный, фиксированный и групповой. Сведения о предоставлении атрибутов наблюдаемых объектов см. в 7.4.5. Форматы отчетов о событиях описаны ниже соответственно в 6.3.9.5.5, 6.3.9.6.5 и А.11.5.

Более специализированные классы сканера произведены от базового класса Scanner.

6.3.9.3.2 Идентификация класса Scanner

Для идентификации класса Scanner используется номенклатурный код MDC_MOC_SCAN.

6.3.9.3.3 Атрибуты класса Scanner

Таблица 14 содержит описание набора атрибутов класса Scanner, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 14 — Атрибуты класса Scanner

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Handle	MDC_ATTR_ID_HANDLE	HANDLE	Сканеры идентифицируются с помощью дескрипторов	Обязательный, статический
Operational-State	MDC_ATTR_OP_STAT	OperationalState	Данный атрибут указывает, отправляет ли сканер отчеты о событиях или нет. Если сканер отправляет отчеты о событиях, значение атрибута должно задаваться активным, иначе — неактивным. Менеджер должен использовать действие SET, чтобы запросить изменение значения этого атрибута	Обязательный, динамический
Scan-Handle-List	MDC_ATTR_SCAN_HANDLE_LIST	HANDLEList	Данный атрибут определяет объекты, произведенные от класса Metric, которые могут сообщаться в отчетах Unbuf-Scan-Report-Var, Buf-Scan-Report-Var, Unbuf-Scan-Report-Fixed, Buf-Scan-Report-Fixed или любых их эквивалентах для нескольких лиц. Для эпизодических сканеров в отчет о событиях добавляется конкретный объект при каждом возникновении набора AttributeChangeSet для этого объекта. Для периодических сканеров наборы AttributeChangeSet, собранные от объектов, включаются в отчет в течение каждого периода. Менеджер не должен предполагать, что порядок объектов типа ObservationScan, содержащихся в отчетах о событиях, совпадает с их порядком в списке Scan-Handle-List. Данный атрибут должен присутствовать, если сканер использует какой-либо из восьми стилей отчетов. Данный атрибут должен задаваться до отправки такого отчета. Пока сканер не активен, значение этого атрибута может изменяться между отчетами о событиях. Информация об изменении значения атрибута передается от объекта MDS к менеджеру с помощью инициированного агентом отчета о событии	Условно обязательный, динамический
Scan-Handle-Attr-Val-Map	MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP	HandleAttrValMap	Данный атрибут определяет объекты, произведенные от класса Metric, атрибуты, а также порядок объектов и значений атрибутов, сообщаемых в отчетах Unbuf-Scan-Report-Grouped, Buf-Scan-Report-Grouped, Unbuf-Scan-Report-MP-Grouped или Buf-Scan-Report-MP-Grouped. Для согласованности структуры сообщения должны присутствовать все значения. Если используется какой-либо из этих четырех стилей отчетов, данный атрибут должен задаваться до отправки такого отчета. Пока сканер не активен, значение этого атрибута может изменяться между отчетами о событиях. Информация об изменении значения атрибута передается от объекта MDS к менеджеру с помощью инициированного агентом отчета о событии	Условно обязательный, динамический

6.3.9.3.4 Методы объектов Scanner

В настоящем стандарте отсутствуют определения методов объектов Scanner, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

6.3.9.3.5 События объектов Scanner

Описания событий производных классов см. в 6.3.9.5.5 и 6.3.9.6.5.

6.3.9.3.6 Прочие службы Scanner

- Служба GET

В настоящем стандарте отсутствует определение службы GET, однако оно может существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

- Служба SET

Агенты, имеющие объекты, произведенные от класса Scanner, должны поддерживать службу SET для атрибута Operational-State этих объектов. Такая служба SET может вызываться как подтверждаемое или неподтверждаемое действие.

6.3.9.4 Класс CfgScanner

6.3.9.4.1 Общие положения

Класс CfgScanner (конфигурируемый сканер) является абстрактным классом, определяющим атрибуты, методы, события и службы, общие для его подклассов. В частности, этот класс определяет коммуникационное поведение объектов этих подклассов. Будучи абстрактным, он не имеет экземпляров.

6.3.9.4.2 Идентификация класса CfgScanner

Для идентификации класса CfgScanner используется номенклатурный код MDC_MOC_SCAN_CFG.

6.3.9.4.3 Атрибуты класса CfgScanner

Таблица 15 содержит описание набора атрибутов класса CfgScanner, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 15 — Атрибуты класса CfgScanner

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Confirm-Mode	MDC_ATTR_CONFIRM_MODE	ConfirmMode	Данный атрибут указывает, являются ли передаваемые отчеты о событиях подтверждаемыми или неподтверждаемыми. В настоящее время атрибут считается динамическим, однако реализации агента должны трактовать его как статический, поскольку, скорее всего, в будущем этот атрибут станет статическим. Изменение значения этого атрибута должно выполняться только после деактивации сканера. Информация об изменении значения атрибута передается от объекта MDS к менеджеру с помощью инициированного агентом отчета о событии	Обязательный, динамический
Confirm-Timeout	MDC_ATTR_CONFIRM_TIMEOUT	RelativeTime	Данный информационный атрибут тайм-аута определяет минимальное время, на протяжении которого агент должен ожидать ответного сообщения от менеджера после генерирования сообщения вызова подтверждаемого отчета о событии, прежде чем перейти в состояние «Не ассоциирован». Атрибут является информационным и используется для улучшения работы менеджера. В случае предоставления этого атрибута необходимо обеспечить его соответствие фактическому значению времени ожидания, которое используется агентом для подтверждаемого отчета о событии, сгенерированного на основе объекта сканера.	Необязательный, динамический

Окончание таблицы 15

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
			Данный атрибут является информационным для менеджера в том смысле, что менеджер не использует этот атрибут при фактической реализации протокола (например, менеджер не инициирует тайм-аут для переданного агентом подтверждаемого отчета о событии). Однако менеджер может пожелать использовать данную информацию для назначения приоритета обработки сообщения от агента с «коротким» тайм-аутом по отношению к сообщению от агента с «длинным» тайм-аутом. В настоящее время атрибут считается динамическим, однако реализации агента должны трактовать его как статический, поскольку, скорее всего, в будущем этот атрибут станет статическим. Изменение значения этого атрибута должно выполняться только после деактивации сканера. Информация об изменении значения атрибута передается от объекта MDS к менеджеру с помощью инициированного агентом отчета о событии	
Transmit-Window	MDC_ATTR_TX_WIND	INT-U16	Данный атрибут указывает информативные данные, предоставляемые агентом, которые могут помочь менеджеру оптимизировать свою конфигурацию. Атрибут Transmit-Window представляет число неквитированных подтверждаемых отчетов о событиях, которым агент позволит оставаться просроченными. Стандарт IEEE 11073-20601-2014 требует, чтобы значение этого атрибута равнялось только 1. В настоящее время атрибут считается динамическим, однако реализации агента должны трактовать его как статический, поскольку, скорее всего, в будущем этот атрибут станет статическим. Изменение значения этого атрибута должно выполняться только после деактивации сканера. Информация об изменении значения атрибута передается от объекта MDS к менеджеру с помощью инициированного агентом отчета о событии	Необязательный, динамический

На рисунке 6 показана обработка необязательной очереди передачи, когда значение Transmit-Window больше 1.

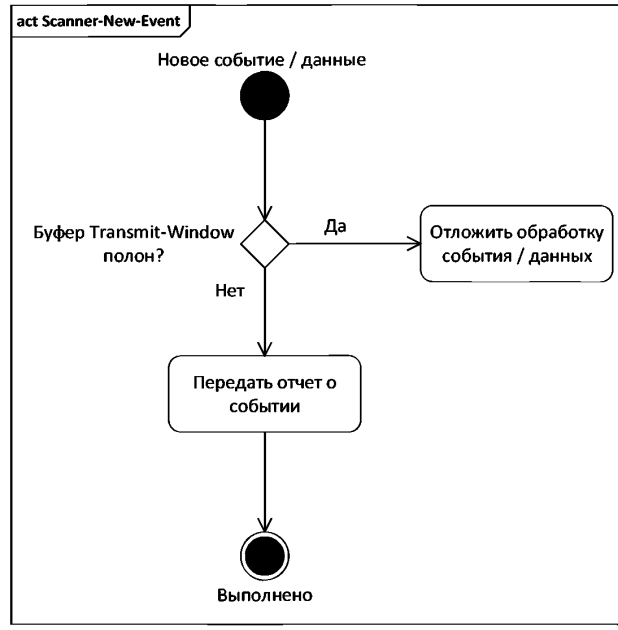


Рисунок 6 — Обработка атрибута Transmit-Window конфигурируемого сканера

6.3.9.4.4 Методы объектов конфигурируемого сканера

В настоящем стандарте отсутствуют определения методов объектов конфигурируемого сканера, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

6.3.9.4.5 События объектов конфигурируемого сканера

Описания событий производных классов см. в 6.3.9.5.5 и 6.3.9.6.5.

6.3.9.4.6 Прочие службы конфигурируемого сканера

На сегодняшний день службы SET или GET конфигурируемого сканера, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые их задействуют.

6.3.9.5 Класс EpiCfgScanner

6.3.9.5.1 Общие положения

Класс EpiCfgScanner (эпизодический конфигурируемый сканер) допускает создание экземпляров. Объекты EpiCfgScanner используются для отправки отчетов, содержащих эпизодические данные, то есть данные, не имеющие постоянного периода между каждым значением данных. Отчет передается при каждом изменении значения одного из наблюдаемых атрибутов, однако интервал времени между двумя последовательными отчетами о событиях не должен быть меньше значения атрибута Min-Reporting-Interval.

6.3.9.5.2 Идентификация класса EpiCfgScanner

Для идентификации класса EpiCfgScanner используется номенклатурный код MDC_MOC_SCAN_CFG_EPI.

6.3.9.5.3 Атрибуты класса EpiCfgScanner

Таблица 16 содержит описание набора атрибутов класса EpiCfgScanner, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 16 — Атрибуты класса EpiCfgScanner

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Min-Reporting-Interval	MDC_ATTR_SCAN_REP_PD_MIN	RelativeTime	Данный атрибут предоставляет оценку ожидаемого минимального времени между любыми двумя последовательными отчетами о событиях	Необязательный, динамический

6.3.9.5.4 Методы объекта EpiCfgScanner

В настоящем стандарте отсутствуют определения методов объектов эпизодического конфигурируемого сканера, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

6.3.9.5.5 События объекта EpiCfgScanner

Таблица 17 указывает потенциальные события, передаваемые объектом эпизодического конфигурируемого сканера. Отчеты о событиях классифицируются как не буферируемые, поскольку агент пересылает событие, когда происходит эпизод, при этом не требуется буферировать информацию в ожидании следующего периода передачи. Агент, поддерживающий взаимодействие с эпизодическим конфигурируемым сканером, должен поддерживать как минимум одно из событий, указанных в таблице 17. Агент, поддерживающий взаимодействие с эпизодическим конфигурируемым сканером, должен поддерживать все события, указанные в таблице 17.

Таблица 17 — События объекта эпизодического конфигурируемого Scanner

Событие	Режим	Тип события	Параметр информации о событии	Event-reply-info
Unbuf-Scan-Report-Var	Подтверждаемый или неподтверждаемый	MDC_NOTI_UNBUF_SCAN_REPORT_VAR	ScanReportInfoVar	—
Unbuf-Scan-Report-Fixed	Подтверждаемый или неподтверждаемый	MDC_NOTI_UNBUF_SCAN_REPORT_FIXED	ScanReportInfoFixed	—
Unbuf-Scan-Report-Grouped	Подтверждаемый или неподтверждаемый	MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED	ScanReportInfoGrouped	—
Unbuf-Scan-Report-MP-Var	Подтверждаемый или неподтверждаемый	MDC_NOTI_UNBUF_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	—
Unbuf-Scan-Report-MP-Fixed	Подтверждаемый или неподтверждаемый	MDC_NOTI_UNBUF_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	—
Unbuf-Scan-Report-MP-Grouped	Подтверждаемый или неподтверждаемый	MDC_NOTI_UNBUF_SCAN_REPORT_MP_GROUPED	ScanReportInfoMPGrouped	—
Примечание — Если от объекта не собрано ни одного набора AttributeChangeSet, то отчеты о сканировании с переменными и фиксированными форматами не будут содержать ни одного набора AttributeChangeSet от этого объекта. Если ни одного набора AttributeChangeSets не собрано, отчет о событии сканирования не отправляется.				

- Unbuf-Scan-Report-Var

Отчет о событии этого стиля содержит сводные данные об объектах и атрибутах, отслеживаемых сканером. Событие инициируется, как только значения данных изменяются, при этом для информирования об изменении данных используется сообщение с переменным форматом (тип/длина/значение).

- Unbuf-Scan-Report-Fixed

Отчет о событии этого стиля используется, как только значения данных меняются, при этом для каждого объекта используется сообщение с фиксированным форматом.

- Unbuf-Scan-Report-Grouped

Отчет о событии этого стиля используется, когда объект сканера используется для передачи данных в наиболее компактном формате. Атрибут Scan-Handle-Attr-Val-Map описывает объекты и атрибуты, включенные в отчет, а также формат сообщения.

- Unbuf-Scan-Report-MP-Var

Стиль события полностью совпадает с Unbuf-Scan-Report-Var, но позволяет включать данные нескольких лиц.

- Unbuf-Scan-Report-MP-Fixed

Стиль события полностью совпадает с Unbuf-Scan-Report-Fixed, но позволяет включать данные нескольких лиц.

- Unbuf-Scan-Report-MP-Grouped

Стиль события полностью совпадает с Unbuf-Scan-Report-Grouped, но позволяет включать данные нескольких лиц.

6.3.9.5.6 Прочие службы эпизодического конфигурируемого сканера

На сегодняшний день службы SET или GET эпизодического конфигурируемого сканера, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые их задействуют.

6.3.9.6 Класс PeriCfgScanner

6.3.9.6.1 Общие положения

Класс PeriCfgScanner (периодический конфигурируемый сканер) допускает создание экземпляров. Объекты PeriCfigScanner используются для передачи отчетов, содержащих периодические данные. Отчеты о событиях должны передаваться с временным интервалом, равным значению атрибута Reporting-Interval.

Количество наблюдений для каждого объекта Metric зависит от интервала обновления объекта Metric и значения атрибута сканера Reporting-Interval.

После того как менеджер активирует периодический конфигурируемый сканер, отчеты о сканировании необходимо передавать в разумные сроки и синхронизировать с интервалом отчетов сканера. Интервал времени между активированием сканера и отправкой первого отчета о сканировании не должен превышать интервал отчетов + 15 секунд.

Примечание — Предполагается, что 15 секунд будет более чем достаточно для инициализации.

Пример — Периодический конфигурируемый сканер настроен на передачу информации о двух объектах Metric с интервалом 1 с. Два объекта периодически обновляют свои соответствующие наблюдаемые значения с интервалом 1 и 1/2 с соответственно. Затем периодический конфигурируемый Scanner генерирует отчеты о событиях каждую секунду, сообщая один скан наблюдения объекта Metric #1 и два скана наблюдения объекта Metric #2. Объекты в атрибуте Scan-Handle-Attr-Val-Map могут содержать две записи для объекта с интервалом обновления ½ с.

6.3.9.6.2 Идентификация объекта PeriCfgScanner

Для идентификации класса PeriCfgScanner используется номенклатурный код MDC_MOC_SCAN_CFG_PERI.

6.3.9.6.3 Атрибуты класса PeriCfgScanner

Таблица 18 содержит описание набора атрибутов класса PeriCfgScanner, поддерживаемых для обмена данными с персональными медицинскими приборами.

Таблица 18 — Атрибуты объекта класса PeriCfgScanner

Имя атрибута	Идентификатор атрибута	Тип атрибута	Примечание	Квалификаторы
Reporting-Interval	MDC_ATTR_SCAN_REP_PD	RelativeTime	Период отправки отчетов о событиях	Обязательный, динамический

6.3.9.6.4 Методы объекта класса PeriCfgScanner

В настоящем стандарте отсутствуют определения методов объектов периодического конфигурируемого сканера, однако они могут существовать в специализациях приборов ИСО/ИИЭР 11073-104zz или в местных документах.

В таблице 19 указаны потенциальные события, передаваемые объектом периодического конфигурируемого сканера. Агент, поддерживающий периодический конфигурируемый сканер, должен поддерживать хотя бы одно из событий, перечисленных в таблице 19. Менеджер, поддерживающий периодические сканеры, должен поддерживать все события, перечисленные в таблице 19.

Таблица 19 — События объекта периодического конфигурируемого сканера

Событие	Режим	Тип события	Параметр информации о событии	Event-reply-info
Buf-Scan-Report-Var	Подтверждаемый или неподтверждаемый	MDC_NOTI_BUF_SCAN_REPORT_VAR	ScanReportInfoVar	—
Buf-Scan-Report-Fixed	Подтверждаемый или неподтверждаемый	MDC_NOTI_BUF_SCAN_REPORT_FIXED	ScanReportInfoFixed	—

Окончание таблицы 19

Событие	Режим	Тип события	Параметр информации о событии	Event-reply-info
Buf-Scan-Report-Grouped	Подтверждаемый или неподтверждаемый	MDC_NOTI_BUF_SCAN_REPORT_GROUPED	ScanReportInfoGrouped	—
Buf-Scan-Report-MP-Var	Подтверждаемый или неподтверждаемый	MDC_NOTI_BUF_SCAN_REPORT_MP_VAR	ScanReportInfoMPVar	—
Buf-Scan-Report-MP-Fixed	Подтверждаемый или неподтверждаемый	MDC_NOTI_BUF_SCAN_REPORT_MP_FIXED	ScanReportInfoMPFixed	—
Buf-Scan-Report-MP-Grouped	Подтверждаемый или неподтверждаемый	MDC_NOTI_BUF_SCAN_REPORT_MP_GROUPED	ScanReportInfoMPGrouped	—

Примечание — Если от объекта не собрано ни одного набора AttributeChangeSet, то отчеты о сканировании с переменными и фиксированными форматами не будут содержать ни одного набора AttributeChangeSet от этого объекта. Если ни одного набора AttributeChangeSets не собрано, то по истечении периода отправляется пустой отчет о событии сканирования.

Все стили отчетов о событиях, перечисленные в таблице 19, являются буферизованными эквивалентами их небуферизованных аналогов, описанных в 6.3.9.5.5. Первое отличие заключается в том, что сканер буферизует данные на протяжении интервала отчетов и посылает единственное сообщение в конце интервала. Второе отличие заключается в том, что каждый отчет содержит одни и те же объекты и атрибуты, независимо от того, изменились ли их значения.

6.3.9.6.5 Прочие службы периодического конфигурируемого сканера

На сегодняшний день службы SET или GET периодического конфигурируемого сканера, определенные в настоящем стандарте, не используются, однако могут существовать специализации приборов, определенные в стандартах серии ИСО/ИИЭР 11073-104zz, или местные определения, которые их задействуют.

6.4 Правила расширения информационной модели

Информационная модель может быть расширена при реализации, используя дополнительные атрибуты объектов, определенных в настоящем стандарте, которые определены в ИСО/ИИЭР 11073-10201:2004 [B17].

Другим возможным расширением может стать использование местных (например, специфичных для производителя) атрибутов объектов и/или методов для объектов, определенных в настоящем стандарте. Местные атрибуты должны идентифицироваться путем присвоения номенклатурных кодов из местного пространства нумерации (0xF000 — 0xFFFF) в рамках соответствующего раздела, определенного в ИСО/ИИЭР 11073-10101 [B16].

Могут быть определены классы, специфичные для производителя. Объекты, специфичные для производителя, могут создаваться на основе этих классов или любых классов, определенных в этой серии стандартов.

Реализация системы менеджера должна полностью обрабатывать сообщение, пропуская любые неизвестные атрибуты (например, атрибуты, специфичные для производителя) и игнорируя значения, присвоенные таким атрибутам, без ошибок протокола. При необходимости реализация может регистрировать наличие таких атрибутов (например, в файлах журналов).

7 Сервисная модель персонального медицинского прибора

7.1 Общие положения

Сервисная модель определяет концептуальные механизмы служб обмена данными. Такие службы отображаются на сообщения, которыми обмениваются агент и менеджер. Сообщения протокола в рамках серии стандартов ИСО/ИИЭР 11073 определены на языке ACH.1. Сообщения, описанные в настоящем стандарте, могут сосуществовать с сообщениями, описанными в других стандартных профилях, определенных в серии стандартов ИСО/ИИЭР 11073.

Сообщения протокола структурируются следующим образом.

- Структура кадра протокола верхнего уровня отделяет сообщения команд, связанных с управлением соединением (сообщения ассоциации), от сообщений верхнего уровня, связанных с объектами (передача данных и служб).

- Структура кадра верхнего уровня, в частности, содержит тип сообщения и поле длины.

- При использовании MDER протокол позволяет агентам хранить заранее определенные шаблоны передачи и изменять только фиксированные места шаблона, меняя части перед отправкой.

- Для управления или указания поведения и/или настроек часто используются установки битов. Во многих случаях не все биты назначены и/или указаны как зарезервированные. Для будущей совместимости менеджеры должны игнорировать установки любых зарезервированных или не назначенных битов (см. А.2.1).

7.2 Служба ассоциации

Служба ассоциации управляет ассоциацией между агентом и менеджером. Следующие сообщения являются частью службы ассоциации:

- Запрос ассоциации устанавливает соединение верхнего уровня поверх существующего транспортного соединения.

- Если соединение двунаправленное, то ответ на запрос ассоциации сообщает об обработке этого запроса.

- Запрос на разъединение корректно завершает ассоциацию верхнего уровня.

- Если соединение двунаправленное, то ответ на разъединение подтверждает завершение ассоциации верхнего уровня.

- Прекращение ассоциации инициирует немедленное завершение ассоциации верхнего уровня без ответа. Обычно это сообщение пересылается в результате сбоя.

7.3 Службы доступа к объектам

Службы доступа к объектам используются для доступа к информационным объектам, определенным в информационной модели предметной области. В частности, эти службы используются для предоставляемых агентом функций отчетов и доступа к данным.

Поддерживаются следующие базовые службы доступа к объектам:

- Служба GET: используется менеджером для получения от агента значений объекта MDS и атрибутов объекта PM-store. Список атрибутов объекта MDS указан в 6.3.2.3, а список атрибутов объекта PM-store — в 6.3.7.3.

- Служба SET: используется менеджером для присвоения значений атрибутов объекта агента. В настоящее время службу SET поддерживают только объекты сканера (см. 6.3.9.3.6).

- Служба EVENT REPORT: используется агентом для передачи менеджеру обновлений конфигурации и результатов измерений. Список отчетов о событиях указан в 6.3.2.5, 6.3.7.5, 6.3.9.5.5 и 6.3.9.6.5.

- Служба ACTION: используется менеджером для вызова действий (или методов), поддерживаемых агентом. Примером может служить действие MDS-Data-Request, используемое для запроса результатов измерений от агента. Список методов указан в 6.3.2.4 и 6.3.7.4.

Доступ к объектам агента с помощью запроса Get должен считаться недействительным до тех пор, пока не выполнено одно из следующих условий.

- Агент находится в состоянии «Выполнение», а служба GET ссылается на объект MDS или дескриптор объекта, объявленный во время конфигурирования.

- Агент находится в состоянии «Ассоциирован», а служба GET ссылается на объект MDS.

Менеджер, получающий от агента подтверждаемый отчет о событии, должен возвратить сообщение rors-cmip-confirmed-event-report или соответствующее сообщение об ошибке roer с подходящим кодом возврата.

Если запрос на подтверждаемое действие получен агентом, который не поддерживает это действие, агент должен возвратить сообщение об ошибке (roer) со значением ошибки no-such-action (такого действия нет). В случае ошибки выполнения подтверждаемого действия ее можно указать с помощью возврата сообщения об ошибке (roer) с соответствующим значением ошибки, и, при необходимости, дополнительная информация об ошибке может быть добавлена в поле параметра, используя один из кодов возврата, указанный в разделе кодов возврата.

Если к какому-либо объекту агента осуществляется доступ с помощью подтверждаемой службы доступа, то в любой момент времени допустимо существование не более одной неквитированной подтверждаемой службы доступа к объектам.

7.4 Специфичное применение служб доступа к объектам EVENT REPORT для персональных медицинских приборов

7.4.1 Общие положения

Для агента служба EVENT REPORT является основным механизмом отправки отчетов о результатах измерений и конфигурациях. В настоящем стандарте отчеты о событиях являются свойствами объектов MDS и сканеров. Такие специфичные отчеты о событиях могут иметь разные формы и свойства, указанные в 7.4.2—7.4.7.

7.4.2 Подтверждаемые и неподтверждаемые отчеты о событиях

Отправитель отчета о событии может выборочно запросить от получателя подтверждение. Если подтверждаемый отчет о событии используется определенным объектом, в любой момент времени допустимо не более одного неквитированного подтверждаемого отчета о событии, исходящего от этого объекта.

7.4.3 Отчет о событиях конфигурации

7.4.3.1 Общие положения

Подразделы 7.4.3.2—7.4.3.4 описывают конфигурации, отчеты о событиях конфигураций и специализации приборов, используемые для описания объектов агента.

7.4.3.2 Конфигурация прибора агента

Набор объектов и атрибутов агента, которые не входят в MDS, обозначает конфигурацию прибора агента и ассоциируется со значением Dev-Configuration-Id (см. таблицу 3). Если агент обладает несколькими конфигурациями прибора, присвоенные значения Dev-Configuration-Id должны быть локально уникальными. На протяжении срока существования ассоциации конфигурация агента должна оставаться постоянной, то есть набор объектов должен оставаться неизменным. Однако агент может добавить объекту новые атрибуты или изменить значения атрибутов в соответствии с 7.4.3.3. Агент, которому требуется другая конфигурация, должен завершить ассоциацию и установить новую ассоциацию с требуемой конфигурацией.

Объект MDS не рассматривается как часть конфигурации. Менеджер, выполняющий повторную ассоциацию с агентом, предоставляющим то же самое значение Dev-Configuration-Id, не может рассчитывать на совпадение значений атрибута MDS. Например, агент может сбросить бит manager-set-time, поскольку время на его часах уже установлено.

7.4.3.3 Отчет о событиях конфигураций

Конфигурация, которую агент желает использовать на время ассоциации с менеджером, указывается в сообщении запроса ассоциации в значении атрибута Dev-Configuration-Id поля dev-config-id. Если менеджеру еще не известна конфигурация прибора агента (например, на основе предыдущего этапа ассоциации), менеджер запрашивает конфигурацию прибора агента. Даже если менеджеру известна конфигурация прибора агента, менеджер может запросить переход в состояние «Конфигурирующий», чтобы проверить атрибуты объекта MDS перед принятием решения о допустимости ассоциации. Агент сообщает менеджеру свою конфигурацию с помощью отчета о событиях конфигурации. Отчет описывает все объекты конфигурации прибора агента вместе с ассоциированным значением Dev-Configuration-Id. На протяжении существования ассоциации конфигурация агента остается неизменной с точки зрения количества объектов. Если агент намеревается использовать другую конфигурацию или желает изменить существующую конфигурацию путем добавления или удаления объектов, агент должен завершить ассоциацию и инициировать новую ассоциацию с новой конфигурацией.

Для каждого объекта кроме MDS отчет о событии конфигурации должен содержать статические атрибуты, а также динамические атрибуты, используемые объектом. Данные атрибуты указываются в списке структур ConfigObject (A.11.5). Значение атрибута Handle указывается в поле obj-handle структуры ConfigObject и не включается в список атрибутов attribute-list структуры ConfigObject. Наблюдаемые атрибуты объектов не должны включаться в структуру ConfigObject. Значения наблюдаемых атрибутов передаются в последующих отчетах о событиях сканирования (см. 7.4.5 и 7.4.6). Агент, находящийся в состоянии «Выполнение», может добавлять новые атрибуты в объект или изменять динамические значения атрибутов, не отправляя новую конфигурацию, даже если данный атрибут первоначально отсутствует в стандартной конфигурации, определенной в некоторой специализации прибора ИСО/ИИЭР 11073-104zz.

Изменения любых значений атрибутов объектов Metric и Scanner должны сообщаться менеджеру в отчетах о событиях сканирования перед отправкой отчетов о событиях, зависящих от этих значений (например, атрибут scan-handle-attr-val-map и отчет о событии в групповом формате, или атрибут единицы измерения unit-code и наблюдаемое значение). Сведения об изменениях значений любых не статических атрибутов объектов PM-store или MDS могут передаваться менеджеру в отчетах о событиях

по усмотрению агента. Добавление новых атрибутов возможно только с помощью отчета о событиях переменного формата (дополнительные сведения о форматах отчетов о событиях см. в 7.4.5). Изменяемые значения атрибутов могут в зависимости от конфигурации использовать переменные, фиксированные или групповые отчеты о событиях.

Примечание — Менеджер всегда может опросить PM-объекты и объект MDS с помощью службы GET, поэтому агенту не обязательно отправлять изменения или дополнения этих объектов в отчетах о событиях сканирования. Однако, если приложение агента считает, что менеджер должен обладать этой информацией в момент изменения, агент может уведомить менеджера о ней с помощью отчета о событии сканирования.

Изменения, вносимые в текущую конфигурацию (стандартную или расширенную), имеют силу только в течение данной ассоциации и не считаются постоянными изменениями конфигурации. Следовательно, атрибут Dev-Configuration-Id представляет конфигурацию, согласованную во время конфигурирования. В последующих ассоциациях, когда указывается ранее использованное значение атрибута Dev-Configuration-Id, ссылочная конфигурация не содержит каких-либо изменений, внесенных во время предыдущей ассоциации. Постоянные изменения конфигурации могут вноситься только после установления новой ассоциации и указания другого значения Dev-Configuration-Id вместе с желаемой новой конфигурацией во время конфигурирования.

Менеджер использует информацию о конфигурации для создания эквивалентной модели информации агента. Затем данная информация обновляется агентом по мере сбора измерений.

7.4.3.4 Специализации приборов

Специализации приборов ИСО/ИИЭР 11073-104zz определяют обязательные объекты и атрибуты, которые должны существовать в конфигурации агента. Далее, каждая специализация определяет обязательные элементы (например, обязательные действия и методы) сервисных и коммуникационных моделей, которые должны поддерживаться агентом, реализующим соответствующую специализацию.

7.4.3.5 Профили

7.4.3.5.1 Общие положения

Профиль дополнительно ограничивает объектную, сервисную и коммуникационную модель специализации. Некоторые специализации написаны с целью охвата обширной категории типов устройств. Они определяют общие объекты, которые полезны для группы устройств, и другие объекты, специфичные для ограниченного числа устройств. При дальнейшем профилировании специализации приборов стандарт предоставляет дополнительные рекомендации по обязательным, необязательным и ненужным объектам. Кроме того, профиль может определять меньшие размеры блоков данных прикладного протокола (APDU), специфичные стандартные конфигурации, а также помогает менеджеру точно знать, какой тип прибора реализован (например, шагомер или велотренажер).

Ожидается, что профиль будет идентифицироваться по названию и номенклатурному значению. Например, при реализации стандарта концентратора активности (ИИЭР 11073-10471 [B11]) разработчик может объявить о следовании профилю датчика дыма или профилю датчика угарного газа.

Реализация профиля датчика дыма или профиля угарного газа потребует выбора из тех объектов, служб и средств связи, которые определены в специализации. Устройство, которое декларирует соответствие этим профилям, соблюдает требования, указанные в подразделах профилей информационной модели предметной области, сервисных и коммуникационных моделей.

7.4.3.5.2 Ограничения информационной модели предметной области

Объекты, обязательные или условно обязательные в специализации, остаются обязательными или условно обязательными в профиле. Объекты, которые не обязательны в специализации, могут остаться необязательными или стать обязательными в профиле. Профиль не предназначен для выполнения следующих действий:

- определение дополнительных объектов;
- расширение объектов с помощью дополнительных атрибутов и диапазонов значений атрибутов;
- расширение условий, ранее определенных в специализации.

7.4.3.5.3 Ограничения сервисной модели

Ожидается, что профиль не изменяет и не расширяет службу ассоциации или службы доступа к объектам. В частности, профиль не предназначен для выполнения следующих действий:

- расширение набора событий, описываемых специализацией;
- расширение набора методов, описываемых специализацией.

Отчет о событиях конфигурации, имеющий отношение к конкретному профилю, может отличаться от других отчетов о событиях конфигурации, указанных в специализации приборов.

7.4.3.5.4 Ограничения коммуникационной модели

Профиль может уменьшить максимальный размер блока APDU по сравнению с максимальным размером, определенным в специализации приборов.

7.4.3.6 Типы конфигурации

Для уменьшения размеров сообщений передачи настоящий стандарт предоставляет возможность эффективного информирования менеджера о конфигурации агента. Существуют два типа конфигураций: стандартная и расширенная.

7.4.3.6.1 Стандартная конфигурация

Стандартная конфигурация — конфигурация, описанная в одной из специализаций ИСО/ИИЭР 11073-104zz и имеющая значение атрибута Dev-Configuration-Id, присвоенное из диапазона от standard-config-start до standard-config-end включительно. Такой диапазон имеет 100 зарезервированных идентификаторов для каждой специализации ИСО/ИИЭР 11073-104zz и содержит значения от zz x 100 до zz x 100 + 99 включительно. Например, диапазон 1500—1599 зарезервирован для ИИЭР 11073-10415™ [B7]. Все неиспользуемые значения из стандартного диапазона зарезервированы для будущего использования. Менеджер, обнаруживший такое зарезервированное значение, должен рассматривать его как соответствующее неопознанной стандартной конфигурации и обработать согласно 8.7.3.3 и 8.8.3.

Менеджер, который поддерживает одну (или несколько) специализаций прибора ИСО/ИИЭР 11073-104zz, должен оказаться способным принять все стандартные конфигурации приборов, описанные для профилей, указанных в строке Gen-4 таблицы 23 (общие спецификации соответствия). При наличии стандартных конфигураций, которые обычно применимы к поддерживаемым специализациям, менеджер должен оказаться способным принять все эти конфигурации. Каждый раз, когда агент запрашивает ассоциацию с менеджером, используя значение атрибута Dev-Configuration-Id, присвоенное стандартной конфигурации, менеджер может принять ассоциацию без запроса конфигурации агента, поскольку она уже известна. После успешной ассоциации менеджер и агент переходят в состояние «Выполнение». Как вариант менеджер может запросить агента об отправке стандартной конфигурации, чтобы перейти в состояние «Конфигурирующий» и проверить атрибуты объекта MDS перед окончательным принятием (или отклонением) агента.

Важно отметить, что по запросу приборы со стандартными конфигурациями должны отправлять свои конфигурации. Данное требование охватывает случай, когда агент устанавливает ассоциацию с менеджером, который предварительно не имел информации о стандартной конфигурации (например, версия менеджера 1.0, а специализация прибора имеет версию 2.0 или выше). Способность менеджера применять конфигурацию зависит от реализации менеджера.

Если агент использует значение Dev-Configuration-Id, присвоенное стандартной конфигурации, он должен также реализовать все дополнительные обязательные элементы (в том числе обязательные действия и методы) сервисных и коммуникационных моделей, определенные в соответствующей специализации прибора.

7.4.3.6.2 Расширенная конфигурация

Расширенная конфигурация агента не является стандартной; она может иметь отличающийся набор объектов, атрибутов и/или значений атрибутов. Агент, использующий расширенные конфигурации, должен выбрать уникальное значение атрибута Dev-Configuration-Id из диапазона от extended-config-start до extended-config-end включительно для каждой расширенной конфигурации. Во время ассоциации агент отправляет значение атрибута Dev-Configuration-Id в поле dev-config-id, чтобы идентифицировать выбранную конфигурацию агента на период существования ассоциации. Если менеджер уже знает эту конфигурацию, поскольку она уже загружена при помощи программы установки или агент ранее уже устанавливал ассоциацию с менеджером, то менеджер должен возвратить сообщение о принятии конфигурации без необходимости пересылки дальнейшей информации о конфигурации. Но если менеджер не знаком с конфигурацией агента, он должен ответить сообщением accepted-unknown-config, после чего агент должен передать информацию о своей конфигурации путем отправки отчета о событии конфигурации. Подробная информация о процедурах ассоциирования и конфигурирования приведена в 8.7 и 8.8. Как только менеджер получает конфигурацию, агент может передать результаты измерений. В целях экономии времени ассоциации агенту следует систематически использовать Dev-Configuration-Id для последующих ассоциаций. Отсюда вытекают следующие два последствия.

a) Один и тот же Dev-Configuration-Id не должен использоваться агентом для последующих ассоциаций, когда необходимо идентифицировать другую конфигурацию устройства.

b) Агенту необходимо использовать то же значение для Dev-Configuration-Id в последующих запросах на ассоциацию с менеджером, чтобы обозначить ту же самую конфигурацию прибора.

В отличие от стандартных конфигураций, два агента с одинаковым расширенным значением Dev-Configuration-Id не обязательно представляют одну и ту же конфигурацию. Менеджер должен различать расширенные конфигурации в пределах каждого агента. Для различия расширенных конфигураций агента может использоваться атрибут System-Id, поскольку он обязателен, должен быть уникальным и высылается во время ассоциации, однако вместо этого могут использоваться другие методы (например, передача сведений о производителе/модели/серийном номере) при условии, что они не приведут к использованию менеджером некорректной конфигурации агента.

В принципе агент с расширенной конфигурацией поддерживает ноль, одну или несколько специализаций приборов, определенных в спецификациях ИСО/ИИЭР 11073-104zz. Агент, поддерживающий одну и более специализаций приборов, должен реализовать все обязательные элементы и допустимый выбор условно обязательных элементов (в том числе объекты, атрибуты, действия и методы) из числа указанных в соответствующих спецификациях.

7.4.4 Передача результатов измерений, инициируемая агентом и менеджером

Передача результатов измерений, инициируемая агентом, реализуется агентом, например, как результат новых проведенных измерений.

Передача результатов измерений, инициируемая менеджером, явно запрашивается менеджером с помощью передачи команды (например, MDS-Data-Request), требующей, чтобы агент начал или завершил передачу результатов измерений. Период времени, на протяжении которого режим отчета остается активным, зависит от возможностей агента (например, фиксированный период или всегда в течение ассоциации).

Менеджер должен поддерживать получение от агента передачи результатов измерений, инициируемой агентом или менеджером. Агент может поддерживать генерирование передачи результатов измерений, инициируемой одним из двух или обоими участниками обмена.

В обоих случаях передача данных измерений, а также содержания сегментов PM-segment, осуществляется с помощью отчетов о событиях, содержащих наборы изменений атрибутов и/или добавлений атрибутов. Отчеты о наборах изменений атрибутов и/или добавлений атрибутов организуются в одну или несколько структур ObservationScan. Менеджер применяет изменения ObservationScan к соответствующим объектам целиком, не придавая семантического значения порядку появления атрибутов в этих структурах ObservationScan конкретного объекта.

Примечание 1 — Пример 1. Если структура ObservationScan объекта температуры содержит набор значений атрибутов, представляющих температуру, и атрибут Metric-Id, указывающий часть тела, корректная семантическая интерпретация состоит в том, что это одно измерение с соответствующими значениями. Если структура ObservationScan определяется на основе объекта RT-SA, содержащего поток значений температуры и указание части тела, правильная интерпретация подразумевает, что эта часть тела относится ко всему потоку значений температуры. Если структура ObservationScan содержит набор значений атрибутов, представляющих код единицы измерения и часть тела (оба атрибута динамические), правильная интерпретация подразумевает, что новые значения будут применяться к следующему полученному результату наблюдений (предполагается, что эти два динамических значения в дальнейшем не обновляются).

Примечание 2 — Пример 2. Если структура ObservationScan содержит только атрибут А, за которым следует структура ObservationScan с измерением температуры, и атрибут А — наблюдаемый (например, состояние измерения), соответствующее значение НЕ может применяться к измерению температуры. Если атрибут А динамический (например, код единицы измерения), соответствующее значение будет применяться к измерению температуры. Если атрибут А и измерение температуры находились в одной структуре ObservationScan, значение атрибута А будет применяться к температуре независимо от квалификатора атрибута А.

7.4.5 Переменный, фиксированный и групповой форматы отчетов о событиях

Для передачи отчетов о событиях можно использовать три формата: переменный, фиксированный или групповой. На рисунке 7 показана связь между этими форматами сообщений.

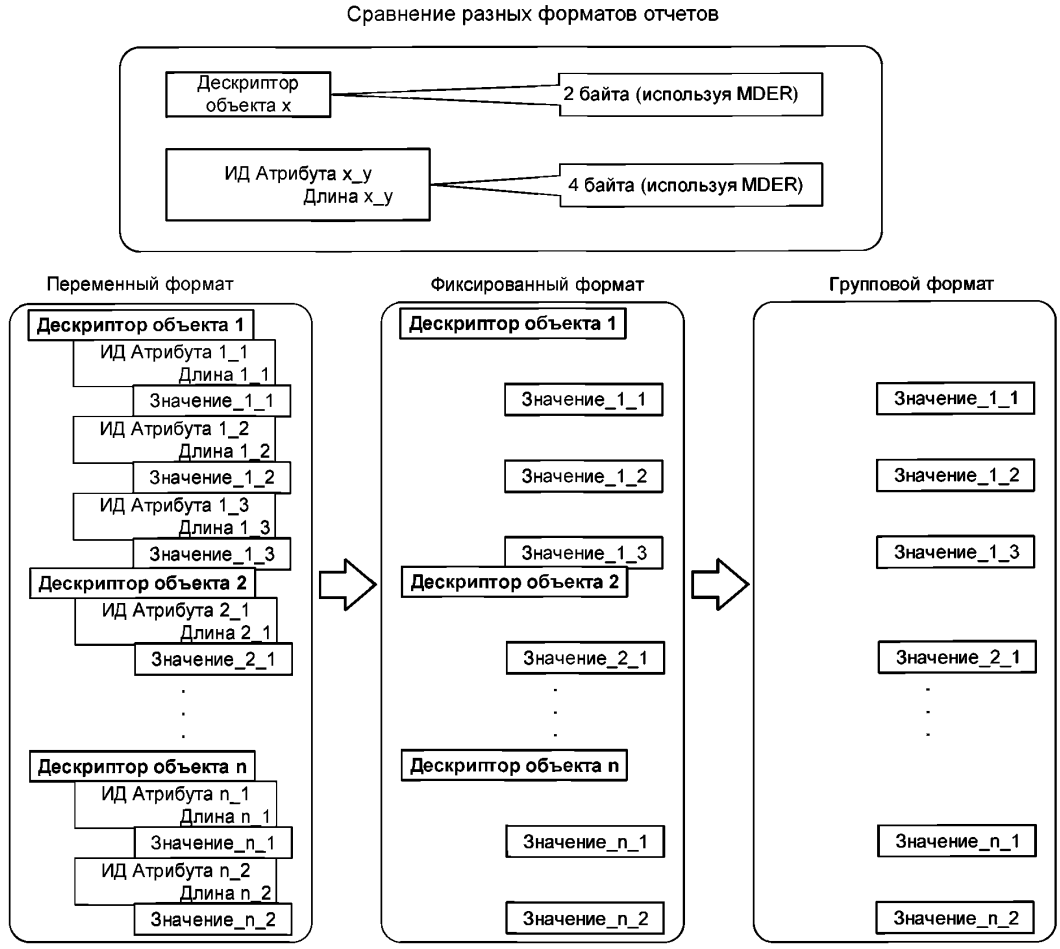


Рисунок 7 — Связи между переменным, фиксированным и групповым форматами

Отчет о событиях, использующий переменный формат, явно определяет каждый сообщаемый атрибут с помощью добавления в сообщение поля идентификации атрибута, длины значения и самого значения. Данный подход обеспечивает гибкость добавления различных наборов атрибутов в каждый отчет о событиях, но увеличивает издержки, связанные с сообщениями.

Отчет о событиях, использующий фиксированный формат, оптимизирует передачу данных с помощью определения конкретного списка передаваемых атрибутов и порядка их следования в сообщении. Это определение включено в атрибут Attribute-Value-Map, соответствующий объекту, и содержит в том числе идентификаторы атрибутов и длины значений атрибутов. Специфичный выбор добавляемых атрибутов зависит от того, какие атрибуты могут изменять свои значения. Например, одна модель весов может передавать результаты измерений веса и метки времени, а другая — результаты измерений веса, статус измерений и метки времени. В первом случае структура attribute-value-map будет содержать идентификатор наблюдаемого атрибута веса (MDC_ATTR_NU_VAL_OBS_SIMP) и его длину (4 байта), за которым следуют идентификатор атрибута метки времени (MDC_ATTR_TIME_STAMP_ABS) и его длина (8 байт). Второй случай похож на первый, однако будут добавлены идентификатор атрибута для статуса измерения (MDC_ATTR_MSMT_STAT) и его длина (2 байта). Атрибут Attribute-Value-Map должен быть определен и передан менеджеру до начала передачи отчета о событии в фиксированном формате. Когда агент передает данные в отчете о событии, имеющем фиксированный формат, он должен сообщать дескриптор объекта и значения атрибутов в том же порядке и с теми же размерами, что указаны в атрибуте Attribute-Value-Map. Таким образом можно избежать издержек, связанных с переда-

чей идентификации и длины атрибутов в каждом отчете о событиях. При получении отчета о событиях в фиксированном формате менеджер использует дескриптор объекта для извлечения ранее предоставленного атрибута Attribute-Value-Map, чтобы получить информацию о способе извлечения данных. Например, в первом ранее описанном случае менеджер знает, что наблюдение веса является первым элементом отчета о событиях с фиксированным форматом и этот элемент имеет длину 4 байта, поэтому его можно извлечь в атрибут Simple-Nu-Observed-Value, а остальные 8 байт — в Absolute-Time-Stamp. Порядок следования этих элементов определяется порядком перечисления идентификаторов атрибутов в атрибуте Attribute-Value-Map. Агент контролирует порядок следования и сообщает его менеджеру с помощью атрибута Attribute-Value-Map.

Групповой формат отчета о событиях дополнительно оптимизирован с помощью определения формата сообщения, содержащего один или несколько объектов, в атрибуте Scan-Handle-Attr-Val-Map объекта сканера. Оно аналогично определению Attribute-Value-Map, однако атрибут Scan-Handle-Attr-Val-Map позволяет агенту предоставлять сведения одновременно о нескольких объектах с помощью ссылки на другие дескрипторы и атрибуты объектов. Данный атрибут должен быть определен до начала передачи отчета о событиях в групповом формате. Агент, передающий данные в групповом формате отчета о событиях, должен сообщать дескриптор объекта сканера вместе со значениями атрибутов объектов сканера в том же порядке и с теми же размерами, что указаны в атрибуте Scan-Handle-Attr-Val-Map. Таким образом можно избежать издержек, связанных с отправкой дескрипторов объектов сканера, идентификацией их атрибутов и длины данных в каждом отчете о событиях. При получении отчета о событии в групповом формате менеджер использует дескриптор объекта сканера для извлечения ранее переданного атрибута Scan-Handle-Attr-Val-Map, чтобы получить информацию о способе извлечения данных.

Менеджер должен поддерживать отчеты о событиях с переменным или фиксированным форматом, при этом менеджер, поддерживающий сканеры, должен также поддерживать групповой формат отчетов о событиях. Агент может поддерживать любой или все отчеты о событиях с переменным, фиксированным и групповым форматами. Менеджер определяет, какой из форматов может использовать агент, проверяя атрибуты Attribute-Value-Map объектов или атрибут Handle-Attr-Val-Map объектов сканера.

7.4.6 Отчеты о событиях с одним и несколькими лицами

Агенты, созданные для работы в среде, где данные могут собираться от нескольких лиц, могут использовать отчет о событиях с несколькими лицами, чтобы передавать все данные от всех лиц в одном отчете о событии. Если такая функциональность не требуется, агент может использовать отчет о событии с одним лицом, чтобы снизить издержки.

Менеджер должен поддерживать отчеты о событиях с одним лицом и с несколькими лицами. Агент может поддерживать только один из этих вариантов, а также оба варианта отчета о событиях с одним и несколькими лицами. В А.11.5 описаны форматы отчетов о событиях с одним или несколькими лицами.

7.4.7 Временно хранящиеся результаты измерений

Агент может по своему усмотрению хранить небольшое число измерений в локальной памяти, пока он не соединен с системой менеджера (т. е. иметь временно хранимые измерения). Когда после этого агент может установить соединение с менеджером, все ранее сохраненные измерения передаются менеджеру.

Примечание — Типичным примером временно хранящихся измерений служат весы: новые измерения выполняются редко. Весы не соединены с менеджером, и их питание отключается после измерения, чтобы не ждать неопределенное время соединения с менеджером и потреблять электроэнергию.

Для поддержки временно хранящихся измерений система агента должна вести себя следующим образом:

- как временно хранящиеся результаты измерений поддерживаются только те объекты Metric, которые не являются объектами RT-SA (например, объекты Numeric и Enumeration);
- для временно хранящихся результатов измерений требуется использовать атрибуты меток времени (например, Date-and-Time, Relative-Time или HiRes-Relative-Time);
- агент не должен передавать временно хранящиеся результаты измерений, если известно, что метка времени не точна (например, если база, используемая для меток времени значений, изменилась в период между измерениями на величину, существенную для соответствующего типа измерения), кроме случаев, когда имеется подходящий параметр Date-and-Time-Adjustment в начале отчета о событиях;
- временно хранящиеся результаты измерений являются частью любого из следующих механизмов отчета о событии: инициируемые агентом или менеджером; фиксированный или переменный форматы; одно или несколько лиц;

- после передачи временно хранящихся результатов измерений менеджеру агент должен удалить их из локальной памяти. Агент должен удостовериться, что владение результатами измерений успешно передано менеджеру с помощью подтверждаемых отчетов о событиях;

- для ограничения объема данных, передаваемых данным механизмом, агент должен предоставить не более 25 временно хранящихся результатов измерений в любом одном отчете о событии. Если требуется хранение более 25 измерений, следует использовать механизм объекта PM-store для архивирования результатов измерений;

- временно хранящиеся результаты измерений должны передаваться в порядке получения (FIFO).

8 Коммуникационная модель

8.1 Общие положения

В общем случае предполагается топология, при которой один или несколько агентов обмениваются данными с менеджером, используя двухточечные соединения. Если менеджеру требуется поддерживать несколько агентов одновременно (например, используя пикосеть Bluetooth), он должен иметь возможность обрабатывать несколько индикаторов соединения и отдельные ассоциации для каждого из этих агентов.

Любой агент, поддерживающий несколько модальностей (специализаций приборов), может выбрать создание одного соединения и одной ассоциации с менеджером или создать несколько соединений и ассоциаций с менеджером (например, по одному экземпляру для каждой модальности). Но если агент выбирает создание нескольких соединений и ассоциаций, то экземпляры объектов в различных ассоциациях должны быть полностью независимыми, как если бы ассоциации были созданы различными приборами. Например, экземпляры объекта MDS для каждого соединения и ассоциации должны действовать в качестве отдельных независимых агентов.

8.2 Контекст системы

Коммуникационный профиль, определенный в настоящем стандарте, учитывает специфичные требования агентов и менеджеров персональных медицинских приборов, которые обычно используются в мобильной среде или на дому. В части служб и свойств, которые должны предоставляться транспортными уровнями, приняты описанные ниже допущения. Кроме того, также охвачен контекст системы за пределами этого коммуникационного профиля (то есть другие не персональные медицинские приборы/поддерживающие функциональность прикладного уровня) и его связь с допущениями, принятыми для транспортных уровней.

Настоящий стандарт предполагает, что технологии передачи данных обладают широким спектром свойств, и в общих чертах рассматривает транспортные технологии, чтобы позволить использование и применение их нативных возможностей. Если этих возможностей транспорта недостаточно, то для получения необходимых характеристик добавляется «промежуточный уровень». См. рисунок 8.



Рисунок 8 — Контекст системы

В настоящем стандарте используется понятие «тип» для группировки и дифференцирования служб, предлагаемых доступными технологиями транспорта, профилированными для использования в серии стандартов ИСО/ИИЭР 11073. А именно в этой серии выделяются следующие типы транспортных профилей:

- Тип 1. Транспортные профили, содержащие «надежные» и «лучшие из возможных» транспортные службы, в которых должны быть один или несколько виртуальных каналов «надежных» транспортных служб и ноль или более виртуальных каналов «лучших из возможных» транспортных служб;

- Тип 2. Транспортные профили, содержащие только однонаправленную транспортную службу.

- Тип 3. Транспортные профили, содержащие только «лучшие из возможных» транспортные службы, в которых должен быть один или несколько виртуальных каналов «лучших из возможных» транспортных служб.

Причина, по которой типам транспортных профилей придается большое значение, состоит в том, что предлагаемые ими различные транспортные службы влияют на некоторую функциональность верхнего уровня. В частности, они влияют на реализацию механизма подтверждаемых служб, описанных в настоящем стандарте. Настоящий стандарт предназначен для использования только с транспортными профилями типа 1.

Более полное описание различных типов транспортных профилей и вариантов их взаимодействия с подтверждаемыми и неподтверждаемыми механизмами служб см. в приложении D.

8.3 Коммуникационные характеристики

8.3.1 Общие положения

В настоящем стандарте должны использоваться транспортные профили типа 1.

Согласно настоящему стандарту каждый прибор должен поддерживать основной виртуальный канал. Основной виртуальный канал должен являться надежным виртуальным каналом (то есть надежной транспортной службой) из транспортного профиля типа 1. См. рисунок 9.

Основной виртуальный канал должен использоваться для следующих целей.

- Все сообщения, относящиеся к процедуре ассоциации:
  - aare, aarq, rlre, rlrq, abrt.
- Все сообщения, относящиеся к подтверждаемому сервисному механизму:
  - prst.ro**iv**-cmip-confirmed-**action**, prst.ro**iv**-cmip-confirmed-**event-report**, prst.ro**iv**-cmip-**get**, prst.ro**iv**-cmip-confirmed-**set**;
  - prst.ro**rs**-cmip-confirmed-**action**, prst.ro**rs**-cmip-confirmed-**event-report**, prst.ro**rs**-cmip-**get**, prst.ro**rs**-cmip-confirmed-**set**.
- Все сообщения, относящиеся к условиям сбоя или аномалии:
  - roer, rorj.

Согласно настоящему стандарту каждый прибор может поддерживать один или несколько вспомогательных виртуальных каналов. Каждый вспомогательный виртуальный канал может являться надежным или лучшим из возможных для транспортного профиля типа 1.

Основной виртуальный канал или любые вспомогательные каналы можно использовать для сообщений, связанных с неподтверждаемым сервисным механизмом.

- prst.ro**iv**-cmip-**action**, prst.ro**iv**-cmip-**event-report**, prst.ro**iv**-cmip-**set**

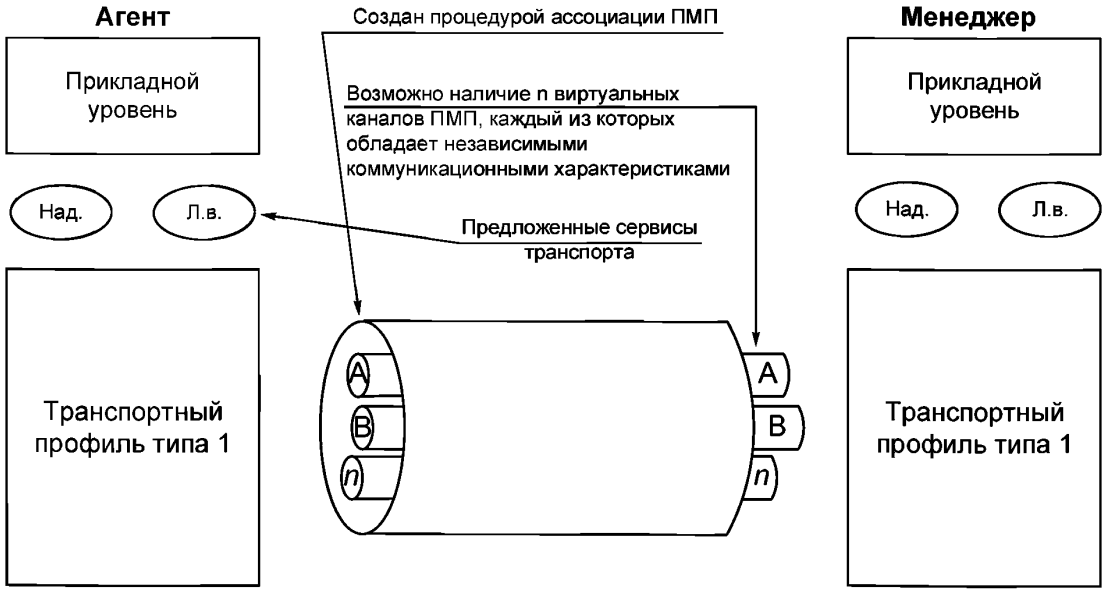


Рисунок 9 — Общая коммуникационная модель

Обычно термин «метаданные» означает данные о данных. В контексте коммуникационных характеристик, описанных в ИСО/ИИЭР 11073-20601, термин «метаданные» используется для обозначения вспомогательной информации или данных, относящихся к блоку APDU. Можно привести следующие примеры:

- специфический транспортно-технологический адрес для передачи блока APDU определенному агенту или менеджеру;
- надежность и/или задержка, необходимая определенному блоку APDU;
- размер или длина определенного блока APDU.

Некоторые метаданные описывают коммуникационные характеристики, представленные как отдельное значение, охватывающее широкий диапазон возможных значений. Для вышеприведенных общих примеров метаданных возможны следующие частные случаи:

- APDU-metadata.address (для конечной точки USB) = 1 — 1023;
- APDU-metadata.address (для сети IPv4) = 0.0.0.0 — 255.255.255.255;
- APDU-metadata.size = 1 — 64512.

Другие метаданные описывают коммуникационные характеристики, представленные как отдельное значение, но обладающие только несколькими дискретными возможными значениями. Для вышеприведенных общих примеров метаданных возможны следующие частные случаи:

- APDU-metadata.latency = (10 мс | 100 мс | 1 с | 10 с);
- APDU-metadata.reliability = (высокий | средний | низкий);
- APDU-metadata.bandwidth = (100 бит/с | 1 Кбит/с | 10 Кбит/с | 100 Кбит/с | 1 Мбит/с).

В последующих подразделах описываются общие характеристики (см. 8.3.2) и уникальные характеристики надежных (см. 8.3.3) и лучших из возможных (см. 8.3.4) виртуальных каналов применительно к настоящему стандарту.

8.3.2 Общие коммуникационные характеристики

Существует ряд следующих общих коммуникационных характеристик, применимых к надежным и лучшим из возможных коммуникациям:

a) Блок APDU может обрабатываться любым способом (например, частями по мере получения блока APDU или как полный блок APDU, буферизированный в памяти), однако он должен обрабатываться таким образом, как если бы это была атомарная транзакция.

b) Блоки APDU могут сегментироваться и собираться повторно во время передачи или они могут отправляться как единое целое.

c) Размер блоков APDU, отправляемых агентом менеджеру, не должен превышать 63 КБ (64 512 байт). Специфичные специализации приборов, профили или реализации могут анализировать переданные сообщения с целью определения конкретного размера реализации для приемного буфера менеджера, меньшего, чем максимальный размер блока APDU, передаваемого агентом менеджеру. Если менеджер получает блок APDU, который больше размера его приемного буфера, то ответное сообщение должно содержать код ошибки (roer) protocol-violation (нарушение протокола). Размер приемного буфера менеджера должен как минимум равняться размеру наибольшего буфера, указанному в специализациях, поддерживаемых менеджером. Ограничения размера буфера, указанные в этом и следующем пунктах перечисления, применяются ко всем блокам APDU независимо от использования стандартной или расширенной конфигурации.

d) Размер блоков APDU, отправляемых менеджером агенту, не должен превышать 8 КБ (8192 байт). Специфичные специализации устройств, профилей или реализаций могут анализировать передаваемые сообщения с целью определения конкретного размера реализации для приемного буфера агента, меньшего, чем максимальный размер блока APDU, передаваемого менеджером агенту. Агент, получивший блок APDU большего размера, должен ответить, указав код ошибки (roer) protocol-violation (нарушение протокола).

e) Общая длина блока APDU должна передаваться между коммуникационными уровнями как метаданные.

f) Коммуникационные уровни должны указывать общую длину блока APDU для его однорангового коммуникационного уровня.

8.3.3 Характеристики надежных коммуникаций

Коммуникационные технологии/методы, которые должны рассматриваться как надежные и пригодные для использования оптимизированным протоколом обмена, имеют следующие характеристики:

a) Блоки APDU должны приниматься в порядке их отправки.

b) Блоки APDU не должны содержать обнаруживаемых ошибок.

c) Блоки APDU не должны дублироваться.

d) Блоки APDU не должны быть утеряны.

e) Блоки APDU, как правило, отправляются оперативно, однако могут быть задержки вследствие повторных передач.

f) Коммуникационные уровни должны предоставлять прикладному уровню механизм, позволяющий определить, что блок APDU получен целиком.

g) Коммуникационные уровни должны предоставлять прикладному уровню механизм, позволяющий определить, что между агентом и менеджером установлено соединение.

h) Коммуникационные уровни по возможности должны предоставлять прикладному уровню механизм, позволяющий определить, что соединение прекращено или разъединено.

i) Коммуникационные уровни должны предоставлять прикладному уровню механизм, позволяющий определить, что отправка блока APDU невозможна.

j) Управление потоком между отправляющим и получающим приложением должно поддерживаться для полных блоков APDU. Нижние слои могут реализовать управление потоком для небольших (подмножество) блоков APDU.

