МЕЖГОСУДАРСТВЕННЫЙ СОВЕТ ПО СТАНДАРТИЗАЦИИ, МЕТРОЛОГИИ И СЕРТИФИКАЦИИ
(МГС)

INTERSTATE COUNCIL FOR STANDARDIZATION, METROLOGY AND CERTIFICATION
(ISC)

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

ГОСТ ISO/HL7 21731— 2013

Информатизация здоровья

HL7, ВЕРСИЯ 3

Эталонная информационная модель

Выпуск 1

(ISO/HL7 21731:2006, IDT)

Издание официальное

Москва
Стандартинформ
2015

gost.e-ecolog.ru

Предисловие

Цели, основные принципы и основной порядок проведения работ по межгосударственной стандартизации установлены ГОСТ 1.0—92 «Межгосударственная система стандартизации. Основные положения» и ГОСТ 1.2—2009 «Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Правила разработки, принятия, применения, обновления и отмены»

Сведения о стандарте

1 ПОДГОТОВЛЕН Федеральным бюджетным учреждением «Консультационно-внедренческая фирма в области международной стандартизации и сертификации — «Фирма «ИНТЕРСТАНДАРТ» (ФБУ «КВФ «Интерстандарт») совместно с Федеральным государственным бюджетным учреждением «Центральный научно-исследовательский институт организации и информатизации здравоохранения Министерства здравоохранения Российской Федерации» (ФГБУ «ЦНИИОИЗ Минздрава России») на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 5

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

3 ПРИНЯТ Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 5 ноября 2013 г. № 61-П)

За принятие проголосовали:

Краткое наименование страны по МК (ИСО 3166) 004—97	Код страны по МК (ИСО 3166) 004—97	Сокращенное наименование национального органа по стандартизации
Армения	AM	Минэкономики Республики Армения
Беларусь	BY	Госстандарт Республики Беларусь
Киргизия	KG	Кыргызстандарт
Молдова	MD	Молдова-Стандарт
Россия	RU	Росстандарт
Узбекистан	UZ	Узстандарт

4 Приказом Федерального агентства по техническому регулированию и метрологии от 28 апреля 2014 г. № 422-ст межгосударственный стандарт ГОСТ ISO/HL7 21731—2013 введен в действие в Российской Федерации для применения в качестве национального стандарта с 1 июля 2015 г.

5 Настоящий стандарт идентичен международному стандарту ISO/HL7 21731:2006 Health informatics — HL7 version 3 — Reference information model — Release 1 (Информатизация здоровья. HL7, версия 3. Эталонная информационная модель. Выпуск 1).

Международный стандарт разработан техническим комитетом Международной организации по стандартизации ISO/TC 215 Health informatics (Информатизация здоровья).

Перевод с английского языка (en).

Официальные экземпляры международного стандарта, на основе которого подготовлен настоящий межгосударственный стандарт, имеются в ФГУП «СТАНДАРТИНФОРМ».

Степень соответствия — идентичная (IDT)

6 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодном информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячном информационном указателе «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

© Стандартинформ, 2015

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

Содержание

1 Область применения . . . 1
2 Соответствие . . . 1
3 Нормативные ссылки . . . 1
4 Термины и определения . . . 2
5 Интерпретация спецификации . . . 4
5.1 Содержание спецификации . . . 4
5.2 Понимание ЭИМ . . . 5
5.3 Графические диаграммы нормативного содержания ЭИМ . . . 6
6 Предметные области ЭИМ HL7 . . . 12
6.1 Предметная область FoundationClasses . . . 12
7 Классы ЭИМ HL7 . . . 13
7.1 Классы предметной области Acts . . . 13
7.2 Классы предметной области Entities . . . 48
7.3 Классы предметной области Roles . . . 59
8 Ассоциации в ЭИМ, выпуск 1 . . . 65
9 Содержание нормативного словаря . . . 65
9.1 Структурированный словарь ЭИМ . . . 65
Приложение А (справочное) История развития ЭИМ . . . 117
А.1 История ЭИМ . . . 117
А.2 Значение утверждения ЭИМ . . . 117
Приложение Б (справочное) Обзор принципов разработки ЭИМ . . . 118
Б.1 Цель . . . 118
Б.2 Общие сведения . . . 118
Б.3 Обоснование конструкции ЭИМ . . . 118
Б.4 Связывание действий: семантика класса ActRelationship . . . 119
Б.5 Определения шести базовых классов ЭИМ . . . 120
Б.6 Типы данных и спецификации словарей . . . 122
Б.7 Методология стандарта HL7 Версия 3 и ЭИМ . . . 122
Приложение В (справочное) Сводный перечень типов данных Версии 3 . . . 123
В.1 Обзор типов данных . . . 123
Приложение Г (справочное) Сведения о соответствии ссылочных межгосударственных стандартов международным стандартам (международным документам) . . . 127

Предисловие

ИСО (International Organization for Standardization — Международная организация по стандартизации) — международная федерация национальных органов стандартизации (членов ИСО). Подготовка международных стандартов обычно выполняется техническими комитетами ИСО. Каждый член ИСО, заинтересованный в предметной области, для стандартизации которой был создан соответствующий технический комитет, имеет право быть представленным в этом комитете. Международные организации, правительственные и неправительственные, имеющие представительство в ИСО, также принимают участие в работе технических комитетов. ИСО тесно сотрудничает с Международной электротехнической комиссией (МЭК) по всем вопросам электротехнической стандартизации.

Международные стандарты разработаны в соответствии с правилами, установленными частью 2 директив ИСО/МЭК.

Главной задачей технических комитетов является подготовка международных стандартов. Проекты международных стандартов, одобренные техническими комитетами, предлагаются членам комитета для голосования. Публикация документа в качестве международного стандарта требует одобрения по крайней мере 75 % членов комитета, принявших участие в голосовании.

Для разработки и сопровождения группы стандартов ISO/HL7 в области медицинских приборов заключено экспериментальное соглашение между ИСО и комитетом Health Level Seven Inc (HL7), одобренное решением Совета 7/2002. В соответствии с этим соглашением на комитет HL7 возлагается ответственность за разработку и сопровождение этих стандартов при участии членов ИСО, представляющих свои предложения.

Особое внимание должно быть уделено возможности, что некоторые из элементов этого документа могут быть предметом патентных прав. ИСО не берет на себя ответственность за идентификацию любых таких патентных прав.

Стандарт ISO/HL7 21731 разработан HL7 и Техническим комитетом ISO/TC 215, Health informatics (Информатизация здоровья).

Введение

Это введение посвящено обсуждению требований, предъявляемых к стандартизации Эталонной информационной модели (ЭИМ). Дальнейшие сведения о развитии этой модели и причинах ее развития можно найти в Приложении А.

Применение ЭИМ в информатизации здоровья

Использование ЭИМ в ИСО ТК 215

Технический комитет ИСО ТК 215 «Информатизация здоровья» в течение ряда лет рассматривал документ ИСО 17113, представляющий собой описание методологии разработки стандартов передачи медицинских данных. Согласно этой методологии такие стандарты должны быть основаны на единственной, всесторонней модели информации о здоровье и медицинской помощи. Одной из таких моделей является ЭИМ, представленая в настоящем документе. Кроме того, ЭИМ может рассматриваться как справочный документ, способствующий гармонизации стандартов информатизации здоровья и соответствующих спецификаций, подготавливаемых комитетом ИСО ТК 215.

Использование ЭИМ в комитете HL7

ЭИМ является критичным компонентом процесса развития стандартов HL7 Версии 3. Она положена в основу всех информационных моделей и структур, разрабатываемых в процессе развития Версии 3.

Разработка стандартов HL7 Версии 3 ведется в соответствии с методологией, ориентированной на применение моделей, согласно которой для описания статических и динамических аспектов требований и разработки стандартов HL7, а также для описания соответствующей семантики и правил деятельности разрабатывается комплекс взаимосвязанных моделей.

ЭИМ представляет собой статическую точку зрения на информационные потребности стандартов HL7 Версии 3. Она включает в себя классы информационных объектов и диаграммы перехода состояний, дополняемые моделями вариантов использования, моделями взаимодействия, моделями типов данных, моделями терминологии и другими типами моделей, необходимых для обеспечения полной точки зрения на требования и архитектуру стандартов HL7. Классы, атрибуты, переходы состояний и ассоциации, описанные в ЭИМ, используются для производства предметно-ориентированных информационных моделей, кото-

рые с помощью ряда процессов уточнения ограничений преобразуются в статическую модель содержания информации, используемую в стандарте HL7.

Процесс разработки стандарта HL7 определяет правила производства предметно-ориентированных информационных моделей из ЭИМ и уточнения этих моделей в спецификациях стандартов HL7. Правила требуют, чтобы все информационные структуры, описанные в производных моделях, могли быть прослежены обратно к ЭИМ и чтобы их семантика и соответствующие правила деятельности не противоречили тем, которые определены в ЭИМ. Поэтому ЭИМ является основным источником всего информационного содержания стандартов HL7 Версии 3.

ЭИМ используется международными филиалами HL7 для адаптации стандартов HL7 Версии 3 к местным особенностям. С помощью процесса, называемого локализацией, спецификации стандарта Версии 3 могут быть расширены, используя ЭИМ как источник нового информационного содержания. Такая новая информация производится из ЭИМ и уточняется тем же способом, что и применяется для создания исходной спецификации.

Использование ЭИМ вне HL7

В основном ЭИМ предназначена для использования комитетом HL7 и его международными филиалами. Однако ЭИМ может использоваться и вне структур комитета HL7. Несмотря на то, что комитет HL7 обладает авторскими правами на данный стандарт, он не требует лицензирования или иного способа контроля использования информационных структур или программ, реализующих настоящий стандарт.

Первые разработчики, принявшие на вооружение методологию разработки стандартов Версии 3, использовали ЭИМ для конструирования спецификаций сообщений, похожих на описанные в стандарте HL7, в своих собственных средах. К этим разработчикам относятся производители информационных систем, большие сети интегрированных медицинских организаций и правительственные агентства как в США, так и в других странах. Эти же самые первые разработчики крайне активны в комитете HL7 и вносят практические предложения по усовершенствованию ЭИМ и других аспектов процесса разработки Версии 3.

Некоторые организации — участники комитета HL7 сообщили об использовании ЭИМ как источника разработки корпоративной информационной архитектуры или как исходной точки для системного анализа и проектирования. ЭИМ действительно может использоваться для таких целей, однако комитет HL7 не дает никаких гарантий, что ЭИМ полезна не только как эталонная модель, предназначенная для разработки стандартов HL7.

ЭИМ — только одна модель информационных потребностей здравоохранения. Абстрактный стиль ЭИМ и возможность расширения ЭИМ с помощью спецификаций словарей данных позволяет применять ЭИМ для любого мыслимого сценария обмена данными между информационными системами здравоохранения. На самом деле она концептуально применима для любой предметной области, в которой определенные сущности выполняют роли и участвуют в действиях.

Универсальная применимость ЭИМ делает ее особенно полезной для организаций, подобных комитету HL7, которые должны учитывать потребности большого и разнообразного членства. Стиль ЭИМ делает ее чрезвычайно стабильной, что является еще одним свойством, важным для комитета HL7.

Процесс разработки стандартов HL7 состоит в создании предметно-ориентированных моделей, являющихся производными от ЭИМ, и последовательного уточнения этих моделей, позволяющего получить модели, специфичные для данной предметной области. Эти специфичные модели предметной области конкретизируют ЭИМ и задают ограничения значений атрибутов и ассоциаций между классами, применимые для конкретных случаев. Внешним организациям, рассматривающим возможность использование HL7 ЭИМ, рекомендуется принять подобный процесс построения производных моделей в форме преобразования ЭИМ.

Полезная информация

С вопросами или комментариям о содержании стандарта можно обратиться к комитету HL7 (www.hl7.org) или к соответствующему национальному филиалу комитета HL7 либо к Секретариату комитета ИСО ТК215 «Информатизация здоровья».

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

Информатизация здоровья

HL7, ВЕРСИЯ 3

Эталонная информационная модель

Выпуск 1

Health informatics. HL7 version 3. Reference information model. Release 1

Дата введения — 2015—07—01

1 Область применения

Эталонная информационная модель (ЭИМ) комитета Health Level Seven (HL7) представляет собой статическую модель информации о здоровье и медицинской помощи в соответствии с областями применения деятельности по развитию стандартов HL7. Она является информационной точкой зрения, согласованной комитетом HL7 и его национальными филиалами. ЭИМ служит основным источником, из которого производится все содержание стандартных спецификаций протокола HL7 Версии 3, связанное с информацией. В контексте деятельности комитета ИСО ТК215 «Информатизация здоровья» ЭИМ является эталонной моделью, которая может использоваться при разработке других стандартов информатизации здоровья.

2 Соответствие

Информационная модель, подобная ЭИМ, описываемой в настоящем стандарте, может служить основой для других информационных моделей, которые непосредственно производятся из нее, и может использоваться в качестве основы для проектирования баз данных и других информационных структур. Однако ни комитет ИСО ТК 215, ни комитет HL7 не предполагают, что целесообразно разработать правила определения того, может ли конкретная реализация соответствовать данному стандарту. Поэтому пользователи данного стандарта не должны требовать соответствия ему. Далее, комитеты ИСО ТК 215 и HL7 как разработчики данного стандарта требуют, чтобы пользователи информировали их о конкретных причинах, которые заставили пользователей отклониться от этого стандарта или расширить его. Это позволит последующим выпускам стандарта удовлетворять более широкому диапазону требований.

3 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие документы, содержание которых неразрывно связано с настоящим стандартом. Если в ссылке указана дата публикации, то должен использоваться только цитируемый документ. Если дата в ссылке не указана, то должно использоваться последнее издание документа (включая все поправки).

ISO 17113, Health informatics — Exchange of information between healthcare information systems — Method for development of messages. (ИСО 17113, Информатизация здоровья — Обмен информацией между медицинскими информационными системами — Метод разработки сообщений).

ISO/IEC 19501, Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2 (ИСО/МЭК 19501, Информационные технологии — Открытая распределенная обработка — Унифицированный язык моделирования (UML), Версия 1.4.2.).

Издание официальное

ANSI/HL7 V3 DT, R1–2004, HL7 Version 3 Standard: Data Types — Abstract Specification, Release 1 (HL7 Версия 3 — Типы данных — Общая спецификация, выпуск 1).

4 Термины и определения

Далее представлены все термины и определения документа.

4.1 **ANSI:** American National Standards Institute — Американский национальный институт по стандартизации.

4.2 **ассоциация** (association): Ссылка от одного класса к другому классу или к себе, или связи между двумя объектами (экземплярами классов).

4.3 **имя роли ассоциации** (association role name): Имя каждого конца ассоциации. Имя — это короткая глагольная фраза, описывающая роль класса, находящегося на противоположном конце ассоциации, по отношению к классу, находящемуся на данном конце ассоциации.

4.4 **атрибут:** Абстрактное описание составной части класса. При передаче сообщений HL7 атрибуты становятся значениями данных.

4.5 **мультимножество** (bag): Форма коллекции, элементы которой не упорядочены и не должны быть уникальными.

4.6 **кратность** (cardinality): Свойство элемента данных (число возможных повторений элемента данных в пределах индивидуального экземпляра представления объекта).

4.7 **класс:** Абстракция сущности или понятия в специфической прикладной области.

4.8 **атрибут классификатора** (classifier attribute): Признак, используемый в иерархиях обобщений для указания, какая из специализаций является фокусом класса.

4.9 **кодированный атрибут** (coded attribute): Атрибут ЭИМ с базовым типом данных CD, CE, CS или CV.

4.10 **сила кодирования** (coding strength): Показатель расширяемости, который определяет, может ли набор кодов быть расширен для учета местных потребностей реализации.

4.11 **система кодирования** (coding system): Схема представления понятий, использующая (обычно) короткие идентификаторы для обозначения понятий, являющихся элементами системы; определяет совокупность уникальных кодов понятий. Примерами систем кодирования служат МКБ-9, LOINC и SNOMED.

4.12 **коллекция** (collection): Агрегат подобных объектов. В стандарте HL7 используются следующие формы коллекций: множество (set), мультимножество (bag) и список (list). Объекты, которые могут быть организованы в коллекции, включают типы данных.

4.13 **связь** (connection): В информационной модели связью является заданное отношение между двумя классами.

4.14 **тип данных** (data type): Структурный формат данных, передаваемых в атрибуте. Он может ограничивать множество значений, принимаемых атрибутом.

4.15 **отдаленный класс** (distal class): Применительно к классу в информационной модели, это класс на противоположном конце ассоциации между двумя классами.

4.16 **домен** (domain): 1. Конкретная предметная область (например, предметной областью стандартов HL7 является здравоохранение). 2. Множество возможных значений типа данных, атрибута, или компонента типа данных. См. также словарный домен.

4.17 **событие** (event): 1. Воздействие, которое вызывает ощутимое изменение состояния объекта, а также сигнал, воздействующий на поведение объекта. 2. В словарном домене — одно из значений «наклонения» (Mood).

4.18 **квалификатор расширяемости** (extensibility qualifier): Квалификатор словарного домена, используемый в спецификации домена для указания, может ли существующий словарный домен быть расширен дополнительными значениями. У него два возможных значения: CNE (coded, no extension — кодированный без расширения) и CWE (coded with extension — кодированный с расширением).

4.19 **обобщение** (generalization): Связь между двумя классами, называемыми суперкласс и подкласс, в которой подкласс является производным от суперкласса. Подкласс наследует все свойства суперкласса, включая атрибуты, ассоциации и состояния, но может иметь и новые, дополнительные свой-

ства, расширяющие возможности родительского класса. Описывает специализацию с точки зрения подкласса.

4.20 **иерархия обобщений** (generalization hierarchy): Все суперклассы и подклассы с общим корневым суперклассом.

4.21 **графическое представление** (graphical expression): Визуальное представление модели, использующее графические символы для представления компонентов модели и отношений между этими компонентами.

4.22 **Health Level Seven** (HL7): Организация, разрабатывающая стандарты, расположенная в США.

4.23 **идентифицирующий атрибут** (identifier attribute): Атрибут, используемый для идентификации экземпляра класса.

4.24 **информационная модель** (information model): Структурированная спецификация требований к информации в предметной области, выраженная графически и/или повествовательно. Информационная модель описывает классы требуемой информации и свойства этих классов, включая атрибуты, отношения и состояния. Примерами в стандартах HL7 служат эталонная информационная модель предметной области (Domain Reference Information Model), ЭИМ, и Уточненная информационная модель сообщений (Refined Message Information Model).

4.25 **наследование** (inheritance): В отношениях обобщения подкласс наследует все свойства от суперкласса, включая атрибуты, отношения и состояния, если не указано иное.

4.26 **экземпляр** (instance): Вариант или реализация. Например, объект является экземпляром класса.

4.27 **список** (list): Форма коллекции, элементы которой упорядочены и не обязаны быть уникальными.

4.28 **словесное выражение** (literary expression): Текстовое представление модели. Словесное выражение стремится уравновесить потребность в строгом, однозначном описании модели с потребностью в его представлении, которое может быть легко прочитано и интерпретировано людьми, понимающими общие понятия объектно-ориентированных моделей, но, возможно, не обученными формальным языкам определения моделей.

4.29 **обязательный** (mandatory): Если атрибут определен как обязательный, то все элементы сообщения, в которых использован этот атрибут, должны содержать непустое значение либо должны иметь непустое значение по умолчанию.

4.30 **обязательная ассоциация** (mandatory association): Ассоциация, минимальная кратность которой на одном конце больше нуля. Полностью обязательная ассоциация имеет ненулевую кратность на каждом конце.

4.31 **методология** (methodology): Методы или правила, которым надо следовать в конкретной дисциплине.

4.32 **модель** (model): Представление предметной области, использующее абстракцию для выражения релевантных понятий.

4.33 **кратность** (multiplicity): В информационной модели кратность — спецификация минимального и максимального числа объектов каждого класса, которые могут участвовать в ассоциации. Кратность определяется для каждого конца ассоциации.

4.34 **пространство имен** (namespace): Пространство имен — часть модели, в которой определены и используются имена, каждое из которых имеет уникальное значение.

4.35 **объект** (object): Экземпляр класса. Часть информационной системы, содержащей коллекцию связанных данных (в форме атрибутов) и процедур (методов), предназначенных для обработки этих данных.

4.36 **идентичность объекта** (object identity): Свойство существования объекта независимо от любых значений, ассоциированных с объектом.

4.37 **объектный** (object-based): Любой метод, язык или система, поддерживающие идентичность объекта, классификацию и инкапсуляцию. Объектная система не поддерживает специализацию. Ада — пример объектного языка.

4.38 **свойство** (property): Любой атрибут, ассоциация, метод или модель состояний, определенная для класса или объекта.

4.39 **эталонная информационная модель** (ЭИМ): Информационная модель HL7, из которой выводятся все другие информационные модели (например, модели R-MIM) и сообщения.

4.40 **роль** (role): 1. Функция или положение. 2. Класс ЭИМ, который определяет компетенцию класса Entity (сущность). Каждая роль выполняется одним классом Entity (сущность, выполняющая роль) и обычно контролируется другой ролью. 3. В языке UML каждый конец ассоциации обозначается как роль, отражающая функцию класса в ассоциации.

4.41 **имя роли** (role name): См. имя роли ассоциации.

4.42 **множество** (set): Форма коллекции, содержащей неупорядоченный список уникальных элементов одного типа.

4.43 **специализация** (specialization): Ассоциация между двумя классами (называемыми суперкласс и подкласс), в котором подкласс произведен от суперкласса. Подкласс наследует все свойства суперкласса, включая атрибуты, отношения и состояния, но может иметь и новые, дополнительные свойства, расширяющие возможности родительского класса.

4.44 **состояние** (state): Именованное состояние экземпляра класса (объекта), которое может быть проверено с помощью анализа атрибутов и ассоциаций этого экземпляра.

4.45 **атрибут состояния** (state attribute): Атрибут, описывающий текущее состояние объекта.

4.46 **диаграмма перехода состояний** (state diagram): Графическое представление модели перехода состояний, показывающее состояния как вершины (узлы) и переходы состояния как направленные дуги (стрелки) между узлами.

4.47 **машина состояний** (state machine): Описание жизненного цикла экземпляров класса, определяемое моделью переходов состояний.

4.48 **переход состояния** (state transition): Изменение состояния объекта в результате изменения его атрибута или ассоциаций.

4.49 **модель переходов состояний** (state transition model): Графическое представление жизненного цикла класса. Модель изображает все уместные состояния класса, и действительные переходы в зависимости от состояния.

4.50 **подкласс** (subclass): Класс, являющийся специализацией другого класса (суперкласса).

4.51 **предметная область** (subject area): Совокупность классов модели, используемая для деления больших моделей на обозримые подмножества.

4.52 **подсостояние** (sub-state): Идентифицируемое состояние класса, который имеет более специфичное определение, чем его суперсостояние, и полностью входит в это суперсостояние.

4.53 **суперкласс** (superclass): Класс, который является обобщением одного или более других классов (подклассов).

4.54 **суперсостояние** (super-state): Состояние класса, которое охватывает два или более независимых подсостояний.

4.55 **унифицированный язык моделирования** (UML): Язык для создания моделей предметных областей. Язык UML был создан для объединения нескольких известных объектно-ориентированных методологий моделирования, в том числе Booch, Rumbaugh, Jacobson и другие.

4.56 **словарь** (vocabulary): Набор допустимых значений кодированного атрибута или поля.

4.57 **словарный домен** (vocabulary domain): Набор всех понятий, которые могут рассматриваться как допустимые значения кодированного атрибута или поля; ограничение, применяемое к кодированным значениям.

4.58 **квалификатор словарного домена** (vocabulary domain qualifier): Часть спецификации словарного домена. Двумя существующими квалификаторами являются расширяемость (extensibility) и сфера действия (realm).

4.59 **W3C** (The World Wide Web Consortium): Консорциум World Wide Web, международный промышленный консорциум

5 Интерпретация спецификации

5.1 Содержание спецификации

ЭИМ состоит из классов, включенных в один или несколько пакетов предметных областей. Атрибуты, отношения и машины состояний связаны с классами.

Каждый класс ЭИМ представляет информацию о понятии, которое должно быть передано в сфере здравоохранения. Имена классов произведены от обычного (английского) языка, но по необходимости они ограничены «пространством имен» ЭИМ. Значение этих классов полностью охвачено определением класса и определениями его свойств (атрибутов и ассоциаций) этого класса. Например, для понимания значения класса Role (роль) достаточно ознакомиться с его определением и определениями его свойств. В контексте пространства имен ЭИМ не являются релевантными определения имени, взятые из другого контекста или из словаря.

ЭИМ представлена на языке UML, расширенном с помощью тегов, специфичных для стандартов HL7 и включенных в метаданные элементов UML-модели. Все стандартные значения метаданных элементов модели UML нормативны, но нормативны также и следующие расширения HL7:

- Class.stateAttribute;
- Class.classCode;
- Attribute.mandatoryInclusion;
- Attribute.cardinality;
- Attribute.vocabDomain;
- Attribute.vocabStrength.

5.2 Понимание ЭИМ

ЭИМ использует очень абстрактный стиль моделирования. Ее ядром служат базовые классы ЭИМ и их структурные атрибуты. Понимание этих классов и атрибутов существенно для понимания ЭИМ. В настоящем разделе описано, каким образом абстракции моделируются на языке UML и контролируются с помощью словаря, являющегося частью настоящего документа. В приложении Б приведено резюме (высокоуровневое руководство), содержащее примеры использования этих абстракций для представления более детальной медицинской информации.

5.2.1 ЭИМ как абстрактная модель

ЭИМ содержит 6 «базовых» классов:

- класс Act (действие), представляющий действия, которые выполняются и должны быть документированы при оказании медицинской помощи,

- класс Participation (участие), представляющий контекст действия, а именно, кто выполнил действие, для кого оно было выполнено, где было выполнено и т. д.;

- класс Entity (сущность), представляющий физические предметы и существа, которые используются при оказании медицинской помощи и принимают участие в ее оказании;

- класс Role (роль), представляющий роли, выполняемые сущностями, участвующими в действиях по оказанию медицинской помощи;

- класс ActRelashionship (связь действий), представляющий связь одного действия с другим, например, связь между направлением на исследование и состоявшимся исследованием;

- класс RoleLink (связь ролей), представляющий отношения между отдельными ролями.

Три из этих классов — Act, Entity и Role (действия, сущности и роли) — детализируются в виде множества классов-специализаций, или подтипов. В представлении стандарта HL7 подтипы добавляются к ЭИМ только в том случае, если требуется определить один или более атрибутов или ассоциаций, которые не могут быть унаследованы от родительского класса. Классы, описывающие отдельными понятия, не нуждающиеся ни в каких дальнейших атрибутах или ассоциациях, представлены исключительно как уникальный код в контролируемом словаре. Поэтому эти три базовых класса имеют следующие кодируемые атрибуты, используемые для дальнейшего определения моделируемого понятия:

- classCode (в классах Act, Entity и Role), уточняющий назначение класса или соответствующего ему понятия независимо от того, представлен ли он как класс в иерархии ЭИМ;

- moodCode (в классе Act) и determinerCode (в классе Entity), с помощью которых можно указать, представляет ли класс экземпляр или разновидность класса Act или Entity. Если класс является специализацией класса Act, то атрибут moodCode позволяет указать, описывает ли экземпляр класса Act свершившееся или планируемое действие;

- code (в классах Act, Entity и Role), обеспечивающий более детальную классификацию значения атрибута classCode, например, конкретный вид исследования в классе Observation (исследование).

Другие три базовых класса ЭИМ — Participation, ActRelationship и RoleLink — не представлены иерархиями обобщения-специализации. Тем не менее эти классы представляют разнообразные понятия, например, разные формы участия или разные типы отношений между действиями. Эти отличия представлены атрибутом typeCode, который должен быть определен в каждом из этих классов.

5.2.2 Представление структуры класса ЭИМ

Как упоминалось ранее, ЭИМ сконструирована с использованием подмножества семантики языка UML. ЭИМ представляет собой набор UML-классов, содержащих один или более атрибутов, которым присвоены типы данных, основанные на независимой спецификации типов данных. Классы связаны рядом отношений ассоциации, идентифицируемых уникальными именами ролей, или отношениями обобщения-специализации.

Каждый из этих элементов имеет текстовое определение. Внешнее представление атрибутов и ассоциаций управляется кратностью и другими связанными ограничениями, применяемыми к атрибутам и ролям, привязывающим ассоциации к классам.

5.2.3 Представление контролируемого словаря

Несколько атрибутов в ЭИМ имеют тип данных CS. Это означает, что множество значений такого атрибута должно быть выбрано из множества кодов, определенных в стандарте HL7. Упомянутые ранее атрибуты classCode и typeCode являются примерами атрибутов с типом данных CS.

Все множества кодированных значений этих атрибутов являются частью настоящего документа и принимаются в соответствии с теми же принципами голосования, что и классы ЭИМ. Каждое множество кодов представлено как словарный домен, т. е. множество всех понятий, которые могут использоваться как допустимые значения кодированного поля или атрибута. Важно отметить, что словарный домен состоит из множества понятий, а не слов или кодов.

5.2.4 Связанные спецификации

Как отмечено ранее, каждому атрибуту в ЭИМ присвоен тип данных. Формальная спецификация этих типов данных зависит от того, используется ли настоящая модель комитетом HL7 или комитетом ИСО ТК215. При использовании комитетом HL7 нормативной спецификацией типов данных служит документ «HL7 Data Types Abstract Specification» (Абстрактная спецификация типов данных HL7). При использовании комитетом ИСО ТК215 документы, специфицирующие релевантные типы данных, имеют обозначение ИСО 22??xx??. Справочная таблица свойств релевантных типов данных включена в Приложение В.

5.3 Графические диаграммы нормативного содержания ЭИМ

Классы, включенные в нормативное содержание ЭИМ, представлены на рисунках 1—4.

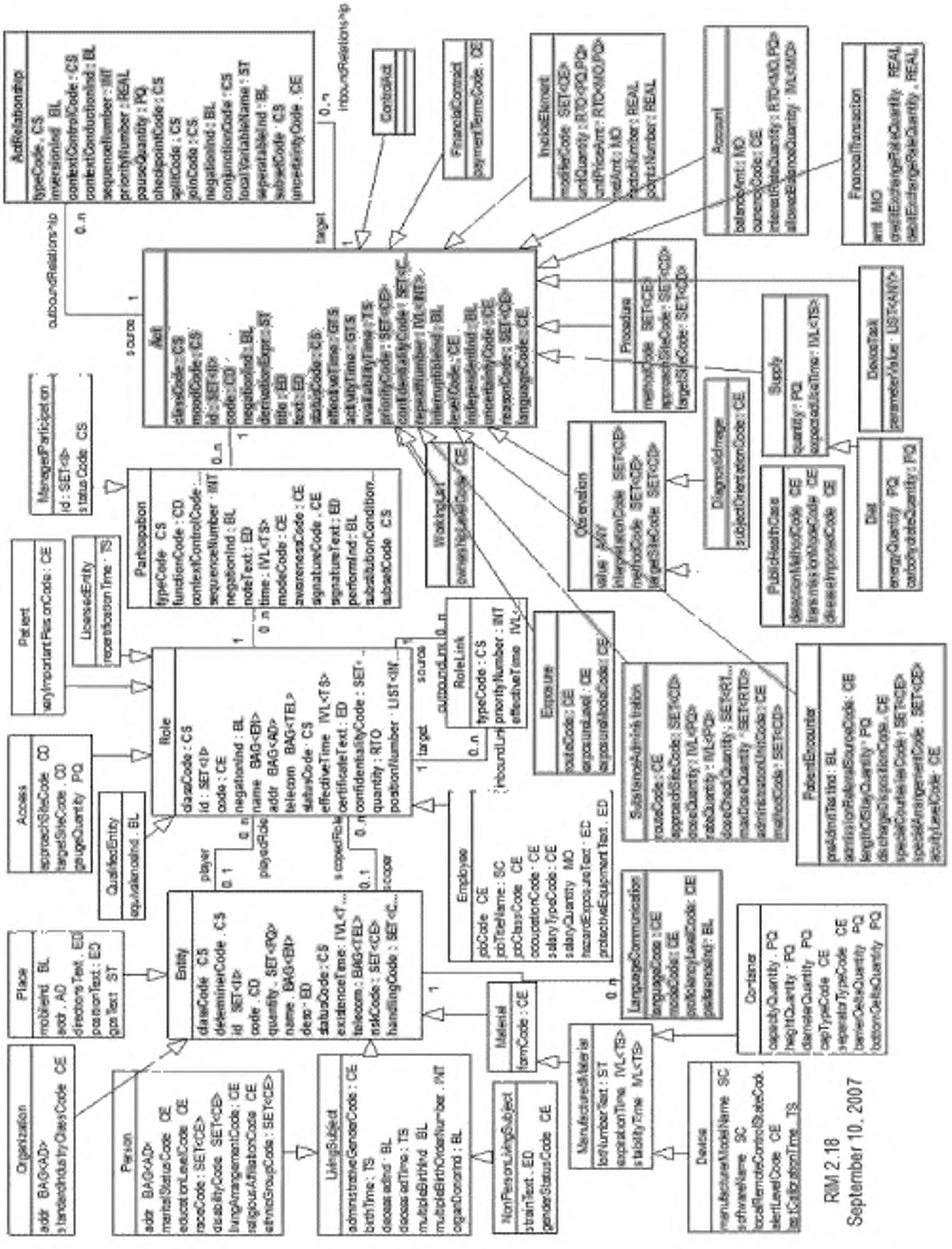

Рисунок 1 — Диаграмма классов всех предметных областей

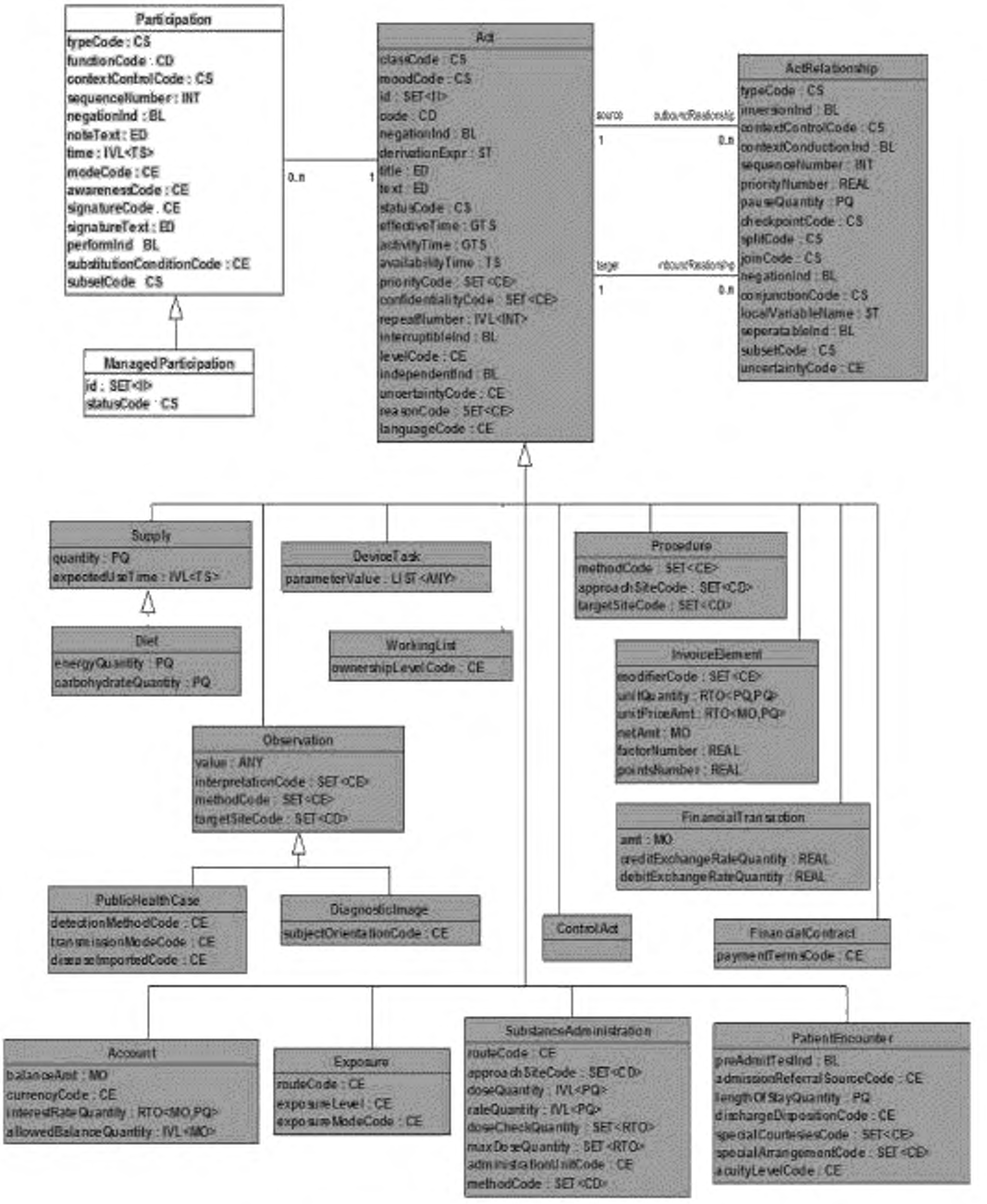

Рисунок 2 — Диаграмма классов предметной области Acts (действия)

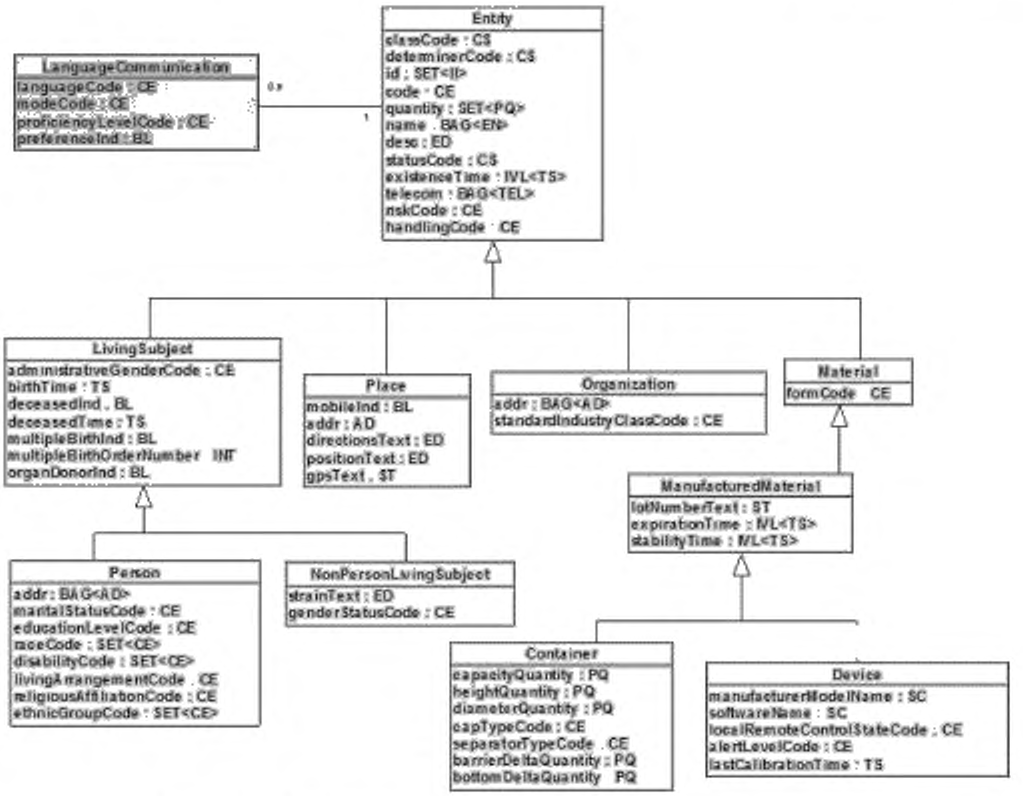

Рисунок 3 — Диаграмма классов предметной области Entities (сущности)

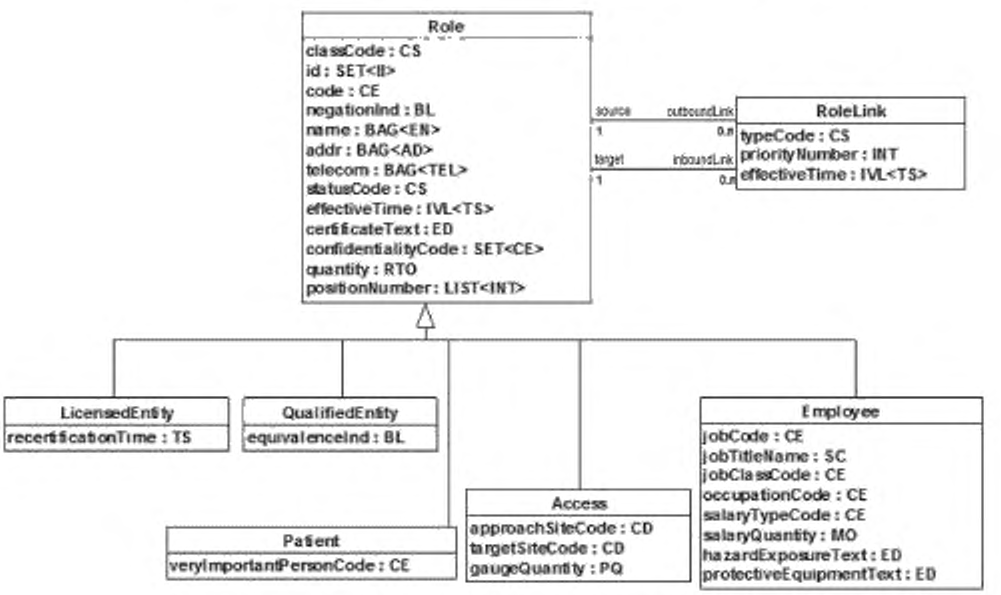

Рисунок 4 — Диаграмма классов предметной области Roles (роли)

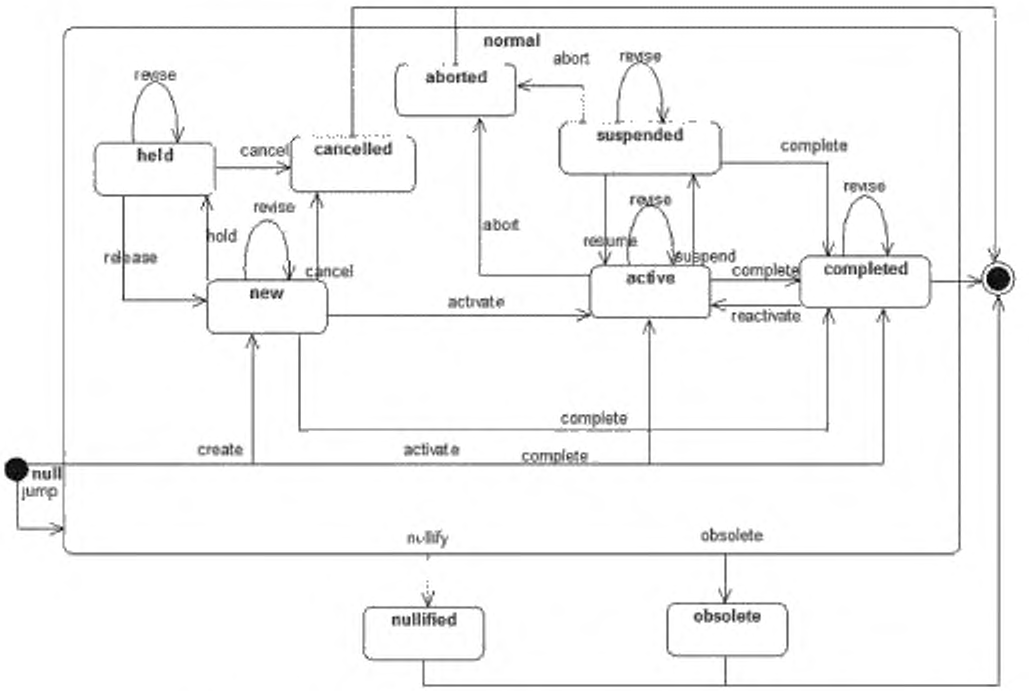

Рисунок 5 — Диаграмма перехода состояний класса Act (действие)

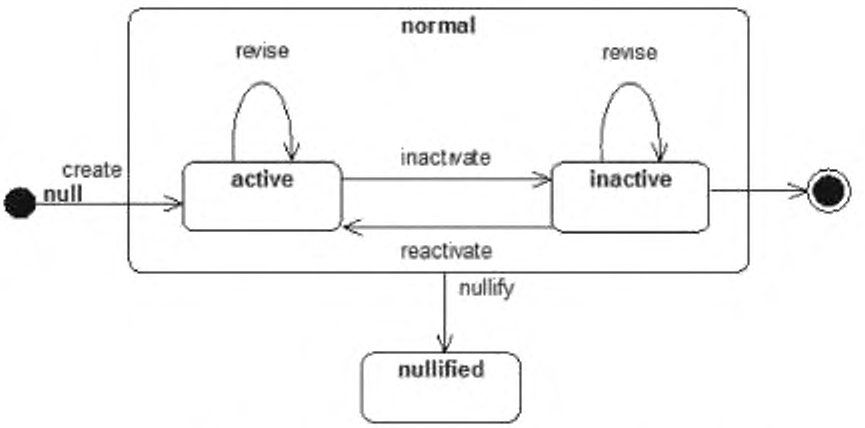

Рисунок 6 — Диаграмма перехода состояний класса Entity (сущность)

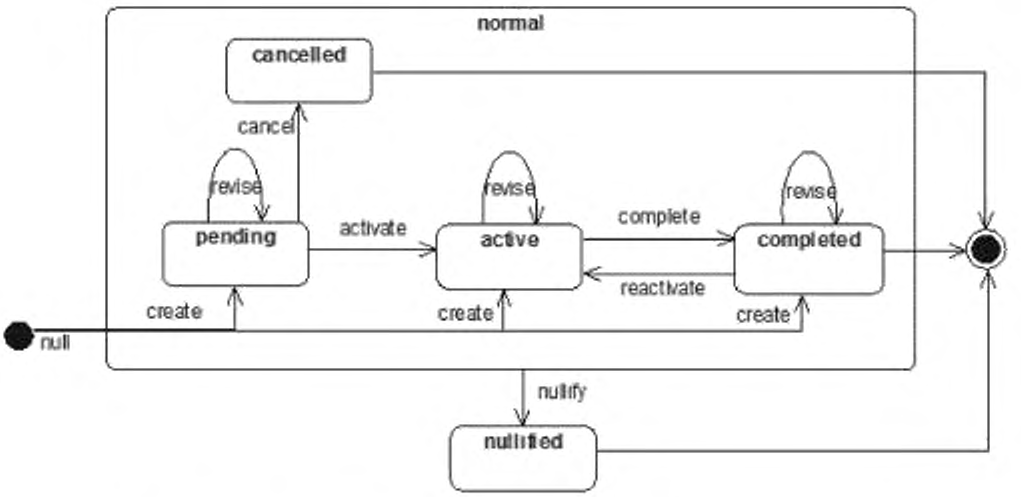

Рисунок 7 — Диаграмма перехода состояний класса ManagedParticipation (управляемое участие)

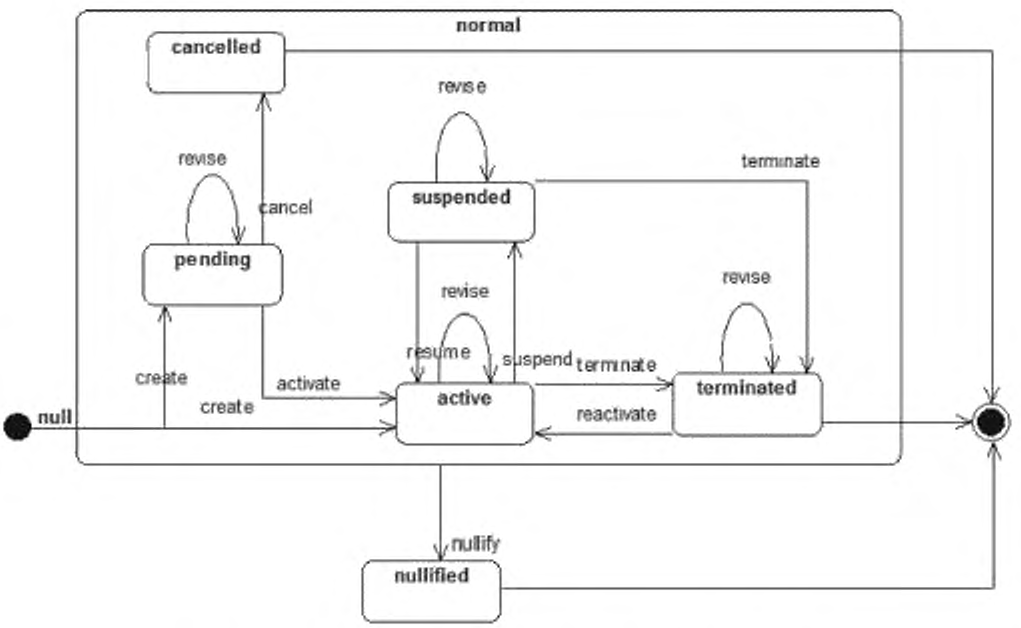

Рисунок 8 — Диаграмма перехода состояний класса Role (роль)

6 Предметные области ЭИМ HL7

6.1 Предметная область FoundationClasses

Эта совокупность классов и их ассоциаций представляет нормативное содержание HL7 ЭИМ. Содержание данной предметной области утверждается в комитете HL7 как нормативный документ.

Диаграмма классов этой предметной области показана на рисунке 5.3.

Предметная область FoundationClasses содержит следующие предметные области:

- Acts (действия)
- Entities (сущности)
- Roles (роли)

6.1.1 Предметная область Acts (в предметной области FoundationClasses)

Совокупность классов, включая класс Act и его специализации. Они относятся к действиям и событиям, связанным с оказанием медицинской помощи.

Диаграмма классов этой предметной области показана на рисунке 2.

Предметная область Acts содержит следующие классы:

- Account (счет);
- Act (действие);
- ActRelationship (связь действий);
- ControlAct (управляющее действие):
- DeviceTask (функция устройства);
- DiagnosticImage (диагностическое изображение);
- Diet (диета);
- Exposure (воздействие),
- FinancialContract (контракт);
- FinancialTransaction (финансовая операция);
- InvoiceElement (элемент счета-фактуры);
- ManagedParticipation (управляемое участие);
- Observation (исследование);
- Participation (участие);
- Procedure (процедура);
- PublicHealthCase (событие общественного здоровья);
- SubstanceAdministration (лекарственное назначение);
- Supply (предоставление материала);
- WorkingList (рабочий лист).

6.1.2 Предметная область Entities (в предметной области FoundationClasses)

Совокупность классов, содержащая класс Entity, его специализации и связанные с ними квалифицирующие классы. Эти классы представляют участников медицинской помощи и другие сущности здравоохранения.

Диаграмма классов этой предметной области показана на рисунке 3.

Предметная область Entities содержит следующие классы:

- Container (контейнер);
- Device (устройство);
- Entity (сущность);
- LanguageCommunication (вербальное общение);
- LivingSubject (живой организм);
- ManufacturedMaterial (изготовленный материал);
- Material (материал);
- NonPersonLivingSubject (нечеловеческое живое существо);
- Organization (организация);
- Person (лицо);
- Place (место).

6.1.3 Предметная область Roles (в предметной области FoundationClasses)

Совокупность классов, содержащая класс Role и его специализации. Эти классы представляют роли участников процесса оказания медицинской помощи.

Диаграмма классов этой предметной области показана на рисунке 4.

Предметная область Roles содержит следующие классы:
- Access (доступ);
- Employee (работник);
- LicensedEntity (лицензированный субъект);
- Patient (пациент);
- Role (роль);
- RoleLink (связь роли).

7 Классы ЭИМ HL7

Далее перечислены все классы ЭИМ. Порядок сортировки основан на следующих трех критериях:
- нормативное содержание;
- имя основной предметной области (в алфавитном порядке);
- имя класса (в алфавитном порядке).

7.1 Классы предметной области Acts

7.1.1 Класс Act (в предметной области Acts)

Код класса: ACT.

Атрибуты класса Act:
- classCode:: CS
- moodCode:: CS
- id:: SET<II>
- code:: CD
- negationInd:: BL
- derivationExpr:: ST
- title:: ED
- text:: ED
- statusCode:: CS
- effectiveTime:: GTS
- activityTime:: GTS
- availabilityTime:: TS
- priorityCode:: SET<CE>
- confidentialityCode:: SET<CE>
- repeatNumber:: IVL<INT>
- interruptibleInd:: BL
- levelCode:: CE
- independentInd:: BL
- uncertaintyCode:: CE
- reasonCode:: SET<CE>
- languageCode:: CE

Ассоциации класса Act:

- outboundRelationship::(0..*) ActRelationship::source::(1..1) (ассоциация с классом ActRelationship, роль source — источник);

- inboundRelationship::(0..*) ActRelationship::target::(1..1) (ассоциация с классом ActRelationship, роль target — цель);

- participation::(0..*) Participation::act::(1..1) (ассоциация с классом Participation, роль act — действие).

Класс Act является обобщением следующих классов:
- Account (счет);
- ActRelationship (связь действий);
- ControlAct (управляющее действие);
- DeviceTask (функция устройства);
- DiagnosticImage (диагностическое изображение);
- Diet (диета);
- Exposure (воздействие);
- FinancialContract (контракт);

- FinancialTransaction (финансовая транзакция);
- InvoiceElement (элемент счета);
- ManagedParticipation (управляемое участие);
- Observation (исследование);
- Participation (участие);
- Procedure (процедура);
- PublicHealthCase (событие общественного здоровья);
- SubstanceAdministration (лекарственное назначение);
- Supply (предоставление материала);
- WorkingList (рабочий лист).

Определение класса Act: запись о чем-то, что делается, что было сделано, что может быть сделано, что планируется или что требуется сделать.

Примеры: видами действий, типичными для здравоохранения, являются:

- клиническое исследование;
- оценка состояния здоровья (например, жалобы и диагнозы);
- цели медицинской помощи;
- услуги лечения (например, лекарственная терапия, хирургическое лечение, физиотерапия и психотерапия);
- ассистирование, мониторинг, ведение пациента;
- санитарное просвещение пациентов и их близких лиц;
- услуги нотариуса (например, упреждающие указания или отказ от искусственного поддержания жизни);
- редактирование и обработка документов, и т. д.

Обсуждение и обоснование: классы действий являются краеугольным камнем ЭИМ; информация всех предметных областей и процессов, в основном, сосредоточена в классах действий. Любая профессия или деятельность, включая здравоохранение, состоит из намеренных действий, выполняемых и регистрируемых ответственными лицами. Экземпляр класса действия представляет собой запись о таком намеренном действии. Намеренные действия отличаются от тех, что происходят в силу естественных причин (естественные события). Такие естественные события не являются действиями как таковыми, но могут регистрироваться как экземпляры класса Observation (исследования).

Соединение классов действий Act с классами сущностей Entity осуществляется с помощью классов участия Participation и ролей Role. Соединение одного класса действий Act с другим классом Act осуществляется с помощью класса связи действий ActRelationship. Классы участия Participation представляют авторов, исполнителей и другие ответственные стороны, а также субъектов и бенефициариев (в том числе инструменты и материал, используемые при выполнении действия, которые также являются субъектами). Использование атрибута moodCode позволяет различить фактически выполненные действия, запланированные и затребованные действия, а также другие варианты определенности действия.

Каждый класс действия Act имеет (в крайнем случае, косвенно) связь с одним из классов Participation, представляющим основного автора, ответственного за выполнения действия и «владеющего» действием. Ответственность за действие означает ответственность за то, что оно означает, и как это записано. А владелец действия имеет право его изменить в рабочем порядке. Ответственность за действие и владение действием, представленным классом Act, не идентичны ответственности и владению по отношению к тому, что соответствует экземпляру этого класса в реальном мире. Одна и та же деятельность, происходящая в реальном мире, может быть описана с точки зрения двух людей, каждый из которых является автором своего экземпляра класса Act, содержащего такое описание. Но один из этих людей может быть свидетелем деятельности, а другой — ее основным исполнителем. Исполнитель несет ответственность за физические действия; свидетель отвечает только за достоверное описание этих действий в силу своих способностей. Эти два экземпляра класса Act могут даже не быть согласованы между собой, но поскольку у каждого из них есть свой автор, такие разногласия могут сосуществовать и рассматриваться получателем этих экземпляров класса Act.

В этом отношении экземпляр класса Act представляет собой «утверждение» в терминах Rector и Nowlan (1991) [Foundations for an electronic medical record. Methods Inf Med. 30]. Rector и Nowlan подчеркнули важность понимания медицинской карты не как собрания фактов, а как «достоверного отчета о том, что слышали, видели, думали и делали медицинские работники». Rector и Nowlan продолжают эту мысль, заявляя, что «другие требования, предъявляемые к ведению медицинской карты, например, что она долж-

на иметь определенную принадлежность и обладать свойством постоянства, естественно вытекают из этой точки зрения». Действительно, класс Act как раз и представляет собой такое «утверждение», содержащее атрибуты, и правила обновления экземпляров класса Act (обсуждаемые в модели перехода состояний, см. атрибут Act.statusCode), применяемые вместо создания новых экземпляров этого класса, разработаны в соответствии с принципом постоянства и принадлежности.

Rector и Nowlan описывают электронную медицинскую карту как совокупность утверждений, которые имеют определенную принадлежность, но ограничиваются изложением фактов. Однако класс Act выходит за это ограничение, относящееся к фактическим утверждениям определенной принадлежности, представляя то, что известно в лингвистике и философии как «речевые акты».

Понятие речевого акта означает, что речевое выражение, кроме изложения факта, имеет определенное прагматическое значение, и что в реальном мире речевые выражения используются для изменения состояния дел, в том числе, чтобы непосредственно инициировать физическую деятельность. Например, заказ исследования представляет собой речевой акт, который (при условии, что оно адекватно) вызывает физическое выполнение заказанного действия. Кульминацией теории речевых актов является плодотворная работа Austin (1962) [How to do things with words. Oxford University Press].

В реальном мире отдельная деятельность может развиваться от ее формулировки до выполнения, проходя этапы планирования и указаний. Отражение этих этапов находит свое представление с помощью атрибута moodCode класса Act. Хотя такую деятельность и можно рассматривать в развитии от плана до выполнения, это развитие должно представляться в форме нескольких экземпляров класса Act, каждый из которых имеет ровно одно значение атрибута moodCode, которое не меняется в течение всего срока жизни этого экземпляра. Это связано с тем, что принадлежность и содержание речевых актов в процессе деятельности могут быть различными, и нередко очень важно вести достоверную и постоянно хранящуюся регистрацию этого процесса. Спецификация указаний, обещаний или планов не должна заменяться на спецификацию того, что было выполнено на самом деле, чтобы иметь возможность сравнить результат с более ранними спецификациями. Экземпляры класса Act, являющиеся отражением развития одной деятельности, должны быть связаны между собой с помощью экземпляров класса ActRelationship, у которых атрибут typeCode имеет значение SEQL (продолжение).