8.3.4 Характеристики наилучших коммуникаций

Согласно оптимизированному протоколу обмена, если коммуникационные технологии не соответствуют критерию надежного канала коммуникации (см. описание выше), они именуются как лучшие из возможных. Для таких каналов типичны следующие характеристики.

a) Порядок доставки блоков APDU может не соответствовать порядку их отправки. Коммуникационный канал может нарушить порядок пакетов независимо от работы передатчика персонального медицинского прибора.

b) Блок APDU может теряться или дублироваться.

c) Скорость передачи пакетов APDU может привести к переполнению буфера получателя.

8.4 Конечные автоматы

8.4.1 Конечный автомат агента

Общая схема конечного автомата агента показана на рисунке 10.

Подробная таблица состояний агента представлена в Е.2.

Таблица 20 содержит описание каждого из этих состояний.

Описание каждого изменения состояния представлено в таблице 21.

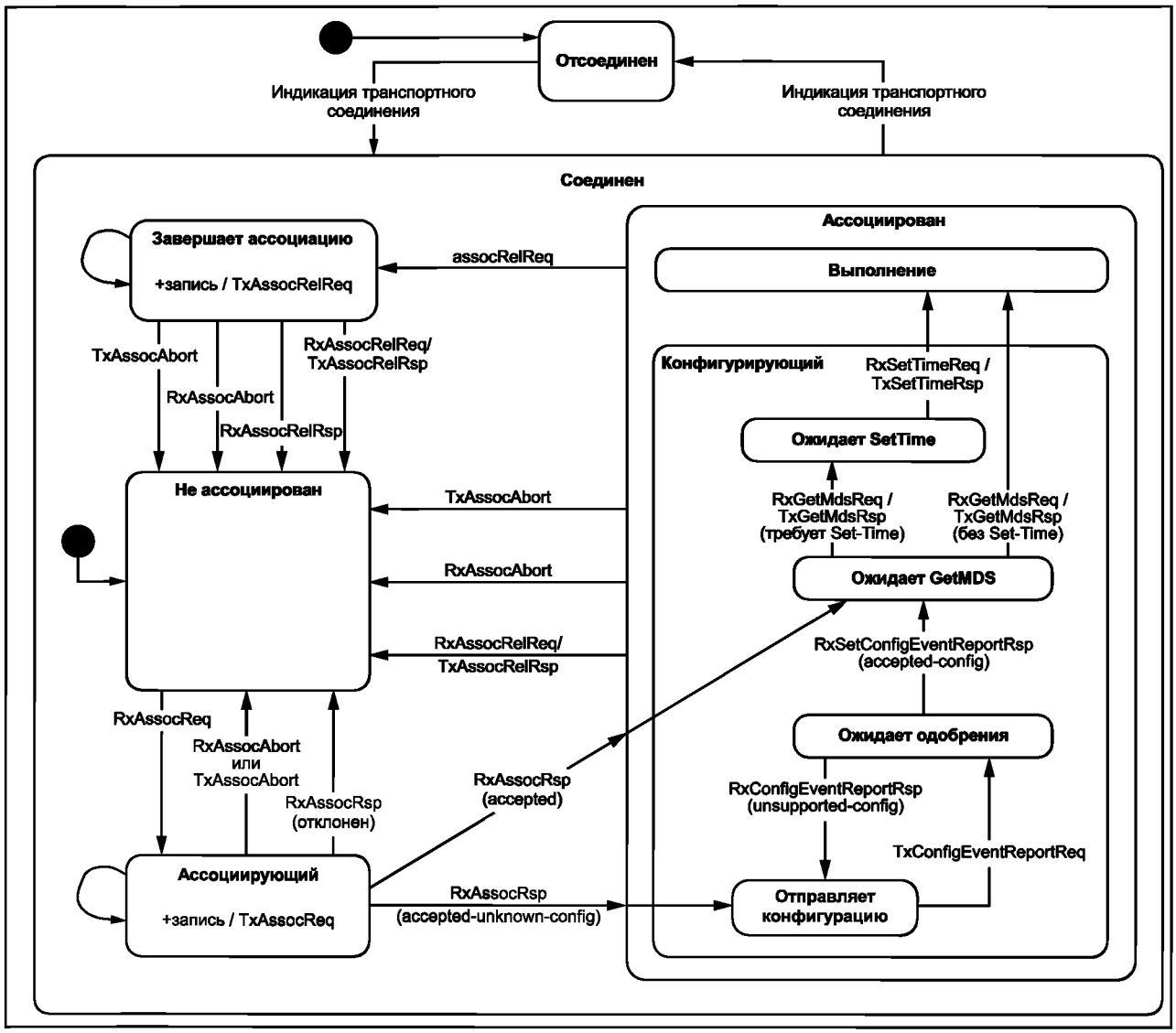


Рисунок 10 — Схема конечного автомата агента

Таблица 20 — Описание состояний агента

Состояние	Описание
Отсоединен	Предполагается, что при начальном включении агент находится в состоянии «Отсоединен», означающем отсутствие транспортного соединения между агентом и менеджером. После установления транспортного соединения можно вернуться в состояние «Отсоединен», если транспортное соединение намеренно или непреднамеренно прекращено
Соединен	После установления транспортного соединения агент получает от транспортного уровня индикацию транспортного соединения, что приводит к переходу в состояние «Соединен» (см. 8.4.3). Агент остается в состоянии «Соединен» до тех пор, пока поддерживается транспортное соединение. Изначально агент начинает работу в состоянии «Не ассоциирован» (подсостояние состояния «Соединен»)

Окончание таблицы 20

Состояние	Описание
Не ассоциирован	Агент находится в состоянии «Не ассоциирован», когда не имеет ассоциацию прикладного уровня с менеджером. Такая ситуация может возникнуть вследствие любого из следующих обстоятельств: - новое соединение только что установлено; - менеджер отклоняет запрос ассоциации; - любая из сторон завершает или разрывает активную ассоциацию в любой момент при наличии соединения. Агент остается в состоянии «Не ассоциирован», пока не определит, что должен начать ассоциирование с менеджером
Ассоциирующий	Когда агент определяет, что должен создать ассоциацию, то переходит в состояние «Ассоциирующий» и посылает менеджеру запрос ассоциации (см. 8.6). Если ассоциацию не удается создать, но возможны альтернативные параметры ассоциации, агент может попытаться связаться с помощью каждого нового набора параметров ассоциации. В случае тайм-аута агент должен продолжать попытки установления ассоциации до тех пор, пока не будет достигнуто максимальное количество повторных попыток или ассоциация не будет создана
Ассоциирован	Если менеджер определяет, что агент и менеджер используют общие версии и протоколы, он посылает агенту ответ на запрос ассоциации с параметром «accepted» (см. 8.7.3.3). Агент, получивший это сообщение, переходит в состояние «Ассоциирован» и остается в этом состоянии до тех пор, пока не отправит или не получит запрос на завершение или разрыв ассоциации. Начальное подсостояние после перехода в состояние «Ассоциирован» зависит от ответа менеджера на запрос ассоциации, в котором указано, распознана ли конфигурация агента
Выполнение	Описания процедур, возможных в состоянии «Выполнение», см. в 8.9.
Конфигурирующий	Если менеджер не распознал конфигурацию агента, то он информирует его об этом, посылая ответ на запрос ассоциации с параметром «accepted-unknown-config», чтобы указать, что ассоциация принята, но необходимо передать конфигурацию. Агент остается в состоянии «Конфигурирующий» до тех пор, пока не передаст информацию о конфигурации, получение которой менеджер подтвердит (см. 8.8) Если менеджер распознал конфигурацию агента, он информирует его об этом, посылая ответ на запрос ассоциации с параметром «accepted», чтобы перевести агента в подсостояние «Ожидает GetMDS». На протяжении TO_{get} секунд после перехода агента в подсостояние «Ожидает GetMDS» менеджер должен вызвать службу GET. Получив этот вызов, отправляет менеджеру ответ на GET. Если бит mds-time-mgr-set-time не установлен, агент сразу переходит в состояние «Выполнение». Если бит mds-time-mgr-set-time установлен, агент должен ожидать TO_{ca} секунд получения от менеджера команды действия Set-time. После получения действия Set-Time агент должен очистить бит mds-time-mgr-set-time перед отправкой подтверждения Set-Time. После отправки подтверждения Set-Time агент переходит в состояние «Выполнение». Если для передачи отчета агент использует базовое время со смещением, в вышеуказанном описании вместо действия Set-Time должно использоваться действие Set-Base-Offset-Time. В случае тайм-аута агент посылает запрос на завершение ассоциации и переходит в состояние отсутствия состояния
Завершает ассоциацию	Когда агент определяет, что должен завершить текущую ассоциацию, то переходит в состояние «Завершает ассоциацию» и посылает менеджеру запрос на завершение ассоциации (см. 8.10). В случае тайм-аута агент посылает запрос на прекращение ассоциации и переходит в состояние «Не ассоциирован»

Таблица 21 — Описание переходов состояния агента

Переход	Описание
Индикация транспортного соединения	Передача индикации транспортного соединения происходит всякий раз, когда транспортный уровень (или вспомогательный промежуточный уровень) указывает на успешное установление соединения
assocReq	Когда агент определяет, что должен попытается создать ассоциацию с менеджером, он переходит в состояние «Ассоциирующий»
RxAssocRsp (accepted или accepted-unknown-config)	Агент, пытающийся создать ассоциацию с менеджером, посылает запрос ассоциации (или несколько запросов при наличии тайм-аута) и ожидает ответа на запрос ассоциации от менеджера. После получения одобрения ассоциации агент переходит в состояние «Ассоциирован»
RxAssocRsp (отклонен)	Если после получения запроса ассоциации менеджер определяет, что не может ассоциироваться с агентом, то он посылает ответ на запрос ассоциации с кодом отклонения ассоциации (постоянного или временного), чтобы перевести агента обратно в состояние «Не ассоциирован»
RxAssocRelReq/TxAssoc-RelRsp	Если агент ассоциирован с менеджером и получает запрос на завершение ассоциации, то он отвечает ему и переходит в состояние «Не ассоциирован»
RxAssocAbort или TxAssocAbort	В любое время, когда агент и менеджер создают ассоциацию, ассоциированы или завершают ассоциацию, агент может отправить или получить сообщение о прекращении ассоциации. При возникновении данной ситуации агент переходит из текущего состояния в состояние «Не ассоциирован». Ассоциированный агент может отправить сообщение о прекращении ассоциации, чтобы сообщить менеджеру о возникновении серьезного сбоя. Оно должно передаваться в самом крайнем случае, поскольку предпочтительнее отправка запроса на завершение ассоциации, чтобы корректно перейти в состояние «Не ассоциирован». Если агент получает сообщение о прекращении ассоциации, ответ от него не требуется, поскольку это сообщение может быть получено только в случае прекращения ассоциации менеджером (например, при аварии)
assocRelReq	Агент, решивший завершить ассоциацию, переходит в состояние «Завершает ассоциацию» и посылает запрос на завершение ассоциации. Такой переход используется во время обычной последовательности отключения с помощью отправки в параметре ReleaseRequestReason кода nomal при обычных условиях или, если конфигурация агента изменилась, из-за чего необходимо завершить ассоциацию, агент отправляет в параметре ReleaseRequestReason код configuration-changed. В любом случае при следующей ассоциации агент указывает в запросе ассоциации конфигурацию, необходимую для использования, и менеджер определяет, знакома ли ему эта конфигурация
RxAssocRelRsp	Данный переход указывает на принятие запроса о завершении текущей ассоциации. Если перед этим агент послал запрос на завершение ассоциации, это означает, что получен ответ, указывающий на одобрение завершения со стороны менеджера
RxGetMdsReq/TxGetMdsR-sp	На протяжении TO_{get} секунд после перехода агента в подсостояние «Конфигурирующий»/«Ожидает GetMDS» менеджер должен вызвать службу GET, при этом агент отправляет менеджеру ответ на действие GET вместе с реализованными атрибутами объекта MDS. После этого агент переходит в состояние «Выполнение» или подсостояние «Ожидает SetTime» (зависит от значения бита mds-time-mgr-set-time). Если бит mds-time-mgr-set-time установлен, агент переходит в подсостояние «Конфигурирующий»/«Ожидает SetTime», иначе он переходит в состояние «Выполнение»
RxSetTimeReq/TxSetTim-eRsp	На протяжении TO_{ca} секунд после перехода агента в подсостояние «Ожидает Set-Time» менеджер должен отправить агенту команду действия Set-Time (или Set-Base-Offset-Time). После завершения операции настройки внутреннего времени агент отправляет менеджеру ответ и переходит в состояние «Выполнен»
Индикация транспортного разъединения	В любой момент времени агент или менеджер может разорвать транспортное соединение. Также возможна потеря соединения вследствие аварийных условий. Если получена индикация об отсоединении транспорта, агент переходит в состояние «Отсоединен»

8.4.2 Конечный автомат менеджера

Общая схема конечного автомата менеджера показана на рисунке 11. Большинство состояний и переходов симметричны элементам, описанным для агента в таблицах 20 и 21. Основные различия заключаются в следующем.

- Менеджер должен ждать в состоянии «Ожидает конфигурацию» как минимум TOconfig секунд перед отправкой сообщения о прекращении ассоциации.
- Если менеджер не принимает конфигурацию, то он должен отправить ответ на конфигурацию с указанием результата unsupported-config.
- Если менеджер принимает конфигурацию, то он должен отправить ответ на конфигурацию с указанием результата accepted-config.
- Менеджер должен вызвать службу GET не позже TOget секунд после перехода агента в подсостояние «Конфигурирующий»/«Ожидает GetMDS».
- Менеджер должен отправить агенту команду действия Set-Time (или Set-Base-Offset-Time) не позже TOca секунд после перехода агента в подсостояние «Конфигурирующий»/«Ожидает SetTime».

Подробная таблица состояний менеджера представлена в Е.4.

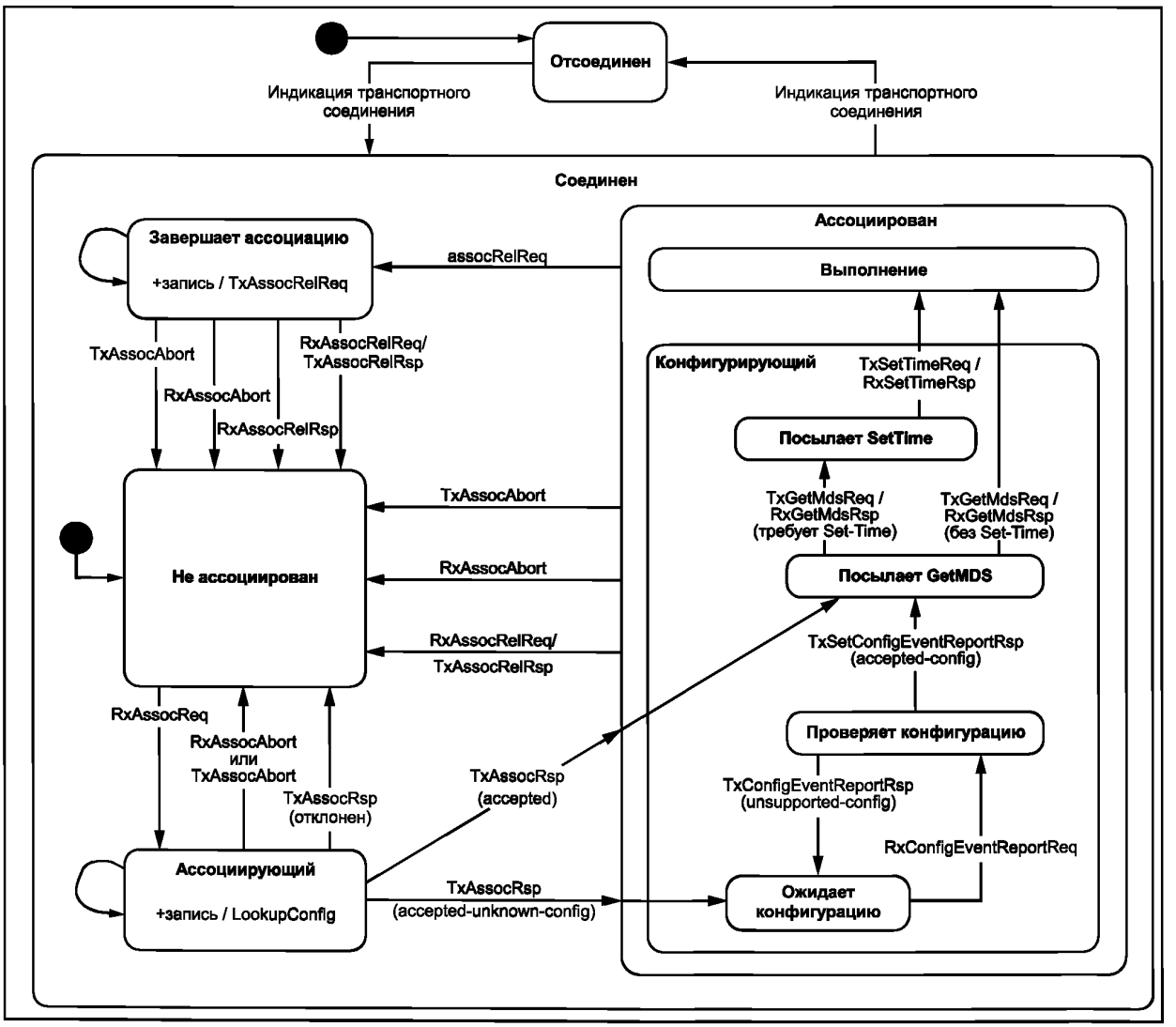


Рисунок 11 — Схема конечного автомата менеджера

8.4.3 Переменные тайм-аута

В протоколе персонального медицинского прибора есть несколько мест, где используются тайм-ауты. Существуют тайм-ауты периодов повтора и счетчики повторов (RC). Чтобы облегчить управление документом в течение длительного периода времени и упростить электронный «поиск» разных значений тайм-аута, конкретные числовые значения удалены из содержания настоящего стандарта и заменены соответствующими переменными тайм-аута. Их отображения на числовые значения указаны в таблице 22.

Таблица 22 — Переменные тайм-аута

	Коммуникационная служба	Тайм-аут		Подраздел
		Переменная	Значение	
Процедура ассоциирования				
	Ассоциация	TO_{assoc}	10 с (и RC_{assoc} = 3)	8.7.5
	Конфигурация	TO_{config}	10 с	8.8.5
	Завершение ассоциации	$TO_{release}$	3 с	8.10.5
Процедура выполнения				
Объект MDS	Подтверждение действия	TO_{ca}	3 с	8.9.5.2
	Подтверждение отчета о событии	$TO_{cer-mds}$	MDS.Confirm-Timeout. Если атрибут отсутствует, агент и менеджер должны использовать значение 3 с	8.9.5.3
	Обработка вызова Get	TO_{get}	MDS.Transport-Timeout. Если атрибут отсутствует, агент и менеджер должны использовать значение 3 с	8.9.5.4
	Настройка подтверждения	TO_{cs}	3 с	8.9.5.5
	<тайм-аут между службами>	TO_{sp-mds}	3 с	8.9.5.6
Объект PM-store	Подтверждение действия	TO_{ca}	3 с	8.9.5.2
	Подтверждение отчета о событии	$TO_{cer-pms}$	Segm.Confirm-Timeout. Если атрибут отсутствует, агент и менеджер должны использовать значение 3 с	8.9.5.3
	Обработка вызова Get	TO_{get}	MDS.Transport-Timeout. Если атрибут отсутствует, агент и менеджер должны использовать значение 3 с	8.9.5.4
	Настройка подтверждения	TO_{cs}	3 с	8.9.5.5
	< тайм-аута конца сегмента>	TO_{sp-pms}	Segm.Transfer-Timeout	8.9.5.6
	Подтверждение действия — SegmClear	$TO_{clr-pms}$	PMS.Clear-Timeout	8.9.5.6
Объект Scanner	Настройка подтверждения	TO_{cs}	3 с	8.9.5.5
	Подтверждение отчета о событии	$TO_{cer-scan}$	Scan.Confirm-Timeout. Если атрибут отсутствует, агент и менеджер должны использовать значение 3 с	8.9.5.3

8.5 Процедура соединения

8.5.1 Общие сведения

В подразделах 8.5.2 — 8.5.5 описываются условия входа, нормальные процедуры, условия выхода и условия ошибок, которые могут возникать на диаграммах состояний для состояния «Соединен».

8.5.2 Условия входа

Агент и менеджер входят в состояние «Соединен», когда от транспортного уровня приходит индикация соединения между агентом и менеджером. Агент и менеджер получают индикацию соединения от собственных транспортных уровней (т.е. в это время обмен данными на прикладном уровне не про-

исходит). После начального входа в состояние «Соединен» и агент, и менеджер переходят в состояние «Не ассоциирован» (подсостояние состояния «Соединен»).

8.5.3 Нормальные процедуры

Так как состояние «Соединен» имеет несколько подсостояний, то фактические условия выполнения описываются как часть этих подсостояний.

8.5.4 Условия выхода

Агент и менеджер по возможности должны выйти из состояния ассоциации, перейдя в состояние «Завершает ассоциацию», отправив запрос на завершение ассоциации и ожидая ответ на этот запрос. Затем агент и менеджер должны закрыть активную ассоциацию и вернуться в состояние «Не ассоциирован». Это нормальное поведение до выхода агента или менеджера из состояния «Соединен». После этого закрытие соединения контролируется транспортным уровнем.

8.5.5 Условия ошибки

Передача данных может прерваться неожиданно (например, беспроводная передача данных выходит за пределы доступности или проводной интерфейс может быть преждевременно удален). В подобных случаях транспортный уровень должен оповестить прикладной уровень предупреждением об отсоединении. Агент и менеджер должны нести ответственность за возвращение в состояние «Отсоединен». Данное требование распространяется на состояние «Соединен» и все его подсостояния.

8.6 Процедура в состоянии «Не ассоциирован»

8.6.1 Общие положения

Подразделы 8.6.2 — 8.6.5 описывают условия входа, нормальные процедуры, условия выхода и условия ошибок, которые могут возникать для состояния «Не ассоциирован» на диаграммах состояний.

8.6.2 Условия входа

Состояние «Не ассоциирован» — состояние по умолчанию, в которое агент или менеджер первоначально переходят при первом уведомлении об установлении соединения. Агент или менеджер также возвращаются в это состояние после завершения или прерывания ассоциации с партнером.

8.6.3 Нормальные процедуры

Обычно агент ничего не делает в этом состоянии.

Менеджер в этом состоянии ожидает получения сообщения запроса ассоциации.

8.6.4 Условия выхода

Каждый раз, когда агент пытается создать ассоциацию с менеджером, он переходит в состояние «Ассоциирующий». Менеджер переходит в это состояние после получения сообщения с запросом ассоциации.

8.6.5 Условия ошибки

В состоянии «Не ассоциирован» может возникнуть ряд условий ошибок. Реакцией на возникновение таких условий может быть игнорирование их наличия или генерирование сообщения о прекращении ассоциации. Для получения дополнительной информации см. таблицу Е.1, состояние 2 (состояние «Не ассоциирован»).

8.7 Процедура ассоциирования

8.7.1 Общие положения

Процедура ассоциирования позволяет агенту и менеджеру согласовать общий протокол данных и общий набор параметров выполнения.

8.7.2 Условия входа

Агент и менеджер должны оставаться в состоянии «Не ассоциирован» до тех пор, пока агент не обнаружит потребность в установлении ассоциации. На этом этапе агент должен перейти в состояние «Ассоциирующий» и отправить запрос ассоциации. Менеджер должен перейти в состояние «Ассоциирующий» после получения от агента запроса ассоциации.

8.7.3 Нормальные процедуры

На рисунках 12 и 13 показаны схемы последовательности операций процедуры ассоциирования агента и менеджера. На рисунке 12 показана ситуация, когда менеджеру уже известна конфигурация агента в результате предварительного соединения с ним или вследствие наличия у агента стандартной конфигурации (т. е. предварительно заданной конфигурации, указанной в стандарте специализации). На рисунке 13 показана ситуация, когда менеджеру не известна конфигурация агента и он информирует агента о том, что запрос на ассоциацию принят, но конфигурация неизвестна; или ситуация, когда менеджеру известна конфигурация агента, но ему требуется перейти в состояние «Конфигурирующий», чтобы проверить атрибуты объекта MDS.

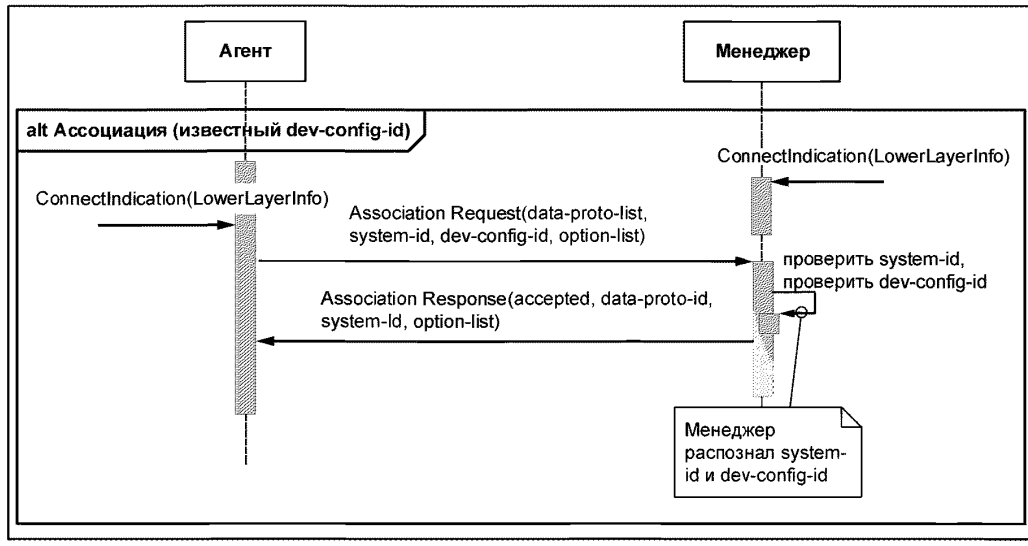


Рисунок 12 — Процедура ассоциации (известная конфигурация)

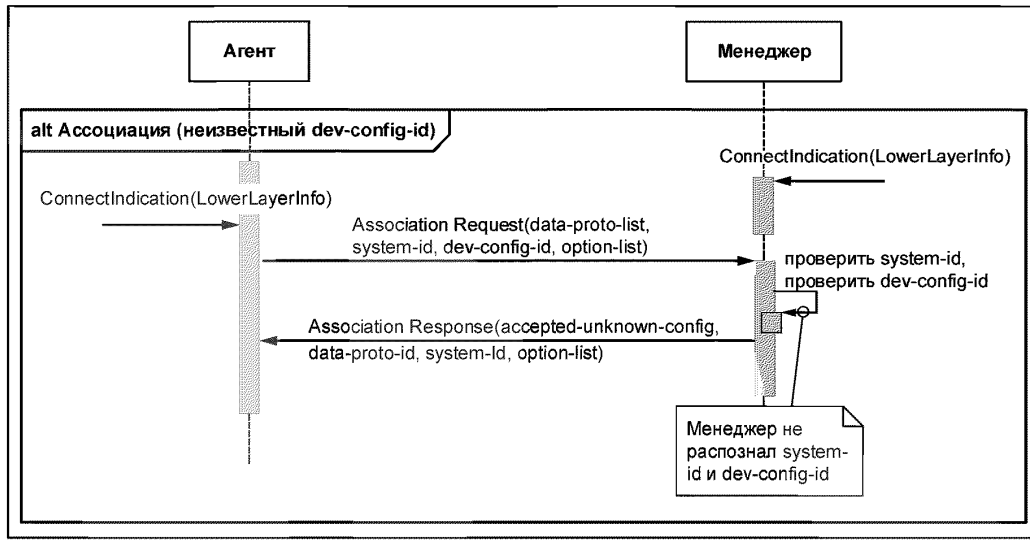


Рисунок 13 — Процедура ассоциации (неизвестная конфигурация)

Подразделы 8.7.3.1 — 8.7.3.2 описывают условия выполнения для двух разных ролей устройств: агента и менеджера.

8.7.3.1 Процедура агента

8.7.3.1.1 Общие положения

Агент, намеренный создать ассоциацию, должен начать с перехода в состояние «Ассоциирующий» и отправки менеджеру сообщения запроса ассоциации. Определение AarqApdu (см. А.8) описывает формат сообщения запроса ассоциации. Пример запроса ассоциации приведен в Н.2.1.1.

Сообщение запроса ассоциации содержит следующие элементы.

- Версия используемого протокола ассоциации (assoc-version). Данное поле позволяет агенту и менеджеру убедиться в том, что они используют одинаковую версию протокола обмена.

- Список протоколов передачи данных, которые поддерживает агент (data-proto-list). Агенту разрешено поддерживать один или несколько протоколов передачи данных для обмена информацией. Агент должен запросить список протоколов передачи данных начиная с наиболее предпочтительного протокола и заканчивая наименее предпочтительным протоколом.

Менеджер выбирает необходимый протокол и сообщает об этом агенту.

Чтобы разрешить выбор протокола передачи данных в ходе ассоциации, список data-proto-list содержит идентификатор, который обозначает, что протокол данных определяется одним из стандартов серии ISO/IEEE 11073 или определяется изготовителем. Данные параметры описаны в следующих двух подразделах. Доступны дополнительные коды, которые, однако, зарезервированы для последующих расширений. Агент должен поместить в data-proto-list не более одного элемента data-proto, содержащего поле data-proto-id, заданное равным data-proto-id-20601.

8.7.3.1.2 Протокол обмена данными, определенный настоящим стандартом

Если агент задает поле data-proto-id (см. А.8) равным data-proto-id-20601, он должен придерживаться определений абстрактного синтаксиса, содержащихся в настоящем стандарте, для типов данных и обмена сообщениями. Кроме того, поле data-proto-info должно заполняться с помощью структуры PhdAssociationInformation, которая определяет следующую информацию.

- Поле protocol-version содержит сведения о версии протокола обмена данными.

- Поле encoding-rules содержит конкретные правила кодирования DataApdu, поддерживаемые агентом. Агент должен задать один или несколько битов encoding-rules.

- Агент должен всегда поддерживать MDER (агентом должен задаваться бит mder поля encoding-rules).

- Помимо MDER, агент может предложить менеджеру другие правила кодирования, задав другие биты в поле encoding-rules.

- Поле nomenclature-version содержит сведения о версии использованной номенклатуры.

- Поле functional-units указывает все функциональные блоки и необязательные функции, поддерживаемые агентом.

- Поле system-type указывает тип системы (в данном случае агента).

- Поле system-id содержит уникальное значение атрибута System-Id (см. таблицу 3) агента. Для идентификации агента используется формат EUI-64. Менеджер может использовать это поле с целью определения идентификации агента, с которым он осуществляет обмен данными, и, возможно, для реализации простой политики ограничения доступа.

- Поле dev-config-id указывает конфигурацию, предлагаемую для начального рассмотрения во время этой ассоциации (см. 7.4.3). Для стандартных конфигураций значение поля dev-config-id должно находиться в диапазоне от standard-config-start до standard-config-end включительно. Для расширенных конфигураций значение поля dev-config-id должно находиться в диапазоне от extended-config-start до extended-config-end включительно.

- Поле data-req-mode-capab определяет режимы запроса данных, поддерживаемые агентом (см. 8.9.3.3.3).

- Поле option-list содержит список дополнительных атрибутов агента, которые он намерен передавать.

8.7.3.1.3 Протокол обмена данными, определенный изготовителем

Прочие спецификации могут использовать начальный запрос ассоциации с целью согласования использования протоколов, определенных изготовителем. В этом случае агент присваивает полю data-proto-id (см. А.8) значение data-proto-id-external. Агент использует структуру ManufSpecAssociationInformation, чтобы предоставить идентификатор UUID, обозначающий конкретный протокол, благодаря чему можно различать многочисленные возможные протоколы, определенные изготовителем. Фактическое поведение протокола за пределами начальной ассоциации находится вне области применения серии стандартов ИСО/ИИЭР 11073. Идентификатор UUID должен генерироваться согласно требованиям ITU-T Rec. X.667 (сентябрь 2004 года).

8.7.3.2 Ответ на запрос ассоциации

Агент, отправивший сообщение запроса ассоциации, должен ожидать получения сообщения ответа на запрос ассоциации от менеджера или тайм-аута (условия тайм-аута см. в 8.7.5).

Определение AareApdu (см. А.8) описывает формат сообщения ответа на запрос ассоциации. Пример ответа на запрос ассоциации приведен в Н.2.1.2. Сообщение ответа на запрос ассоциации содержит следующее.

- Поле result представляет результат выполнения процедуры ассоциации.

- Поле protocol-version указывает версию общего протокола данных, выбранного менеджером, если поле result содержит значение accepted или accepted-unknown-config.

- Стандарт ИИЭР 11073-20601-2014 не полностью совместим с ИИЭР 11073-20601-2008 и ИИЭР 11073-20601a™-2010. Любой менеджер, которому необходимо обменяться данными с агентом, используя protocol-version1 или protocol-version2, должен задать соответствующий бит в ответе на запрос ассоциации.

- Если поле result содержит значение accepted или accepted-unknown-config, то поле encoding-rules указывает одно и только одно правило кодирования DataApdu, выбранное менеджером,.

- Менеджер должен всегда поддерживать MDER для обеспечения интероперабельности.

- Кроме правил MDER, менеджер может выбрать одно из других правил кодирования, предложенных агентом.

Примечание — MDER всегда поддерживается агентом и менеджером. Но если агент предлагает менеджеру дополнительные правила кодирования, можно сделать вывод о том, что агент имеет на это важную причину (т.е. реализация поддержки дополнительных правил кодирования не выполняется без веских причин, связанных с продукцией). Следовательно, если агент предлагает дополнительные правила кодирования помимо MDER, предполагается, что менеджер выполнит одно из дополнительных предложенных правил кодирования при наличии возможности. Например, если агент предлагает MDER и правила уплотненного кодирования (PER), предполагается, что менеджер выберет кодирование PER при наличии возможности. Если агент предлагает MDER и правила кодирования XML (XER), предполагается, что менеджер выполнит правила кодирования XER при наличии возможности. Для агентов, предлагающих MDER, PER и XER, настоящий стандарт не дает рекомендаций в отношении выбора предпочтительного правила кодирования.

- Если поле result содержит значение accepted или accepted-unknown-config, то поле nomenclature-version указывает версию номенклатуры, выбранную менеджером.

- Если значение поля результата равно accepted или accepted-unknown-config, то поле functional-units указывает общие функциональные единицы и дополнительные функции, которые выбираются менеджером.

- Поле system-type указывает тип системы (в данном случае менеджера, поскольку сообщение посылается менеджером).

- Поле system-id содержит уникальный идентификатор системы менеджера. Для уникальной идентификации менеджера используется EUI-64. Агент может использовать это поле для определения, установлена ли связь с требуемым менеджером.

- Поле dev-config-id в ответе должно иметь значение manager-config-response.

- Поле data-req-mode-capab в ответе должно быть нулем.

Поле result в сообщении ответа на запрос ассоциации указывает результат запроса. Возможны (см. описание AssociateResult в А.8) следующие результаты:

- «accepted» означает, что ассоциация принимается и конфигурация известна. Агент должен перейти в состояние «Ожидает GetMDS» (более подробные сведения о рабочих процедурах доступны в 8.9);

- «accepted-unknown-config» означает, что ассоциация принимается, но агенту необходимо направить менеджеру свою конфигурацию. Агент, получивший ответ о том, что конфигурация не известна, должен перейти в состояние «Конфигурирующий» и следовать процедурам из подраздела 8.7.6 для передачи конфигурации;

- «rejected-unsupported-assoc-version» означает, что агент и менеджер не используют общую версию ассоциации.

- «rejected-no-common-protocol» означает, что менеджер отклоняет запрос на ассоциацию, поскольку в списке DataProtoList не найден общий протокол данных, совместно используемый менеджером и агентом;

- «rejected-no-common-parameter» означает, что менеджер отклоняет запрос ассоциации, поскольку менеджер и агент не имеют общего набора рабочих параметров в информации об ассоциации, специфичной для протокола (PhdAssociationInformation);

- «rejected-unauthorized» используется, когда менеджер определяет, что агент не авторизован для подключения. Способ принятия решения устанавливается изготовителем;

- «rejected-transient» используется, когда менеджер не может принять ассоциацию из-за временных ситуаций, например ограниченность ресурсов;

- «rejected-permanent» означает, что менеджер не может установить ассоциацию с агентом, но никакая дальнейшая информация о причине не доступна;

- «rejected-unknown» необходимо использовать с осторожностью и только тогда, когда вышеуказанные коды возврата не применяются.

При всех условиях rejected-* агент должен перейти в состояние «Не ассоциирован».

8.7.3.3 Процедура менеджера

Менеджер, получивший запрос на ассоциацию, должен сравнить параметры протокола и параметры выполнения с собственными параметрами и определить, совместимы ли агент и менеджер между собой. Если соединение является двунаправленным, менеджер должен сообщить о результатах этой оценки в поле result ответа на запрос ассоциации.

Менеджер может отклонить ассоциацию по любой из возможных причин отклонения, перечисленных в 8.7.3.2. Менеджер, отклонивший ассоциацию, должен перейти в состояние «Не ассоциирован».

Если запрос не отклоняется менеджером, то поле result в сообщении ответа на запрос ассоциации, исходящем от менеджера, указывает, понимает ли менеджер конфигурацию. Если менеджер распознает значение поля dev-config-id как представляющее известную стандартную специализацию приборов или конфигурацию из предыдущей ассоциации, то он должен отправить ответное сообщение на запрос ассоциации с полем результата accepted и перейти в состояние «Посылает GetMDS» или может отправить ответное сообщение на запрос ассоциации с полем результата accepted-unknown-config, чтобы вынудить агента перейти в состояние «Конфигурирующий» с целью проверки атрибутов объекта MDS перед окончательным принятием ассоциации.

Если менеджер не распознает значение поля dev-config-id, то он должен отправить сообщение ответа на ассоциацию с указанием accepted-unknown-config в поле result и перейти в состояние «Конфигурирующий».

Менеджер, принявший общий протокол, должен вернуть в ответе на запрос ассоциации предпочтительный общий протокол данных и общий набор рабочих параметров, выбранных из списка, переданного в запросе ассоциации.

8.7.4 Условия выхода

Менеджер выходит из состояния «Ассоциирующий» после отправки ответа на запрос ассоциации. Агент выходит из состояния «Ассоциирующий» после получения ответа на запрос ассоциации.

8.7.5 Условия ошибки

Агент должен ждать ответное сообщение на запрос ассоциации в течение периода TOassoc (тайм-аут: процедура ассоциации). После истечения периода TOassoc агент должен повторить передачу запроса ассоциации с новым периодом TOassoc. Данная процедура должна повторяться до момента получения ответа на запрос ассоциации или превышения числа попыток RCassoc (счетчик повторений процедуры ассоциации), предпринятых после первого тайм-аута (в зависимости от того, что произойдет раньше). В результате возможно не более RCassoc + 1 запросов на ассоциацию. Агент, который после данной последовательности повторных попыток не может успешно получить какие-либо сообщения ответа на ассоциацию, должен отправить сообщение о прекращении ассоциации с менеджером и вернуться в состояние «Не ассоциирован».

Агент или менеджер, получивший сообщение о прекращении ассоциации, находясь в состоянии «Ассоциирующий», должен перейти в состояние «Не ассоциирован».

8.7.6 Тестовая ассоциация

Тестовая ассоциация создается агентом и менеджером с целью обмена данными, предназначенными для тестирования. Настоящий стандарт определяет только процесс входа устройств в процесс тестовой ассоциации и выхода из нее, но не формат и семантику таких обменов. Отдельные специализации приборов могут определить стандартизированные тестовые ресурсы, идентификаторы конфигураций и процессы, которые можно использовать во время тестовой ассоциации. Тестовая ассоциация может использоваться в целях тестирования, специфичного для изготовителя.

Так как настоящий стандарт не определяет семантику тестовой ассоциации, то в нем не определены и специфичные механизмы надлежащего управления тестовыми данными. Однако очень важно, чтобы приборы обеспечивали защиту, снижающую риск обработки тестовых данных в качестве фактических результатов измерений. В общем случае результаты измерений, созданные с помощью тестовой ассоциации, должны быть доступны только тем элементам, которые воспринимают понятие тестовой ассоциации. Разработчикам следует предпринять следующие действия:

- Установить бит test-data или demo-data атрибута MeasurementStatus при генерировании имитационных результатов измерений. Если атрибут MeasurementStatus не поддерживается, необходимо использовать альтернативные средства пометки таких данных.

- Попытаться убедиться, что местные устройства изображения и хранилища результатов измерений игнорируют тестовые и демонстрационные данные, если они не могут должным образом пометить такие данные пользователю и обнаружить вход и выход из тестовой ассоциации. Местный компонент агента, не участвующий в протоколе ИИЭР 11073-20601, не является хорошим кандидатом на получение тестовых результатов измерений.

- Попытаться убедиться, что результаты измерений, помещенные в PM-store или другую постоянную структуру хранения, никогда не видны за пределами тестовой ассоциации. Для этой цели может использоваться пометка и/или очистка постоянных хранилищ данных.

- Попытаться убедиться, что устройства, которые отображают или хранят тестовые или демонстрационные данные, должным образом обновляются, когда такие события, как отсоединение, вызывают необходимость завершения тестовой ассоциации.

Для формирования тестовой ассоциации менеджер и агент должны поддерживать тестовые ассоциации и оказаться готовыми создать тестовую ассоциацию в определенный момент времени. Для однозначного создания тестовой ассоциации используется трехэтапный протокол.

На первом этапе агент передает менеджеру два бита информации в поле fun-units структуры PhdAssociationInformation. Бит fun-unit-havetestcap указывает, что агент имеет возможности тестирования, которые можно использовать при тестовой ассоциации. Бит fun-unit-createtestassociation используется агентом для запроса менеджеру о создании тестовой ассоциации. Агент не должен задавать бит fun-unit-createtestassociation, если не задан бит fun-unit-havetestcap. Агент, указавший бит fun-unit-havetestcap в структуре PhdAssociationInformation, не должен завершать ассоциацию вследствие получения ответа с битом fun-unit-createtestassociation. То есть если агент задает бит fun-unit-havetestcap и предлагает несколько конфигураций со стандартизированными возможностями выполнения тестирования, агент должен оказаться готовым создать тестовую ассоциацию, используя любую из этих конфигураций.

На втором этапе протокола менеджер сообщает агенту о своем намерении создать тестовую ассоциацию. Менеджер передает эту информацию агенту с помощью бита fun-unit-createtestassociation. Бит задается менеджером, чтобы обозначить свой вход в тестовую ассоциацию. Менеджер должен задать этот бит исключительно в том случае, если запрос агента содержит бит fun-unit-havetestcap. Согласно настоящему стандарту менеджер не обязан входить в тестовую ассоциацию даже по запросу агента. Агент должен игнорировать бит fun-unit-havetestcap в ответе на запрос ассоциации.

Заключительный этап протокола тестовой ассоциации подразумевает, что агент принимает решение о продолжении или завершении тестовой ассоциации. Агент не должен входить в состояние тестовой ассоциации, если менеджер не установил бит fun-unit-createtestassociation. Тестовая ассоциация завершается, когда конечный автомат ассоциации переходит в состояние «Не ассоциирован».

8.8 Процедура конфигурации

8.8.1 Общие положения

Состояние «Конфигурирующий» активно, когда агенту необходимо передать менеджеру информацию о конфигурации.

8.8.2 Условия входа

Сообщение ответа на ассоциацию с полем result = accepted-unknown-config должно инициировать переход агента в подсостояние «Конфигурирующий»/«Отправляет конфигурацию» и отправку конфигурации менеджеру. Менеджер переходит в подсостояние «Конфигурирующий»/«Ожидает конфигурацию» сразу после отправки ответа на запрос ассоциации с результатом accepted-unknown-config.

Сообщение ответа на запрос ассоциации с результатом accepted должно инициировать переход агента в подсостояние «Конфигурирующий»/«Ожидает GetMDS». Менеджер переходит в подсостояние «Конфигурирующий»/«Посылает GetMDS» сразу после отправки ответа на запрос ассоциации с результатом accepted.

Обратите внимание, что частью процесса конфигурации также является присваивание значения атрибута Handle экземплярам объектов. Менеджер, знающий конфигурацию агента, также знает значения, присвоенные атрибуту Handle. Это подразумевает, что стандартная конфигурация (например, конфигурация, определенная в специализации ИСО/ИИЭР 11073-104zz) определяет фиксированные значения атрибутов Handle.

8.8.3 Нормальные процедуры

На рисунке 14 показана диаграмма последовательности для процедуры конфигурации. Во время процедуры конфигурации агент должен передавать конфигурационную информацию обо всех объектах, которые он поддерживает (кроме объекта MDS), а также все статические атрибуты объектов. Агенты, как правило, имеют очень статичную конфигурацию, поэтому передача всех статических частей во время этапа однократного конфигурирования уменьшает общий объем передаваемых данных. Новые типы измерений не добавляются динамически, многие атрибуты не изменяются, а наборы переданных атрибутов объекта часто остаются прежними. Повторное конфигурирование требуется только при изменении агента (например, как часть процедуры начальной установки, когда возможна настройка определенных возможностей измерений).

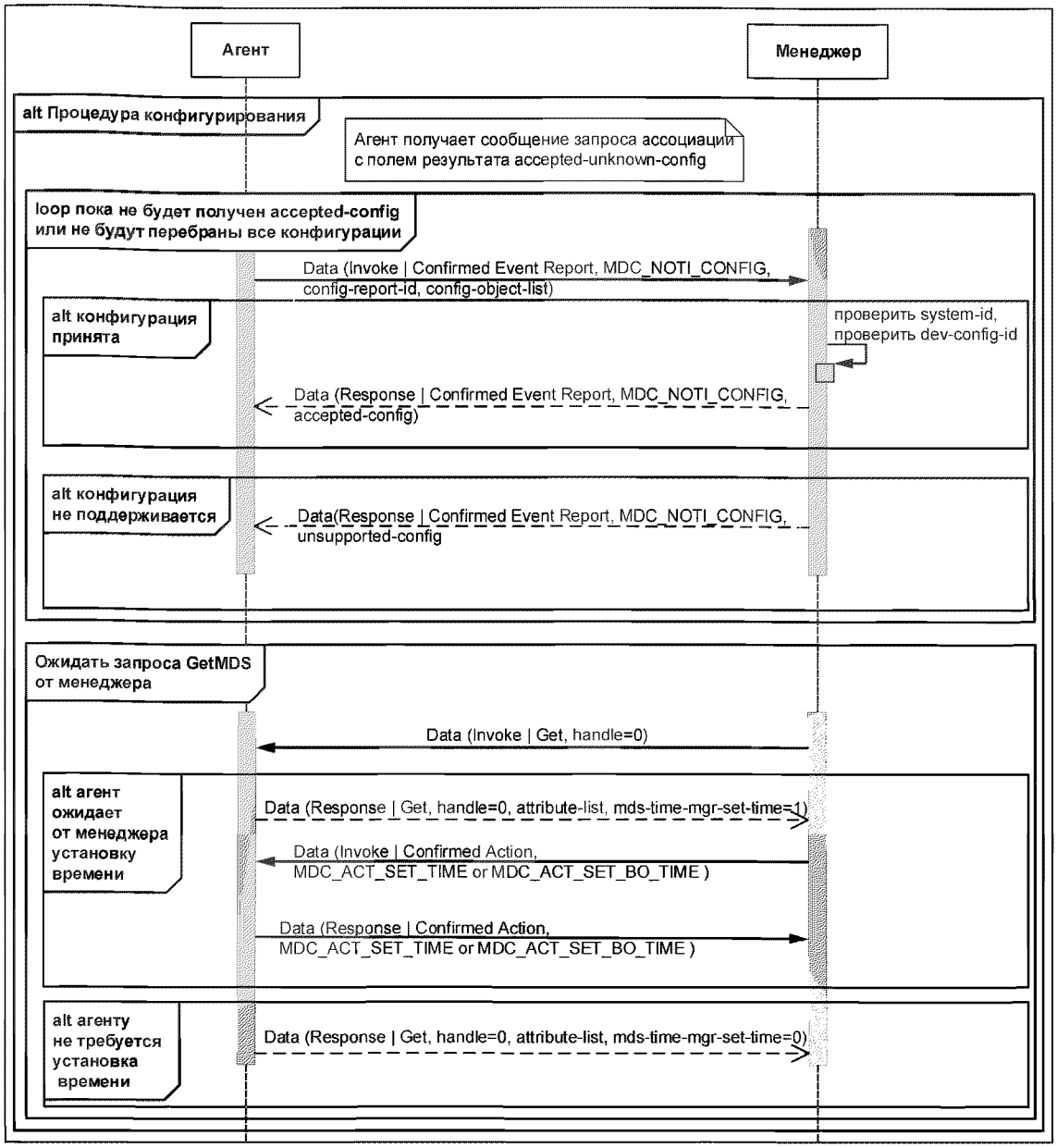


Рисунок 14 — Процедура конфигурации

Агент выполняет процедуру конфигурирования, посылая менеджеру сообщение подтверждаемого запроса события MDC_NOTI_CONFIG. Сообщение с уведомлением о конфигурации идентифицирует:

- атрибут Dev-Configuration-Id объекта MDS, соответствующий описываемой конфигурации;
- все объекты, поддерживаемые агентом, кроме объекта MDS; и
- набор статических атрибутов каждого объекта.

Атрибуты содержат идентификацию номенклатуры классов объекта (см. 6.3.4.2, 6.3.5.2 и 6.3.6.2), физиологический идентификатор (номенклатурный код), идентификатор единицы/размерности (номенклатурный код), а также дополнительно строки для маркировки и любые другие статические атрибуты, которые могут оказаться полезными. Такая информация рассматривается как плоское (не иерархическое) и статическое дерево состава агента. Объект MDS исключается из конфигурации, так как большая часть информации является динамической или специфичной для изготовителя. Отдельная команда Get MDS Object предоставляет механизм извлечения этой информации (см. 6.3.2.6.1).

Для объектов, сообщающих каждый раз те же самые атрибуты, рекомендуется использовать отчет о событии в фиксированном формате (см. 7.4.5), при этом агент должен отправить атрибут Attribute-

Value-Map, описывающий структуру сообщения, до отправки отчета о событии для такого объекта в фиксированном формате. Для объектов Scanner, использующих отчеты о событиях в групповом формате, агент должен передать атрибут Handle-Attr-Val-Map, описывающий структуру, до отправки отчетов о событиях сканирования в групповом формате. Обычно такие схемы структур передаются в отчете о событии конфигурации.

Если набор передаваемых атрибутов объекта не постоянен, рекомендуется использовать отчет о событии в переменном формате. Такой формат позволяет сообщать атрибуты конфигурации как часть обновлений значений. В этом случае атрибут Attribute-Value-Map не предоставляется в отчете о событии конфигурации или является пустым списком.

Конфигурация агента идентифицируется атрибутом Dev-Configuration-Id объекта MDS, и агент передает это значение менеджеру в поле dev-config-id сообщения запроса ассоциации или в поле config-report-id сообщения отчета о событии конфигурации.

Для передачи своей конфигурации агент должен использовать сообщение данных «Remote Operation Invoke | Confirmed Event Report» (см. исходное определение EventReportArgumentSimple в А.10.3) с типом события MDC_NOTI_CONFIG (остальную часть структуры см. в описании ConfigReport, приведенном в А.11.5). Менеджер должен ответить сообщением «Remote Operation Response | Confirmed Event Report» (определение EventReportResultSimple см. в А.10.3) с указанием типа события MDC_NOTI_CONFIG, включенного в структуру ConfigReportRsp. Пример запроса события конфигурации, отправленного агентом, приведен в Н.2.2, за которым следует пример ответа от менеджера.

Агенты могут поддерживать несколько конфигураций. В этом случае агент должен послать каждую из доступных конфигураций, начиная с предпочтительной конфигурации. Если менеджер принял конфигурацию, то он отвечает сообщением о принятии конфигурации (accepted-config), после чего менеджер переходит в состояние «Посылает GetMDS», а агент — в состояние «Ожидает GetMDS». Если менеджер не принимает конфигурацию, он должен отправить ответ о неподдерживаемой конфигурации (unsupported-config). После получения ответа о неподдерживаемой конфигурации агент должен отправить следующую конфигурацию. Данный процесс повторяется до тех пор, пока агент не попытается отправить все конфигурации. После этого он должен отправить сообщение о завершении ассоциации, содержащее код причины no-more-configurations, чтобы указать на невозможность работы с менеджером.

Агент, соответствующий одной или нескольким специализациям и/или профилям приборов, которые определяют стандартные конфигурации (то есть специализации ИСО/ИИЭР 11073-104zz), должен поддерживать одну или несколько стандартных конфигураций и может поддерживать одну или несколько расширенных конфигураций. В целях интероперабельности такой агент должен отправлять поддерживаемые стандартные конфигурации как резервные, если расширенные конфигурации не поддерживаются. Однако если расширенная конфигурация специализации, реализованной агентом, позволяет использовать объекты PM-store и результаты измерений передаются только с помощью событий данных сегментов PM-Segment и при этом стандартная конфигурация не поддерживает объекты PM-store, то агент не обязан поддерживать соответствующую стандартную конфигурацию. Если агент реализует несколько специализаций ИСО/ИИЭР 11073-104zz, то его атрибут System-Type-Spec-List содержит список пар «тип — версия», каждая из которых ссылается на соответствующую специализацию приборов и версию специализации.

Если агент соответствует стандартной конфигурации, то он должен использовать идентификатор Dev-Configuration-Id согласно определению в конкретной специализации ИСО/ИИЭР 11073-104zz. Значения Dev-Configuration-Id стандартной конфигурации задаются в диапазоне от standard-config-start до standard-config-end включительно.

Когда агент отправляет отчет о событии конфигурации, соответствующей стандартной конфигурации, сообщение конфигурации не обязано содержать информацию о конфигурации и может отправить тип события MDC_NOTI_CONFIG с идентификатором стандартной конфигурации в поле config-report-id и пустым списком config-obj-list. Если менеджер не распознает стандартную конфигурацию (например, менеджер создан до реализации специализации прибора), он должен отравить ответ standard-config-unknown (стандартная конфигурация не известна). Агент может повторить передачу информации о стандартной конфигурации, однако в этом случае он должен отправить полную информацию о конфигурации, а не пустой список config-obj-list.

Примечание — Если менеджер способен взаимодействовать с помощью предоставленной стандартной конфигурации, то он может принять такую конфигурацию. Если менеджер хранит конфигурации, он может сохранить конфигурацию для последующего обращения, когда какой-либо агент использует идентификацию стандартной конфигурации, и впредь рассматривать такую конфигурацию как распознанную.

Агент, имеющий нестандартную конфигурацию, должен присвоить уникальный идентификатор своей конфигурации, генерируя значение для атрибута Dev-Configuration-Id в диапазоне от extended-config-start до extended-config-end включительно.

Агенту следует постоянно использовать одно и то же значение атрибута Dev-Configuration-Id в последующих запросах ассоциации. Это имеет два следствия. То же самое значение Dev-Configuration-Id не должно использоваться агентом при последующих ассоциациях для идентификации другой конфигурации прибора. Агенту следует использовать то же самое значение Dev-Configuration-Id в последующих запросах ассоциации с менеджером, чтобы обозначить ту же конфигурацию прибора. Выбранное значение dev-config-id должно сообщаться в атрибуте Dev-Configuration-Id объекта MDS.

Если агент изменяет свою конфигурацию таким образом, что больше не может поддерживать старую, или определяет, что новая конфигурация должна использоваться как предпочтительная, он должен завершить любую существующую ассоциацию, отправив сообщение о завершении ассоциации с указанием причины изменения конфигурации (configuration-changed). Если новая конфигурация представляет собой новую расширенную конфигурацию, то агент должен назначить этой конфигурации новый идентификатор. Когда в следующий раз агент установит ассоциацию, он выполнит процедуру согласования с менеджером, предъявляя каждую конфигурацию в порядке приоритетности, как описано выше.

Агент может отправить расширенную конфигурацию с пустым списком config-object-list. Такая ситуация может возникать, например, когда в процессе работы агент загружает подключаемые модули, которые в настоящее время отсутствуют. Ответ менеджера содержит accepted-config или unsupported-config.

На протяжении периода TOget после принятия менеджером конфигурации, отправленной агентом, менеджер должен запросить атрибуты объектов MDS агента, отправив сообщение данных с командой «Remote Operation Invoke | Get» и зарезервированным значением дескриптора 0 либо конкретным списком атрибутов, содержащим как минимум Mds-Time-Info. Дополнительную информацию см. в 8.9.3.2.

Если бит mds-time-mgr-set-time не установлен, агент сразу переходит в состояние «Выполнение». Если бит mds-time-mgr-set-time установлен, агент должен ожидать на протяжении периода TOca получения от менеджера команды действия Set-time. После получения действия Set-Time агент должен сбросить бит mds-time-mgr-set-time перед отправкой подтверждения действия Set-Time. После отправки подтверждения действия Set-Time агент переходит в состояние «Выполнение». Дополнительную информацию см. в 8.12.2.1.

Если для передачи отчета агент использует базовое время со смещением, в вышеуказанной процедуре вместо действия Set-Time должно использоваться действие Set-Base-Offset-Time.

8.8.4 Условия выхода

Если менеджер принял предпочтительную конфигурацию, то он должен отправить агенту ответ о принятии конфигурации и перейти в состояние «Посылает GetMDS». Если менеджер получает запрос на завершение ассоциации по причине отсутствия у агента дополнительных конфигураций (no-more-configurations), менеджер должен перейти в состояние «Не ассоциирован».

Агент, получивший от менеджера ответ о принятии конфигурации (accepted-config), должен перейти в состояние «Ожидает GetMDS». Если агент получает от менеджера ответ об отсутствии поддержки (unsupported-config), то он должен отправлять менеджеру следующую конфигурацию до тех пор, пока не останется конфигураций. После этого агент должен перейти в состояние «Завершает ассоциацию» и отправить сообщение с запросом на завершение ассоциации, указав причину no-more-configurations.

В случае тайм-аута агент или менеджер должен отправить запрос на прерывание ассоциации и перейти в состояние «Не ассоциирован».

8.8.5 Условия ошибки

Агент должен ожидать получения сообщения «Remote Operation Response | Confirmed Event Report | MDC_NOTI_CONFIG» в течение периода TOconfig (тайм-аут: процедура конфигурации). После истечения периода TOconfig агент должен отправить менеджеру сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

Менеджер, находясь в состоянии «Ожидает конфигурацию», должен ждать информацию о конфигурации минимум TOconfig секунд, прежде чем отправить сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

Если в какой-либо момент времени агент или менеджер получает или посылает сообщение о прекращении ассоциации, он должен перейти в состояние «Не ассоциирован».

8.9 Процедура выполнения

8.9.1 Общие положения

Передача медицинских данных и информации о статусе агента происходит в состоянии «Выполнение».

8.9.2 Условия входа

Агент и менеджер переходят в состояние «Выполнение», когда конфигурация агента уже известна менеджеру или после того, как агент передал менеджеру приемлемую конфигурацию.

8.9.3 Нормальные процедуры

8.9.3.1 Общие положения

Подразделы 8.9.3.2 — 8.9.3.4 содержат описание процедур, которые могут осуществляться в состоянии «Выполнение».

8.9.3.2 Атрибуты объектов MDS

В любое время при нахождении в состоянии «Выполнение» или «Посылает GetMDS» менеджер может запросить атрибуты объектов MDS агента, отправив сообщение данных с командой «Remote Operation Invoke | Get». Агент должен сообщить менеджеру реализованные атрибуты объектов MDS, используя ответное сообщение «Remote Operation Response | Get». Если значение дескриптора в команде «Remote Operation Invoke | Get» равно 0, должны сообщаться все реализованные атрибуты объектов MDS агента, иначе агент информирует только о запрошенных реализованных атрибутах. Примеры использования данного набора сообщений приведены в Н.2.3. Агенты должны поддерживать команду Get, запрашивающую все атрибуты (т. е. список attribute-id-list пуст), и должны поддерживать извлечение конкретного списка атрибутов. Дескриптор указывается в поле obj-handle (А.10.4) и отсутствует в списке идентификаторов атрибутов запроса или в списке атрибутов ответа. Если менеджер запрашивает определенные атрибуты объектов MDS, указанные с помощью элементов в списке attribute-id-list, то агент должен возвратить сообщение rors-cmip-get, в котором attribute-list содержит список тех запрошенных атрибутов объекта MDS, которые реализованы.

На рисунке 15 показана диаграмма последовательности операций менеджера, запрашивающего от агента атрибуты объектов MDS.

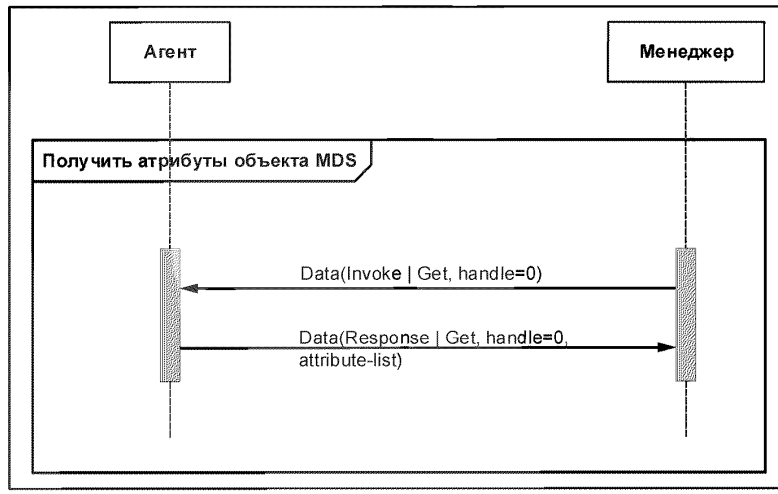


Рисунок 15 — Диаграмма последовательности получения атрибутов объектов MDS

8.9.3.3 Передача результатов измерений

8.9.3.3.1 Общие положения

Передача результатов измерений может инициироваться агентом или менеджером (см. 7.4.4). Передача, инициированная агентом, как правило, ожидается от агентов, которые передают небольшое количество редкой эпизодической информации или требуют минимальной пропускной способности. Агентам с большими объемами данных, частыми передачами данных или потоковыми данными необходимо использовать передачу, инициированную менеджером. Передача, инициированная менеджером, пред-

почтительна во всех случаях, поскольку такой подход предоставляет механизмы управления потоком данных. Обратите внимание, что получение запроса на передачу результатов измерений не обязывает агента выполнять измерения, а требует от него передать любые доступные результаты измерений.

В каждом случае, кроме режима одиночного ответа, передача результатов измерений осуществляется с использованием отчета о событиях, подтверждаемого или неподтверждаемого, по выбору агента.

Все варианты двух стилей подробно описаны в 8.9.3.3.2 — 8.9.3.3.8.

8.9.3.3.2 Инициируемая агентом передача результатов измерений

Агент, поддерживающий передачу, инициированную агентом, должен указать, что реализует поддержку через структуру DataReqModeCapab, или иметь в своей конфигурации один или несколько экземпляров объекта Scanner.

Для отправки менеджеру результата спонтанного измерения, который не был запрошен менеджером, агент должен использовать службу EVENT REPORT (см. 7.3). Для этой цели должны использоваться сообщение DataApdu в команде «Remote Operation Invoke | Event Report» и один из типов событий MDC_NOTI_SCAN_REPORT_*. При использовании подтверждаемого отчета о событии менеджер должен передать сообщение DataApdu с ответом «Remote Operation Response | Confirmed Event Report» (см. рисунок 16). Если используется неподтверждаемый отчет о событиях, менеджер не должен отвечать.

У агентов, использующих двустороннюю коммуникацию, в начале работы атрибут Operational-State объектов Scanner должен быть неактивным, пока менеджер не активирует его. Менеджер должен сделать состояние объектов Scanner активным, когда ему необходимо получить данные.

Для инициированной агентом передачи результатов измерений через объект MDS полю data-req-id в отчете о сканировании (MDC_NOTI_SCAN_REPORT_ *) должно быть присвоено значение data-req-id-agent-initiated-confirmed или data-req-id-agent-initiated-unconfirmed (в зависимости от того, является ли отчет о событии подтверждаемым или неподтверждаемым).

Менеджер может остановить инициированную агентом передачу результатов измерений, отправив агенту сообщение о завершении или прекращении ассоциации, чтобы закончить ассоциацию. Если агент использует объект Scanner, менеджер может деактивировать объект Scanner, присвоив соответствующее значение его атрибуту Operational-State с помощью службы SET.

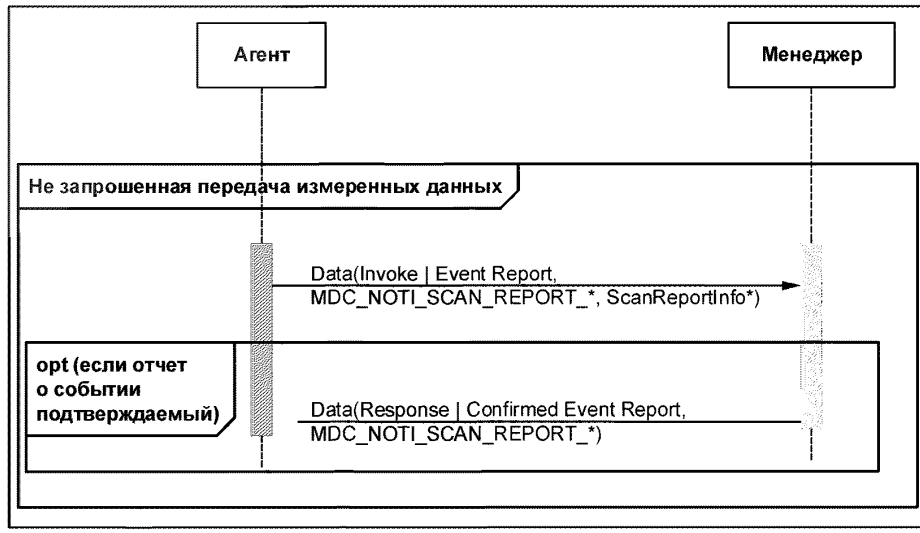


Рисунок 16 — Инициированная агентом передача результатов измерений

8.9.3.3.3 Обзор инициируемой менеджером передачи результатов измерений

Если агент поддерживает передачу, инициированную менеджером, то он должен указать, какие функции поддерживает, используя структуру DataReqModeCapab. Если агент не установил бит в поле DataReqModeFlags структуры DataReqModeCapab, то менеджер должен предполагать, что агент не поддерживает соответствующую функцию.

При инициированной менеджером передаче результатов измерений менеджер использует службу ACTION (см. 7.3), предоставленную агентом, чтобы запросить передачу результатов измерений со стороны агента. Менеджер, которому необходимо сделать это, должен послать подтверждаемый запрос DataApdu ActionArgumentSimple с типом действия MDC_ACT_DATA_REQUEST, за которым следует информация DataRequest. Такой запрос данных может оказаться запросом начала или остановки в зависимости от значения бита data-req-start-stop режима data-req-mode (см. A.11.5) либо запросом продолжения, указанным с помощью бита data-req-continuation.

Для запроса начала можно использовать три режима: одиночный ответ (data-req-mode-single-rsp), период времени (data-req-mode-time-period) и без ограничения времени (data-req-mode-time-no-limit). В зависимости от режима запроса начала агент может отправить менеджеру один или несколько отчетов о событиях. При инициации начала режима передачи данных менеджер предоставляет идентификатор data-req-id, который должен использоваться агентом во всех отчетах о событиях, связанных с таким запросом начала. Когда активен режим передачи данных, то в случае получения нового запроса начала с тем же самым значением data-req-id этот запрос должен считаться приоритетным и инициировать новый режим. Агент обрабатывает новый запрос начала, как если бы это был запрос остановки, за которым следует запрос начала передачи данных. Режимы одиночного ответа и периода времени имеют четко определенные конечные моменты времени, после которых ресурсы, поддерживающие запрос, могут быть освобождены. Запрос без ограничения времени не имеет четко определенного конечного момента времени. Менеджер должен сгенерировать запрос остановки, когда больше не заинтересован в потоке измерений (особенно при запросах без ограничения времени), чтобы освободить ресурсы агента.

Для каждого такого режима может выбираться один из трех различных вариантов, соответствующих области объектов, к которым относится запрос передачи данных: все доступные данные агента (data-req-scope-all), доступные данные агента с учетом конкретного класса объектов (data-req-scope-class) и доступные данные агента с учетом конкретных объектов, идентифицированных их дескрипторами (data-req-scope-handle).

При использовании data-req-scope-all агент должен учесть все объекты, кроме объекта MDS, при определении содержания каждого отчета о событии.

При использовании data-req-scope-class менеджер должен указать класс объектов, включаемых в отчет, в поле data-req-class. В этом случае агент должен включать в отчеты о событиях только объекты, относящиеся к указанному классу. Разрешены следующие идентификаторы классов: MDC_MOC_ VMO_METRIC_NU, MDC_MOC_ VMO_METRIC_SA_RT и MDC_MOC_VMO_METRIC_ENUM.

При отправке data-req-scope-handle менеджер должен предоставить список дескрипторов в поле data-req-obj-handle-list. В этом случае агент должен включать в отчеты о событиях только объекты, описанные валидными дескрипторами, принадлежащие этому списку. В данном контексте термин «валидный» означает все дескрипторы, ассоциированные с объектами, произведенными от класса Metric (например, Numeric, RT-SA или Enumeration) и поддерживаемыми агентом.

Запрос остановки может использоваться менеджером с целью завершения начатой ранее передачи результатов измерений в режиме периода времени или без ограничения по времени.

При использовании режима ограниченного времени, если менеджеру необходимо продлить время, которое предоставлено агенту для передачи данных, менеджер должен задать бит data-req-continuation в этом режиме и присвоить data-req-time значение времени, отведенного агенту на продолжение передачи.

Поле data-req-id в запросе данных используется для дифференциации ответов на несколько запросов данных у одного и того же агента (если агент позволяет несколько одновременных запросов данных). Менеджер должен присвоить полю data-req-id значение из диапазона от data-req-id-manager-initiated-min до data-req-id-manager-initiated-max включительно. Агент должен использовать одно и то же значение data-req-id во всех ассоциированных отчетах о событиях.

Необходимо отметить, что менеджер может присвоить полю data-req-id любое значение из допустимого диапазона. Поэтому агент не должен рассчитывать, что поле data-req-id можно использовать для каких-либо выводов, например о порядке, в котором разные запросы данных были созданы менеджером.

Потоковым агентам необходимо использовать инициированную менеджером передачу результатов измерений (или объекты Scanner), чтобы позволить менеджеру управлять процессом получения данных. Менеджерам следует активировать потоковых агентов как можно раньше, чтобы информация агента стала легко доступной.

В подразделах 8.9.3.3.4 — 8.9.3.3.6 описаны три режима передачи результатов измерений, инициируемой менеджером.

8.9.3.3.4 Режим одиночного ответа, инициированный менеджером

Режим одиночного ответа позволяет менеджеру запрашивать данные у агента и получать их в ответном сообщении (см. рисунок 17). Не существует каких-либо требований, чтобы для составления ответа агент собирал какие- либо данные (например, надул манжету тонометра). Если на момент запроса агент не располагает данными, он должен возвратить пустой список данных. Если агент располагает данными и статус результата — data-req-result-no-error, он должен послать сообщение DataResponse, содержащее статус результата запроса (DataReqResult), а также данные измерений (ScanReportInfo*). Это ответное сообщение должно завершить доступ к результатам измерений.

Режим одиночного ответа не позволяет агенту удостовериться, что менеджер получил данные измерений. Если такое подтверждение имеет большую важность, используется команда запроса данных в режиме периода времени с нулевым значением тайм-аута (см. 8.9.3.3.5).

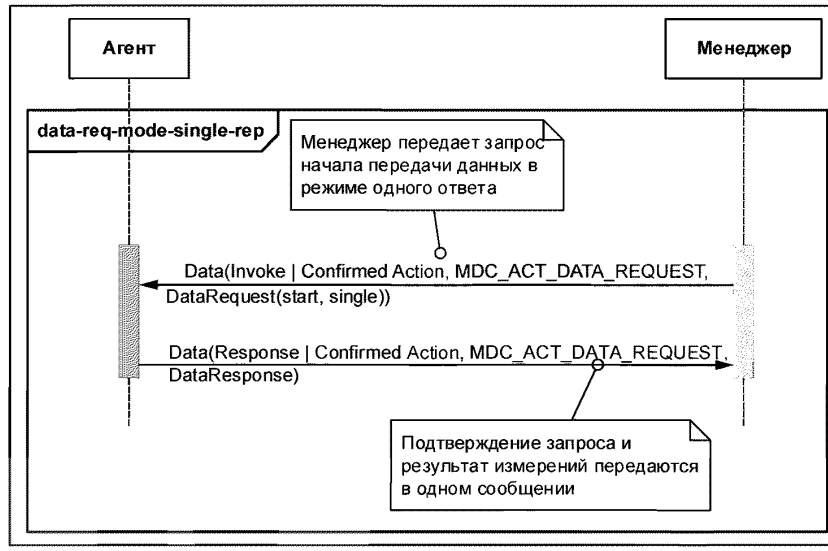


Рисунок 17 — Передача результатов измерений, инициированная менеджером (data-req-mode-single-rep)

8.9.3.3.5 Режим периода времени, инициированный менеджером

Режим периода времени используется менеджером с целью активации агента для передачи данных, которые он собирает на протяжении запрошенного периода времени (см. рисунок 18). Когда агент получает от менеджера сообщение DataRequest о начале работы, он должен отправить ответное сообщение DataResponse, подтверждающее статус результата запроса (DataReqResult), без передачи каких-либо результатов измерений в этом сообщении. Если в сообщении DataReqResult передан код результата data-req-result-no-error, то каждый раз, когда данные становятся доступными, агент должен использовать службу EVENT REPORT для передачи менеджеру отчетов о событиях, содержащих результаты измерений, пока не истечет период времени, указанный в запросе данных, или не будет получен от менеджера запрос остановки, или не будет закончена ассоциация агента и менеджера. Агент определяет, необходимо ли использовать для передачи данных сообщение подтверждаемого или неподтверждаемого отчета о событии.

Если менеджеру требуется продлить период времени, он должен послать data-req-id, задать режим продолжения, установив бит data-req-continuation, и присвоить data-req-time значение времени, до истечения которого агент может продолжать передавать данные. Все остальные параметры в сообщении DataRequest должны игнорироваться, при этом должны использоваться параметры исходной команды начала измерений. С момента получения команды агент должен применить новый период времени измерений. Если команда продолжения получена для значения data-req-id, которое не соответствует режиму ограниченного времени, то агент должен возвратить результат data-req-result-continuation-not-supported. Если команда продолжения получена для несуществующего значения data-req-id, агент должен возвратить data-req-result-invalid-req-id. Например, если время измерений истекает до получения команды на продолжение, то измерения останавливаются и data-req-id удаляется.

В режиме ограниченного времени, если значение data-req-time равно 0, агент должен подтвердить запрос и, если менеджер подтвердил его получение, немедленно начать передачу всех данных, доступных на текущий момент времени, в отчетах о событиях, а затем остановиться. В отличие от режима одиночного ответа (см. 8.9.3.3.4) режим ограниченного времени позволяет агенту использовать сообщения подтверждаемого или не подтверждаемого отчета о событиях. Например, агент может использовать подтверждаемый отчет о событиях, чтобы убедиться, что данные получены менеджером, прежде чем удалить их из локального кэша.

При получении запроса остановки данных для активного data-req-id агент должен немедленно прекратить отправку отчетов о событиях для этого data-req-id.

Поле data-req-id в этих отчетах о событиях используется менеджером для привязки результатов измерений к соответствующему запросу данных.

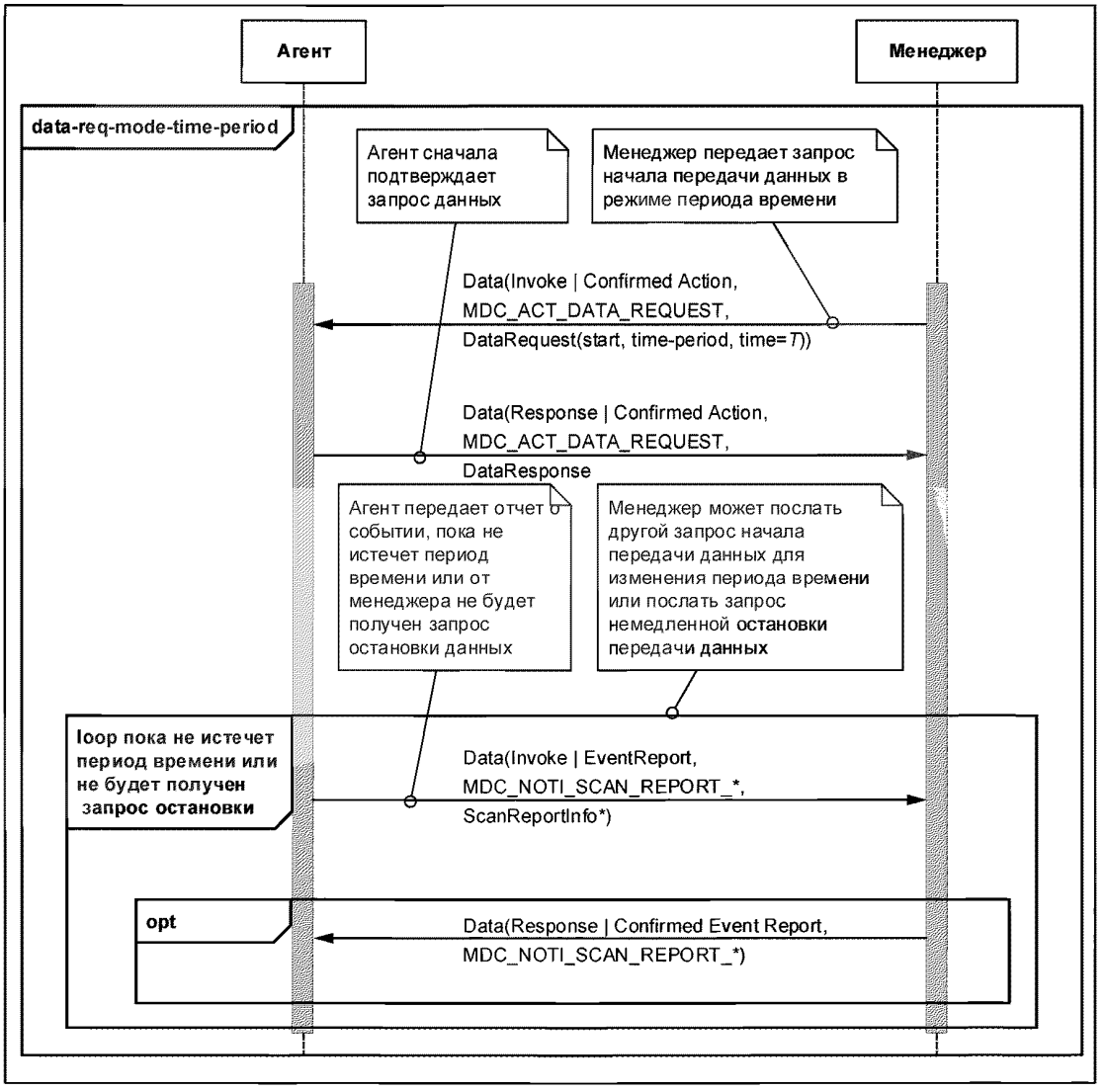


Рисунок 18 — Передача результатов измерений, инициированная менеджером (data-req-mode-time-period)

8.9.3.3.6 Инициируемый менеджером режим без ограничения времени

Режим без ограничения времени должен использоваться для команды агенту на непрерывную отправку отчетов о событиях до тех пор, пока не будет получена команда остановки или ассоциация между агентом и менеджером не будет закончена (см. рисунок 19).

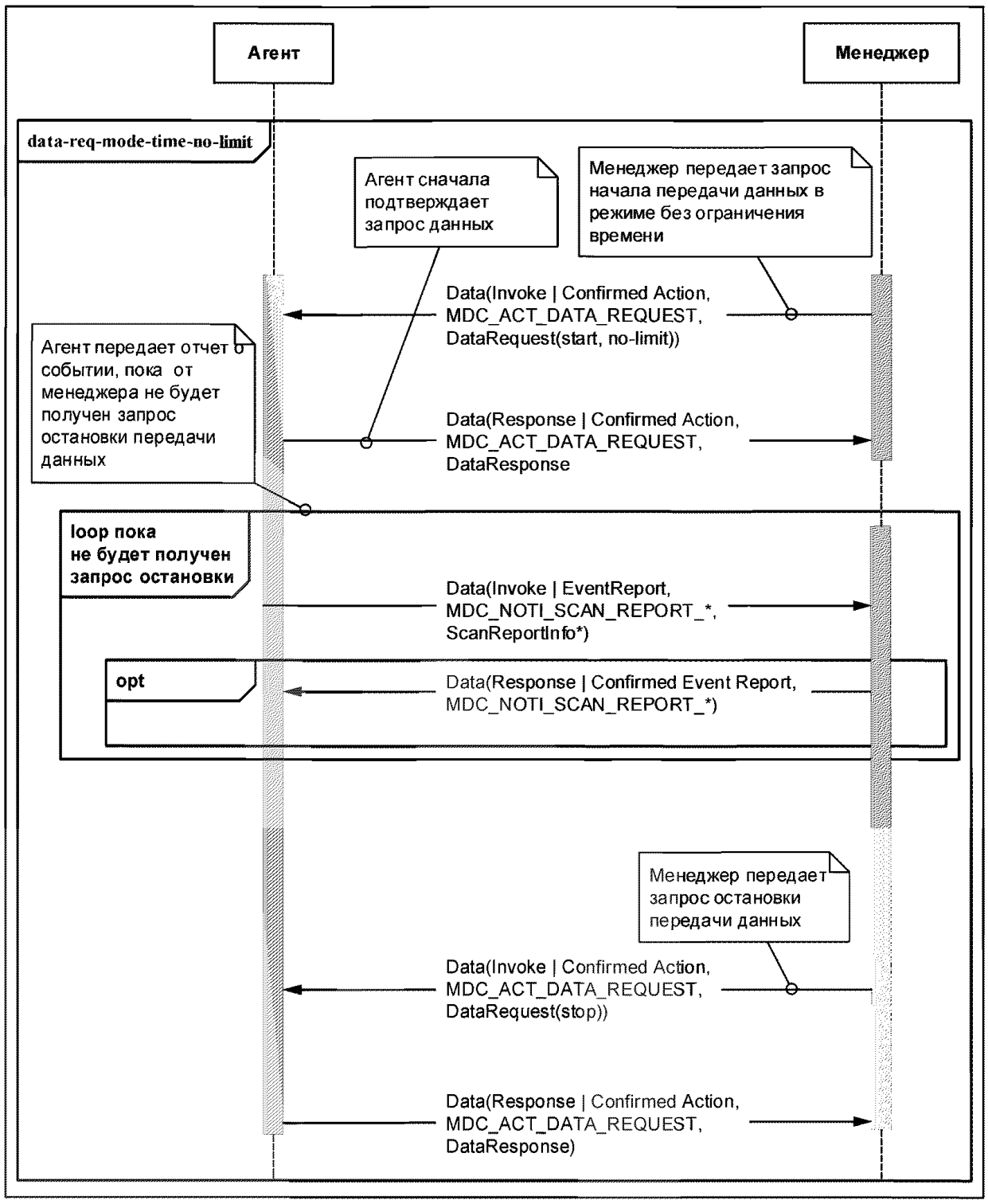


Рисунок 19 — Инициируемая менеджером передача результатов измерений (data-req-mode-time-no-limit)

8.9.3.3.7 Управление номерами отчетов о сканировании

Запрос данных, в котором установлен бит data-req-start-stop, формирует новый поток одного или нескольких результатов измерений от агента к менеджеру в контексте системы MDS. После формирования потока MDS агент создает для этого потока новый экземпляр счетчика scan-report-no. Для каждого потока обязательно наличие одного экземпляра счетчика scan-report-no, дифференцируемого с помощью data-req-id. Показания этого счетчика должны начинаться с 0 и увеличиваться на 1 для каждого отчета о событиях, отправленного в поток, начатый с 0. Если агент получает запрос на передачу данных, в котором установлен бит data-req-start-stop и содержится значение data-req-id, которое уже используется в потоке MDS, агент должен сбросить счетчик scan-report-no идентифицированного потока до 0. Если агент получает запрос на передачу данных, в котором установлен бит data-req-continuation, то счетчик scan-report-no должен прирастать без сброса.

При передаче результатов измерений, инициированной менеджером в режиме одиночного ответа (data-req-mode-single-rsp), ответное сообщение должно содержать 0 в поле scan-report-no структуры ScanReportInfo*. Это вызвано тем, что новый поток создается при каждом запросе режима data-req-mode-single-rsp и заканчивается после отправки запроса.

Напротив, передача данных, инициированная агентом от объектов системы MDS или Scanner, образует поток, завершаемый только в том случае, когда заканчивается ассоциация. Поэтому при передаче данных, инициированной агентом, счетчик scan-report-no начинается с 0, но не может быть сброшен менеджером в контексте ассоциации. Деактивирование атрибута Operational-State Scanner останавливает передачу отчетов о событиях (внутреннее наблюдение объектов Metric прекращается и возобновляется после повторного активирования атрибута Operational-State. Счетчик scan-report-no в этом случае продолжает отсчет с момента своей остановки.

8.9.3.3.8 Несколько потоков MDS, ссылающихся на один объект измерения

Агент может инициировать и получать запросы на потоки, которые ассоциируют data-req-ids с объектами Metric в контексте MDS. Если объект Metric, ассоциированный с несколькими потоками, генерирует результаты измерений, наблюдения данных должны сообщаться в каждом потоке.

В процессе ассоциации агент должен сообщить максимальное число параллельных потоков, инициированных менеджером, которые поддерживаются счетчиком data-req-init-manager-count. Менеджер должен ограничить число запрошенных параллельных потоков, инициированных менеджером, чтобы оно не превышало значение, сообщенное агентом. Если из-за недостатка ресурсов агенту не удается сформировать новый поток, инициированный менеджером, то в ответном сообщении агент должен присвоить data-req-result значение data-req-result-init-manager-overflow.

8.9.3.4 Передача постоянно хранящихся данных объектов Metric

8.9.3.4.1 Общие положения

Агент, реализующий один или несколько объектов PM-store, сообщает о существовании объекта PM-store на этапе конфигурирования. Менеджер использует эту информацию для запроса объектов PM-store агента. Взаимодействие менеджера и агента при извлечении информации из объекта PM-store описано в 8.9.3.4.2.

8.9.3.4.2 Передача постоянно хранящихся данных объектов Metric

а) Извлечение атрибутов PM-store. Когда агент и менеджер находятся в состоянии «Выполнение», менеджер может проверить конфигурацию, согласованную с агентом, чтобы определить количество объектов PM-store, хранящихся агентом. Менеджер может запросить информацию о каждом объекте PM-store для определения числа сегментов PM-segment, существующих в объекте PM-store. На рисунке 20 показана диаграмма последовательности операций этой процедуры. Менеджер отправляет агенту команду Get с запросом информации об атрибутах конкретного объекта PM-store. Для ссылки на требуемый объект PM-store менеджер использует номер дескриптора. Значение дескриптора помещается в поле obj-handle сообщения (A.10.4) и не присутствует в списке attribute-id-list запроса или ответа. Список attribute-id-list должен оставаться пустым, если запрашиваются все атрибуты объекта PM-store. Как вариант конкретные атрибуты объекта могут запрашиваться путем перечисления идентификаторов требуемых атрибутов, приведенных в таблице 10. Агент не обязан поддерживать эту возможность. Если эта возможность не реализована, агент должен ответить сообщением об ошибке (roer) со значением not-allowed-by-object параметра error-value.

Менеджер может анализировать атрибуты, чтобы определить конфигурации хранилища PM-store. Например, атрибут PM-Store-Capab описывает возможности хранения, а Number-Of-Segments указывает число сегментов, присутствующих в объекте PM-store. Полный список атрибутов и их определения см. в таблице 10.

Если агент поддерживает несколько экземпляров объектов PM-store, то запрос Get требуется для каждого объекта PM-store.

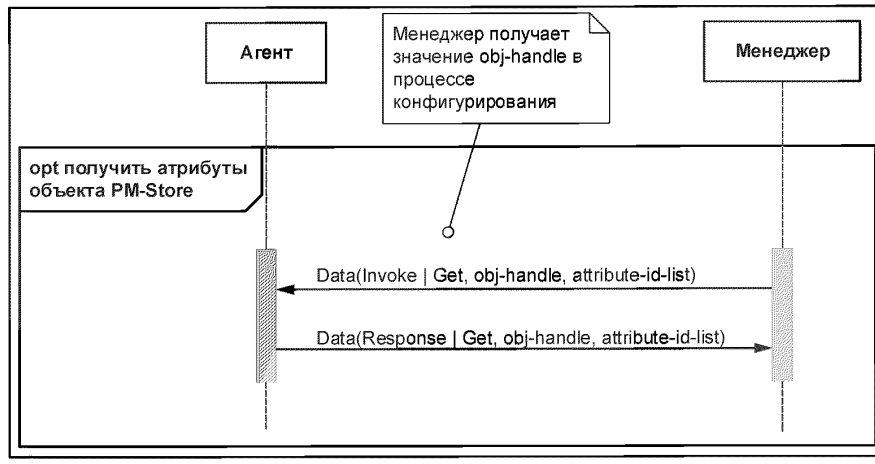


Рисунок 20 — Извлечение атрибутов PM-store

b) Извлечение информации из сегмента PM-segment. Менеджер извлекает информацию о сегментах объекта PM-store с помощью отправки команды ACTION.Get-Segment-Info или ACTION.Get-Segment-Id-List конкретному объекту PM-store (см. рисунки 21 и 22) с запросом предоставить информацию из всех сегментов определенного списка сегментов или любых сегментов в заданном диапазоне времени. Если в любом из трех указанных случаев сегменты отсутствуют, то агент возвращает пустой список. Агент должен поддерживать первый критерий отбора и может поддерживать второй и третий критерии отбора. Менеджер способен определить, обеспечивает ли агент поддержку критерия, проверяя поле pmsc-abs-time-select атрибута PM-Store-Capab, содержащегося в ранее извлеченной информации объекта PM-store.

Агент отвечает на команду ACTION.Get-Segment-Info списком номеров сегментов, за которым следует полный список атрибутов каждого сегмента. Агент отвечает на команду ACTION.Get-Segment-Id-List списком номеров экземпляров сегментов.

Если менеджер вызывает один из необязательных методов Get-Segment-Info или Get-Segment-Id-List, но агент не поддерживает определенное необязательное действие (список сегментов или диапазон сегментов в периоде времени), то агент должен возвратить сообщение roer DataApdu, в котором поле RoerErrorValue имеет значение «not-allowed-by-object».

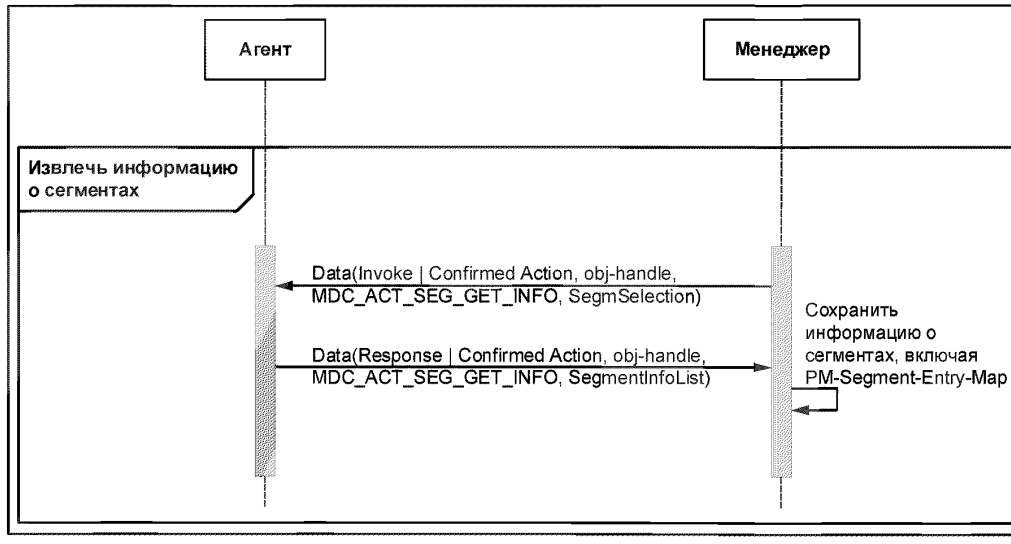


Рисунок 21 — Извлечение информации PM-segment с помощью Get-Segment-Info

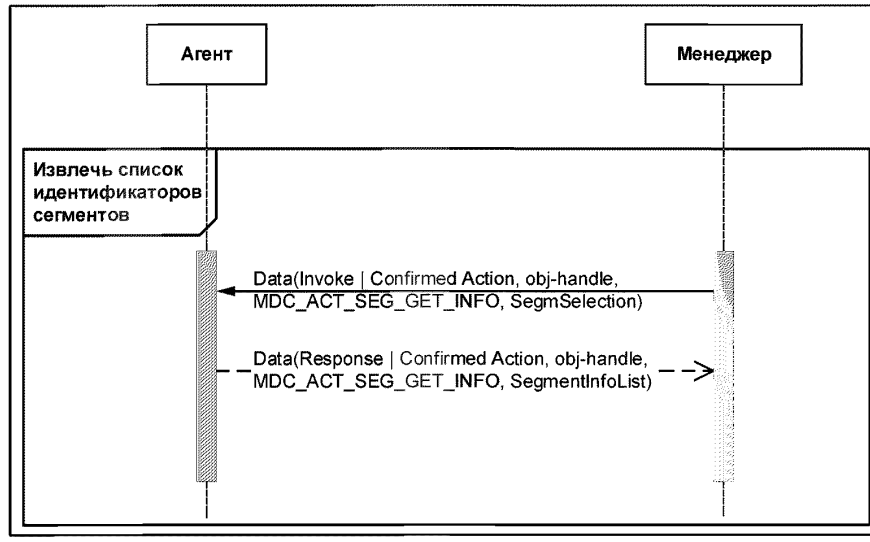


Рисунок 22 — Извлечение информации PM-segment с помощью Get-Segment-Id-List

c) Передача содержания сегмента PM-segment. Менеджер получает конкретные сегменты PM-segment с помощью метода Trig-Segment-Data-Xfer ACTION, инициирующего передачу данных (см. рисунок 23). На первом шаге менеджер отправляет метод ACTION агенту вместе с дескриптором объекта PM-store, к которому требуется доступ. Аргументом этого метода ACTION служит номер передаваемого экземпляра сегмента.

Агент должен принять решение о возможности обработки запроса. Он проверяет валидный номер сегмента, доступны ли данные сегмента (например, сегмент мог находиться в процессе обновления) или любые другие условия возникновения ошибок. При возникновении ошибки агент должен сообщить подходящий код ошибки в ответе и игнорировать запрос на передачу данных. В противном случае агент должен направить код ответа tsxr-successful, чтобы подтвердить получение запроса и возможность его обработки.

Менеджер может отправить сообщение вызова метода Trig-Segment-Data-Xfer ACTION в любое время. Но если менеджер отправляет это сообщение, когда сообщение вызова метода Clear-Segments ACTION находится в состоянии обработки, агент может сгенерировать ответное сообщение Trig-Segment-Data-Xfer ACTION с кодом возврата trig-segm-xfer-rsp = tsxr-fail-clear-in-process. Примером возможной отправки этого кода возврата является ситуация, при которой носителем данных объекта PM-store представляет собой одно флэш-устройство. Удаление данных флэш-устройства может привести к недоступности всего этого устройства.

Агент должен отправлять подтверждаемые отчеты о событиях Segment-Data-Event до тех пор, пока все записи сегмента PM не будут отправлены менеджеру или передача не будет прервана битом sevtsta-agent-abort или sevtsta-manager-abort, описанным далее. Агент заполняет структуру SegmentDataEvent информацией о передаваемом сегменте. Агент должен передавать все записи сегментов в порядке «первым пришел — первым ушел» (FIFO). Агент сообщает менеджеру дескриптор объекта PM-store и использует структуру SegmDataEventDescr для описания передаваемого номера сегмента, номера первой записи в поле segm-data-event-entries, числа записей в сообщении и информации о текущем состоянии. Агент должен всегда присваивать всем битам sevtsta-manager-* значение 0. Если сообщение содержит первую запись и/или последнюю запись данных, агент должен установить соответственно бит sevtsta-first-entry и/или sevtsta-last-entry. Если агент намерен прервать передачу, он должен присвоить биту sevtsta-agent-abort значение 1.

При передаче сегмента агент использует поле segm-data-event-entries для отправки всех записей. Агент должен начать с первой собранной записи, после нее указать следующую запись и так далее. Для оптимизации передачи агент должен упаковать в структуру события максимально возможное число записей. Каждая запись должна иметь формат, соответствующий структуре, определенной в атрибуте PM-Segment-Entry-Map сегмента PM-segment.

Менеджер, получивший отчет о событии, должен передать ответ SegmentDataResult, который должен содержать те же самые поля store-handle, segm-instance, segm-evt-entry-index и segm-evt-entry-count. В поле segm-evt-status менеджер должен установить бит sevtsta-manager-confirm.

Если агент установил бит sevtsta-agent-abort, менеджер должен подтвердить отключение агента, установив тот же бит. Если менеджеру требуется прервать обмен данными, он должен установить бит sevtsta-manager-abort.

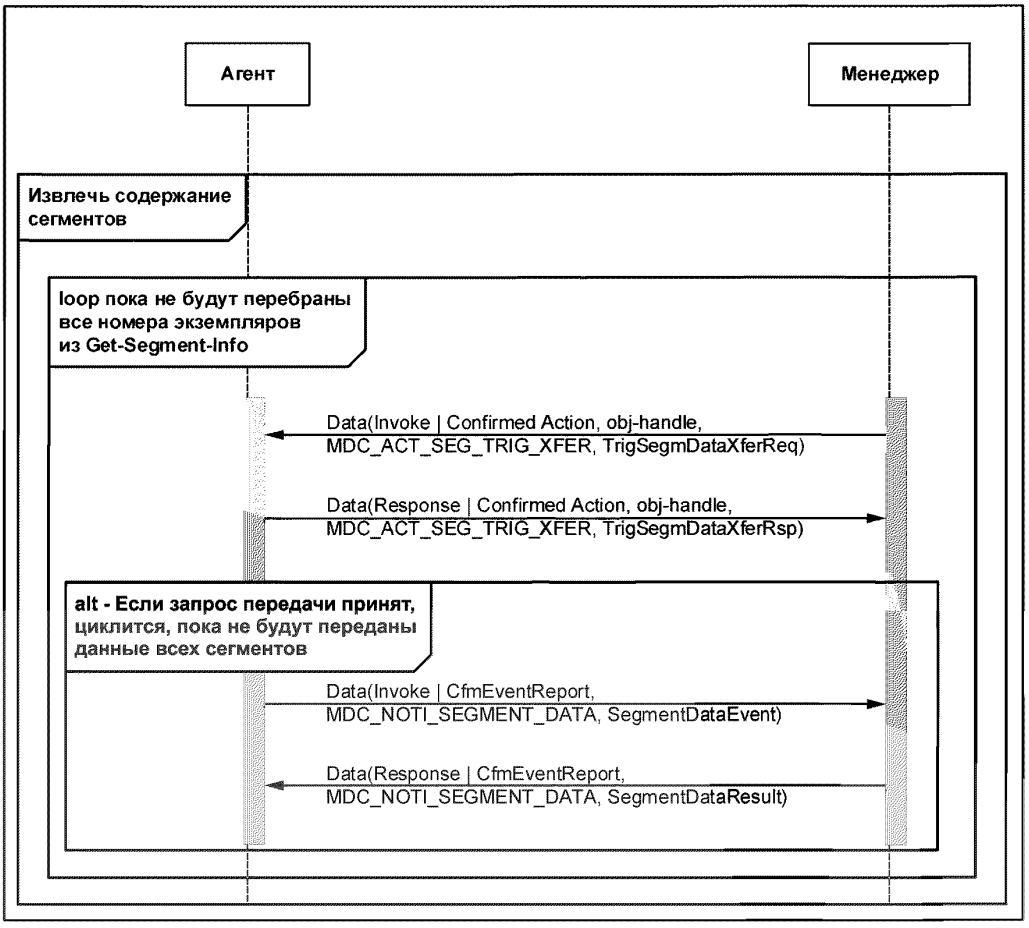


Рисунок 23 — Извлечение содержания сегментов PM-segment

d) Очистка сегментов PM-segment. Агент может поддерживать очистку сегментов PM-segment. Менеджер определяет, поддерживает ли агент какие-либо функции очистки, проверяя флаги pmsc-clear-segm-all-sup, pmsc-clear-segm-by-list-sup и pmsc-clear-segm-by-time-sup в атрибуте PM-Store-Capab.

Менеджер может очистить сегмент PM-segment в любое время, используя последовательность действий, показанную на рисунке 24. Обычно время для очистки сегмента наступает непосредственно после передачи менеджеру всего сегмента. Менеджер распознает это условие, когда получает структуру SegmEvtStatus с набором битов sevtsta-last-entry.

Когда менеджеру требуется очистить сегменты, он посылает агенту команду ACTION вместе с методом Clear-Segments и критериями отбора всех сегментов (все сегменты, конкретный список сегментов или любой сегмент в заданном диапазоне времени). Если агент поддерживает эту функцию, то должен поддерживать очистку всех сегментов (pmsc-clear-segm-all-sup), а также может поддерживать очистку сегментов из конкретного списка (pmsc-clear-segm-by-list-sup) и может поддерживать критерий выбора сегментов в диапазоне времени (pmsc-clear-segm-by-time-sup). Менеджер определяет поддерживаемые возможности, проверяя отдельные биты атрибута PM-Store-Capab.

Если менеджер вызывает метод Clear-Segments, но агент вообще не поддерживает эту функцию (флаги pmsc-clear-segm-remove-* в атрибуте PM-Store-Capab не установлены), то он должен возвратить сообщение roer DataApdu со значением «no-such-action» поля RoerErrorValue. Если менеджер вызывает метод Clear-Segments, но агент не поддерживает определенное действие (список сегментов или диапазон сегментов), то должен возвратить сообщение roer DataApdu со значением для «not-allowed-by-object» поля RoerErrorValue.

Агент, получивший команду Clear-Segment, может удалить все присутствующие записи и оставить или удалить сегмент. Менеджер определяет поддерживаемые возможности, проверяя бит pmsc-clear-segm-remove атрибута PM-Store-Capab.

Согласно 6.3.7.4 метод Clear-Segment может очищать не все заданные сегменты PM-segment. В целях проверки менеджер может сгенерировать действие Get-Segment-Info или Get-Segment-Id-List и запросы GET, чтобы отследить фактическую очистку и/или удаление любых сегментов.

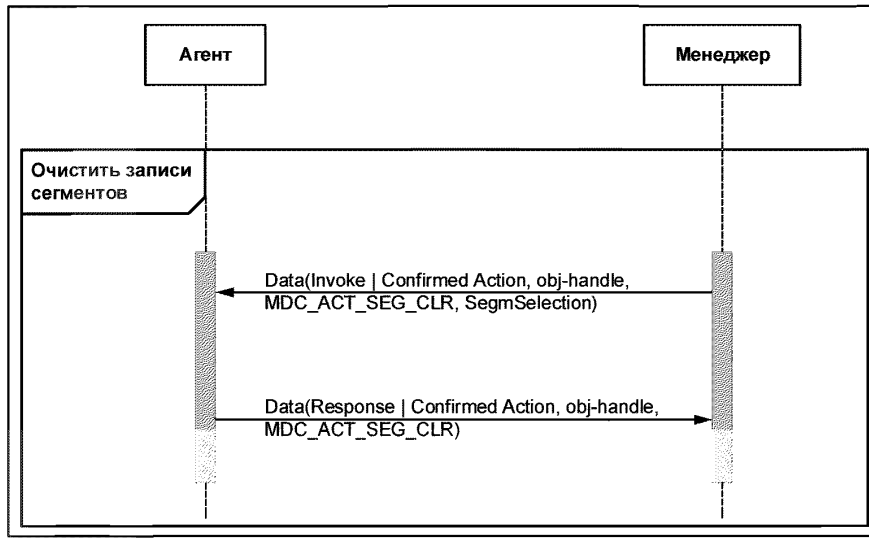


Рисунок 24 — Удаление записей сегментов

8.9.4 Условия выхода

Нормальный выход из состояния «Выполнение» происходит, когда агент или менеджер решает закончить ассоциацию. В этом случае агент или менеджер должен перейти в состояние «Завершает ассоциацию» и выполнить процедуру завершения ассоциации (см. 8.10).

Агент или менеджер, получивший запрос завершения ассоциации, должен отправить ответ о завершении ассоциации и перейти в состояние «Не ассоциирован».

Агент, выходящий из состояния «Выполнение» нормальным или аномальным способом, должен остановить все механизмы передачи данных, в том числе передачу результатов измерений, инициированную агентом или менеджером, передачу сегментов PM-segment и передачу данных объектов Scanner.

8.9.5 Условия ошибки

8.9.5.1 Общие положения

Как и в 8.9.4, агент, выходящий из состояния «Выполнение» нормальным или аномальным способом, должен остановить все механизмы передачи данных, в том числе инициированную агентом или менеджером передачу результатов измерений, передачу сегментов PM-segment или передачу данных объектов Scanner.

Если в какой-либо момент времени происходит тайм-аут транспортного уровня со стороны надежного транспортного уровня, агент или менеджер должен сделать следующее:

- При тайм-ауте надежных транспортных протоколов, зависящих от соединения (например, протокол TCP), осуществить возврат в состояние «Отсоединен», поскольку транспортные тайм-ауты передаются на верхние уровни как «индикация отсоединения транспортного уровня».

- При тайм-ауте протоколов, не зависящих от соединения (например, USB), выполняются: попытка восстановить транспортный канал, отправка сообщения о прекращении ассоциации своему партнеру и возврат в состояние «Не ассоциирован».

8.9.5.2 Подтверждаемое действие

После отправки сообщения вызова подтверждаемого действия менеджер должен ожидать ответного сообщения, подтверждающего действие, на протяжении периода TOca (тайм-аут: служба подтверждаемого действия) по умолчанию, если не применяется другой тайм-аут (например, TOclr-pms переопределяет TOca согласно 8.9.5.6). После истечения периода TOca менеджер должен отправить агенту сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

8.9.5.3 Подтверждаемый отчет о событиях

После отправки сообщения вызова подтверждаемого отчета о событии агент должен в течение периода TOcer-* (тайм-аут: служба подтверждаемых отчетов о событиях) ожидать ответного сообщения, подтверждающего отчет о событии. После истечения периода TOcer-* агент должен отправить менеджеру сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

Период TOcer-* определяется отдельно для каждого объекта. В рамках настоящего стандарта каждый объект, генерирующий отчеты о событиях, имеет свое значение тайм-аута, сообщаемое с помощью подходящего атрибута объекта:

- TOcer.mds (тайм-аут для объекта MDS) MDS.Confirm-Timeout;
- TOcer.pms (тайм-аут для объекта PM-store) Segm.Confirm-Timeout;
- TOcer.scan (тайм-аут для объекта Scanner) Scan.Confirm-Timeout.

8.9.5.4 Служба GET

После отправки сообщения вызова службы GET менеджер должен ожидать ответное сообщение службы GET в течение периода TOget (тайм-аут: служба GET). После истечения периода TOget менеджер должен отправить агенту сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

8.9.5.5 Подтверждаемая служба SET

После отправки сообщения вызова подтверждаемой службы SET менеджер должен ожидать ответное сообщение, подтверждающее ее выполнение, в течение периода TOcs (тайм-аут: подтверждаемая служба SET). После истечения периода TOcs менеджер должен отправить агенту сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

8.9.5.6 Специальные тайм-ауты

Помимо типичных вышеописанных тайм-аутов коммуникационной службы существуют три специальных тайм-аута, которые также используются в протоколе персональных медицинских приборов.

- TOclr-pms: специальный тайм-аут, связанный с очисткой объекта PM-store PMS.Clear-Timeout 3 с
- TOsp-mds: специальный тайм-аут между службами для объекта MDS
- TOsp-pms: специальный тайм-аут передачи сегмента PM-store Segm.Transfer-Timeout

Назначение TOclr-pms: после отправки сообщения вызова подтверждаемого действия (MDC_ACT_ SEG_CLR) менеджер должен ожидать ответного сообщения, подтверждающего действие, на протяжении периода TOclr-pms (тайм-аут: служба подтвержденного действия для очистки объекта PM-store). После истечения периода TOclr-pms менеджер должен отправить агенту сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

Назначение TOsp-mds: после отправки сообщения вызова подтверждаемого действия (MDC_ACT_ DATA_REQUEST, начало передачи, период времени, время = 0) менеджер должен ожидать сообщение вызова подтверждаемого отчета о событии на протяжении TOsp-mds (тайм-аут: специальный тайм-аут между службами для объекта MDS). После истечения периода TOsp-mds менеджер должен отправить агенту сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован».

После отправки сообщения вызова подтверждаемого действия (MDC_ACT_SEG_TRIG_XFER) и получения ответа менеджер должен на протяжении периода TOsp-pms (тайм-аут: специальный тайм-аут передачи сегмента объекта PM-store) ожидать сообщение вызова подтверждаемого отчета о событии (segm-evt-status=sevtsta-last-entry, segm-data-event-entries). После истечения периода TOsp-pms менеджер должен отправить агенту сообщение о прекращении ассоциации и вернуться в состояние «Не ассоциирован». Менеджер должен обрабатывать сообщение вызова подтверждаемого действия (MDC_ACT_SEG_TRIG_XFER), как и любое другое действие; он должен соблюдать процедуры тайм-аута, описанные в 8.9.5 2 для подтверждаемых действий.

8.10 Процедура завершения ассоциации

8.10.1 Общие положения

Процедура завершения ассоциации предоставляет агенту или менеджеру возможность нормального завершения ассоциации.

8.10.2 Условия входа

Агент или менеджер, решивший закрыть ассоциацию, должен вернуться в состояние «Не ассоциирован» и инициировать процедуру завершения ассоциации.

8.10.3 Нормальные процедуры

Находясь в состоянии «Не ассоциирован», агент или менеджер отправляет своему партнеру запрос завершения ассоциации и ожидает ответа. Запрос завершения ассоциации содержит информацию о причине завершения ассоциации (ReleaseRequestReason).

- Причина no-more-configurations (отсутствие конфигураций) используется агентом, находящимся в состоянии «Конфигурирующий», чтобы указать, что все возможные конфигурации испробованы и отклонены менеджером.

- Причина configuration-changed (изменение конфигурации) используется агентом, находящимся в состоянии «Выполнение», чтобы указать, что конфигурация агента изменена и больше нет возможности отправлять данные в соответствии с ранее согласованной конфигурацией. Обычно агент отвечает на это сообщение новым запросом ассоциации с другим значением Dev-Configuration-Id в поле dev-config-id, однако этот шаг не обязателен.

- Обычная причина (normal) используется агентом или менеджером для выхода из состояния «Выполнение» без указания особого условия.

Агент или менеджер, получивший запрос на завершение ассоциации с неопознанным идентификатором вызова invoke-id, должен отправить ответ с запросом завершения ассоциации и предположить, что ответа на этот запрос не последует.

8.10.4 Условия выхода

Агент или менеджер, получивший ответ на запрос завершения ассоциации, должен перейти в состояние «Не ассоциирован».

Если агент или менеджер получают запрос завершения ассоциации, находясь в состоянии «Завершает ассоциацию», он должен отправить ответ на запрос завершения ассоциации и оставаться в состоянии «Завершает ассоциацию», ожидая ответ на собственный запрос о завершении ассоциации.

8.10.5 Условия ошибки

После отправки сообщения о завершении ассоциации агент или менеджер должны ожидать ответа на запрос завершения ассоциации в течение периода TOrelease (тайм-аут: процедура завершения ассоциации). По прошествии периода TOrelease агент или менеджер должны отправить сообщение о прекращении ассоциации своему партнеру и вернуться в состояние «Не ассоциирован».

Агент или менеджер могут отправить или получить сообщение о прекращении ассоциации для других условий сбоя и должны немедленно перейти в состояние «Не ассоциирован».

8.11 Кодирование сообщений

Используемый в настоящем стандарте язык ACH.1 для описания абстрактного синтаксиса поддерживает преобразование в несколько форматов передачи данных. Менеджер и агент должны поддерживать правила MDER, сформулированные в стандарте ИСО/ИИЭР 11073-20101:2004 [B21]. Правила кодирования MDER представлены в приложении F вместе с дополнительными оптимизациями, специфичными для настоящего стандарта. Далее, при двоичной передаче должен использоваться сетевой порядок передачи байтов (кодирование big-endian — от старшего бита к младшему). Настоящий стандарт также позволяет агенту или менеджеру использовать альтернативные правила кодирования — PER [B23] и XER [B24].

Приложение G содержит один из примеров того, как структуры данных на языке ACH.1 можно перекодировать в синтаксис языка C.

Приложение H содержит дополнительные примеры двоичного кодирования сообщений, определенных в настоящем стандарте.

Все номенклатурные коды, использованные в настоящем стандарте, определены с помощью представления ссылочного идентификатора (например, MDC_...), однако во время передачи данных должны использоваться номенклатурные числовые коды независимо от согласованных правил кодирования. Приложение I содержит список определенных значений всех кодов, использованных в настоящем стандарте.

8.12 Координация времени

8.12.1 Общие положения

Агент может использовать четыре типа часов: абсолютного времени, базового времени со смещением по местному времени, относительного времени и относительного времени высокой точности. Во всех случаях информацию о возможностях часов агента и возможности синхронизации одних или нескольких часов с внешним источником времени можно определить по значениям атрибута Mds-Time-Info, приведенным в таблице 3. Все ссылки на биты в следующих подразделах относятся к данному атрибуту. Если агент имеет часы какого-либо типа и этот атрибут отсутствует, менеджер должен предположить, что разрешение по умолчанию равно 0 (то есть не известно). Агент, хранящий данные (в объекте PM-store или как «временно хранящиеся измерения»), должен сопоставлять с данными одну и только одну метку времени (Absolute-Time-Stamp, Base-Offset-Time-Stamp, Relative-Time-Stamp или HiRes-Time-Stamp).

8.12.2 Абсолютное время

8.12.2.1 Общие положения

Агенты с внутренними часами реального времени (RTC) должны сообщать об этой возможности, устанавливая бит mds-time-capab-real-time-clock (см. А.11.1). Агенты, поддерживающие действие Set-Time (см. 6.3.2.4 и А.4), должны установить бит mds-time-capab-set-clock.

Агенты могут поддерживать независимый метод синхронизации внутренних часов реального времени относительно внешних источников синхронизации. Используемый метод синхронизации не входит в область применения настоящего стандарта. Однако агент должен указать, синхронизируется ли абсолютное время, используя бит mds-time-capab-sync-abs-time. Если синхронизация поддерживается, то информация о протоколе, используемом для синхронизации внутренних часов реального времени (например, NTP и Simple Network Time Protocol (SNTP)), указывается в поле time-sync-protocol с помощью таких идентификаторов, как MDC_TIME_SYNC_NTPV3. Бит mds-time-state-abs-time-synced должен задаваться в том и только в том случае, если агент считает, что его атрибут Date-and-Time синхронизируется с внешним источником времени.

Агентам может понадобиться указать менеджеру, необходимо ли задать время с помощью действия Set-Time. Если агент предполагает, что его показания времени не точны, то он должен установить бит mds-time-mgr-set-time, при этом менеджер должен использовать команду действия Set-Time для установки абсолютного времени на часах агента. Команда Set-Time должна быть передана в течение периода времени TOconfig после получения атрибута из сообщения MDS Get. Получив действие Set-Time, агент должен сбросить бит mds-time-mgr-set-time перед отправкой подтверждения этого действия. Если в последующем агенту потребуется заново установить время, он должен установить бит mds-time-mgr-set-time и ожидать запроса GET от менеджера или отправить обновленный атрибут в отчете о событиях сканирования. Любой агент, запрашивающий менеджера об установке своего времени с помощью указания бита mds-time-mgr-set-time, не должен отправлять никакие отчеты о событиях сканирования до момента настройки времени менеджером. Данное ограничение позволяет агенту необходимым образом скорректировать постоянно и/или временно хранимые результаты измерений. Кроме того, такое ограничение помогает менеджеру и агенту работать с согласованной базой времени при получении данных.

В некоторых случаях агенту не требуется установка часов менеджером. Такая ситуация может возникнуть, когда агент синхронизирует свои часы через внешний источник времени или пользователь настроил часы локально. В этом случае биты mds-time-mgr-set-time и mds-time-capab-set-clock не должны устанавливаться и менеджер не должен пытаться настроить часы.

Если для передачи отчета агент использует базовое время со смещением, в указанном выше описании вместо действия Set-Time должно использоваться действие Set-Base-Offset-Time.

8.12.2.2 Сопоставимое время

В настоящем стандарте для всех четырех вариантов применения управления метками времени измерений используется понятие «сопоставимое время». Оно имеет следующие основные аспекты.

- Агент, сообщающий информацию о времени, должен подтвердить, что все измерения, переданные в одном наборе, имеют одну непрерывную шкалу времени. Для временно хранящихся результатов измерений набор содержит все измерения, передаваемые в одном отчете о событии. Для объекта PM-store набор эквивалентен сегменту PM-segment.

- Если набор измерений получен при иных настройках часов реального времени, агент должен сообщить данные вместе с количеством 1/100 секунд, которые необходимо добавить к каждому времени измерения, чтобы разместить их на той же шкале времени, что и текущий атрибут Date-and-Time агента.

Если агент не располагает информацией о связи между двумя шкалами времени до и после изменения времени, то он должен передать специальное значение 7FFFFFF в поле Date-and-Time-Adjustment, чтобы указать на наличие корректировки неизвестного количества времени, благодаря чему любой получатель этих данных уведомляется об этом и может предпринять подходящее согласованное действие.

- Два указанных выше аспекта применяются только в случае, если временная ось, использованная для отметки значений времени, значительно изменилась с точки зрения типа измерения. Другими словами, нет необходимости сообщать о небольших смещениях времени или незначительном изменении настроек часов, чтобы поддерживать синхронизацию с внешним источником времени.

Абсолютное время должно интерпретироваться как сопоставимое для агента следующим образом.

- После настройки атрибута Date-and-Time агент, ассоциированный с менеджером, должен отправить отчет о событии, содержащий новое значение атрибута Date-and-Time. Единственным исключением является случай, когда для изменения времени агента менеджер использует команду Set-Time. В этом случае агент может принять решение не посылать отчет о событии, так как менеджер уже знает об изменении времени.

- Если агент собирает временно хранимые измерения, то при изменении атрибута Date-and-Time он должен подтвердить, что в отчете о событии все измерения имеют общую непрерывную шкалу времени, то есть корректировка времени не выполнялась для интервала, ограниченного метками времени, содержащимися в отчете о событии. Кроме того, все отчеты о событиях, которые содержат результаты измерений, собранные до момента настройки текущего времени агента, должны иметь атрибут MDS Date-and-Time-Adjustment в качестве начальной информации, указанной в отчете о событии. Данный атрибут должен указывать количество 1/100 секунд, которые необходимо добавить к каждой метке времени в отчете о событии, чтобы выполнить согласование с текущими показаниями часов (например, если часы переведены на 60 минут вперед, в отчете это будет обозначаться как 360 000). Такая корректировка времени должна быть накопительной и определяться как разность между временем часов агента в момент наблюдения и временем часов агента в момент передачи отчетов о каждом наблюдении в этом отчете о событии.

- Если агент собирает результаты измерений в объекте PM-store, то при изменении атрибута Date-and-Time он должен подтвердить, что каждый сегмент PM-segment содержит результаты измерений только для одной непрерывной шкалы времени. Кроме того, в каждом сегменте PM-segment, содержащем результаты измерений, собранные при других настройках часов, должен присутствовать свой атрибут Date-and-Time-Adjustment.

- Необходимо отметить, что в тех случаях, когда результаты измерений получены в автономном режиме и настройки часов изменялись несколько раз перед выгрузкой данных, значение Date-and-Time-Adjustment является накопительным. Другими словами, когда после сбора результатов измерений часы переведены назад на 30 мин, а затем собраны дополнительные результаты измерений и часы переведены назад еще на 30 мин, первый набор данных необходимо указать в отчете со смещением — 60 мин, а второй со смещением — 30 мин.

8.12.3 Базовое время со смещением

В качестве формата меток времени настоящий стандарт использует базовое время со смещением в минутах по местному времени. Время, отображаемое пользователю, или местное время агента состоит из двух частей: базовое время и смещение в минутах относительно базового времени по местному времени. Такой формат предоставляет преимущества над абсолютным временем, поскольку может поддержать непрерывность шкалы времени и учесть изменения времени (например, летнее время (DST) или часовые пояса). Базовое время со смещением более эффективно, чем механизмы сопоставимого времени, поэтому его рекомендуется применять в приборах, хранящих результаты наблюдений, полученные на протяжении продолжительных периодов времени.

Однако обратите внимание, что если базовое время меняется, необходимо указать корректировку времени, использующую те же методы, что и для абсолютного времени.

Базовое время в настоящем стандарте представляется согласно определению протокола NTP. Базовое время представляется в виде количества секунд, прошедших с 00:00 часов 1 января 1900 года. Для этого используются 32-битовое беззнаковое целое число (без учета корректировочных секунд для високосного года аналогично протоколу NTP) и 16-битовая беззнаковая дробная часть поля секунд, выраженная как x/65 536 с. Поле 16-битового целого числа со знаком указывает смещение относительно базового времени в минутах по местному времени.

Базовое время необходимо задавать относительно некоторого начала отсчета времени таким образом, чтобы смещение относительно любого местного времени согласовывалось с максимальным значением поля смещения. Если базовое время согласуется с отчетом от 00:00 часов 1 января 1900

года в часовом поясе UTC (точность соответствует приложению), то в структуре MdsTimeCapState необходимо установить бит mds-time-state-bo-time-utc-aligned(14).