Экземпляры класса Act, представляющие утверждения или речевые акты, являются единственным способом представления фактов реального мира или процессов в HL7 ЭИМ. Правда о реальном мире конструируется только с помощью комбинаций (или выбора) таких утверждений, имеющих определенную принадлежность, и в ЭИМ нет ни одного класса, экземпляры которого представляли бы «реальное положение дел» или «реальные процессы» независимо от этих утверждений. В силу этого обстоятельства не проводится никаких различий между деятельностью и ее описанием. Каждый экземпляр класса Act характеризует и то, и другое в различной степени. Например, фактическое утверждение о недавних (но уже завершенных) действиях, сделанное (и подписанное) исполнителем этих действий, обычно известное как протокол или первичная документация (например, протокол хирургической операции, дневниковые записи и т.д.). А изменение состояния текущей деятельности, документируемое исполнителем (или непосредственным наблюдателем), рассматривается как сбор информации об этой деятельности (которая позже заменяется полным протоколом процедуры). Однако и обновление состояния, и протокол процедуры представляются экземплярами класса Act одного рода, которые можно различить по значениям атрибутов наклонения (moodCode) и состояния (statusCode) и по завершенности информации.

В следующих подпунктах описаны атрибуты класса Act.

7.1.1.1 Act.classCode:: CS (1..1) Mandatory

Словарный домен: ActClass (CNE)

Определение: Код, устанавливающий главный тип класса Act, представляющий данный экземпляр этого класса.

Ограничения: домен classCode представляет собой строго контролируемый словарь, который не может быть внешним или пользовательским.

Каждый экземпляр класса Act должен иметь атрибут classCode. Если этот класс не специализирован, то атрибут Act.classCode должен иметь наиболее общее значение (ACT).

Значение атрибута Act.classCode должно быть обобщением определенного понятия (например, заданного значением атрибута Act.code), другими словами, понятия, моделируемые классом Act и передаваемые в его экземпляре, должны быть специализациями класса понятий, заданного значением атрибута Act.classCode. В частности, атрибут classCode не является «модификатором» атрибута кода класса Act.code. (Для сравнения см. описание атрибута Act.code).

7.1.1.2 Act.moodCode:: CS (1..1) Mandatory

Словарный домен: ActMood (CNE)

Определение: Код, позволяющий различить, что именно представляет данный экземпляр класса Act: фактическое утверждение, команду, возможность, цель и т. д.

Ограничение: Экземпляр класса Act должен иметь одно и только одно значение атрибута moodCode.

Значение атрибута moodCode конкретного экземпляра класса Act никогда не изменяется. Стадия деятельности, характеризуемая этим атрибутом, не является состоянием объекта.

Чтобы описать развитие конкретной деятельности от ее плана до выполнения, необходимо создать несколько экземпляров классов Act, имеющих разные значения атрибута moodCode, и связать их между собой с помощью экземпляров класса ActRelationship, у которых атрибут typeCode имеет значение SEQL (продолжение).

Обсуждение: Значение атрибута Act.moodCode описывает следующие понятия: (1) событие, т. е. фактическое описание произошедших действий; (2) определение возможных действий и планов действия (на уровне нормативно-справочного файла); (3) намерение, т. е. план действий, который для пациента выражен в форме плана лечения или направления; (4) цель, т. е. желательный результат медицинской помощи пациенту; (4) задача, т. е. желаемый результат, приложенный к проблемам пациента и планам; и (5) критерий, т. е. предикат, используемый для вычисления логического выражения.

Значение атрибута Act.moodCode контролируемым образом изменяет смысл класса Act подобно тому, как в естественном языке грамматическая форма глагола определенным образом изменяет смысл предложения. Например, если значение атрибута Act.moodCode является признаком фактического события, то весь экземпляр класса Act представляет известный факт. Если оно является признаком плана (намерения), то весь экземпляр класса Act представляет описание того, что должно быть сделано. Значение атрибута Act.moodCode не меняет каким-либо особым способом конкретные свойства класса Act.

Поскольку значение атрибута Act.moodCode определяет смысл экземпляра класса Act, то оно должно быть всегда известно. Это означает, что всякий раз, когда создается экземпляр класса Act, его атрибуту moodCode должен быть присвоен допустимый код, который не может меняться в течение всего срока жизни этого экземпляра.

Поскольку смысл экземпляра класса Act задается кодом, присвоенным атрибуту moodCode, то значение этого кода влияет на интерпретацию всего этого экземпляра, включая каждое его свойство (атрибут или ассоциацию). Коль скоро значение атрибута moodCode влияет на интерпретацию экземпляра класса, то смысл этого экземпляра, в свою очередь, влияет на смысл его атрибутов. Однако значение атрибута moodCode не может оказать непосредственное влияние на смысл отдельного атрибута.

Классы Act имеют два типа свойств действий — инертные и описательные. Смысл инертных свойств не зависит от значения атрибута Act.moodCode, а интерпретация описательных свойств зависит. Например, у класса Act есть атрибут Act.id, который обеспечивает уникальную идентификацию экземпляра этого класса. Уникальная идентификация объекта никоим образом не зависит от значения атрибута Act.moodCode. Поэтому «интерпретация» идентификатора Act.id является инертной по отношению к атрибуту Act.moodCode.

Напротив, большинство из других атрибутов класса Act является описательным по отношению к утверждению, передаваемому в форме экземпляра класса Act. Эти атрибуты дают ответы на вопросы, кто выполнил действие, для кого, где, что использовал, как и когда было выполнено это действие. Ответы на вопросы, кто, для кого, где, что использовал, передаются в описательных атрибутах классов Paricipation, а ответы на вопросы, как и когда — в описательных атрибутах и экземплярах классов ActRelationship.

Для иллюстрации влияния атрибута moodCode ниже рассмотрены экземпляры класса Act, относящиеся к процессу определения сахара крови.

Экземпляр специализации класса Act, у которого атрибут moodCode имеет значение DEF (definition — описание), содержит справочное описание процесса «определение сахара крови». Связанные с ним экземпляры классов Participation содержат характеристики субъектов, которые должны участвовать в этом процессе, и требуемых для него объектов, например, биоматериал, подразделение, лабораторное оборудование и т. д. Значение атрибута Observation.value указывает абсолютный диапазон значений (домен) результата анализа (например, «15—500 мг/дл»).

Если атрибут moodCode имеет значение INT (intent — намерение), это означает, что автор, указанный в экземпляре класса, намерен назначить анализ концентрации сахара в крови («надо определить сахар крови»). Связанные с ним экземпляры классов Participation содержат информацию о тех субъектах и объектах, которые фактически или предположительно участвуют в этом назначении, в первую очередь об авторе намерения или о любом отдельном лице при групповом намерении, а также о передаваемом биоматериале,

о требованиях к лабораторному оборудованию и т. д. Атрибут Observation.value в этом случае обычно отсутствует, поскольку речь идет о намерении провести анализ концентрации сахара, а не измерить концентрацию сахара в указанном диапазоне значений. (Иная ситуация будет, если атрибут moodCode имеет значение GOL.)

Экземпляр специализации класса Act, у которого атрибут moodCode имеет значение RQO (request — требование, что можно рассматривать как разновидность намерения), содержит направление на анализ концентрации сахара в крови («определите сахар крови»). Связанные с ним экземпляры классов Participation содержат информацию о субъектах и объектах, которые фактически или предположительно должны участвовать в процессе выполнения анализа, в первую очередь о заказчике анализа и о выбранном исполнителе, а также о передаваемом биоматериале, о требованиях к лабораторному оборудованию и т. д. Атрибут Observation.value в этом случае обычно отсутствует, поскольку речь идет о намерении провести анализ концентрации сахара, а не измерить концентрацию сахара в указанном диапазоне значений.

Экземпляр специализации класса Act, у которого атрибут moodCode имеет значение EVN (event — событие), содержит результат определения сахара крови ("сахар крови определен"). Связанные с ним экземпляры классов Participation содержат информацию о субъектах и объектах, фактически участвовавших в процессе определения (включая биоматериал, подразделение, лабораторное оборудование). Атрибут Observation.value содержит измеренное значение (например, «80 мг/дл» или «<15 мг/дл»).

Если атрибут moodCode имеет значение EVN.CRT (event-criterion — критерий события), это означает, что автор, указанный в экземпляре класса, рассматривает некоторый класс процессов "определения сахара крови", возможно, с определенным критерием оценки (диапазоном). Связанные с ним экземпляры классов Participation содержат критерий, применяемый, например, к пациенту. Атрибут Observation.value содержит диапазон значений критерия (например, «>180 мг/дл» или «200—300 мг/дл»).

Экземпляр специализации класса Act, у которого атрибут moodCode имеет значение GOL (goal — цель, что можно рассматривать как разновидность критерия), содержит информацию о цели, которую требуется достичь («целью является определенный уровень (диапазон) концентрации сахара в крови»). Связанные с ним экземпляры классов Participation содержат информацию, близкую к той, что была указана в намерении определения сахара крови, в первую очередь сведения об авторе цели и о пациенте, по отношению к которому эта цель поставлена. Атрибут Observation.value содержит целевой диапазон значений (например, «80—120 мг/дл»).

Объяснение: смысл атрибута moodCode заимствован от наклонения глагола в грамматике естественного языка (лат. modus verbi).

Применение атрибута moodCode напоминает также различные расширения логики фактов в модальной логике и логике с модальностями. Значение этого атрибута эквивалентно модальности (факт, возможность, намерение, цель и т. д.), которая является или не является подходящей для «утверждения», передаваемого в экземпляре класса Act.

7.1.1.3 Act.id:: SET<II> (0..*)

Определение: уникальный идентификатор экземпляра класса Act.

7.1.1.4 Act.code:: CD (0..1)

Словарный домен: ActCode (CWE)

Определение: код, указывающий конкретный вид действия, представленного экземпляром класса Act.

Ограничения: для указания вида действия (например, физикальное исследование, определение калия в сыворотке крови, госпитализация, финансовая транзакция по оплате лечения и т. д.) используется код, который берется из какой-либо (обычно внешней) системы кодирования. Система кодирования зависит от конкретной специализации класса Act, например, для класса Observation, описывающего исследования, может использоваться система кодирования LOINC, и т. д.

Концептуально значение атрибута Act.code должно быть специализацией значения атрибута Act.classCode. Поэтому структура словарного домена ActClass должна найти свое отражение на верхнем уровне структуры словарного домена ActCode и отдельные коды или внешние словари должны быть подчинены структуре словарного домена ActClass.

Атрибуты Act.classCode и Act.code не являются модификаторами друг для друга, однако понятие, передаваемое в атрибуте Act.code, должно логически вытекать из понятия, передаваемого в атрибуте Act.classCode. Негативным примером служит использование атрибута Act.code для передачи понятия «калий» одновременно в экземпляре класса Observation, у которого атрибут Act.classCode имеет значение SPCOBS (specimen observation — лабораторное исследование биоматериала), чтобы он означал «лабораторное исследование содержания калия», и в экземпляре класса Medication, у которого атрибут Act.classCode

имеет значение SBADM (substance administration — лекарственное назначение), чтобы он означал "замещение калия". Такое взаимное изменяющее использование сочетаний атрибутов Act.classCode и Act.code не допускается.

Обсуждение: атрибут Act.code не является обязательным в классе Act. Вместо конкретизации вида действия с помощью атрибута Act.code можно воспользоваться атрибутом Act.classCode и другими свойствами класса Act. Более общий и чаще встречающийся прием состоит в задании вида действия с помощью экземпляра класса Act, в котором атрибут Act.moodCode имеет значение DEF, связанного с другим экземпляром класса Act с помощью экземпляра класса ActRelationship. Вид действия без труда можно указать и без такой привязки к его определению, используя другие атрибуты, а также классы ActRelationship и Participation. Например, вид лекарственного назначения, передаваемого в экземпляре класса SubstanceAdministarion, можно указать с помощью ассоциации ActRelationship с экземпляром класса Entity, содержащим информацию о конкретном лекарстве.

7.1.1.5 Act.negationInd:: BL (0..1)

Определение: индикатор, указывающий, что к описательным атрибутам утверждения, передаваемого в экземпляре класса Act, должно быть применено отрицание.

Примеры: использование этого индикатора в экземпляре класса Observation, у которого атрибут Act.moodCode имеет значение EVN (событие), позволяет передать утверждение «у пациента НЕТ боли в груди». А в экземпляре класса Observation, у которого атрибут Act.moodCode имеет значение EVN.CRT (критерий), использование идентификатора отрицания позволяет передавать критерии выполнения действия вида «если у пациента НЕТ боли в груди в течение 3 дней...», или «если систолическое давление НЕ находится в пределах 90—100 мм рт. ст…».

Обсуждение: атрибут negationInd используется как отрицание квантора существования. Его смысл лучше всего объяснять для экземпляров класса Act, у которых атрибут Act.moodCode имеет значение EVN.CRT (критерий). Если критерий используется без индикатора отрицания, то обычно в экземпляре класса Act надо передать только несколько тех критичных атрибутов и связей (параметров), которые нужны для проверки критерия. Чем больше параметров указано, тем более специфичным (ограниченным) будет критерий. Например, для задания критерия «систолическое давление находится в пределах 90—100 мм рт. ст.» в экземпляре класса Observation достаточно указать описательные атрибуты Act.code, указывающий «измерение кровяного давления», и Observation.value, указывающий диапазон значений «90—100 мм рт. ст.». Если будет указан еще и атрибут effectiveTime, скажем, со значением «вчера», то критерий станет более узким. Если при таком критерия атрибут negationInd будет иметь значение TRUE (истина), то критерий приобретет следующий смысл: «не существует вчерашнего значения систолического кровяного давления в диапазоне 90—100 мм рт. ст.» (вне зависимости от того, измерялось вчера кровяное давление или нет).

Значение атрибута negationInd воздействует указанным ранее образом на описательные атрибуты класса Act (включая Act.code, Act.effectiveTime, Observation.value, Act.doseQty, и т. д.), и на любые его компоненты. Инертные свойства, например, Act.id, Act.moodCode, Act.confidentialityCode, и особенно ассоциация с классом Participation, имеющая роль автора, остаются неизменными. Эти инертные свойства всегда имеют одно и то же значение: т. е., автор остается и автором отрицательного исследования. Кроме того, атрибут negationInd не воздействует и на большинство связей ActRelationship (за исключением компонентов).

Например, крайне конфиденциальное указание, записанное д-ром Джонсом в форме «не применять сукцинилхолин» в связи (класс ActRelationship) с имевшейся злокачественной гипертермией (класс Observation), является отрицанием положительного указания «применить сукцинилхолин» (атрибут Act.code), но тем не менее остается указанием, написанным д-ром Джонсом для пациента Джона Смита, и причина этого указания — имевшаяся у пациента злокачественная гипертермия.

Однако дополнительные детали, передаваемые в описательных атрибутах, будут частью отрицания, ограничивая его воздействие. Например, если в указании не применять субстанцию присутствует атрибут дозы doseQuantity, то это означает, что нельзя давать эту конкретную дозу субстанции (но любая другая доза могла бы оставаться допустимой).

Экземпляр класса Act, у которого атрибут negationInd имеет значение TRUE, тем не менее, остается утверждением об определенном факте, описанном в этом экземпляре. Например, отрицание утверждения «1 июля выявлена одышка» означает, что его автор определенно отрицает, что 1 июля была одышка, и что он несет ту же самую ответственность за это утверждение и те же самые требования к его доказательству, как если бы он не использовал отрицание. И наоборот, признак отрицания, переданный в атрибуте negationInd, никоим образом не отрицает того, что факт был подтвержден или что утверждение имело место. Это равным образом относится ко всем наклонениям утверждения, задаваемым атрибутом moodCode, например, при-

менение отрицания к направлению является указанием не делать того, что в нем написано, а вовсе не лапидарное утверждение, что такого направления нет.

7.1.1.6 Act.derivationExpr:: ST (0..1)

Определение: строка символов, содержащая выражение на формальном языке, которое указывает, каким образом значения атрибутов экземпляра класса Act выводятся из входных параметров, связанных с этим экземпляром отношениями «произведен из».

Обсуждение: производный результат исследования, передаваемый в экземпляре класса Observation, может быть определен с помощью ассоциаций ActRelationship, у которых атрибут typeCode имеет значение DRIV (is derived from — произведен из) и которые связывают данный экземпляр с другими экземплярами класса Observation. Например, для определения производного исследования «среднее содержание гемоглобина в эритроците» (MCH) надо связать исследование MCH с исследованием концентрации гемоглобина (Hb) и абсолютным содержанием эритроцитов (RBC). В этом случае производное выражение должно задаваться формулой MCH = Hb / RBC.

Производное выражение представляется в виде строки символов.

Примечание — Синтаксис такого выражения должен быть полностью определен. Следует выбрать единственный стандартный язык выражений вместо необязательного выбора из нескольких языков. Синтаксис может быть основан на фактическом стандарте многих объектно-ориентированных языков, например, C++, Java, OCL и т. д. Конкретная спецификация такого языка выражений разрабатывается.

7.1.1.7 Act.text:: ED (0..1)

Определение: текстовое или мультимедийное описание действия.

Примеры: в экземпляре класса Act, у которого атрибут moodCode имеет значение DEF (определение), атрибут Act.text может содержать справочную информацию о действии. В экземпляре класса Act, у которого атрибут moodCode имеет значение RQO (направление), атрибут Act.text будет содержать специфические инструкции, применяемые только к этому направлению.

Ограничения: никаких ограничений ни на содержание описания, ни на его длину не накладывается.

Содержание этого описания не считается частью функциональной информации, передаваемой между компьютерными системами.

Описания в форме свободного текста используются, чтобы человек мог интерпретировать содержание и контекст действия, но вся информация, необходимая для автоматизированных функций, должна быть передана, используя надлежащие атрибуты и ассоциированные объекты.

Если предполагается, что информация, переданная в экземпляре класса Act, предназначена, в том числе, для людей (читателей и исполнителей), то автоматизированная система должна показать пользователю поле Act.text либо каким-то образом уведомить его о том, что такое поле существует, и показать его по требованию.

7.1.1.8 Act.statusCode:: CS (0..1)

Словарный домен: ActStatus (CNE)

Определение: Код, описывающий состояние действия.

7.1.1.9 Act.effectiveTime:: GTS (0..1)

Определение: выражение, указывающее «эффективное время» — момент или оперативное время действия, основное время осуществления действия, планируемое время действия.

Примеры: для клинических исследований атрибут effectiveTime содержит время проведения исследования (эффективное время) пациента.

Для контрактов атрибут effectiveTime указывает срок действия контракта.

Для информированных согласий атрибут effectiveTime указывает срок действия согласия.

Для лекарственных назначений атрибут effectiveTime указывает время, в течение которого надо принимать лекарство, включая режим приема (например, трижды в день в течение 10 дней).

Для хирургической процедуры (операции) атрибут effectiveTime указывает время, значимое для пациента, например, время между разрезом и последним швом.

Для действий транспортировки атрибут effectiveTime указывает время в пути.

Для обращений пациентов атрибут effectiveTime указывает «административное время», т.е. даты начала и завершения обращения, определяемые в соответствии с административным регламентом, в отличие от фактического времени оказания медицинской помощи по данному обращению.

Обсуждение: понятие «эффективного времени» называется также «основным» временем (в стандарте Arden Syntax) или «биологически значимым временем» (в стандарте HL7 v2.x). Этот атрибут надо отличать от атрибута activityTime, описывающего время деятельности.

Для исследований время деятельности может быть много позже времени наблюдения. Например, при анализе газов, содержащихся в артериальной крови, результат всегда будет получен через несколько минут после взятия биоматериала, а за это время физиологическое состояние пациента может значительно измениться.

Для существенно физической деятельности (хирургические процедуры, транспортировка и т. д.) эффективным временем является целевое время действия, например, коль скоро целью транспортировки является доставка груза из пункта А в пункт Б, то эффективным временем является время в пути. Однако действие обычно также включает в себя дополнительную работу, которая необходима для достижения цели действия, но не является существенной для этой цели.

Например, время, необходимое для приезда на место погрузки А и последующего возвращения на свою автобазу из пункта разгрузки Б, включается в физическую деятельность, но не включается во время транспортировки полезного груза. Другой пример: рабочие часы, считающиеся эффективным временем, могут быть с 8 часов утра до 5 часов вечера независимо от того, тратит ли человек на дорогу 10 минут или 2 часа. Приход на работу необходим, но на рабочие часы не влияет.

7.1.1.10 Act.activityTime:: GTS (0..1)

Определение: Выражение, указывающее время деятельности, т. е. когда происходило действие, описанное в экземпляре класса Act, либо, в зависимости от значения атрибута Act.moodCode, когда оно должно было происходить по предварительному плану, по текущему плану, когда оно могло происходить, и т. д.

Обсуждение: см. также описание атрибута effectiveTime. Атрибут activityTime содержит полное время деятельности, в том числе время, затрачиваемое на ее подготовку и на необходимые действия после ее завершения, если таковые считаются релевантными для основной деятельности. Следовательно, промежуток времени, передаваемый в атрибуте activityTime, может быть более долгим по сравнению со значением атрибута effectiveTime того же самого экземпляра класса Act либо совсем отличаться от него. Например, при ретроспективных исследованиях время деятельности намного позже эффективного времени.

В большинстве случаев разработчики должны рассматривать эффективное время как наиболее релевантное время действия.

7.1.1.11 Act.availabilityTime:: TS (0..1)

Определение: момент времени, когда информация об экземпляре класса Act (в зависимости от значения атрибута Act.moodCode) впервые становится доступной системе, воспроизводящей действие, описанное в этом экземпляре.

Примеры: в экземпляре класса Act может передаваться информация о том, что 3 часа назад у пациента произошел инфаркт миокарда правого желудочка (см. описание атрибута Act.effectiveTime), но информация об этом необычном случае поступила только несколько минут назад (значение атрибута Act.availabilityTime). Соответственно, с момента Act.effectiveTime до момента Act.availabilityTime любые вмешательства, скорее всего, предпринимались, исходя из предположения о более распространенном инфаркте миокарда левого желудочка. Новое знание может объяснить, почему эти вмешательства (например, назначение нитрата) могли оказаться не подходящими.

Обсуждение: значение атрибута availabilityTime — субъективная вторичная часть информации, добавленной (или измененной) системой, воспроизводящей действие. Она не может быть приписана автору действия (и не должна включаться в информацию, которую автор засвидетельствует своей подписью). Система, воспроизводящая действие, нередко отличается от системы, зарегистрировавшей действие, т. е. получает информацию о действии из другой системы. Время получения этой информации должно быть присвоено атрибуту availabilityTime, поскольку, начиная с этого момента, пользователи системы-получателя должны получить возможность узнать об этом экземпляре класса Act.

При передаче атрибута availabilityTime другой системе время доступности экземпляра А класса Act передается в экземпляре В этого класса, который содержит экземпляр А или ссылку на него. Тогда автор экземпляра В становится автором значения атрибута availabilityTime. Например, если выписка из медицинской карты подготовлена для отчета о нежелательной побочной реакции, то автор отчета становится и автором значений availabilityTime, указанных в этом отчете.

7.1.1.12 Act.priorityCode:: SET<CE> (0..*)

Словарный домен: ActPriority (CWE)

Определение: код или последовательность кодов (описывающих, например, обычную или экстренную срочность действия), указывающих срочность, с которой действие случилось, может случиться, происходит, планируется или требуется.

Обсуждение: этот атрибут используется в направлениях для обозначения требуемой срочности действия, а в документации о совершенном действии он указывает фактическую срочность действия. В экземплярах класса Act, у которых атрибут moodCode имеет значение DEF, в этом атрибуте указаны разрешенные значения срочности.

7.1.1.13 Act.confidentialityCode:: SET<CE> (0..*)

Словарный домен: Confidentiality (CWE)

Определение: Код, контролирующий раскрытие информации о данном действии независимо от значения атрибута moodCode.

Обсуждение: важно отметить, что необходимая конфиденциальность медицинской карты не может быть достигнута только с помощью кодов конфиденциальности, предназначенных для маскировки отдельных частей этой карты от определенных типов пользователей. При обеспечении конфиденциальности на основе кодов конфиденциальности, присваиваемых отдельным частям карты, возникают две проблемы: одна связана с возможностью логического вывода, а другая — с риском отсутствия доступа к информации, которая может быть критичной в определенных случаях оказания медицинской помощи. Возможность вывода означает, что фильтрованная чувствительная информация может быть выведена из другой, не отфильтрованной информации. Простейшей формой вывода может служить пример, когда из факта наличия направления на анализ иммуноблота или на анализ количества Т4- и Т8-клеток с большой вероятностью можно сделать вывод, что пациент ВИЧ-инфицирован, даже если результат анализа не известен. Очень часто диагнозы могут быть выведены из лекарственных назначений, например, назначение зидовудина свидетельствует о лечении ВИЧ. Проблема сокрытия отдельных частей медицинской карты становится особенно трудной при наличии текущих лекарственных назначений, поскольку должно быть обеспечено продолжение приема лекарств.

Чтобы ослабить некоторые риски вывода, следует считать, что агрегированные данные имеют уровень конфиденциальности самого конфиденциального действия в агрегате.

7.1.1.14 Act.repeatNumber:: IVL<INT> (0..1)

Определение: диапазон целых чисел, задающий минимальное и максимальное число повторений действия.

Примеры: после удаления зуба хирург-стоматолог может дать следующий совет пациенту: «Меняйте тампон каждый час от одного до трех раз, пока кровотечение не остановится полностью». Этот совет преобразуется в значение атрибута repeatNumber с нижней границей 1 и верхней границей 3.

Обсуждение: этот атрибут принадлежит к совокупности атрибутов управления рабочим процессом.

Число повторений дополнительно ограничивается временем. Действие будет повторяться, по меньшей мере, минимальное число раз и, самое большее, максимальное число раз. Повторения также закончатся, если время превысит максимальное значение атрибута Act.effectiveTime, если это случится раньше.

7.1.1.15 Act.interruptibleInd:: BL (0..1)

Определение: индикатор, указывающий, может ли действие прерываться асинхронными событиями.

Обсуждение: этот атрибут принадлежит к совокупности атрибутов управления рабочим процессом. Активные действия могут быть прерваны разными способами. Различаются следующие события прерывания: (1) когда получено прямое требование прекращения действия; (2) когда истекло время, выделенное для выполнения этого действия (тайм-аут); (3) когда «условие» выполнения действия перестает быть истинным (см. описание атрибута ActRelationship.checkpointCode); (4) когда экземпляр класса Act является компонентом, у которого атрибут joinCode имеет значение K (kill — прекращение) и все другие компоненты этой же группы завершены (см. описание атрибута Act.joinCode), и (5) когда прерывается объемлющий экземпляр класса Act.

Если действие получило прерывание и само оно может быть прервано, но в настоящее время имеет активные компоненты, которые не могут быть прерваны, то действие будет прервано тогда, когда все его активные не прерываемые компоненты будут завершены.

7.1.1.16 Act.levelCode:: CE (0..1)

Словарный домен: ActContextLevel (CWE)

Определение: Код, определяющий уровень в иерархической структуре составного действия и тип контекста составных действий («контейнеров»), распространяемый на компоненты действия в пределах этих контейнеров. Значение атрибута levelCode обозначает положение в этой иерархии включения и применяемые ограничения.

Обсуждение: Читатели должны иметь в виду, что этот признак может быть объявлен «устаревающим» в следующем нормативном выпуске HL7 ЭИМ. Вместо него рассматривается альтернативное понятие уровня,

использующее иерархию значений атрибута classCode. Если это изменение будет принято, то согласно процедурам сопровождения модели HL7 ЭИМ атрибут levelCode будет объявлен «устаревающим» в следующем выпуске ЭИМ, а затем «устаревшим» в выпуске после этого. Прежде чем использовать этот атрибут, пользователям рекомендуется проверить самые последние внутренние определения ЭИМ.

Понятия уровня определены в целях удовлетворения специфичных требований к передаче медицинских карт. Хотя эти понятия и применимы к некоторым другим типам транзакций, они не образуют полностью закрытый список. Существуют варианты других наборов ортогональных уровней, которые должны удовлетворять деловым требованиям (например, сообщения о нескольких пациентах можно подразделить с помощью вышестоящего уровня предметных областей).

Примеры: экземпляры класса Act, находящиеся на «уровне выписки из медицинской карты» (значение атрибута Act.levelCode равно «EXTRACT») и «уровне папки» (значение атрибута Act.levelCode равно «FOLDER») должны содержать данные о единственном лице, в то время как на «уровне нескольких субъектов» эти экземпляры могут содержать данные более чем об одном лице. В то время как «выписка из медицинской карты» может быть сделана из нескольких источников, «папка» должна содержать данные из одного источника. Уровень «композиции» (значение атрибута Act.levelCode равно «COMPOSITION») обычно имеет единственного автора.

Ограничения: ограничения, применимые к специфическому уровню, могут включать разные требования к участиям (например, к пациенту, к организации-источнику, к автору или другому лицу, подписывающему данные), к ассоциациям или включениям других экземпляров класса Act, к документам или к использованию шаблонов. Ограничения, применимые к уровню, могут также определить допустимые уровни экземпляров, которые могут быть компонентами этого уровня. Несколько вложенных уровней с тем же самым значением атрибута levelCode могут быть допустимыми, запрещенными (или ограниченными). Экземпляры класса Act следующего подчиненного уровня обычно разрешены на каждом уровне, но некоторые уровни могут быть опущены в модели, и допускается пропустить несколько уровней.

7.1.1.17 Act.independentInd:: BL (0..1)

Определение: индикатор, указывающий, можно ли управлять данным экземпляром класса Act независимо от других экземпляров, или управление этим экземпляром возможно только из вышестоящего составного действия, для которого данный экземпляр является компонентом. По умолчанию атрибут independentInd должен иметь значение TRUE.

Примеры: определению действия иногда присваивается значение атрибута independentInd = FALSE, если деловые правила не разрешают назначать это действие, не назначая группу действий, которая его содержит.