Если агент способен применять изменения перехода на летнее время подходящим образом, чтобы обеспечить корректное использование времени для каждого наблюдения, то в структуре MdsTimeCapState необходимо установить бит mds-time-dst-rules-enabled(15).

8.12.4 Относительное время

Агенты могут применять таймер относительного времени с разрешающей способностью до 125 мкс [младший значащий бит (LSB)]. Такая разрешающая способность достаточна для частот дискретизации до 8 кГц, позволяет измерять периоды относительного времени с высокой точностью и охватывает периоды времени до 6,2 дня. Если относительное время используется для временно хранящихся результатов измерений или для объекта PM-store, агенты должны гарантировать, что продолжительность периода хранения никогда не превышает разрешающую способность таймера (т. е. 6,2 дня). Такая гарантия со стороны агента позволяет менеджеру запрашивать текущее относительное время агента и вычислять, как давно проведены измерения. Если требуется большее время хранения, используются атрибуты абсолютного времени или атрибуты относительного времени высокой точности. Агенты должны указывать поддержку относительного времени, устанавливая бит mds-time-capab-relative-time в атрибуте Mds-Time-Info. Данный таймер должен инициализироваться перед ассоциацией. За исключением случаев перезапуска счетчика, он должен монотонно увеличивать показания и не изменять свое значение после инициализации. Точность фактического времени (то есть внутренний период обновления) определяется агентом, но должна соответствовать назначению прибора.

Агенты могут поддерживать метод синхронизации своего внутреннего таймера с внешним источником времени (например, пикосеть Bluetooth). Используемый метод синхронизации не входит в область применения настоящего стандарта. Однако агент должен указывать, синхронизирует ли он относительное время, используя бит mds-time-capab-sync-rel-time. Если синхронизация поддерживается, то бит mds-time-state-rel-time-synced должен устанавливаться только в том случае, когда агент считает, что его часы относительного времени синхронизируются с внешним источником. Все агенты, соединенные с одним и тем же менеджером и указывающие синхронизацию их внутренних часов, должны показывать одно и то же относительное время для событий, синхронизированных по времени.

Если агент предоставляет метку относительного времени для числового результата измерения, то она должна находиться строго в пределах заявленной точности синхронизации времени и времени считывания числового значения. При использовании в качестве времени события и текущего времени метка относительного времени может предоставить точную информацию об интервалах между событиями. Метка относительного времени может обеспечить точные измерения абсолютного времени, когда менеджер получает атрибуты относительного и абсолютного времени от объекта MDS агента и определяет время относительно собственных внутренних часов.

Если агент предоставляет метку относительного времени для массива измерений RT-SA, то метка времени должна относиться к первому считыванию в этом массиве и иметь точность в пределах заявленной точности атрибута относительного времени и времени считывания массива RT-SA.

Отчеты о событиях могут содержать метки относительного времени, указывающие момент возникновения события. Если объекты, произведенные от класса Metric и содержащиеся в отчете о событиях, не имеют собственную перекрывающую метку времени, то в качестве времени измерения должно использоваться время события. Если агент предоставил метку относительного времени для времени события, то ее точность находится в пределах заявленной точности атрибута относительного времени, времени события и любых атрибутов времени считываний, ассоциированных с событием.

8.12.5 Относительное время высокой точности

Агенты могут применять внутренний таймер высокой точности с разрешающей способностью до 1 мкс (младший значащий бит). Данный таймер высокой точности достаточен для частот дискретизации до 1 МГц, позволяет измерять периоды относительного времени с высокой точностью и охватывает периоды времени до 584 000 лет. Агенты должны указывать поддержку данного свойства, устанавливая бит mds-time-capab-high-res-relative-time в атрибуте Mds-Time-Info. Данный таймер должен инициализироваться перед ассоциацией. За исключением случаев перезапуска счетчика, он должен монотонно увеличивать показания и не изменять свое значение после инициализации. Точность фактического времени (то есть внутренний период обновления) определяется агентом, но должна соответствовать назначению прибора. Относительное время высокого разрешения должно сохранять частотную синхронизацию с относительным временем.

Агенты могут поддерживать метод синхронизации своего внутреннего таймера высокой точности с внешним источником времени (например, пикосеть Bluetooth). Используемый метод синхронизации не вхо-

дит в область применения настоящего стандарта. Однако агент должен указывать, синхронизирует ли он относительное время высокой точности, используя бит mds-time-capab-sync-hi-res-relative-time. Если синхронизация поддерживается, то бит mds-time-state-hi-res-relative-time-synced должен устанавливаться только в том случае, когда агент считает, что его часы относительного времени синхронизируются с внешним источником. Когда агент отключается от источника синхронизации часов, он должен сбросить бит синхронизации сразу после выхода за пределы точности параметров синхронизации часов.

Если агент предоставляет метку относительного времени высокой точности для числового результата измерения, то метка времени должна находиться строго в пределах заявленной точности атрибута относительного времени и времени считывания числового значения.

Если агент предоставляет метку относительного времени высокой точности для массива RT-SA, метка времени должна относиться к первому считыванию в данном массиве. Метка времени имеет точность в пределах заявленной точности атрибута относительного времени и времени считывания RT-SA.

9 Модель соответствия

9.1 Применимость

Ожидается, что другие стандарты серии ИСО/ИИЭР 11073 будут ссылаться на настоящий стандарт при определении приложений (например, обменов результатами персональных медицинских измерений) или функциональных коммуникационных профилей (например, профилей интероперабельности персональных медицинских приборов).

В частности, для создания интероперабельной системы необходима серия специализаций ИСО/ИИЭР 11073-104zz. Таким образом, интероперабельность требует проверки соответствия требованиям настоящего стандарта и набора специализаций реализуемых приборов. Специализации приборов определяют подходящую модель соответствия, которая включает в себя требования соответствия представлений персональных медицинских приборов настоящему стандарту. Специализации приборов используют информацию, специфическую для прибора, при определении дополнительных критериев соответствия, которые не входят в область применения настоящего стандарта.

Соответствие определениям настоящего стандарта указывается преимущественно для определенных прикладных или системных интерфейсов. Кроме того, поведение, указанное в настоящем стандарте (например, следование описанной в нем модели конечных автоматов), также является частью спецификации. С точки зрения соответствия поведение на этом уровне считается критически важным для обеспечения надлежащей и точной работы протокола в целом. Спецификации соответствия не распространяются на детали реализации, например на язык программирования, выделение уровней программного обеспечения, внутренние интерфейсы и так далее.

9.2 Спецификация соответствия

Настоящий стандарт по представлению персональных медицинских данных обладает высокой степенью гибкости применения описанной в нем модели для определенного медицинского прибора, в особенности для следующих областей:

- информационная модель конкретного прибора;
- использование атрибутов, диапазонов значений и доступа;
- использование расширенных коммуникационных служб (например, сканирование, периоды сканирования и конфигурации объектов Scanner).

Для обеспечения интероперабельности приложений и систем реализация, основанная на настоящем стандарте, должна предоставлять специфические подробности того, каким образом применяются определения настоящего стандарта в сочетании с требованиями соответствия любым производным специализациям приборов.

Такие спецификации принимают форму набора заявлений о соответствии реализации (ЗСР). ЗСР имеет форму перечня, раскрывающего подробности определенной реализации и указывающего предоставляемые возможности. Специфичные приложения или функциональные коммуникационные профили, основанные на настоящем стандарте, должны определять более конкретные требования к соответствию, дополняющие ЗСР, определенные в настоящем стандарте.

Примечание — ЗСР, определенные в последующих подразделах, способствуют пониманию деталей реализации. Однако их одних недостаточно для обеспечения интероперабельности приборов или приложений. Для нее необходимо учитывать дополнительные спецификации (например, тайминг, задержки и предположения о нагрузке системы). Такие спецификации не входят в область применения настоящего стандарта.

9.3 Заявления о соответствии реализации (ЗСР)

Обычно ЗСР представляются в виде таблиц. Шаблоны таблиц ЗСР представлены в таблицах 23 — 30. Таблицы должны заполняться и предоставляться в качестве полного документа заявления о соответствии.

Как правило, заголовки граф таблицы ICS содержат следующую информацию:

- индекс, представляющий собой идентификатор (например, номер) определенного свойства;
- свойство, которое кратко описывает характеристику, которой соответствует заявление о соответствии;
- ссылка на определение свойства (может быть не заполнена);
- статус, устанавливающий требования соответствия (например, требования к претендующей на соответствие реализации, связанные с рассматриваемым свойством). Для некоторых случаев настоящий стандарт не указывает требования соответствия, но, так или иначе, требует определения статуса конкретного свойства;
- поддержка, обеспечиваемая разработчиком и описывающая реализованные характеристики свойства;
- комментарий, содержащий дополнительную информацию, предоставленную разработчиком.

9.4 Общее соответствие

Таблицы 23—25 предназначены для использования при описании общего базового соответствия настоящему стандарту. В 9.4.1—9.4.3 сформулированы фундаментальные аспекты поддержки, необходимой прибору для обеспечения соответствия.

9.4.1 Общее ЗСР

В общем ЗСР верхнего уровня разработчик указывает версии/выпуски, поддерживаемые реализацией, а также некоторые определения высокого уровня, характеризующие поведение системы.

Общее ЗСР представлено в таблице 23.

Таблица 23 — Общая декларация о соответствии реализации

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
GEN-1	Описание реализации		Идентификация устройства/ приложения. Описание функциональности		
GEN-2	Стандартная версия документа	(Стандартные документы)	Обозначение поддерживаемых версий по стандарту IEEE 11073-20601	(Набор поддерживаемых редакций IEEE 11073-20601)	
GEN-3	Соблюдение соответствия - Уровень 1 -		Базовая декларация о соответствии прибора следующим требованиям стандарта IEEE 11073-20601™ к соответствию: A) все минимальные обязательные требования (некоторые из наиболее важных аспектов см. в пункте 9.4.2, таблица 24); B) все условно обязательные элементы реализованы в соответствии с заявленными условиями; C) все реализованные дополнительные элементы определяются как часть заявления о соответствии (например, в таблицах ЗСР, описывающих атрибуты (см. таблицу 28))	Да/Нет («Нет» подразумевает несоответствие)	

Окончание таблицы 23

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
GEN-4	Соблюдение соответствия - Уровень 2 -		В дополнение к GEN-3 прибор соответствует одной или нескольким специализациям, основанным на стандарте IEEE 11073-20601	(Перечислите ряд реализованных специализаций и профилей прибора по IEEE 11073-20601, а также подготовьте информацию, указанную в 9.5)	
GEN-5	Коммуникационный профиль и оборудование		Описание требований по сопряжению, предъявляемых к коммуникационной инфраструктуре и оборудованию		

Для каждой реализации должно быть предоставлено одно общее ЗСР.

9.4.2 Минимальные требования ЗСР

В таблице 24 указаны минимальные требования к соответствию настоящему стандарту.

Таблица 24 — Минимальные требования ИИЭР 11073-20601

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
REQ-1	Конечный автомат		-Обязательный- Имеет ли реализация строгое соответствие поведению конечного автомата персонального медицинского прибора, сформулированному в стандарте IEEE 11073-20601?	Да/Нет («Нет» подразумевает несоответствие)	
REQ-2	Сообщения протокола		-Обязательный- Соответствует ли реализация сообщениям протокола персональных медицинских приборов согласно стандарту IEEE 11073-20601?	Да/Нет («Нет» подразумевает несоответствие)	
REQ-3	Объекты		-Рекомендованный- Соответствуют ли все объекты стандарту IEEE 11073-20601 или специализациям прибора, основанным на стандарте IEEE 11073-20601? Настоятельно рекомендуется соответствие данному набору объектов, полей, значений и поведения	Да/Нет. Если «Нет», перечислите расширения, как описано в 9.5.2	
REQ-4	Кодирование		-Обязательный- Поддерживаются ли правила кодирования MDER? Сообщения протокола кодируются на основе описания ACH.1 и формата передачи данных с использованием правил кодирования. Поддержка MDER обязательна. Эти правила кодирования определены в приложении F стандарта IEEE 11073-20601. Существует возможность использования альтернативных правил кодирования. Перечислите все поддерживаемые правила кодирования	Да/Нет («Нет» подразумевает несоответствие) (Список поддерживаемых альтернативных правил кодирования)	

Окончание таблицы 24

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
REQ-5	Номенклатура		-Обязательный- Стандарт IEEE 11073-20601 основан на стандарте ISO/IEEE 11073-10101 [B16] для базовой номенклатуры. Стандарт IEEE 11073-20601 и связанные с ним специализации приборов вносят в эту номенклатуру свои дополнения. Соответствуют ли все номенклатурные коды одному из этих источников?	Да/Нет («Нет» подразумевает несоответствие)	
REQ-6	Транспорт		-Обязательный- Перечислите все классы транспорта (т.е. надежные и/или лучшие из возможных), которые поддерживаются реализацией. Выполнены ли для них все требования к транспорту, документированные в IEEE 11073-20601?	(Список транспортных классов) Да/Нет («Нет» подразумевает несоответствие)	

9.4.3 ЗСР для поддержки служб

ЗСР для поддержки служб определяет, какие из служб, описанных в сервисной модели, реализованы. Подобное ЗСР необходимо только для взаимодействующих устройств.

ЗСР для поддержки служб приведено в таблице 25.

Таблица 25 — ЗСР для поддержки служб

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
SRV-1	Служба GET	7.3	Поддерживает ли реализация службу GET? (условно обязательное)	Отправляет команду и/или принимает команду или не поддерживает	
SRV-2	Служба SET	7.3	Поддерживает ли реализация службу SET? (условно обязательное)	Отправляет команду и/или принимает команду или не поддерживает	
SRV-3	Подтверждаемая служба SET	7.3	Поддерживает ли реализация подтверждаемую службу SET? (необязательное)	Отправляет команду и/или принимает команду или не поддерживает	
SRV-4	Служба EVENT REPORT	7.3	Поддерживает ли реализация службу EVENT REPORT? (условно обязательное)	Отправляет команду и/или принимает команду или не поддерживает	
SRV-5	Подтверждаемая служба EVENT REPORT	7.3	Поддерживает ли реализация подтверждаемую службу EVENT REPORT? (условно обязательное)	Отправляет команду и/или принимает команду или не поддерживает	
SRV-6	Служба ACTION	7.3	Поддерживает ли реализация службу ACTION? (условно обязательное)	Отправляет команду и/или принимает команду или не поддерживает	
SRV-7	Подтверждаемая служба ACTION	7.3	Поддерживает ли реализация подтверждаемую службу ACTION? (необязательное)	Отправляет команду и/или принимает команду или не поддерживает	

В графе «Поддержка» заполненной таблицы следует указать, вызывает ли реализация службу (например, отправляет блок GET PDU), предоставляет службу (например, обрабатывает полученный блок GET PDU) или не реализует службу вообще.

Кроме того, должны быть перечислены специфичные ограничения (например, если определенная служба ограничена только одним классом объектов).

9.5 Дополнения/расширения ЗСР для устройств

Таблицы 26—30 предназначены для использования при описании ЗСР для любых дополнений или расширений, используемых устройством за пределами настоящего стандарта и его специализаций. Ожидается, что все условные или необязательные типы поведения сформулированы как часть соответствующего заявления о соответствии для конкретных специализаций приборов.

9.5.1 Общие дополнения/расширения ЗСР

Общие дополнения/расширения ЗСР определяют основную справочную информацию по объему поддерживаемых дополнений/расширений.

Таблица 26 — Общие дополнения/расширения ЗСР

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
ADD-1	Использование местных объектов	—	Используются ли в реализации объекты, не указанные в стандарте IEEE 11073-20601 или одной из перечисленных специализаций прибора?	Да/Нет [Если «Да»: ЗСР ОК ПМП (см. 9.5.2) должно быть использовано для разъяснения подробностей реализации]	
ADD-2	Использование кодов из стандарта ISO/IEEE 11073-10101 [B16], не входящих в номенклатуру 20601	—	Используются ли в реализации номенклатурные коды из стандарта ISO/IEEE 11073-10101 [B16], не указанные в стандарте IEEE 11073-20601 или одной из перечисленных специализаций прибора?	Да/Нет (Если «Да»: объясните в подходящем ЗСР, см. 9.5.6)	
ADD-3	Использование местных расширений номенклатуры	—	Используются ли в реализации местные расширения номенклатуры? Местные расширения номенклатуры допускаются, только если стандартная номенклатура не включает специфичные термины, требуемые приложению	Да/Нет (Если «Да»: объясните в подходящем ЗСР, см. 9.5.6)	
ADD-4	Формат полезной нагрузки	—	Были ли введены дополнительные форматы полезной нагрузки кроме тех, что указаны в стандарте IEEE 11073-20601 или одной из перечисленных специализаций прибора?	Да/Нет (Если «Да», то полностью объяснить цели и расположение. Следует описать на языке ACH.1)	

9.5.2 ЗСР объекта и класса (ОК ПМП) информационной модели предметной области персонального медицинского прибора

ЗСР ОК ПМП определяет, какие управляемые объекты из настоящего стандарта реализованы, и обозначает класс каждого объекта. Таблица 27 представляет собой всего лишь шаблон. Для каждого объекта, поддерживаемого реализацией, должна быть заполнена одна строка.

Таблица 27 — Шаблон ЗСР ОК ПМП

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
POC-*n*	Описание объекта	Класс объекта (например, Numeric)	Реализован	Обозначить ограничения (такие, как максимальное число поддерживаемых экземпляров класса)	

Для реализаций с предопределенными объектами букву n в графе «Индекс» следует заменить на дескриптор объекта. В ином случае она должна быть просто уникальным числом (1 .. m).

Все местные объекты должны быть обозначены, включая ссылку на определение объекта. Если публично доступная ссылка отсутствует, то определение объекта должно быть приложено к заявлению о соответствии.

В графе «Поддержка» следует указать все ограничения реализации объекта.

В качестве составной части ЗСР ОК ПМП должна быть предоставлена диаграмма включения объектов (диаграмма экземпляров класса).

9.5.3 ЗСР атрибутов ОК ПМП

Для каждого поддерживаемого объекта, указанного в ЗСР ОК ПМП, предоставляется ЗСР атрибутов ОК ПМП, определяющая условно обязательные, необязательные или расширенные атрибуты, используемые/поддерживаемые реализацией, включая все наследуемые атрибуты. Обязательные атрибуты не нужно указывать, так как их реализация требуется для соответствия.

Таблица 28 представляет собой всего лишь шаблон.

Таблица 28 — Шаблон для ЗСР атрибутов ОК ПМП

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
ATTR-*n*-*x*	Имя атрибута. Имена расширенных атрибутов должны также включать в себя идентификаторы атрибута	Если атрибут не указан в настоящем стандарте или в одной из перечисленных специализаций прибора, включите ссылку на структуру ACH.1	Реализован	Опишите: доступ, диапазоны значений, дополнительные ограничения значений	

Буква n в графе «Индекс» означает идентификатор управляемого объекта, для которого составлена таблица (например, индекс управляемого объекта, указанный в 9.5.2 для ЗСР ОК ПМП). Для каждого поддерживаемого управляемого объекта составляется отдельная таблица.

Буква x в графе «Индекс» представляет собой просто порядковый номер (1..m).

Должны быть указаны все атрибуты, не входящие в настоящий стандарт или одну из перечисленных специализаций прибора, включая ссылку на определение атрибута. Если публично доступная ссылка отсутствует, то определение атрибута должно быть приложено к заявлению о соответствии.

Поле описания доступа в графе «Поддержка» обозначено на тот случай, если реализация предоставляет службы доступа к атрибутам.

В графе «Поддержка» должны быть также указаны диапазоны значений атрибута (если применимо), советы по конкретным ограничениям доступа к атрибутам или по доступности атрибутов и по информации, а также статичность или динамичность атрибутов при данной реализации.

Примечание — Таблицы с определениями атрибутов в настоящем стандарте определяют минимальный обязательный набор атрибутов каждого объекта.

9.5.4 ЗСР поведения ОК ПМП

ЗСР поведения ОК ПМП определяет все методы реализованных объектов, которые можно вызвать с помощью службы ACTION. Таблица 29 представляет собой всего лишь шаблон. Отдельная таблица составляется для каждого объекта, поддерживающего специальные методы.

Таблица 29 — Шаблон ЗСР поведения ОК ПМП

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
ACT-*n*-*x*	Имя метода. Имена методов, не включенных в стандарты, должны также содержать идентификатор метода	Если метод не указан в настоящем стандарте или одной из перечисленных специализаций устройства, включите ссылку на структуру ACH.1		Специфичные ограничения	

Буква n в графе «Индекс» означает идентификатор управляемого объекта, для которого составлена таблица (например, индекс управляемого объекта, указанный в ЗСР ОК ПМП). Для каждого управляемого объекта, который поддерживает специфичные методы объектов (т.е. действия), составляется отдельная таблица.

Буква x в графе «Индекс» представляет собой просто порядковый номер (1..m).

Должны быть указаны все методы, не входящие в настоящий стандарт или одну из перечисленных специализаций прибора, включая ссылку на определение метода. Если публично доступная ссылка отсутствует, то определение метода должно быть приложено к заявлению о соответствии.

В графе «Поддержка» следует указать все ограничения метода.

9.5.5 ЗСР уведомлений ОК ПМП

ЗСР уведомлений ОК ПМП указывает все реализованные уведомления (обычно в форме службы EVENT REPORT), которые выдаются поддерживаемыми объектами. Таблица 30 представляет собой всего лишь шаблон. Таблица составляется для каждого объекта, поддерживающего специальные уведомления от объекта.

Таблица 30 — Шаблон для ЗСР уведомления от класса медицинского объекта (КМО)

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
NOTI-*n*-*x*	Имя и идентификатор уведомления	Ссылка на подпункт настоящего стандарта, где определено событие		Специфичные ограничения, идентификатор и описание каждого вовлеченного объекта	

Буква n в столбце «Индекс» соответствует идентификатору управляемого объекта, для которого составлена таблица (например, индекс управляемого объекта, указанного в POC ICS). Для каждого управляемого объекта, который поддерживает специфичные уведомления (т.е. события), выдаваемые объектами, составляется отдельная таблица.

Буква x в графе «Индекс» представляет собой просто порядковый номер (1 .. m).

Должны быть указаны все местные уведомления, включая ссылку на определение уведомления. Если публично доступная ссылка отсутствует, то определение уведомления должно быть приложено к заявлению о соответствии.

В графе «Поддержка» следует указать все ограничения уведомления.

9.5.6 ЗСР номенклатуры ОК ПМП

ЗСР номенклатуры ОК ПМП указывает все реализованные номенклатуры, использованные агентом. Таблица 31 представляет собой всего лишь шаблон. Для каждого элемента номенклатуры должна быть использована отдельная строка таблицы.

Таблица 31 — Шаблон ЗСР номенклатуры ОК ПМП

Индекс	Свойство	Ссылка	Статус	Поддержка	Комментарий
NOME-*n*	Имя и значение номенклатуры	Ссылка на подраздел стандарта или другую часть текста, где определяется или используется номенклатура		Описание способа использования номенклатуры. Описание каких-либо конкретных ограничений	

Буква n в столбце «Индекс» — порядковый номер для обеспечения уникальности (1 .. m).

Приложение А
(обязательное)

Определения ACH.1

А.1 Общие положения

Настоящее приложение содержит определения языка ACH.1, релевантные для протокола персонального медицинского прибора. Некоторые определения заимствованы из других частей серии стандартов ISO/IEEE 11073, а другие созданы специально для предметной области персональных медицинских приборов. Если необходимо понять, какие структуры заимствованы, а какие созданы заново, см. приложение J. Настоящее приложение предоставляет информацию обо всех структурах данных, необходимых для реализации настоящего стандарта.

Соглашение об именах, содержащихся в этом приложении, предполагает использование дефиса (-) для разделения слов в атрибутах и смешанных регистров в именах типов данных; однако конструкции, заимствованные из других спецификаций, подчиняются существующим правилам использования прописных букв и дефисов.

А.2 Общие типы данных

Настоящий подраздел определяет набор типов данных языка ACH.1, используемых в определениях объектов.

А.2.1 Типы данных целых чисел и строк битов

Для представления целых чисел определения объектов используют только типы данных фиксированного размера. Тип данных строки битов представляет собой битовое поле, где каждый отдельный бит имеет определенное значение (например, поля флагов). Используются следующие типы данных целых чисел и строк битов:

```
--
-- 8-битовое целое число без знака
--
INT-U8 ::= INTEGER (0..255)
--
-- 8-битовое целое число со знаком
--
INT-I8 ::= INTEGER (-128..127)
--

-- 16-битовое целое число без знака
--
INT-U16 ::= INTEGER (0..65535)
--
-- 16-битовое целое число со знаком
--
INT-I16 ::= INTEGER (-32768..32767)
--
-- 32-битовое целое число без знака
--
INT-U32 ::= INTEGER (0..4294967295)
--
-- 32-битовое целое число со знаком
--
INT-I32 ::= INTEGER (-2147483648..2147483647)
--
-- Если не указано иное, все неиспользуемые (зарезервированные) биты в любых структурах BITS-* должны задаваться равными 0 на
-- стороне отправителя и, если не указано иное, должны игнорироваться получателем, если они заданы равными 1.
--
-- 8-битовая строка
--
BITS-8 ::= BIT STRING (SIZE(8))
```

```
--
-- 16-битовая строка
--
BITS-16 ::= BIT STRING (SIZE(16))
--
-- 32-битовая строка
--
BITS-32 ::= BIT STRING (SIZE(32))
```

Необходимо отметить, что в определениях объектов целочисленные и битовые строковые типы данных с именованными константами или именованными битами для простоты используют вышеприведенные базовые типы данных. Такой подход обеспечивает сокращенное обозначение, но не допустим с точки зрения синтаксиса языка АСН.1. Его можно легко преобразовать в правильный синтаксис. Например, определению

```
NamedConstant ::= INT-U16 {
                        const1(1),
                        const2(2)
}
```

соответствует следующий правильный синтаксис языка АСН.1:

```
NamedConstant ::= INTEGER {
                    const1(1),
                    const2(2)
} (0..65535)
```

А.2.2 Тип идентификационных данных

Все элементы (например, классы, объекты и типы измерений), которым необходима уникальная идентификация, получают идентификатор объекта (OID). Ряд допустимых OID, используемых в настоящем стандарте, определен в ISO/IEEE 11073-10101 [B16]. Номенклатура состоит из набора разделов, каждый из которых охватывает определенное понятие и имеет собственные 16-битовые коды. Другими словами, конкретный код идентифицируется номером раздела и OID внутри этого раздела или его использование контекстно зависимо. Конкретный раздел, к которому принадлежит контекстно-зависимый код, указан в настоящем стандарте.

16-битовый тип идентификационных данных определяется следующим образом:

```
--
-- Тип OID согласно определению в номенклатуре
-- (не путать с OID языка ASN.1)
--
OID-Type ::= INT-U16 -- 16-битовый целочисленный тип
```

Местный раздел может содержать коды и идентификаторы, которые еще предстоит стандартизировать, или коды, задаваемые изготовителем.

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

А.2.3 Тип данных дескриптора

Тип данных дескриптора используется для эффективной и локально уникальной идентификации всех управляемых экземпляров объектов («локально уникальный» означает уникальность в пределах контекста одной системы MDS). Такой тип данных определяется следующим образом.

```
--
-- дескриптор
--
HANDLE ::= INT-U16
```

А.2.4 Тип данных номера экземпляра

Номер экземпляра используется для различения экземпляров классов или объектов одинакового типа или экземпляров объектов, которыми нельзя управлять непосредственно (используемых, например, в качестве идентифицирующих атрибутов объектов PM-segment).

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

А.2.5 Тип данных идентификатора

Тип данных идентификатора типа используется для идентификации типа всех элементов (например, классов, объектов и типов измерений). Он похож на тип OID (см. В.2.2), но содержит раздел номенклатуры и код для обеспечения уникальной идентификации элемента и должен использоваться в тех случаях, когда контекст не подразумевается. Этот тип данных определяется следующим образом.

```
--
-- Тип идентификатора
--
TYPE ::= SEQUENCE {
                    partition       NomPartition,
                code            OID-Type
}
--
-- Существуют следующие разделы номенклатуры:
--
NomPartition ::= INT-U16 {
                        nom-part-unspec(0),         -- не определен
                        nom-part-obj(1),            -- объектно-ориентированный раздел
                        nom-part-metric(2),         -- раздел метрик (SCADA)
                        nom-part-alert(3),          -- раздел уведомлений/событий
                        nom-part-dim(4),            -- раздел размерностей
                        nom-part-vattr(5),          -- раздел виртуальных атрибутов объектов
                                                    операций
                        nom-part-pgrp(6),           -- раздел идентификаторов групп параметров
                        nom-part-sites(7),          -- части тела и места измерений
                        nom-part-infrastruct(8),    -- раздел элементов инфраструктуры
                        nom-part-fef(9),            -- раздел форматов обмена файлами
                        nom-part-ecg-extn(10),      -- раздел расширений для
                                                    электрокардиограммы
                        nom-part-idco-extn(11),     -- расширения для имплантируемых
                                                    устройств — сердечных наблюдений
                        nom-part-phd-dm(128),       -- управление течением заболевания
                        nom-part-phd-hf(129),       -- здоровье и фитнес
                        nom-part-phd-ai(130),       -- одинокое старение
                        nom-part-ret-code(255),     -- раздел кодов возврата
                        nom-part-ext-nom(256),      -- идентификаторы других номенклатур и
                                                    словарей
                        nom-part-priv(1024)         -- местный раздел
}
```

А.2.6 Тип данных объявления значения атрибута (AVA)

Тип данных AVA полностью определяет атрибут объекта с помощью его идентификатора атрибута и значения. Поскольку структура значения зависит от атрибута, то ее тип определяется конструкцией ANY DEFINED BY. Этот тип данных поддерживает ряд служб, используемых для доступа к атрибутам объекта (например, GET и SET). Значения идентификаторов атрибутов определяются для каждого типа объекта в графе «Идентификатор атрибута» таблиц идентификации объектов (см., например, таблицы 3, 6—10, 13—16 и 18). Структура, использованная для attribute-value, определена в столбце «Тип атрибута» тех же таблиц. Тип данных AVA определяется следующим образом:

```
--
AVA-Type ::= SEQUENCE {
                attribute-id        OID-Type,       -- Идентификатор атрибута заимствуется из раздела nom-part-obj
                attribute-value     ANY DEFINED BY attribute-id
}
```

А.2.7 Тип данных списка атрибутов

Зачастую необходим список пар «идентификатор атрибута — значение атрибута». Тип данных списка атрибутов — специальный тип данных, предусмотренный для этой ситуации и определяемый следующим образом:

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

А.2.8 Тип данных списка идентификаторов атрибутов

Список идентификаторов атрибутов имеет частое применение. Тип данных списка идентификаторов атрибутов — специальный тип, предусмотренный для удобства и определяемый следующим образом.

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

А.2.9 Тип данных с плавающей точкой (FLOAT-Type)

Тип данных с плавающей точкой (FLOAT-Type) определен для представления числовых значений нецелочисленного типа. Тип FLOAT-Type определен как 32-битовое значение с 24-битовой мантиссой и 8-битовой экспонентой. Полное определение этого типа данных см. в F.7. Такой тип данных определяется следующим образом:

```
--
-- 32-битовый тип с плавающей точкой; целочисленный тип является только заполнителем
--
FLOAT-Type ::= INT-U32
```

32 бита содержат 8-битовую экспоненту со знаком и основанием 10, за которой следует 24-битовое целое число со знаком (мантисса).

Специальные значения закреплены для выражения следующего:

— NaN (не число) [экспонента 0, мантисса +(2**23 –1) → 0x007FFFFF]
— NRes (не при этой точности) [экспонента 0, мантисса –(2**23) → 0x00800000]
— + INFINITY [экспонента 0, мантисса +(2**23 –2) → 0x007FFFFE];
— – INFINITY [экспонента 0, мантисса –(2**23 –2) → 0x00800002]
— Зарезервировано для последующего использования [экспонента 0, мантисса –(2**23 –1) → 0x00800001].

А.2.10 Тип данных укороченного формата с плавающей точкой (SFLOAT-Type)

Тип данных укороченного формата с плавающей точкой (SFLOAT-Type) определен для представления числовых значений нецелочисленного типа с ограниченной точностью. Тип SFLOAT-Type определен как 16-битовое значение с 12-битовой мантиссой и 4-битовой экспонентой. Полное определение этого типа данных см. в F.7. Такой тип данных определяется следующим образом:

```
--
-- 16-битовый тип с плавающей точкой; целочисленный тип является только заполнителем
--
SFLOAT-Type ::= INT-U16
```

16-битовое значение содержит 4-битовую экспоненту с основанием 10, за которым следует 12-битовая мантисса. Каждый из них выражен в форме дополнительного кода.

Специальные значения закреплены для выражения следующего:

— NaN [экспонента 0, мантисса +(2**11 –1) → 0x07FF]
— NRes [экспонента 0, мантисса –(2**11) → 0x0800]
— + INFINITY [экспонента 0, мантисса +(2**11 –2) → 0x07FE]
— – INFINITY [экспонента 0, мантисса –(2**11 –2) → 0x0802]
— Зарезервировано для последующего использования [экспонента 0, мантисса –(2**11 –1) → 0x0801]

А.2.11 Тип данных относительного времени

Тип данных относительного времени представляет собой счетчик времени, используемый для определения относительного времени между событиями. Этот тип данных используется для позицирования событий друг относительно друга и определяется следующим образом:

```
--
-- Относительное время имеет точность до 125 мкс (LSB), достаточную для считывания
-- показаний с частотой до 8 кГц, и охватывает периоды времени до 6,2 дня.
-- Значение 0xFFFFFFFF должно использоваться, если агенту необходимо отправить относительное время в
   структуре на языке ACH.1, но у него отсутствует поддержка часов относительного времени.
--
RelativeTime ::= INT-U32
```

Следует учитывать, что фактическая точность времени определяется агентом.

А.2.12 Тип данных относительного времени высокой точности

Тип данных относительного времени высокой точности представляет собой счетчик времени высокой точности, используемый для определения относительного времени между событиями. Этот тип данных используется для позицирования событий друг относительно друга и определяется следующим образом:

```
--
-- Время высокой точности имеет точность до 1 мкс и может охватывать
-- периоды времени более 584 000 лет. Теоретически это может моделироваться как INT-U64,
```

```
-- однако из-за ограничений компиляторов языка ASN.1, поддержки 64-битовых целых чисел
-- встроенными устройствами и спецификаций MDER вместо него использован тип строки байтов.
--
HighResRelativeTime ::= OCTET STRING (SIZE(8))
```

Необходимо отметить, что фактическая точность времени определяется агентом.

```
--
-- Сдвиг абсолютного времени имеет точность до 1/100 секунды и может представлять
-- сдвиги времени вперед или назад на 44 505 лет. Теоретически это можно моделировать как INT-I48,
-- однако из-за потенциальных ограничений компиляторов языка ASN.1, поддержки 48-битовых целых чисел
-- встроенными устройствами и спецификаций MDER вместо него использован тип строки байтов.
--
AbsoluteTimeAdjust ::= OCTET STRING (SIZE(6))
```

A.2.13 Тип данных абсолютного времени

Тип данных абсолютного времени определяет время суток с точностью до 1/100 секунды. Поле часов должно представляться в 24-часовом формате времени (т. е. от 0 до 23). Значения в структуре должны представляться с использованием двоично-десятичного кодирования (т. е. 4-битовых полубайтов). Например, 1996 год должен представляться шестнадцатеричным значением 0x19 в поле века и шестнадцатеричным значением 0x96 в поле года. Такой формат легко преобразуется в формат, ориентированный на символы или целые числа. Тип данных абсолютного времени определяется следующим образом:

```
--
AbsoluteTime ::= SEQUENCE {
        century         INT-U8,
        year            INT-U8,
        month           INT-U8,
        day             INT-U8,
        hour            INT-U8,
        minute          INT-U8,
        second          INT-U8,
        sec-fractions   INT-U8      -- 1/100 секунды, если доступно
}
```

Необходимо отметить, что фактическая точность времени определяется агентом (например, если точность часов равна 1 с, доли секунды всегда нулевые). Агентам следует иметь точность 1 с или более высокую.

A.2.14 Тип данных базового времени со смещением

Тип данных базового времени со смещением определяет время суток и содержит поле смещения времени, позволяющее указать разность в минутах между базовым и местным временем. Базовое время кодируется как количество секунд, прошедшее с полуночи 1 января 1900 года, указанное в формате INT-U32, а доля секунды x/65 536 — в формате INT-U16. Поле смещения времени определяется как INT-I16. Тип данных базового времени со смещением определяется следующим образом:

```
--
BaseOffsetTime ::= SEQUENCE {
        bo-seconds      INT-U32,
        bo-fraction         INT-U16,
        bo-time-offset  INT-I16
}
```

A.2.15 Тип данных состояния выполнения

Тип данных состояния выполнения определяет, активирован или деактивирован определенный объект или другое свойство.

```
--
OperationalState ::= INT-U16 {
        disabled(0),
        enabled(1),
        notAvailable(2)         -- значение notAvailable не используется в этом стандарте
}
```

A.3 Типы данных атрибутов

A.3.1 Атрибуты MDS

```
--
-- SystemModel содержит имя изготовителя и информацию о модели, указанную изготовителем.
-- Поле номера модели предполагает наличие числа, однако требование о наличии
```

```
-- в поле числовых значений отсутствует. Форматы имени изготовителя и строк номеров моделей
-- определяются поставщиком агента, но должны позволять печать в кодировке ASCII.
--
SystemModel ::= SEQUENCE {
                manufacturer        OCTET STRING,      -- Размер строки должен быть четным
                model-number        OCTET STRING       -- Размер строки должен быть четным
}
-- Атрибут ProductionSpec содержит серийные номера, номера деталей, выпусков и т. д.
-- Следует отметить, что агент может состоять из нескольких компонентов, поэтому prod-spec должен
-- представляться в виде печатной строки ASCII формата «spec-type: vendor-specified-str», где spec-type
-- заменяется строковым представлением spec-type. Формат vendor-specified-str
-- определяется поставщиком.
--
ProductionSpec ::= SEQUENCE OF ProdSpecEntry
ProdSpecEntry ::= SEQUENCE {
                spec-type           INT-U16 {
                                    unspecified(0),
                                    serial-number(1),
                                    part-number(2),
                                    hw-revision(3),
                                    sw-revision(4),
                                    fw-revision(5),
                                    protocol-revision(6),
                                    prod-spec-gmdn(7)      -- см. примечание по GMDN ниже
                                    },
                  component-id      PrivateOid,
                  prod-spec         OCTET STRING           -- размер строк должен быть четным
}
-- Примечание — Международная номенклатура медицинских приборов (Global Medical Device Nomenclature,
GMDN)
-- основана на стандарте ISO 15225 [B13] и разработана при содействии CEN TC257 SC1.[11]
--
-- PowerStatus определяет, работает ли прибор от аккумулятора или сети. Старшие биты определяют состояние
-- заряда.
--
PowerStatus ::= BITS-16 {
        onMains(0),
        onBattery(1),
        chargingFull(8),
        chargingTrickle(9),
        chargingOff(10)
}
--
-- Результатом всех измерений характеристик аккумулятора являются значения с их размерностями.
-- Сведения о допустимых единицах см. в описании Remaining-Battery-Time (таблица 3).
--
BatMeasure ::= SEQUENCE {
                value               FLOAT-Type,
                unit                OID-Type            -- из раздела nom-part-dim
}
```

А.3.2 Атрибуты класса Metric

Данная группа содержит заимствованные определения атрибутов, применяемые к объектам Numeric, Enumeration и RT-SA.

```
--
-- Состояние измерения
-- Значения битов 14 и 15 используются в других стандартах ISO/IEEE 11073 и не должны использоваться для
-- других целей.
--
```

[11] Дополнительная информация об этом техническом комитете доступна по адресу:

gost.e-ecolog.ru

```
MeasurementStatus ::= BITS-16 {
                        invalid(0),
                        questionable(1),
                        not-available(2),
                        calibration-ongoing(3),
                        test-data(4),
                        demo-data(5),
                        validated-data(8),              -- релевантный, например, в архиве
                        early-indication(9),            -- ранняя оценка значения
                        msmt-ongoing(10),               -- указывает, что новое измерение еще проводится
                                                        -- (эпизодическое)
                        msmt-value-exceed-boundaries(14), -- указывает, что результат измерения находится за
                                                        -- пределами допустимого диапазона.
                        msmt-state-ann-inhibited(15)    -- указывает, что индикация порогового значения
                                                        -- отключена и не должна вызывать отображаемое
                                                        -- оповещение. Если этот бит установлен, бит 14
                                                        -- не должен устанавливаться.
}
```

A.3.3 Атрибуты класса Numeric

```
--
-- NuObsValue (числовое наблюдаемое значение) всегда содержит информацию об идентификации, состоянии и
   размерности.
--
NuObsValue ::= SEQUENCE {
                    metric-id           OID-Type,       -- Данный код извлекается из раздела, указанного в
                                                        -- атрибуте Metric::Type объекта Numeric.
                    state               MeasurementStatus,
                    unit-code           OID-Type,       -- Из раздела номенклатуры размерностей nom-part-dim
                    value               FLOAT-Type
    }
--
-- Наблюдаемое значение для составных числовых значений
--
NuObsValueCmp ::= SEQUENCE OF NuObsValue
```

A.3.4 Атрибуты класса RT-SA

```
--
-- SaSpec описывает массив считываний.
--
SaSpec ::= SEQUENCE {
                array-size  INT-U16,        -- количество считываний в каждом периоде обновления объекта Metric
                sample-type         SampleType,
                flags               SaFlags
}
--
-- SampleType описывает одно считывание в массиве наблюдаемых значений.
--
SampleType ::= SEQUENCE {
                sample-size     INT-U8,     -- например, 8 для 8-битовых считываний, 16 для 16-битовых,
                                            -- должен делиться на 8
                significant-bits  INT-U8    -- определяет значащие биты в одном считывании
                {signed-samples(255)}       -- если значение равно 255, считывания
                                            -- в Simple-Sa-Observed-Value и
                                            -- lower-scaled-value и upper-scaled-value в
                                            -- ScaleRangeSpec должны интерпретироваться как
                                            -- целые числа со знаком, выраженные в форме дополнительного кода.
}
```

```
--
-- SaFlags определяет дополнительные свойства формы биосигнала.
--
SaFlags ::= BITS-16 {
        smooth-curve(0),        -- для оптимального отображения используется алгоритм сглаживания
        delayed-curve(1),       -- кривая отложена (не реальное время)
        static-scale(2),        -- ScaleRangeSpec не меняется
        sa-ext-val-range(3)     -- Не значащие биты считывания не равны 0, например,
                                -- когда они используются для аннотаций или маркеров.
                                -- Получатель должен применить битовую маску, чтобы извлечь
                                -- значащие биты из считывания. Если считывания имеют
                                -- знак, бит sa-ext-val-range не должен быть установлен
                                -- (так как по определению в поле не могут содержаться
                                -- не значащие биты).
}
--
-- Атрибут определения масштаба и диапазона описывает преобразование масштабируемых
-- значений в абсолютные и определяет ожидаемый диапазон абсолютных и масштабируемых значений.
-- В зависимости от диапазона масштабируемых значений существует множество типов атрибутов.
-- Отображение значений считываний в преобразованные абсолютные значения выполняется по
-- формуле Scale-and-Range-Specification, приведенной в 6.3.5.3.
--
ScaleRangeSpec8 ::= SEQUENCE {
                    lower-absolute-value    FLOAT-Type,
                    upper-absolute-value    FLOAT-Type,
                    lower-scaled-value      INT-U8,         -- учтите, что интерпретируется как INT-I8,
                    upper-scaled-value      INT-U8          -- если атрибут Sa-Specification
                                                            -- указывает считывания со знаком
}
ScaleRangeSpec16 ::= SEQUENCE {
                    lower-absolute-value    FLOAT-Type,
                    upper-absolute-value    FLOAT-Type,
                    lower-scaled-value      INT-U16,        -- учтите, что  интерпретируется как INT-I16,
                    upper-scaled-value      INT-U16         -- если атрибут Sa-Specification
                                                            -- указывает считывания со знаком
}
ScaleRangeSpec32 ::= SEQUENCE {
                    lower-absolute-value    FLOAT-Type,
                    upper-absolute-value    FLOAT-Type,
                    lower-scaled-value      INT-U32,        -- учтите, что  интерпретируется как INT-I32,
                    upper-scaled-value      INT-U32         -- если атрибут Sa-Specification
                                                            -- указывает считывания со знаком
}
```

A.3.5 Атрибуты класса Enumeration

```
--
-- EnumObsValue описывает наблюдаемое значение Enumeration
--
EnumObsValue ::= SEQUENCE {
                metric-id       OID-Type,       -- Данный код извлекается из раздела, определенного в
                                                -- атрибуте Metric-Id-Partition (при наличии значения). Иначе
                                                -- код извлекается из того же раздела, что и
                                                -- атрибут Type.
                state           MeasurementStatus,
                value           EnumVal         -- поддерживает различные типы данных величин
}
-- EnumVal используется для обозначения различных конкретных наблюдаемых типов данных следующим образом
-- (необходимо отметить, что тип измерения кодируется в структуре верхнего уровня EnumObsVal::metric-id).
--
--              enum-obj-id:            используется для предоставления OID Metric, например, в
--                                      качестве аннотации или
```

```
--                                  другого события, указанного в разделе Metric::Type
--          enum-text-string:       используется для предоставления строки произвольного текста
--                                  (например, сообщения о состоянии);
--          enum-bit-str:           для кодирования значений битовых строк; тип данных битовой строки
--                                  должен определяться отдельно, например в рамках номенклатуры или
--                                  стандарта для конкретного устройства
-- Описание других типов данных, определенных в ISO/IEEE 11073-10201:2004 [B17], отсутствует в настоящем
-- стандарте, поскольку они не имеют отношения к персональным медицинским приборам.
--
EnumVal ::= CHOICE {
        enum-obj-id         [1] OID-Type,       -- Данный код извлекается из раздела, определенного в
                                                -- атрибуте Enum-Observed-Value-Partition
                                                -- (при наличии значений). Иначе код извлекается из того же
                                                -- раздела, что и атрибут Type.
        enum-text-string    [2] OCTET STRING,           -- печатаемый текст ASCII, четный размер
        enum-bit-str        [16] BITS-32                -- битовая строка.
}
```

A.3.6 Атрибуты класса Scanner

Нет.

A.3.7 Атрибуты класса CfgScanner

```
--
-- ConfirmMode определяет, используются ли отчеты о подтверждаемых или не подтверждаемых событиях.
--
ConfirmMode ::= INT-U16 {
        unconfirmed(0),
        confirmed(1)
}
```

A.3.8 Атрибуты класса EpiCfgScanner

Нет.

A.3.9 Атрибуты класса PeriCfgScanner

Нет.

A.3.10 Атрибуты классов PM-store и PM-segment

```
--
-- StoSampleAlg описывает извлечение и усреднение считываний.
--
StoSampleAlg ::= INT-U16 {
        st-alg-nos(0),                      -- не указано иное
        st-alg-moving-average(1),
        st-alg-recursive(2),
        st-alg-min-pick(3),
        st-alg-max-pick(4),
        st-alg-median(5),
        st-alg-trended(512),                -- используются значения тренда
        st-alg-no-downsampling(1024),       -- не предполагает усреднение, это реально
                                            -- измеренное считывание
        st-alg-manuf-specific-start(61440), -- начало диапазона, зарезервированного для
                                            -- специфики изготовителя
        st-alg-manuf-specific-end(65535)    -- конец, зарезервированного для специфики изготовителя
}
```

A.4 Типы данных, связанные с методом действия ACTION

```
--
-- SetTimeInvoke выбирает устанавливаемую дату и время
--
SetTimeInvoke ::= SEQUENCE {
        date-time   AbsoluteTime,
        accuracy    FLOAT-Type  -- точность установки времени (например, ошибка 2 мин);
```

```
                                        -- значение определяется в секундах. Данный параметр
                                        -- унаследован из ISO/IEEE 11073-10201:2004
                                        -- [B17], однако не используется. Следовательно, значение параметра
                                        -- должно равняться нулю (0).
}
--
-- SetBOTimeInvoke выбирает устанавливаемую дату и время с использованием формата базового времени со
-- смещением. Если поля секунд и их долей одновременно содержат значение 0x0, они игнорируются при
-- выполнении действия. Изменяется только смещение.
--
SetBOTimeInvoke ::= SEQUENCE {
        date-time       BaseOffsetTime
}
--
-- SegmSelection выбирает сегменты PM-segment, являющиеся предметами метода.
--
SegmSelection ::= CHOICE {
        all-segments    [1] INT-U16,    -- если этот тип используется для выбора всех сегментов,
                                        -- фактическое содержимое поля «не имеет значения»
                                        -- и должно равняться нулю.
        segm-id-list    [2] SegmIdList, -- Использование этого списка требует, чтобы менеджер уже
                                        -- знал атрибуты Instance-Number
                                        -- сегментов PM-segment, например, из предыдущего
                                        -- вызова метода Get-Segment-Info
                                        -- или Get-Segment-Id-List.
        abs-time-range  [3] AbsTimeRange,
                                        -- Поддержка abs-time-range не обязательна и указывается в
                                        -- атрибуте PM-Store-Capab.
        bo-time-range   [4] BOTimeRange
                                        -- Поддержка bo-time-range не обязательна и указывается в
                                        -- атрибуте PM-Store-Capab.
}
--
-- SegmIdList выбирает сегменты PM-segment по идентификатору.
--
SegmIdList ::= SEQUENCE OF InstNumber
--
-- AbsTimeRange позволяет отбирать сегменты PM-segment по периоду времени.
AbsTimeRange ::= SEQUENCE {
        from-time       AbsoluteTime,
        to-time         AbsoluteTime
}
--
-- BOTimeRange позволяет отбирать сегменты PM-segment по периоду времени, указанному в форме базового
-- времени со смещением.
--
BOTimeRange ::= SEQUENCE {
        from-time       BaseOffsetTime,
        to-time         BaseOffsetTime
}
--
-- SegmentInfoList возвращает атрибуты объектов (кроме Fixed-Segment-Data) всех
-- выбранных экземпляров объектов PM-segment в ответ на метод Get-Segment-Info или Get-Segment-Id-List
-- объекта PM-store.
-- Это требуется менеджеру для извлечения динамической информации о сегментах.
--
SegmentInfoList ::= SEQUENCE OF SegmentInfo
SegmentInfo ::= SEQUENCE {
        seg-inst-no     InstNumber,
        seg-info        AttributeList
}
```

А.5 Типы данных, связанные с сообщениями

```
ObservationScan ::= SEQUENCE {
        obj-handle          HANDLE,
        attributes          AttributeList
}
```

А.6 Прочие типы данных

```
--
-- TimeProtocolId указывает протоколы времени, поддерживаемые или используемые прибором.
--
TimeProtocolId ::= OID-Type        -- Из раздела номенклатуры nom-part-infrastruct
```

А.7 Кадр протокола персонального медицинского прибора

Следующий тип данных представляет собой кадр сообщения верхнего уровня протокола персонального медицинского прибора. Данные блока Apdu (инкапсулируются в PrstApdu) интерпретируются согласно требованиям настоящего стандарта как результат взаимодействия при выполнении процедуры ассоциации (см. 8.7.3.1).

К структурам из А.7 всегда должны применяться правила кодирования MDER.

```
ApduType ::= CHOICE {
        aarq    [57856] AarqApdu,    -- Запрос ассоциации [0xE200]
        aare    [58112] AareApdu,    -- Ответ на запрос ассоциации [0xE300]
        rlrq    [58368] RlrqApdu,    -- Запрос завершения ассоциации
                                     -- [0xE400]
        rlre    [58624] RlreApdu,    -- Ответ на запрос завершения ассоциации
                                     -- [0xE500]
        abrt    [58880] AbrtApdu,    -- Прекращение ассоциации [0xE600]
        prst    [59136] PrstApdu     -- Представление APDU [0xE700]
}
```

А.8 Определения протокола ассоциации

К структурам из А.8 всегда должны применяться правила кодирования MDER.

```
AarqApdu ::= SEQUENCE {
        -- Параметр assoc-version определяет версию процедуры ассоциации,
        -- использованной агентом. Агент должен задать только один
        -- бит версии. Если менеджер не поддерживает данную версию, он должен
        -- отклонить запрос на ассоциацию с кодом rejected-unsupported-assoc-version.
        assoc-version           AssociationVersion,
        data-proto-list         DataProtoList
}
DataProtoList ::= SEQUENCE OF DataProto
-- Если параметр data-proto-id имеет значение data-proto-id-20601, то data-proto-info должен
-- заполняться с помощью структуры PhdAssociationInformation.
-- Если параметр data-proto-id имеет значение data-proto-id-external, то data-proto-info должен
-- заполняться с помощью структуры ManufSpecAssociationInformation.
-- Если параметр data-proto-id имеет значение data-proto-id-empty, то data-proto-info должен
-- иметь пустое значение (используется только в случае отклонения AareApdu).
DataProto ::= SEQUENCE {
        data-proto-id           DataProtoId,
        data-proto-info         ANY DEFINED BY data-proto-id
}
-- Все другие значения DataProtoId зарезервированы и не должны использоваться.
DataProtoId ::= INT-U16 {
        data-proto-id-empty(0),          -- должно использоваться в AareApdu только при условии, что
                                         -- результатом является отклонение
        data-proto-id-20601(20601),      -- указывает, что протокол обмена соответствует требованиям
                                         -- настоящего стандарта
        data-proto-id-external(65535)    -- Указывает, что указанный изготовителем
                                         -- UUID протокола передачи данных является частью
                                         -- ManufSpecAssociationInformation
}
-- Ответ на запрос ассоциации
AareApdu ::= SEQUENCE {
```

```
        result                          AssociateResult,
        selected-data-proto             DataProto
}
-- Запрос завершения ассоциации
RlrqApdu ::= SEQUENCE {
        reason                          ReleaseRequestReason
-- Ответ на запрос завершения ассоциации
RlreApdu ::= SEQUENCE {
        reason                          ReleaseResponseReason
}
-- Прекращение ассоциации
AbrtApdu ::= SEQUENCE {
        reason                          Abort-reason
}
-- Причина прекращения
-- Все не назначенные значения «Abort-reason» зарезервированы для последующего расширения и
-- не должны использоваться.
Abort-reason ::= INT-U16 {
        undefined(0),
        buffer-overflow(1),
        response-timeout(2),
        configuration-timeout(3)        -- Сообщение о конфигурации не получено своевременно
}
-- Описание использования см. в 8.7.3.2.
-- Все не назначенные значения «AssociateResult» зарезервированы для последующего расширения
-- и не должны использоваться.
AssociateResult ::= INT-U16 {
        accepted(0),
        rejected-permanent(1),
        rejected-transient(2),
        accepted-unknown-config(3),
        rejected-no-common-protocol(4),
        rejected-no-common-parameter(5),
        rejected-unknown(6),
        rejected-unauthorized(7),
        rejected-unsupported-assoc-version(8)
}
-- Все не назначенные значения «ReleaseRequestReason» зарезервированы для последующего
-- расширения и не должны использоваться.
ReleaseRequestReason ::= INT-U16 {
        normal(0),                      -- используется, когда агент или менеджер решает
                                        -- завершить ассоциацию при нормальных условиях
        no-more-configurations(1),      -- используется агентом, когда все возможные конфигурации
                                        -- испробованы и менеджер
                                        -- отклонил их все.
        configuration-changed(2)        -- используется агентом, когда изменения в его конфигурации
                                        -- требуют от агента прекращения ассоциации. За этим
                                        -- может следовать запрос ассоциации вместе с
                                        -- информацией о новой конфигурации.
}
-- Все не назначенные значения «ReleaseResponseReason» зарезервированы для последующего
-- расширения и не должны использоваться.
ReleaseResponseReason ::= INT-U16 {
         normal(0)
}
-- Значения запроса ассоциации DataProto отображены в PhdAssociationInformation.
-- Эта информация используется для объявления и получения версии протокола, профиля и т. д.
PhdAssociationInformation ::= SEQUENCE {
        -- Информация protocolVersion используется для передачи допустимых версий. Если
        -- агент отправляет protocolVersion, он должен установить биты для
        -- каждой поддерживаемой версии. В ответе менеджер должен установить один бит
```

```
        -- для обозначения версии протокола, которая должна использоваться агентом и менеджером.
        -- В случае отсутствия общей версии протокола менеджер должен отклонить запрос
        -- ассоциации и присвоить всем битам ProtocolVersion нулевые значения.
        protocol-version            ProtocolVersion,
        encoding-rules              EncodingRules,
        nomenclature-version        NomenclatureVersion,
        functional-units            FunctionalUnits,
        system-type                 SystemType,
        system-id                   OCTET STRING,
        dev-config-id               ConfigId,
        data-req-mode-capab         DataReqModeCapab,
        option-list                 AttributeList
}
--
-- Специфичная для изготовителя информация об ассоциации для собственного протокола
-- передачи данных
--
ManufSpecAssociationInformation : := SEQUENCE {
        data-proto-id-ext           UuidIdent,
        data-proto-info-ext         ANY DEFINED BY data-proto-id-ext
}
-- Все не назначенные значения битов структуры «AssociationVersion» зарезервированы
-- для последующего расширения и должны равняться нулю.
AssociationVersion ::= BITS-32 {
        assoc-version1(0)           -- Этот бит должен быть установлен, если поддерживается
                                    -- версия 1 протокола ассоциации
}
-- Все не назначенные значения битов структуры «ProtocolVersion» зарезервированы
-- для последующего расширения и должны равняться нулю.
ProtocolVersion ::= BITS-32 {
        protocol-version1(0),       -- Этот бит должен быть установлен, если поддерживается
                                    -- стандарт IEEE 11073-20601-2008
        protocol-version2(1),       -- Этот бит должен быть установлен, если поддерживается
                                    -- стандарт IEEE 11073-20601a-2010.
        protocol-version3(2),       -- Этот бит должен быть установлен, если поддерживается
                                    -- стандарт IEEE 11073-20601-2014.
}
--
-- Агент и менеджер всегда должны всегда поддерживать правила кодирования MDER.
-- Кроме MDER, агент и менеджер могут согласовать другие правила кодирования.
-- Все не назначенные значения битов структуры «EncodingRules» зарезервированы
-- для последующего расширения и должны равняться нулю.
--
EncodingRules ::= BITS-16 {
        mder(0),        -- Этот бит должен быть установлен, если поддерживаются/выбраны MDER
        xer(1),         -- Этот бит должен быть установлен, если поддерживаются/выбраны XER
        per(2)          -- Этот бит должен быть установлен, если поддерживаются/выбраны PER
}
-- Все не назначенные значения битов структуры «NomenclatureVersion» зарезервированы
-- для последующего расширения и должны равняться нулю.
NomenclatureVersion ::= BITS-32 {   -- значения ссылаются на конкретный стандарт,
                                    -- регламентирующий номенклатуру
nom-version1(0)                     -- Этот бит должен быть установлен, если
                                    -- поддерживается версия 1
}
-- Все не назначенные значения битов структуры «FunctionalUnits» зарезервированы
-- для последующего расширения и должны равняться нулю.
FunctionalUnits ::= BITS-32 {
        fun-units-unidirectional(0),    -- Зарезервировано для последующего использования.
        fun-units-havetestcap(1),       -- Этот бит указывает, может ли прибор установить
                                        -- тестовую ассоциацию
```

```
        fun-units-createtestassoc(2)        -- Этот бит используется, чтобы указать намерение
                                            -- сформировать тестовую ассоциацию
}
-- Все не назначенные значения битов структуры «SystemType» зарезервированы
-- для последующего расширения и должны равняться нулю.
SystemType ::= BITS-32 {
        sys-type-manager(0),
        sys-type-agent(8)
}
ConfigId ::= INT-U16 {
        manager-config-response(0),
        standard-config-start(1),
        standard-config-end(16383),
        extended-config-start(16384),
        extended-config-end(32767),
        reserved-start(32768),
        reserved-end(65535)
}
```

А.9 Определения протокола представления данных

К структурам из А.9 всегда должны применяться правила кодирования MDER.

```
--
-- Байтовая строка OCTET STRING содержит блок APDU данных, закодированный в соответствии
-- с абстрактным синтаксисом и синтаксисом передачи, согласованными при создании ассоциации.
-- Если для data-proto-id согласовано значение data-proto-id-20601, то OCTET STRING должна быть
-- закодированной версией DataApdu.
--
PrstApdu ::= OCTET STRING
```

А.10 Определения протокола данных

А.10.1 Общие положения

Блок DataApdu и соответствующие структуры из А.10 должны использовать правила кодирования, согласованные процедурой ассоциации (см. 8.7.3.1). Агент и менеджер должны всегда поддерживать правила кодирования MDER, но могут согласовать другие правила кодирования.