Назначение может иметь компонент, который нельзя прервать независимо от других компонентов.

7.1.1.18 Act.uncertaintyCode:: CE (0..1)

Словарный домен: ActUncertainty (CNE)

Определение: код, указывающий, было ли в целом утверждение, передаваемое в экземпляре класса Act, объявлено как недостаточно точное.

Примеры: пациенту могли в прошлом сделать операцию по холецистэктомии (однако он в этом не уверен).

Ограничения: отсутствие точности, объявленное с помощью этого атрибута, относится к объединенному смыслу утверждения, передаваемого в экземпляре класса Act с помощью всех описательных атрибутов (например, Act.code, Act.effectiveTime, Observation.value, SubstanceAdministration.doseQuantity и т. д.), и к смыслу всех компонентов.

Обсуждение: этот атрибут не предназначен для замены или конкуренции с отсутствием точности значения атрибута Observation.value или других отдельных атрибутов класса. Такие точечные указания отсутствия точности должны быть определены с помощью расширения типов данных PPD, UVP или UVN, применяемых к конкретному атрибуту. В частности, если отсутствие точности относится к значению количественного измерения, то его надо указать с помощью присваивания этому значению типа данных PPD<PQ>, а не с помощью атрибута uncertaintyCode. Если, к примеру, дифференциальные диагнозы перенумерованы или им присвоены вероятностные веса, то надо использовать типы данных UVP<CD> или UVN<CD>, а не атрибут uncertaintyCode. Использование атрибута uncertaintyCode возможно только в том случае, если точность всего действия и зависящих от него действий подвергаются сомнению.

Можно было бы подумать, что если точность совсем уж мала, то вместо атрибута uncertaintyCode надо воспользоваться атрибутом отрицания negationInd, однако эти два понятия совершенно независимы. Можно быть очень неуверенным в том, что событие имело место, но это не означает уверенности в его отрицании.

7.1.1.19 Act.reasonCode:: SET<CE> (0..*)

Словарный домен: ActReason (CWE)

Определение: код, указывающий мотивацию, причину, или логическое обоснование действия, если такое обоснование не было представлено с помощью ассоциации ActRelationship, у которой атрибут typeCode имеет значение RSON (has reason — имеет причину) и которая связывает данное действие с другим.

Примеры: примерами причин, которые могли бы заслуживать передачи в этом поле, служат «обычное назначение», «требование сообщить об инфекционном заболевании», «по запросу пациента», «требование закона».

Обсуждение: большинство причин действий могут быть четко описаны с помощью связывания нового действие с предшествующим, использующего ассоциацию ActRelationship, у которой атрибут typeCode имеет значение RSON. Такая связь означает, что предшествующее действие служит причиной для нового (см. описание класса ActRelationship). Это предшествующее действие может быть специфичным существующим действием или текстовым разъяснением. Такой подход пригоден для большинства случаев, и чем более специфична причина, тем более надо использовать ассоциацию ActRelationship, а не атрибут reasonCode.

Атрибут reasonCode остается как место для указания общих причин, которые не связаны с предшествующим действием или любыми другими условиями, выраженными с помощью экземпляров класса Act. Примером могут служить указания, что таковы требования закона или что причиной послужил запрос пациента и т. д. Но если требуется более точно сослаться на конкретную статью закона, правила, контракта или запроса пациента, то надо их представить в форме экземпляра класса Act (обычно так и делается) и не использовать атрибут reasonCode.

7.1.1.20 Act.languageCode:: CE (0..1)

Словарный домен: HumanLanguage (CWE)

Определение: основной язык, на котором описано данное действие, в особенности язык значения атрибута Act.text.

Переходы состояний экземпляра класса Act

Диаграмма перехода состояний класса Act показана на рисунке 5. Действие может иметь следующие состояния:

- aborted (прервано) — подсостояние состояния normal: активный объект услуги был неожиданно завершен;
- active (активно) — подсостояние состояния normal: объект услуги активен;
- cancelled (отменено) — подсостояние состояния normal: объект услуги был отменен до того, как стал активным;
- completed (завершено) — подсостояние состояния normal: объект услуги завершен;
- held (отложено) — подсостояние состояния normal: объект услуги, все еще находящийся на подготовительной стадии. Он не может стать активным, пока не будет выведен из этого состояния;
- new (новое) — подсостояние состояния normal: объект услуги, который готовится стать активным;
- normal (нормальное): охватывает все ожидаемые состояния объекта услуги, за исключением nullified и obsolete, которые представляют необычные терминальные состояния жизненного цикла;
- nullified (аннулировано): объект услуги не должен был создаваться, поэтому он аннулирован;
- obsolete (устарело): объект услуги заменен новым объектом;
- suspended (приостановлено) — подсостояние состояния normal: объект активной услуги временно приостановлен.

Между состояниями действия возможны следующие переходы:

- abort (прервать) — из состояния active в состояние aborted;
- revise (пересмотреть) — из состояния active в состояние active;
- complete (завершить) — из состояния active в состояние completed;
- suspend (приостановить) — из состояния active в состояние suspended;
- reactivate (активировать заново) — из состояния completed в состояние active;
- revise (пересмотреть) — из состояния completed в состояние completed;
- cancel (отменить) — из состояния held в состояние cancelled;
- revise (пересмотреть) — из состояния held в состояние held;
- release (освободить) — из состояния held в состояние new;
- activate (активировать) — из состояния new в состояние active;
- cancel (отменить) — из состояния new в состояние cancelled;

- complete (завершить) — из состояния new в состояние completed;
- hold (задержать) — из состояния new в состояние held;
- revise (пересмотреть) — из состояния new в состояние new;
- nullify (аннулировать) — из состояния normal в состояние nullified;
- obsolete (сделать устаревшим) — из состояния normal в состояние obsolete;
- activate (активировать) — из начального (пустого) состояния в состояние active;
- complete (завершить) — из начального (пустого) состояния в состояние completed;
- create (создать) — из начального (пустого) состояния в состояние new;
- jump (перейти) — из начального (пустого) состояния в состояние normal;
- abort (прервать) — из состояния suspended в состояние aborted;
- resume (возобновить) — из состояния suspended в состояние active;
- complete (завершить) — из состояния suspended в состояние completed;
- revise (пересмотреть) — из состояния suspended в состояние suspended.

7.1.2 Класс: ActRelationship (в предметной области Acts)

Атрибуты класса ActRelationship:

- typeCode:: CS
- inversionInd:: BL
- contextControlCode:: CS
- contextConductionInd:: BL
- sequenceNumber:: INT
- priorityNumber:: REAL
- pauseQuantity:: PQ
- checkpointCode:: CS
- splitCode:: CS
- joinCode:: CS
- negationInd:: BL
- conjunctionCode:: CS
- localVariableName:: ST
- seperatableInd:: BL

Ассоциации класса ActRelationship:

- target :: (1..1) Act :: inboundRelationship :: (0..*) (ассоциация с классом Act, роль inboundRelationship — входящая связь)
- source :: (1..1) Act :: outboundRelationship :: (0..*) (ассоциация с классом Act, роль outboundRelationship — исходящая связь)

Определение класса ActRelationship: класс ActRelationship моделирует направленную ассоциацию между классом-источником Act и классом-целью Act. Ассоциация класса ActRelationship с классом-источником Act моделирует «исходящую» часть направленной ассоциации, а ассоциация класса ActRelationship с классом-целью Act моделирует «входящую» часть направленной ассоциации. Смысл и назначение экземпляра класса ActRelationship указаны в атрибуте ActRelationship.typeCode.

Примеры:

1) Компонентами панели электролитов являются натрий, калий, pH и бикарбонат. Тогда экземпляр класса Act, описывающий панель электролитов, имела бы 4 «исходящие» ассоциации с экземплярами классов ActRelationship, у которых атрибут ActRelationship.typeCode имеет значение COMP (has component — имеет компонент).

2) Панель электролитов исследуется по направлению на лабораторный анализ. Тогда экземпляр класса Act, содержащий результат исследования, имел бы «исходящую» ассоциацию с экземпляром класса ActRelationship, у которого атрибут ActRelationship.typeCode имеет значение FLFS (fulfills — выполняет) и который имеет «входящую» ассоциацию с экземпляром класса Act, содержащим направление на анализ.

3) Операция «холецисэктомия» выполнена в связи с результатом исследования «желчекаменная болезнь». Тогда экземпляр класса Act, содержащий сведения об этой операции, имел бы «исходящую» ассоциацию с экземпляром класса ActRelationship, у которого атрибут ActRelationship.typeCode имеет значение RSON (has reason — имеет причину) и который имеет «входящую» ассоциацию с экземпляром класса Observation, в котором указан диагноз «желчекаменная болезнь».

Обсуждение: каждый экземпляр класса ActRelationship можно рассматривать как стрелу с наконечником (упирающимся в целевой объект) и хвостовиком (упирающимся в объект-источник). Функции (иногда называемые «ролями»), которые выполняют экземпляры класса Act при таком связывании, определяются

для каждого типа класса экземпляр класса ActRelationship по-разному. Когда экземпляр класса ActRelationship обеспечивать связь «композиция», то источник является составным объектом, а цель — его компонентом. Когда экземпляр класса ActRelationship обеспечивать причинно-следственную связь, то источником может быть любой экземпляр класса Act, а целью — экземпляр класса Act, описывающий причину появления источника.

Связи, ассоциированные с экземпляром класса Act, должны рассматриваться как свойства экземпляра-источника. Это означает, что автор экземпляра класса Act считается автором всех связей с этим экземпляром, имеющих его в качестве источника. Из этого правила нет исключений.

Более подробные сведения о различных типах экземпляров класса ActRelationship см. в описании атрибута ActRelationship.typeCode.

Класс ActRelationship используется для представления планов действий и клинических выводов или суждений о связи действий. Предшествующие действия могут быть связаны с вновь появившимися в качестве их причин. Известные факты могут быть связаны с клиническими гипотезами в качестве их доказательства. Списки диагнозов и другие сети связанных суждений о клинических событиях могут быть образованы с помощью экземпляров класса ActRelationship.

Одним из наиболее распространенных применений класса ActRelationship является описание композиций и декомпозиций действий, в которых экземпляры этого класса используются с атрибутом typeCode, имеющим значение COMP (has component — имеет компонент). Этот тип связи позволяет описывать действия с разной степенью детализации.

Связь композиции COMP позволяет группировать действия в «панели», например, панели LYTES, CHEM12 или CBC, с помощью которых можно сделать групповой заказ нескольких рутинных лабораторных анализов. Некоторые группировки, скажем, CHEM12, представляются более случайными, другие — более обоснованными, например, измерение артериального давления естественным образом состоит из систолического и диастолического давления.

Отношения композиции могут быть организованы в последовательности, чтобы формировать временные и условные (не временные) ряды планов действий (например, план лечения, критичные пути, протоколы клинических испытаний, протоколы испытаний лекарственных средств). Как в классе Act, так и в классе ActRelationship есть группа атрибутов, называемая «комплексом атрибутов рабочих процессов», с помощью которых можно формировать детальные спецификации планов выполняемых действий. Они имеют следующие функции.

С помощью атрибута ActRelationship.sequenceNumber можно упорядочить компоненты экземпляра класса Act в форме последовательной или параллельной коллекции, выражая логические ветвления, а также параллельные задачи (задачи, выполняемые в одно и то же время). С помощью атрибута ActRelationship.splitCode и ActRelationship.joinCode можно описывать выбор ветвей или параллельное выполнение задач;

С помощью атрибутов Act.activityTime и ActRelationship.pauseQty можно явным образом задавать время выполнения планируемых действий;

С помощью экземпляров класса ActRelationship, у которых атрибут ActRelationship.typeCode имеет значение PRCN (has precondition — имеет предусловие), можно задавать условия выполнения шагов плана в зависимости от состояния или результата предшествующих действий. С помощью атрибута ActRelationhsip.checkpointCode можно указать, когда проверяется предусловие действия в процессе передачи управления.

С помощью композиций экземпляров класса ActRelationship можно организовывать многие уровни вложения, позволяющие полностью представить управление рабочими процессами. Такое вложение и такое использование атрибутов рабочих процессов сконструировано по аналогии с конструкциями языков структурного программирования, поддерживающих параллелизм (ветвление, соединение, прерывания) и не требующих применения операторов перехода GOTO. Важно отметить, что ВСЕ планы описываются с помощью упорядоченных компонентов (шагов) составного действия (блока), которые могут быть отображены с помощью диаграмм Насси-Шнейдерман (известных также как «щелевые схемы» (Chap Charts) или структограммы), а не в форме цепочек связанных действий, как в блок-схемах.

С помощью отношений композиции детали действия могут быть представлены на разных уровнях для разных целей, не требуя переупорядочения структуры иерархии классов Act. Это позволяет отображать разные точки зрения на один и тот же процесс деятельности. Например, с позиции платежной системы панель лабораторных анализов может считаться простой оплачиваемой услугой. С клинической позиции та же самая панель лабораторных анализов является совокупностью отдельных анализов независимо от того,

как они были заказаны. Точка зрения лаборатории на эту панель может быть более детальной и включать в себя такие шаги, которые никогда не сообщаются клиницисту (центрифугирование, отбор, аликвотирование, использование определенных устройств и т. д.). Лабораторная точка зрения обеспечивает исчерпывающую спецификацию планов действий (которые могут быть автоматизированы). При составлении такой спецификации будут описаны все более и более вложенные подчиненные действия. Тем не менее действие остается тем же самым, только его детали скрываются с помощью декомпозиции до необходимого уровня.

Имея в виду природу варьирования деталей, действия можно назвать «фракталами», и они даже допускают больший уровень декомпозиции, подобно тому как движение руки робота можно представить в виде большого числа точных шагов управления.

В следующих подпунктах описаны атрибуты класса ActRelationship.

7.1.2.1 ActRelationship.typeCode:: CS (1..1) Mandatory

Словарный домен: ActRelationshipType (CNE)

Описание: код, указывающий смысл и назначение каждого экземпляра класса ActRelationship. Каждое из его значений предполагает определенное ограничение того, какие виды экземпляров класса Act могут быть связаны и каким именно способом.

Обсуждение: типы экземпляров класса ActRelationship попадают в одну из 5 категорий:

1) Композиция или декомпозиция, с составным объектом (источником) и компонентом (цель).

2) Продолжение, например, долечивание, выполнение, конкретизация, замена, преобразование и т. д., имеющие то общее, что источником и целью связи являются экземпляры класса Act принципиально того же типа, но отличающиеся значениями атрибута moodCode и других атрибутов, а также то, что цель связи существует до того, как появился источник, и источник содержит ссылку на цель, а цель содержит обратную ссылку на этот источник.

3) Предусловие, триггер, причина, противопоказание, при которых условно выполняемое действие является источником связи, а условие или причина — целью.

4) Постусловие, результат, цель или риск, при которых действие-источник имеет связь с результатом или целью.

5) Множество функциональных связей, например, поддержка, причина, производное, обобщаемых понятием «отношение».

7.1.2.2 ActRelationship.inversionInd:: BL (0..1)

Определение: индикатор, указывающий, что значение атрибута ActRelationship.typeCode должен быть интерпретировано таким образом, как если бы роли источника и целевых действий поменялись местами. Этот индикатор реверса используется, когда смысл значения атрибута ActRelationship.typeCode должен быть обращен.

7.1.2.3 ActRelationship.contextControlCode :: CS (0..1)

Словарный домен: ContextControl (CNE)

Определение: код, указывающий, какой вклад вносит данный экземпляр класса ActRelationship в контекст текущего экземпляра класса Act, и будет ли этот контекст распространяться на действия-потомки, чьи связи разрешают такое распространение (см. описание атрибута ActRelationship.contextConductionInd).

Обоснование: в целях сокращения дублирования люди склонны при интерпретации информации полагаться на ее контекст. Например, читая документ, извлеченный из папки, содержащей медицинскую карту пациента, пользователь сделает вывод, что этот документ относится к данному пациенту, даже если в нем нет прямого указания на принадлежность к этому пациенту. Однако другие части информации, например, автор папки (больница, в которой ведется эта медицинская карта), могут иногда применяться к содержанию папки (например, документ, составленный врачом этой больницы), а могут не применяться (например, в папку вложена копия документа, полученного из другого медицинского учреждения). Люди вполне хорошо делают необходимые выводы о том, какая часть контекста должна распространяться на элемент данных, а какая — нет. Однако случаются и неправильные выводы (например, в медицинскую карту пациента вложен документ с информацией о его родственнике). Более того, автоматизация подобных выводов представляет собой очень сложную задачу, даже если такие выводы необходимы для систем обеспечения принятия решений.

Обсуждение: с помощью этого атрибута можно точно указать, вносит ли ассоциация дополнение к контексту, связанному с конкретным элементом (например, добавляя дополнительного автора), или заменяет (отменяет) часть контекста, связанного с конкретным элементом (например, указывая единственного автора независимо от прежнего содержания элемента). С его помощью можно также указать, применяется ли ассоциация только к этому действию (не распространяемый контекст), или может применяться также и к производным действиям (распространяемый контекст).

Рассматриваемый атрибут тесно связан с атрибутом ActRelationship.contextConductionInd, указывающий, будут ли ассоциации, маркированные как распространяемые контекст, действительно распространять его на дочернее действие. Например, участие автора могло быть маркировано как распространяемое, но при этом не будет применяться к гиперссылке на внешний документ.

Если этот атрибут не имеет значения (например, у него пустое значение) или у него нет значения, то о контексте нельзя сделать никаких выводов. В системах могут использоваться собственные допущения на основе передаваемых данных. (По этой причине филиалам комитета HL7 рекомендуется в своих спецификациях указывать для этого атрибута значение по умолчанию или фиксированное значение, чтобы гарантировать его согласованную интерпретацию.)

Примеры: пусть экземпляр класса Observation имеет исходящую ассоциацию к экземпляру класса ActRelationship, содержащему информацию об участии пациента в исследовании, и при этом атрибут ActRelationship.contextControlCode имеет значение AP (additive, propagating — аддитивная, распространяемая). Пусть этот же экземпляр класса Observation имеет связи с компонентами исследования, описанные с помощью экземпляров класса ActRelationship, которые маркированы как распространяющие контекст. Это означает, что участие пациента, описанное для родительского экземпляра класса Observation, распространяется и на эти компоненты исследования.

Пусть создано составное назначение, содержащее заказ в аптеку, а также заказ нескольких лабораторных анализов. Это составное назначение имеет связи участия с пациентом и автором назначения, а также связь с диагнозом, и каждая из этих связей маркирована как «аддитивная, распространяемая». Пусть связь между составным назначением и заказом в аптеку маркирована как распространяющая связь (ее атрибут contextConductionInd имеет значение TRUE). При этом заказ в аптеку имеет связь участия с автором, маркированную как AN (additive, non-propagating — аддитивная, не распространяемая), и причинно-следственную связь с диагнозом, маркированную как OP (overriding, propagating — замещающая, распространяемая). Кроме того, заказ в аптеку имеет связь с информацией об отпуске лекарства, а также связь с формуляром лекарства, которая маркирована как не распространяемая (ее атрибут contextConductionInd имеет значение FALSE). Такая совокупность объектов и связей трактуется следующим образом.

Заказ в аптеку интерпретируется как наследующий пациента от составного назначения. Он имеет двух авторов (одного, унаследованного от составного назначения, и другого, явно указанного автором заказа в аптеку). Диагнозом, в связи с которым сделан заказ в аптеку, будет считаться только тот, что связан с заказом в аптеку, а не с составным назначением. Событие отпуска унаследует пациента от составного назначения и диагноз заказа в аптеку, но не авторов. Формуляр не будет связан ни с пациентом, ни с диагнозом, ни с автором.

7.1.2.4 ActRelationship.contextConductionInd:: BL (0..1)

Определение: если этот атрибут имеет значение TRUE, то связи родительского действия распространяются через данный экземпляр класса ActRelationship на дочернее действие.

Обсуждение: распространяются только те связи, которые были добавлены к контексту родительского действия и были маркированы как «распространяемые» (см. описания атрибута contextControlCode в классах ActRelationship и Participation).

Идентификация экземпляра класса Act как родительского или дочернего (и, следовательно, идентификация направления распространения контекста) определяется пересечением связи при ее сериализации. Первое обнаруженное действие рассматривается как родительское. Контекст распространяется через экземпляр класса ActRelationship на второе (дочернее) действие.

Обоснование и примеры указаны в описании атрибута ActRelationship.contextControlCode.

7.1.2.5 ActRelationship.sequenceNumber:: INT (0..1)

Определение: целое значение, указывающее относительное положение данной связи среди других связей похожих типов, у которых источником является один и тот же экземпляр класса Act.

Обсуждение: этот атрибут принадлежит к группе атрибутов управления рабочим процессом. План действия представляет собой составное действие, связанное с действиями-компонентами. В упорядоченном плане каждый экземпляр класса ActRelationship, связывающий составное действие с компонентом, имеет атрибут sequenceNumber, значение которого определяет порядок шагов плана. Если у нескольких компонентов значение атрибута sequenceNumber одинаково, то эти компоненты являются ветвями. Ветви могут быть исключающими (как в переключателе case) или могут допускать параллельное выполнение, что можно указать с помощью атрибута splitCode.

7.1.2.6 ActRelationship.priorityNumber:: REAL (0..1)

Определение: целое значение, указывающее относительный приоритет данной связи среди других связей похожих типов, у которых источником является один и тот же экземпляр класса Act. Связи с меньшими значениями атрибута priorityNumber рассматриваются раньше и выше тех, что имеют более высокие значения.

Примеры: если указано несколько критериев, то с помощью этого атрибута можно указать, какой из них должен быть рассмотрен раньше других. Если связи с компонентами имеют одно и то же значение атрибута sequenceNumber, то атрибут приоритета позволяет указать, какая из них должна быть рассмотрена раньше других. Если альтернативы или варианты выбираются людьми, то значение priorityNumber указывает предпочтение.

Обсуждение: упорядочение может быть полным, при котором все значения приоритета уникальны, или частичным, при котором один и тот же приоритет назначается нескольким связям.

7.1.2.7 ActRelationship.pauseQuantity:: PQ (0..1)

Определение: промежуток времени между готовностью действия к выполнению и фактическим началом его выполнения.

Обсуждение: этот атрибут принадлежит к группе атрибутов управления рабочим процессом. План действия представляет собой составное действие, связанное с действиями-компонентами. В упорядоченном плане каждый экземпляр класса ActRelationship, связывающий составное действие с компонентом, имеет атрибут sequenceNumber, значение которого определяет порядок шагов плана. Если у шага есть предусловия, то его выполнение инициируется в том случае, когда они удовлетворяются. В этот момент запускает таймер со значением pauseQuantity, и действие начинает выполняться, когда пройдет время, указанное в атрибуте pauseQuantity.

7.1.2.8 ActRelationship.checkpointCode:: CS (0..1)

Словарный домен: ActRelationshipCheckpoint (CNE)

Определение: код, указывающий моменты проверки выполнения предусловия действия, (например, перед тем, как действие начнется впервые, после каждого повторения действия, но не перед первым, или в процессе всего времени действия).

Обсуждение: этот атрибут принадлежит к группе атрибутов управления рабочим процессом. План действия представляет собой составное действие, связанное с действиями-компонентами. В упорядоченном плане каждый экземпляр класса ActRelationship, связывающий составное действие с компонентом, имеет атрибут sequenceNumber, значение которого определяет порядок шагов плана. Если у шага есть предусловия, то его выполнение инициируется в том случае, когда они удовлетворяются. С помощью атрибута repeatNumber можно указать, что выполнение действия может повторяться. А с помощью атрибута checkpointCode можно указать, когда проверяется предусловие, что аналогично различным условным операторам и циклам в языках программирования: while-do по сравнению с do-while или repeat-until по сравнению с loop-exit.

Для всех значений атрибута checkpointCode, кроме E (end — конец), предусловия проверяются в момент завершения предшествующего шага плана при условии, что данный шаг является следующим согласно значению атрибута sequenceNumber.

Если атрибут checkpointCode критерия повторяющегося действия имеет значение E (end — конец), то этот критерий проверяется только в конце каждого повторения действия. Если критерий повторения удовлетворен, то следующее повторение действия готово к выполнению.

Если атрибут checkpointCode имеет значение S (entry — вход), то критерий проверяется в начале каждого повторения (если таковые имеются), при этом «начало» означает, что критерий проверяется однократно при старте «циклического» повторения.

Если атрибут checkpointCode имеет значение T (through — в течение), то оно задает особый случай, когда критерий проверяется в процессе повторения. Как только критерий перестал выполняться, должно быть инициировано событие прерывания действия (см. описание атрибута Act.interruptibleInd) и в принципе действие должно быть завершено.

Атрибут checkpointCode со значением X (exit — выход) используется в специальном шаге плана, представляющем выход из цикла. С его помощью можно обеспечить завершение плана действий в связи с выполнением определенного условия, проверяемого при выполнении этого плана. Такие критерии выхода упорядочены относительно других компонентов плана с помощью атрибута ActRelationship.sequenceNumber.

7.1.2.9 ActRelationship.splitCode:: CS (0..1)

Словарный домен: ActRelationshipSplit (CNE)

Определение: код, указывающий, какие ветви плана действия выбираются среди других ветвей.

Обсуждение: этот атрибут принадлежит к группе атрибутов управления рабочим процессом. План действия представляет собой составное действие, связанное с действиями-компонентами. В упорядоченном плане каждый экземпляр класса ActRelationship, связывающий составное действие с компонентом, имеет атрибут sequenceNumber, значение которого определяет порядок шагов плана. Если для нескольких компонентов значение атрибута sequenceNumber одинаково, то эти компоненты являются ветвями. Атрибут splitCode указывает, является ли ветвь исключающей (как в переключателе case) или включающей, т. е. может допускать параллельное выполнение других ветвей.

В дополнение к исключающему и включающему ветвлению с помощью атрибута splitCode можно указать, как проверяется предусловие (называемое также сторожевым условием) ветви. Сторожевое условие может проверяться однократно при переходе к ветви, и если оно не выполнено, то ветвь отвергается. В другом варианте выполнение ветви может быть отложено, пока это условие не будет выполнено.

При исключающем ветвлении с ожиданием первая ветвь, у которой ее условие ветвления станет выполненным, начнет выполняться, а все остальные ветви будут отвергнуты. При включающем ветвлении с ожиданием некоторые ветви могут уже выполняться, в то время как другие все еще будут ожидать, пока их сторожевое условие не будет выполнено.

7.1.2.10 ActRelationship.joinCode:: CS (0..1)

Словарный домен: ActRelationshipJoin (CNE)

Определение: код, указывающий способ восстановления синхронизации параллельно выполняемых действий.

Обсуждение: этот атрибут принадлежит к группе атрибутов управления рабочим процессом. План действия представляет собой составное действие, связанное с действиями-компонентами. В упорядоченном плане каждый экземпляр класса ActRelationship, связывающий составное действие с компонентом, имеет атрибут sequenceNumber, значение которого определяет порядок шагов плана. Если для нескольких компонентов значение атрибута sequenceNumber одинаково, то эти компоненты являются ветвями. Ветви могут выполняться параллельно, если атрибут splitCode указывает, что в одно и то же время может исполняться более одной ветви. В этом случае с помощью атрибута joinCode можно указать, будет ли восстанавливаться синхронизация ветвей, и если да, то каким образом.

Основные способы восстановления синхронизации следующие:

- поток управления ждет, пока выполнение каждой ветви не завершится (ожидание ветвей);
- как только выполнится одна ветвь, выполнение остальных ветвей прекращается (прекращение ветвей);
- синхронизация ветвей не восстанавливается, и они продолжают выполняться (отложенные ветви).

Прекращение ветвей происходит только в том случае, когда имеется, как минимум, одна активная ожидающая (или исключающая ожидающая) ветвь. Если активных ожидающих ветвей нет, то процесс прекращения ветвей не инициируется (а не прекращается после инициации).

Поскольку отложенная ветвь не связана с другими ветвями, наличие активных отложенных ветвей не мешает прекращению другой ветви.

7.1.2.11 ActRelationship.negationInd:: BL (0..1)

Определение: индикатор, указывающий, что значения связи инвертировано.

Примеры: Если не инвертированная связь указывает, что у действия А есть компонент В, то ее инверсия означает, что действие В не является компонентом действия А. Если действие В описывает причину действия А, то инверсия означает, что действие В не является причиной действия А. Если действие В описывает предусловие действия А, то инверсия означает, что действие В не является предварительным условием действия А.

Обоснование: как показывают примеры, использование этого атрибута довольно ограничено, особенно по сравнению с атрибутом Act.negationInd, с помощью которого фактически можно указать, что описанное действие не существует, не выполняется и т. д., в то время как с помощью атрибута ActRelationship.negationInd можно лишь отрицать данную связь между действием-источником и действием-целью действия, а не изменить смысл каждого действия. Поэтому данный атрибут используется, главным образом, в разъяснительных целях.

Учите также различие между отрицанием и противоположностью. Противопоказание является противоположностью показания (причины). То обстоятельство, что наличие боли в пояснице не является причиной назначения антибиотиков, не означает, что боль в пояснице является противопоказанием для применения антибиотиков.

7.1.2.12 ActRelationship.conjunctionCode:: CS (0..1)

Словарный домен: RelationshipConjunction (CNE)

Определение: код, указывающий логическое соединение критериев, описанных условными связями между действиями (например, AND («и»), OR («или»), XOR («исключающее или»)).

Ограничения: Все критерии, соединенные с помощью AND, должны выполняться. Если в одну группу входят критерии, соединяемые с помощью OR и AND, то должны выполняться хотя бы один из критериев OR и все критерии AND. Если в одну группу входят критерии, соединяемые с помощью XOR, OR и AND, то должен выполняться ровно один критерий XOR, по крайней мере один из критериев OR и все критерии AND. Другими словами, множества критериев AND, OR и XOR, в свою очередь, объединены логическим оператором AND (все критерии AND, по крайней мере один из критериев OR и ровно один критерий XOR). Если требуется иное, то можно воспользоваться вложенными критериями.