А.10.2 Кадр протокола данных

```
--
-- Комбинированный тип примитива удаленных операций и тип операции
-- В сообщениях вызова удаленных операций (roiv-*) параметр invoke-id представляет собой
-- непрозрачный идентификатор, позволяющий отправителю сообщения идентифицировать
-- ассоциированное ответное сообщение (при наличии).
-- Отправитель сообщения roiv-* должен выбрать значение invoke-id, позволяющее ему отличать это
-- сообщение от любых других сообщений roiv-*, которые не устарели. Сообщения устаревают при
-- получении ответа (rors-*, roer или rorj) или превышении значения времени ожидания
-- подтверждения.
-- При возвращении ответного сообщения (rors-*, roer или rorj) значение параметра invoke-id
-- из сообщения вызова должно копироваться в invoke-id ответа. Благодаря этому инициатор
-- может сопоставить ответы с ожидающими запросами. Поскольку идентификатор не прозрачен,
-- получатель не может делать никакие предположения относительно invoke-id. В частности, нельзя
-- предполагать, что идентификатор уникален для любой последовательности номеров или периода времени.
--
DataApdu ::= SEQUENCE {
        invoke-id                           InvokeIDType,
        message                             CHOICE {
        roiv-cmip-event-report              [256] EventReportArgumentSimple, -- [0x0100]
        roiv-cmip-confirmed-event-report    [257] EventReportArgumentSimple, -- [0x0101]
        roiv-cmip-get                       [259] GetArgumentSimple, -- [0x0103]
        roiv-cmip-set                       [260] SetArgumentSimple, -- [0x0104]
        roiv-cmip-confirmed-set             [261] SetArgumentSimple, -- [0x0105]
        roiv-cmip-action                    [262] ActionArgumentSimple, -- [0x0106]
        roiv-cmip-confirmed-action          [263] ActionArgumentSimple, -- [0x0107]
```

```
                rors-cmip-confirmed-event-report        [513] EventReportResultSimple, -- [0x0201]
                rors-cmip-get                           [515] GetResultSimple, -- [0x0203]
                rors-cmip-confirmed-set                 [517] SetResultSimple, -- [0x0205]
                rors-cmip-confirmed-action              [519] ActionResultSimple, -- [0x0207]
                roer                                    [768] ErrorResult, -- [0x0300]
                rorj                                    [1024] RejectResult – [0x0400]
                }
}
-- Отправителю необходимо ограничить количество сообщений, одновременно ожидающих ответа.
-- В действительности получатель может оказаться неспособным обработать одновременно
-- более одного сообщения.
InvokeIDType ::= INT-U16
-- Если в результате выполнения действия, вызванного DataApdu (roiv-*), произошла ошибка,
-- получатель отправляет обратно ErrorResult. Параметр invokeID используется для определения
-- вызова, ставшего причиной возникновения ошибки. Параметру error-value присваивается код
-- ошибки из списка RoerErrorValue (см. ниже). Параметр содержит дополнительную информацию,
-- если это допускает код ошибки. Использование значения параметра указывается в комментариях
-- для RoerErrorValue.
ErrorResult ::= SEQUENCE {
        error-value             RoerErrorValue,
        parameter               ANY DEFINED BY error-value
}
-- Все не назначенные значения «RoerErrorValue» зарезервированы для последующего расширения и
-- не должны использоваться.
-- Необходимо отметить, что стандарт ISO/IEEE 11073-20101:2004 [B21] определяет ряд значений
-- RoerErrorValue, которые не упомянуты в настоящем стандарте. Для обеспечения согласованности
нумерация RoerErrorValue игнорирует все значения, которые уже определены
-- в стандарте ISO/IEEE 11073-20101:2004.
RoerErrorValue ::= INT-U16 {
        -- no-such-object-instance возвращается при ссылке на недопустимый дескриптор или
        -- попытке доступа к любому объекту, кроме системы медицинского прибора, до
        -- согласования конфигурации, например, агент и менеджер не находятся
        -- в состоянии «Выполнение».
        no-such-object-instance(1),
        -- no-such-action возвращается для недопустимой команды действия.
        no-such-action(9),
        -- invalid-object-instance возвращается в тех случаях, когда объект существует, но команда
        -- не допустима для объекта такого типа (например, служба Get применяется к
        -- любому объекту, кроме MDS или PM-store).
        invalid-object-instance(17),
        -- protocol-violation возвращается при наличии нарушения протокола (например, блок APDU
        -- превысил максимальный размер)
        protocol-violation(23),
        -- not-allowed-by-object возвращается при попытке применения действия к объекту,
        -- не позволяющему выполнение этого действия
        -- Более высокий уровень может сообщить причину прекращения действия в форме OID-Type
        -- в поле параметра, используя код возврата, заимствованный из раздела кодов возврата
        not-allowed-by-object(24),
        -- action-timed-out возвращается при прекращении действия или превышении
        -- предельного времени ожидания завершения действия.
        -- Более высокий уровень может сообщить причину прекращения действия в форме OID-Type
        -- в поле параметра, используя код возврата, заимствованный из раздела кодов возврата
        action-timed-out(25),
        -- action-aborted возвращается в случае прерывания действия вследствие
        -- существования определенных причин на более высоких уровнях (например, превышена
        -- емкость системы хранения).
        -- Более высокий уровень может сообщить причину прекращения действия в форме OID-Type
        -- в поле параметра, используя код возврата, заимствованный из раздела кодов возврата
        action-aborted(26),
        -- unsupported-choice возвращается при попытке применения действия к объекту,
        -- когда объект не поддерживает вариант выбора, используемый действием.
```

```
        -- Более высокий уровень может сообщить причину прекращения действия в форме OID-Type
        -- в поле параметра, используя код возврата, заимствованный из раздела кодов возврата
        unsupported-choice(27)
        -- invalid-choice возвращается при попытке применения действия к объекту,
        -- когда вариант выбора, используемый этим действием, не определен в настоящем стандарте и не
           понимается или не предоставляется агентом. Более высокий уровень может сообщить причину
           прекращения действия в форме OID-Type в поле параметра, используя код возврата, заимствованный
           из раздела кодов возврата
        invalid-choice(28)
}
-- Если действие, вызванное DataApdu (roiv-*), требует от получателя отклонения
-- операции перед попыткой ее выполнения, получатель должен послать
-- обратно RejectResult. Параметр invokeID используется для определения вызова, ставшего
-- причиной отклонения. Поле проблемы должно заполняться значением из списка RorjProblem
-- (см. ниже).
RejectResult ::= SEQUENCE {
                problem         RorjProblem
}
-- Все не назначенные значения «RorjProblem» зарезервированы для последующего расширения и
-- не должны использоваться.
RorjProblem ::= INT-U16 {
        -- unrecognized-apdu возвращается в тех случаях, когда DataApdu не распознан.
        unrecognized-apdu(0),
        -- badly-structured-apdu возвращается в тех случаях, когда получатель не может
        --  распознать DataApdu из-за его структуры (или ее отсутствия)
        -- (например, некорректная длина данных)
        badly-structured-apdu(2),
        -- unrecognized-operation отправляется в тех случаях, когда получатель
        -- не может распознать запрошенную операцию
        unrecognized-operation(101),
        -- resource-limitation отправляется в тех случаях, когда получатель не может обработать
        -- сообщение вследствие ограниченности ресурсов
        resource-limitation(103),
        -- unexpected-error соответствует условиям ошибки, когда невозможно
        -- определить более конкретный код ошибки.
        unexpected-error(303)
}
```

A.10.3 Служба EVENT REPORT

```
-- В отчетах о событиях, определенных в настоящем стандарте, идентификатор obj-handle должен
-- иметь значение 0, соответствующее объекту MDS, или значение дескриптора, представляющего
-- объект Scanner или PM-store.
-- Если агент не поддерживает RelativeTime (указывается с помощью бита mds-time-capab-relative-time
-- в MdsTimeCapState), параметру event-time необходимо присвоить значение 0xFFFFFFFF.
-- Менеджеры должны игнорировать параметр event-time, если агент сообщает, что он
-- не поддерживает RelativeTime.
-- Для типов событий event-type, указанных в таблицах 5, 12, 17 и 19, должна использоваться
-- соответствующая структура event-info. Таким образом, event-info будет иметь один из
-- следующих типов: ConfigReport, ScanReportInfoFixed, ScanReportInfoVar, ScanReportInfoMPFixed,
-- ScanReportInfoMPVar, ScanReportInfoGrouped, ScanReportInfoMPGrouped
-- или SegmentDataEvent.
EventReportArgumentSimple ::= SEQUENCE {
                obj-handle      HANDLE,
                event-time      RelativeTime,
                event-type      OID-Type,               -- Из раздела nom-part-ob
                                                        -- Подраздел NOTI (MDC_NOTI_*)
                event-info      ANY DEFINED BY event-type
}
-- Для отчетов о событиях, определенных в настоящем стандарте, идентификатор obj-handle должен
-- иметь значение 0, соответствующее объекту MDS, или значение дескриптора, представляющего
-- объект Scanner или PM-store.
```

```
-- Тип события (event-type) результатов должен быть копией event-type из вызова.
-- Для типов событий event-type, указанных в таблицах 5, 12, 17 и 19, должна использоваться
-- соответствующая структура event-reply-info. Таким образом, event-reply-info
-- будет пустой или будет иметь тип ConfigReportRsp либо SegmentDataResult.
EventReportResultSimple ::= SEQUENCE {
                    obj-handle      HANDLE,
                    currentTime     RelativeTime,
                    event-type      OID-Type,           -- Выбирается из раздела nom-part-obj
                                                        -- Подраздел NOTI (MDC_NOTI_*)
                    event-reply-info ANY DEFINED BY event-type
}
```

A.10.4 Служба GET

```
-- В запросах GET, определенных в настоящем стандарте, идентификатор obj-handle должен иметь
-- значение 0, соответствующее объекту MDS, или значение дескриптора, представляющего
-- объект PM-store.
-- Параметр attribute-id-list должен оставаться пустым, если необходимо запросить все атрибуты
-- объекта MDS или объекта PM-store.
-- Иначе конкретные атрибуты объекта могут запрашиваться путем перечисления необходимых
-- идентификаторов, указанных в таблице 3 или 10.
GetArgumentSimple ::= SEQUENCE {
                    obj-handle          HANDLE,
                    attribute-id-list   AttributeIdList
}
-- Идентификатор obj-handle для ответов на запросы GET, определенные в настоящем стандарте,
-- должен совпадать с переданным в запросе.
-- Параметр attribute-list содержит все запрошенные атрибуты, используя переменный формат.
-- Если запрошенный идентификатор атрибута отсутствует в объекте MDS, он не должен
-- возвращаться в attribute-list.
GetResultSimple ::= SEQUENCE {
                obj-handle          HANDLE,
                attribute-list      AttributeList
}
TypeVerList ::= SEQUENCE OF TypeVer
-- Поскольку тип должен заимствоваться из подраздела DEVspec раздела коммуникационной
-- инфраструктуры nom-part-infrastruct, описанного в ISO/IEEE 11073-10101 [B16], то вместо
-- TYPE используется простой OID-Type.
-- Отдельные специализации IEEE 11073-104zz определяют, какая спецификация
-- классифицируется как версия 1, 2, ... и так далее. Следовательно, версия 3 может
-- соответствовать версии спецификации 1.5.
TypeVer ::= SEQUENCE {
                    type            OID-Type,
                    version         INT-U16
}
```

A.10.5 Служба SET

```
-- Для служб SET, определенных в настоящем стандарте, идентификатор obj-handle должен иметь
-- значение, указывающее на объект Scanner.
SetArgumentSimple ::= SEQUENCE {
                    obj-handle          HANDLE,
                    modification-list   ModificationList
}
ModificationList ::= SEQUENCE OF AttributeModEntry
AttributeModEntry ::= SEQUENCE {
                    modify-operator     ModifyOperator,
                    attribute           AVA-Type
}
-- Все не назначенные значения структуры «ModifyOperator» зарезервированы для последующего
-- расширения и не должны использоваться.
ModifyOperator ::= INT-U16 {
                    replace(0),
```

```
                addValues(1),           -- используется для изменения значений, содержащихся в
                                        -- списочных типах данных
                removeValues(2),        -- используется для изменения значений, содержащихся в
                                        -- списочных типах данных
                setToDefault(3)
}
--
-- Идентификатору obj-handle должно присваиваться значение, полученное в SetArgumentSimple.
-- Список attribute-list должен содержать все идентификаторы attribute-id измененных атрибутов и
-- возвращать новое значение атрибута. Обычно это значение берется из
-- команды Set, однако возможны ситуации, когда вследствие округления или
-- ошибки возвращенное значение может отличаться от запрошенного значения.
SetResultSimple ::= SEQUENCE {
        obj-handle          HANDLE,
        attribute-list      AttributeList
}
```

A.10.6 Служба ACTION

```
-- Для запросов действий, определенных в настоящем стандарте, идентификатор obj-handle должен
-- иметь значение 0, соответствующее объекту MDS, или значение дескриптора, представляющего
-- объект PM-store.
-- Для значений параметра action-type, указанных в таблицах 4 и 11, должны использоваться
-- соответствующие структуры action-info-args. Таким образом, action-info-args будет иметь один из
-- следующих типов: DataRequest, SetTimeInvoke, SetBOTimeInvoke, SegmSelection
-- или TrigSegmDataXferReq.
ActionArgumentSimple ::= SEQUENCE {
        obj-handle          HANDLE,
        action-type         OID-Type,           -- Из раздела nom-part-obj
                                                -- Подраздел ACT (MDC_ACT_*)
        action-info-args    ANY DEFINED BY action-type
}
-- Идентификатор obj-handle в ответах на запросы действий, определенных в настоящем стандарте,
-- должен совпадать с идентификатором из соответствующего запроса.
-- Тип действия action-type должен копироваться из типа действия сообщения вызова.
-- Для значений параметра action-type, указанных в таблицах 4 и 11, должна использоваться
-- результирующая структура action-info-args. Таким образом, action-info-args будет пустой или
-- будет иметь один из следующих типов: DataResponse, SegmentInfoList или TrigSegmDataXferRsp.
ActionResultSimple ::= SEQUENCE {
        obj-handle          HANDLE,
        action-type         OID-Type,           -- Из раздела nom-part-obj
                                                -- Подраздел ACT (MDC_ACT_*)
        action-info-args    ANY DEFINED BY action-type
}
```

A.11 Типы данных новых атрибутов и служб объектов

A.11.1 Общие типы данных

```
AttrValMap ::= SEQUENCE OF AttrValMapEntry
AttrValMapEntry ::= SEQUENCE {
        attribute-id        OID-Type,           -- Заимствуется из раздела nom-part-obj
        attribute-len       INT-U16
}
```

A.11.2 Типы данных, связанные с MDS

```
UuidIdent ::= OCTET STRING(SIZE(16))
-- Параметр time-sync-accuracy позволяет агенту сообщить, насколько точно синхронизированы его
-- внутренние часы с настройками мастера синхронизации часов (если используется синхронизация
-- времени). Данное значение представляет собой накопленную разность между показаниями
-- внутренних часов и внешнего эталона с момента последней синхронизации.
MdsTimeInfo ::= SEQUENCE {
        mds-time-cap-state          MdsTimeCapState,
        time-sync-protocol          TimeProtocolId,     -- номенклатурный код из
                                                        -- раздела nom-part-infrastruct
```

```
        time-sync-accuracy              RelativeTime,   -- 0xFFFFFFFF, если не известно
                                                        -- 0, если точнее, чем 1/8 мс.
        time-resolution-abs-time        INT-U16,        -- если
                                                        -- бит mds-time-capab-real-time-clock
                                                        -- установлен, это значение
                                                        -- указывает точность часов
                                                        -- абсолютного времени агента.
                                                        -- 0, если не известно, иначе
                                                        -- количество 1/100 с,
                                                        -- прошедшее с каждого отсчета
                                                        -- часов. Например, если
                                                        -- агент имеет часы с точностью
                                                        -- отсчета 1 с, данное значение
                                                        -- будет равно 100.
                                                        -- Если бит mds-time-capab-bo-time
                                                        -- установлен, это значение
                                                        -- указывает точность часов
                                                        -- базового времени агента.
                                                        -- 0, если не известно, иначе
                                                        -- количество 1/65536 с,
                                                        -- прошедшее с каждого отсчета
                                                        -- часов. Значение 0xFFFF
                                                        -- зарезервировано, чтобы указать
                                                        -- интервал 1 с.
        time-resolution-rel-time        INT-U16,        -- Точность часов
                                                        -- относительного времени агента.
                                                        -- 0, если неизвестно,
                                                        -- иначе количество
                                                        -- 125 мкс, прошедшее
                                                        -- с каждого отсчета часов. Например,
                                                        -- если агент имеет часы
                                                        -- с точностью 1 с,
                                                        -- это значение будет равно 8000.
        time-resolution-high-res-time   INT-U32         -- Точность часов
                                                        -- времени высокой точности.
                                                        -- 0, если не известно, иначе
                                                        -- количество микросекунд,
                                                        -- прошедшее с каждого отсчета
                                                        -- часов. Например, если
                                                        -- агент имеет часы с точностью
                                                        -- отсчета 1 с, данное значение
                                                        -- будет равно 1 000 000.
}
-- Должен указываться только mds-time-capab-real-time-clock или mds-time-capab-bo-time.
-- Должен указываться только mds-time-capab-sync-abs-time или mds-time-capab-sync-bo-time.
-- Должен указываться только mds-time-state-abs-time-synced или mds-time-state-bo-time-synced.
-- Все не назначенные значения битов структуры «MdsTimeCapState» зарезервированы для
-- последующего расширения и должны равняться нулю.
MdsTimeCapState ::= BITS-16 {
        mds-time-capab-real-time-clock(0),          -- Прибор поддерживает внутренние
                                                    -- часы реального времени,
                                                    -- учитывающие абсолютное время
        mds-time-capab-set-clock(1),                -- Прибор поддерживает действие
                                                    -- Set-Time или Set-Base-OffsetTime.
        mds-time-capab-relative-time(2),            -- Прибор поддерживает
                                                    -- относительное время
        mds-time-capab-high-res-relative-time(3),   -- Прибор поддерживает
                                                    -- относительное время
                                                    -- высокой точности
        mds-time-capab-sync-abs-time(4),            -- Прибор синхронизирует
                                                    -- абсолютное время
```

```
                mds-time-capab-sync-rel-time(5),                -- Прибор синхронизирует
                                                                -- относительное время
                mds-time-capab-sync-hi-res-relative-time(6),    -- Прибор синхронизирует
                                                                -- относительное время
                                                                -- высокой точности
                mds-time-capab-bo-time(7),                      -- Прибор поддерживает базовое
                                                                -- время со смещением
                mds-time-state-abs-time-synced(8),              -- синхронизируется абсолютное
                                                                -- время
                mds-time-state-rel-time-synced(9),              -- синхронизируется относительное
                                                                -- время
                mds-time-state-hi-res-relative-time-synced(10), -- синхронизируется относительное
                                                                -- время высокой точности
                mds-time-mgr-set-time(11),                      -- менеджер должен установить
                                                                -- время
                mds-time-capab-sync-bo-time(12),                -- прибор синхронизирует базовое
                                                                -- время со смещением
                mds-time-state-bo-time-synced(13),              -- синхронизируется базовое время.
                mds-time-state-bo-time-UTC-aligned(14)          -- базовое время согласуется с UTC.
                mds-time-dst-rules-enabled(15)                  -- прибор поддерживает и применяет
                                                                -- правила перехода на летнее время
}
-- **************
-- Список различных элементов, связанных с соответствием нормативным документам и
-- сертификации, которым следует агент.
-- **************
RegCertDataList ::= SEQUENCE OF RegCertData
RegCertData ::= SEQUENCE {
                auth-body-and-struc-type        AuthBodyAndStrucType,
                auth-body-data                  ANY DEFINED BY auth-body-and-struc-type
}
AuthBodyAndStrucType ::= SEQUENCE {
                auth-body                       AuthBody,
                auth-body-struc-type            AuthBodyStrucType
}
--
-- Все не назначенные значения структуры «AuthBody» зарезервированы для последующего
-- расширения и не должны использоваться.
AuthBody ::= INT-U8 {
                auth-body-empty(0),
                auth-body-ieee-11073(1),
                auth-body-continua(2),
                auth-body-experimental(254),
                auth-body-reserved(255)
}
--
-- Некоторые другие возможные/ожидаемые уполномоченные органы:
-- auth-body-eu(),
-- auth-body-ieee(),
-- auth-body-iso(),
-- auth-body-us-fda(),
-- конкретные значения будут назначены, когда определенный уполномоченный орган
-- присвоит свой первый тип AuthBodyStrucType
-- структуре auth-body-data.
-- AuthBodyStrucType контролируется и назначается уполномоченным органом
AuthBodyStrucType ::= INT-U8
```

А.11.3 Типы данных, связанные с классом Metric

```
--
-- SupplementalTypeList предоставляет эффективную возможность перечисления дополнительной
-- информации об объекте.
```

```
-- Данная категория может содержать такую информацию, как местоположение датчика или реакция
-- объекта.
--
SupplementalTypeList ::= SEQUENCE OF TYPE
--
-- Атрибут MetricSpecSmall представляет собой сокращенный вариант атрибута MetricSpec,
-- определенного в ISO/IEEE 11073-10201:2004 [B17]. Он определяет доступность, периодичность и
-- категорию измерения.
-- Группа битов 0—5 используется преимущественно для информационных целей; они должны
-- устанавливаться в тех случаях, когда условие выполнено, но менеджер не может предположить,
-- что поведение будет наблюдаться, если они установлены.
-- Все не назначенные значения структуры «MetricSpecSmall» зарезервированы для последующего
-- расширения и должны равняться нулю.
--
MetricSpecSmall ::= BITS-16 {
        mss-avail-intermittent(0),      -- Значение доступно только время от времени
        mss-avail-stored-data(1),       -- Агент может хранить и отправлять несколько
                                        -- исторических значений (например, возможно
                                        -- хранение до 25 значений веса).
        mss-upd-aperiodic(2),           -- Значение передается только апериодически
                                        -- (например, после изменения).
        mss-msmt-aperiodic(3),          -- Измерение выполняется апериодически.
        mss-msmt-phys-ev-id(4),         -- Измерение является лишь физиологическим
                                        -- триггером (например, маркировка
                                        -- обнаружения сокращения сердца)
        mss-msmt-btb-metric(5),         -- Измерение от удара сердца до удара или вдоха
                                        -- до выдоха
        mss-acc-manager-initiated(8),   -- Доступ к значению объекта возможен с помощью
                                        -- инициированной менеджером передачи
                                        -- результатов измерений
        mss-acc-agent-initiated(9),     -- Значение объекта обновляется с помощью
                                        -- инициированной агентом передачи
                                        -- результатов измерений
                                        -- Примечания к использованию битов mss-cat-*.
                                        -- Если измерения осуществляются автоматически,
                                        -- то биты mss-cat-setting и mss-cat-calculation
                                        -- не устанавливаются. Результат соответствует
                                        -- нормальному, стандартно измеренному
                                        -- значению. Отсюда вытекает, что в результатах
                                        -- автоматических измерений, предоставленных
                                        -- агентом, ни один из битов mss-cat-*
                                        -- не установлен (по умолчанию).
        mss-cat-manual(12),             -- Если этот бит установлен, результат собирается
                                        -- вручную
                                        -- (например, человек вводит значение вручную).
                                        -- Если этот бит не установлен, результат собирается
                                        -- автоматически (например, прибор измеряет
                                        -- значение).
        mss-cat-setting(13),            -- Если этот бит установлен, результат представляет
                                        -- параметры устройства. Его значение собирается
                                        -- вручную или автоматически, как указано
                                        -- в бите mss-cat-manual.
        mss-cat-calculation(14)         -- Если этот бит установлен, результат представляет
                                        -- вычисленное значение. Оно может рассчитываться
                                        -- вручную или автоматически, как указано
                                        -- в бите mss-cat-manual.
                                        -- Вычисленные значения определяются
                                        -- на основе автоматически собранных результатов
                                        -- измерений и/или вручную введенных значений
}
-- Данный атрибут частично унаследован из ISO/IEEE 11073-10201:2004 [B17] и дополняется
-- значением ms-struct::ms-struct-compound-fix. В соответствии со стандартом IEEE 11073-20601-2014
```

```
-- ms-struct-compound и ms-struct-compound-fix должны использоваться только для объектов Numeric.
-- Для обеспечения использования составных структур в объектах RT-SA и Enumeration
-- необходимо наличие дополнительных структур.
-- При использовании составных структур агент должен отправить только наблюдаемые
-- значения ms-comp-no.
--
MetricStructureSmall ::= SEQUENCE {
        ms-struct INT-U8 {
                ms-struct-simple(0),
                ms-struct-compound(1),          -- несколько наблюдаемых значений,
                                                -- тот же динамический контекст
                ms-struct-reserved(2),          -- для ISO/IEEE 11073-10201:2004
                ms-struct-compound-fix(3)       -- [B17] похоже на compound(1),
                                                -- однако размер массива
                                                -- наблюдаемого составного значения
                                                -- не должен быть динамическим
                                                -- во время ассоциации
        },
        ms-comp-no INT-U8               -- Максимальное количество компонентов/элементов в
                                        -- составном наблюдаемом значении (0, если ms-struct
                                        -- задано ms-struct-simple).
}
-- Данный атрибут определяет список идентификаторов MetricId.
--
MetricIdList ::= SEQUENCE OF OID-Type
--
-- EnumPrintableString — тип данных, используемый для представления перечисляемых наблюдаемых
-- значений в виде печатаемых строк ASCII.
--
EnumPrintableString ::= OCTET STRING            -- размер строки должен быть четным
PersonId ::= INT-U16 {
        unknown-person-id(65535)        -- 0xFFFF
}
```

A.11.4 Типы данных, связанные с классом Scanner

```
HandleAttrValMap ::= SEQUENCE OF HandleAttrValMapEntry
HandleAttrValMapEntry ::= SEQUENCE {
        obj-handle      HANDLE,
        attr-val-map    AttrValMap
}
HANDLEList ::= SEQUENCE OF HANDLE
```

A.11.5 Службы MDS

```
-- Следующие определения поддерживают указанные выше определения EventReportArgumentSimple
-- и ActionArgumentSimple.
--
-- Типы информации о сканировании используются в качестве типов данных результатов для
-- различных событий семейства MDS-Dynamic-Data-Update* (подробнее см. в 6.3.2.5).
--
-- Определения ScanReport* используются при передаче информации об изменениях
-- значений атрибутов объектов (наборы изменений атрибутов). Существуют два вектора: A) одно
-- лицо или несколько лиц и B) переменный, фиксированный или групповой формат.
-- Сочетание этих векторов позволяет сформулировать шесть определений
-- верхнего уровня: ScanReportInfoVar, ScanReportInfoFixed, ScanReportInfoGrouped,
-- ScanReportInfoMPVar, ScanReportInfoMPFixed и ScanReportInfoMPGrouped.
-- SEQUENCE OF ObservationScan или ObservationScanFixed может содержать несколько
-- экземпляров одного и того же дескриптора, пока существует метка времени, позволяющая
-- различать экземпляры.
-- Во всех случаях значение scan-report-no должно обнуляться во время ассоциации и монотонно
-- возрастать на единицу до момента сброса.
```

```
-----------------------------------------------------------------------------------------------------------
ScanReportInfoVar ::= SEQUENCE {
                data-req-id             DataReqId,
                scan-report-no          INT-U16,        -- счетчик для обнаружения потери отчетов о
                                                        -- сканировании
                obs-scan-var                    SEQUENCE OF ObservationScan
}
-----------------------------------------------------------------------------------------------------------
ScanReportInfoFixed ::= SEQUENCE {
                data-req-id             DataReqId,
                scan-report-no          INT-U16,        -- счетчик для обнаружения потери отчетов о
                                                        -- сканировании
                obs-scan-fixed          SEQUENCE OF ObservationScanFixed
}
ObservationScanFixed ::= SEQUENCE {
                obj-handle              HANDLE,         -- уникальная идентификация объекта
                obs-val-data            OCTET STRING    -- наблюдаемые значения данных,
                                                        -- определяемые obj-handle
}
-----------------------------------------------------------------------------------------------------------
-- obs-scan-grouped определен как SEQUENCE OF, поэтому при эпизодических измерениях можно
-- объединить несколько отчетов в один отчет о сканировании. Периодические отчеты следует
-- формировать таким образом, чтобы не требовалось помещать несколько отчетов
-- в один ScanReport.
ScanReportInfoGrouped ::= SEQUENCE {
                data-req-id             DataReqId,
                scan-report-no          INT-U16,        -- счетчик для обнаружения потери
                                                        -- отчетов о сканировании
                obs-scan-grouped        SEQUENCE OF ObservationScanGrouped
}
ObservationScanGrouped ::= OCTET STRING                 -- Формат определяется HandleAttrValMap
-----------------------------------------------------------------------------------------------------------
ScanReportInfoMPVar ::= SEQUENCE {
                data-req-id             DataReqId,
                scan-report-no          INT-U16,        -- счетчик для обнаружения потери
                                                        -- отчетов о сканировании
                scan-per-var            SEQUENCE OF ScanReportPerVar
}
DataReqId ::= INT-U 16 {
                data-req-id-manager-initiated-min(0),           -- 0x0000
                data-req-id-manager-initiated-max(61439),       -- 0xEFFF
                -- Значения в диапазоне от data-req-id-manager-initiated-min до
                -- data-req-id-manager-initiated-max включительно должны использоваться при
                -- инициируемой менеджером передаче результатов измерений.
                --
                data-req-id-agent-initiated-confirmed(61440)    -- 0xF000
                -- Параметр data-req-id-agent-initiated-confirmed должен использоваться при
                -- инициированной агентом передаче результатов измерений
                -- с помощью подтверждаемого отчета о событиях.
                --
                data-req-id-agent-initiated-unconfirmed(61441)  -- 0xF001
                -- Параметр data-req-id-agent-initiated-unconfirmed должен использоваться при
                -- инициированной агентом передаче результатов измерений
                -- с помощью не подтверждаемого отчета о событиях.
                --
                -- Значения в диапазоне от 0xF002 до 0xFFFF включительно зарезервированы.
}
--
-- Значение, используемое для параметра person-id, определяется поставщиком (например, если
-- агент имеет две кнопки для идентификации двух лиц, то он может использовать
-- идентификаторы 1 и 2 или идентификаторы 35 и 97).
```

```
-- В настоящем стандарте не описывается процесс присвоения идентификатора
-- отдельным лицам.
--
ScanReportPerVar ::= SEQUENCE {
        person-id           PersonId,
        obs-scan-var        SEQUENCE OF ObservationScan
}
-------------------------------------------------------------------------------------------------------------------------------
ScanReportInfoMPFixed ::= SEQUENCE {
        data-req-id         DataReqId,
        scan-report-no      INT-U16,        -- счетчик для обнаружения потери
                                            -- отчетов о сканировании
        scan-per-fixed      SEQUENCE OF ScanReportPerFixed
}
ScanReportPerFixed ::= SEQUENCE {
        person-id           PersonId,
        obs-scan-fixed      SEQUENCE OF ObservationScanFixed
}
-------------------------------------------------------------------------------------------------------------------------------
ScanReportInfoMPGrouped ::= SEQUENCE {
        data-req-id         DataReqId,
        scan-report-no      INT-U16,        -- счетчик для обнаружения потери
                                            -- отчетов о сканировании
        scan-per-grouped    SEQUENCE OF ScanReportPerGrouped
}
ScanReportPerGrouped ::= SEQUENCE {
        person-id           PersonId,
        obs-scan-grouped    ObservationScanGrouped
}
-------------------------------------------------------------------------------------------------------------------------------
-- Определение ConfigReport используется при передаче менеджеру конфигурации агента (см.
-- таблицу 5).
ConfigReport ::= SEQUENCE {
        config-report-id    ConfigId,
        config-obj-list     ConfigObjectList
}
ConfigObjectList ::= SEQUENCE OF ConfigObject
ConfigObject ::= SEQUENCE {
        obj-class           OID-Type,       -- Из раздела nom-part-obj
                                            -- Подраздел MOC/BASE (MDC_MOC_VMD_*)
        obj-handle          HANDLE,
        attributes          AttributeList
}
ConfigReportRsp ::= SEQUENCE {
        config-report-id    ConfigId,
        config-result       ConfigResult
}
-- Все не назначенные значения структуры «ConfigResult» зарезервированы для последующего
-- расширения и не должны использоваться.
ConfigResult ::= INT-U16 {
        accepted-config(0),
        unsupported-config(1),
        standard-config-unknown(2)
}
DataRequest ::= SEQUENCE {
        data-req-id         DataReqId,      -- Позволяет отличать
                                            -- ответы для нескольких запросов
                                            -- данных (если
                                            -- прибор поддерживает несколько
                                            -- одновременных запросов данных).
                                            -- Зеркально отражается в параметре
```

```
                                                    -- data-req-id структур ScanReportInfo*
        data-req-mode           DataReqMode,        -- Определяет режим путем установки
                                                    -- одного или нескольких битов.
        data-req-time           RelativeTime,       -- Указывает разрешенную агенту
                                                    -- продолжительность передачи данных.
                                                    -- Используется только для
                                                    -- data-req-mode-time-period.
        data-req-person-id      INT-U16,            -- Значение 0xFFFF соответствует всем
                                                    -- доступным лицам
        data-req-class          OID-Type,           -- Из раздела nom-part-obj
                                                    -- Подраздел MOC/BASE
                                                    -- (MDC_MOC_VMD_*)
        data-req-obj-handle-list HANDLEList
}
-- Все не назначенные значения битов структуры «DataReqMode» зарезервированы для
-- последующего расширения и должны равняться нулю.
DataReqMode ::= BITS-16 {
        data-req-start-stop(0),             -- запрос начала передачи данных: 1 | запрос
                                            -- остановки передачи данных: 0
        data-req-continuation(1),           -- продолжение запроса данных в режиме
                                            -- периода времени
                                            -- Значение 1 позволяет увеличить время,
                                            -- выделенное для передачи данных. Если задано
                                            -- значение 1, все остальные биты должны
                                            -- игнорироваться и необходимо использовать их
                                            -- значения из команды начала передачи.
        -- должен быть установлен только один из следующих битов data-req-scope-*
        data-req-scope-all(4),
        data-req-scope-class(5),
        data-req-scope-handle(6),
        -- должен быть установлен только один из следующих битов data-req-mode-*.
        data-req-mode-single-rsp(8),        -- ответ включается непосредственно
                                            -- в DataResponse
        data-req-mode-time-period(9),       -- Ограниченный по времени запрос данных с
                                            -- ответами в форме отчетов о событиях. Период
                                            -- времени указывается в поле data-req-time
                                            -- структуры DataRequest.
        data-req-mode-time-no-limit(10),    -- не ограниченный по времени запрос данных с
                                            -- ответами в форме отчетов о событиях
        data-req-person-id(12)
}
DataReqModeCapab ::= SEQUENCE {
        data-req-mode-flags                 DataReqModeFlags,
        data-req-init-agent-count INT-U8,   -- максимальное число параллельных
                                            -- запросов/потоков данных,
                                            -- инициируемых агентом. В настоящее время
                                            -- должно задаваться равным только 0 или 1.
        data-req-init-manager-count INT-U8  -- максимальное количество параллельных
                                            -- запросов данных, инициируемых менеджером
}
-- Все не назначенные значения битов структуры «DataReqModeFlags» зарезервированы для
-- последующего расширения и должны равняться нулю.
DataReqModeFlags ::= BITS-16 {              -- это поле используется в ассоциации, чтобы
                                            -- обозначить возможности запроса данных
        data-req-supp-stop(0),              -- поддерживает остановку текущего запроса
        data-req-supp-scope-all(4),         -- данных, поддерживает запрос всех объектов
        data-req-supp-scope-class(5),       -- поддерживает запрос на основе
                                            -- класса объектов
        data-req-supp-scope-handle(6),      -- поддерживает запрос объектов на основе
                                            -- их дескрипторов
        data-req-supp-mode-single-rsp(8),   -- поддерживает одиночный ответ
```

```
                data-req-supp-mode-time-period(9),      -- поддерживает запрос данных,
                                                        -- ограниченный по времени
                data-req-supp-mode-time-no-limit(10),   -- поддерживает запрос данных, не
                                                        -- ограниченный по времени
                data-req-supp-person-id(11),
                data-req-supp-init-agent(15)            -- агент использует запросы/потоки данных,
                                                        -- инициируемые агентом
}
-- DataResponse возвращается как результат запроса MDS-Data-Request (см. таблицу 4). Однако поля
-- event-type и event-info заполняются с помощью тех же самых параметров, что указаны в событиях
-- объектов MDS. Для ознакомления с допустимыми значениями event-type и соответствующей
-- структурой поля event-info см. таблицу 5; однако в этом случае тип ConfigReport не должен
-- использоваться. Таким образом, поле event-info должно иметь один из следующих типов:
-- ScanReportInfoFixed, ScanReportInfoVar, ScanReportInfoMPFixed или ScanReportInfoMPVar.
DataResponse ::= SEQUENCE {
                rel-time-stamp      RelativeTime,       -- задается равным 0xFFFFFFFF, если
                                                        -- относительное время не поддерживается
                data-req-result     DataReqResult,
                event-type          OID-Type,           -- event-type и event-info только
                                                        -- в случае data-req-mode-single-rsp,
                                                        -- иначе event-type должно быть 0 и
                                                        -- event-info.length = 0.
                                                        -- Из раздела nom-part-obj
                                                        -- Подраздел NOTI (MDC_NOTI_*)
                event-info          ANY DEFINED BY event-type
}
-- Значения из DataReqResult используются в поле data-req-result структуры DataResponse. Оно
-- возвращается в ответ на DataRequest. Если запрос оказался успешным, агент должен вернуть
-- значение data-req-result-no-error. Иначе возвращается одна из определенных ошибок.
-- Все не назначенные значения структуры «DataReqResult» зарезервированы для последующего
-- расширения и не должны использоваться.
DataReqResult ::= INT-U16 {
                data-req-result-no-error(0),
                data-req-result-unspecific-error(1),
                -- Возвращаются следующие коды ошибок, если запрос менеджера содержит
                -- режим DataReqMode, не поддерживаемый агентом.
                data-req-result-no-stop-support(2),
                data-req-result-no-scope-all-support(3),
                data-req-result-no-scope-class-support(4),
                data-req-result-no-scope-handle-support(5),
                data-req-result-no-mode-single-rsp-support(6),
                data-req-result-no-mode-time-period-support(7),
                data-req-result-no-mode-time-no-limit-support(8),
                data-req-result-no-person-id-support(9),
                -- Возвращаются следующие коды ошибок, если запрос менеджера содержит
                -- неизвестные значения в поддерживаемых полях (например, data-req-person-id).
                data-req-result-unknown-person-id(11),
                data-req-result-unknown-class(12),
                data-req-result-unknown-handle(13),
                -- Следующие элементы описывают ситуацию, когда менеджер задает несколько
                -- битов режима или области.
                data-req-result-unsupp-scope(14),           -- заданы не поддерживаемые
                                                            -- биты области
                data-req-result-unsupp-scope(15),           -- заданы не поддерживаемые
                                                            -- биты режима
                data-req-result-init-manager-overflow(16),  -- менеджер попытался установить
                                                            -- более чем
                                                            -- data-req-init-manager-count потоков
                data-req-result-continuation-not-supported(17), -- менеджер попытался продолжить
                                                            -- передачу данных, проводящуюся
                                                            -- не в режиме ограниченного времени
```

```
                data-req-result-invalid-req-id(18)      -- менеджер попытался продолжить
                                                        -- передачу данных для  не
                                                        -- существующего data-req-id.
}
```

A.11.6 Службы Scanner

Применительно к службам Scanner используются определения типов служб MDS, перечисленные в разделе A.11.5, а именно:

ScanReportInfoVar
ScanReportInfoFixed
ScanReportInfoGrouped
ScanReportInfoMPVar
ScanReportInfoMPFixed
ScanReportInfoMPGrouped

A.11.7 Типы данных, связанные с классом Numeric

```
-- Простое числовое наблюдаемое значение представляет собой число с плавающей точкой.
--
SimpleNuObsValue ::= FLOAT-Type
-- Тип списка SimpleNuObsValue
--
SimpleNuObsValueCmp ::= SEQUENCE OF SimpleNuObsValue
-- Во многих случаях базовое наблюдаемое числовое значение может выражаться меньшим
-- значением с плавающей точкой.
--
BasicNuObsValue ::= SFLOAT-Type
-- Тип списка BasicNuObsValue
--
BasicNuObsValueCmp ::= SEQUENCE OF BasicNuObsValue
```

A.11.8 Типы данных, связанные с классами PM-store и PM-segment

```
--
-- Атрибут PM-Store-Capab определяет специфические статические параметры и свойства
-- экземпляра объекта PM-store. По умолчанию значение этого атрибута
-- равно 0 (биты не установлены).
-- Все не назначенные значения битов структуры «PmStoreCapab» зарезервированы для
-- последующего расширения и должны равняться нулю.
--
PmStoreCapab ::=BITS-16 {
                pmsc-var-no-of-segm(0),         -- указывает, что количество сегментов
                                                -- PM-segment в этом объекте PM-store динамично
                                                -- и может меняться
                pmsc-segm-id-list-select(3),    -- сегменты PM-segment в типе данных
                                                -- SegmSelection можно выбрать, указав список
                                                -- идентификаторов сегментов
                pmsc-epi-seg-entries(4),        -- некоторые/все сегменты PM-segment содержат
                                                -- эпизодические/апериодические записи, поэтому
                                                -- они должны иметь явную информацию о
                                                -- метке времени
                pmsc-peri-seg-entries(5),       -- некоторые/все сегменты PM-segment содержат
                                                -- периодически собираемые записи, поэтому
                                                -- сегмент PM-segment или объект PM-store
                                                -- должен поддерживать атрибут Sample-Period
                pmsc-abs-time-select(6),        -- сегмент PM-segment в типе данных
                                                -- SegmSelection можно выбрать,
                                                -- задавая abs-time-range или bo-time-range
                                                -- в зависимости от режима времени,
                                                -- поддерживаемого устройством
                pmsc-clear-segm-by-list-sup(7), -- поддерживается очистка списка сегментов
                pmsc-clear-segm-by-time-sup(8), -- очистка сегментов с помощью abs-time-range
                                                -- или bo-time-range возможна в зависимости
                                                -- от режима времени, поддерживаемого
                                                -- устройством
```

```
                pmsc-clear-segm-remove(9),      -- если этот бит установлен, агент полностью
                                                -- удалит указанный экземпляр сегмента
                                                -- PM-segment в рамках метода Clear-Segment.
                                                -- Если этот бит не установлен, будут удалены все
                                                -- записи из указанного сегмента PM-segment
                pmsc-clear-segm-all-sup(10),    -- поддерживается очистка всех сегментов
                pmsc-multi-person(12),          -- объект PM-store позволяет сегментам
                                                -- PM-segment хранить данные нескольких лиц
                pmsc-get-segm-info-sup(13),     -- поддерживается метод Get-Segment-Info.
                pmsc-get-segm-id-list-sup(14),  -- поддерживается метод Get-Segment-Id-List.
}
--
-- Все записи в сегменте должны соответствовать формату, определяемому этим атрибутом.
-- Во-первых, необязательный заголовок должен соответствовать описанию в segm-entry-header. Это
-- позволяет перед каждой записью в сегменте иметь необязательный заголовок (например, для
-- информации о метке времени), применимый ко всем элементам записи.
-- Во-вторых, элементы должны соответствовать формату и порядку, описанным
-- в segm-entry-elem-list.
-- Обычно элемент представляет измерение. Для каждого элемента хранимые данные определяются
-- в форме карты значений атрибутов, тем же способом, что и объекты Metric.
--
PmSegmentEntryMap ::= SEQUENCE {
                segm-entry-header       SegmEntryHeader,    -- определяет необязательные
                                                            -- элементы перед каждой записью
                                                            -- (SegmentEntryHeader)
                segm-entry-elem-list    SegmEntryElemList
}
--
-- Следующая битовая строка определяет необязательные элементы данных перед каждой записью
-- сегмента.
-- Определяется несколько элементов данных. В этом случае элемент данных с меньшим номером
-- бита должен располагаться перед элементами, обладающими более высокими номерами битов.
-- Заголовок позволяет определить элементы данных, общие для всех элементов записи. Если все
-- биты равны нулю, то отчет о событии записи сегмента должен начинаться с данных из первого
-- элемента.
-- Все не назначенные значения битов структуры «SegmEntryHeader» зарезервированы для
-- последующего расширения и должны равняться нулю.
-- Если какой-либо из битов установлен помимо ожидаемых битов (например, в более новую версию
-- добавлен новый бит), не следует извлекать данные, поскольку нельзя вычислить отклонение от
-- первого элемента данных
--
SegmEntryHeader ::= BITS-16 {
                seg-elem-hdr-absolute-time(0),      -- перед записью указывается абсолютное время
                                                    -- (тип данных AbsoluteTime)
                seg-elem-hdr-relative-time(1),      -- перед записью указывается
                                                    -- относительное время
                                                    -- (тип данных RelativeTime)
                seg-elem-hdr-hires-relative-time(2), -- перед записью указывается
                                                    -- относительное время высокой точности
                                                    -- (тип данных HighResRelativeTime)
                seg-elem-hdr-bo-time(3)             -- перед записью указывается
                                                    -- базовое время со смещением
                                                    -- (тип данных BaseOffsetTime)
                                                    -- варианты (0) и (3) взаимно исключают друг друга
}
SegmEntryElemList ::= SEQUENCE OF SegmEntryElem
--
-- Значение SegmEntryElem должно ссылаться на экземпляр объекта Metric в конфигурации агента
-- с помощью его значения дескриптора. Такой объект ссылки должен существовать в
-- конфигурации агента, при этом metric-type и class-id должны равняться значениям соответствующих
-- атрибутов ссылочного объекта Metric.
```

```
--
SegmEntryElem ::= SEQUENCE {
        class-id        OID-Type,       -- содержит номенклатурный код из
                                        -- объектно-ориентированного раздела nom-part-obj,
                                        -- определяющего класс объекта
                                        -- (например, Numeric)
        metric-type     TYPE,           -- конкретный статичный тип хранящегося элемента
        handle          HANDLE,         -- дескриптор объекта ссылки
        attr-val-map    AttrValMap      -- карта значений атрибутов,
                                        -- описывающая хранящиеся данные
}
--
-- Запрос начала передачи указанного сегмента
--
TrigSegmDataXferReq ::= SEQUENCE {
        seg-inst-no     InstNumber
}
TrigSegmDataXferRsp ::= SEQUENCE {
        seg-inst-no     InstNumber,
        trig-segm-xfer-rsp  TrigSegmXferRsp
}
-- Все не назначенные значения битов структуры «TrigSegmXferRsp» зарезервированы для
-- последующего расширения и не должны использоваться.
TrigSegmXferRsp ::= INT-U16 {
        tsxr-successful(0),             -- агент начнет передачу сегмента
        tsxr-fail-no-such-segment(1),   -- идентификатор сегмента не обнаружен
        tsxr-fail-clear-in-process(2),  -- идет очистка среды хранения данных. Доступ
                                        -- запрещен.
        tsxr-fail-segm-empty(3),        -- запрошенный сегмент пуст
        tsxr-fail-not-otherwise-specified(512)
}
--
-- Примечания:
--  - агент должен передать все записи сегментов в порядке «первый пришел — первый ушел» (FIFO).
SegmentDataEvent ::= SEQUENCE {
        segm-data-event-descr   SegmDataEventDescr,
        segm-data-event-entries OCTET STRING        -- содержит указанные записи
                                                    -- сегмента в непрозрачной
                                                    -- структуре данных.
                                                    -- Данное поле может содержать
                                                    -- только полные записи.
}
SegmentDataResult ::= SEQUENCE {
        segm-data-event-descr   SegmDataEventDescr
}
--
-- Дескриптор события данных сегмента определяет, какие записи данных сегмента передаются в
-- сообщении о событии.
--
SegmDataEventDescr ::= SEQUENCE {
        segm-instance           InstNumber,     -- номер экземпляра передаваемого сегмента
        segm-evt-entry-index    INT-U32,        -- индекс массива первой записи для этого
        события
        segm-evt-entry-count    INT-U32,        -- число записей для этого события
        segm-evt-status         SegmEvtStatus
}
-- Все не назначенные значения битов структуры «SegmEvtStatus» зарезервированы для
-- последующего расширения и должны равняться нулю.
SegmEvtStatus ::= BITS-16 {
        sevtsta-first-entry(0),     -- данное событие содержит первую запись сегмента.
        sevtsta-last-entry(1),      -- данное событие содержит последнюю запись
```

```
                                        -- сегмента (одновременно можно задать первый и
                                        -- последний биты, если все записи укладываются
                                        -- в одно событие)
                sevtsta-agent-abort(4),     -- передача прекращена агентом (менеджер должен
                                        -- ответить тем же статусом).
                sevtsta-manager-confirm(8), -- включить в ответ, если сегмент получен правильно
                                        -- (если не включен в ответ, то агент должен
                                        -- остановить передачу сегмента и послать ответ с
                                        -- кодом ошибки (roer), используя параметр
                                        -- protocol-violation)
                sevtsta-manager-abort(12)   -- передается в ответе менеджера (агент должен
                                        -- остановить отправку сообщений)
}
SegmentStatistics ::= SEQUENCE OF SegmentStatisticEntry
SegmentStatisticEntry ::= SEQUENCE {
                segm-stat-type      SegmStatType,
                segm-stat-entry     OCTET STRING    -- Этот атрибут содержит одну запись
                                                    -- сегмента в формате, определяемом
                                                    -- PmSegmentEntryMap.
}
-- Все не назначенные значения структуры «SegmStatType» зарезервированы для последующего
-- расширения и не должны использоваться.
-- Значения в диапазоне от 0xF000 до 0xFFFF зарезервированы для расширений, определяемых
-- изготовителями.
SegmStatType ::= INT-U16 {
                segm-stat-type-undefined(0),
                segm-stat-type-minimum(1),
                segm-stat-type-maximum(2),
                segm-stat-type-average(3)
}
```

Приложение В
(справочное)

Пример спецификации масштаба и диапазона

В.1 Общие положения

Алгоритм определения масштаба и диапазона значений объекта RT-SA описан в 6.3.5.3, однако приводится повторно в настоящем приложении для справки.

$$Y = M \times X + B,$$

где Y = преобразованная абсолютная величина,

M = (верхнее абсолютное значение — нижнее абсолютное значение) / (верхнее масштабированное значение — нижнее масштабированное значение),

B = верхнее абсолютное значение — (M × верхнее масштабированное значение),

X = масштабированное значение.

Необходимо учесть, что термин *абсолютное значение* не обозначает математическое абсолютное значение, которое всегда положительно, а относится к фактическому измеренному значению.

Формула позволяет заменить измеренные значения (с учетом диапазона смещения и ограниченной точности) на целочисленную скалярную величину, что может уменьшить объем данных, передаваемых между агентом и менеджером. Структуры ScaleRangeSpec8, -16 и -32, описанные в А.3.4, передают верхние и нижние абсолютные значения, а также верхние и нижние масштабированные значения, позволяя менеджеру определить параметры формулы преобразования масштабируемых значений в соответствующие абсолютные значения и подтвердить, что полученные значения находятся в ожидаемом диапазоне.

В рамках агента масштабируемое значение, получаемое на основе фактического измеренного значения, может вычисляться по следующей формуле:

$$X = (R - B) / M,$$

где R = фактическое измеренное значение.

Подходящее значение М позволяет обеспечить наличие масштабируемых значений, чтобы передать абсолютные измеренные значения с требуемой точностью. На практике параметры М и В могут задаваться с учетом разрешающей способности АЦП и прочих характеристик оборудования.

В.2 Пример определения показаний термометра

Ниже приводится пример этого алгоритма. Показания термометра, способного определять температуру по шкале Цельсия от -45 °С до +50 °С при точности 0,5 °С, должны передаваться в виде беззнаковых значений, используя структуру ScaleRangeSpec8.

Использование для структуры ScaleRangeSpec8 следующих значений:

нижнее абсолютное значение = -45.0;

верхнее абсолютное значение = 50,0;

нижнее масштабированное значение = 0;

верхнее масштабированное значение = 190,

позволяет получить:

M = (50,0 - (-45,0))/(190 - 0) = 0,5;

B = 50,0 - (0,5 × 190) = -45,0.

Некоторые характерные масштабированные и преобразованные значения представлены в таблице В.1 и на рисунке В.1.

Таблица В.1 — Схема преобразования

Масштабированный (x)	Преобразованный (y)
0	-45,0
50	-20,0
100	5,0
150	30,0
190	50,0

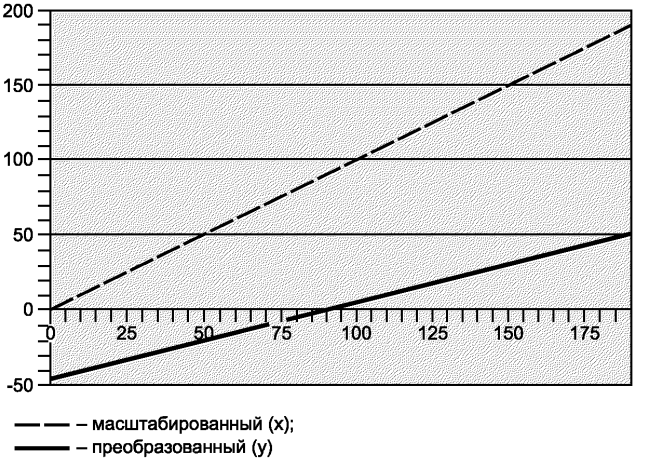


Рисунок В.1 — Графическое представление преобразования

Приложение С
(справочное)

Концепция PM-store

С.1 Общие положения

Понятие объекта PM-store предлагает метод представления, доступа и передачи большого количества измеренных данных, хранящихся в памяти агента. Информация организована в виде иерархической объектной модели с возможностью структурирования данных в соответствии с их природой.

На верхнем уровне объект PM-store представляет собой точку первичного доступа ко всей информации о хранящихся измеренных данных. Агент, поддерживающий постоянно хранящиеся данные измерений, может создавать экземпляры одного или нескольких объектов PM-store. Объект PM-store является частью конфигурации устройства. Прямой доступ к нему можно получить с помощью служб доступа к объектам, описанных в настоящем стандарте.

Каждый PM-store может хранить 0, 1 или несколько сегментов PM-segment, которые представляют собой контейнеры фактических данных. Количество сегментов PM-segment может меняться в результате работы агента. Другими словами, агент может создавать новые сегменты PM-segment на основе периодов времени, объема хранимых данных или даже ручного управления пользователем.

Понятие PM-store предлагает информационную модель с двухуровневой иерархией с несколькими объектами PM-segment внутри нескольких объектов PM-store.

К типичному варианту использования нескольких объектов PM-store относится следующий случай:

- Если агент хранит данные с разными характеристиками (например, апериодические и периодические измерения), отдельные объекты PM-store используются для определения оптимизированных типов хранимых данных, благодаря чему экономится память.

К типичному варианту использования нескольких PM-segment относится следующий случай:

- Если агенту необходимо структурировать хранимые данные в более иерархической форме, можно использовать несколько экземпляров объектов PM-store с несколькими экземплярами объектов PM-segment для моделирования такой иерархии (например, можно задействовать объект PM-store для представления сеанса обучения, а затем использовать объект PM-segment для моделирования индивидуальных упражнений в рамках сеанса обучения).

Для хранения фактических данных используется понятие той же карты значений атрибутов, что используется для атрибутов объектов Metric. Специальный сопоставляющий атрибут позволяет задать структуру двоичных хранящихся данных без излишних затрат ресурсов на идентификацию, полей длины и т. д. в фактических хранящихся и передаваемых двоичных данных. Предполагается, что хранящиеся данные по сути представляют собой большой массив одинаково форматированных данных.

После проверки информации в объектах PM-store менеджер инициирует передачу хранящихся данных. Менеджер может выбрать эпизоды данных для передачи. После этого фактическая передача выполняется агентом с помощью квитированных сообщений отчетов о событиях. Ожидается, что агент заполняет данными структуру SegmentDataEvent до максимально доступного размера.

С.2 Иерархия постоянно хранящихся объектов класса Metric

С.2.1 Общие положения

Постоянное хранилище результатов измерений (Persistent metric, PM) состоит из следующих четырех ключевых частей:

- PM-store — Данный объект относится к верхнему уровню и содержит атрибуты объекта хранения, а также ноль или более сегментов PM-segment;
- PM-segment — Данный объект содержит атрибуты, описывающие сегмент, а также ноль или более записей;
- Запись — Каждая запись содержит необязательный заголовок записи и один или несколько элементов;
- Элемент — Каждый элемент содержит результаты одного или нескольких измерений.

На рисунке С.1 показан пример схемы связи этих четырех частей. Более подробную информацию см. в оставшейся части настоящего приложения.

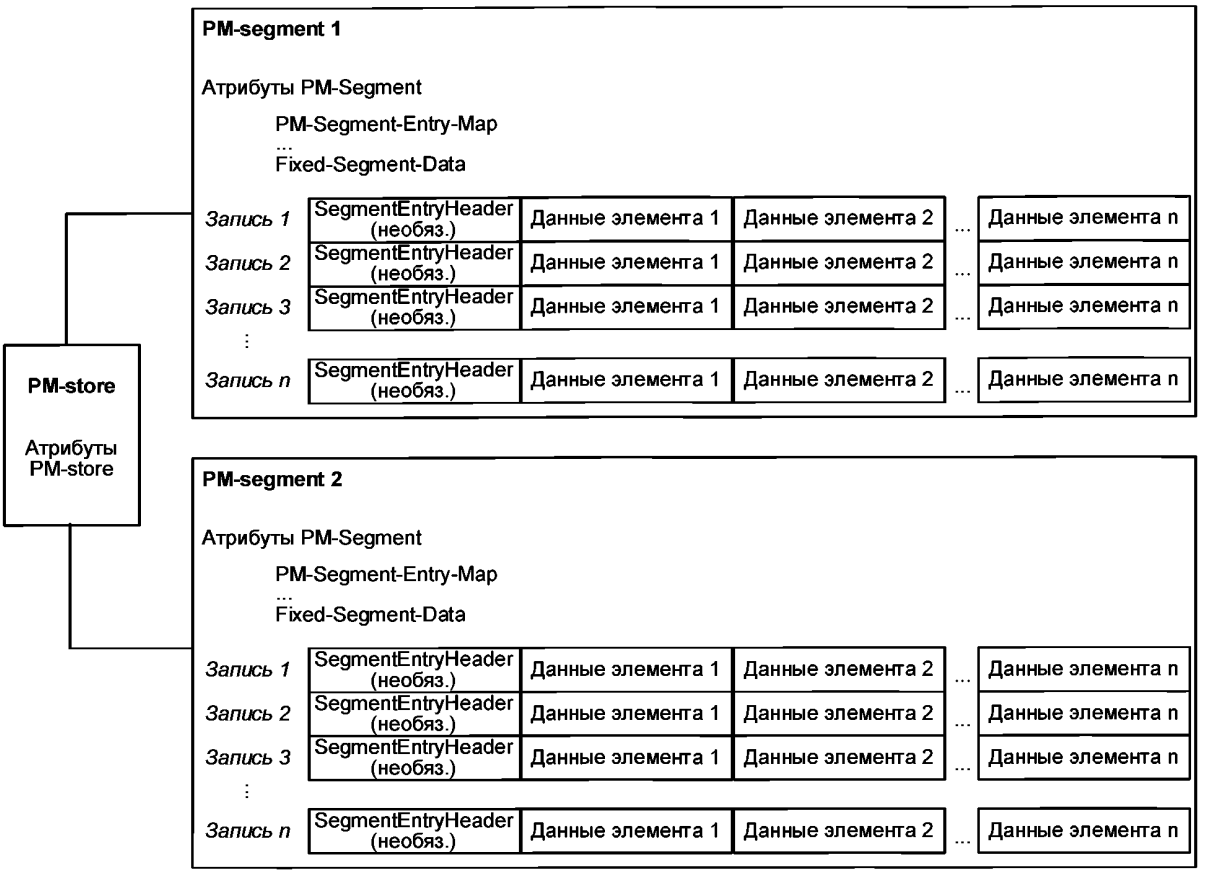


Рисунок С.1 — Объект PM-store с двумя сегментами PM-segment и атрибутами fixed-segment-data в сегментах

С.2.2 Объект PM-store

Поддержка объекта PM-store не обязательна. Поддержка объектов, атрибутов, методов и сопутствующих событий объектов PM-store необходима лишь тем агентам, которые хотят постоянно хранить результаты измерений. Менеджер получает информацию обо всех поддерживаемых объектах PM-store из конфигурации агента. Атрибуты объекта PM-store описывают общие характеристики хранящихся данных (например, периодическое или эпизодическое хранение значений).

Агент может использовать более одного хранилища PM-store. Несколько хранилищ используются для представления данных с разными форматами или характеристиками, а также для группировки данных по различным логическим группам. Кроме того, объект PM-store служит точкой входа для всех методов, связанных с хранящимися результатами измерений (например, при извлечении данных менеджером).

Агент управляет количеством сегментов PM-segment, существующих в объекте PM-store. Если данные отсутствуют, то объект PM-store может содержать нулевое число элементов. Если данные присутствуют, то объект PM-store содержит один или несколько объектов PM-segment. Поскольку количество объектов PM-segment не постоянно, то эти объекты не входят в конфигурацию агента. Вместо этого объект PM-store содержит информацию о доступных сегментах PM-segment в форме атрибутов PM-store, которые запрашиваются с помощью службы GET.

С.2.3 Объект PM-segment

Базовый формат данных сегмента PM-segment показан на рисунке С.2.



Рисунок С.2 — Формат данных сегмента PM-segment

Сегмент содержит *k* записей. Формат записи сегмента определяется его атрибутом PM-Segment-Entry-Map. В записи отображаются данные, хранящиеся в определенный момент времени. Каждой записи может предшествовать необязательный заголовок (например, содержащий метку времени), общий для всех элементов записи. В записи содержится n элементов, формат которых определяется картой значений атрибута. Обычно запись содержит результат измерения (например, числовое значение и перечисление). Итоговая структура данных не содержит каких-либо полей длины или идентификаторов атрибутов, поэтому очень компактна.