7.1.2.13 ActRelationship.localVariableName:: ST (0..1)

Определение: строка символов, указывающая имя входного параметра, по значению которого экземпляр класса Act, который служит источником ассоциации с данным экземпляром класса ActRelationship, может вычислить некоторые из своих атрибутов. Областью действия имени локальной переменной является значение атрибута Act.derivationExpr, который может содержать это имя в качестве входного параметра.

7.1.2.14 ActRelationship.seperatableInd:: BL (0..1)

Определение: этот индикатор указывает, должно ли действие-источник связи интерпретироваться независимо от целевого действия связи. Его значение не может препятствовать человеку или приложению обрабатывать действия независимо друг от друга, оно лишь обозначает желание и готовность автора аттестовать содержание действия-источника, если оно будет отделено от содержания целевого действия. Имейте в виду, что типичным значением этого атрибута по умолчанию будет TRUE. Также учтите, что этот атрибут ортогонален механизму наследования контекста и никак с ним не связан. Если контекст действия распространен на вложенные действия, то тем самым предполагается, что эти вложенные действия не могут интерпретироваться без распространенного контекста.

7.1.3 Класс Participation (в предметной области Acts)

Атрибуты класса Participation:

- typeCode:: CS
- functionCode:: CD
- contextControlCode:: CS
- sequenceNumber:: INT
- negationInd:: BL
- noteText:: ED
- time:: IVL<TS>
- modeCode:: CE
- awarenessCode:: CE
- signatureCode:: CE
- signatureText:: ED
- performInd:: BL
- substitutionConditionCode:: CE

Ассоциации класса Participation:

- act::(1..1)Act::participation::(0..*) (ассоциация с классом Act, роль participation — участие)
- role::(1..1)Role::participation::(0..*) (ассоциация с классом Role, роль participation — участие)

Класс Participation является обобщением следующих классов:

- ManagedParticipation (управляемое участие)

Определение класса Participation: класс Participation моделирует ассоциацию между классом действия Act и классом сущности Entity, принимающим участие в действии, описанном классом Act, и выполняющим в этом участии роль, описанную классом Role. Каждая пара сущность-роль связана с экземпляром класса Act с помощью одного экземпляра класса Participation. Тип вовлечения этой пары в действие, описываемое экземпляром класса Act, задается с помощью атрибута Participation.typeCode.

Примеры:

1) Исполнители действий (хирурги, исследователи, врачи общей практики);
2) Субъекты действий, пациент, устройства;
3) Местоположения;
4) Автор, поручитель, свидетель, информатор;
5) Адресат, получатель информации.

Обоснование: экземпляры класса Participation описывают выполняемую функцию, в то время как экземпляры класса Role описывают компетенцию. Экземпляры класса Participation описывают фактическое участие сущности в определенном действии и, следовательно, вид участия зависит от специфики этого действия. Напротив, роль описывает компетенцию сущности (например, какое принципиальное участие она может принимать в каких типах действий) независимо от конкретного действия.

Например, профессиональные полномочия лица (описанные экземпляром класса Role) могут значительно отличаться от его фактических действий (описанных экземпляром класса Participation). Типичным примером может служить выполнение интерном или ординатором анестезии или хирургической операции под присмотром (более или менее тщательным) лечащего специалиста.

Один и тот же вид участия в действии могут принимать несколько сущностей, что бывает при коллективном сотрудничестве или групповых действиях. Понятие кратных участий частично перекрывается с понятием подчиненных действий (компонентов действий). Когда в действие вовлечено несколько действующих лиц, каждое из них выполняет свою задачу (за крайне редким исключением таких симметричных действий, примером которых служат два человека, которые тянут одну веревку, каждый со своего конца). Поэтому участие нескольких действующих лиц вполне удовлетворительно может быть представлено как составное действие, образованное из нескольких действий, каждое из которых выполняется ровно одним лицом.

Например, в протокол хирургической операции могли быть внесены три лица:

а) медицинская сестра, которая должна получить согласие на проведение операции;

б) ведущий хирург;

в) анестезиолог.

Эти три лица выполняют разные задачи, которые можно представить в виде трех связанных действий:

а) получение согласия,

б) собственно операция,

в) анестезия, выполняемая параллельно с операцией.

Если используются три таких компонента, то тип действующего лица у медсестры, хирурга и анестезиолога будет одним и тем же, а именно, «исполнитель». Чем более детальные компоненты действий используются, тем меньше требуется градаций действующих лиц. И наоборот, чем меньше компонентов у действия, тем больше требуется градаций действующих лиц.

Как эмпирическое правило, использование подзадач вместо нескольких действующих лиц полезно в тех случаях, когда каждая подзадача требует специального планирования, или отдельной оплаты, или если полная ответственность за выполнение подзадач различна. Однако в большинстве случаев ресурсы персонала планируются в форме бригад (а не отдельных сотрудников), методы расчетов по оплате лечения имеют тенденцию группировки мелких подзадач в одну строку счета, а полная ответственность часто возлагается на одного лечащего врача, старшую медсестру или заведующего отделением. Такая модель позволяет сочетать подходы моделирования действий как выполняемых несколькими лицами и как состоящих из нескольких задач, что отражает реалии процесса оказания медицинской помощи. При этом наблюдается небольшой уклон в пользу объединения мелких подзадач в одно действие.

В следующих подпунктах описаны атрибуты класса Participation.

7.1.3.1 Participation.typeCode:: CS (1..1) Mandatory

Словарный домен: ParticipationType (CNE)

Определение: код, указывающий вид участия (вовлечения) сущности, выполняющей определенную роль, связанную с этим участием, в действии, описанном в ассоциированном экземпляре класса Act.

Ограничения: допустимые значения атрибута Participant.typeCode имеют строгие семантические определения, соответствующие области применения стандарта HL7. Это атрибут с кодируемыми значениями без исключений. Альтернативные системы кодирования этого атрибута не допускаются.

7.1.3.2 Participation.functionCode:: CD (0..1)

Словарный домен: ParticipationFunction (CWE)

Определение: необязательный код, указывающий дополнительную информацию о функции данного участия в действии, если она однозначно не вытекает из значения атрибута Participation.typeCode.

Примеры: ведущий хирург, второй хирург (или первый ассистент хирурга, стоящий напротив ведущего хирурга), второй ассистент (часто стоящий за ведущим хирургом), потенциально третий ассистент, старшая операционная сестра, операционная сестра, медрегистратор, ведущий анестезиолог, анестезиолог, сестра-анестезистка, санитар, сестра послеоперационного мониторинга, ассистенты, акушерки, студенты и т. д.

Ограничения: если этот код указан, его значение не должно конфликтовать со значением атрибута Participation.typeCode.

Не будет написано ни одной спецификации стандарта HL7, технически зависящей от атрибута functionCode. Если возникнет потребность в дополнительных понятиях, то они должны быть определены для значений атрибута Participation.typeCode.

Обсуждение: с помощью этого кода можно указать большое разнообразие и нюансы функций участников в действии. Число и виды этих функций зависят от конкретного типа действий. Например, для каждого вида операции и метода ее проведения может требоваться разное число ассистентов хирурга или медицинских сестер.

Поскольку функции участия характеризуют то, что именно люди выполняют в процессе действия, то в действительности они эквивалентны подзадачам, которые могут выполняться параллельно. Если нужны дополнительные сведения об этих подзадачах, кроме списка их участников, то надо использовать компоненты действия.

7.1.3.3 Participation.contextControlCode:: CS (0..1)

Словарный домен: ContextControl (CNE)

Определение: код, указывающий, какой вклад вносит данный экземпляр класса Participation в контекст текущего экземпляра класса Act, и будет ли этот контекст распространяться на действия-потомки, чьи связи разрешают такое распространение (см. описание атрибута ActRelationship.contextConductionInd).

Обсуждение: обоснование, обсуждение и примеры см. в описании атрибута ActRelationship.contextControlCode.

7.1.3.4 Participation.sequenceNumber:: INT (0..1)

Определение: целое значение, указывающее относительный порядок данного участия среди других участий в том же самом действии.

Обоснование: упорядочение экземпляров класса Participation, описывающих участие в данном действии, может быть предпринято по функциональным причинам или для присвоения участиям приоритета. Примером может служить указание порядка участия выгодоприобретателей, необходимое для координации возмещений по счету на оплату страхового случая.

7.1.3.5 Participation.negationInd:: BL (0..1)

Определение: значение TRUE этого атрибута указывает, что участие, описанное в экземпляре класса Participation, не имело, не имеет или не должно иметь место (в зависимости от значения атрибута moodCode).

Обоснование: этот атрибут имеет два основных приложения:

- чтобы указать, что конкретная роль не имела или не должна иметь участие в действии;
- чтобы удалить участника из контекста, распространяемого на другие действия.

Обсуждение: экземпляр класса Participation, у которого атрибут negationInd имеет значение TRUE, обладает преимуществом по отношению к экземпляру класса Participation, у которого атрибут negationInd имеет значение FALSE. Другими словами, если возник конфликт, то отрицание участия имеет больший приоритет.

Примеры: Др. Смит не участвовал; пациент Джонс не подписывал информированное согласие.

7.1.3.6 Participation.noteText:: ED (0..1)

Определение: текстовое или мультимедийное представление комментария к данному участию. Относится только к этому участнику.

7.1.3.7 Participation.time:: IVL<TS> (0..1)

Определение: интервал времени, в течение которого сущность, связанная с действием с помощью конкретного экземпляра класса Participation, была вовлечена в это действие.

Обоснование: указание времени участия необходимо, если сущность принимала участие в действии не на протяжении всего времени выполнения действия. Время участия также используется для указания времени, в течение которого выполнялись определенные очень общие подзадачи, не заслуживающие упоминания в полном действии.

Примеры:

1. Время ввода данных в систему-источник передается в атрибуте Participation.time экземпляра класса Participation, связывающем действие с ролью «ввод данных».
2. Концом времени участия автора считается время подписи данных, передаваемых в экземпляре класса Act.
3. Временем участия поручителя, передаваемым в атрибуте Participation.time, считается момент его подписи.

7.1.3.8 Participation.modeCode:: CE (0..1)

Словарный домен: ParticipationMode (CWE)

Определение: код, указывающий модальность участия сущности в действии, выполняя определенную роль.

Примеры: физическое присутствие, по телефону, письменная коммуникация.

Обоснование: этот атрибут особенно часть используется для участников «автор» («создатель»), чтобы определить, обеспечивалась ли информация, представленная действием, первоначально устно, письменно (рукописно) или в электронном виде.

7.1.3.9 Participation.awarenessCode:: CE (0..1)

Словарный домен: TargetAwareness (CWE)

Определение: код, указывающий степень осведомленности сущности о действии, в которой она участвует в определенной роли (обычно в качестве целевого субъекта).

Примеры: в записях медицинской карты может быть указана степень осведомленности пациента, члена его семьи или другого участника о наличии у пациента смертельного заболевания.

Определение: если степень осведомленности, отказ, отсутствие сознания и т. д. — предмет медицинского рассмотрения (например, часть списка проблем), то нужно передавать эту информацию в экземплярах класса Observation, а не полагаться исключительно на этот простой атрибут класса Participation.

7.1.3.10 Participation.signatureCode:: CE (0..1)

Словарный домен: ParticipationSignature (CNE)

Определение: код, указывающий, заверила ли сущность свое участие подписью, а также указывающий необходимость такой подписи.

Примеры: протокол хирургической операции (представленный в форме экземпляра класса Procedure) должен быть подписан ответственным оперировавшим хирургом и, возможно, другими участниками операции. (См. также описание атрибута Participation.signatureText.)

7.1.3.11 Participation.signatureText:: ED (0..1)

Определение: Текстовое или мультимедийное представление подписи, подтверждающей определенный вид участия сущности в действии (указанный в атрибуте Participation.typeCode) и ее согласие принять на себя связанную с этим ответственность.

Примеры:

1) Участник «автор» берет на себя ответственность за достоверность содержания экземпляра класса Act в меру его осведомленности.

2) Получатель информации подтверждает лишь факт ее получения.

Обсуждение: подпись может быть представлена многими разными способами либо по значению, либо по ссылке в соответствии с типом данных ED. Типичные случаи таковы:

1) Подписи бумажных документов: значение, имеющее тип данных ED, может содержать ссылку на некоторый документ или файл, который может быть найден с помощью электронного интерфейса к архиву бумажных документов.

2) Электронная подпись: с помощью типа данных ED можно представить фактически любую схему электронной подписи.

3) Стандартная электронная подпись: в частности, с помощью типа данных ED можно представить стандартные электронные подписи, например, с помощью ссылки на блок данных подписи, сконструированный в соответствии со стандартом электронной подписи, например, XML-DSIG, PKCS#7, PGP и т. д.

7.1.3.12 Participation.performInd:: BL (0..1)

Определение: указывает, должен ли быть зарезервирован ресурс этого участия до начала его участия (т. е. участие ресурса управляется расписанием).

Обоснование: этот признак используется для очень специфичных нужд в контексте планирования ресурсов. В большинстве описаний участия он не требуется. Чаще всего он применяется при описании участия, для которого требуется конкретное помещение или определенное оборудование, использование которого контролируется расписанием.

7.1.3.13 Participation.substitutionConditionCode:: CE (0..1)

Словарный домен: SubstitutionCondition (CWE)

Определение: указывает условия, при которых один участвующий предмет может быть заменен другим.

7.1.4 Класс Account (в предметной области Acts)

Код класса: ACCT.

Атрибуты класса Account:
- balanceAmt:: MO
- currencyCode:: CE
- interestRateQuantity:: RTO<MO,PQ>
- allowedBalanceQuantity:: IVL<MO>

Класс Account является специализацией класса Act.

Определение класса Account: специализация класса Act, представляющая категорию финансовых операций, которые проводятся на отдельном балансе.

Обсуждение: этот класс может использоваться для представления накопленной общей суммы, полученной за товары или услуги, оплаченной товары или услуги, а также для представления счетов дебита и кредита, между которыми осуществляются операции.

Примеры: лицевые счета пациентов, счета за случаи обращения; центры затрат; счета дебиторов.

В следующих подпунктах описаны атрибуты класса Account.

7.1.4.1 Account.name::ST (0..1)

Определение: описательное имя счета, взятое из книги, в которой ведется этот счет.

Обсуждение: это имя не является идентификатором счета (для идентификаторов счета используйте атрибут id, а для описательных или смысловых названий счета — данный атрибут.)

Примеры: «затраты июня 2002 года», «дебиторская задолженность Джона Смита».

7.1.4.2 Account.balanceAmt:: MO (0..1)

Определение: сумма операций по дебету и кредиту счета (остаток) с момента его открытия.

Обсуждение: остаток счета обычно сообщается в валюте, указанной в атрибуте Account.currencyCode. Однако сумму остатка можно сообщать в альтернативных валютах.

7.1.4.3 Account.currencyCode:: CE (0..1)

Словарный домен: Currency (CWE)

Определение: указывает валюту счета.

Обсуждение: конкретные суммы можно сообщать в другой валюте, однако данный атрибут должен представлять валюту, используемую по умолчанию для операций по этому счету.

7.1.4.4 Account.interestRateQuantity:: RTO<MO,PQ> (0..1)

Определение: дробь, указывающая процентную ставку остатка по этому счету, и период, для которого она действует.

Обсуждение: в этом атрибуте может быть указана процентная ставка для платежей (например, по ссудам, просроченным счетам) или для поступлений (по вложениям капитала и т. д.) в зависимости от типа счета.

Примеры: 0.10/1a (10% в год); 0.0005895/1d (0,05895% в день).

Ограничения: должна иметься возможность перевода единиц измерения знаменателя, имеющего тип данных PQ, в секунды. (Т. е. знаменателем должно быть время.)

7.1.4.5 Account.allowedBalanceQuantity:: IVL<MO> (0..1)

Определение: интервал, описывающий минимально и максимально допустимое значение остатка счета.

Обсуждение: этот интервал не обязательно задает «жесткие» пределы (т.е. остаток может быть и выше, и ниже указанных пределов). Он указывает такой «целевой» диапазон остатка счета, выход из которого может повлечь за собой определенные последствия. Для остатка счета не обязательно одновременно задавать и верхний, и нижний предел (или какой-либо из них).

Примеры: пределы для «прекращения потерь»; пределы для кредита.

7.1.5 Класс DeviceTask (в предметной области Acts)

Код класса: CONTREG.

Атрибуты класса DeviceTask:
- parameterValue:: LIST<ANY>

Класс DeviceTask является специализацией класса Act.

Определение класса DeviceTask: специализация класса Act, представляющая деятельность автоматизированной системы.

Обсуждение: этот класс может использоваться для представления действий, вызванных внешней командой или запланированных и самостоятельно инициированных устройством (например, периодическая калибровка или очистка). Если экземпляр класса DeviceTask содержит описание команды, то его атрибут moodCode должен иметь значение ORD; если он содержит описание задачи (в том числе находящейся

в стадии выполнения), то атрибут moodCode должен иметь значение EVN; для задачи, автоматически выполняемой по расписанию, атрибут moodCode должен иметь значение APT.

В следующих подпунктах описаны атрибуты класса DeviceTask.

7.1.5.1 DeviceTask.parameterValue:: LIST<ANY> (0..*)

Определение: параметры задачи, отправленные на устройство при передаче ему команды (либо при настройке расписания самостоятельно инициируемых задач).

Обоснование: некоторые параметры задач однозначно определяются конкретной моделью устройства. Наиболее критичные параметры задачи (например, контейнер, необходимый для работы устройства, позиционирование, временные параметры и т. д.) определены в стандартизованной структуре, и их не надо включать в список параметров. Этот список используется только для тех параметров, которые не могут быть стандартизованы, поскольку являются уникальными для конкретной модели оборудования.

Примечание — Семантика и интерпретация данных, передаваемых в атрибуте parameterValue, зависят исключительно от содержания спецификации или документации конкретного устройства. Эта информация не передается как часть сообщения.

Ограничения: параметры задаются таким способом только в том случае, если они не включены в отдельную структуру, определенную в стандарте HL7. Параметры представляют собой список значений любых типов данных, интерпретируемых устройством. Параметрам должны быть присвоены типы данных из числа определенных в стандарте HL7 (например, коды для номинальных значений, к примеру, для флажков, типы данных REAL и INT для чисел, тип данных TS для моментов времени, тип данных PQ для физических количеств и т.д.). Однако, помимо типов данных, использование параметров не входит в область применения стандарта HL7.

7.1.6 Класс DiagnosticImage (в предметной области Acts)

Код класса: DGIMG.

Атрибуты класса DiagnosticImage:

- subjectOrientationCode:: CE

Класс DiagnosticImage является специализацией класса Observation.

Определение класса DiagnosticImage: разновидность исследования, имеющая в качестве непосредственного и основного результата (постусловие) новые данные о субъекте в форме визуальных изображений.

В следующих подпунктах описаны атрибуты класса DiagnosticImage.

7.1.6.1 DiagnosticImage.subjectOrientationCode:: CE (0..1)

Словарный домен: ImagingSubjectOrientation (CWE)

Определение: код, содержащий качественную характеристику пространственного положения изображаемого объекта по отношению к пленке или детектору.

7.1.7 Класс Diet (в предметной области Acts)

Код класса: DIET.

Атрибуты класса Diet:

- energyQuantity:: PQ
- carbohydrateQuantity:: PQ

Класс Diet является специализацией класса Supply.

Определение класса Diet: действие кормления субъекта или предоставления ему питания.

Обсуждение: детали диеты передаются в экземпляре класса Material, ассоциированного с помощью экземпляра класса Participation, у которого атрибут typeCode имеет значение PRD (product — товар). Номер диеты, имеющий медицинское значение, можно передать в атрибуте Diet.code, однако детали пищи, входящей в диету, и различные сочетания блюд должны передаваться в форме экземпляров класса Material.

Примеры: безглютеновая диета, бессолевая диета.

В следующих подпунктах описаны атрибуты класса Diet.

7.1.7.1 Diet.energyQuantity:: PQ (0..1)

Определение: предоставляемая биологическая энергия (калории) в день.

Обсуждение: должно существовать преобразование единицы измерения этого физического количества в ккал/день (или кДж/день). Обратите внимание на возможную путаницу между «большой калорией» и «малой калорией». Этикетки на продуктах питания указывают калорийность в «больших калориях». Однако в качестве единицы измерения лучше использовать малую калорию, являющуюся 1/1000 большой калории. Это различие строго проводится в таблицах единиц измерения, включенных в стандарт HL7.

7.1.7.2 Diet.carbohydrateQuantity:: PQ (0..1)

Определение: предоставляемое количество углеводов (грамм) в день.

Обсуждение: для диабетической диеты типично ограничение количества усваиваемых углеводов в день (например, 240 г/д). Это ограничение может быть передано как значение атрибута carbohydrateQuantity.

7.1.8 Класс FinancialContract (в предметной области Acts)

Код класса: FCNTRCT.

Атрибуты класса FinancialContract:

- paymentTermsCode:: CE

Класс FinancialContract является специализацией класса Act.

Определение класса FinancialContract: контракт, имеющий денежное выражение.

Примеры: договор страхования, соглашение о закупках.

В следующих подпунктах описаны атрибуты класса FinancialContract.

7.1.8.1 FinancialContract.paymentTermsCode:: CE (0..1)

Словарный домен: PaymentTerms (CWE)

Определение: сроки платежа по контракту или обязательствам.

Примеры: «в течение 30 дней с даты выставления счета», «по получению счета», «по завершению услуги».

7.1.9 Класс FinancialTransaction (в предметной области Acts)

Код класса: XACT.

Атрибуты класса FinancialTransaction:

- amt:: MO
- creditExchangeRateQuantity:: REAL
- debitExchangeRateQuantity:: REAL

Класс FinancialTransaction является специализацией класса Act.

Определение класса FinancialTransaction: действие, представляющее движение денежных сумм между двумя счетами.

Обсуждение: финансовые операции всегда осуществляются между двумя счетами (дебет и кредит), но могут быть случаи, когда один или оба счета определяются более общей моделью или наследуются от нее.

Если у экземпляра класса FinancialTransaction атрибут moodCode имеет значение ORD, то этот экземпляр трактуется как требование выполнения операции.

Если атрибут moodCode имеет значение EVN, то этот экземпляр класса FinancialTransaction содержит информацию об уже выполненной операции.

Примеры: затраты на услугу; оплата услуги; оплата счета.

В следующих подпунктах описаны атрибуты класса FinancialTransaction.

7.1.9.1 FinancialTransaction.amt:: MO (0..1)

Определение: указывает денежную сумму, перемещаемую с дебета на кредит счета.

Обсуждение: если денежные единицы суммы отличаются от денежных единиц дебита или кредита счета, то должен быть указан используемый обменный курс.

7.1.9.2 FinancialTransaction.creditExchangeRateQuantity:: REAL (0..1)

Определение: десятичное число, указывающее обменный курс между валютой кредита счета и валютой операции.

Примеры: при регистрации оплаты услуг, оцененных в мексиканских песо (MXP) и оплаченных в долларах США (USD) со счета, который ведется в канадских долларах (CAD), в качестве обменного курса валюты кредита должно быть передано десятичное число r, для которого «y (USD) * r = x (CAD)».

Обоснование: с помощью этого атрибута операция может быть указана в валюте, отличающейся от валюты кредита и дебета. Это также позволяет вести дебет и кредит в разных валютах.

7.1.9.3 FinancialTransaction.debitExchangeRateQuantity:: REAL (0..1)

Определение: десятичное число, указывающее обменный курс между валютой дебета счета и валютой операции.

Примеры: при регистрации оплаты услуг, оцененных в мексиканских песо (MXP) и оплаченных в долларах США (USD) со счета, который ведется в канадских долларах (CAD), в качестве обменного курса валюты дебита должно быть передано десятичное число r, для которого «y (USD) * r = x (MXP)».

Обоснование: с помощью этого атрибута операция может быть указана в валюте, отличающейся от валюты кредита и дебета. Это также позволяет вести дебет и кредит в разных валютах.

7.1.10 Класс InvoiceElement (в предметной области Acts)

Код класса: INVE.

Атрибуты класса InvoiceElement:

- modifierCode:: SET<CE>
- unitQuantity:: RTO<PQ,PQ>
- unitPriceAmt:: RTO<MO,PQ>
- netAmt:: MO
- factorNumber:: REAL
- pointsNumber:: REAL

Класс InvoiceElement является специализацией класса Act.

Определение класса InvoiceElement: действие, представляющее объявление и обоснование суммы, подлежащей оплате.

Обсуждение: эта информация является частью «расшифровки» счета. Она часто объединяется с информацией о финансовой операции, представляющей сумму, подлежащую оплате, сумму, согласованную к оплате, или фактически оплаченную сумму.

Чтобы разбить один элемент счета-фактуры, передаваемый в экземпляре класса InvoiceElement, на несколько составляющих элементов, можно использовать рекурсивные отношения.

Если у экземпляра класса InvoiceElement атрибут moodCode имеет значение DEF, то этот экземпляр описывает «возможное» согласование строки счета-фактуры при его будущем рассмотрении. Если атрибут moodCode имеет значение EVN, то этот экземпляр класса InvoiceElement содержит информацию о сумме, которую должен оплатить получатель.

В следующих подпунктах описаны атрибуты класса InvoiceElement.

7.1.10.1 InvoiceElement.modifierCode:: SET<CE> (0..*)

Словарный домен: InvoiceElementModifier (CWE)

Определение: указывает модификатор кода, передаваемого в атрибуте code, для представления дополнительной информации об элементе счета-фактуры.

Примеры: удаленная территория, внеурочное обслуживание.

Обоснование: этот модификатор не рассматривается как часть пре-координированной классификации с кодом, передаваемым в атрибуте code, поскольку система кодирования значений атрибута modifierCode не обязательно специально является детализацией системы кодирования атрибута code. Это не соответствует смыслу имени modifier (модификатор), поскольку обычно словарный домен модификатора должен быть определен как часть базового системы кодирования или должен быть разработан специально для нее.

7.1.10.2 InvoiceElement.unitQuantity:: RTO<PQ,PQ> (0..1)

Определение: описание количества товара или услуги, которое должно быть оплачено или уже оплачено.

Примеры: 4 часа, 4 мг, 4 коробки, 15 штук из контейнера, содержащего 1000 штук, и т. д.

Обсуждение: каждый экземпляр класса InvoiceElement, описывающий элемент счета-фактуры, подлежащий оплате или оплаченный, идентифицируется кодом товара или услуги, передаваемым в атрибуте InvoiceElement.code. В некоторых случаях этот код берется из пре-координированной классификации и идентифицирует контейнер (например, универсальный код продукта (УКП), присвоенный контейнеру, содержащему 1000 таблеток, и другой УКП-код для контейнера, содержащего 100 таких же таблеток). УКП-код используется при выставлении счетов, но при его применении возникает необходимость указать в форме дроби, что только часть контейнера (например, флакона) подлежит оплате или оплачена. Если товар, информация о котором передается в экземпляре класса InvoiceElement, не является контейнером, то знаменатель дроби не указывается.

Например, пусть оплачены 15 таблеток из контейнера, содержащего 1000 таблеток. В этом случае числитель дроби может быть указан как «15{таблеток}» или просто «15», а знаменатель как «1000{флакон}» или просто «1000» (см. обсуждение, следующее за обоснованием использования описательного текста для исчисляемых величин).

Ограничения: единицы товара или услуги должны быть ограничены такими измеряемыми единицами как литры, миллиграммы и часы. Не измеряемые, но исчисляемые единицы, например, короба, пакеты, посещения, таблетки и контейнеры, не должны указываться в компоненте единиц измерения типа данных PQ иначе как в аннотации, указанной в фигурных скобках ({xxx}). См. спецификацию типов данных в документе Data Types Part II Unabridged Specification, Appendix A: Unified Code for Units of Measure.

Указание исчисляемых единиц может быть выполнено с помощью следующих методов:

- определить исчисляемую единицу в атрибуте InvoiceElement.code. В этом случае конкретное значение атрибута InvoiceElement.code будет содержать указание, что предмет, описанный в данном экземпляре класса InvoiceElement, является упаковкой, содержащей 20 элементов. Для упаковки, содержащей 40 элементов, должен использоваться другой код, и т. д.

Например, если значение InvoiceElement.code соответствует упаковке, содержащей 20 элементов, а значение атрибута количества InvoiceElement.unitQuantity = 2 (без единиц измерения), то данный экземпляр класса InvoiceElement описывает 2 упаковки по 20 элементов в каждой, т. е. всего 40 элементов.

Если требуется указать больше деталей (например, описать состав, упаковку, производителя товара), то эта информация должна быть описана как экземпляр класса Entity, связанный с данным экземпляром класса InvoiceElement с помощью экземпляров классов Role и Participation, при этом атрибут Participation.typeCode должен иметь значение PRD.

7.1.10.3 InvoiceElement.unitPriceAmt:: RTO<MO,PQ> (0..1)

Определение: стоимость единицы товара или услуги.

Ограничения: при указании дроби числитель должен иметь тип данных MO, а знаменатель — тип данных PQ. Детали указания дроби см. в описании атрибута unitQuantity.

Примеры: $0.20/мг; $250/день; $50.

7.1.10.4 InvoiceElement.netAmt:: MO (0..1)

Определение: указывает полную стоимость элемента счета-фактуры, включая сумму его компонентов.

Определение: для листовых элементов счета-фактуры эта сумма вычисляется по формуле unitQuantity*unitPriceAmt[*factorNumber][*pointsNumber]. Для группировок элементов счета-фактуры эта сумма вычисляется как сумма значений атрибутов netAmt всех элементов группы.

7.1.10.5 InvoiceElement.factorNumber:: REAL (0..1)

Определение: указывает коэффициент, используемый при определении полной стоимости оказанных услуг и/или полученных товаров.

Примеры: 10 (число услуг в качестве единиц) * $3.00 (стоимость единицы) * 1.5 (коэффициент) = $45.00 (сумма).

Обсуждение: это понятие часто используется в Европе, чтобы устанавливать разные цены услуги для обязательного и добровольного медицинского страхования.

Простейшая формула для вычисления полной суммы такова: unitQuantity * unitPriceAmount = netAmt.

С помощью коэффициента можно учесть скидки или доплаты, применяемые к полной сумме. Например, с учетом коэффициента формула для вычисления полной суммы примет следующий вид: unitQuantity * unitPrice (Cost/Point) * factorNumber = netAmt. См. аналогичное примечание к описанию атрибута InvoiceElement.pointsNumber, когда стоимость вычисляется с помощью баллов. При применении коэффициента и баллов формула для вычисления полной суммы примет следующий вид: unitQuantity * unitPriceAmt * pointsNumber * factorNumber = netAmt

7.1.10.6 InvoiceElement.pointsNumber:: REAL (0..1)

Определение: этот атрибут используется в ситуации, когда количество товара или услуги выражается в «баллах», позволяющих задать весовой коэффициент к стоимости товара или услуги (основанный на трудоемкости, стоимости и/или интенсивности ресурса).

Примеры: 5 (число услуг в качестве единиц) * 3 (число баллов, присвоенных одной услуге) * $20.00 (стоимость балла) = $300.00 (сумма).

Обсуждение: такой подход широко используется в системах, где предоставленные услуги оцениваются в «условных единицах трудоемкости» (УЕТ), которым назначается фиксированная цена. В этом случае организация может пересматривать цены услуг, увеличивая или уменьшая стоимость УЕТ, чтобы учесть инфляцию, перегрузку и т. д.

Простейшая формула для вычисления полной суммы такова: unitQuantity * unitPriceAmount = netAmt.

Понятие баллов может использоваться для расценки услуг и/или товаров, при которой количество услуг или товара измеряется в баллах, а одному баллу назначается определенная стоимость в долларах. В этом случае для расчета полной стоимости используется следующая формула: unitQuantity * pointsNumber * unitPriceAmt = netAmt.

См. соответствующее примечание к описанию атрибута factorNumber. При использовании коэффициентов и баллов для вычисления полной суммы применяется следующая формула: unitQuantity * unitPriceAmt * pointsNumber * factorNumber = netAmt.

7.1.11 Класс ManagedParticipation (в предметной области Acts)

Атрибуты класса ManagedParticipation:

- id:: SET<II>
- statusCode:: CS>

Класс ManagedParticipation является специализацией класса Participation.

Определение класса ManagedParticipation: участие, которое может меняться с течением времени, в связи с чем его состоянием и идентичностью надо управлять.

Примеры: лечащий врач пациента может уйти в отпуск, поэтому важно знать, когда его участие возобновится.

Обоснование: класс ManagedParticipation определен как подкласс класса Participation, чтобы явно указать, что не все участия не имеют состояния. В общем случае, если о подзадаче, реализуемой с помощью участия, надо иметь больше информации, и этой подзадачей надо управлять, то вместо применения парадигмы участия НАДО моделировать эту подзадачу как компонент основного действия.

Однако в некоторых случаях представление о том, в чем именно состоят эти подзадачи и что именно выполняют участники, является не очень определенным и поэтому их моделирование в форме компонентов оказывается неоднозначным или затруднительным. Для таких случаев предназначен класс ManagedParticipation, который расширяет базовый класс Participation двумя атрибутами: идентификатором id и кодом состояния statusCode. Классы ManagedParticipation должны применяться с чрезвычайными предосторожностями, чтобы избежать путаницы с действиями, моделируемыми в виде классов Act, и не создавать для участий инфраструктуру управления, подобную той, что уже существует для действий.

В следующих подпунктах описаны атрибуты класса ManagedParticipation.

7.1.11.1 ManagedParticipation.id:: SET<II> (0..*)

Определение: уникальный идентификатор, с помощью которого можно идентифицировать конкретный экземпляр класса ManagedParticipation среди всех экземпляров этого класса, имеющих ассоциации с тем же самым экземпляром класса Act и с тем же самым экземпляром класса Role. (См. описание класса ManagedParticipation.)

7.1.11.2 ManagedParticipation.statusCode:: CS (0..1)

Словарный домен: ManagedParticipationStatus (CNE)

Определение: код, указывающий, описывает ли экземпляр класса ManagedParticipation готовящееся, активное, завершенное или отмененное участие. (См. описание класса ManagedParticipation.)

7.1.11.3 Переходы состояний класса ManagedParticipation

Диаграмма перехода состояний класса ManagedParticipation показана на рисунке 7. Управляемое участие может иметь следующие состояния:

- active (активно) — подсостояние состояния normal: это состояние отражает тот факт, что участие продолжается;
- cancelled (отменено) — подсостояние состояния normal: участие было отменено до того, как стало активным;
- completed (завершено) — подсостояние состояния normal: участие успешно завершено;
- normal (нормальное): «типичное» состояние. Исключает состояние nullified, которое указывает, что экземпляр участия был создан по ошибке;
- nullified (аннулировано): это состояние является терминальным состоянием экземпляра участия, созданного по ошибке;
- pending (готовящееся) — подсостояние состояния normal: это состояние отражает тот факт, что участие еще не стало активным.

Между состояниями действия возможны следующие переходы:

- revise (пересмотреть) — из состояния active в состояние active;
- complete (завершить) — из состояния active в состояние completed;
- reactivate (активизировать заново) — из состояния completed в состояние active;
- revise (пересмотреть) — из состояния completed в состояние completed;
- nullify (аннулировать) — из состояния normal в состояние nullified;
- create (создать) — из начального (пустого) состояния в состояние active;
- create (создать) — из начального (пустого) состояния в состояние completed;
- create (создать) — из начального (пустого) состояния в состояние pending;
- activate (активизировать) — из состояния pending в состояние active;
- cancel (отменить) — из состояния pending в состояние cancelled;
- revise (пересмотреть) — из состояния pending в состояние pending.

7.1.12 Класс Observation (в предметной области Acts)

Код класса: OBS.

Атрибуты класса Observation:

- modifierCode:: SET<CE>
- unitQuantity:: RTO<PQ,PQ>
- unitPriceAmt:: RTO<MO,PQ>
- netAmt:: MO
- factorNumber:: REAL
- pointsNumber:: REAL

Класс Observation является генерализацией следующих классов:

- DiagnosticImage
- PublicHealthCase

Класс Observation является специализацией класса Act.

Определение класса Observation: действие исследования, т. е. действие получения информации о субъекте и сообщения этой информации. Непосредственным и основным результатом исследования (постусловие) является новая информация о субъекте. Исследования нередко осуществляются с помощью измерений или других сложных методов, но иногда сводятся к простым утверждениям.

Обсуждение: структура многих результатов исследований может быть представлена в форме пар имя-значение, при этом атрибут Observation.code (унаследованный от класса Act) — это имя свойства, а атрибут Observation.value — значение свойства. Такая конструкция часто называется «переменной» (т. е. именованное свойство, которому может быть присвоено значение). Таким образом, класс Observation всегда используется для передачи общих пар имя-значение, т. е. переменных, даже если вычисление переменной и не является результатом сложного метода исследования. Оно может быть просто ответом на вопрос, или утверждением, или присваиванием значения параметру.

Как и все другие специализации класса Act, класс Observation используется для описания того, что сделано, и в случае класса Observation такое описание указывает, что было в действительности выявлено («результаты» или «ответы»), и эти «результаты» или «ответы» являются частью исследовании и не расщепляются на объекты других типов.

Исследование может состоять из нескольких других исследований (компонентов), у каждого из которых свои значения атрибутов Observation.code и Observation.value. В этом случае составное исследование может не иметь собственного атрибута Observation.value. Например, дифференцированный подсчет белых клеток крови включает в себя отдельные результаты подсчета гранулоцитов, лейкоцитов и других нормальных или аномальных клеток крови (к примеру, незрелые клетки). Поэтому экземпляр класса Observation, описывающий составное исследование (подсчет белых клеток крови), может не иметь собственного значения (хотя его и можно получить, просуммировав все счетчики-компоненты). Таким образом, всякое действие, которое по своей природе является действием получения информации о субъекте и сообщения этой информации, представляется в форме экземпляра класса Observation независимо от того, имеет ли оно собственное простое значение или состоит из нескольких других исследований.

В следующих подпунктах описаны атрибуты класса Observation.

7.1.12.1 Observation.value:: ANY (0..*)

Определение: Информация, которая создана или определена действием исследования.

Ограничения: если иное не указано, то атрибут Observation.value может иметь любой тип данных.

Тип данных, который может принимать значение атрибута Observation.value, зависит от типа исследования и обычно указывается в описании исследования или определяется с помощью простого правила по типу исследования, передаваемому в атрибуте code.

Далее приведены общие рекомендации по выбору типа данных:

В количественных результатах, в основном, используется тип данных Physical Quantity (PQ). Этот тип данных, по существу, представляет собой вещественное число с единицей измерения. Это общие предпочтения для всех числовых значений. Некоторые исключения описаны далее.

Числовые значения не должны быть передаваться в форме простой строки символов (тип данных ST).

Для титра (например, 1:64) и очень немногих других отношений используется тип данных RTO. У титров числителем и знаменателем отношения являются целые числа (например, 1:128). В других отношениях могут использоваться разные количественные типы данных, например, «цена» с типом данных MO.

Иногда по местным соглашениям для титров передаются только знаменатели (например, 32 вместо 1/32). Такие соглашения могут вносить путаницу и в сообщениях стандарта HL7 вместо них должны использоваться правильные отношения.

Для передачи значений индекса (числа без единицы измерения) используется вещественный тип данных (REAL). Если количество не имеет подходящей единицы измерения, то его можно передать как вещественное число. В качестве альтернативы можно использовать тип данных PQ с безразмерной единицей (например, 1 или %). Целое число можно передавать только в том случае, если согласно определению исследования результатами могут являться только целые числа, что является редким случаем, например, если результат исследования имеет перечислимые значения (см. ниже).

Диапазоны (например, <3; 12—20) должны быть представлены в форме диапазонов физических количеств (тип данных IVL<PQ>) или в форме интервалов значений других количественных типов данных. Иногда такие интервалы используются, чтобы указать неопределенность измеренного значения. Однако на этот случай имеются специальные расширения типов данных.

Для передачи перечислимых значений (например, +, ++, +++; или I, IIa, IIb, III, IV) используется тип данных CO (coded ordinal — кодированное перечислимое значение).

Номинальные результаты («таксоны», например, тип организма) передаются с помощью любого из кодируемых типов данных (CD, CE), которые указывают, по меньшей мере, код, систему кодирования и необязательный исходный текст, преобразование в другую систему кодирования, а иногда — еще и квалификаторы.

Для представления мультимедийных данных используется тип данных ED (Encapsulated Data — инкапсулированные данные). С его помощью можно передать изображение (например, рентгенограмму грудной клетки) или видеозапись (например, результат коронарной ангиографии или эхокардиограмму) в виде вложенных двоичных данных или ссылки на внешние адреса, откуда они могут быть загружены по запросу.

Записи биосигналов можно передавать, используя шаблоны коррелируемых последовательностей результатов (Correlated Observation Sequences), обеспечивающих передачу всей необходимой информации в структуре, предложенной в стандарте HL7. Эту информацию можно также передать, вложив ее в значение типа ED. Это позволяет использовать другие форматы цифрового кодирования биосигналов, кроме предложенных в стандарте HL7, а также передавать ссылки на внешние адреса, откуда оцифрованные записи биосигналов могут быть загружены по требованию.

Иногда для передачи формализованных выражений, для которых не подходит ни один из указанных выше типов данных, может использовать строковый тип данных. Однако строковый тип данных не должен использоваться, если значение может быть представлено одним из существующих типов данных.

Временные метки не должны передаваться как значения результатов исследования, если для них можно найти более подходящие места, например, атрибут Act.effectiveTime некоторого действия (к примеру, «время получения биоматериала лабораторией» можно передать в атрибуте effectiveTime действия, описывающее транспорт биоматериала в лабораторию, а не как результат исследования).

Множества значений любого типа данных, перечисляемые данные, а также интервалы часто используются в экземплярах класса Observation, описывающих критерии (т.е. в экземплярах, у которых атрибут moodCode имеет значение EVN.CRT), чтобы указать «референтные пределы» или «пороговые значения» (для тревожных сигналов) и т. д.

Для последовательностей результатов (повторяемые измерения тех же свойств в течение относительно короткого времени) используется тип данных LIST (list — список). Дополнительные детали см. в спецификации коррелируемых последовательностей результатов (Correlated Observation Sequences).

Степень неопределенности значений можно указать, используя расширения типа данных вероятности и распределения вероятности (UVP, PPD). Для передачи статистики абсолютных частот категорий вполне пригоден тип мультимножества BAG.

7.1.12.2 Observation.interpretationCode:: SET<CE> (0..*)

Словарный домен: ObservationInterpretation (CWE)

Определение: один или несколько кодов, указывающих грубую качественную интерпретацию исследования, например, «норма», «патология», «ниже нормы», «возрастает», «устойчивый», «чувствительный» и т. д.

Обсуждение: эти коды интерпретации иногда называют «флагами аномалий», однако оценка нормы представляет собой только один пример грубой интерпретации и часто не применима. Например, оценка чувствительности микроорганизмов не имеет никакого отношения к «норме», и при любом исследовании патологического состояния нет смысла говорить о норме, так как патологические состояния никогда не считаются «нормальными».

7.1.12.3 Observation.methodCode:: SET<CE> (0..*)

Словарный домен: ObservationMethod (CWE)

Определение: код, указывающий дополнительную информацию о методике или способе исследования.

Примеры: способы измерения кровяного давления (артериальная пункция, сфигмоманометр (Riva-Rocci), сидя, лежа на спине и т. д.).

Ограничения: при всех исследованиях метод их выполнения частично определяется по значению атрибута Act.code. В этом случае атрибут methodCode не должен был бы использоваться вообще. Однако в тех случаях, когда метод исследования надо указать с большей точностью, нежели это позволяет значение атрибута Act.code, необходимую дополнительную информацию можно передать в атрибуте methodCode. При этом обработка информации об исследовании, осуществляемая информационной системой или процессом, не должна зависеть от детальных данных, переданных в атрибуте methodCode в дополнение к информации о методе исследования, вытекающей из значения атрибута Act.code.

Если атрибут methodCode используется для указания деталей метода, который идентифицируется по значению атрибута Act.code, то значение атрибута methodCode не должно противоречить этому методу.

Обсуждение: при всех исследованиях метод их выполнения частично определяется по типу исследования (определению исследования, атрибут Act.code) и эту косвенную информацию о методе не надо явным образом указывать в атрибуте Observation.methodCode. Например, в классификации LOINC многим видам исследований присвоены разные коды, если методики исследования различаются и это может практически повлиять на интерпретацию результатов исследования. Например, в этой классификации проводится различие между исследованием чувствительности к антибиотикам методом «минимальной ингибирующей концентрации» (МИК) и «методом диффузии в агаре» (Кирби-Бауэр), так что этим видам исследований специально присвоены разные коды. Следовательно, значение атрибута methodCode может лишь служить дополнительным квалификатором, указывающим то, что не выводится непосредственно из значения атрибута Act.code.

Кроме того, некоторые вариации методов могут быть связаны с конкретным используемым устройством. Но значение атрибута methodCode не должно использоваться для указания конкретного устройства или использованного диагностикума. Такую информацию надо связывать с данным экземпляром класса Observation, используя экземпляр класса Participation, у которого атрибут typeCode имеет значение DEV (device — устройство).

7.1.12.4 Observation.targetSiteCode:: SET<CD> (0..*)

Словарный домен: ActSite (CWE)

Определение: код, указывающий анатомическую локализацию или систему организма, являющуюся предметом наблюдения, если такая информация не вытекает из определения исследования или из значения атрибута Act.code.

Ограничения: если значение атрибута targetSiteCode присутствует, то оно не должно противоречить той информации об анатомической локализации или системе организма, которая вытекает из определения исследования или из значения атрибута.

Обсуждение: в большинстве случаев исследуемая анатомическая локализация вытекает из определения исследования, значения атрибута Act.code или значения атрибута Observation.value. Например, «шумы сердца» заведомо относятся к сердцу. Этот атрибут используется только в тех случаях, когда анатомическая локализация должна быть детализирована, например, чтобы различать правую и левую сторону, и т. д.

Если предметом исследования не является человек или животное, то этот атрибут используется аналогичным образом для указания структурного ориентира предмета исследования. Например, если предмет — озеро, то исследуемой локализацией может быть приток или исток, и т. д. Если предметом является лимфоузел, то в качестве локализации можно указать «ворота», «периферию» и т. д.

7.1.13 Класс PatientEncounter (в предметной области Acts)

Код класса: ENC

Атрибуты класса PatientEncounter:

- admissionReferralSourceCode:: CE
- lengthOfStayQuantity:: PQ
- dischargeDispositionCode:: CE
- acuityLevelCode:: CE
- preAdmitTestInd:: BL
- specialCourtesiesCode:: SET<CE>
- specialArrangementCode:: SET<CE>

Класс PatientEncounter является специализацией класса Act.

Определение класса PatientEncounter: взаимодействие между пациентом и поставщиком (поставщиками) медицинской помощи с целью предоставления одной или нескольких услуг в сфере здравоохранения. К таким услугам относится в том числе оценка состояния здоровья.

Примеры: амбулаторное посещение нескольких отделений, медицинская помощь на дому (включая физиотерапию), госпитализация, посещение травмпункта, полевая медицинская помощь (например, при дорожном происшествии), посещение кабинета врачебного приема, производственная терапия, телефонный звонок.

В следующих подпунктах описаны атрибуты класса PatientEncounter.

7.1.13.1 PatientEncounter.admissionReferralSourceCode:: CE (0..1)

Словарный домен: EncounterReferralSource (CWE)

Определение: источник направления, инициировавшего данное обращение.

7.1.13.2 PatientEncounter.lengthOfStayQuantity:: PQ (0..1)

Определение: общее время, которое субъект проведет или провел в медицинской организации как часть обращения.

Обсуждение: фактическое число дней пребывания не может быть получено вычитанием даты выписки из даты поступления из-за возможных отпусков (на выходные).

7.1.13.3 PatientEncounter.dischargeDispositionCode:: CE (0..1)

Словарный домен: EncounterDischargeDisposition (CWE)

Определение: код, указывающий фактическое место выписки пациента во время выписки (например, выписан домой, умер, выписался вопреки рекомендациям врача, и т. д.).

7.1.13.4 PatientEncounter.acuityLevelCode:: CE (0..1)

Словарный домен: EncounterAcuity (CWE)

Определение: код, указывающий оценку состояния заболевания (сложности лечения, интенсивности использования ресурсов лечения) при поступлении пациента.

7.1.13.5 PatientEncounter.preAdmitTestInd:: BL (0..1)

Определение: индикатор необходимости предварительного выполнения лабораторных анализов.

7.1.13.6 PatientEncounter.specialCourtesiesCode:: SET<CE> (0..*)

Словарный домен: EncounterSpecialCourtesy (CWE)

Определение: код, указывающий признак особого обслуживания пациента. Например, обычное обслуживание, особое обслуживание, особое профессиональное обслуживание, обслуживание очень важного лица.

7.1.13.7 PatientEncounter.specialArrangementCode:: SET<CE> (0..*)

Словарный домен: SpecialArrangement (CWE)

Определение: код, указывающий тип специальных мер, которые должны быть предприняты до обращения пациента (например, инвалидное кресло, носилки, переводчик, дежурный врач, собака-поводырь). Если атрибут moodCode экземпляра класса PatientEncounter указывает будущее действие, то этот экземпляр может использоваться для указания специальных мер, которые надо предпринять перед ожидаемым поступлением пациента.

7.1.14 Класс Procedure (в предметной области Acts)

Код класса: PROC

Атрибуты класса Procedure:

- methodCode:: SET<CE>
- approachSiteCode:: SET<CD>
- targetSiteCode:: SET<CD>

Класс Procedure является специализацией класса Act.

Определение класса Procedure: непосредственным и основным результатом процедуры (постусловие) является измененное физическое состояние субъекта.

Примеры: изменения физического состояния могут включать в себя нарушение целостности некоторой поверхности тела (например, разрез при хирургической операции), выполнение консервативных процедур, например, вправление вывихнутого сустава, физиотерапию, манипуляции хиропрактика, массаж, бальнеотерапия, иглоукалывание, шиятсу и т. д. За пределами клинической медицины можно привести такие примеры процедур как изменение окружающей среды (например, спрямление рек, осушение болот, возведение дамб), ремонт или модернизация машин и т. д.

Обсуждение: применительно к клинической медицине процедура — один из нескольких типов клинических действий, например, исследование, лекарственные назначения и коммуникативные воздействия (к примеру, обучение, консультация, психотерапия, представленные в форме экземпляров класса Act без специальных атрибутов). Понятие процедуры не поглощает эти другие понятия и, в свою очередь, не поглощается ими. Примечательно, что оно не включает в себя все действия, целью которых является вмешательство или лечение. Будет ли физическое изменение полезным субъекту или предполагается полезным,

не имеет значения, существенно лишь то, что целью действия является изменение физического состояния субъекта.

Выбор между способами представлениями реальных действий в форме экземпляров классов основан на том, применимы ли специфические свойства процедуры и будет ли физическое изменение необходимым постусловием деятельности или ее части. Например, получение рентгеновского изображения иногда называется «процедурой», но в ЭИМ оно не отображается на экземпляр класса Procedure, поскольку целью рентгеновского снимка не является изменение физического состояния тела.

При многих видах клинической деятельности отдельные действия, которые по своей природе являются исследованиями и процедурами, объединяются в одно составное действие. Например, при выполнении процедур лечебной радиологии (к примеру, интраартериальный тромболиз) производятся как исследования, так и лечение, и большинство хирургических процедур включает в себя намеренные и документированные шаги исследований. Такие клинические воздействия лучше всего представлять в виде нескольких действий-компонентов, каждый из которых имеет соответствующий тип.

В следующих подпунктах описаны атрибуты класса Procedure.

7.1.14.1 Procedure.methodCode:: SET<CE> (0..*)

Словарный домен: ProcedureMethod (CWE)

Определение: указывает средства или методику выполнения процедуры.

Обсуждение: у любой процедуры может быть несколько разных методов. Хотя они обеспечивают, в основном, те же самые результаты, для более тщательной интерпретации протокола процедуры может требоваться указание, какой именно метод был применен (например, открытая или лапароскопическая холецистэктомия). Понятия метода могут быть «пре-координированы» в определении действия. Поскольку существует много возможных методов, существенно зависящих от конкретных видов процедур, конструирование словарного домена, который бы охватывал все эти методы, представляется затруднительным. Однако для конкретного типа процедуры такую систему кодирования сконструировать можно. Таким образом, пользователь, направляющий пациента на процедуру, может указать конкретный метод ее выполнения, выбрав требуемый вариант из такой системы кодирования. Возможные варианты методов выполнения могут быть описаны в справочнике процедур. В записях такого справочника, передаваемых с помощью экземпляров класса Procedure, у которых атрибут moodCode имеет значение DEF, атрибут methodCode может содержать список методов выполнения процедуры, используемый для выбора нужного метода при оформлении направления на процедуру или для проверки полученного протокола процедуры.

7.1.14.2 Procedure.approachSiteCode:: SET<CD> (0..*)

Словарный домен: ActSite (CWE)

Определение: анатомическая локализация или система организма, через которую процедура достигает места своего применения (см. описание атрибута targetSiteCode).

Примеры:

1) При нефрэктомии доступ к почке может быть трансабдоминальным или, главным образом, ретроперитонеальным.

2) Целью катетеризации легочной артерии является легочная артерия, но при этом местом доступа обычно является внутренняя яремная вена или подключичная вена, в шее или, соответственно, в подключичной ямке.

3) Для неинвазивных процедур, например, иглоукалывания, место доступа — прокалываемое место кожи.

Обсуждение: если субъектом процедуры является не человек и не животное, то этот атрибут используется аналогичным образом, чтобы определить структурный ориентир на предмете действия.

Списки мест доступа могут быть «пре-координированы» с описанием процедуры, чтобы исключить неправильный выбор анатомической локализации. Одна и та же информационная структура может использоваться в обоих подходах: пре-координированном и пост-координированном.

7.1.14.3 Procedure.targetSiteCode:: SET<CD> (0..*)

Словарный домен: ActSite (CWE)

Определение. Анатомическая локализация или система организма, являющаяся местом применения процедуры.

Примеры:

1) Местом применения нефрэктомии является правая или левая почка.

2) Местом применения катетеризации легочной артерии является легочная артерия.

3) Для неинвазивных процедур, например, иглоукалывания, местом применения является орган/система, на которые рассчитано воздействие (например, «печень»).

Обсуждение: если субъектом процедуры является не человек и не животное, то этот атрибут используется аналогичным образом, чтобы определить структурный ориентир на предмете действия.

Списки мест применения могут быть «пре-координированы» с описанием процедуры, чтобы исключить неправильный выбор анатомической локализации. Одна и та же информационная структура может использоваться в обоих подходах: пре-координированном и пост-координированном.

7.1.15 Класс PublicHealthCase (в предметной области Acts)

Код класса: CASE

Атрибуты класса PublicHealthCase:

- detectionMethodCode:: CE
- transmissionModeCode:: CE
- diseaseImportedCode:: CE

Класс PublicHealthCase является специализацией класса Observation.

Определение класса PublicHealthCase: описание события, имеющего определенное значение для общественного здоровья, представляется как специализация класса Observation. Обычно это информация об одном или нескольких случаях инфекционного заболевания или другого события, подлежащего регистрации. Описание события общественного здоровья может включать в себя сведения о состоянии здоровья одного лица или сведения о нескольких случаях того же самого заболевания или условия, которые рассматриваются как угроза общественному здоровью. Типичным примером может служить вспышка заболевания. Определение события общественного здоровья (передаваемое в экземпляре класса PublicHealthCase, у которого атрибут moodCode имеет значение DEF) включает в себя описание клинических, лабораторных и эпидемиологических показателей, связанных с заболеванием или условием, влияющим на общественное здоровье. Есть определения событий для условий, которые подлежат регистрации, а также для тех, которые не подлежат. Есть также определения событий вспышек. Определение события общественного здоровья используется здравоохранением для статистики событий и не должно использоваться в качестве клинических рекомендаций, предназначенных для лечения. Примерами служат СПИД, синдром токсического шока, сальмонеллез и связанные с ними симптомы, используемые при определении угрозы.

В следующих подпунктах описаны атрибуты класса PublicHealthCase.

7.1.15.1 PublicHealthCase.detectionMethodCode:: CE (0..1)

Словарный домен: CaseDetectionMethod (CWE)

Определение: код, указывающий метод получения санитарно-эпидемиологической службой информации о событии общественного здоровья. Примерами могут служить сообщение поставщика медицинской помощи, самостоятельное обращение пациента, сообщение лаборатории, отчет о результате исследования угрозы или вспышки, выявление контактов, активное наблюдение, рутинный физикальный осмотр, дородовое исследование, перинатальное исследование, мониторинг профессиональных заболеваний, анализ медицинских карт и т.д.

7.1.15.2 PublicHealthCase.transmissionModeCode:: CE (0..1)

Словарный домен: CaseTransmissionMode (CWE)

Определение: код, указывающий механизм приобретения заболевания живым субъектом, вовлеченным в событие общественного здоровья. Примерами служат половой путь, воздушно-капельный, передача через кровь, трансмиссивный путь, передача через продукты питания, зоонозный путь, внутрибольничная инфекция, механический путь, кожный, врожденный, воздействие внешней среды, неопределенный.

7.1.15.3 PublicHealthCase.diseaseImportedCode:: CE (0..1)

Словарный домен: CaseDiseaseImported (CWE)

Определение: код, указывающий, была ли болезнь, вероятно, приобретена за пределами территории санитарно-эпидемиологической службы, и какова природа межтерриториальных отношений. Примерами возможных значений служат: не занесенная, занесенная из другой страны, занесенная из другого штата, занесенная с другой территории, недостаточно информации для определения.

7.1.16 Класс SubstanceAdministration (в предметной области Acts)

Код класса: SBADM

Атрибуты класса SubstanceAdministration:

- routeCode:: CE
- approachSiteCode:: SET<CD>
- doseQuantity:: IVL<PQ>:: CE
- rateQuantity:: IVL<PQ>
- doseCheckQuantity:: SET<RTO>
- maxDoseQuantity:: SET<RTO>

Класс SubstanceAdministration является специализацией класса Act.

Определение класса SubstanceAdministration: лекарственное назначение, т. е. действие введения или иного применения субстанции к субъекту.

Обсуждение: эффект применения субстанции обычно вызывается биохимическими процессами, однако это не обязательно. Например, лучевую терапию в целом можно описать аналогичным образом, особенно соматическую терапию, к примеру, радиоактивным йодом. К этому же классу действий относится применение химического лечения к определенной области.

Примеры: протокол химиотерапии, рецепт, запись о вакцинации.

В следующих подпунктах описаны атрибуты класса SubstanceAdministration.

7.1.16.1 SubstanceAdministration.routeCode:: CE (0..1)

Словарный домен: RouteOfAdministration (CWE)

Определение: метод введения терапевтического материала внутрь или на поверхность субъекта.

Обсуждение: путь введения лекарственного средства, место его введения (атрибут approachSiteCode) и устройство, используемое для введения, тесно связаны. Все три (если указаны) должны быть хорошо согласованы между собой. В некоторых случаях используемая система кодирования должна быть прекоординирована с одной или несколькими другими системами.

Когда лекарство применяется к окружающей среде или к некоторому физическому месту, то код способа введения указывает это место в его «теле».