Содержание сегмента PM-segment обычно представляет один эпизод измерений. Такой эпизод имеет временной контекст (например, данные, хранящиеся в этом сегменте, собраны в интервале 12:00 - 15:00), некоторые связанные атрибуты и массив хранения, содержащий фактические (измеренные) данные для эпизода, содержащегося в атрибуте Fixed-Segment-Data (см. рисунок С.3).

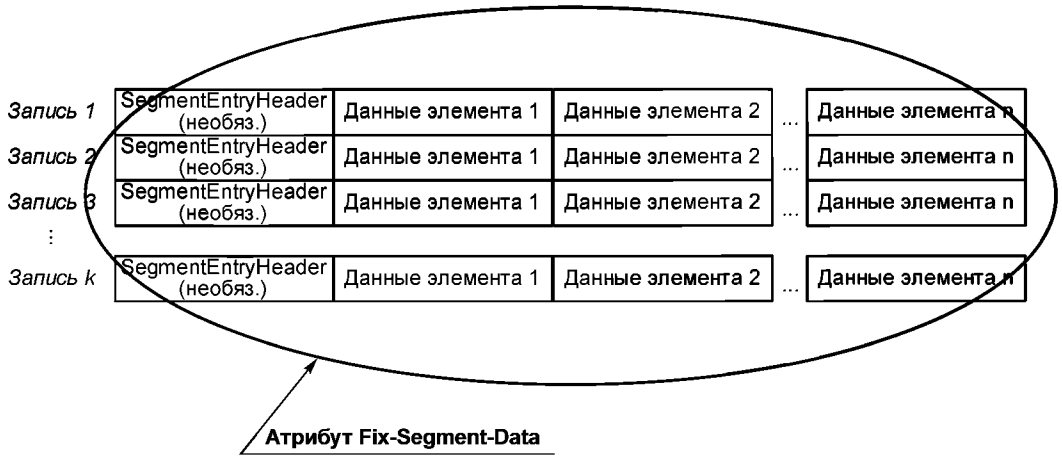


Рисунок С.3 — Атрибут fixed-segment-data, содержащий фактические хранимые данные

Хранилище PM-store может содержать ноль или более сегментов PM-segment (например, ни одного, если данные еще не сохранены; один или несколько в зависимости от хранящихся эпизодов и возможностей агента).

К примеру, сегмент PM-segment часов для бега может содержать сохраненную информацию об одном цикле тренировки (как вариант — 5-минутный отрезок времени с 12:00). Устройство может хранить несколько сегментов (например, несколько таких тренировочных циклов).

С.2.4 Запись PM-segment (в fixed-segment-data)

Атрибут Fixed-Segment-Data содержит записи и элементы. На рисунке С.4 записи отображены в виде строк. Все записи в сегменте обладают структурой, определяемой атрибутом PM-Segment-Entry-Map,. Он очень похож на атрибут Attribute-Value-Map, определенный для объектов Metric. Однако существует возможность группировки нескольких результатов измерений в одной записи.



Рисунок С.4 — Запись (массив элементов в fixed-segment-data)

Атрибут PM-Segment-Entry-Map определяет список измерений, хранящихся в одной записи. Для каждого измерения также определен список хранящихся атрибутов. Кроме того, при необходимости можно задать заголовок (например, чтобы добавить общую метку времени), общий для всех элементов записи.

Используя приведенный выше пример с часами для бега, предположим, что агент каждую секунду сохраняет информацию о частоте сердечных сокращений, текущей скорости бега и значении SpO2. Единственным атрибутом, хранящимся для этих измерений, будет числовое значение (определенное в атрибуте PM-Segment-Entry-Map). В этом случае заголовок записи не требуется, поскольку измерения периодичны и метка времени не обязательна. Для периодических измерений время конкретного сохраненного измерения вычисляется на основе начального и конечного времени, периода считывания и индекса записи. Вследствие этого отдельная метка времени для каждого измерения, равно как и заголовок с меткой времени, не требуется.

Таким образом, каждая строка записи имеет следующие три элемента:

ЧСС	**Скорость бега**	**SpO2**
120	10	98

С.2.5 Элемент записи сегмента PM-segment

Элемент содержит двоичное представление определенных атрибутов одного объекта Metric (см. рисунок С.5). Структура SegmEntryElem (см. А.11.8) в атрибуте PM-Segment-Entry-Map определяет каждый элемент записи.

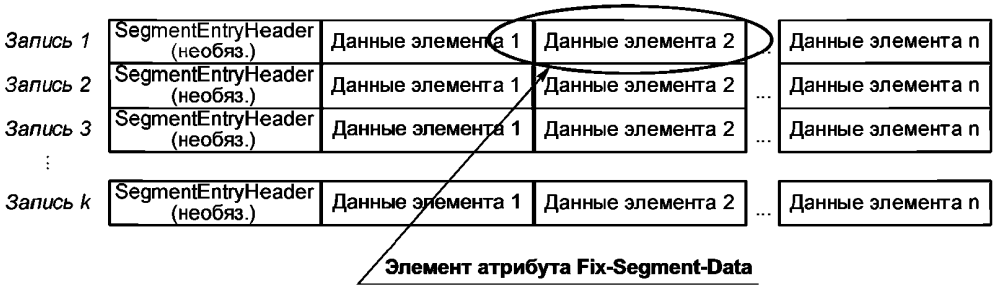


Рисунок С.5 — Элемент: набор атрибутов одного измерения

В примере с часами для бега одна запись содержит три результата измерений. Для каждого результата определен один атрибут, числовое наблюдаемое значение. Таким образом, значения частоты сердечных сокращений, скорости и насыщенности периферийным кислородом являются отдельными элементами записи.

Однако атрибут PM-Segment-Entry-Map может содержать не только атрибуты, связанные с наблюдаемыми значениями. Например, можно добавить такие атрибуты, как действительность, метки времени, коды единиц измерения и так далее.

Приложение D
(справочное)

Типы транспортных профилей

D.1 Общие положения

В настоящем стандарте используется понятие «тип», группирующее и дифференцирующее службы, предлагаемые доступными технологиями транспорта, профилированными для использования в серии стандартов ИСО/ИИЭР 11073. А именно в этой серии выделяются следующие типы транспортных профилей:

Тип 1. Транспортные профили, содержащие «надежные» и «лучшие из возможных» транспортные службы. Они имеют один или несколько виртуальных каналов «надежных» транспортных служб и ноль или более виртуальных каналов «лучших из возможных» транспортных служб.

- Тип 2. Транспортные профили, содержащие только однонаправленную транспортную службу.
- Тип 3. Транспортные профили, содержащие только «лучшие из возможных» транспортные службы. Они имеют один или несколько виртуальных каналов «лучших из возможных» транспортных служб.

Причина, по которой типам транспортных профилей придается большое значение, состоит в том, что предлагаемые ими различные транспортные службы влияют на реализацию некоторой функциональности верхнего уровня. В частности, они влияют на реализацию механизма подтверждаемых служб, описанных в настоящем стандарте.

Прилагательное «подтверждаемый» в термине «механизм подтверждаемых служб» имеет следующее определение:

- на уровне данных (службы EVENT REPORT) позволяет агенту знать, когда менеджер «принял ответственность» за часть данных, и агент может удалить такие данные;
- на уровне управления (службы ACTION, GET и SET) позволяет менеджеру знать, когда агент «завершил» запрошенную транзакцию.

D.2 Тип 1

Транспортный профиль типа 1 предусматривает одновременно как надежные, так и лучшие из возможных транспортные службы. Учитывая определение и цели механизма подтверждаемой службы, подтверждаемые сообщения достаточно чувствительны к потере пакетов данных. Следовательно, самой подходящей транспортной службой для всех подтверждаемых сообщений будет надежная транспортная служба.

Кроме того, согласно настоящему стандарту (см. 8.4) конечные автоматы агента и менеджера синхронизированы. Наличие синхронизированных конечных автоматов косвенно подразумевает использование надежной транспортной службы между двумя конечными автоматами, обеспечивающей доставку сообщений или уведомляющей о сбое доставки. Таким образом, все сообщения, связанные с процессом ассоциации, передаются через надежную транспортную службу. В целях упрощения ссылок конечные автоматы агента и менеджера, описанные в 8.4, именуются конечными автоматами типа 1, чтобы соотнести их с транспортным профилем типа 1.

Для неподтверждаемых сообщений прикладное программное обеспечение может самостоятельно принимать решение об использовании надежной или лучшей из возможных транспортных служб (см. таблицу D.1).

Таблица D.1 — Использование транспортного профиля типа 1

Транспортная служба	Сообщения ИИЭР 11073-20601	
	Процедура ассоциации и подтверждаемые сообщения	Неподтверждаемые сообщения
Лучшая из возможных	Не поддерживаются	Поддерживаются
Надежная	Поддерживаются	Поддерживаются

Транспортный профиль, который считается профилем типа 1, поддерживает один или несколько надежных виртуальных каналов или ноль или более лучших из возможных виртуальных каналов.

D.3 Тип 2

Транспортный профиль типа 2 позволяет использовать только однонаправленную транспортную службу. Учитывая определение и цели механизма подтверждаемой службы, однонаправленная транспортная служба не пригодна для подтверждаемых сообщений. (Менеджер не может отправлять подтверждающие сообщения обратно агенту.)

Будучи однонаправленной, такая служба по своей сути является лучшей из возможных транспортных служб. Транспортный уровень менеджера не может запрашивать повторную передачу на транспортном уровне, если потерян блок данных транспортного протокола (PDU). Таким образом, надежная транспортная служба невозможна при однонаправленной транспортной службе.

При отсутствии надежной транспортной службы конечный автомат типа 1 не может корректно взаимодействовать с транспортным профилем типа 2. Таким образом, для среды типа 2 необходим конечный автомат типа 2 (см. таблицу D.2).

Таблица D.2 — Использование транспортного профиля типа 2

Транспортная служба	Сообщения ИИЭР 11073-20601	
	Процедура ассоциации и подтверждаемые сообщения	Неподтверждаемые сообщения
Лучшая из возможных	Не поддерживаются	Поддерживаются
Надежная	Не поддерживаются	Не поддерживаются

D.4 Тип 3

D.4.1 Общие положения

Транспортный профиль типа 3 позволяет использовать только лучшую из возможных транспортных служб. Отсутствие надежной транспортной службы вызывает определенные трудности с использованием конечного автомата типа 1 и механизма подтверждаемой службы, описанным в настоящем стандарте. Для решения этих проблем существуют различные методы, которые не исключают друг друга.

D.4.2 Тип 3a

Одним из методов разрешения такой ситуации является добавление к лучшей из возможных транспортных служб дополнительной функции надежной транспортной службы. После применения такого метода транспортный профиль типа 3 (только лучший из возможных) фактически станет транспортным профилем типа 1 (надежный и лучший из возможных). В этом случае возможно использование конечного автомата типа 1 и механизма подтверждаемой службы.

Таким образом, транспортный профиль типа 3a представляет собой особый вариант транспортного профиля типа 1.

D.4.3 Тип 3b

Еще одним способом использования только лучшей из возможных транспортных служб будет перенос функциональности надежной транспортной службы в протокол персональных медицинских приборов. Результатом данного метода являются конечный автомат типа 3 и механизм службы подтверждения типа 3 (см. таблицу D.3).

Таблица D.3 — Использование транспортного профиля типа 3b

Транспортная служба	Сообщения IEEE 11073-20601	
	Процедура ассоциации и подтверждаемые сообщения	Неподтверждаемые сообщения
Лучшая из возможных	Поддерживаются	Поддерживаются
Надежная	Не поддерживаются	Не поддерживаются

D.4.4 Тип 3c

Третьим методом использования только лучшей из возможных транспортных служб будет добавление ослабленной дополнительной функции надежного транспорта в лучшую из возможных транспортных служб. Данный метод похож на стратегию для типа 3a. Однако в транспортной службе типа 3c некоторые характеристики надежной транспортной службы будут ослаблены. Ожидается, что такое ослабление будет способствовать более компактной и простой реализации транспорта по сравнению с полностью надежной транспортной службой.

Фактически ослабленные характеристики надежной транспортной службы можно определить, если конечный автомат типа 1 и механизм подтверждаемой службы будут правильно функционировать в среде транспортного профиля типа 3c.

D.5 Выводы

Сводная информация о типах транспортных профилей представлена в таблице D.4.

Таблица D.4 — Типы транспортных профилей

Транспортный профиль	Описание	Вид «2 × 2»			Ассоц. конечный автомат и подтверждаемые сообщения	Режимы передачи данных
Тип 1/3a	Надежный и лучший из возможных		Подтв.	Не подтв.	Тип 1	3
		Лучший из возможных	НЕТ	да		
		Надежный	да	да		
Тип 2	Только однонаправленный		~~Подтв.~~	Неподтв.	Новый тип 2	1
		Лучший из возможных	НЕТ	да		
		~~Надежный~~	НЕТ	НЕТ		
Тип 3b	Только лучший из возможных		Подтв.	Неподтв.	Новый тип 3	2
		Лучший из возможных	да	да		
		~~Надежный~~	НЕТ	НЕТ		
Тип 3c	Ослабленно надежный и лучший из возможных		Подтв.	Неподтв.	Требует определения (возможен тип 1)	3
		Лучший из возможных	НЕТ	да		
		Ослабленно надежный	да	да		

Приложение Е
(обязательное)

Таблицы состояний

Е.1 Общие положения

Целевой аудиторией таблицы Е.1 являются преимущественно пользователи, которые поддерживают стандарты и стремятся обеспечить согласованное использование номеров состояний, приведенных в таблицах Е.3 и Е.4.

Все состояния, приведенные в таблицах состояний агента и менеджера, показаны в таблице Е.1.

Таблица Е.1 — Состояния

Номер состояния	Состояние	Используется агентом	Используется менеджером
1	Отсоединен	Да	Да
2	Соединен, не ассоциирован	Да	Да
3	Соединен, ассоциирующий	Да	Да
4	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Да	
5	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Да	
6	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию		Да
7	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию		Да
8	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Да	
9	Соединен, ассоциирован, конфигурирующий, посылает GetMDS		Да
10	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Да	
11	Соединен, ассоциирован, конфигурирующий, посылает SetTime		Да
12	Соединен, ассоциирован, выполнение	Да	Да
13	Соединен, завершает ассоциацию	Да	Да

Е.2 События

Целевой аудиторией таблицы Е.2 являются преимущественно пользователи, которые поддерживают стандарты и стремятся обеспечить согласованное использование номеров состояний, приведенных в таблицах Е.3 и Е.4.

Все события агента и менеджера определены в таблице Е.2.

В таблице событий используются следующие обозначения.

REQ — запрос от прикладного программного обеспечения, взаимодействующего через интерфейс с конечным автоматом.

IND — состояние, объявленное нижним уровнем программного обеспечения через хорошо определенный прикладной программный интерфейс (API).

Rx — блок данных протокола (PDU), полученный из потока входных данных.

Таблица Е.2 — События

Номер события	Событие
1	IND, транспортное соединение
2	IND, транспортное отсоединение
3	IND, тайм-аут, максимальное количество повторных попыток не достигнуто
4	IND, тайм-аут, достигнуто максимальное количество повторных попыток
5	REQ, ассоциация
6	REQ, завершение ассоциации
7	REQ, прекращение ассоциации

Окончание таблицы Е.2

Номер события	Событие
8	$R_{x\ aarq}$ (*)
9	REQ, допустимая и известная конфигурация
10	REQ, допустимая и неизвестная конфигурация
11	REQ, недопустимая конфигурация
12	Rx, aare (*)
13	Rx, aare (accepted)
14	Rx, aare (accepted-unknown-config)
15	Rx, aare (rejected-*)
16	Rx, rlrq
17	Rx, rlre
18	Rx, abrt
19	Rx, apdu(*), любой полученный APDU, который явно не охвачен этим состоянием (например, поврежден, не известен, не ожидался)
20	REQ, доступная структура ConfigEventReport
21	Rx, roiv-*
22	Rx, roiv-cmip-get, handle = 0
23	Rx, roiv-*, но не (roiv-cmip-get, handle = 0)
24	Rx, roiv-confirmed-event-report
25	Rx, roiv-*, но не (roiv-confirmed-event-report)
26	Rx (rors-*, roer-* или rorj-*)
27	Rx, rors-cmip-confirmed-event-report (unsupported-config), доступны дополнительные конфигурации
28	Rx, rors-cmip-confirmed-event-report (unsupported-config), дополнительные конфигурации не доступны
29	Rx, rors-cmip-confirmed-event-report (accepted-config)
30	Rx (rors-*, roer-* или rorj-*), но не Rx: rors-cmip-confirmed-event-report
31	REQ, объявлена неподдерживаемая конфигурация
32	REQ, объявлена поддерживаемая конфигурация
33	IND, приложение: доступно ConfigEventReport
34	REQ, roiv-cmip-confirmed-*
35	Rx, roiv-cmip-confirmed-action (заданное время)
36	Rx, roiv-cmip-confirmed-action (но не заданное время)
37	Rx, rors-cmip-confirmed-action (заданное время)
38	Rx (rors-*, roer-* или rorj-*), но не Rx: rors-cmip-confirmed-action (установка времени)
39	REQ, roiv-cmip-get, handle = 0
40	REQ, roiv-*, но не roiv-cmip-get, handle = 0
41	Rx, rors-cmip-get, handle = 0
42	IND, тайм-аут, Rx, roiv-cmip-get, handle = 0
43	IND, тайм-аут, Rx, roiv-cmip-confirmed-action (заданное время)
44	Rx, roiv-confirmed-event-report, содержащий конфигурацию
45	Rx, roiv-*, но не roiv-confirmed-event-report, содержащий конфигурацию

Е.3 Таблица состояний агента

Конечный автомат агента должен быть реализован в соответствии с описанием в таблице Е.3.

В таблице состояний агента используются следующие условные обозначения.

REQ — запрос от прикладного программного обеспечения, взаимодействующего через интерфейс с конечным автоматом.

IND — условие, объявленное нижним уровнем программного обеспечения через хорошо определенный прикладной программный интерфейс (API).

Rx — блок данных протокола (PDU), полученный из потока входных данных.

Tx — пакет данных протокола прикладного уровня, отправленный в поток выходных данных.

Идентификатор сигнала — x.y = state.event, где x указано в таблицах Е.1 и Е.2.

Все значения тайм-аута указывают продолжительность ожидания перед объявлением условия «IND, тайм-аут».

Таблица Е.3 — Таблица состояний агента

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
1.1	Отсоединен	IND транспортное соединение	Соединен, не ассоциирован	«Должен» указывает на прикладной уровень	Нет
2.2	Соединен, не ассоциирован	IND транспортное отсоединение	Отсоединен	«Следует» указывает на прикладной уровень	Нет
2.5	Соединен, не ассоциирован	REQ Assoc	Соединен, ассоциирующий	Тайм-аут = TO_{assoc}, retry = сброс	Tx aarq
2.6	Соединен, не ассоциирован	REQ AssocRel	Соединен, не ассоциирован <нет смены состояний>	Нет	Нет
2.7	Соединен, не ассоциирован	REQ AssocAbort	Соединен, не ассоциирован <нет смены состояний>	Можно использовать для синхронизации состояния обеих сторон	Tx abrt (причина не определена)
2.8	Соединен, не ассоциирован	Rx aarq(*)	Соединен, не ассоциирован	Ассоциация агент-агент	Tx aare (отклонено постоянно)
2.12	Соединен, не ассоциирован	Rx aare (*)	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить	Tx abrt (причина не определена)
2.16	Соединен, не ассоциирован	Rx rlrq	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить	Tx abrt (причина не определена)
2.17	Соединен, не ассоциирован	Rx rlre	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить. Игнорировать	Нет
2.18	Соединен, не ассоциирован	Rx abrt	Соединен, не ассоциирован <нет смены состояний>	Нет	Нет
2.19	Соединен, не ассоциирован	Rx apdu(*) Любой APDU, который не охвачен 2.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить	Tx abrt (причина не определена)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
3.2	Соединен, ассоциирующий	IND транспортное отсоединение	Отсоединен	Нет	Нет
3.3	Соединен, ассоциирующий	IND тайм-аут, максимальное количество повторных попыток не достигнуто	Соединен, ассоциирующий <нет смены состояний>	Тайм-аут = TO_{assoc}, прирастить счетчик попыток	Tx aarq
3.4	Соединен, ассоциирующий	IND тайм-аут, достигнуто максимальное количество повторных попыток	Соединен, не ассоциирован	Нет	Tx abrt (причина: тайм-аут ответа)
3.6	Соединен, ассоциирующий	REQ AssocRel	Соединен, не ассоциирован	Нет	Tx abrt (причина не определена)
3.7	Соединен, ассоциирующий	REQ AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
3.8	Соединен, ассоциирующий	Rx aarq(*)	Соединен, не ассоциирован	Ассоциация агент-агент	Tx aare (отклонено постоянно)
3.13	Соединен, ассоциирующий	Rx aare (принято)	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Вызывает прямой переход в состояние «конфигурирующий, ожидает GetMDS»	Нет
3.14	Соединен, ассоциирующий	Rx aare (принята неизвестная конфигурация)	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Менеджер принял ассоциацию, но не имеет конфигурации	Нет
3.15	Соединен, ассоциирующий	Rx aare (отклонено-*)	Соединен, не ассоциирован	Отсутствуют дальнейшие попытки соединения	Нет
3.16	Соединен, ассоциирующий	Rx rlrq	Соединен, не ассоциирован	Не должен происходить. Агент получил запрос на завершение ассоциации, но еще не установил ее	Tx abrt (причина не определена)
3.17	Соединен, ассоциирующий	Rx rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
3.18	Соединен, ассоциирующий	Rx abrt	Соединен, не ассоциирован	Нет	Нет
3.19	Соединен, ассоциирующий	Rx apdu(*) Любой APDU, не охваченный в 3.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
4.2	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	IND транспортное отсоединение	Отсоединен	Нет	Нет
4.4	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	IND тайм-аут	Соединен, не ассоциирован	Нет ответа	Tx abrt (причина: тайм-аут ответа)
4.6	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	REQ AssocRel (*)	Соединен, завершает ассоциацию	Программа запрашивает завершение ассоциации. Тайм-аут = $TO_{release}$	Tx rlrq (*)
4.7	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	REQ AssocAbort	Соединен, не ассоциирован	Прерывание выполнения программы	Tx abrt, причина определяется приложением
4.8	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
4.12	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx aare	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
4.16	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx rlrq	Соединен, не ассоциирован	Нет	Tx rlre (нормальный)
4.17	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
4.18	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx abrt	Соединен, не ассоциирован	Нет	Нет
4.19	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx apdu(*) Любой APDU, не охваченный в 4.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
4.22	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx roiv-cmip-get, handle = 0	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию <нет смены состояний>	Запрещен до момента достижения состояния «Конфигурирующий, ожидает GetMDS»	Tx roer (no-such-object-instance)
4.23	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx roiv-*, но не (roiv-cmip-get, handle = 0)	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию <нет смены состояний>	Запрещен до момента достижения «Выполнение»	Tx roer (no-such-object-instance)
4.26	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Rx (rors-*, roer или rorj)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
4.32	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	REQ Send(ConfigReport)	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Конфигурация агента еще не опробована менеджером. Тайм-аут = TO_{config}	Tx EventReport(ConfigReport)
5.2	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	IND транспортное отсоединение	Отсоединен	Нет	Нет
5.4	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	IND тайм-аут	Соединен, не ассоциирован	Нет ответа	Tx abrt (причина: тайм-аут конфигурации)
5.6	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	REQ AssocRel(*)	Соединен, завершает ассоциацию	Программа запрашивает завершение ассоциации. Тайм-аут = $TO_{release}$	Tx rlrq (*)
5.7	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	REQ AssocAbort	Соединен, не ассоциирован	Прерывание выполнения программы	Tx abrt, причина определяется приложением
5.8	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
5.12	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
5.16	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx rlrq	Соединен, не ассоциирован	Нет	Tx rlre (нормальный)
5.17	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
5.18	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx abrt	Соединен, не ассоциирован	Нет	Нет
5.19	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx apdu(*), Любой APDU, не охваченный в 5.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
5.22	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx roiv-cmip-get, handle = 0	Соединен, ассоциирован, конфигурирующий, ожидает одобрения <нет смены состояний>	Запрещен до момента достижения состояния «Конфигурирующий, ожидает GetMDS»	Tx roer (no-such-object-instance)
5.23	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx roiv-*, но не (roiv-cmip-get, handle = 0)	Соединен, ассоциирован, конфигурирующий, ожидает одобрения <нет смены состояний>	Запрещен до момента достижения состояния «Выполнение»	Tx roer (no-such-object-instance)
5.27	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx rors-cmip-confirmed-event-report (unsupported-config) и доступны дополнительные конфигурации	Соединен, ассоциирован, конфигурирующий, отправляет конфигурацию	Менеджер отклонил конфигурацию, доступны дополнительные конфигурации	Нет
5.28	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx rors-cmip-confirmed-event-report (unsupported-config), дополнительные конфигурации не доступны	Соединен, завершает ассоциацию	Менеджер отклонил конфигурацию, дополнительные конфигурации не доступны	Tx rlrq (*) (причина: больше нет конфигураций)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
5.29	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx rors-cmip-confirmed-event-report (accepted-config)	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Менеджер принял конфигурацию	Нет
5.30	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx (rors-*, roer-* или rorj-*), но не Rx: rors-cmip-confirmed-event-report	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
5.35	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx roiv-cmip-confirmed-action (установка времени)	Соединен, ассоциирован, конфигурирующий, ожидает одобрения <нет смены состояний>	Запрещен до момента достижения состояния «Конфигурирующий, ожидает SetTime»	Tx roer (no-such-object-instance)
5.36	Соединен, ассоциирован, конфигурирующий, ожидает одобрения	Rx roiv-cmip-confirmed-action (но не установка времени)	Соединен, ассоциирован, конфигурирующий, ожидает одобрения <нет смены состояний>	Запрещен до момента достижения состояния «Выполнение»	Tx roer (no-such-object-instance)
8.2	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	IND транспортное отсоединение	Отсоединен	Нет	Нет
8.4	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	IND тайм-аут	Соединен, не ассоциирован	Нет ответа. Не получено roiv-cmip-get, handle = 0. Менеджер должен отправить «roiv-cmip-get, handle=0» в течение периода TO_{get} после перехода в состояние 9	Tx abrt (причина: тайм-аут ответа)
8.6	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	REQ AssocRel(*)	Соединен, завершает ассоциацию	Программа запрашивает завершение ассоциации. Тайм-аут = $TO_{release}$	Tx rlrq (*)
8.7	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	REQ AssocAbort	Соединен, не ассоциирован	Прерывание выполнения программы	Tx abrt, причина определяется приложением
8.16	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Rx rlrq	Соединен, не ассоциирован	Нет	Tx rlre (нормальный)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
8.17	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Rx rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
8.18	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Rx abrt	Соединен, не ассоциирован	Нет	Нет
8.19	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Rx apdu(*) Любой APDU, который не охвачен в 8.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
8.22	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Rx roiv-cmip-get, handle = 0	Соединен, ассоциирован, конфигурирующий, ожидает SetTime (если бит mds-time-mgr-set-time установлен) или соединен, ассоциирован, выполнение (если бит mds-time-mgr-set-time не установлен)	Менеджер исследует MDS. См. 6.3.2.6.1.	rors-cmip-get (атрибуты MDS)
8.23	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS	Rx roiv-*, но не (roiv-cmip-get, handle = 0)	Соединен, ассоциирован, конфигурирующий, ожидает GetMDS <нет смены состояний>	Запрещен, пока не достигнуто состояние «Конфигурирующий, ожидает SetTime» или «Выполнение»	Tx roer (no-such-object-instance)
10.2	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	IND транспортное отсоединение	Отсоединен	Нет	Нет
10.4	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	IND тайм-аут	Соединен, не ассоциирован	Нет ответа. Команда действия Set-Time (или Set-Base-Offset-Time) не получена. Если бит mds-time-mgr-set-time установлен, менеджер должен запросить команду Set-Time (или Set-Base-Offset-Time) в течение периода TO_{ca} после получения «rors-cmip-get, handle=0»	Tx abrt (причина: тайм-аут ответа)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
10.6	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	REQ AssocRel(*)	Соединен, завершает ассоциацию	Программа запрашивает завершение ассоциации. Тайм-аут = $TO_{release}$	Tx rlrq (*)
10.7	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	REQ AssocAbort	Соединен, не ассоциирован	Прерывание выполнения программы	Tx abrt, причина определяется приложением
10.16	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Rx rlrq	Соединен, не ассоциирован	Нет	Tx rlre (нормальный)
10.17	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Rx rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
10.18	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Rx abrt	Соединен, не ассоциирован	Нет	Нет
10.19	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Rx apdu(*), Любой APDU, не охваченный в 10.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
10.35	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Rx roiv-cmip-confirmed-action (установка времени)	Соединен, ассоциирован, выполнение	Менеджер устанавливает агенту местное время. Дополнительную информацию см. в 8.8.3 и 8.12.2.1.	rors-cmip-confirmed-action (заданное время)
10.36	Соединен, ассоциирован, конфигурирующий, ожидает SetTime	Rx roiv-cmip-confirmed-action (но не установка времени)	Соединен, ассоциирован, конфигурирующий, ожидает SetTime <нет смены состояний>	Запрещен до момента достижения состояния «Выполнение»	Tx roer (no-such-object-instance)
12.2	Соединен, ассоциирован, выполнение	IND транспортное отсоединение	Отсоединен	Нет	Нет
12.4	Соединен, ассоциирован, выполнение	IND тайм-аут	Соединен, не ассоциирован	Нет ответа	Tx abrt (причина: тайм-аут ответа)

Продолжение таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
12.6	Соединен, ассоциирован, выполнение	REQ AssocRel	Соединен, завершает ассоциацию	Нет. Тайм-аут = $TO_{release}$	Tx rlrq (нормальный)[12]
12.7	Соединен, ассоциирован, выполнение	REQ AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
12.8	Соединен, ассоциирован, выполнение	Rx aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.12	Соединен, ассоциирован, выполнение	Rx aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.16	Соединен, ассоциирован, выполнение	Rx rlrq	Соединен, не ассоциирован	Если агент имеет вызовы, требующие обработки, предполагается, что он не получит ответ на свой запрос	Tx rlre (нормальный)
12.17	Соединен, ассоциирован, выполнение	Rx rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.18	Соединен, ассоциирован, выполнение	Rx abrt	Соединен, не ассоциирован	Нет	Нет
12.19	Соединен, ассоциирован, выполнение	Rx apdu(*), Любой APDU, не охваченный в 12.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.21	Соединен, ассоциирован, выполнение	Rx roiv-*	Соединен, ассоциирован, выполнение <нет смены состояний>	Нормальная обработка сообщений. Это нормальное состояние «Выполнение»	Tx (rors-*, roer или rorj)
12.26	Соединен, ассоциирован, выполнение	Rx (rors-*, roer или rorj)	Соединен, ассоциирован, выполнение <нет смены состояний>	Нормальная обработка сообщений. Это нормальное состояние «Выполнение»	Нет[13]
13.2	Соединен, завершает ассоциацию	IND транспортное разъединение	Отсоединен	Нет	Нет

[12] AssocRel отправляется только после того, как все текущие идентификаторы вызова устареют.

[13] Если rors-* получены с неизвестным идентификатором вызова, прикладной уровень должен инициировать отправку менеджеру сообщения о прекращении ассоциации, передав «REQ abrt» конечному автомату менеджера.

Окончание таблицы Е.3

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
13.4	Соединен, завершает ассоциацию	IND тайм-аут	Соединен, не ассоциирован	Нет ответа на rlrq	Tx abrt (причина: тайм-аут ответа)
13.6	Соединен, завершает ассоциацию	REQ AssocRel	Соединен, завершает ассоциацию	Ассоциация уже завершается. Игнорировать	Нет
13.7	Соединен, завершает ассоциацию	REQ AssocAbort	Соединен, не ассоциирован	Прекращение нормальной процедуры завершения ассоциации	Tx abrt, причина определяется приложением
13.8	Соединен, завершает ассоциацию	Rx aarq	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
13.12	Соединен, завершает ассоциацию	Rx aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
13.16	Соединен, завершает ассоциацию	Rx rlrq	Соединен, завершает ассоциацию <нет смены состояний>	Обе стороны завершают ассоциацию. Ответ и ожидание своего rlre	Tx rlre (нормальный)
13.17	Соединен, завершает ассоциацию	Rx rlre	Соединен, не ассоциирован	Процесс завершения ассоциации закончен, переход в состояние «Не ассоциирован»	Нет
13.18	Соединен, завершает ассоциацию	Rx abrt	Соединен, не ассоциирован	Нет	Нет
13.19	Соединен, завершает ассоциацию	Rx apdu(*), Любой APDU, не охваченный в 13.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
13.21	Соединен, завершает ассоциацию	Rx roiv-*	Соединен, завершает ассоциацию <нет смены состояний>	Менеджер отправил сообщение вызова, так как агент отправил rlrq. Агент вышел из состояния «Выполнение», поэтому не предоставит ответ	Нет
13.26	Соединен, завершает ассоциацию	Rx (rors-*, roer или rorj)	Соединен, не ассоциирован	Пример 1. Прикладной уровень имеет незавершенные вызовы, но, несмотря на это, ранее запросил завершение ассоциации. Пример 2. Не запрошенные rors-*.	Tx abrt (причина не определена)

Е.4 Таблица состояний менеджера

Конечный автомат менеджера должен быть реализован в соответствии с таблицей Е.4.

В таблице состояний менеджера присутствуют следующие обозначения.

REQ — запрос от прикладного программного обеспечения, взаимодействующего через интерфейс с конечным автоматом.

IND — состояние, утвержденное нижним уровнем программного обеспечения через конкретный программный интерфейс.

Rx — блок данных протокола прикладного уровня, полученный из потока входных данных.

Tx — пакет данных протокола прикладного уровня, отправленный через поток выходных данных.

Таблица Е.4 — Таблица состояний менеджера

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
1.1	Отсоединен	IND, транспортное соединение	Соединен, не ассоциирован	«Должен» указывает на прикладной уровень	Нет
2.2	Соединен, не ассоциирован	IND, транспортное отсоединение	Отсоединен	«Следует» указывает на прикладной уровень	Нет
2.6	Соединен, не ассоциирован	REQ, AssocRel	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить. Игнорировать	Нет
2.7	Соединен, не ассоциирован	REQ, AssocAbort	Соединен, не ассоциирован <нет смены состояний>	Можно использовать для синхронизации состояния обеих сторон	Tx abrt (причина не определена)
2.8	Соединен, не ассоциирован	Rx, aarq(*)	Подключено, ассоциируется	Запрос ассоциации	Нет
2.12	Соединен, не ассоциирован	Rx, aare (*)	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить	Tx abrt (причина не определена)
2.16	Соединен, не ассоциирован	Rx, rlrq	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить	Tx abrt (причина не определена)
2.17	Соединен, не ассоциирован	Rx, rlre	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить. Игнорировать	Нет
2.18	Соединен, не ассоциирован	Rx, abrt	Соединен, не ассоциирован <нет смены состояний>	Нет	Нет
2.19	Соединен, не ассоциирован	Rx, apdu(*) Любой APDU, не охваченный в 2.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован <нет смены состояний>	Не должен происходить	Tx abrt (причина не определена)
3.2	Соединен, ассоциирующий	IND, транспортное отсоединение	Отсоединен	Нет	Нет

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
3.6	Соединен, ассоциирующий	REQ, Assoc-Rel	Соединен, не ассоциирован	Нет	Tx abrt (причина не определена)
3.7	Соединен, ассоциирующий	REQ, AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
3.8	Соединен, ассоциирующий	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
3.9	Соединен, ассоциирующий	REQ, ассоциация приемлема, конфигурация известна	Соединен, ассоциирован, выполнение	Агент и конфигурация известны менеджеру	Tx aare (принято)
3.10	Соединен, ассоциирующий	REQ, ассоциация приемлема, конфигурация не известна	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Менеджер принимает запрос ассоциации, но не обладает действительной информацией о конфигурации агента. Тайм-аут = TO_{config}	Tx, aare (принята неизвестная конфигурация)
3.11	Соединен, ассоциирующий	REQ, недопустимая конфигурация	Соединен, не ассоциирован	Менеджер определяет, что ассоциация неприемлема	Tx aare (отклонено-*)
3.12	Соединен, ассоциирующий	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
3.16	Соединен, ассоциирующий	Rx, rlrq	Соединен, не ассоциирован	Не должен происходить. Менеджер получил запрос завершения ассоциации, но еще не установил ее	Tx abrt (причина не определена)
3.17	Соединен, ассоциирующий	Rx, rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
3.18	Соединен, ассоциирующий	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
3.19	Соединен, ассоциирующий	Rx, apdu(*) Любой APDU, не охваченный в 3.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
6.2	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	IND, транспортное отсоединение	Отсоединен	Нет	Нет
6.4	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	IND, тайм-аут	Соединен, не ассоциирован	Нет ответа. Отчет о конфигурации не получен	Tx abrt (причина: тайм-аут конфигурации)

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
6.6	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	REQ, Assoc-Rel	Соединен, завершает ассоциацию	Нет. Тайм-аут = $TO_{release}$	Tx, rlrq (нормальный)
6.7	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	REQ, AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
6.8	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
6.12	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
6.16	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, rlrq	Соединен, не ассоциирован	Менеджер получил запрос завершения ассоциации	Tx rlre (нормальный)
6.17	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
6.18	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
6.19	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, apdu (*) Любой APDU, не охваченный в 6.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
6.26	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx (rors-*, roer или rorj)	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию <нет смены состояний>	Менеджер может отправить roiv-cmip-get (handle=0). См. 6.3.2.6.1. Ожидается, что в дальнейшем это устареет	Нет[14]

[14] Если rors-* получены с неизвестным идентификатором вызова, прикладной уровень должен инициировать отправку агенту сообщения о прекращении ассоциации, передав «REQ abrt» конечному автомату агента.

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
6.44	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, roiv-confirmed-event-report, содержащий конфигурацию	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Отчет о событиях, содержащий конфигурацию агента. Тайм-аут = TO_{config}	Нет
6.45	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Rx, roiv-*, но не roiv-confirmed-event-report, содержащий конфигурацию	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
7.2	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	IND, транспортное отсоединение	Отсоединен	Нет	Нет
7.4	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	IND, тайм-аут	Соединен, не ассоциирован	Нет ответа	Tx abrt (причина: тайм-аут ответа)
7.6	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	REQ, Assoc-Rel	Соединен, завершает ассоциацию	Нет	Tx, rlrq (нормальный)
7.7	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	REQ, AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
7.8	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
7.12	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
7.16	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, rlrq	Соединен, не ассоциирован	Менеджер получил запрос на разрыв ассоциации.	Tx rlre (нормальный)
7.17	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)[15]

[15] Если rors-* получены с неизвестным идентификатором вызова, прикладной уровень должен инициировать отправку агенту сообщения о прекращении ассоциации, передав «REQ abrt» конечному автомату агента.

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
7.18	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
7.19	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, apdu(*), Любой APDU, не охваченный в 7.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
7.24	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, roiv-confirmed-event-report	Соединен, не ассоциирован	Агент отправляет отчеты о событиях до согласования конфигурации	Tx abrt (причина не определена)
7.25	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx, roiv-*, но не (roiv-confirmed-event-report)	Соединен, не ассоциирован	Агент отправляет только сообщения отчетов о событиях. Никогда не должно происходить	Tx abrt (причина не определена)
7.26	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	Rx (rors-*, roer или rorj)	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию <нет смены состояний>	Менеджер мог отправить roiv-cmip-get (handle=0). См. 6.3.2.6.1. Ожидается, что в дальнейшем это устареет	Нет
7.31	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	REQ, агент предоставил неподдерживаемую конфигурацию	Соединен, ассоциирован, конфигурирующий, ожидает конфигурацию	Тайм-аут = TO_{config}	Tx rors-cmip-configuration-event (неподдерживаемая конфигурация)
7.32	Соединен, ассоциирован, конфигурирующий, проверяет конфигурацию	REQ, агент предоставил поддерживаемую конфигурацию	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Нет	Tx rors-cmip-configuration-event (поддерживаемая конфигурация)
9.2	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	IND, транспортное отсоединение	Отсоединен	Нет	Нет
9.4	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	IND, тайм-аут	Соединен, не ассоциирован	Нет ответа. Не получено rors-cmip-get, handle = 0	Tx abrt (причина: тайм-аут ответа)
9.6	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	REQ, AssocRel	Соединен, завершает ассоциацию	Нет. Тайм-аут = $TO_{release}$	Tx, rlrq (нормальный)

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
9.7	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	REQ, AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
9.8	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
9.12	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
9.16	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, rlrq	Соединен, не ассоциирован	Менеджер получил запрос завершения ассоциации	Tx rlre (нормальный)
9.17	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
9.18	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
9.19	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, apdu(*) Любой APDU, не охваченный в 9.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
9.39	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	REQ, roiv-cmip-get, handle = 0	Соединен, ассоциирован, конфигурирующий, посылает GetMDS <нет смены состояний>	Сразу после перехода в подсостояние «Конфигурирующий, посылает GetMDS» менеджер должен передать агенту команду службы GET с handle = 0, чтобы извлечь значения всех реализованных атрибутов объектов MDS. См. 6.3.2.6.1. Менеджер должен отправить «roiv-cmip-get, handle=0» в течение периода TO_{get} после перехода в состояние 9	Tx, roiv-cmip-get, handle = 0

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
9.40	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	REQ, roiv-*, но не roiv-cmip-get, handle = 0	Соединен, ассоциирован, конфигурирующий, посылает GetMDS <нет смены состояний>	Запрещен, пока не достигнуто состояние «Конфигурирующий, ожидает SetTime» или «Выполнение»	Нет
9.41	Соединен, ассоциирован, конфигурирующий, посылает GetMDS	Rx, rors-cmip-get, handle = 0	Соединен, ассоциирован, конфигурирующий, посылает SetTime (если бит mds-time-mgr-set-time установлен) или соединен, ассоциирован, выполнение (если бит mds-time-mgr-set-time не установлен)	В зависимости от значения бита mds-time-mgr-set-time менеджер решает, необходимо ли установить агенту местное время. Если бит mds-time-mgr-set-time установлен, менеджер должен вызвать команду Set-Time (или Set-Base-Offset-Time) в течение периода TO_{ca} после получения «rors-cmip-get, handle=0»	Tx, roiv-cmip-confirmed-action (установка времени), если бит mds-time-mgr-set-time установлен или Нет, если бит mds-time-mgr-set-time не установлен
11.2	Соединен, ассоциирован, конфигурирующий, посылает SetTime	IND, транспортное отсоединение	Отсоединен	Нет	Нет
11.4	Соединен, ассоциирован, конфигурирующий, посылает SetTime	IND, тайм-аут	Соединен, не ассоциирован	Нет ответа. Не получено rors-cmip-get, handle = 0	Tx abrt (причина: тайм-аут ответа)
11.6	Соединен, ассоциирован, конфигурирующий, посылает SetTime	REQ, AssocRel	Соединен, завершает ассоциацию	Нет. Тайм-аут = $TO_{release}$	Tx, rlrq (нормальный)
11.7	Соединен, ассоциирован, конфигурирующий, посылает SetTime	REQ, AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
11.8	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
11.12	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
11.16	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, rlrq	Соединен, не ассоциирован	Менеджер получил запрос завершения ассоциации	Tx rlre (нормальный)
11.17	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
11.18	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
11.19	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, apdu(*) Любой APDU, не охваченный в 11.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
11.37	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx, rors-cmip-confirmed-action (установка времени)	Соединен, ассоциирован, выполнение	Менеджер устанавливает агенту местное время. Дополнительную информацию см. в 8.8.3 и 8.12.2.1	Нет
11.38	Соединен, ассоциирован, конфигурирующий, посылает SetTime	Rx (rors-*, roer-* или rorj-*), но не Rx: rors-cmip-confirmed-action (установка времени)	Соединен, ассоциирован, конфигурирующий, посылает SetTime <нет смены состояний>	Запрещен до момента достижения состояния «Выполнение»	Нет
12.2	Соединен, ассоциирован, выполнение	IND, транспортное отсоединение	Отсоединен	Нет	Нет
12.4	Соединен, ассоциирован, выполнение	IND, тайм-аут	Соединен, не ассоциирован	Нет ответа	Tx abrt (причина: тайм-аут ответа)
12.6	Соединен, ассоциирован, выполнение	REQ, AssocRel	Подключено, завершение ассоциации	Нет	Tx, rlrq (нормальный)[16]
12.7	Соединен, ассоциирован, выполнение	REQ, AssocAbort	Соединен, не ассоциирован	Нет	Tx abrt, причина определяется приложением
12.8	Соединен, ассоциирован, выполнение	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)

[16] AssocRel отправляется только после того, как все текущие идентификаторы вызова устареют.

Продолжение таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
12.12	Соединен, ассоциирован, выполнение	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.16	Соединен, ассоциирован, выполнение	Rx, rlrq	Соединен, не ассоциирован	Нет	Tx rlre (нормальный)
12.17	Соединен, ассоциирован, выполнение	Rx, rlre	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.18	Соединен, ассоциирован, выполнение	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
12.19	Соединен, ассоциирован, выполнение	Rx, apdu(*) Любой APDU, не охваченный в 12.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.26	Соединен, ассоциирован, выполнение	Rx (rors-*, roer или rorj)	Соединен, ассоциирован, выполнение <нет смены состояний>	Нормальная обработка сообщений. Это нормальное состояние «Выполнение»	Нет[17]
12.44	Соединен, ассоциирован, выполнение	Rx, roiv-confirmed-event-report, содержащий конфигурацию	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
12.45	Соединен, ассоциирован, выполнение	Rx, roiv-*, но не roiv-confirmed-event-report, содержащий конфигурацию	Соединен, ассоциирован, выполнение <нет смены состояний>	Нормальная обработка сообщений. Это нормальное состояние «Выполнение»	Tx (rors-*, roer или rorj)
13.2	Соединен, завершает ассоциацию	IND, транспортное отсоединение	Отсоединен	Нет	Нет
13.4	Соединен, завершает ассоциацию	IND, тайм-аут	Соединен, не ассоциирован	Нет ответа	Tx abrt (причина: тайм-аут ответа)
13.6	Соединен, завершает ассоциацию	REQ, Assoc-Rel	Соединен, завершает ассоциацию <нет смены состояний>	Уже завершает ассоциацию. Игнорировать.	Нет

[17] Если rors-* получены с неизвестным идентификатором вызова, прикладной уровень должен инициировать отправку агенту сообщения о прекращении ассоциации, передав «REQ abrt» конечному автомату агента.

Окончание таблицы Е.4

Идентификатор сигнала	Начальное состояние	Событие/ входной поток	Следующее состояние	Семантическое поведение/примечание	Поток Tx (выходной)/ передача события
13.7	Соединен, завершает ассоциацию	REQ, AssocAbort	Соединен, не ассоциирован	Прекращение нормальной процедуры завершения ассоциации	Tx abrt, причина определяется приложением
13.8	Соединен, завершает ассоциацию	Rx, aarq(*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
13.12	Соединен, завершает ассоциацию	Rx, aare (*)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
13.16	Соединен, завершает ассоциацию	Rx, rlrq	Соединен, завершает ассоциацию <нет смены состояний>	Обе стороны завершают ассоциацию. Ответ и ожидание своего rlre	Tx rlre (нормальный)
13.17	Соединен, завершает ассоциацию	Rx, rlre	Соединен, не ассоциирован	Процесс завершения ассоциации закончен. Переход в состояние «Не ассоциирован»	Нет
13.18	Соединен, завершает ассоциацию	Rx, abrt	Соединен, не ассоциирован	Нет	Нет
13.19	Соединен, завершает ассоциацию	Rx, apdu(*) Любой APDU, не охваченный в 13.* (например, поврежден, неизвестный, не ожидался)	Соединен, не ассоциирован	Не должен происходить	Tx abrt (причина не определена)
13.21	Соединен, завершает ассоциацию	Rx, roiv-*	Соединен, завершает ассоциацию <нет смены состояний>	Агент отправляет сообщение вызова, когда менеджер посылает rlrq. Менеджер вышел из состояния «Выполнение», поэтому не предоставит какой-либо ответ	Нет
13.26	Соединен, завершает ассоциацию	Rx (rors-*, roer или rorj)	Соединен, не ассоциирован	Пример 1. Прикладной уровень имеет незавершенные вызовы, но, несмотря на это, ранее запросил завершение ассоциации. Пример 2. Незапрошенные rors-*	Tx abrt (причина не определена)[18]

[18] Если rors-* получены с неизвестным идентификатором вызова, прикладной уровень должен инициировать отправку агенту сообщения о прекращении ассоциации, передав «REQ abrt» конечному автомату агента.

Приложение F
(обязательное)

Правила кодирования медицинских приборов (MDER)

F.1 Общие положения

Настоящее приложение заимствовано из ISO/IEEE 11073-20101:2004 [21] (см. А.1—А.4) для удобства представления информации.

Настоящее приложение определяет специализированные правила MDER, связанные с представлением последовательных двоичных строк в вычислительной сети относительно организации в памяти компьютера, а также с их представлением в абстрактном синтаксисе (например, в языке программирования) или на диаграммах, используемых в спецификациях. Предполагается, что данная спецификация согласована со всеми без исключения альтернативами нижнего уровня ISO/IEEE 11073; таким образом, реализация на верхних уровнях может обеспечить прозрачность, основанную на конкретном профиле нижнего уровня.

Основные цели MDER заключаются в предоставлении возможности оптимизации производительности форматирования и синтаксического разбора, а также минимизации использования пропускной способности. Оптимизация форматирования фокусируется на способности процессора передачи данных определять предварительно заданные шаблоны передачи, содержащие только динамически изменяемые данные для относительно высокочастотных сообщений, в частности биосигналов.

F.2 Поддерживаемый синтаксис языка ACH.1

ACH.1 представляет собой стандартную нотацию, используемую для определения типов данных, значений и ограничений значений. Эта нотация широко используется в стандартах ВОЗ (взаимосвязь открытых систем) и серии стандартов ISO/IEEE 11073 (например, в информационной модели предметной области, где все определения данных формализуются с помощью ACH.1).

В целях выполнения требований, предъявляемых к характеристикам кодирования и декодирования, и поддержке предварительно заданных шаблонов передачи, правила MDER определяют методы преобразования синтаксиса ACH.1 в поток байтов, пригодный для передачи.

В отличие от других стандартов ISO/OSI, регламентирующих правила кодирования ACH.1 (например, BER и PER), правила MDER оптимизированы только для подмножества ACH.1. Правила MDER не поддерживают полный набор типов данных ACH.1, а лишь определенный, ограниченный набор конструкций ACH.1.

Стандарты серии ISO/IEEE 11073 используют этот ограниченный набор ACH.1 для определения типов данных, применяемых только в управляемых объектах. Следовательно, правила MDER пригодны и достаточны для кодирования структур данных в рамках этих стандартов.

Ограниченный набор конструкций ACH.1, используемый для компонентов PDU в стандартах ISO/IEEE 11073, представляет собой строгое подмножество допустимых типов данных ACH.1, поэтому во время ассоциации возможно использование и согласование других общих стандартных правил кодирования (например, XER и PER).

В таблице F.1 определена специализация ACH.1, пригодная для кодирования с помощью правил MDER. Все компоненты PDU, описанные в нотации ACH.1, предназначенные для кодирования по правилам MDER, являются предметами этой специализации.

Для каждого типа данных ACH.1 эта специализация обозначается символами «I» (добавление с ограничением), «R» (ограничения использования) и «E» (исключение).

Таблица F.1 — Поддерживаемые типы данных ACH.1

Тип ACH.1	Статус	Примечания
INTEGER	R	Ограничения размера должны использоваться для всех типов данных INTEGER, чтобы задать диапазон целочисленных значений. Краткие имена поддерживаемых типов ограничений определяются следующим образом: INT-U8 ::= INTEGER (0..255) INT-I8 ::= INTEGER (-128.. 127) 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 необходимо использовать в определениях типов данных, предназначенных для кодирования по правилам MDER

Окончание таблицы F.1

Тип ACH.1	Статус	Примечания
BIT STRING	R	Ограничения размера должны использоваться для всех типов данных BIT STRING, чтобы задать диапазон значений битовой строки. Краткие имена 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 необходимо использовать с определениями типов данных для кодирования по правилам MDER
OCTET STRING	I	—
SEQUENCE	R	Может не использовать OPTIONAL, DEFAULT или автоматическое тегирование
SEQUENCE OF	I	—
CHOICE	R	Может использоваться явное и неявное тегирование
ANY DEFINED BY	I	Тип ANY DEFINED BY должен идентифицировать компонент в структуре данных (обычно SEQUENCE), который определяет структуру этих данных для декодера/синтаксического анализатора
ANY DEFINED BY	I	Тип ANY DEFINED BY должен идентифицировать компонент в структуре данных (обычно SEQUENCE), который определяет структуру этих данных для декодера/синтаксического анализатора

F.3 Порядок байтов

На рисунке F.1 показано, как представления различных двоичных строк в вычислительной сети отображаются на их представления в памяти. На диаграммах используется сетевой порядок байтов (Network byte order, NBO). Нижеследующие правила пронумерованы для удобства упоминания.

1) Для представления на диаграммах используется формат NBO, показанный на рисунке F.1.

2) В правилах MDER не используется выравнивание. Другими словами, в байтовые строки дополнительные байты не добавляются, например, для получения длин, которые делятся на два или четыре. Однако из соображений эффективности элементы данных переменной длины (например, строки) должны содержать четное число байтов. Например, поскольку большинство элементов данных 16-битовые, то они не будут неправильно выравнены, если строки имеют четное число байтов.

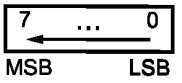
3) Стороны, использующие прикладные профили медицинских приборов (Medical Device Application Profile, MDAP), ограничены использованием соглашения NBO (big endian — прямой порядок байтов).

4) Протокол ассоциации должен использовать правила ISO MDER для обеспечения универсальной интероперабельности при согласовании соглашений о применении MDER. Все остальные блоки PDU, обмен которыми происходит на протяжении жизненного цикла коммуникаций, например PDU CMIP* и ROSE*, будут основаны на MDER,. Суффикс звездочка (*) означает, что MDER используется для оптимизации протокола ISO, основой которого обычно служат правила двоичного кодирования (BER).

Многобайтовые структуры отображаются между вычислительной сетью и компьютерной памятью, при этом они упорядочиваются в памяти компьютера двумя основными способами: *big endian (прямой порядок байтов)* и *little endian (обратный порядок байтов)*. Формат big endian согласуется с NBO, a little endian — не согласуется. Например, в последнем примере на рисунке F.1 структура ABCD будет упорядочена как DCBA. В этом случае, если согласованным протоколом является прямой порядок байтов, компьютеру с обратным порядком байтов необходимо подходящим образом переставить компоненты этих структур при обмене сетевыми данными с памятью. Макросы на языках программирования и машинно-зависимые команды перестановки байтов, которые обычно способствуют нормализации, относятся к вопросам реализации, но их можно упростить, используя ненормативные определения, содержащиеся в настоящем и связанных с ним стандартах.

- Сетевой порядок байтов (NBO)
  - Однобайтовая битовая строка (октет)
    - Последовательность битов: по порядку от младшего значащего бита (least significant bit, LSB) к старшему значащему биту (most significant bit, MSB) (т. е. 0, ..., 7 или 24, ..., 31; порядок

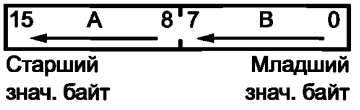
следования битов представляется на диаграммах с помощью стрелки ◄———, которая направлена в сторону бита, передаваемого последним:



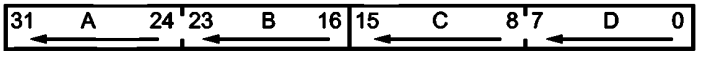
- Многобайтовая строка
  - Неструктурированная строка: массив октетов (октетная строка):
    - Последовательность битов: для каждого байта, как определено для октета.
    - Последовательность байтов: обычно нумеруется от [0] до [*n*-1], например от A[0] до A[*n*-1], где <*n*> = длина в октетах.



  - Структурированная строка: многобайтовая последовательность битов, обычно кратная двум байтам (например, короткое целое число — 16 бит, длинное целое число — 32 бита); представления чисел с плавающей точкой обычно кратны 16 битам, однако в настоящем стандарте определен только 32-битовый формат FLOAT. Можно привести два общих примера (ABCD соответствует порядку байтов).
    - 16-битовая структура (например, короткое целое число)
      - Последовательность битов: каждый байт пересылается согласно определению для октета.
      - Последовательность байтов: пересылается в порядке от старшего значащего байта к младшему значащему байту.
      - Для целых чисел со знаком в качестве знакового бита используется старший значащий бит (MSB) старшего значащего байта.



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



    - По традиции сочетания разных структур изображаются в порядке их расположения внутри сериализованной строки, например:

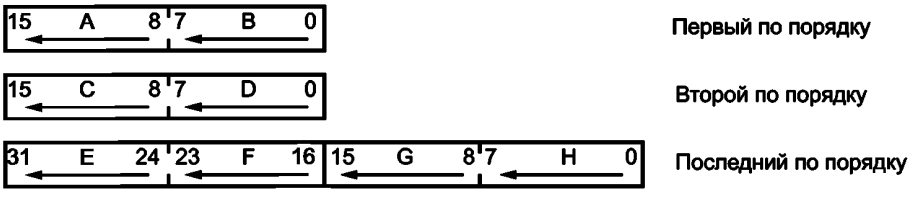


Рисунок F.1 — Пример представления двоичной строки — NBO

F.4 Кодирование

F.4.1 Общие положения

В правилах MDER отсутствует тегирование простых типов. Теги используются только там, где декодеру необходимо различать типы (например, CHOICE). Поля длины используются только для элементов переменной длины и ограничены 16 битами (64 Кбайт (65 536 байт)), которых должно быть достаточно для большинства целей.

Простые типы имеют фиксированную длину, чтобы оптимизировать общий размер кодируемых данных.

Типы SEQUENCE имеют фиксированную длину, так как синтаксические компоненты OPTIONAL не используются.

F.4.2 Тип INTEGER

Кодирование целых значений является примитивным, и октеты представляют значение целых чисел со знаком в дополнительном двоичном коде и абсолютное значение чисел без знака.

На рисунке F.2 показано кодирование октетов для ограниченных по размеру целых значений, поддерживаемых правилами MDER[19].

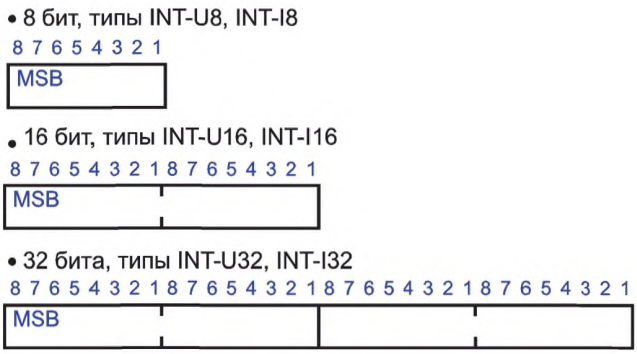


Рисунок F.2 — Кодирование целых чисел

Октеты содержат закодированные целые значения в дополнительном двоичном коде.

F.4.3 Тип BIT STRING

Кодирование значения битовой строки является примитивным, и октеты ее содержания представляют собой набор битов в битовой строке. Длина битовой строки ограничена 8, 16 или 32 битами.

При кодировании бит 0 соответствует старшему значащему биту (MSB), бит 1 представлен следующим битом октета и т. д.

На рисунке F.3 показано кодирование октетов для ограниченных по длине значений битовых строк, поддерживаемых правилами MDER.

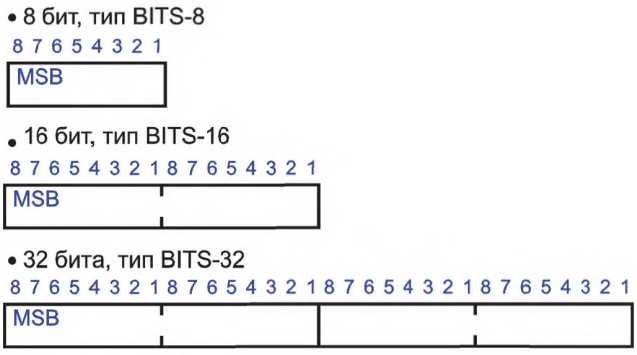


Рисунок F.3 — Кодирование битовой строки

Пример:

Определение

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

следующим образом может быть представлено типом на языке С:

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

(аналогично именованным битам в BIT STRING).

F.4.4 Тип OCTET STRING

Кодирование значения строки октетов является примитивным, и октеты ее содержания представляют собой элементы строки. Для октетов используется кодирование, унаследованное от определения типа строки.

---

[19] В целях содействия стандартизации языка программирования С для этих целых типов данных используются определения из ISO/IEC 9899 [B15].

Октеты могут содержать печатаемые символы ASCII или инкапсулированные двоичные данные. Типы OCTET STRING, содержащие печатаемые символы ASCII, должны иметь четную длину и использовать символ NULL в качестве заполнителя.

Необходимо учесть, что строки, имеющие изначально четное число октетов, могут не завершаться символом NULL.