Примеры: перорально — per os (PO), сублингвально — sublingual (SL), суппозиторий — rectal (PR), ингаляция — per inhalationem (IH), глазные капли — ophtalmic (OP), назальное применение — nasal (NS), ушное применение — otic (OT), пессарий — vaginal (VG), внутрикожно — intra-dermal (ID), подкожно — subcutaneous (SC), внутривенно — intra-venous (IV), внутрисердечно — intra-cardial (IC)

7.1.16.2 SubstanceAdministration.approachSiteCode:: SET<CD> (0..*)

Словарный домен: ActSite (CWE)

Определение: детальное указание анатомической локализации, куда лекарство вводится или где оно применяется к субъекту.

Обсуждение: этот атрибут необходим только в том случае, если значение атрибута routeCode требует дальнейшего уточнения. Например, если атрибут routeCode имеет значение PO (перорально), то никакая дальнейшая спецификация места введения не требуется. Если, однако, атрибут routeCode имеет значение IV (внутривенно) или IM (внутримышечно), то в атрибуте approachSiteCode можно указать точное место введения (например, правое предплечье или левая дельтовидная мышца соответственно).

Способ введения, место его применения (approachSiteCode) и устройство, используемое для введения, тесно связаны. Все три (если указаны) должны быть хорошо согласованы между собой. В некоторых случаях используемая система кодирования должна быть пре-координирована с одной или несколькими другими системами.

7.1.16.3 SubstanceAdministration.doseQuantity:: IVL<PQ> (0..1)

Определение: количество терапевтического агента или другой субстанции, полученное за один прием.

Обсуждение: доза может определяться или как физическое количество активного ингредиента (например, 200 мг), или как количество единиц назначения (например, таблетки, капсулы, «штуки»). Какой выбрать подход, зависит от исполнителя «потребляемого» участия (который идентифицирует вводимое лекарство). Если потребляемое лекарство имеет неисчисляемую форму дозировки (например, измеряется в миллиграммах или в литрах), тогда доза должна выражаться в этих единицах. Если потребляемое лекарство имеет исчисляемую форму дозировки (таблетки, капсулы, «штуки»), тогда доза должна быть выражена как безразмерное число (т. е. никакая единица измерения не указывается).

Единицы измерения должны быть ограничены такими измеряемыми единицами как литры, миллиграммы. Не измеряемые, но исчисляемые единицы, например, таблетки и капсулы, не должны указываться в компоненте единиц измерения типа данных PQ иначе как в аннотации, указанной в фигурных скобках ({xxx}). См. спецификацию типов данных в документе Data Types Part II Unabridged Specification, Appendix A. Unified Code for Units of Measure.

7.1.16.4 SubstanceAdministration.rateQuantity:: IVL<PQ> (0..1)

Определение: указывает скорость введения субстанции в субъект. Выражается как физическое (экстенсивное) количество, введенное за единицу времени (например, 100 мл/ч, 1 г/сут, 40 ммоль/ч, и т. д.).

Обсуждение: это понятие применимо к непрерывно делимым формам доз (например, жидкости, газы). Если значение указанного атрибута является интервалом, то скорость должна быть в этом интервале. Указанный атрибут определяется как физическое (экстенсивное) количество, введенное за единицу времени, т.е. количество, которое передано в атрибуте doseQuantity, деленное на продолжительность введения.

7.1.16.5 SubstanceAdministration.doseCheckQuantity:: SET<RTO> (0..*)

Определение: в этом атрибуте указано ожидаемое количество субстанции, которое должно быть введено за период времени. Используется как критерий проверки значений, переданных в других атрибутах.

Обсуждение: обычно этот атрибут не используется; он предназначен только для специальных целей. В некоторых странах, в частности, в Японии, есть норма, требующая указывать общую суточную дозу в рецепте и сопутствующей документации. Цель этого требования, очевидно, в том, чтобы обеспечить и облегчить контроль общей назначенной дозы во избежание передозировки (или слишком малой дозировки).

Примеры:

1) При назначении эритромицина в дозировке 250 мг (1 таблетка 3 раза в день) общую суточную дозу можно рассчитать по формуле doseCheckQuantity = doseQuantity (1) * Ingredient.quantity (250 мг) * effectiveTime (3 / сут) = 750 мг/сут.

2) При внутривенном введении ожидаемое количество субстанции в час можно рассчитать по формуле doseCheckQuantity = doseQuantity (100 мл) * Ingredient.quantity (5 мг/л) / rateQuantity (1 час) = 0.5 мг/ч, что можно преобразовать в суточную дозу по формуле doseCheckQuantity = 0.5 мг/ч * 24 ч/сут = 12 мг/сут.

Обоснование: вместо определения атрибута «общая суточная доза», как это было сделано в стандарте HL7 v2.3, здесь введено более общее понятие doseCheckQuantity, имеющее тип данных «отношение» (RTO).

Ограничения: invariant(SubstanceAdministration med, RTO max) where med.doseCheckQuantity.contains(max) {max.numerator.compares(med.doseQuantity); max.denominator.compares(1 s);}. Числитель должен быть в единицах, сопоставимых с единицами измерения значения doseQuantity, а знаменатель должен быть мерой времени.

7.1.16.6 SubstanceAdministration.maxDoseQuantity:: SET<RTO> (0..*)

Определение: указывает максимальное общее количество терапевтической субстанции, которое может быть введено субъекту за период времени.

Обсуждение: этот атрибут особенно полезен, если разрешенная дозировка задается в форме диапазона либо режим введения является переменным или по мере необходимости. С его помощью можно указать предел общего количества субстанции, которое может быть применено за период времени. Чтобы указать разные пределы для разных периодов времени, можно указать несколько значений атрибута maxDoseQuantity.

Примеры: 500 мг/сут; 1200 мг/неделя.

Ограничения: invariant(SubstanceAdministration med, RTO max) where med.maxDoseQuantity.contains(max) {max.numerator.compares(med.doseQuantity); max.denominator.compares(1 s);}. Числитель должен выражаться в единицах, сопоставимых с единицами измерения значения doseQuantity, а знаменатель должен быть мерой времени.

7.1.17 Класс Supply (в предметной области Acts)

Код класса: SPLY

Атрибуты класса Supply:

- quantity:: PQ
- expectedUseTime:: IVL<TS>

Класс Supply является генерализацией класса Diet.

Класс Supply является специализацией класса Act.

Определение класса Supply: действие, включающее в себя передачу материала (товара) от одной сущности к другой.

Обсуждение: информация о передаваемом материале связывается с экземпляром класса Supply с помощью экземпляра класса Participation, у которого атрибут typeCode имеет значение PRD (product — товар). При этом важна точная идентификация материала (производитель, серийный номер и т.д.). Большая часть детальной информации о материале должна передаваться в экземпляре класса Material. Если требуется отдельно описать планирование доставки, доставку и оплату материала, то с экземпляром класса Supply можно связать экземпляр класса Transportation. Для описания услуги отпуска лекарственного средства используется экземпляр класса Supply, связанный с экземпляром класса SubstanceAdministration. В этом случае экземпляр класса SubstanceAdministration описывает применение лекарства, а экземпляр класса Supply — отпуск.

Примеры: заказ простыней, отпуск лекарства, отпуск медицинских расходных материалов со склада.

В следующих подпунктах описаны атрибуты класса Supply.

7.1.17.1 Supply.quantity:: PQ (0..1)

Определение: количество материала, которое было предоставлено или должно быть предоставлено (в зависимости от значения атрибута moodCode).

Обсуждение: этот атрибут может использоваться как альтернатива атрибуту expectedUseTime или вместе с этим атрибутом. Если оба эти атрибута указаны, то значение, указанное в атрибуте quantity, представляет собой количество, которое предполагается израсходовать в течение времени, указанного в атрибуте expectedUseTime.

Единицы измерения должны быть ограничены такими измеряемыми единицами как литры, миллиграммы. Не измеряемые, но исчисляемые единицы, например, таблетки и капсулы, не должны указываться в компоненте единиц измерения типа данных PQ иначе как в аннотации, указанной в фигурных скобках ({xxx}). См. спецификацию типов данных в документе Data Types Part II Unabridged Specification, Appendix A: Unified Code for Units of Measure. Тип «исчисляемой» информации определен информацией сущности «продукта».

7.1.17.2 Supply.expectedUseTime:: IVL<TS> (0..1)

Определение: указывает период времени, в течение которое предполагается использовать предоставляемый материал, или продолжительность времени, в течение которого будет длиться его предоставление.

В некоторых случаях этот атрибут может использоваться вместо атрибута Supply.quantity для косвенного указания предоставляемого количества в форме продолжительности предоставления. Например, применение лекарства в течение 90 дней лечения (вычисление количества основано на дозировке лекарства), количество реактивного топлива на 10 часов полета, и т. д.

Примечание — По возможности количество лучше передавать в атрибуте Supply.quantity, поскольку это точнее. Определение количества по значению атрибута Supply.expectedUseTime всегда будет оценкой, подверженной влиянию внешних факторов.

7.1.18 Класс WorkingList (в предметной области Acts)

Код класса: LIST

Атрибуты класса WorkingList:

- ownershipLevelCode:: CE

Класс WorkingList является специализацией класса Act.

Определение класса WorkingList: динамический список отдельных экземпляров класса Act, составленный для отражения потребностей отдельного сотрудника, бригады или организации по просмотру группы действий, выполняемых в клинических или административных целях.

Обсуждение: группируемые действия связываются с экземпляром класса WorkingList с помощью экземпляров класса ActRelationship, у которых атрибут typeCode имеет значение COMP (component — компонент).

Примеры: список жалоб, список целей лечения, список аллергий, листы назначений и т.д.

В следующих подпунктах описаны атрибуты класса WorkingList.

7.1.18.1 WorkingList.ownershipLevelCode:: CE (0..1)

Словарный домен: ListOwnershipLevel (CWE)

Определение: указывает, какому уровню подчинения предназначен данный рабочий лист — отдельному лицу, бригаде или организации.

7.2 Классы предметной области Entities

7.2.1 Класс Entity (в предметной области Entities)

Код класса: ENT.

Атрибуты класса Entity:

- classCode:: CS
- determinerCode:: CS
- id:: SET<II>
- code:: CD
- quantity:: SET<PQ>
- name:: BAG<EN>
- desc:: ED
- statusCode:: CS
- existenceTime:: IVL<TS>

- telecom:: BAG<TEL>
- riskCode:: SET<CE>
- handlingCode :: CE

Ассоциации класса Entity:

- languageCommunication::(0..*) LanguageCommunication::entity::(1..1) (ассоциация с классом LanguageCommunication, роль entity — сущность)
- playedRole::(0..*) Role::player::(0..1) (ассоциация с классом LanguageCommunication, роль player — исполнитель)
- scopedRole::(0..*) Role::scoper::(0..1) (ассоциация с классом LanguageCommunication, роль scoper — контролер)

Класс Entity является генерализацией следующих классов:

- LivingSubject
- Material
- Organization

Класс Entity является специализацией класса InfrastructureRoot.

Определение класса Entity: сущность (физический предмет, группа физических предметов или организация), способная участвовать в действиях, выполняя некоторую роль.

Обсуждение: сущность — физический объект, который существует, существовал или будет существовать. Единственное исключение — организация, которая, не имея физического облика, обладает другими характеристиками класса Entity. Иерархия специализаций класса Entity включает в себя представления живых существ (включая людей), организации, материал, места и их специализации. Характеристики класса Entity не содержат указаний на выполняемые роли или действия, в которых эти сущности могут участвовать.

Ограничения: понятие сущности не включает в себя события/дела/действия или выполняемые роли (например, пациент, поставщик).

В следующих подпунктах описаны атрибуты класса Entity.

7.2.1.1 Entity.classCode:: CS (1..1) Mandatory

Словарный домен: EntityClass (CNE)

Определение: код, определяемый стандартом HL7 и указывающий вид или категорию экземпляра класса Entity.

Примеры: человек, животное, химическая субстанция, группа, организация.

Обоснование: вследствие чрезвычайно большого числа потенциальных значений, которые должна охватывать система кодирования, представляющая все физические объекты в мире, атрибут classCode указывает как подтип ветви в иерархии специализаций класса Entity, так и высокоуровневый классификатор экземпляра класса Entity. Его можно использовать для ограничения допустимых областей значений атрибута Entity.code.

7.2.1.2 Entity.determinerCode:: CS (1..1) Mandatory

Словарный домен: EntityDeterminer (CNE)

Определение код, определяемый стандартом HL7 и указывающий, представляет ли экземпляр класса Entity вид сущностей (kind-of) или конкретный экземпляр сущности.

Примеры: 1 человек (экземпляр), 3 шприца (вид сущностей, количество которых определено) или население Индианаполиса (вид группы)

Обоснование: экземпляр класса Entity может представлять информацию о конкретном экземпляре сущности (наиболее общий случай), о перечисляемой группе сущностей, обладающих общими характеристиками, или об общем виде сущностей. Эти варианты представлений различаются с помощью значения атрибута determinerCode.

7.2.1.3 Entity.id:: SET<II> (0..*)

Определение: уникальный идентификатор экземпляра класса Entity.

Обоснование: для успешной передачи данных требуется лишь наличие у сущности уникального идентификатора. Но поскольку различные системы ведут различные базы данных, то одной и той же сущности в них могут быть присвоены разные идентификаторы. Учтите, что идентификатор экземпляра сущности представляет собой чистый идентификатор, а не классификатор. Для представления серийных номеров, присваиваемых производителями, или артикулов в каталогах поставщиков, или инвентарных номеров материалов и товаров, информация о которых передается в экземплярах класса Material, могут использоваться атрибуты Role.id, что позволяет более ясно отразить тот факт, что такие номера или артикулы присвоены определенной стороной, связанной с этим материалом или товаром.

7.2.1.4 Entity.code:: CD (0..1)

Словарный домен: EntityCode (CWE)

Определение: значение, указывающее определенный вид сущности, представляемой в форме экземпляра класса Entity.

Примеры: здание больницы, доберман-пинчер, пробирка для сбора крови, биоптат.

Обоснование: система кодирования значений этого атрибута зависит от значения атрибута Entity.classCode, например, своя система кодирования живых субъектов (таксономии животного и растительного мира), своя для химических субстанций (например, код IUPAC), системы кодирования организаций, страховых компаний, органов государственного управления, больниц, парков, озер, шприцев и т. д. В принципе система кодирования значений Entity.code может быть до такой степени детализирована, что будет содержать единственное значение. Примером может служить код CDC производителя вакцины, моделируемый как словарь понятий, в котором каждое понятие фактически относится к единственному экземпляру.

7.2.1.5 Entity.quantity:: SET<PQ> (0..*)

Определение: число сущностей или количество сущности, с соответствующими единицами измерения, что определяется значением атрибута Entity.determinerCode.

Примеры: для неопределенных видов значение этого атрибута представляет собой справочное значение для спецификации пропорций ингредиентов или компонентов (например, используя ассоциации RoleLink с экземплярами класса Role, у которых атрибут RoleLink.typeCode имеет значение PART, означающий «имеет часть», «имеет компонент», или «имеет содержание»). К примеру, популяция, в составе которой 60 % женщин, может быть описана в виде связи «имеет часть» экземпляра класса Person, у которого атрибут quantity имеет значение 100, с экземпляром класса Person, у которого атрибут quantity имеет значение 60, а атрибут sex — значение F, т. е. женский пол. Амоксициллин в таблетках с дозировкой 500 мг может быть описан в виде связи «имеет ингредиент» экземпляра класса Material, у которого атрибут formCode имеет значение TAB (tablet — таблетка), а атрибут quantity имеет значение 1, с экземпляром класса Material, у которого атрибут code имеет значение кода амоксициллина, а атрибут quantity имеет значение 500 мг. Раствор глюкозы 5 % (D5W) можно описать в виде связи «имеет ингредиент» экземпляра класса Material, у которого атрибут code имеет значение D5W, а атрибут quantity — значение 1 кг, с экземпляром класса Material, у которого атрибут code имеет значение кода глюкозы, а атрибут quantity имеет значение 50 г.

Количественные отношения, специфичные для материалов, выражаются, используя тот факт, что тип данных этого атрибута — множество физических количеств (SET<PQ>). Если это множество имеет более одного значения, то каждый его элемент рассматривается как определяющий одно и то же количество материала. Например, для одного литра воды можно было указать множество физических количеств 1 л, 1 кг, 55,56 моль, задающих, соответственно, объем, массу и количество вещества для одного и того же самого количества воды, что эквивалентно плотности массы (объемной массе) 1 кг/л и молярной массе (18 г/моль). Для глюкозы можно указать множество значений 180 г, 1 моль в соответствии с ее молярной массой (180 г/моль).

Обсуждение: если экземпляр класса Entity описывает часть субстанции, то атрибут quantity указывает количество этой части субстанции. Для неопределенной субстанции (вид) это количество удовлетворяет одновременно двум целям: а) предоставляет информацию об отношениях между определенными количествами этой субстанции, и б) является справочным количеством для спецификации ингредиентов или компонентов. Во всех случаях в атрибуте quantity передается экстенсивный вид «количества» (например, число, длина, объем, масса, площадь поверхности, энергия и т.д.). Учтите, что большинство относительных величин и фракций не представляют количества, в частности, фракция массы, концентрация, массовые отношения, проценты, и т.д. не представляют экстенсивные количества и являются запрещенными значениями для этого атрибута.

Ограничения: экземпляры класса Entity, у которых атрибут determinerCode имеет значение INSTANCE, должны иметь атрибут quantity со значением 1. Экземпляры класса Entity, у которых атрибут determinerCode имеет значение QUANTIFIED_KIND, должны иметь атрибут quantity со значением, равным числу членов группы; значение атрибута quantity экземпляров класса Entity, у которых атрибут determinerCode имеет значение KIND, не определено.

7.2.1.6 Entity.name:: BAG<EN> (0..*)

Определение: неуникальный текстовый идентификатор или псевдоним экземпляра класса Entity.

Примеры: собственные имена, псевдонимы, юридические имена людей, мест или предметов.

Обоснование: большинство сущностей имеет общеиспользуемое название, которое может использоваться, чтобы отличить их от других сущностей, но не является уникальным идентификатором.

7.2.1.7 Entity.desc:: ED (0..1)

Определение: текстовое или мультимедийное описание сущности.

Обсуждение: содержание описания не рассматривается в качестве части функциональной информации, передаваемой другой системе. Оно предназначено для восприятия человеком. Вся информация, существенная для автоматизированных функций, должна передаваться в надлежащих атрибутах и связанных объектах.

Обоснование: имена и описания сущностей обычно более значимы человеку, читающему информацию, нежели числовые, мнемонические или сокращенные кодированные значения. В атрибуте desc можно передать дополнительный контекст сущности, предназначенный человеку и не влияющий на функциональные компоненты сообщения.

7.2.1.8 Entity.statusCode:: CS (0..1)

Словарный домен: EntityStatus (CNE)

Определение: значение, указывающее, является ли информация, связанная с экземпляром класса Entity, активной или неактивной в отношении возможности участия в действиях.

7.2.1.9 Entity.existenceTime:: IVL<TS> (0..1)

Определение: интервал времени физического существования сущности.

Примеры: дата рождения/ дата смерти, дата производства/ дата списания.

Ограничения: этот интервал может указывать прошлые, настоящие и будущие периоды времени.

Обоснование: физические сущности имеют определенные периоды существования. Люди рождаются, живут и умирают. Оборудование производится, вводится в эксплуатацию, выводится из эксплуатации и списывается. Значение этого атрибута полезно учитывать при планировании и оценке доступности, а также при ретроспективном анализе.

7.2.1.10 Entity.telecom:: BAG<TEL> (0..*)

Определение: телекоммуникационный адрес сущности.

7.2.1.11 Entity.riskCode:: SET<CE> (0..*)

Словарный домен: EntityRisk (CWE)

Определение: значение, указывающее тип риска или опасности, связанной с сущностью.

Примеры: нефтехимические или органические химикаты являются очень огнеопасными, что в определенных условиях увеличивает риск пожара. Материалы с естественными или привнесенными радиоактивными свойствами представляют риск для тех, кто с ними обращается. Предметы, содержащие биоматериал больных людей, представляют повышенный риск инфекции для тех, кто обращается с ними. Вспыльчивые люди или животные могут представлять опасность для медицинского персонала.

Обоснование: некоторые сущности обладают свойствами, представляющими потенциально повышенную опасность нанесения вреда или повреждения другой сущности. В атрибуте riskCode передается тип риска без указания степени риска.

7.2.1.12 Entity.handlingCode:: SET<CE> (0..*)

Словарный домен: EntityHandling (CWE)

Определение: значение, указывающее специальные требования обработки сущности.

Примеры: хранить при комнатной температуре, хранить замороженным при температуре ниже 0° С, хранить в сухом месте; хранить вертикально, не переворачивая вверх дном.

Обоснование: этот атрибут используется для описания требования специальной обработки сущности во избежание ее повреждения или повреждения других сущностей.

Переходы состояний экземпляра класса Entity

Диаграмма перехода состояний класса Entity показана на рисунке 6.

Сущность может иметь следующие состояния:

- active (активное) — подсостояние состояния normal: сущность активна;
- inactive (не активное) — подсостояние состояния normal: сущность более не может быть активным участником событий;
- normal (нормальное): охватывает все ожидаемые состояния объекта услуги, за исключением nullified, которое представляет терминальные состояния экземпляра класса Entity, созданного по ошибке;
- nullified (аннулированное): терминальное состояние экземпляра класса Entity, созданного по ошибке.

Между состояниями действия возможны следующие переходы:

- revise (пересмотреть) — из состояния active в состояние active;
- inactivate (сделать неактивным) — из состояния active в состояние inactive;
- reactivate (активировать заново) — из состояния inactive в состояние active;

- revise (пересмотреть) — из состояния inactive в состояние inactive;
- cancel (отменить) — из состояния held в состояние cancelled;
- nullify (аннулировать) — из состояния normal в состояние nullified;
- create (создать) — из начального (пустого) состояния в состояние active.

7.2.2 Класс LanguageCommunication (в предметной области Entities)

Атрибуты класса LanguageCommunication:

- languageCode:: CE
- modeCode:: CE
- proficiencyLevelCode:: CE
- preferenceInd:: BL

Ассоциации класса LanguageCommunication:

- entity::(1..1) Entity:: languageCommunication:: (0..*) (ассоциация с классом Entity, роль languageCommunication — язык общения)

Определение класса LanguageCommunication: способности сущности к языковому общению.

Примеры: пациент, ранее приехавший из Мексики, может иметь беглые языковые возможности чтения и письма на испанском языке, но при этом имеет только самые начальные способности общения на английском языке. Выходец из России может одинаково хорошо владеть разговорным русским, армянским или украинским языком, но предпочитает говорить на армянском языке.

Обсуждение: хотя на первый взгляд использование этого класса должно быть ограничено только подтипами класса LivingSubject (живой организм), но устройства, представляемые классом Device, также могут использовать языковое общение, например, автоматизированные телефонные устройства, перенаправляющие пациента к живым операторам в зависимости от его нужд, или устройства, автоматически читающие врачам результаты лабораторных анализов.

Обоснование: каждая сущность, способная к вербальному общению, имеет свои собственные степени владения разными языками. В экземплярах класса LanguageCommunication можно указать языки, которыми владеет сущность, способ общения (речь, чтение, письмо), степень владения этим способом и предпочтительный язык общения.

В следующих подпунктах описаны атрибуты класса LanguageCommunication.

7.2.2.1 LanguageCommunication.languageCode:: CE (0..1)

Словарный домен: HumanLanguage (CWE)

Определение: значение, указывающее язык, которым сущность в определенной степени владеет для разговорного или письменного общения.

Примеры: испанский, итальянский, немецкий, английский, американский язык жестов и т.д.

Обсуждение: разговорное или письменное языковое общение присуще не только живым существам. Для устройств, которые взаимодействуют с людьми, используя человеческий язык, также надо указать, на какие языки они рассчитаны. Автоматизированные системы речевого ответа передают сообщения на естественном языке и взаимодействуют с другими устройствами или людьми, используя естественный язык.

Обоснование: многие люди и устройства способны общаться на разных языках, имея разные степени владения языками. Значение атрибута languageCode указывает декларируемую способность сущности к языковому общению.

7.2.2.2 LanguageCommunication.modeCode:: CE (0..1)

Словарный домен: LanguageAbilityMode (CWE)

Определение: значение, указывающее метод применения языка.

Примеры: речевое выражение, письменное выражение, выражение на языке жестов, восприятие речи, восприятие письменного языка, восприятие языка жестов.

7.2.2.3 LanguageCommunication.proficiencyLevelCode:: CE (0..1)

Словарный домен: LanguageAbilityProficiency (CWE)

Определение: значение, указывающее степень владения языком.

Примеры: превосходно, хорошо, удовлетворительно, плохо.

7.2.2.4 LanguageCommunication.preferenceInd:: BL (0..1)

Определение: индикатор, указывающий, является ли язык предпочтительным для метода применения, указанного в атрибуте modeCode.

7.2.3 Класс Container (в предметной области Entities)

Код класса: CONT.

Атрибуты класса Container:
- capacityQuantity:: PQ
- heightQuantity:: PQ
- diameterQuantity:: PQ
- capTypeCode:: CE
- separatorTypeCode:: CE
- barrierDeltaQuantity:: PQ
- bottomDeltaQuantity:: PQ:

Класс Container является специализацией класса ManufacturedMaterial.

Определение класса Container: класс Container является специализацией класса ManufacturedMaterial, предназначенной для описания контейнера, в который помещаются другие сущности в целях транспортировки или защиты их содержания от утери или повреждения.

Обоснование: спецификация этого класса появилась в результате сотрудничества комитета HL7 и организации NCCLS. Многие из определений атрибутов позаимствованы из стандарта NCCLS или ссылаются на него. Контейнер необходим для хранения аморфных субстанций (жидкости, газы). Однако содержание емкости всегда можно отличить от самого контейнера и относительно легко отделяется от него, в отличие от содержания (ингредиента) микстуры.

Применение: экземпляр класса Container связан с материалом содержания с помощью экземпляра класса Role, у которого атрибут classCode имеет значение CONT (content — содержание).

В следующих подпунктах описаны атрибуты класса Container.

7.2.3.1 Container.capacityQuantity:: PQ (0..1)

Определение: функциональная емкость контейнера.

7.2.3.2 Container.heightQuantity:: PQ (0..1)

Определение: высота контейнера.

7.2.3.3 Container.diameterQuantity:: PQ (0..1)

Определение: внешний диаметр контейнера.

7.2.3.4 Container.capTypeCode:: CE (0..1)

Словарный домен: ContainerCap (CWE)

Определение: тип крышки контейнера, с которой могут осуществляться такие автоматизированные манипуляции, как открытие, прокалывание и т.д.

7.2.3.5 Container.separatorTypeCode:: CE (0..1)

Словарный домен: ContainerSeparator (CWE)

Определение: материал, добавляемый в контейнер для обеспечения и облегчения физического разделения компонентов биоматериала различной плотности.

Примеры: в пробирки для сбора крови добавляется гель, который при последующем центрифугировании создает физический барьер между клетками крови и сывороткой или плазмой.

Обоснование: состав или тип разделяющего вещества может оказывать воздействие на выполняемый анализ. Знание разделяющего материала помогает интерпретировать результаты.

7.2.3.6 Container.barrierDeltaQuantity:: PQ (0..1)

Определение: расстояние от контрольной точки до разделяющего материала (барьера) внутри контейнера.

Обоснование: это расстояние может передаваться автоматизированной лабораторной системой измерительному инструменту и/или устройству обработки биоматериала, чтобы пробник, вставляемый в биоматериал, не касался разделителя. См. определение контрольной точки (Point of Reference) в стандарте NCCLS AUTO5 Laboratory Automation: Electromechanical Interfaces.

7.2.3.7 Container.bottomDeltaQuantity:: PQ (0..1)

Определение: расстояние от контрольной точки до внешнего дна контейнера.

Обоснование: см. определение контрольной точки (Point of Reference) в стандарте NCCLS AUTO5 Laboratory Automation: Electromechanical Interfaces.

7.2.4 Класс Device (в предметной области Entities)

Код класса: DEV.

Атрибуты класса Device:
- manufacturerModelName:: SC
- softwareName:: SC
- localRemoteControlStateCode:: CE

- alertLevelCode:: CE
- lastCalibrationTime:: TS

Класс Device является специализацией класса ManufacturedMaterial.

Определение класса Device: класс Device является специализацией класса ManufacturedMaterial, предназначенный для описания сущности, которая используется в деятельности, не претерпевая существенных изменений. Тип устройства указывается в атрибуте code, унаследованном от класса Entity.

Использование: понятие устройства включает в себя как долговечное медицинское оборудование (многократно используемое), так и одноразовое оборудование.

В следующих подпунктах описаны атрибуты класса Device.

7.2.4.1 Device.manufacturerModelName:: SC (0..1)

Словарный домен: ManufacturerModelName (CWE)

Определение: человеко-читаемое имя модели устройства, присвоенное производителем.

Примеры: масс-спектрометр на основе индуктивно связанной плазмы Perkin Elmer 400.

7.2.4.2 Device.softwareName:: SC (0..1)

Словарный домен: SoftwareName (CWE)

Определение: наименование, версия и выпуск программного обеспечения, управляющего устройством, присвоенные производителем или разработчиком программного обеспечения.

Примеры: Agilent Technologies Chemstation A.08.xx.

7.2.4.3 Device.localRemoteControlStateCode:: CE (0..1)

Словарный домен: LocalRemoteControlState (CWE)

Определение: значение, указывающее текущее состояние управления устройством.

Примеры: устройство может работать автономно (атрибут localRemoteControlStateCode имеет значение L (local — местный), или управляться другой системой (атрибут localRemoteControlStateCode имеет значение R (remote — дистанционный)).

Обоснование: состояние управления устройством может быть передано другому устройству до дистанционной передачи команд управления. Если состояние управления устройством не имеет значение R, то внешние команды будут игнорироваться.

7.2.4.4 Device.alertLevelCode:: CE (0..1)

Словарный домен: DeviceAlertLevel (CWE)

Определение: значение, указывающее текущее состояние функционирования автоматизированного устройства.

Примеры: нормальное, предупреждение, критическое

Ограничения: допустимые значения атрибута зависят от устройства.

7.2.4.5 Device.lastCalibrationTime:: TS (0..1)

Определение: дата/время последней калибровки устройства.

Обоснование: чтобы гарантировать, что устройства работают в соответствии со спецификациями, их надо с определенной периодичностью калибровать. Допустимый интервал между калибровками зависит от регламентов. Следовательно, для обеспечения правильности результатов точное время/дата последней калибровки надо рассматривать как критичный параметр.

7.2.5 Класс LivingSubject (в предметной области Entities)

Код класса: LIV.

Атрибуты класса LivingSubject:

- administrativeGenderCode:: CE
- birthTime:: TS
- deceasedInd:: BL
- deceasedTime:: TS
- multipleBirthInd:: BL
- multipleBirthOrderNumber:: INT
- organDonorInd:: BL

Класс LivingSubject является генерализацией классов NonPersonLivingSubject, Person.

Класс LivingSubject является специализацией класса Entity.

Определение класса LivingSubject: специализация класса Entity, описывающая организм или сложное животное, живое или мертвое.

Примеры: человек, собака, микроорганизм или растение любой таксономической группы.

Ограничения: экземпляры этого класса могут содержать информацию о млекопитающих, птицах, рыбах, бактериях, паразитах, грибах и вирусах. Человек является специализацией этого класса.

Обоснование: этот класс содержит «статические» или «административные» атрибуты, представляющие интерес для медицины и отличающие живые организмы от других сущностей.

В следующих подпунктах описаны атрибуты класса LivingSubject.

7.2.5.1 LivingSubject.administrativeGenderCode:: CE (0..1)

Словарный домен: AdministrativeGender (CWE)

Определение: значение, указывающее пол живого организма.

Примеры: женский, мужской

Обсуждение: значения этого атрибута не включают в себя термины, относящиеся к клиническому полу. Пол — это сложное физиологическое, генетическое и социологическое понятие, для всестороннего описания которого необходим ряд наблюдений. Цель этого атрибута — обеспечить высокоуровневую классификацию, которая дополнительно может использоваться для соответствующего размещения пациентов на больничные койки.

Ограничение: этот код используется в административных целях.

7.2.5.2 LivingSubject.birthTime:: TS (0..1)

Определение: дата и время рождения или выведения живого организма.

7.2.5.3 LivingSubject.deceasedInd:: BL (0..1)

Определение: признак смерти организма.

7.2.5.4 LivingSubject.deceasedTime:: TS (0..1)

Определение: дата и время смерти живого организма.

7.2.5.5 LivingSubject.multipleBirthInd:: BL (0..1)

Определение: признак, появился ли живой организм в результате многоплодных родов.

7.2.5.6 LivingSubject.multipleBirthOrderNumber:: INT (0..1)

Определение: порядок рождения живого организма при многоплодных родах.

7.2.5.7 LivingSubject.organDonorInd:: BL (0..1)

Определение: признак того, что живой организм является кандидатом в доноры органа.

Обсуждение: этот атрибут указывает, пожертвовал ли данный живой организм свой орган или готов его пожертвовать.

7.2.6 Класс ManufacturedMaterial (в предметной области Entities)

Код класса: MMAT.

Атрибуты класса ManufacturedMaterial:

- lotNumberText:: ST
- expirationTime:: IVL<TS>
- stabilityTime:: IVL<TS>

Класс ManufacturedMaterial является генерализацией классов Container, Device.

Класс ManufacturedMaterial является специализацией класса Material.

Определение класса ManufacturedMaterial: специализация класса Material, описывающая сущность или комбинацию сущностей, преобразованную в специальных целях с помощью искусственного или производственного процесса.

Примеры: обработанные пищевые продукты, одноразовые шприцы, химический анализатор, физиологический раствор и т.д.

Обсуждение: к понятиям этого класса относятся контейнеры, устройства, программные модули и средства.

Обоснование: этот класс используется для дальнейшего определения характеристик сущностей, которые созданы из других сущностей. Эти сущности идентифицируются и отслеживаются с помощью ассоциаций и механизмов, уникальных для данного класса, например, lotName, stabilityTime и expirationTime.

В следующих подпунктах описаны атрибуты класса ManufacturedMaterial.

7.2.6.1 ManufacturedMaterial.lotNumberText:: ST (0..1)

Определение: алфавитно-цифровая строка, используемая для идентификации конкретной партии изготовленного материала.

Обсуждение: номер партии обычно печатается на ярлыке, приложенном к контейнеру, содержащему субстанцию, и/или на упаковке, в которую вложен контейнер. Следует отметить, что номер партии не обязан быть уникальным идентификатором, он имеет значение только в сочетании с видом продукта и производителем.

7.2.6.2 ManufacturedMaterial.expirationTime:: IVL<TS> (0..1)

Определение: дата и время, после которого производитель больше не гарантирует безопасность, качество, и/или надлежащее функционирование материала.

Обоснование: во многих ситуациях требуется, чтобы используемые материалы имели конкретное качество или обладали определенной эффективностью, либо имели необходимое функциональное состояние. Конечная дата этой гарантии определяется производителем. Хотя после указанной даты материал может все еще обладать теми же самыми свойствами, производитель с этого момента снимает с себя ответственность за надлежащее функционирование материала и возможные последствия его использования.

7.2.6.3 ManufacturedMaterial.stabilityTime:: IVL<TS> (0..1)

Определение: время, в течение которого материал считается пригодным к использованию после активирования.

Примеры: время годности жидкости после открытия флакона; смесь двух химических реагентов для анализа, которая должна быть использована в течение двух часов после смешивания, иначе ее активность снижается.

Обсуждение: если экземпляр класса ManufacturedMaterial описывает вид материала (его атрибут determinerCode имеет значение KIND), то может быть известна только длина интервала, например, время после открытия контейнера с реагентом, в течение которой вещество реагента считается пригодным для употребления в его стандартном аналитическом применении. Для определенного экземпляра реагента (например, конкретный флакон), значение stabilityTime.low типа TS указывает время, когда флакон был открыт (или реагент был активирован иным образом). Добавляя к нему типичное время стабильности, можно получить значение stabilityTime.high типа TS, после которого реагент уже не считается пригодным для употребления в его стандартном аналитическом применении.

7.2.7 Класс Material (в предметной области Entities)

Код класса: MAT.

Атрибуты класса Material:

- formCode:: CE

Класс Material является генерализацией класса ManufacturedMaterial.

Класс Material является специализацией класса Entity.

Определение класса Material: специализация класса Entity, описывающая неодушевленную сущность, не зависящую от местонахождения.

Примеры: фармацевтические субстанции (включая активные вакцины, содержащие ослабленные вирусные клетки), одноразовые предметы, оборудование длительного пользования, имплантаты, продукты питания (включая мясные или растительные продукты), отходы, продаваемые товары и т.д.

Обсуждение: изготовленные или обработанные продукты считаются материалом, даже если они происходят от живой материи. Существует широкое разнообразие физических форм материалов, которые могут принимать различные состояния (например, газ, жидкость, твердое вещество), все еще сохраняя свой физический состав и материальные характеристики.

Обоснование: есть сущности, которые обладают дополнительными атрибутами по сравнению с классом Entity, но не могут быть классифицированы как живой организм или место.

В следующих подпунктах описаны атрибуты класса Material.

7.2.7.1 Material.formCode:: CE (0..1)

Словарный домен: MaterialForm (CWE)

Определение: значение, указывающее состояние (твердое, жидкое, газообразное) и природу материала.

Примеры: для терапевтических субстанций могут быть указаны такие формы, как таблетка, мазь, гель и т. д.

7.2.8 Класс NonPersonLivingSubject (в предметной области Entities)

Код класса: NLIV.

Атрибуты класса NonPersonLivingSubject:

- strainText:: ED
- genderStatusCode:: CE

Класс NonPersonLivingSubject является специализацией класса LivingSubject.

Определение класса NonPersonLivingSubject: специализация класса LivingSubject, описывающая все живые организмы, кроме человека.

Примеры: крупный рогатый скот, птицы, бактерии, формы растений и грибы и т. д.

Обсуждение: для описания живого организма, не являющегося человеком, может понадобиться дополнительная характерная информация, например, генетическая идентификация рода, которая не может быть передана в атрибуте code.

В следующих подпунктах описаны атрибуты класса NonPersonLivingSubject.

7.2.8.1 NonPersonLivingSubject.strainText:: ED (0..1)

Определение: текстовая строка, указывающая определенный генотипический или фенотипический вариант организма, информация о котором передается в экземпляре класса NonPersonLivingSubject.

Примеры: Миннесота 5 (порода поросят), DXL (порода домашней птицы), RB51 (штамм противобруцеллезной вакцины).

Обоснование: универсальное руководство по именованию или каталогизированию породы, вида или штамма организмов отсутствует. Многие обозначения пород, видов и штаммов создавались, а затем исчезали, а другие по разным причинам приживались в различных областях производства (изготовление вакцин, популярность племенного животноводства, и т. д.). И поскольку каждый, кому это нужно, может обозначить организм как «новый» вид, вместо кодированного значения данный атрибут должен содержать описательный текст, в котором можно передать все такие обозначения.

7.2.8.2 NonPersonLivingSubject.genderStatusCode:: CE (0..1)

Словарный домен: GenderStatus (CWE)

Определение: значение, указывающее, имеются ли основные репродуктивные органы у организма, информация о котором передается в экземпляре класса NonPersonLivingSubject.

7.2.9 Класс Organization (в предметной области Entities)

Код класса: ORG.

Атрибуты класса Organization:

- addr:: BAG<AD>
- standardIndustryClassCode:: CE

Класс Organization является специализацией класса Entity.

Определение класса Organization: специализация класса Entity, описывающая формализованную группу сущностей с общей целью (например, административной, юридической, политической) и инфраструктуру, необходимую для осуществления этой цели.

Примеры: предприятия и институты, правительственная структура, корпорация по управлению недвижимостью, страховая компания.

В следующих подпунктах описаны атрибуты класса Organization.

7.2.9.1 Organization.addr:: BAG<AD> (0..*)

Определение: почтовый и/или юридический адрес организации.

7.2.9.2 Organization.standardIndustryClassCode:: CE (0..1)

Словарный домен: OrganizationIndustryClass (CWE)

Определение: значение, указывающее код организации согласно классификатору отраслей народного хозяйства.

Примеры: 11231 — производство куриных яиц, 6211 — амбулаторная помощь, 621511 — медицинские лаборатории, 524114 — непосредственное страхование здоровья и медицинской помощи.

7.2.10 Класс Person (в предметной области Entities)

Код класса: PSN.

Атрибуты класса Person:

- addr:: BAG<AD>
- maritalStatusCode:: CE
- educationLevelCode:: CE
- disabilityCode:: SET<CE>
- livingArrangementCode:: CE
- raceCode:: SET<CE>
- ethnicGroupCode:: SET<CE>

Класс Person является специализацией класса LivingSubject.

Определение класса Person: специализация класса LivingSubject, описывающая человека.

Ограничения: в зависимости от значений атрибутов determinerCode и quantity экземпляр этого класса может содержать информацию об одном лице или о группе лиц.

В следующих подпунктах описаны атрибуты класса Person.

7.2.10.1 Person.addr:: BAG<AD> (0..*)

Определение: почтовый адрес или адрес места жительства лица.

7.2.10.2 Person.maritalStatusCode:: CE (0..1)

Словарный домен: MaritalStatus (CWE)

Определение: значение, указывающее статус семейного партнерства лица.

Примеры: женат (замужем), раздельное проживание, в разводе, вдовец (вдова), гражданский брак.

Обоснование: эта информация передается в поле FL 16 универсального счета на оплату лечения (UB).

7.2.10.3 Person.educationLevelCode:: CE (0..1)

Словарный домен: EducationLevel (CWE)

Определение: самый высокий уровень образования, достигнутый лицом (например, начальная школа, средняя школа или полная степень средней школы, колледж или полная ученая степень бакалавра).

7.2.10.4 Person.disabilityCode:: SET<CE> (0..*)

Словарный домен: PersonDisabilityType (CWE)

Определение: значение, указывающее инвалидность человека.

Примеры: ослабленное зрение, ослабленный слух.

7.2.10.5 Person.livingArrangementCode:: CE (0..1)

Словарный домен: LivingArrangement (CWE)

Определение: значение, указывающее условия проживания лица.

Примеры: собственный дом, организация, интернат, реабилитационный центр, дом престарелых и т. д.

Обсуждение: используется при планировании выписки, оценке необходимости социального обеспечения, психосоциологической оценке.

7.2.10.6 Person.religiousAffiliationCode:: CE (0..1)

Словарный домен: ReligiousAffiliation (CWE)

Определение: основное религиозное предпочтение лица (например, индуизм, ислам, римско-католическая церковь).

7.2.10.7 Person.raceCode:: SET<CE> (0..*)

Словарный домен: Race (CWE)

Определение: значение, указывающее расу лица.

7.2.10.8 Person.ethnicGroupCode:: SET<CE> (0..*)

Словарный домен: Ethnicity (CWE)

Определение: этническая группа лица.

7.2.11 Класс Place (в предметной области Entities)

Код класса: PLC.

Атрибуты класса Place:

- mobileInd:: BL
- addr:: BAG<AD>
- directionsText:: ED
- positionText:: ED
- gpsText:: ST

Класс Place является специализацией класса Entity.

Определение класса Place: специализация класса Entity, описывающая ограниченное физическое место или участок с содержащимися в нем структурами, если таковые имеются.

Примеры: поле, озеро, город, графство, штат, страна, участок (земли), здание, трубопровод, линия электропередачи, детская площадка, судно, грузовик.

Ограничения: место может быть естественным или созданным человеком. Географическое положение места может быть непостоянным.

Обсуждение: места могут быть производственными зданиями (где выполняются релевантные действия), домами (где живут люди) или конторами (где люди работают). Места могут содержать другие места (этаж, палата, кабина, койка). Экземпляры класса Place могут содержать информацию о местах, идентифицируемых в контексте оказания медицинской помощи, социальной помощи, санитарно-эпидемиологических мероприятий (например, здания, поляны для пикника, дневные стационары, тюрьмы, округа, штаты, и другие места эпидемиологических событий).

В следующих подпунктах описаны атрибуты класса Place.

7.2.11.1 Place.mobileInd:: BL (0..1)

Определение: признак того, может ли место свободно перемещаться из одного местоположения в другое.

Примеры: перемещаемыми местами оказания медицинской помощи могут быть суда, самолеты и санитарные машины.

7.2.11.2 Place.addr:: AD (0..1)

Определение: физический адрес данного места.

Ограничения: значением этого атрибута должен быть адрес, с помощью которого можно определить физическое положение места на карте.

7.2.11.3 Place.directionsText:: ED (0..1)

Определение: свободный текст, содержащий информацию о данном месте, которая полезна для сущностей при поиске этого места.

Обсуждение: в этом атрибуте могут передаваться указания для поиска места, если информация об адресе не адекватна, его координаты в глобальной системе навигации (GPS) недоступны и/или сущность, которая ищет это место, не может непосредственно использовать координаты GPS. В нем можно передавать информацию, полезную для посетителей места. Например, «последний дом справа», «если владельца нет, узнайте в следующем доме, где его найти».

7.2.11.4 Place.positionText:: ED (0..1)

Определение: совокупность кодов, задающих расположение места на карте.

Примеры: координаты на картах, издаваемых агентством US Geological Survey.

7.2.11.5 Place.gpsText:: ST (0..1)

Определение: координаты места в глобальной системе навигации (GPS).

Обсуждение: значения координат GPS должны соответствовать стандартам USGS Spatial Data Transmission. К числу этих параметров относятся способы получения значений широты и долготы, ошибки сдвига, проекция.

Обоснование: в некоторых полевых условиях физический адрес интересующего места может отсутствовать. Поскольку все места поверхности земли имеют уникальные географические координаты, для точного определения и передачи информации о положении места можно использовать GPS-координаты.

7.3 Классы предметной области Roles

7.3.1 Класс Role (в предметной области Roles)

Код класса: ROL.

Атрибуты класса Role:

- classCode:: CS
- id:: SET<II>
- code:: CE
- negationInd:: BL
- name:: BAG<EN>
- addr:: BAG<AD>
- telecom:: BAG<TEL>
- statusCode:: CS
- effectiveTime:: IVL<TS>
- certificateText:: ED
- quantity:: RTO
- positionNumber:: LIST<INT>

Ассоциации класса Role:

player::(0..1) Entity::playedRole::(0..*) (ассоциация с классом Entity, роль playedRole — выполняемая роль)

scoper::(0..1) Entity::scopedRole::(0..*) (ассоциация с классом Entity, роль scopedRole — контролируемая роль)

participation::(0..*) Participation::role::(1..1) (ассоциация с классом Participation, роль role — роль)

outboundLink::(0..*) RoleLink::source::(1..1) (ассоциация с классом RoleLink, роль source — источник)

inboundLink::(0..*) RoleLink::target::(1..1) (ассоциация с классом RoleLink, роль target — цель)

Класс Role является обобщением следующих классов:

- Access
- Employee
- LicensedEntity
- Patient

Определение класса Role: компетентность сущности, выполняющей роль в соответствии с указаниями, определениями, гарантиями или признанием другой сущности, контролирующей эту роль.

Обсуждение: сущность участвует в действии, выполняя определенную роль. Следует отметить, что конкретная сущность в конкретной роли может участвовать в действии многими способами. Например, лицо в роли врача может участвовать в ведении пациента как дежурант или как лечащий врач. Роль определяет компетентность сущности, не зависящую от любого действия, в противоположность участию, которое ограничено рамками действия.

Каждая роль «выполняется» одной сущностью, называемой «исполнителем», и «контролируется» другой сущностью, называемой «контролером». Например, роль «пациент» может выполняться некоторым лицом и контролироваться организацией, оказывающей пациенту медицинскую помощь. Аналогичным образом работодатель контролирует роль «работника».

Идентификатор роли идентифицирует сущность, выполняющую эту роль. Этот идентификатор присваивается исполнителю контролирующей сущностью. Контролирующая сущность не обязана создавать новые идентификаторы. Она может повторно использовать идентификатор, ранее присвоенный контролируемой сущности, чтобы идентифицировать эту сущность в контролируемой ею роли.

Большинство атрибутов класса Role представляют собой характеристики исполнителя роли, присущие ему в процессе выполнения конкретной роли.

В следующих подпунктах описаны атрибуты класса Role.

7.3.1.1 Role.classCode:: CS (1..1) Mandatory

Словарный домен: RoleClass (CNE)

Определение: код, указывающий основную категорию роли в соответствии со словарем, определенным в стандарте HL7.

7.3.1.2 Role.id:: SET<II> (0..*)

Определение: уникальный идентификатор сущности, выполняющей данную роль.

7.3.1.3 Role.code:: CE (0..1)

Словарный домен: RoleCode (CWE)

Определение: код, уточняющий вид роли.

Обсуждение: значения атрибута Role.code должны концептуально представлять собой специализации понятия, указанного в атрибуте Role.classCode. Атрибут Role.code не должен использоваться как модификатор значения атрибута Role.classCode. Каждый из этих атрибутов представляет завершенное понятие или ролевые отношения между двумя сущностями, но значение атрибута Role.code может быть более специфичным, нежели значение атрибута Role.classCode.

Значение Role.code может быть не кодированным, если данный тип роли обычно обозначается только не кодированным именем.

7.3.1.4 Role.negationInd:: BL (0..1)

Определение: индикатор, указывающий, что данный экземпляр класса Role описывает компетенцию, которая отсутствует у сущности, выполняющей эту роль.

Примеры:

1) Это лицо **не** является нашим работником.

2) Эта жидкость для полоскания рта **не** содержит алкоголь в качестве ингредиента.

Ограничение: обычно все роли рассматриваются как утвердительные (по умолчанию этот атрибут имеет значение FALSE).

7.3.1.5 Role.addr:: BAG<AD> (0..*)

Определение: адрес сущности во время выполнения роли.

7.3.1.6 Role.telecom:: BAG<TEL> (0..*)

Определение: телекоммуникационный адрес сущности во время выполнения роли.

7.3.1.7 Role.statusCode:: CS (0..1)

Словарный домен: RoleStatus (CNE)

Определение: код, указывающий состояние данной роли, определенное в модели перехода состояний.

7.3.1.8 Role.effectiveTime:: IVL<TS> (0..1)

Определение: интервал времени, указывающий время выполнения роли, если такое время применимо и известно.

7.3.1.9 Role.certificateText:: ED (0..*)

Определение: текстовое или мультимедийное представление сертификата, изданного сущностью, контролирующей данную роль, и удостоверяющего, что данная роль действительно принадлежит исполнителю.

Примеры: сертификат может быть представлен многими разными способами либо по значению, либо по ссылке в соответствии с типом данных ED. Типичные случаи таковы:

1) Бумажный сертификат: значение, имеющее тип данных ED, может содержать ссылку на некоторый документ или файл, который может быть найден с помощью электронного интерфейса к архиву бумажных документов.

2) Электронный сертификат: в этом атрибуте можно передать практически любую электронную схему сертификата, например, электронный текстовый документ с электронной подписью (в том числе стандартной).

3) Цифровой сертификат (сертификат открытого ключа): в частности, в этом атрибуте можно передавать цифровые сертификаты по значению или по ссылке. Блок данных сертификата может конструироваться в соответствии со стандартом цифровых сертификатов, например, X.509, SPKI, PGP и т.д.

Субъектом сертификата является сущность, выполняющая данную роль. Издателем сертификата является сущность, контролирующая данную роль.

7.3.1.10 Role.quantity:: RTO (0..1)

Определение: отношение (числитель:знаменатель), указывающее относительное количество исполнителя роли в сущности, контролирующей роль. Оно используется для ролей, описывающих отношения композиции между контролирующими сущностями и исполнителями.

Примеры:

1) Ингредиентом этого сиропа (контролирующая сущность) являются 160 мг (числитель) ацетаминофена (исполнитель) на столовую ложку (знаменатель).

2) Стадо (контролирующая сущность) состоит из 500 (числитель) голов крупного рогатого скота (исполнитель).

3) У компьютера VAX 6630 (контролирующая сущность) имеются 3 (числитель) центральных процессора (исполнитель).

4) Эта упаковка (контролирующая сущность) содержит 100 (числитель) таблеток (исполнитель).

Обсуждение: в отношениях композиции (например, имеет части, имеет ингредиенты, имеет содержание) значение атрибута Role.quantity указывает, что количество числителя целевой сущности отношения содержит в себе количество сущности-источника, указанное в знаменателе. Например, если коробка (источник отношения) содержит 10 яиц (цель отношения), то атрибут quantity должен иметь значение 10:1; если 0,6 мл смеси содержат 75 мг $FeSO_4$, то атрибут quantity должен иметь значение 75мг:0,6 мл. Как числитель, так и знаменатель должны быть количественными величинами (экстенсивными количествами, например, число подсчетов, масса, объем, количество субстанции, количество энергии и т.д.).

7.3.1.11 Role.positionNumber:: LIST<INT> (0..*)

Определение: целое число, указывающее положение исполнителя роли относительно сущности, контролирующей эту роль.

Обсуждение: этот атрибут, в основном, используется в экземплярах класса Role, описывающих содержание в чем-либо. Например, у некоторых контейнеров есть дискретные положения, в которых может быть размещено содержимое. В зависимости от геометрии контейнера для указания конкретного положения может использоваться скалярное порядковое число или вектор порядковых чисел (координаты). Начальным значением координаты должно быть число 1.

Для некоторых контейнеров может использоваться собственный способ нумерации положений; способа нумерации может и не быть. В отсутствие какого-либо определенного способа позиционирования у конкретного типа контейнеров эмпирическое правило нумерации состоит в том, что координата, изменяющаяся раньше, должна позиционироваться первой. Для автоматизированного биохимического анализатора крови со штативом квадратной формы это означает, что первой координатой будет та, в каком направлении перемещается штатив на каждом шаге, а второй — та, в каком направлении штатив перемещается через каждые 10 (или иное число) шагов.

Переходы состояний экземпляра класса Role

Диаграмма перехода состояний класса Role показана на рисунке 5.3. Роль может иметь следующие состояния:

- active (активно) — подсостояние состояния normal: сущность в настоящий момент активна в данной роли;

- normal (нормальное): «типичное» состояние. Исключает состояние nullified, которое указывает, что экземпляр класса Role был создан по ошибке;

- nullified (аннулировано): это состояние является терминальным состоянием экземпляра класса Role, созданного по ошибке;

- suspended (приостановлено) — подсостояние состояния normal: выполнение роли временно приостановлено. Переход в это состояние возможен из состояния active (активно);
- terminated (завершено) — подсостояние состояния normal: успешное завершение выполнения роли.

Между состояниями действия возможны следующие переходы:
- revise (пересмотреть) — из состояния active в состояние active;
- suspend (приостановить) — из состояния active в состояние suspended;
- terminate (завершить) — из состояния active в состояние terminated;
- nullify (аннулировать) — из состояния normal в состояние nullified;
- create (создать) — из начального (пустого) состояния в состояние active;
- resume (продолжить) — из состояния suspended в состояние active;
- revise (пересмотреть) — из состояния suspended в состояние suspended;
- terminate (завершить) — из состояния suspended в состояние terminated;
- reactivate (активизировать заново) — из состояния terminated в состояние active;
- revise (пересмотреть) — из состояния terminated в состояние terminated.

7.3.2 Класс RoleLink (в предметной области Roles)

Атрибуты класса RoleLink:
- typeCode:: CS
- effectiveTime:: IVL<TS>

Ассоциации класса RoleLink:

target::(1..1) Role::inboundLink::(0..*) (ассоциация с классом Role, роль inboundLink — входящая связь)

source::(1..1) Role::outboundLink::(0..*) (ассоциация с классом Role, роль outboundLink — исходящая связь)

Определение класса RoleLink: связь двух экземпляров класса Role, выражающая зависимость соответствующих ролей.

Примеры:

1) Роль должности или агента зависит от другой роли, а именно, роли работника. Если роль работника завершается, то прекращается и роль должности. Тем самым определяется зависимость роли должности от роли работника, или, другими словами, должность «является частью» роли работника.

2) Одна из ролей может иметь полномочия управления другой ролью (в иерархии подчинения). Например, работник категории «управляющий» может иметь полномочия управления служащими категории «аналитик», что может быть указано с помощью экземпляра класса RoleLink, у которого атрибут typeCode имеет значение DIRAUTH (has direct authority over — имеет прямые полномочия управления).

Обсуждение: экземпляр класса RoleLink описывает отношения между ролями, а не между людьми (или другими сущностями). Люди (или другие сущности), в основном, связаны между собой прямыми отношениями исполнитель/контролер, возникающими у роли исполнителя, а в более общем плане — отношениями взаимодействия (т. е. их участием в действиях).

В следующих подпунктах описаны атрибуты класса RoleLink.

7.3.2.1 RoleLink.typeCode:: CS (1..1) Mandatory

Словарный домен: RoleLinkType (CNE)

Определение: код, указывающий вид связи, представленной данным экземпляром класса RoleLink, например, PART (has part — имеет частью), DIRAUTH (has direct authority over — имеет прямые полномочия управления).

7.3.2.2 RoleLink.effectiveTime:: IVL<TS> (0..1)

Определение: временной интервал, указывающий период действия связи между ролями.

7.3.3 Класс Access (в предметной области Roles)

Атрибуты класса Access:
- approachSiteCode:: CD
- targetSiteCode:: CD
- gaugeQuantity:: PQ

Класс Access является специализацией класса Role.

Определение класса Access: специализация класса Role, описывающая роль устройства, используемого для введения терапевтических агентов (лекарств и элементов, поддерживающих жизнь) в тело пациента, или устройства, используемого для вывода веществ (например, экссудатов, гноя, мочи, воздуха, крови) из его тела.

Обсуждение: В общем случае доступ описывает роль изготовленного материала или устройств, иногда специально изготовленного или созданного для определенной цели, например, роль катетера или канюли, помещенной в полый орган тела.

Устройства, выполняющие роль доступа, обычно используются в наблюдениях поглощения/оттока, а также для введения лекарственных средств. Микробиологические исследования самого материала или жидкостей, выделяемых из дренажа, также являются обычными примерами.

Обоснование: роль доступа, в основном, используется для описания материала, фактически применяемого для доступа, а не нового материала, полученного от изготовителя. Например, для представления заказа коробки катетеров у поставщика не обязательно использовать класс роли Access, так как атрибутов класса Material обычно вполне достаточно для описания и идентификации заказываемого изделия. А экземпляры класса Access используются для передачи данных об эксплуатации, поглощении/оттоке и своевременной замене трубок и дренажа.

В следующих подпунктах описаны атрибуты класса Access.

7.3.3.1 Access.approachSiteCode:: CD (0..1)

Словарный домен: ActSite (CWE)

Определение: кодированное представление анатомической локализации места, через которое устройство, участвующее в доступе, впервые проникает в тело и, если применимо, направление от места первого входа к целевому месту.

Примеры: катетер легочной артерии предназначен для легочной артерии, но при этом местом доступа обычно является внутренняя яремная вена в шее или подключичная вена в подключичной ямке.

Ограничения: система кодирования значений атрибута Access.approachSiteCode та же, что у атрибута Procedure.approachSiteCode; действительно, атрибут approachSiteCode был скопирован из класса Procedure в класс Access. Значение атрибута Access.approachSiteCode должно быть идентично значению атрибута Procedure.approachSiteCode соответствующей процедуры введения устройства.

Обоснование: поскольку устройства доступа обычно помещаются в тело на значительный период времени и используются как ресурс для многих действий, то целевое