На рисунке F.4 показано, что правила MDER различают тип OCTET STRING с переменной длиной строки и тип OCTET STRING с фиксированной длиной строки (ограничение по размеру).

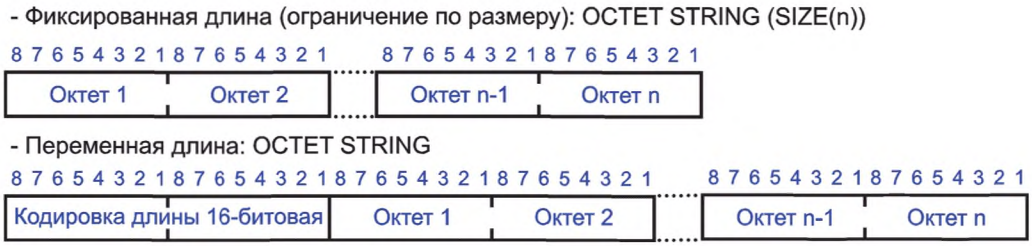


Рисунок F.4 — Кодирование типа OCTET STRING

Типы OCTET STRING с фиксированной длиной кодируются без поля длины и имеют только октеты содержания.

Типы OCTET STRING с переменной длиной кодируются в виде поля длиной 16 бит (целое число без знака в дополнительном двоичном коде), за которым следует указанное в нем число октетов содержания.

Пример:

Определения

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

следующим образом могут быть представлены типами на языке С:

```
typedef unsigned char   fixed_size_label[12];
typedef struct {
    unsigned short      length;
    unsigned char       data[1];  /* это заполнитель для массива подходящего размера */
} variable_label;
```

F.4.5 Тип SEQUENCE

Каждый элемент типа данных SEQUENCE кодируется в порядке его определения в типе ACH.1 SEQUENCE. Никакое выравнивание не производится.

Пример:

Определение

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

следующим образом может быть представлено типом на языке С:

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

Его кодирование по правилам MDER показано на рисунке F.5.



Рисунок F.5 — Пример кодирования SEQUENCE

F.4.6 Тип SEQUENCE OF

Тип SEQUENCE OF кодируется с помощью заголовка, содержащего поле счетчика закодированных элементов (n) и за ним поле длины, указывающее общее количество октетов (m). Длина (m) должна равняться сумме длин всех n закодированных элементов, при этом элементы счетчика и длины не учитываются. После заголовка следуют упорядоченные закодированные элементы. См. рисунок F.6.

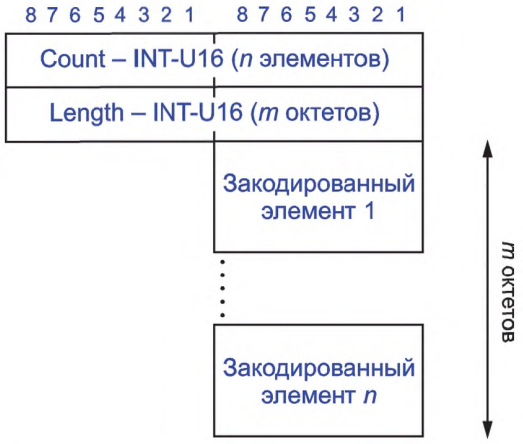


Рисунок F.6 — Кодирование типа SEQUENCE OF

Значения «0» полей счетчика и длины допустимы и означают структуру данных пустого списка.

Пример

Определение

Array1 ::= SEQUENCE OF Entry

следующим образом преобразуется в представление типа языка С:

```
typedef struct {
        unsigned        short   count;
        unsigned        short   length;
        Entry           data[1];            /* Заполнитель для достаточного количества записей */
} Array1;
```

F.4.7 Тип CHOICE

Конструкция CHOICE (выбор) кодируется с помощью заголовка, содержащего поле тега, указывающее выбранную альтернативу, и за ним поле длины, указывающее количество октетов в кодировании этой альтернативы, следующей за заголовком. См. рисунок F.7.



Рисунок F.7 — Кодирование типа CHOICE

Пример

Определения

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

следующим образом преобразуются в представления типа языка С:

```
        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 присваивания значений тегов формулируются следующим образом.

- Теги задаются явно или неявно.

- Абстрактный синтаксис для неявно заданных тегов не содержит явно заданного номера выбора, поэтому требуется правило присвоения значений полю choice_id. Для неявно заданных тегов значения поля choice_id должны начинаться с «1» и последовательно возрастать в порядке выборов, определенных с помощью абстрактного синтаксиса. В приведенном выше примере значения поля choice_id для полей one_type_chosen и two_type_chosen будут соответственно равны 1 и 2.

- Абстрактный синтаксис для явно заданных тегов содержит явно заданный номер выбора, который непосредственно сопоставляется полю choice_id в только что определенном правиле кодирования. В этом случае номера альтернатив являются последовательными, но в зависимости от приложения могут быть несвязными, как показано в следующем примере:

```
choice-type ::= CHOICE {
        one     [1] OneType,    -- Определяет значение 1 тега в MDER.
        four    [4] FourType    -- Определяет значение 4 тега в MDER.
}
```

F.4.8 Типы ANY DEFINED BY и instance-of

Типы ANY DEFINED BY (ASN.1 1988/90) и instance-of (ASN.1 1994) кодируются с помощью заголовка поля длины, указывающего количество октетов в кодировании выбранного значения, следующего за заголовком. См. рисунок F.8.

Данный тип ссылается на встроенные синтаксические определения, заданные с помощью зарегистрированного идентификатора объекта (OID). Сведения о совместимости с предыдущими версиями синтаксиса см. в приложении H стандарта ISO/IEEE 11073-20101:2004 [21].

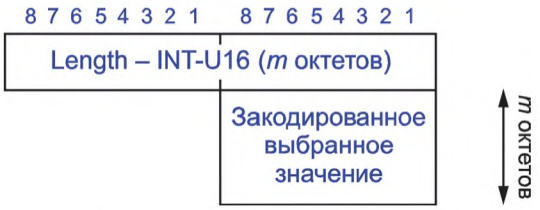


Рисунок F.8 — Кодирование ANY DEFINED BY (instance-of)

Пример

Определения

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

следующим образом преобразуются в представление типа языка C:

```
typedef struct {
        OIDType                 type-id,
        unsigned short          any_length;
        char                    any_data;       /* Заполнитель для типа кодированных данных */
} TestType;
```

Данный пример демонстрирует кодирование байтов SEQUENCE, содержащее контекстно-зависимый идентификатор объекта и значение типа ANY DEFINED BY.

В предыдущем преобразовании поле type-id представляет собой контекстно-свободный идентификатор объекта. Приложение использует 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. Длина может равняться 0, что соответствует отсутствию поля any_data.

Тип instance-of кодирует конструкцию TYPE-IDENTIFIER нотации ACH.1 и идентичен конструкции ANY DEFINED BY, обеспечивающей кодирование в целях обратной совместимости.

F.5 Числа с плавающей точкой

Ограниченное подмножество ACH.1, используемое правилами MDER, не содержит тип данных FLOAT, определенный в нотации ACH.1. Вместо него настоящий стандарт определяет собственные типы данных с плавающей точкой: FLOAT-Type и SFLOAT-Type.

F.6 Структура данных с плавающей точкой — FLOAT-Type

Тип FLOAT-Type отображается как 32-битовая структура, форматированная в соответствии с цифровым форматом медицинских приборов MDNF(medical device numeric format).

Формат MDNF описывает 32-битовое слово, содержащее 8-битовую целочисленную экспоненту со знаком, за которой следует 24-битовая целочисленная мантисса со знаком. См. рисунок F.9.



Рисунок F.9 — Кодирование MDNF

Число имеет следующее представление: (мантисса) × (10**экспонента)[20]. Экспонента и мантисса записываются в двоичном дополнительном коде. Для обозначения точности мантисса и экспонента выравниваются согласно F.8.

Существуют специальные значения, представленные в таблице F.2.

Таблица F.2 — Специальные значения MDNF

Специальное значение	Мантисса	Битовое значение
NaN (не число)	+(2**23 -1)	0x007FFFFF
NRes (не с этой точностью)	-(2**23)	0x00800000
+ INFINITY (БЕСКОНЕЧНОСТЬ)	+(2**23 -2)	0x007FFFFE
- INFINITY (БЕСКОНЕЧНОСТЬ)	-(2**23 -2)	0x00800002
Зарезервировано для последующего использования	-(2**23 -1)	0x00800001

В каждом из этих специальных случаев экспонента должна равняться 0. При этом для нормального представления чисел остаются доступными следующие диапазоны:

- -128 ≤ экспонента ≤ 127;
- -2(2**23 -3) ≤ мантисса ≤ +(2**23 -3);
- NaN =+(2**23 -1);
- NRes = -(2**23);
- ± INFINITY (БЕСКОНЕЧНОСТЬ) = ±(2**23 -2).

Значение NaN следует применять для указания неверного результата вычислительного шага или указания данных, отсутствующих вследствие неспособности оборудования предоставить корректные измерения (возможно, из-за сбоя датчика). Менеджер должен отреагировать на эту информацию заполнением изображаемого поля пробелами или каким-либо другим подходящим способом.

Значение NRes необходимо применять для индексации того, что значение не может быть представлено при доступном диапазоне и точности. Такая ситуация может быть последствием переполнения снизу или сверху, когда количество требуемых значащих цифр выходит за максимальный или минимальный диапазон экспоненты или когда число выходит за максимальный или минимальный диапазон экспоненты.

Значение бита 0x00800001 зарезервировано для последующего использования, чтобы сохранить непрерывность диапазона специальных значений.

F.7 Структура коротких данных с плавающей точкой – SFLOAT-Type

Тип SFLOAT-Type может использоваться для представления чисел с плавающей точкой в очень ограниченном диапазоне значений, что значительно сокращает размер передаваемой информации.

Тип SFLOAT-Type описывается как 16-битовое слово, содержащее 4-битовую целочисленную экспоненту со знаком, за которой следует 12-битовая мантисса со знаком (см. рисунок F.10).

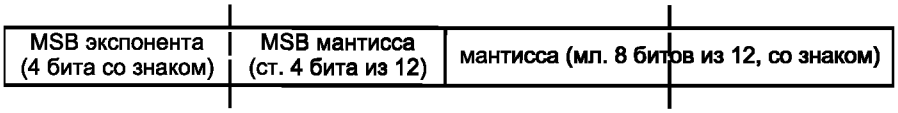


Рисунок F.10 — Короткое представление чисел с плавающей точкой

[20] Две звездочки (**) используются для обозначения экспоненциальной функции.

Число имеет следующее представление: (мантисса) × (10**экспонента). Экспонента и мантисса записываются в двоичном дополнительном коде. Для обозначения точности мантисса и экспонента выравниваются согласно F.8.

Существуют специальные значения, которые представлены в таблице F.3.

Таблица F.3 — Специальные значения SFLOAT-Type

Специальное значение	Мантисса	Битовое значение
NaN	+(2**11 -1)	0x07FF
NRes	-(2**11)	0x0800
+ INFINITY (БЕСКОНЕЧНОСТЬ)	+(2**11 -2)	0x07FE
-INFINITY (БЕСКОНЕЧНОСТЬ)	-(2**11 -2)	0x0802
Зарезервировано для последующего использования	-(2**11 -1)	0x0801

Специальные значения выражают следующие значения:

- NaN [экспонента 0, мантисса +(2**11 -1) → 0x07FF];
- NRes [экспонента 0, мантисса -(2**11) → 0x0800];
- + INFINITY [экспонента 0, мантисса +(2**11 -2) → 0x07FE];
- - INFINITY [экспонента 0, мантисса -(2**11 -2) → 0x0802];

В каждом из этих специальных случаев экспонента должна равняться 0. Использование экспоненты для указания действительных цифр в описании SFLOAT совпадает с тем, что указано в F.8.

Битовое значение 0x0801 зарезервировано, чтобы сохранить непрерывность диапазона специальных значений, но не представляет определенного числового значения.

F.8 Представление точности чисел с плавающей точкой

Число с плавающей точкой может быть представлено методом, обозначающим точность значения, за счет представления мантиссы в целочисленной форме для обозначения количества значащих цифр числа и соответствующего выбора экспоненты. Ниже приведены следующие примеры.

- Если экспонента < 0, целочисленное значение экспоненты показывает количество значащих цифр после десятичной точки. Примеры см. в таблице F.4.

Таблица F.4 — Примеры, соответствующие случаю отрицательной экспоненты

Экспонента	Мантисса	Значение
-3	32 000	32.000
-1	320	32.0

- Если экспонента ≥ 0, количество значащих цифр после десятичной точки равно нулю и мантисса представляет точность значения. Примеры см. в таблице F.5.

Таблица F.5 — Примеры, соответствующие случаю положительной экспоненты

Экспонента	Мантисса	Значение
1	320	3200
2	32	3200

Значения, для которых не требуются цифры справа от десятичной точки (например, частота пульса), представляются с помощью помещения в младший значащий байт мантиссы и нулевой экспоненты. Например, значение 72 будет представлено как 0x00000048. Процент насыщения кислородом может быть аналогичным образом представлен типом SFLOAT-Type, помещая его значение в младший байт мантиссы. Например, значение 98 будет представлено как 0x0062. В то же время, если доступна более высокая точность (например, показание с точностью до 0,1 единицы измерения), значение 72,0 будет представлено как 720×10^{-1}, с мантиссой 0x0002D0 и экспонентой 0xFF, что дает общее представление 0xFF0002D0. При использовании типа SFLOAT-Type значение 98,0 будет представлено как 980×10^{-1}, с мантиссой 0x3D4 и экспонентой 0xF, что дает общее представление 0xF3D4.

Приложение G
(справочное)

Определения типов кодированных данных

Настоящее приложение содержит пример определения заголовочного файла, генерируемого из определений на языке АСН.1, представленных в приложении А. Оно не содержит какие-либо коды, необходимые для преобразования этих структур в машинно независимые буферы двоичных данных, используемых для передачи данных между компьютерами с прямым или обратным порядком байтов.

```
#ifndef PHD_TYPES
#define PHD_TYPES

/*
        В следующих определениях типов могут потребоваться изменения с учетом архитектуры компилятора и
        компьютера.
*/
typedef unsigned char intu8;
typedef unsigned short intu16;
typedef unsigned int intu32;
typedef short int inti16;

typedef struct Any
{
        Intu16   length;
        intu8    value[1];          /* Первый элемент массива */
}       Any;
typedef intu16 OID_Type;

typedef intu16 PrivateOid;

typedef intu16 HANDLE;

typedef intu16 InstNumber;

typedef intu16 NomPartition;
#define         NOM_PART_UNSPEC         0
#define         NOM_PART_OBJ            1
#define         NOM_PART_SCADA          2
#define         NOM_PART_EVT            3
#define         NOM_PART_DIM            4
#define         NOM_PART_VATTR          5
#define         NOM_PART_PGRP           6
#define         NOM_PART_SITES          7
#define         NOM_PART_INFRA          8
#define         NOM_PART_FEF            9
#define         NOM_PART_ECG_EXTN       10
#define         NOM_PART_IDCO_EXTN      11
#define         NOM_PART_PHD_DM         128
#define         NOM_PART_PHD_HF         129
#define         NOM_PART_PHD_AI         130
#define         NOM_PART_RET_CODE       255
#define         NOM_PART_EXT_NOM        256
#define         NOM_PART_PVT            1024
typedef struct TYPE
{
        NomPartition partition;
        OID_Type code;
}       TYPE;
```

```
typedef struct AVA_Type
{
        OID_Type attribute_id;
        Any attribute_value;
}    AVA_Type;

typedef struct AttributeList
{
        intu16 count;
        intu16 length;
        AVA_Type value[1];              /* Первый элемент массива */
}       AttributeList;

typedef struct AttributeIdList
{
        intu16 count;
        intu16 length;
        OID_Type value[1];              /* Первый элемент массива */
} AttributeIdList;

typedef intu32 FLOAT_Type;

typedef intul6 SFLOAT_Type;

typedef intu32 RelativeTime;

typedef struct HighResRelativeTime
{
        intu8 value[8];
}               HighResRelativeTime;

typedef struct AbsoluteTimeAdjust
{
        intu8 value[6];
}               AbsoluteTimeAdjust;

typedef struct AbsoluteTime
{
        intu8 century;
        intu8 year;
        intu8 month;
        intu8 day;
        intu8 hour;
        intu8 minute;
        intu8 second;
        intu8 sec_fractions;
}       AbsoluteTime;
typedef struct BaseOffsetTime
        intu32 bo_seconds;
        intu16 bo_fraction;
        inti16  bo_time_offset;
} BaseOffsetTime;

typedef intu16 OperationalState;
#define         OS_DISABLED             0
#define         OS_ENABLED              1
#define         OS_NOT_AVAILABLE        2

typedef struct octet_string
{
        intu16 length;
        intu8 value[1];                 /* Первый элемент массива */
}       octet_string;
```

```
typedef struct SystemModel
{
        octet_string manufacturer;
        octet_string model_number;
}               SystemModel;

typedef struct ProdSpecEntry
{
        intu16 spec_type;
#define         UNSPECIFIED             0
#define         SERIAL_NUMBER           1
#define         PART_NUMBER             2
#define         HW_REVISION             3
#define         SW_REVISION             4
#define         FW_REVISION             5
#define         PROTOCOL_REVISION       6
#define         PROD_SPEC_GMDN          7
        PrivateOid component_id;
        octet_string prod_spec;
}               ProdSpecEntry;

typedef struct ProductionSpec
{
        intu16 count;
        intu16 length;
        ProdSpecEntry value[1];                 /* Первый элемент массива */
}               ProductionSpec;

typedef intu16 PowerStatus;
#define         ON_MAINS        0x8000
#define         ON_BATTERY              0x4000
#define         CHARGING_FULL           0x0080
#define         CHARGING_TRICKLE                0x0040
#define         CHARGING_OFF    0x0020

typedef struct BatMeasure
{
        FLOAT_Type value;
        OID_Type unit;
}       BatMeasure;

typedef intu16 MeasurementStatus;
#define         MS_INVALID              0x8000
#define         MS_QUESTIONABLE         0x4000
#define         MS_NOT_AVAILABLE        0x2000
#define         MS_CALIBRATION_ONGOING  0x1000
#define         MS_TEST_DATA            0x0800
#define         MS_DEMO_DATA            0x0400
#define         MS_VALIDATED_DATA       0x0080
#define         MS_EARLY_INDICATION     0x0040
#define         MS_MSMT_ONGOING         0x0020

typedef struct NuObsValue
{
        OID_Type metric_id;
        MeasurementStatus state;
        OID_Type unit_code;
        FLOAT_Type value;
}               NuObsValue;

typedef struct NuObsValueCmp
```

```
{
        intu16 count;
        intu16 length;
        NuObsValue value[1];                    /* Первый элемент массива */
}               NuObsValueCmp;

typedef struct SampleType
{
        intu8 sample_size;
        intu8 significant_bits;
}               SampleType;
#define SAMPLE_TYPE_SIGNIFICANT_BITS_SIGNED_SAMPLES 255

typedef intu16 SaFlags;
#define         SMOOTH_CURVE            0x8000
#define         DELAYED_CURVE           0x4000
#define         STATIC_SCALE            0x2000
#define         SA_EXT_VAL_RANGE        0x1000

typedef struct SaSpec
{
        intu16 array_size;
        SampleType sample_type;
        SaFlags flags;
}               SaSpec;

typedef struct ScaleRangeSpec8
{
        FLOAT_Type lower_absolute_value;
        FLOAT_Type upper_absolute_value;
        intu8 lower_scaled_value;
        intu8 upper_scaled_value;
}               ScaleRangeSpec8;
typedef struct ScaleRangeSpec16
{
        FLOAT_Type lower_absolute_value;
        FLOAT_Type upper_absolute_value;
        intu16 lower_scaled_value;
        intu16 upper_scaled_value;
}               ScaleRangeSpec16;

typedef struct ScaleRangeSpec32
{
        FLOAT_Type lower_absolute_value;
        FLOAT_Type upper_absolute_value;
        intu32 lower_scaled_value;
        intu32 upper_scaled_value;
}         ScaleRangeSpec32;

typedef struct EnumVal
{
        intu16 choice;
        intu16 length;
#define         OBJ_ID_CHOSEN           0x0001
#define         TEXT_STRING_CHOSEN      0x0002
#define         BIT_STR_CHOSEN          0x0010
        union
        {
                OID_Type enum_obj_id;
                octet_string enum_text_string;
                intu32 enum_bit_str;            // BITS-32
        } u;
}   EnumVal;
```

```
typedef struct EnumObsValue
{
        OID_Type metric_id;
        MeasurementStatus state;
        EnumVal value;
}         EnumObsValue;

typedef struct AttrValMapEntry
{
        OID_Type attribute_id;
        intu16 attribute_len;
}         AttrValMapEntry;

typedef struct AttrValMap
{
        intu16 count;
        intu16 length;
        AttrValMapEntry value[1];                    /* Первый элемент массива */
}         AttrValMap;

typedef struct HandleAttrValMapEntry
{
        HANDLE obj_handle;
        AttrValMap attr_val_map;
}         HandleAttrValMapEntry;
typedef intu16 ConfirmMode;
#define         UNCONFIRMED         0x0000
#define         CONFIRMED           0x0001

typedef struct HandleAttrValMap
{
        intu16 count;
        intu16 length;
        HandleAttrValMapEntry value[1];              /* Первый элемент массива */
}         HandleAttrValMap;

typedef intu16 StoSampleAlg;
#define         ST_ALG_NOS                          0x0000
#define         ST_ALG_MOVING_AVERAGE               0x0001
#define         ST_ALG_RECURSIVE                    0x0002
#define         ST_ALG_MIN_PICK                     0x0003
#define         ST_ALG_MAX_PICK                     0x0004
#define         ST_ALG_MEDIAN                       0x0005
#define         ST_ALG_TRENDED                      0x0200
#define         ST_ALG_NO_DOWNSAMPLING              0x0400

typedef struct SetTimeInvoke
{
        AbsoluteTime date_time;
        FLOAT_Type accuracy;
}         SetTimeInvoke;

typedef struct SetBOTimeInvoke
{
        BaseOffsetTime date_time;
}         SetBOTimeInvoke;

typedef struct SegmIdList
{
        intu16 count;
        intu16 length;
        InstNumber value[1];                                  /* Первый элемент массива */
}         SegmIdList;
```

```
typedef struct AbsTimeRange
{
        AbsoluteTime    from_time;
        AbsoluteTime    to_time;
}         AbsTimeRange;

typedef struct BOTimeRange
{
        BaseOffsetTime          from_time;
        BaseOffsetTime          to_time;
}         BOTimeRange;

typedef struct SegmentInfo
{
        InstNumber              seg_inst_no;
        AttributeList           seg_info;
}         SegmentInfo;
typedef struct SegmentInfoList
{
        intu16 count;
        intu16 length;
        SegmentInfo value[1];                           /* Первый элемент массива */
}         SegmentInfoList;

typedef struct SegmSelection
{
        intu16 choice;
        intu16 length;
#define         ALL_SEGMENTS_CHOSEN             0x0001
#define         SEGM_ID_LIST_CHOSEN             0x0002
#define         ABS_TIME_RANGE_CHOSEN           0x0003
#define         BO_TIME_RANGE_CHOSEN            0x0004
        union
        {
          intu16 all_segments;
          SegmIdList segm_id_list;
          AbsTimeRange abs_time_range;
          BOTimeRange bo_time_range;
        } u;
}         SegmSelection;

typedef intu16 PMStoreCapab;
#define         PMSC_VAR_NO_OF_SEGM             0x8000
#define         PMSC_SEGM_ID_LIST_SELECT        0x1000
#define         PMSC_EPI_SEG_ENTRIES            0x0800
#define         PMSC_PERI_SEG_ENTRIES           0x0400
#define         PMSC_ABS_TIME_SELECT            0x0200
#define         PMSC_CLEAR_SEGM_BY_LIST_SUP     0x0100
#define         PMSC_CLEAR_SEGM_BY_TIME_SUP     0x0080
#define         PMSC_CLEAR_SEGM_REMOVE          0x0040
#define         PMSC_CLEAR_SEGM_ALL_SUP         0x0020
#define         PMSC_MULTI_PERSON               0x0008
#define         PMSC_GET_SEGM_INFO_SUP          0x0004
#define         PMSC_GET_SEGM_ID_LIST_SUP       0x0002

typedef intu16 SegmEntryHeader;
#define         SEG_ELEM_HDR_ABSOLUTE_TIME              0x8000
#define         SEG_ELEM_HDR_RELATIVE_TIME              0x4000
#define         SEG_ELEM_HDR_HIRES_RELATIVE_TIME        0x2000
#define         SEG_ELEM_HDR_BO_TIME                    0x1000
```

```
typedef struct SegmEntryElem
{
        OID_Type        class_id;
        TYPE            metric_type;
        HANDLE          handle;
        AttrValMap      attr_val_map;
}       SegmEntryElem;

typedef struct SegmEntryElemList
{
        intu16 count;
        intu16 length;
        SegmEntryElem value[1];                 /* Первый элемент массива */
}       SegmEntryElemList;

typedef struct PmSegmentEntryMap
{
        SegmEntryHeader   segm_entry_header;
        SegmEntryElemList   segm_entry_elem_list;
}       PmSegmentEntryMap;

typedef struct SegmElemStaticAttrEntry
{
        OID_Type        class_id;
        TYPE            metric_type;
        AttributeList   attribute_list;
}       SegmElemStaticAttrEntry;

typedef struct PmSegmElemStaticAttrList
{
        intu16 count;
        intu16 length;
        SegmElemStaticAttrEntry value[1];       /* Первый элемент массива */
}       PmSegmElemStaticAttrList;

typedef struct TrigSegmDataXferReq
{
        InstNumber      seg_inst_no;
}       TrigSegmDataXferReq;

typedef intu16 TrigSegmXferRsp;
#define  TSXR_SUCCESSFUL                0
#define  TSXR_FAIL_NO_SUCH_SEGMENT      1
#define  TSXR_FAIL_SEGM_TRY_LATER       2
#define  TSXR_FAIL_SEGM_EMPTY           3
#define  TSXR_FAIL_OTHER                512

typedef struct TrigSegmDataXferRsp
{
        InstNumber      seg_inst_no;
        TrigSegmXferRsp     trig_segm_xfer_rsp;
}       TrigSegmDataXferRsp;
typedef intu16 SegmEvtStatus;
#define         SEVTSTA_FIRST_ENTRY             0x8000
#define         SEVTSTA_LAST_ENTRY              0x4000
#define         SEVTSTA_AGENT_ABORT             0x0800
#define         SEVTSTA_MANAGER_CONFIRM         0x0080
#define         SEVTSTA_MANAGER_ABORT           0x0008
```

```
typedef struct SegmDataEventDescr
{
        InstNumber              segm_instance;
        intu32                  segm_evt_entry_index;
        intu32                  segm_evt_entry_count;
        SegmEvtStatus           segm_evt_status;
}         SegmDataEventDescr;

typedef struct SegmentDataEvent
{
        SegmDataEventDescr              segm_data_event_descr;
        octet_string                    segm_data_event_entries;
}         SegmentDataEvent;

typedef struct SegmentDataResult
{
        SegmDataEventDescr              segm_data_event_descr;
}         SegmentDataResult;

typedef intu16 SegmStatType;
#define  SEGM_STAT_TYPE_UNDEFINED          0
#define  SEGM_STAT_TYPE_MINIMUM            1
#define  SEGM_STAT_TYPE_MAXIMUM            2
#define  SEGM_STAT_TYPE_AVERAGE            3

typedef struct SegmentStatisticEntry
{
        SegmStatType            segm_stat_type;
        octet_string            segm_stat_entry;
}         SegmentStatisticEntry;
typedef struct SegmentStatistics
{
        intu16 count;
        intu16 length;
        SegmentStatisticEntry value[1];                 /* Первый элемент массива */
}         SegmentStatistics;
typedef struct ObservationScan
{
        HANDLE obj_handle;
        AttributeList attributes;
}         ObservationScan;
typedef OID_Type TimeProtocolId;
typedef intu32    AssociationVersion;
#define    ASSOC_VERSION1                         0x80000000
typedef intu32    ProtocolVersion;
#define                 PROTOCOL_VERSION1         0x80000000
#define                 PROTOCOL_VERSION2         0x40000000
#define                 PROTOCOL_VERSION3         0x20000000
typedef intu16    EncodingRules;
#define           MDER          0x8000
#define           XER           0x4000
#define           PER           0x2000
typedef struct UUID_Ident
{
     intu8 value[16];
}         UUID_Ident;
typedef intu16 DataProtoId;
#define                 DATA_PROTO_ID_20601             20601
#define                 DATA_PROTO_ID_EXTERNAL          65535
```

```
typedef struct DataProto
{
        DataProtoId data_proto_id;
        Any data_proto_info;
}         DataProto;

typedef struct DataProtoList
{
        intu16 count;
        intu16 length;
        DataProto value[1];                              /* Первый элемент массива */
}         DataProtoList;

typedef struct AARQ_apdu
{
        AssociationVersion assoc_version;
        DataProtoList data_proto_list;
}               AARQ_apdu;

typedef intu16 Associate_result;
#define         ACCEPTED                                 0
#define         REJECTED_PERMANENT                       1
#define         REJECTED_TRANSIENT                       2
#define         ACCEPTED_UNKNOWN_CONFIG                  3
#define         REJECTED_NO_COMMON_PROTOCOL              4
#define         REJECTED_NO_COMMON_PARAMETER             5
#define         REJECTED_UNKNOWN                         6
#define         REJECTED_UNAUTHORIZED                    7
#define         REJECTED_UNSUPPORTED_ASSOC_VERSION       8

typedef struct AARE_apdu
{
        Associate_result result;
        DataProto selected_data_proto;
}               AARE_apdu;

typedef intu16 Release_request_reason;
#define       RELEASE_REQUEST_REASON_NORMAL                          0
#define       RELEASE_REQUEST_REASON_NO_MORE_CONFIGURATIONS          1
#define       RELEASE_REQUEST_REASON_CONFIGURATION_CHANGED           2

typedef struct RLRQ_apdu
{
        Release_request_reason reason;
}         RLRQ_apdu;

typedef intu16 Release_response_reason;
#define       RELEASE_RESPONSE_REASON_NORMAL                         0

typedef struct RLRE_apdu
{
        Release_response_reason reason;
}         RLRE_apdu;

typedef intu16    Abort_reason;
#define                 ABORT_REASON_UNDEFINED                       0
#define                 ABORT_REASON_BUFFER_OVERFLOW                 1
#define                 ABORT_REASON_RESPONSE_TIMEOUT                2
#define                 ABORT_REASON_CONFIGURATION_TIMEOUT           3
```

```
typedef struct ABRT_apdu
{
      Abort_reason reason;
}     ABRT_apdu;

typedef octet_string PRST_apdu;

typedef intu16 InvokeIDType;

typedef struct EventReportArgumentSimple
{
          HANDLE obj_handle;
          RelativeTime event_time;
          OID_Type event_type;
          Any event_info;
}           EventReportArgumentSimple;

typedef struct GetArgumentSimple
{
          HANDLE obj_handle;
          AttributeIdList attribute_id_list;
}           GetArgumentSimple;

typedef intu16 ModifyOperator;
#define         REPLACE                         0
#define         ADD_VALUES                      1
#define         REMOVE_VALUES                   2
#define         SET_TO_DEFAULT                  3

typedef struct AttributeModEntry
{
          ModifyOperator modify_operator;
          AVA_Type attribute;
}           AttributeModEntry;

typedef struct ModificationList
{
          intu16 count;
          intu16 length;
          AttributeModEntry value[1];                 /* Первый элемент массива */
}                   ModificationList;

typedef struct SetArgumentSimple
{
          HANDLE obj_handle;
          ModificationList modification_list;
}           SetArgumentSimple;

typedef struct ActionArgumentSimple
{
          HANDLE obj_handle;
          OID_Type action_type;
          Any action_info_args;
}           ActionArgumentSimple;

typedef struct EventReportResultSimple
{
          HANDLE obj_handle;
          RelativeTime currentTime;
          OID_Type event_type;
          Any event_reply_info;
}           EventReportResultSimple;
```

```
typedef struct GetResultSimple
{
        HANDLE obj_handle;
        AttributeList attribute_list;
}         GetResultSimple;

typedef struct TypeVer
{
        OID_Type type;
        intu16 version;
}         TypeVer;

typedef struct TypeVerList
{
        intu16 count;
        intu16 length;
        TypeVer value[1];                       /* Первый элемент массива */
}         TypeVerList;

typedef struct SetResultSimple
{
        HANDLE obj_handle;
        AttributeList attribute_list;
}         SetResultSimple;

typedef struct ActionResultSimple
{
        HANDLE obj_handle;
        OID_Type action_type;
        Any action_info_args;
}         ActionResultSimple;

typedef intu16 ERROR;
#define         NO_SUCH_OBJECT_INSTANCE         1
#define         NO_SUCH_ACTION                  9
#define         INVALID_OBJECT_INSTANCE         17
#define         PROTOCOL_VIOLATION              23
#define         NOT_ALLOWED_BY_OBJECT           24
#define         ACTION_TIMED_OUT                25
#define         ACTION_ABORTED                  26

typedef struct ErrorResult
{
        ERROR error_value;
        Any parameter;
}         ErrorResult;
typedef intu16 RorjProblem;
#define  UNRECOGNIZED_APDU                      0
#define  BADLY_STRUCTURED_APDU                  2
#define  UNRECOGNIZED_OPERATION                 101
#define  RESOURCE_LIMITATION                    103
#define  UNEXPECTED_ERROR                       303

typedef struct RejectResult
{
    RorjProblem problem;
}   RejectResult;

typedef struct DATA_apdu
{
        InvokeIDType invoke_id;
        struct
```

```
        {
          intu16 choice;
          intu16 length;
#define       ROIV_CMIP_EVENT_REPORT_CHOSEN                          0x0100
#define       ROIV_CMIP_CONFIRMED_EVENT_REPORT_CHOSEN                0x0101
#define       ROIV_CMIP_GET_CHOSEN                                   0x0103
#define       ROIV_CMIP_SET_CHOSEN                                   0x0104
#define       ROIV_CMIP_CONFIRMED_SET_CHOSEN                         0x0105
#define       ROIV_CMIP_ACTION_CHOSEN                                0x0106
#define       ROIV_CMIP_CONFIRMED_ACTION_CHOSEN                      0x0107
#define       RORS_CMIP_CONFIRMED_EVENT_REPORT_CHOSEN                0x0201
#define       RORS_CMIP_GET_CHOSEN                                   0x0203
#define       RORS_CMIP_CONFIRMED_SET_CHOSEN                         0x0205
#define       RORS_CMIP_CONFIRMED_ACTION_CHOSEN                      0x0207
#define       ROER_CHOSEN                                            0x0300
#define       RORJ_CHOSEN                                            0x0400
        union
        {
          EventReportArgumentSimple roiv_cmipEventReport;
          EventReportArgumentSimple roiv_cmipConfirmedEventReport;
          GetArgumentSimple roiv_cmipGet;
          SetArgumentSimple roiv_cmipSet;
          SetArgumentSimple roiv_cmipConfirmedSet;
          ActionArgumentSimple roiv_cmipAction;
          ActionArgumentSimple roiv_cmipConfirmedAction;
          EventReportResultSimple rors_cmipConfirmedEventReport;
          GetResultSimple rors_cmipGet;
          SetResultSimple rors_cmipConfirmedSet;
          ActionResultSimple rors_cmipConfirmedAction;
          ErrorResult roer;
          RejectResult rorj;
        } u;
  } choice;
} DATA_apdu;

typedef struct APDU
{
        intu16 choice;
        intu16 length;
#define       AARQ_CHOSEN          0xE200
#define       AARE_CHOSEN          0xE300
#define       RLRQ_CHOSEN          0xE400
#define       RLRE_CHOSEN          0xE500
#define       ABRT_CHOSEN          0xE600
#define       PRST_CHOSEN          0xE700
        union
        {
        AARQ_apdu aarq;
        AARE_apdu aare;
        RLRQ_apdu rlrq;
        RLRE_apdu rlre;
        ABRT_apdu abrt;
        PRST_apdu prst;
        } u;
} APDU;

typedef intu32 NomenclatureVersion;
#define       NOM_VERSION1                  0x80000000

typedef intu32 FunctionalUnits;
#define       FUN_UNITS_UNIDIRECTIONAL            0x80000000
#define       FUN_UNITS_HAVETESTCAP               0x40000000
#define       FUN_UNITS_CREATETESTASSOC           0x20000000
```

```
typedef intu32 SystemType;
#define     SYS_TYPE_MANAGER                0x80000000
#define     SYS_TYPE_AGENT                  0x00800000

typedef intu16 ConfigId;
#define     MANAGER_CONFIG_RESPONSE                     0x0000
#define     STANDARD_CONFIG_START                       0x0001
#define     STANDARD_CONFIG_END                         0x3FFF
#define     EXTENDED_CONFIG_START                       0x4000
#define     EXTENDED_CONFIG_END                         0x7FFF
#define     RESERVED_START                              0x8000
#define     RESERVED_END                                0xFFFF

typedef intu16 DataReqModeFlags;
#define     DATA_REQ_SUPP_STOP                          0x8000
#define     DATA_REQ_SUPP_SCOPE_ALL                     0x0800
#define     DATA_REQ_SUPP_SCOPE_CLASS                   0x0400
#define     DATA_REQ_SUPP_SCOPE_HANDLE                  0x0200
#define     DATA_REQ_SUPP_MODE_SINGLE_RSP               0x0080
#define     DATA_REQ_SUPP_MODE_TIME_PERIOD              0x0040
#define     DATA_REQ_SUPP_MODE_TIME_NO_LIMIT            0x0020
#define     DATA_REQ_SUPP_PERSON_ID                     0x0010
#define     DATA_REQ_SUPP_INIT_AGENT                    0x0001

typedef struct DataReqModeCapab
{
        DataReqModeFlags data_req_mode_flags;
        intu8 data_req_init_agent_count;
        intu8 data_req_init_manager_count;
}       DataReqModeCapab;

typedef struct PhdAssociationInformation
{
        ProtocolVersion protocolVersion;
        EncodingRules encodingRules;
        NomenclatureVersion nomenclatureVersion;
        FunctionalUnits functionalUnits;
        SystemType systemType;
        octet_string_system_id;
        intu16 dev_config_id;
        DataReqModeCapab data_req_mode_capab;
        AttributeList optionList;
}       PhdAssociationInformation;

typedef struct ManufSpecAssociationInformation
{
        UUID_Ident data_proto_id_ext;
        Any data_proto_info_ext;
}       ManufSpecAssociationInformation;

typedef intu16 MdsTimeCapState;
#define     MDS_TIME_CAPAB_REAL_TIME_CLOCK              0x8000
#define     MDS_TIME_CAPAB_SET_CLOCK                    0x4000
#define     MDS_TIME_CAPAB_RELATIVE_TIME                0x2000
#define     MDS_TIME_CAPAB_HIGH_RES_RELATIVE_TIME       0x1000
#define     MDS_TIME_CAPAB_SYNC_ABS_TIME                0x0800
#define     MDS_TIME_CAPAB_SYNC_REL_TIME                0x0400
#define     MDS_TIME_CAPAB_SYNC_HI_RES_RELATIVE_TIME    0x0200
#define     MDS_TIME_CAPAB_BO_TIME                      0x0100
#define     MDS_TIME_STATE_ABS_TIME_SYNCED              0x0080
#define     MDS_TIME_STATE_REL_TIME_SYNCED              0x0040
```

```
#define         MDS_TIME_STATE_HI_RES_RELATIVE_TIME_SYNCED      0x0020
#define         MDS_TIME_MGR_SET_TIME                           0x0010
#define         MDS_TIME_CAPAB_SYNC_BO_TIME                     0x0008
#define         MDS_TIME_STATE_BO_TIME_SYNCED                   0x0004
#define         MDS_TIME_STATE_BO_TIME_UTC_ALIGNED              0x0002
#define         MDS_TIME_DST_RULES_ENABLED                      0x0001

typedef struct MdsTimeInfo
{
        MdsTimeCapState mds_time_cap_state;
        TimeProtocolId time_sync_protocol;
        RelativeTime time_sync_accuracy;
        intu16 time_resolution_abs_time;
        intu16 time_resolution_rel_time;
        intu32 time_resolution_high_res_time;
}       MdsTimeInfo;

typedef intu8 AuthBody;
#define         AUTH_BODY_EMPTY                 0
#define         AUTH_BODY_IEEE_11073            1
#define         AUTH_BODY_CONTINUA              2
#define         AUTH_BODY_EXPERIMENTAL          254
#define         AUTH_BODY_RESERVED              255

typedef intu8 AuthBodyStrucType;

typedef struct AuthBodyAndStrucType
{
AuthBody auth_body;
        AuthBodyStrucType auth_body_struc_type;
}       AuthBodyAndStrucType;

typedef struct RegCertData
{
        AuthBodyAndStrucType auth_body_and_struc_type;
        Any auth_body_data;
}       RegCertData;

typedef struct RegCertDataList
{
        intu16 count;
        intu16 length;
        RegCertData value[1];                   /* Первый элемент массива */
}               RegCertDataList;

typedef octet_string EnumPrintableString;

typedef intu16 PersonId;
#define         UNKNOWN_PERSON_ID               0xFFFF

typedef intu16 MetricSpecSmall;
#define     MSS_AVAIL_INTERMITTENT              0x8000
#define     MSS_AVAIL_STORED_DATA               0x4000
#define     MSS_UPD_APERIODIC                   0x2000
#define     MSS_MSMT_APERIODIC                  0x1000
#define     MSS_MSMT_PHYS_EV_ID                 0x0800
#define     MSS_MSMT_BTB_METRIC                 0x0400
#define     MSS_ACC_MANAGER_INITIATED           0x0080
#define     MSS_ACC_AGENT_INITIATED             0x0040
#define     MSS_CAT_MANUAL                      0x0008
#define     MSS_CAT_SETTING                     0x0004
#define     MSS_CAT_CALCULATION                 0x0002
```

```
typedef struct MetricStructureSmall
{
intu8 ms_struct;
#define       MS_STRUCT_SIMPLE                  0
#define       MS_STRUCT_COMPOUND                      1
#define       MS_STRUCT_RESERVED                      2
#define       MS_STRUCT_COMPOUND_FIX                          3
        intu8 ms_comp_no;
}    MetricStructureSmall;

typedef struct MetricIdList
{
        intu16 count;
        intu16 length;
        OID_Type value[1];                              /* Первый элемент массива */
}         MetricIdList;

typedef struct SupplementalTypeList
{
        intu16 count;
        intu16 length;
        TYPE value[1];                                  /* Первый элемент массива */
}         SupplementalTypeList;

typedef struct ObservationScanList
{
        intu16 count;
        intu16 length;
        ObservationScan value[1];                       /* Первый элемент массива */
}         ObservationScanList;

typedef struct ScanReportPerVar
{
        intu16 person_id;
        ObservationScanList obs_scan_var;
}         ScanReportPerVar;

typedef struct ScanReportPerVarList
{
        intu16 count;
        intu16 length;
        ScanReportPerVar value[1];                      /* Первый элемент массива */
}         ScanReportPerVarList;

typedef intu16 DataReqId;
#define       DATA_REQ_ID_MANAGER_INITIATED_MIN                 0x0000
#define       DATA_REQ_ID_MANAGER_INITIATED_MAX                 0xEFFF
#define       DATA_REQ_ID_AGENT_INITIATED_CONFIRMED             0xF000
#define       DATA_REQ_ID_AGENT_INITIATED_UNCONFIRMED           0xF001

typedef struct ScanReportInfoMPVar
{
        DataReqId data_req_id;
        intu16 scan_report_no;
        ScanReportPerVarList scan_per_var;
}         ScanReportInfoMPVar;

typedef struct ObservationScanFixed
{
        HANDLE obj_handle;
        octet_string obs_val_data;
}         ObservationScanFixed;
```

```
typedef struct ObservationScanFixedList
{
        intu16 count;
        intu16 length;
        ObservationScanFixed value[1];                  /* Первый элемент массива */
}       ObservationScanFixedList;

typedef struct ScanReportPerFixed
{
        intu16 person_id;
        ObservationScanFixedList obs_scan_fix;
}       ScanReportPerFixed;

typedef struct ScanReportPerFixedList
{
        intu16 count;
        intu16 length;
        ScanReportPerFixed value[1];                    /* Первый элемент массива */
}       ScanReportPerFixedList;

typedef struct ScanReportInfoMPFixed
{
        DataReqId data_req_id;
        intu16 scan_report_no;
        ScanReportPerFixedList scan_per_fixed;
}       ScanReportInfoMPFixed;

typedef struct ScanReportInfoVar
{
        DataReqId data_req_id;
        intu16 scan_report_no;
        ObservationScanList obs_scan_var;
}       ScanReportInfoVar;

typedef struct ScanReportInfoFixed
{
        DataReqId data_req_id;
        intu16 scan_report_no;
        ObservationScanFixedList obs_scan_fixed;
}       ScanReportInfoFixed;

typedef octet_string ObservationScanGrouped;

typedef struct ScanReportInfoGroupedList
{
        intu16 count;
        intu16 length;
        ObservationScanGrouped value[1];                /* Первый элемент массива */
}       ScanReportInfoGroupedList;

typedef struct ScanReportInfoGrouped
{
        DataReqId data_req_id;
        intu16 scan_report_no;
        ScanReportInfoGroupedList obs_scan_grouped;
}       ScanReportInfoGrouped;

typedef struct ScanReportPerGrouped
{
        PersonId person_id;
        ObservationScanGrouped obs_scan_grouped;
}       ScanReportPerGrouped;
```

```
typedef struct ScanReportPerGroupedList
{
        intu16 count;
        intu16 length;
        ScanReportPerGrouped value[1];                  /* Первый элемент массива */
}         ScanReportPerGroupedList;

typedef struct ScanReportInfoMPGrouped
{
        DataReqId data_req_id;
        intu16 scan_report_no;
        ScanReportPerGroupedList scan_per_grouped;
}         ScanReportInfoMPgrouped;

typedef struct ConfigObject
{
        OID_Type obj_class;
        HANDLE obj_handle;
        AttributeList attributes;
}    ConfigObject;

typedef struct ConfigObjectList
{
        intu16 count;
        intu16 length;
        ConfigObject value[1];                          /* Первый элемент массива */
}         ConfigObjectList;

typedef struct ConfigReport
{
        ConfigId config_report_id;
        ConfigObjectList config_obj_list;
}         ConfigReport;

typedef intu16 ConfigResult;
#define       ACCEPTED_CONFIG                   0x0000
#define       UNSUPPORTED_CONFIG                    0x0001
#define       STANDARD_CONFIG_UNKNOWN                       0x0002

typedef struct ConfigReportRsp
{
        ConfigId config_report_id;
        ConfigResult config_result;
}         ConfigReportRsp;

typedef intu16 DataReqMode;
#define       DATA_REQ_START_STOP                   0x8000
#define       DATA_REQ_CONTINUATION                 0x4000
#define       DATA_REQ_SCOPE_ALL                    0x0800
#define       DATA_REQ_SCOPE_TYPE                   0x0400
#define       DATA_REQ_SCOPE_HANDLE                 0x0200
#define       DATA_REQ_MODE_SINGLE_RSP              0x0080
#define       DATA_REQ_MODE_TIME_PERIOD             0x0040
#define       DATA_REQ_MODE_TIME_NO_LIMIT           0x0020
#define       DATA_REQ_MODE_DATA_REQ_PERSON_ID      0x0008

typedef struct HANDLEList
{
        intu16 count;
        intu16 length;
        HANDLE value[1];                                /* Первый элемент массива */
}         HANDLEList;
```

```
typedef struct DataRequest
{
        DataReqId data_req_id;
        DataReqMode data_req_mode;
        RelativeTime data_req_time;
        intu16 DataRequest_data_req_person_id;
        OID_Type data_req_class;
        HANDLEList data_req_obj_handle_list;
}       DataRequest;

typedef intu16 DataReqResult;
#define     DATA_REQ_RESULT_NO_ERROR                          0
#define     DATA_REQ_RESULT_UNSPECIFIC_ERROR                  1
#define     DATA_REQ_RESULT_NO_STOP_SUPPORT                   2
#define     DATA_REQ_RESULT_NO_SCOPE_ALL_SUPPORT              3
#define     DATA_REQ_RESULT_NO_SCOPE_CLASS_SUPPORT            4
#define     DATA_REQ_RESULT_NO_SCOPE_HANDLE_SUPPORT           5
#define     DATA_REQ_RESULT_NO_MODE_SINGLE_RSP_SUPPORT        6
#define     DATA_REQ_RESULT_NO_MODE_TIME_PERIOD_SUPPORT       7
#define     DATA_REQ_RESULT_NO_MODE_TIME_NO_LIMIT_SUPPORT     8
#define     DATA_REQ_RESULT_NO_PERSON_ID_SUPPORT              9
#define     DATA_REQ_RESULT_UNKNOWN_PERSON_ID                 11
#define     DATA_REQ_RESULT_UNKNOWN_CLASS                     12
#define     DATA_REQ_RESULT_UNKNOWN_HANDLE                    13
#define     DATA_REQ_RESULT_UNSUPP_SCOPE                      14
#define     DATA_REQ_RESULT_UNSUPP_MODE                       15
#define     DATA_REQ_RESULT_INIT_MANAGER_OVERFLOW             16
#define     DATA_REQ_RESULT_CONTINUATION_NOT_SUPPORTED        17
#define     DATA_REQ_RESULT_INVALID_REQ_ID                    18

typedef struct DataResponse
{
        RelativeTime rel_time_stamp;
        DataReqResult data_req_result;
        OID_Type event_type;
        Any event_info;
}       DataResponse;

typedef FLOAT_Type SimpleNuObsValue;

typedef struct SimpleNuObsValueCmp
{
        intu16 count;
        intu16 length;
        SimpleNuObsValue value[1];                  /* Первый элемент массива */
}       SimpleNuObsValueCmp;

typedef SFLOAT_Type BasicNuObsValue;

typedef struct BasicNuObsValueCmp
{
        intu16 count;
        intu16 length;
        BasicNuObsValue value[1];                   /* Первый элемент массива */
}       BasicNuObsValueCmp;

#endif /* PHD_TYPES */
```

Приложение Н
(справочное)

Примеры

Н.1 Общие положения

Настоящее приложение содержит примеры двоичных сообщений, которыми обмениваются агент и менеджер.

Н.2 Весы

Н.2.1 Ассоциация

Н.2.1.1 Запрос ассоциации

Агент отправляет менеджеру следующее сообщение. Данный пример предполагает, что агент представляет собой весы с расширенными функциями.

0xE2	0x00			APDU CHOICE Type (AarqApdu)
0x00	0x32			CHOICE.length = 50
0x80	0x00	0x00	0x00	assoc-version
0x00	0x01	0x00	0x2A	data-proto-list.count = 1 | length = 42
0x50	0x79			data-proto-id = 20601
0x00	0x26			data-proto-info length = 38
0x20	0x00	0x00	0x00	protocol-version[21]
0xA0	0x00			encoding-rules = MDER или PER
0x80	0x00	0x00	0x00	nomenclature-version
0x00	0x00	0x00	0x00	functional-units (например, флаг возможности перейти к тестовой ассоциации)
0x00	0x80	0x00	0x00	system-type = sys-type-agent
0x00	0x08			system-id length = 8 и значение
0x88	0x77	0x66	0x55	0x44 0x33 0x22 0x11
0x40	0x00			dev-config-id
0x00	0x81	0x01	0x01	data-req-mode-capab
0x00	0x00	0x00	0x00	option-list.count = 0 | option-list.length = 0

[21] Обратите внимание, что агенту разрешается поддерживать несколько версий протокола и иметь установленные здесь соответствующие биты.

H.2.1.2 Ответ на запрос ассоциации

Менеджер отвечает агенту, что может ассоциироваться, но необходима конфигурация агента.

0xE3	0x00			APDU CHOICE Type (AareApdu)
0x00	0x2C			CHOICE.length = 44
0x00	0x03			result = accepted-unknown-config
0x50	0x79			data-proto-id = 20601
0x00	0x26			data-proto-info length = 38
0x20	0x00	0x00	0x00	protocol-version
0x80	0x00			encoding-rules = MDER
0x80	0x00	0x00	0x00	nomenclature-version
0x00	0x00	0x00	0x00	functional-units
0x80	0x00	0x00	0x00	system-type = sys-type-manager
0x00	0x08			system-id length = 8 и значение
0x11	0x22	0x33	0x44	0x55 0x66 0x77 0x88
0x00	0x00			Ответ менеджера в dev-config-id всегда 0
0x00	0x00	0x00	0x00	Ответ менеджера в data-req-mode-capab всегда 0
0x00	0x00	0x00	0x00	option-list.count = 0 | option-list.length = 0

H.2.2 Обмен информацией о конфигурации

H.2.2.1 Удаленный вызов конфигурации отчета о событии

Агент сообщает менеджеру свою конфигурацию.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x70			CHOICE.length = 112
0x00	0x6E			OCTET STRING.length = 110
0x43	0x21			invoke-id = 0x4321 (начало DataApdu. Кодирование MDER.)
0x01	0x01			CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00	0x68			CHOICE.length = 104
0x00	0x00			obj-handle = 0 (объект MDS)
0xFF	0xFF	0xFF	0xFF	event-time = у агента нет часов относительного времени
0x0D	0x1C			event-type = MDC_NOTI_CONFIG
0x00	0x5E			event-info.length =94 (начало ConfigReport)
0x40	0x00			config-report-id
0x00	0x02			config-obj-list.count = 2, будут объявляться объекты измерений
0x00	0x58			config-obj-list.length = 88
0x00	0x06			obj-class = MDC_MOC_VMO_METRIC_NU
0x00	0x01			obj-handle =1 (→ Первое измерение соответствует весу тела)
0x00	0x04			attributes.count = 4
0x00	0x24			attributes.length = 36
0x09	0x2F			attribute-id = MDC_ATTR_ID_TYPE
0x00	0x04			attribute-value.length = 4
0x00	0x02	0xE1	0x40	MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL
0x0A	0x46			attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00	0x02			attribute-value.length = 2
0xD0	0x40			intermittent, stored data, msmt aperiodic, agent init, measured
0x09	0x96			attribute-id = MDC_ATTR_UNIT_CODE
0x00	0x02			attribute-value.length = 2
0x06	0xC3			MDC_DIM_KILO_G
0x0A	0x55			attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00	0x0C			attribute-value.length = 12
0x00	0x02			AttrValMap.count = 2
0x00	0x08			AttrValMap.length = 8
0x0A	0x56	0x00	0x04	MDC_ATTR_NU_VAL_OBS_SIMP | value length = 4
0x09	0x90	0x00	0x08	MDC_ATTR_TIME_STAMP_ABS | value length = 8
0x00	0x06			obj-class = MDC_MOC_VMO_METRIC_NU
0x00	0x02			obj-handle = 2 (→ Второе измерение соответствует жирам тела)
0x00	0x04			attributes.count = 4
0x00	0x24			attributes.length = 36
0x09	0x2F			attribute-id = MDC_ATTR_ID_TYPE
0x00	0x04			attribute-value.length = 4
0x00	0x02	0xE1	0x4C	MDC_PART_SCADA | MDC_BODY_FAT

0x0A	0x46			attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00	0x02			attribute-value.length = 2
0xD0	0x42			intermittent, stored data, msmt aperiodic, agent init, calculated
0x09	0x96			attribute-id = MDC_ATTR_UNIT_CODE
0x00	0x02			attribute-value.length = 2
0x02	0x20			MDC_DIM_PERCENT
0x0A	0x55			attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00	0x0C			attribute-value.length = 12
0x00	0x02			AttrValMap.count = 2
0x00	0x08			AttrValMap.length = 8
0x0A	0x56	0x00	0x04	MDC_ATTR_NU_VAL_OBS_SIMP, 4
0x09	0x90	0x00	0x08	MDC_ATTR_TIME_STAMP_ABS, 8

H.2.2.2 Удаленный ответ для конфигурации отчета о событии

Менеджер отвечает, что может использовать конфигурацию агента.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x16			CHOICE.length = 22
0x00	0x14			OCTET STRING.length = 20
0x43	0x21			invoke-id = 0x4321 (начало DataApdu. Кодирование MDER.)
0x02	0x01			CHOICE (Remote Operation Response | Confirmed Event Report)
0x00	0x0E			CHOICE.length = 14
0x00	0x00			obj-handle = 0 (объект MDS)
0xTT	0xTT	0xTT	0xTT	currentTime = текущий счетчик менеджера (1/8 миллисекунды)
0x0D	0x1C			event-type = MDC_NOTI_CONFIG
0x00	0x04			event-reply-info.length = 4
0x40	0x00			ConfigReportRsp.config-report-id = 0x4000
0x00	0x00			ConfigReportRsp.config-result = accepted-config

H.2.3 Служба GET для запроса атрибутов MDS

H.2.3.1 Общие положения

Служба GET для запроса атрибутов MDS вызывается в любое время, когда устройство находится в состоянии ассоциации.

H.2.3.2 Запрос GET для получения всех атрибутов MDS

Менеджер запрашивает у агента атрибуты объекта MDS.

0xE7	0x00		APDU CHOICE Type (PrstApdu)
0x00	0x0E		CHOICE.length = 14
0x00	0x0C		OCTET STRING.length = 12
0x34	0x56		invoke-id = 0x3456 (начало DataApdu. Кодирование MDER.)
0x01	0x03		CHOICE (Remote Operation Invoke | Get)
0x00	0x06		CHOICE.length = 6
0x00	0x00		obj-handle = 0 (объект MDS)
0x00	0x00		attribute-id-list.count = 0 (все атрибуты)
0x00	0x00		attribute-id-list.length = 0

H.2.3.3 Передача ответа на запрос всех атрибутов MDS

Агент сообщает менеджеру свои атрибуты. В настоящем примере предполагается, что агент поддерживает AbsoluteTime (абсолютное время), но не поддерживает RelativeTime (относительное время). Кроме того, также передаются некоторые дополнительные поля.

0xE7	0x00		APDU CHOICE Type (PrstApdu)
0x00	0x6E		CHOICE.length = 110
0x00	0x6C		OCTET STRING.length = 108
0x34	0x56		invoke-id = 0x3456 (зеркалировано из запроса) (начало DataApdu. Кодирование MDER.)
0x02	0x03		CHOICE (Remote Operation Response | Get)
0x00	0x66		CHOICE.length = 102
0x00	0x00		obj-handle = 0 (объект MDS)
0x00	0x06		attribute-list.count = 6
0x00	0x60		attribute-list.length = 96
0x0A	0x5A		attribute-id = MDC_ATTR_SYS_TYPE_SPEC_LIST
0x00	0x08		attribute-value.length = 8
0x00	0x01		TypeVerList count = 1
0x00	0x04		TypeVerList length = 4
0x10	0x0F		type = MDC_DEV_SPEC_PROFILE_SCALE

```
0x00  0x01                          version = версия 1 специализации
0x09  0x28                          attribute-id = MDC_ATTR_ID_MODEL
0x00  0x1A                          attribute-value.length = 26
0x00  0x0A  0x54  0x68              string.length = 10 | “TheCompany”
0x65  0x43  0x6F  0x6D
0x70  0x61  0x6E  0x79
0x00  0x0C  0x54  0x68              string.length = 12 | «TheScaleABC\0»
0x65  0x53  0x63  0x61
0x6C  0x65  0x41  0x42
0x43  0x00
0x09  0x84                          attribute-id = MDC_ATTR_SYS_ID
0x00  0x0A                          attribute-value.length = 10
0x00  0x08  0x88  0x77              0x66      0x55      0x44      0x33
0x22  0x11                          octet string length = 8 | EUI-64
0x0A  0x44                          attribute-id = MDC_ATTR_DEV_CONFIG_ID
0x00  0x02                          attribute-value.length = 2
0x40  0x00                          dev-config-id = 16384 (extended-config-start)
0x09  0x2D                          attribute-id = MDC_ATTR_ID_PROD_SPECN
0x00  0x12                          attribute-value.length = 18
0x00  0x01                                    ProductionSpec.count = 1
0x00  0x0E                                    ProductionSpec.length = 14
0x00  0x01                                    ProdSpecEntry.spec-type = 1 (серийный номер)
0x00  0x00                                    ProdSpecEntry.component-id = 0
0x00  0x08  0x44  0x45  string length = 8 | prodSpecEntry.prod-spec = “DE124567”
0x31  0x32  0x34  0x35
0x36  0x37
0x09  0x87                          attribute-id = MDC_ATTR_TIME_ABS
0x00  0x08                          attribute-value.length = 8
0x20  0x07  0x02  0x01              AbsoluteTime = 2007-02-01T12:05:0000
```

H.2.4 Создание отчетов

H.2.4.1 Инициируемая агентом передача результатов измерений

Агент отправляет менеджеру спонтанный отчет о событии, содержащий результаты измерений.

```
0xE7  0x00                          APDU CHOICE Type (PrstApdu)
0x00  0x3A                          CHOICE.length = 58
0x00  0x38                          OCTET STRING.length = 56
0x43  0x21                          invoke-id = 0x4321 (начало DataApdu. Кодирование MDER.)
0x01  0x01                          CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00  0x32                          CHOICE.length = 50
0x00  0x00                          obj-handle = 0 (объект MDS)
0xFF  0xFF  0xFF  0xFF              event-time = у агента нет часов относительного времени
0x0D  0x1D                          event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00  0x28                          event-info.length = 40
0xF0  0x00                          ScanReportInfoFixed.data-req-id = 0xF000
0x00  0x00                          ScanReportInfoFixed.scan-report-no = 0
0x00  0x02                          ScanReportInfoFixed.obs-scan-fixed.count = 2
0x00  0x20                          ScanReportInfoFixed.obs-scan-fixed.length = 32
0x00  0x01                          ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00  0x0C                          ScanReportInfoFixed.obs-scan-fixed.value[0].obs-val-data.length = 12
0xFF  0x00  0x02  0x70              Simple-Nu-Observed-Value = 62.4
0x20  0x07  0x02  0x01              Absolute-Time-Stamp = 2007-02-01T12:10:0000
0x12  0x10  0x00  0x00
0x00  0x02                          ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 2
0x00  0x0C                          ScanReportInfoFixed.obs-scan-fixed.value[1].obs-val-data.length = 12
0xFF  0x00  0x01  0x00              Simple-Nu-Observed-Value = 25.6
0x20  0x07  0x02  0x01              Absolute-Time-Stamp = 2007-02-01T12:10:0000
0x12  0x10  0x00  0x00
```

H.2.4.2 Ответ на инициированную агентом передачу результатов измерений

Менеджер подтверждает получение отчета о событии, отправленного агентом.

```
0xE7  0x00          APDU CHOICE Type (PrstApdu)
0x00  0x12          CHOICE.length = 18
0x00  0x10          OCTET STRING.length = 16
```

0x43	0x21			invoke-id = 0x4321 (начало DataApdu. Кодирование MDER.)
0x02	0x01			CHOICE(Remote Operation Response | Confirmed Event Report)
0x00	0x0A			CHOICE.length = 10
0x00	0x00			obj-handle = 0 (объект MDS)
0xTT	0xTT	0xTT	0xTT	currentTime = текущий счетчик менеджера (1/8 миллисекунды)
0x0D	0x1D			event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00	0x00			event-reply-info.length = 0

H.2.4.3 Удаленный вызов подтверждаемого действия по запросу данных в режиме одиночного ответа

Менеджер запрашивает у агента измерения в режиме одиночного ответа.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x1E			CHOICE.length = 30
0x00	0x1C			OCTET STRING.length = 28
0x76	0x54			invoke-id = 0x7654 (начало DataApdu. Кодирование MDER.)
0x01	0x07			CHOICE(Remote Operation Invoke | Confirmed Action)
0x00	0x16			CHOICE.length = 22
0x00	0x00			obj-handle = 0 (объект MDS)
0x0C	0x1B			action-type = MDC_ACT_DATA_REQUEST
0x00	0x10			action-info-args.length = 16
0x01	0x00			data-req-id = 0x0100
0x84	0x80			data-req-mode = Start | Scope Class | Mode SingleRsp
0x00	0x00	0x00	0x00	data-req-time = не используется в этом режиме
0x00	0x00			data-req-person-id = не используется в этом режиме
0x00	0x06			data-req-class = MDC_MOC_VMO_METRIC_NU
0x00	0x00	0x00	0x00	data-req-obj-handle-list = не используется в этом режиме (count = 0, length = 0)

H.2.4.4 Ответ на удаленный вызов подтверждаемого действия по запросу данных в режиме одиночного ответа

Агент отвечает менеджеру на запрос его измерений.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x40			CHOICE.length = 64
0x00	0x3E			OCTET STRING.length = 62
0x76	0x54			invoke-id = 0x7654 (начало DataApdu. Кодирование MDER.)
0x02	0x07			CHOICE (Remote Operation Response | Confirmed Action)
0x00	0x38			CHOICE.length = 56
				ActionResultSimple
0x00	0x00			obj-handle = 0 (объект MDS)
0x0C	0x1B			action-type = MDC_ACT_DATA_REQUEST
0x00	0x32			action-info-args.length = 50
				DataResponse
0xFF	0xFF	0xFF	0xFF	rel-time-stamp = у агента нет часов относительного времени
0x00	0x00			data-req-result = 0
0x0D	0x1D			event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00	0x28			event-info.length = 40
0x01	0x00			ScanReportInfoFixed.data-req-id = 0x0100
0x00	0x00			ScanReportInfoFixed.scan-report-no = 0
0x00	0x02			ScanReportInfoFixed.obs-scan-fixed.count = 2
0x00	0x20			ScanReportInfoFixed.obs-scan-fixed.length = 32
0x00	0x01			ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1
0x00	0x0C			ScanReportInfoFixed.obs-scan-fixed.value[0].obs-val-data.length = 12
0xFF	0x00	0x02	0x70	Simple-Nu-Observed-Value = 62.4
0x20	0x07	0x02	0x01	Absolute-Time-Stamp = 2007-02-01T12:10:0000
0x12	0x10	0x00	0x00	
0x00	0x02			ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 2
0x00	0x0C			ScanReportInfoFixed.obs-scan-fixed.value[1].obs-val-data.length = 12
0xFF	0x00	0x01	0x00	Simple-Nu-Observed-Value = 25.6
0x20	0x07	0x02	0x01	Absolute-Time-Stamp = 2007-02-01T12:10:0000
0x12	0x10	0x00	0x00	

H.3 Пульсоксиметр

H.3.1 Общие положения

Следующие транзакции демонстрируют последовательность сообщений при инициированной менеджером передаче результатов измерений, когда менеджер задает значение поля data-req-id равным 1.

H.3.2 Начальные условия

Плетизмографическая волна, моделируемая с помощью объекта RT-SA, обладает следующими свойствами:

- SaSpec::array-size = 5
- SampleType::sample-size = 16
- SampleType::significant-bits = 16
- Sample-Period:160 = 20 мс

Значения атрибутов SaSpec::array-size и Sample-Period подразумевают, что период обновления наблюдаемых значений в объекте RT-SA равен 100 мс.

Числовой объект SpO2 обновляется каждую секунду.

В каждое 10-е сообщение будет добавлено значение SpO2.

Агент отправляет менеджеру девять сообщений, которые приблизительно соответствуют представленному ниже формату, при этом изменяется только информация в каждом сообщении (например, изменяются invoke-id и scan-report-no).

0xE7	0x00			APDU CHOICE Type (DataApdu)
0x00	0x2A			CHOICE.length = 42
0x00	0x28			OCTET STRING.length = 40
0x43	0x21			invoke-id = 0x4321 (начало DataApdu. Кодирование MDER.)
0x01	0x00			CHOICE (Remote Operation Invoke | Event Report)
0x00	0x22			CHOICE.length = 34
0x00	0x00			obj-handle = 0 (объект MDS)
0xFF	0xFF	0xFF	0xFF	event-time = у агента нет часов относительного времени
0x0D	0x1D			event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00	0x18			event-info.length = 24
0x00	0x01			ScanReportInfoFixed.data-req-id = 1
0x00	0x00			ScanReportInfoFixed.scan-report-no = 0
0x00	0x01			ScanReportInfoFixed.obs-scan-fixed.count = 1
0x00	0x0E			ScanReportInfoFixed.obs-scan-fixed.length = 14
0x00	0x01			ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1 PlethWave
0x00	0x0C			ScanReportInfoFixed.obs-scan-fixed.value[0].obs-val-data.length = 12
0x00	0x0A			Simple-Sa-Observed-Value OCTET STRING length = 10
0XSS	0xSS	0xSS	0xSS	Считывания
0XSS	0xSS	0xSS	0xSS	Считывания
0xSS	0xSS			Считывания

H.3.3 Каждое 10-е сообщение

В каждое 10-е сообщение агент добавляет также значение SpO2.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x32			CHOICE.length = 50
0x00	0x30			OCTET STRING.length = 48
0x43	0x2A			invoke-id = 0x432A (начало DataApdu. Кодирование MDER.)
0x01	0x00			CHOICE (Remote Operation Invoke | Event Report)
0x00	0x2A			CHOICE.length = 42
0x00	0x00			obj-handle = 0 (объект MDS)
0xFF	0xFF	0xFF	0xFF	event-time = у агента нет часов относительного времени
0x0D	0x1D			event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00	0x20			event-info.length = 32
0x00	0x01			ScanReportInfoFixed.data-req-id = 1
0x00	0x09			ScanReportInfoFixed.scan-report-no = 9
0x00	0x02			ScanReportInfoFixed.obs-scan-fixed.count = 2
0x00	0x16			ScanReportInfoFixed.obs-scan-fixed.length = 22
0x00	0x01			ScanReportInfoFixed.obs-scan-fixed.value[0].obj-handle = 1 PlethWave
0x00	0x0C			ScanReportInfoFixed.obs-scan-fixed.value[0].obs-val-data.length = 12
0x00	0x0A			Simple-Sa-Observed-Value OCTET STRING length = 10
0XSS	0xSS	0xSS	0xSS	Считывания
0XSS	0xSS	0xSS	0xSS	Считывания
0xSS	0xSS			Считывания
0x00	0x02			ScanReportInfoFixed.obs-scan-fixed.value[1].obj-handle = 2 SpO2
0x00	0x04			ScanReportInfoFixed.obs-scan-fixed.value[1].obs-val-data.length = 4
0xFF	0x00	0x03	0xDF	Simple-Nu-Observed-Value = 99.1

Н.4 Транзакции с объектами PM-store и PM-segment

Н.4.1 Общие положения

Следующие транзакции демонстрируют последовательность сообщений для выполнения задач по использованию объектов PM-store и PM-segment. После того как агент представляет свою конфигурацию и менеджер предположительно принимает такую конфигурацию, менеджер инициирует последовательность операций передачи результатов измерений. Приведен пример, показывающий очистку сегмента PM-segment.

Учтите, что перед тем, как менеджер вызовет действие Get-Segment-Info, он может вызвать службу GET для считывания последних значений динамических атрибутов Store-Usage-Count, Operation-State, Number-Of-Segments или Clear-Timeout. В настоящем примере этот шаг опущен.

Н.4.2 Сообщение о конфигурации

Агент сообщает менеджеру свою запись конфигурации.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0xA0			CHOICE.length = 160
0x00	0x9E			OCTET STRING.length = 158
0x03	0x21			invoke-id = 0x0321 (начало DataApdu. Кодирование MDER.)
0x01	0x01			CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00	0x98			CHOICE.length = 152
0x00	0x00			obj-handle = 0 (объект MDS)
0xFF	0xFF	0xFF	0xFF	event-time = у агента нет часов относительного времени
0x0D	0x1C			event-type = MDC_NOTI_CONFIG
0x00	0x8E			event-info.length =142 (Начало ConfigReport)
0x40	0x10			config-report-id находится в расширенном диапазоне
0x00	0x03			config-obj-list.count = 3, будут «объявляться» объекты измерений
0x00	0x88			config-obj-list.length = 136 байт
0x00	0x06			obj-class = MDC_MOC_VMO_METRIC_NU
0x00	0x01			obj-handle = 1 (→ Первое измерение соответствует SpO_2)
0x00	0x04			attributes.count = 4
0x00	0x24			attributes.length = 36
0x09	0x2F			attribute-id = MDC_ATTR_ID_TYPE
0x00	0x04			attribute-value.length = 4
0x00	0x02	0x4B	0xB8	MDC_PART_SCADA | MDC_PULS_OXIM_SAT_O2
0x0A	0x46			attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00	0x02			attribute-value.length = 2
0x40	0xC0			avail-stored-data, acc-manager-init, acc-agent-init, measured
0x09	0x96			attribute-id = MDC_ATTR_UNIT_CODE
0x00	0x02			attribute-value.length = 2
0x02	0x20			MDC_DIM_PERCENT
0x0A	0x55			attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00	0x0C			attribute-value.length = 12
0x00	0x02			AttrValMap.count = 2
0x00	0x08			AttrValMap.length = 8
0x0A	0x4C	0x00	0x02	MDC_ATTR_NU_VAL_OBS_BASIC | value length = 2
0x09	0x90	0x00	0x08	MDC_ATTR_TIME_STAMP_ABS | value length = 8
0x00	0x06			obj-class = MDC_MOC_VMO_METRIC_NU
0x00	0x0A			obj-handle = 10 (→ Второе измерение соответствует частоте пульса)
0x00	0x04			attributes.count = 4
0x00	0x24			attributes.length = 36
0x09	0x2F			attribute-id = MDC_ATTR_ID_TYPE
0x00	0x04			attribute-value.length = 4
0x00	0x02	0x48	0x1A	MDC_PART_SCADA | MDC_PULS_OXIM_PULS_RATE
0x0A	0x46			attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00	0x02			attribute-value.length = 2
0x40	0xC0			avail-stored-data, acc-manager-init, acc-agent-init, measured
0x09	0x96			attribute-id = MDC_ATTR_UNIT_CODE
0x00	0x02			attribute-value.length = 2
0x0A	0xA0			MDC_DIM_BEAT_PER_MIN
0x0A	0x55			attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00	0x0C			attribute-value.length = 12
0x00	0x02			AttrValMap.count = 2

0x00	0x08			AttrValMap.length = 8
0x0A	0x4C	0x00	0x02	MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x09	0x90	0x00	0x08	MDC_ATTR_TIME_STAMP_ABS, 8
0x00	0x3D			obj-class = MDC_MOC_VMO_PMSTORE
0x00	0x64			obj-handle = 100 (→ PM-store)
0x00	0x06			attributes.count = 6
0x00	0x28			attributes.length = 40 байт
0x0A	0x4D			attribute-id = MDC_ATTR_PM_STORE_CAPAB
0x00	0x02			attribute-value.length = 2
0x84	0x60			pmsc-var-no-of-segm | pmsc-peri-seg-entries | pmsc-clear-segm-remove_| pmsc-clear-segm-all-sup
0x09	0x43			attribute-id = MDC_ATTR_STORE_SAMPLE_ALG
0x00	0x02			attribute-value.length = 2
0x04	0x00			st-alg-no-downsampling
0x09	0x53			attribute-id = MDC_ATTR_OP_STAT
0x00	0x02			attribute-value.length = 2
0x00	0x00			деактивированные записи в настоящее время не добавлены
0x09	0x8D			attribute-id = MDC_ATTR_TIME_PD_SAMP
0x00	0x04			attribute-value.length = 4
0x00	0x00	0x7D	0x00	4 секунды = 32 000 * 125 микросекунд (такт часов относительного времени)
0x09	0x51			attribute-id = MDC_ATTR_NUM_SEG
0x00	0x02			attribute-value.length = 2
0x00	0x01			в настоящее время хранится 1 сегмент PM-segment
0x0A	0x63			attribute-id = MDC_ATTR_CLEAR_TIMEOUT
0x00	0x04			attribute-value.length = 4
0x00	0x02	0x71	0x00	20 секунд = 160 000 * 125 микросекунд (такт часов относительного времени)

H.4.3 Вызов менеджером действия Get-Segment-Info

Далее менеджер выполняет действие Get-Segment-Info для получения информации о сегменте PM-segment агента. В последующем сообщении менеджер должен запросить информацию обо всех сегментах PM-segment, поскольку агент не установил биты pmsc-segm-id-list-select или pmsc-abs-time-select в атрибуте PM-Store-Capab.

0xE7	0x00	APDU CHOICE Type (PrstApdu)
0x00	0x14	CHOICE.length = 18
0x00	0x12	OCTETSTRING.length =16
0x03	0x22	invoke-id = 0x0322 (начало DataApdu. Кодирование MDER.)
0x01	0x07	CHOICE(Remote Operation Invoke | Confirmed Action)
0x00	0x0C	CHOICE.length = 12
0x00	0x64	obj-handle = 100 (объект PM-store)
0x0C	0x0D	action-type = MDC_ACT_SEG_GET_INFO
0x00	0x06	actionInfoArgs.length = 6
0x00	0x01	CHOICE(all-segments)
0x00	0x02	CHOICE.length = 2
0x00	0x00	Должно быть нулевым

H.4.4 Ответ агента на Get-Segment-Info со списком сегментов SegmentInfoList

В этом примере агент хранит один сегмент PM-segment.

0xE7	0x00	APDU CHOICE Type (PrstApdu)
0x00	0x68	CHOICE.length = 104
0x00	0x66	OCTET STRING.length = 102
0x03	0x22	invoke-id = 0x0322 (начало DataApdu. Кодирование MDER.)
0x02	0x07	CHOICE(Remote Operation Response | Confirmed Action)
0x00	0x60	CHOICE.length = 96
0x00	0x64	obj-handle = 100 (объект PM-store)
0x0C	0x0D	action-type = MDC_ACT_SEG_GET_INFO
0x00	0x5A	actionInfoArgs.length = 96
0x00	0x01	SegmentInfoList.count = 1
0x00	0x56	SegmentInfoList.length = 86
0x00	0x01	идентификатор экземпляра сегмента равен 1
0x00	0x05	attributes.count = 5
0x00	0x50	attributes.length = 80 байт
0x0A	0x64	attribute-id #1 = MDC_ATTR_TRANSFER_TIMEOUT

0x00	0x04			attribute-value.length = 4
0x00	0x0E	0xA6	0x00	120 секунд = 120 * 8000
0x09	0x53			attribute-id #2 = MDC_ATTR_OP_STAT
0x00	0x02			attribute-value.length = 2
0x00	0x00			отключено
0x0A	0x4E			attribute-id #3 = MDC_ATTR_PM_SEG_MAP
0x00	0x26			attribute-value.length = 38
0x00	0x00			SegmEntryHeader: метки времени нет
0x00	0x02			SegmEntryElemList: count = 2
0x00	0x20			SegmEntryElemList: length = 32
0x00	0x06			SegmEntryElemList 1.class-id = MDC_MOC_VMO_METRIC_NU
0x00	0x02	0x4B	0xB8	.metric-type = MDC_PART_SCADA |_PULS_OXIM_SAT_O2
0x00	0x01			handle = 1
0x00	0x01			attr-val-map.count = 1
0x00	0x04			attr-val-map.length = 4
0x0A	0x4C	0x00	0x02	MDC_ATTR_NUVAL_OBS_BASIC | length = 2
0x00	0x06			SegmEntryElemList 2.class-id = MDC_MOC_VMO_METRIC_NU
0x00	0x02	0x48	0x1A	.metric-type = MDC_PART_SCADA | PULS_OXIM_PULS_RATE
0x00	0x0A			handle = 10
0x00	0x01			attr-val-map.count = 1
0x00	0x04			attr-val-map.length = 4
0x0A	0x4C	0x00	0x02	MDC_ATTR_NU_VAL_OBS_BASIC | length = 2
0x09	0x92			attribute-id#4 = MDC_ATTR_TIME_START_SEG
0x00	0x08			attribute-value.length = 8
0x20	0x10	0x02	0x11	Feb 11 2010
0x21	0x55	0x15	0x00	9:55:15 PM
0x09	0x8A			attribute-id#5 = MDC_ATTR_TIME_END_SEG
0x00	0x08			attribute-value.length = 8
0x20	0x10	0x02	0x12	Feb 12 2010
0x06	0x15	0x22	0x00	6:15:22 AM

H.4.5 Инициируемая менеджером передача с помощью действия Trig-SegmData-Xfer, запрашивающего данные сегмента 1

Менеджер готов к тому, что агент начнет передачу накопленных результатов измерений. Следующий блок APDU дает агенту команду начать передачу данных с сегмента 1.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x10			CHOICE.length = 16
0x00	0x0E			OCTET STRING.length = 14
0x03	0x23			invoke-id = 0x0323(начало DataApdu. Кодирование MDER.)
0x01	0x07			CHOICE(Remote Operation Invoke | Confirmed Action)
0x00	0x08			CHOICE.length = 8
0x00	0x64			obj-handle = 100 (объект PM-store)
0x0C	0x10			action-type = MDC_ACT_SEG_TRIG_XFER
0x00	0x02			actionInfoArgs.length = 2
0x00	0x01			Идентификатор сегмента, 1

H.4.6 Ответ агента на действие Trig-Seg-Data-Transfer

Следующий ответ агента информирует, что менеджер сгенерировал допустимый запрос на передачу доступного сегмента.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x12			CHOICE.length = 18
0x00	0x10			OCTET STRING.length = 16
0x03	0x23			invoke-id = 0x0323 (начало DataApdu. Кодирование MDER.)
0x02	0x07			CHOICE(Remote Operation Response | Confirmed Action)
0x00	0x0A			CHOICE.length = 10
0x00	0x64			obj-handle = 100 (объект PM-store)
0x0C	0x1C			action-type = MDC_ACT_SEG_TRIG_XFER
0x00	0x04			actionInfoArgs.length = 4
0x00	0x01	0x00	0x00	Segment = 1 | response code: txsr-successful

Н.4.7 Отправка агентом первого блока измерений, хранящегося в сегменте PM-segment, с помощью отчетов Segment-Data-Event

Поскольку менеджер готов получать данные, агент начинает отправку отчетов Segment-Data-Event.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x34			CHOICE.length = 52
0x00	0x32			OCTET STRING.length = 50
0x03	0x24			invoke-id = 0x0324 (начало DataApdu. Кодирование MDER.)
0x01	0x01			CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00	0x2C			CHOICE.length = 44
0x00	0x64			obj-handle = 100 (объект PM-store)
0xZZ	0xZZ	0xZZ	0xZZ	RelativeTime
0x0D	0x21			event-type = MDC_NOTI_SEGMENT_DATA
0x00	0x22			event-info.length = 34
0x00	0x01			SegmDataEventDescr.segm-instance
0x00	0x00			SegmDataEventDescr.segm-evt-entry-index
0x00	0x00			Индекс записи, 0
0x00	0x00			SegmDataEventDescr.segm-evt-entry-count
0x00	0x05			5 записей
0x80	0x00			SegmDataEventDescr.segm-evt-status: первая запись
0x00	0x14			20 байт 5 измерений SpO2 и частоты пульса
0x00	0x62			Измерение № 0: SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту
0x00	0x62			Измерение № 1: SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту
0x00	0x62			Измерение № 2: SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту
0x00	0x62			Измерение № 3: SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту
0x00	0x62			Измерение № 4: SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту

Н.4.8 Подтверждение менеджером получения первого блока

Менеджер отвечает на передачу агентом первого набора результатов измерений.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x1A			CHOICE.length = 26
0x00	0x18			OCTET STRING.length = 24
0x03	0x24			invoke-id = 0x0324 (начало DataApdu. Кодирование MDER.)
0x02	0x01			CHOICE(Remote Operation Response | Confirmed Event Report)
0x00	0x12			CHOICE.length = 18
0x00	0x64			obj-handle = 100 (объект PM-store)
0x0D	0x21			event-type = MDC_NOTI_SEGMENT_DATA
0x00	0x0C			event-info.length = 12
0x00	0x01			SegmDataEventDescr.segm-instance
0x00	0x00			SegmDataEventDescr.segm-evt-entry-index
0x00	0x00			Индекс записи, 0
0x00	0x00			SegmDataEventDescr.segm-evt-entry-count
0x00	0x05			5 записей
0x80	0x80			SegmDataEventDescr.segm-evt-status: Первая запись, подтверждено

Н.4.9 Отправка агентом второго блока измерений PM-segment

Блок APDU имеет ту же структуру, что и первый блок данных, однако в структуре SegmEvtStatus бит sevtsta-first-entry уже не установлен.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x34			CHOICE.length = 52
0x00	0x32			OCTET STRING.length = 50
0x03	0x25			invoke-id = 0x0325 (начало DataApdu. Кодирование MDER.)
0x01	0x01			CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00	0x2C			CHOICE.length = 44
0x00	0x64			obj-handle = 100 (объект PM-store)
0xZZ	0xZZ	0xZZ	0xZZ	RelativeTime
0x0D	0x21			event-type = MDC_NOTI_SEGMENT_DATA

0x00	0x22			event-info.length = 34
0x00	0x01			SegmDataEventDescr.segm-instance
0x00	0x00			SegmDataEventDescr.segm-evt-entry-index
0x00	0x05			Индекс записи, 5
0x00	0x00			SegmDataEventDescr.segm-evt-entry-count
0x00	0x05			5 записей
0x00	0x00			SegmDataEventDescr.segm-evt-status: непрерывные данные
0x00	0x14			20 байт 5 измерений SpO2 и частоты пульса
0x00	0x62			Измерение № 5: SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту
0x00	0x60			Измерение № 6: SpO2: 96%
0x00	0x39			Частота пульса: 57 ударов в минуту
0x00	0x5C			Измерение № 7: SpO2: 92%
0x00	0x42			Частота пульса: 66 ударов в минуту
0x00	0x5A			Измерение № 8: SpO2: 90%
0x00	0x48			Частота пульса: 72 удара в минуту
0x00	0x60			Измерение № 9: SpO2: 96%
0x00	0x46			Частота пульса: 70 ударов в минуту

H.4.10 Подтверждение менеджером получения второго блока

Менеджер отвечает на передачу агентом второго набора результатов измерений.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x1A			CHOICE.length = 26
0x00	0x18			OCTET STRING.length = 24
0x03	0x25			invoke-id = 0x0325 (начало DataApdu. Кодирование MDER.)
0x02	0x01			CHOICE(Remote Operation response | Confirmed Event Report)
0x00	0x12			CHOICE.length = 18
0x00	0x64			obj-handle = 100 (объект PM-store)
0x0D	0x21			event-type = MDC_NOTI_SEGMENT_DATA
0x00	0x0C			event-info.length = 12
0x00	0x01			SegmDataEventDescr.segm-instance
0x00	0x00			SegmDataEventDescr.segm-evt-entry-index
0x00	0x05			Индекс записи, 5
0x00	0x00			SegmDataEventDescr.segm-evt-entry-count
0x00	0x05			5 записей
0x00	0x80			SegmDataEventDescr.segm-evt-status: подтверждено

H.4.11 Отправка агентом последнего блока измерений, хранящегося в сегменте PM-segment

Блок APDU имеет ту же структуру, что и первый блок данных, однако теперь в segm-evt-status бит sevtsta-last-entry установлен.

0xE7	0x00			APDU CHOICE Type (PrstApdu)
0x00	0x2C			CHOICE.length = 44
0x00	0x2A			OCTET STRING.length = 42
0x03	0x26			invoke-id = 0x0326 (начало DataApdu. Кодирование MDER.)
0x01	0x01			CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00	0x24			CHOICE.length = 36
0x00	0x64			obj-handle = 100 (объект PM-store)
0xZZ	0xZZ	0xZZ	0xZZ	RelativeTime
0x0D	0x21			event-type = MDC_NOTI_SEGMENT_DATA
0x00	0x1A			event-info.length =26
0x00	0x01			SegmDataEventDescr.segm-instance
0x00	0x00			SegmDataEventDescr.segm-evt-entry-index
0x00	0x0A			Индекс записи, 10
0x00	0x00			SegmDataEventDescr.segm-evt-entry-count
0x00	0x03			3 записи
0x40	0x00			SegmDataEventDescr.segm-evt-status: последняя запись
0x00	0x0C			12 байт 5 измерений SpO2 и частоты пульса
0x00	0x62			Измерение № 10: SpO2: 98%
0x00	0x3E			Частота пульса: 62 удара в минуту
0x00	0x62			Измерение № 11: SpO2: 98%
0x00	0x39			Частота пульса: 57 ударов в минуту
0x00	0x62			Измерение № 12 SpO2: 98%
0x00	0x37			Частота пульса: 55 ударов в минуту

Н.4.12 Подтверждение менеджером получения последнего блока

Менеджер отвечает на передачу агентом третьего набора результатов измерений. Обратите внимание, что бит sevtsta-last-entry установлен.

0xE7	0x00	APDU CHOICE Type (PrstApdu)
0x00	0x1A	CHOICE.length = 26
0x00	0x18	OCTET STRING.length = 24
0x03	0x26	invoke-id = 0x0326 (начало DataApdu. Кодирование MDER.)
0x02	0x01	CHOICE(Remote Operation response | Confirmed Event Report)
0x00	0x12	CHOICE.length = 18
0x00	0x64	obj-handle = 100 (объект PM-store)
0x0D	0x21	event-type = MDC_NOTI_SEGMENT_DATA
0x00	0x0C	event-info.length = 12
0x00	0x01	SegmDataEventDescr.segm-instance
0x00	0x00	SegmDataEventDescr.segm-evt-entry-index
0x00	0x0A	Индекс записи, 10
0x00	0x00	SegmDataEventDescr.segm-evt-entry-count
0x00	0x03	3 записи
0x40	0x80	SegmDataEventDescr.segm-evt-status: последний блок данных подтвержден

Н.4.13 Удаление менеджером PM-segment

Следующий код вызывает действие удаления всех сегментов агента.

0xE7	0x00	APDU CHOICE Type (PrstApdu)
0x00	0x14	CHOICE.length = 20
0x00	0x12	OCTET STRING.length = 18
0x03	0x27	invoke-id = 0x0327(начало DataApdu. Кодирование MDER.)
0x01	0x07	CHOICE(Remote Operation Invoke | Confirmed Action)
0x00	0x0C	CHOICE.length = 12
0x00	0x64	obj-handle = 100 (объект PM-store)
0x0C	0x0C	action-type = MDC_ACT_SEG_CLR
0x00	0x06	actionInfoArgs.length = 6
0x00	0x01	CHOICE = all-segments
0x00	0x02	CHOICE.length = 2
0x00	0x00	Не используется

Н.4.14 Агент удаляет сегмент и отвечает менеджеру

Агент отвечает на запрос менеджера об удалении всех сегментов.

0xE7	0x00	APDU CHOICE Type (PrstApdu)
0x00	0x10	CHOICE.length = 16
0x00	0x0E	OCTET STRING.length = 14
0x03	0x27	invoke-id = 0x0327(начало DataApdu. Кодирование MDER.)
0x02	0x07	CHOICE(Remote Operation Response | Confirmed Action)
0x00	0x08	CHOICE.length = 8
0x00	0x64	obj-handle = 100 (объект PM-store)
0x0C	0x0C	action-type = MDC_ACT_SEG_CLR
0x00	0x02	actionInfoArgs.length = 2
0x00	0x00	(пусто)

Приложение I
(обязательное)

Номенклатурные коды

Это приложение содержит номенклатурные коды, используемые в настоящем стандарте. Номенклатурные коды скопированы из ИСО/ИИЭР 11073-10101 [B16] или созданы для этого стандарта и добавлены в ИСО/ИИЭР 11073-10101.

Используется формат, определенный в стандарте ИСО/ИИЭР 11073-10101.

```
/* Коды разделов                                                    */
#define MDC_PART_UNSPEC                 0       /* Не указано                                  */
#define MDC_PART_OBJ                    1       /* Инфраструктура объектов                     */
#define MDC_PART_SCADA                  2       /* SCADA (Идентификаторы
                                                /*физиологических параметров)                 */
#define MDC_PART_EVT                    3       /* Оповещения/события                          */
#define MDC_PART_DIM                    4       /* Размерность                                 */
#define MDC_PART_VATTR                  5       /* Виртуальные атрибуты                        */
#define MDC_PART_PGRP                   6       /* Идентификатор группы параметров             */
#define MDC_PART_SITES                  7       /* Места частей тела                           */
#define MDC_PART_INFRA                  8       /* Инфраструктура                              */
#define MDC_PART_FEF                    9       /* Формат обмена файлами                       */
#define MDC_PART_ECG_EXTN               10      /* Расширения для ЭКГ                          */
#define MDC_PART_IDCO_EXTN              11      /* Расширения для IDCO                         */
#define MDC_PART_PHD_DM                 128     /* Управление течением заболевания             */
#define MDC_PART_PHD_HF                 129     /* Здоровье и фитнес                           */
#define MDC_PART_PHD_AI                 130     /* Одинокое старение                           */
#define MDC_PART_RET_CODE               255     /* Коды возврата                               */
#define MDC_PART_EXT_NOM                256     /* Расширенная номенклатура                    */
#define MDC_PART_PVT                    1024    /* Местный раздел                              */

/*****************************************************************************************************************************
* Инфраструктура объектов (MDC_PART_OBJ)
*****************************************************************************************************************************
#define MDC_MOC_VMO_METRIC              4       /*                                             */
#define MDC_MOC_VMO_METRIC_ENUM         5       /*                                             */
#define MDC_MOC_VMO_METRIC_NU           6       /*                                             */
#define MDC_MOC_VMO_METRIC_SA_RT        9       /*                                             */
#define MDC_MOC_SCAN                    16      /*                                             */
#define MDC_MOC_SCAN_CFG                17      /*                                             */
#define MDC_MOC_SCAN_CFG_EPI            18      /*                                             */
#define MDC_MOC_SCAN_CFG_PERI           19      /*                                             */
#define MDC_MOC_VMS_MDS_SIMP            37      /*                                             */
#define MDC_MOC_VMO_PMSTORE             61      /*                                             */
#define MDC_MOC_PM_SEGMENT              62      /*                                             */
#define MDC_ATTR_CONFIRM_MODE           2323    /*                                             */
#define MDC_ATTR_CONFIRM_TIMEOUT        2324    /*                                             */
#define MDC_ATTR_TRANSPORT_TIMEOUT      2694    /*                                             */
#define MDC_ATTR_ID_HANDLE              2337    /*                                             */
#define MDC_ATTR_ID_INSTNO              2338    /*                                             */
#define MDC_ATTR_ID_LABEL_STRING        2343    /*                                             */
#define MDC_ATTR_ID_MODEL               2344    /*                                             */
#define MDC_ATTR_ID_PHYSIO              2347    /*                                             */
#define MDC_ATTR_ID_PROD_SPECN          2349    /*                                             */
#define MDC_ATTR_ID_TYPE                2351    /*                                             */
#define MDC_ATTR_METRIC_STORE_CAPAC_CNT 2369    /*                                             */
#define MDC_ATTR_METRIC_STORE_SAMPLE_ALG 2371   /*                                             */
#define MDC_ATTR_METRIC_STORE_USAGE_CNT 2372    /*                                             */
#define MDC_ATTR_MSMT_STAT              2375    /*                                             */
#define MDC_ATTR_NU_ACCUR_MSMT          2378    /*                                             */
#define MDC_ATTR_NU_CMPD_VAL_OBS        2379    /*                                             */
```

```
#define MDC_ATTR_NU_VAL_OBS                     2384    /*      */
#define MDC_ATTR_NUM_SEG                        2385    /*      */
#define MDC_ATTR_OP_STAT                        2387    /*      */
#define MDC_ATTR_POWER_STAT                     2389    /*      */
#define MDC_ATTR_SA_SPECN                       2413    /*      */
#define MDC_ATTR_SCALE_SPECN_I16                2415    /*      */
#define MDC_ATTR_SCALE_SPECN_I32                2416    /*      */
#define MDC_ATTR_SCALE_SPECN_I8                 2417    /*      */
#define MDC_ATTR_SCAN_REP_PD                    2421    /*      */
#define MDC_ATTR_SEG_USAGE_CNT                  2427    /*      */
#define MDC_ATTR_SYS_ID                         2436    /*      */
#define MDC_ATTR_SYS_TYPE                       2438    /*      */
#define MDC_ATTR_TIME_ABS                       2439    /*      */
#define MDC_ATTR_TIME_BATT_REMAIN               2440    /*      */
#define MDC_ATTR_TIME_END_SEG                   2442    /*      */
#define MDC_ATTR_TIME_PD_SAMP                   2445    /*      */
#define MDC_ATTR_TIME_REL                       2447    /*      */
#define MDC_ATTR_TIME_STAMP_ABS                 2448    /*      */
#define MDC_ATTR_TIME_STAMP_REL                 2449    /*      */
#define MDC_ATTR_TIME_START_SEG                 2450    /*      */
#define MDC_ATTR_TX_WIND                        2453    /*      */
#define MDC_ATTR_UNIT_CODE                      2454    /*      */
#define MDC_ATTR_UNIT_LABEL_STRING              2457    /*      */
#define MDC_ATTR_VAL_BATT_CHARGE                2460    /*      */
#define MDC_ATTR_VAL_ENUM_OBS                   2462    /*      */
#define MDC_ATTR_TIME_REL_HI_RES                2536    /*      */
#define MDC_ATTR_TIME_STAMP_REL_HI_RES          2537    /*      */
#define MDC_ATTR_DEV_CONFIG_ID                  2628    /*      */
#define MDC_ATTR_MDS_TIME_INFO                  2629    /*      */
#define MDC_ATTR_METRIC_SPEC_SMALL              2630    /*      */
#define MDC_ATTR_SOURCE_HANDLE_REF              2631    /*      */
#define MDC_ATTR_SIMP_SA_OBS_VAL                2632    /*      */
#define MDC_ATTR_ENUM_OBS_VAL_SIMP_OID          2633    /*      */
#define MDC_ATTR_ENUM_OBS_VAL_SIMP_STR          2634    /*      */
#define MDC_REG_CERT_DATA_LIST                  2635    /*      */
#define MDC_ATTR_NU_VAL_OBS_BASIC               2636    /*      */
#define MDC_ATTR_PM_STORE_CAPAB                 2637    /*      */
#define MDC_ATTR_PM_SEG_MAP                     2638    /*      */
#define MDC_ATTR_PM_SEG_PERSON_ID               2639    /*      */
#define MDC_ATTR_SEG_STATS                      2640    /*      */
#define MDC_ATTR_SEG_FIXED_DATA                 2641    /*      */
#define MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP       2643    /*      */
#define MDC_ATTR_SCAN_REP_PD_MIN                2644    /*      */
#define MDC_ATTR_ATTRIBUTE_VAL_MAP              2645    /*      */
#define MDC_ATTR_NU_VAL_OBS_SIMP                2646    /*      */
#define MDC_ATTR_PM_STORE_LABEL_STRING          2647    /*      */
#define MDC_ATTR_PM_SEG_LABEL_STRING            2648    /*      */
#define MDC_ATTR_TIME_PD_MSMT_ACTIVE            2649    /*      */
#define MDC_ATTR_SYS_TYPE_SPEC_LIST             2650    /*      */
#define MDC_ATTR_METRIC_ID_PART                 2655    /*      */
#define MDC_ATTR_ENUM_OBS_VAL_PART              2656    /*      */
#define MDC_ATTR_SUPPLEMENTAL_TYPES             2657    /*      */
#define MDC_ATTR_TIME_ABS_ADJUST                2658    /*      */
#define MDC_ATTR_CLEAR_TIMEOUT                  2659    /*      */
#define MDC_ATTR_TRANSFER_TIMEOUT               2660    /*      */
#define MDC_ATTR_ENUM_OBS_VAL_SIMP_BIT_STR      2661    /*      */
#define MDC_ATTR_ENUM_OBS_VAL_BASIC_BIT_STR     2662    /*      */
#define MDC_ATTR_METRIC_STRUCT_SMALL            2675    /*      */
#define MDC_ATTR_NU_CMPD_VAL_OBS_SIMP           2676    /*      */
#define MDC_ATTR_NU_CMPD_VAL_OBS_BASIC          2677    /*      */
#define MDC_ATTR_ID_PHYSIO_LIST                 2678    /*      */
```

```
#define MDC_ATTR_SCAN_HANDLE_LIST                2679    /*                  */
#define MDC_ATTR_TIME_BO                         2689    /*                  */
#define MDC_ATTR_TIME_STAMP_BO                   2690    /*                  */
#define MDC_ATTR_TIME_START_SEG_BO               2691    /*                  */
#define MDC_ATTR_TIME_END_SEG_BO                 2692    /*                  */
/* Раздел: ACT */
#define MDC_ACT_SEG_CLR                          3084    /*                  */
#define MDC_ACT_SEG_GET_INFO                     3085    /*                  */
#define MDC_ACT_SET_TIME                         3095    /*                  */
#define MDC_ACT_DATA_REQUEST                     3099    /*                  */
#define MDC_ACT_SEG_TRIG_XFER                    3100    /*                  */
#define MDC_ACT_SET_BO_TIME                      3101    /*                  */
#define MDC_ACT_SEG_GET_ID_LIST                  3102    /*                  */
/* Раздел: NOTI */
#define MDC_NOTI_CONFIG                          3356    /*                  */
#define MDC_NOTI_SCAN_REPORT_FIXED               3357    /*                  */
#define MDC_NOTI_SCAN_REPORT_VAR                 3358    /*                  */
#define MDC_NOTI_SCAN_REPORT_MP_FIXED            3359    /*                  */
#define MDC_NOTI_SCAN_REPORT_MP_VAR              3360    /*                  */
#define MDC_NOTI_SEGMENT_DATA                    3361    /*                  */
#define MDC_NOTI_UNBUF_SCAN_REPORT_VAR           3362    /*                  */
#define MDC_NOTI_UNBUF_SCAN_REPORT_FIXED         3363    /*                  */
#define MDC_NOTI_UNBUF_SCAN_REPORT_GROUPED       3364    /*                  */
#define MDC_NOTI_UNBUF_SCAN_REPORT_MP_VAR        3365    /*                  */
#define MDC_NOTI_UNBUF_SCAN_REPORT_MP_FIXED      3366    /*                  */
#define MDC_NOTI_UNBUF_SCAN_REPORT_MP_GROUPED    3367    /*                  */
#define MDC_NOTI_BUF_SCAN_REPORT_VAR             3368    /*                  */
#define MDC_NOTI_BUF_SCAN_REPORT_FIXED           3369    /*                  */
#define MDC_NOTI_BUF_SCAN_REPORT_GROUPED         3370    /*                  */
#define MDC_NOTI_BUF_SCAN_REPORT_MP_VAR          3371    /*                  */
#define MDC_NOTI_BUF_SCAN_REPORT_MP_FIXED        3372    /*                  */
#define MDC_NOTI_BUF_SCAN_REPORT_MP_GROUPED      3373    /*                  */

/**********************************************************************************************************************
* Из медицинского диспетчерского управления и сбора данных (MDC_PART_SCADA)
**********************************************************************************************************************/
#define MDC_TEMP_BODY                            19292   /* Температура тела */
#define MDC_MASS_BODY_ACTUAL                     57664   /*                  */
/* Раздел: SCADA/Прочее
#define MDC_BODY_FAT                             57676   /*                  */

/**********************************************************************************************************************
* Из размерностей (MDC_PART_DIM)
**********************************************************************************************************************/
#define MDC_DIM_PERCENT                          544     /* %                */
#define MDC_DIM_KILO_G                           1731    /* кг               */
#define MDC_DIM_MIN                              2208    /* мин              */
#define MDC_DIM_HR                               2240    /* ч                */
#define MDC_DIM_DAY                              2272    /* д                */
#define MDC_DIM_DEGC                             6048    /* °C               */

/**********************************************************************************************************************
* Из коммуникационной инфраструктуры (MDC_PART_INFRA)
**********************************************************************************************************************/
#define MDC_DEV_SPEC_PROFILE_HYDRA               4096    /* Гидроустройство  */
#define MDC_DEV_SPEC_PROFILE_PULS_OXIM           4100    /* Пульсоксиметр    */
#define MDC_DEV_SPEC_PROFILE_ECG                 4102    /* Базовая ЭКГ      */
#define MDC_DEV_SPEC_PROFILE_BP                  4103    /* Артериальное давление */
#define MDC_DEV_SPEC_PROFILE_TEMP                4104    /* Термометр        */
#define MDC_DEV_SPEC_PROFILE_SCALE               4111    /* Весы             */
#define MDC_DEV_SPEC_PROFILE_GLUCOSE             4113    /* Глюкометр        */
```

```
#define MDC_DEV_SPEC_PROFILE_RESP_RATE 4114 /* Частота дыхания */
#define MDC_DEV_SPEC_PROFILE_INSULIN_PUMP 4115 /* Дозатор инсулина */
#define MDC_DEV_SPEC_PROFILE_BCA 4116 /* Анализатор состава тканей тела */
#define MDC_DEV_SPEC_PROFILE_PEFM 4117 /* Монитор пиковой скорости выдоха */
#define MDC_DEV_SPEC_PROFILE_COAG 4118 /* Международное нормализованное отношение */
#define MDC_DEV_SPEC_PROFILE_URINE_ANALYZER 4119 /* Анализатор мочи */
#define MDC_DEV_SPEC_PROFILE_SLEEP_QUALITY 4120 /* Монитор качества сна */
#define MDC_DEV_SPEC_PROFILE_SLEEP_APONEA 4121 /* Терапевтическое устройство апноэ сна */
#define MDC_DEV_SPEC_PROFILE_CGM 4122 /* Глюкометр непрерывного действия */
#define MDC_DEV_SPEC_PROFILE_HF_CARDIO 4137 /* Сердечно-сосудистый фитнес и монитор активности */
#define MDC_DEV_SPEC_PROFILE_HF_STRENGTH 4138 /* Оборудование силового фитнеса */
#define MDC_DEV_SPEC_PROFILE_AI_ACTIVITY_HUB 4167 /* Концентратор активности при независимом проживании */
#define MDC_DEV_SPEC_PROFILE_AI_MED_MINDER 4168 /* Монитор приема лекарств */
/* Диапазон от 4196 до 5119 зарезервирован для подклассов устройств в рамках других специализаций (профилей). Например, специализация -10441 (кардио) определяет шагомер в качестве профиля */

/* Диапазон от 4236 до 4243 используется для стандарта IEEE 11073-10406™ [B19] (Базовая ЭКГ) */
#define #define MDC_DEV_SUB_SPEC_PROFILE_ECG 4236 /* */
#define #define MDC_DEV_SUB_SPEC_PROFILE_HR 4237 /* */

/* Диапазон от 4196 до 4211 используется для стандарта IEEE 11073-10441™ [B9] (кардио) */
#define MDC_DEV_SUB_SPEC_PROFILE_STEP_COUNTER 4196 /* Шагомер */
#define MDC_DEV_SUB_SPEC_PROFILE_ACTIVITY 4197 /* Монитор активности */

/* Диапазон от 4212 до 4235 используется для стандарта IEEE 11073-10471™ [B11] (концентратор активности) */
#define MDC_DEV_SUB_SPEC_PROFILE_FALL_SENSOR 4213 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_PERS_SENSOR 4214 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_SMOKE_SENSOR 4215 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_CO_SENSOR 4216 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_WATER_SENSOR 4217 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_GAS_SENSOR 4218 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_MOTION_SENSOR 4219 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_PROPEXIT_SENSOR 4220 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_ENURESIS_SENSOR 4221 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_CONTACTCLOSURE_SENSOR 4222 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_USAGE_SENSOR 4223 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_SWITCH_SENSOR 4224 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_DOSAGE_SENSOR 4225 /* */
#define MDC_DEV_SUB_SPEC_PROFILE_TEMP_SENSOR 4226 /* */

/* Начинается на 256 ранее начала последнего раздела: OptionalPackageIdentifiers (например, 8192–256) */
#define MDC_TIME_SYNC_NONE 7936 /* Протокол синхронизации времени не поддерживается */
#define MDC_TIME_SYNC_NTPV3 7937 /* RFC 1305, март 1992: зам. 1119, 1059, 958 */
#define MDC_TIME_SYNC_NTPV4 7938 /* <На стадии разработки по адресу ntp.org> */
#define MDC_TIME_SYNC_SNTPV4 7939 /* RFC 2030, октябрь 1996: зам. 1769 */
#define MDC_TIME_SYNC_SNTPV4330 7940 /* RFC 4330, январь 2006: зам. 2030, 1769 */
#define MDC_TIME_SYNC_BTV1 7941 /* Профиль медицинского прибора Bluetooth */
#define MDC_TIME_SYNC_RADIO 7942 /* Синхронизация по атомным часам с помощью радиосигнала */
```

```
#define MDC_TIME_SYNC_HL7_NCK          7943    /* Синхронизация с помощью Health
                                               Level 7 NCK (сетевые часы)        */
#define MDC_TIME_SYNC_CDMA             7944    /* Синхронизация мобильной
                                               телекоммуникационной системы
                                               CDMA                              */
#define MDC_TIME_SYNC_GSM              7945    /* GSM — Сетевая идентификация
                                               и часовой пояс (NITZ)             */
#define MDC_TIME_SYNC_EBWW             7946    /* Время, задаваемое вручную
                                               с помощью 'глазного яблока
                                               и наручных часов'                 */
#define MDC_TIME_SYNC_USB_SOF          7947    /* Синхронизация с помощью
                                               USB-часов 1 кГц по "началу кадра" */
#define MDC_TIME_SYNC_OTHER            7948    /* Метод синхронизации времени,
                                               не регламентируемый
                                               IEEE 11073-20601                  */

/**************************************************************************************************************************************************
* Из кодов возврата (MDC_PART_RET_CODE)
**************************************************************************************************************************************************/
#define MDC_RET_CODE_UNKNOWN           1       /* Код общей ошибки                  */

/* Раздел ошибок объектов MDC_PART_RET_CODE/OBJ */
#define MDC_RET_CODE_OBJ_BUSY          1000    /* Объект занят, поэтому не может
                                               обработать запрос                 */

/* Раздел ошибок хранения MDC_PART_RETURN_CODES/STORE */
#define MDC_RET_CODE_STORE_EXH         2000    /* Ошибка, связанная
                                               с переполнением диска             */
#define MDC_RET_CODE_STORE_OFFLN       2001    /* Ошибка, связанная с отключенным
                                               диском                            */
```

Приложение J
(справочное)
История происхождений и изменений

J.1 Общие положения

Многие аспекты информационных, сервисных и коммуникационных моделей определены в других стандартах ИСО/ИИЭР 11073. Настоящее приложение описывает происхождение структур ACH.1, номенклатуры и правил кодирования, а также любых модификаций, адаптирующих их к предметной области персональных медицинских приборов. Целевой аудиторией настоящего приложения являются прежде всего пользователи, которые поддерживают и стремятся обеспечить согласованность стандартов ИСО/ИИЭР 11073.

J.2 Структуры ACH.1

Следующие структуры ACH.1 заимствованы из ИСО/ИИЭР 11073-10201:2004 [B17] без изменения:

INT-U8, INT-I8, INT-U16, INT-I16, INT-U32, INT-I32, BITS-16, BITS-32, OID-Type, PrivateOid, HANDLE, InstNumber, TYPE, AVA-Type, AttributeList, AttributeIdList, FLOAT-Type, RelativeTime, HighResRelativeTime, AbsoluteTime, OperationalState, SystemModel, ProductionSpec, ProdSpecEntry, PowerStatus, BatMeasure, NuObsValue, NuObsValueCmp, SaSpec, SampleType, SaFlags, ScaleRangeSpec8, ScaleRangeSpec16, ScaleRangeSpec32, EnumObsValue, ConfirmMode, SetTimeInvoke, SegmSelection, SegmIdList, AbsTimeRange, SegmentInfoList, SegmentInfo, ObservationScan и TimeProtocolId.

Следующие структуры ACH.1 заимствованы из ISO/IEEE 11073-10201:2004 [B17] с изменениями:

NomPartition, StoSampleAlg, MeasurementStatus и EnumVal.

Остальные структуры ACH.1 созданы специально для этого стандарта.

J.3 Правила кодирования медицинских приборов (MDER)

Правила MDER заимствованы из ИСО/ИИЭР 11073-10101 [B16]. Подавляющее большинство изменений, внесенных в заимствованные правила, представляют собой редакторские правки, однако в нескольких случаях необязательные элементы стали обязательными (например, в таблице F.1).

Оптимизация и пояснения, приведенные в F.7 и F.8, специфичны для настоящего стандарта.

J.4 Номенклатурные коды

J.4.1 Общие положения

Подразделы J.4.2 — J.4.6 описывают происхождение кодов, перечисленных в приложении I.

J.4.2 Коды разделов

Добавлены четыре новых кода разделов:
MDC_PART_PHD_DM, MDC_PART_PHD_HF,
MDC_PART_PHD_AI и MDC_PART_RET_CODE.
Все остальные коды заимствованы из ИСО/ИИЭР 11073-10101 [B16].

J.4.3 Коды инфраструктуры объектов

J.4.3.1 Коды MDC_MOC

Все номенклатурные коды, начинающиеся с MDC_MOC_, заимствованы из ISO/IEEE 11073-10101 [B16].

J.4.3.2 Коды MDC_ATTR

Добавлены следующие коды, начинающиеся с MDC_ATTR:
MDC_ATTR_DEV_CONFIG_ID,
MDC_ATTR_MDS_TIME_INFO,
MDC_ATTR_METRIC_SPEC_SMALL,
MDC_ATTR_SOURCE_HANDLE_REF,
MDC_ATTR_SIMP_SA_OBS_VAL,
MDC_ATTR_ENUM_OBS_VAL_SIMP_OID,
MDC_ATTR_ENUM_OBS_VAL_SIMP_STR,
MDC_ATTR_NU_VAL_OBS_BASIC,
MDC_ATTR_ PM_STORE_CAPAB,
MDC_ATTR_PM_SEG_MAP,
MDC_ATTR_PM_SEG_PERSON_ID,
MDC_ATTR_SEG_STATS,
MDC_ATTR_SEG_FIXED_DATA,
MDC_ATTR_PM_SEG_ELEM_STAT_ATTR,
MDC_ATTR_SCAN_HANDLE_ATTR_VAL_MAP,
MDC_ATTR_SCAN_REP_PD_MIN,

MDC_ATTR_ATTRIBUTE_VAL_MAP,
MDC_ATTR_NU_VAL_OBS_SIMP,
MDC_ATTR_PM_STORE_LABEL_STRING,
MDC_ATTR_PM_SEG_LABEL_STRING,
MDC_ATTR_TIME_PD_MSMT_ACTIVE,
MDC_ATTR_SYS_TYPE_SPEC_LIST,
MDC_ATTR_METRIC_STRUCT_SMALL,
MDC_ATTR_NU_CMPD_VAL_OBS_SIMP,
MDC_ATTR_NU_CMPD_VAL_OBS_BASIC,
MDC_ATIR_ID_PHYSIO_LIST.
Все остальные коды заимствованы из ISO/IEEE 11073-10101 [B16].

J.4.3.3 Коды MDC_ACT

Добавлены следующие коды, начинающиеся с MDC_ACT:
MDC_ACT_DATA_REQUEST и MDC_ACT_SEG_TRIG_XFER.
Все остальные коды заимствованы из ИСО/ИИЭР 11073-10101 [B16].

J.4.3.4 Коды MDC_NOTI

Все коды MDC_NOTI вновь созданы для настоящего стандарта.

J.4.3.5 Коды MDC_RET_CODE

Все коды MDC_RET_CODE вновь созданы для настоящего стандарта.

J.4.4 Медицинское диспетчерское управление и сбор данных

Добавлен код MDC_BODY_FAT. Все остальные коды заимствованы из ИСО/ИИЭР 11073-10101 [B16].

J.4.5 Коды размерностей

Все коды размерностей заимствованы из ИСО/ИИЭР 11073-10101 [B16].

J.4.6 Коды коммуникационной инфраструктуры

J.4.6.1 Коды MDC_DEV_SPEC_PROFILE

Добавлены следующие коды, начинающиеся с MDC_DEV_SPEC_PROFILE:
MDC_DEV_SPEC_PROFILE_GLUCOSE,
MDC_DEV_SPEC_PROFILE_HF_CARDIO,
MDC_DEV_SPEC_PROFILE_HF_STRENGTH,
MDC_DEV_SPEC_PROFILE_AI_ACTIVITY_HUB и
MDC_DEV_SPEC_PROFILE_AI_MED_MINDER.
Все остальные коды заимствованы из ИСО/ИИЭР 11073-10101 [B16].

J.4.6.2 Коды MDC_TIME_SYNC

Все коды MDC_TIME_SYNC вновь созданы для настоящего стандарта.

Приложение К
(справочное)

Библиография

Библиографические ссылки представляют собой ресурсы, содержащие дополнительную или полезную информацию, которую не обязательно понимать или использовать для применения настоящего стандарта. Ссылки на эти ресурсы даны только для информационных целей. Для удобства читателей номера в скобках, добавленные в текст настоящего стандарта, соответствуют номерам библиографических ссылок.

[B1] IEC 60601-1 (2005) +A1 (2012), Ed. 3.1, Medical electrical equipment — Part 1: General requirements for basic safety and essential performance (Изд. 3.1. Медицинское электрооборудование. Часть 1. Общие требования, предъявляемые к базовой безопасности и основным характеристикам)[22]

[B2] IEC 60601-2, Medical electrical equipment — Part 2: Particular requirements for the basic safety and essential performance for specific device (See the entire series of standards, Part 2-1 through Part 2-51) (Медицинское электрооборудование. Часть 2. Частные требования, предъявляемые к базовой безопасности и основным характеристикам конкретного устройства. См. всю серию стандартов, части с 2-1 по 2-51)

[B3] IEC 62304 (2006)/EN 62304 (2006), Medical device software — Software life-cycle processes (Программное обеспечение медицинского оборудования. Процессы жизненного цикла программного обеспечения).

[B4] IEC 80001-1 (2010), Application of risk management for IT-networks incorporating medical devices — Part 1: Roles, responsibilities and activities (Применение управления рисками информационно-вычислительных сетей, содержащих медицинские приборы. Часть 1. Роли, обязанности и действия)

[B5] IEEE 11073-00103™, Health informatics — Personal health device communication — Part 00103: Technical report — Overview (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 00103. Технический отчет. Обзор)[23, 24]

[B6] IEEE 11073-10408™, Health informatics — Personal health device communication — Part 10408: Device specialization — Thermometer (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10408. Специализация прибора. Термометр)

[B7] IEEE 11073-10415™, Health informatics — Personal health device communication — Part 10415: Device specialization — Weighing scale (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10415. Специализация прибора. Весы)

[B8] IEEE 11073-10417™, Health informatics — Personal health device communication — Part 10417: Device specialization — Glucose meter (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10417. Специализация прибора. Глюкометр)

[B9] IEEE 11073-10441™, Health informatics — Personal health device communication — Part 10441: Device specialization — Cardiovascular fitness and activity monitor (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10441. Специализация прибора. Монитор сердечно-сосудистого фитнеса и активности)

[B10] IEEE 11073-10442™, Health informatics — Personal health device communication — Part 10442: Device specialization — Strength fitness equipment (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10442. Специализация прибора. Оборудование силового фитнеса)

[B11] IEEE 11073-10471™, Health informatics — Personal health device communication — Part 10471: Device specialization — Independent living activity hub (Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10471. Специализация прибора. Концентратор активности при независимом проживании)

[B12] ISO 14971:2007, Medical devices — Application of risk management to medical devices (Медицинские приборы. Применение управления рисками к медицинским приборам)[25]

[B13] ISO 15225, Nomenclature — Specification for a nomenclature system for medical devices for the purpose of regulatory data exchange (Номенклатура. Спецификация номенклатурной системы медицинских приборов для целей обмена регуляторными данными)

---

[22] Публикации IEC распространяются Международной электротехнической комиссией (gost.e-ecolog.ru

[23] Публикации IEEE распространяются Институтом инженеров по электротехнике и электронике (http://standards.ieee.org/).

[24] Стандарты или продукция IEEE, упомянутые в настоящем приложении, являются товарными знаками, принадлежащими Институту инженеров по электротехнике и электронике.

[25] Публикации ISO распространяются Международной организацией по стандартизации (gost.e-ecolog.ru

[B14] ISO/IEC 646:1991 Информационные технологии. Набор символов ISO с 7-битным кодированием для обмена информацией[26]

[B15] ISO/IEC 9899 Информационные технологии. Языки программирования — С

[B16] ISO/IEEE 11073-10101 Информатизация здоровья. Обмен данными с медицинскими приборами, расположенными в местах оказания медицинской помощи. Часть 10101. Номенклатура

[B17] ISO/IEEE 11073-10201:2004 Информатизация здоровья. Обмен данными с медицинскими приборами, расположенными в местах оказания медицинской помощи. Часть 10201. Информационная модель предметной области

[B18] ISO/IEEE 11073-10404 Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10404. Специализация прибора. Пульсовый оксиметр

[B19] ISO/IEEE 11073-10406 Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10406. Специализация прибора. Базовый электрокардиограф (ECG) (1—3-проводный ECG)

[B20] ISO/IEEE 11073-10407 Информатизация здоровья. Обмен данными с персональными медицинскими приборами. Часть 10407. Специализация прибора. Блок контроля кровяного давления

[B21] ISO/IEEE 11073-20101:2004 Информатизация здоровья. Обмен данными с медицинскими приборами, расположенными в местах оказания медицинской помощи. Часть 20101. Прикладные профили. Базовый стандарт

[B22] ITU-T Rec. X.680 (07/02) Информационные технологии. Язык для описания абстрактного синтаксиса версии 1 (ASN.1): спецификация на базовое описание[27]

[B23] ITU-T Rec. X.691 (07/02) Информационные технологии. Правила кодирования ASN.1: спецификация на правила уплотненного кодирования (PER)

[B24] ITU-T Rec. X.693 (12/01) Информационные технологии. Правила кодирования ASN.1: правила кодирования XML (XER)

[26] Публикации ISO/IEC распространяются Международной организацией по стандартизации (gost.e-ecolog.ru ch/). На территории США публикации ISO/IEC распространяются компанией Global Engineering Documents (http://global.ihs.com/). Электронные копии на территории США распространяются Американским национальным институтом стандартов (gost.e-ecolog.ru

[27] Публикации ITU-T распространяются Международным союзом электросвязи (gost.e-ecolog.ru

УДК 004:61:006.354 ОКС 35.240.80

Ключевые слова: здравоохранение, информатизация здоровья, обмен данными с медицинскими приборами, персональные медицинские приборы

БЗ 10—2019/132

Редактор *П.К. Одинцов*
Технический редактор *И.Е. Черепкова*
Корректор *Л.С. Лысенко*
Компьютерная верстка *А.А. Ворониной*

Сдано в набор 13.09.2019. Подписано в печать 25.10.2019. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 25,11. Уч.-изд. л. 21,34.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении во ФГУП «СТАНДАРТИНФОРМ» для комплектования Федерального информационного фонда стандартов, 117418 Москва, Нахимовский пр-т, д. 31, к. 2.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru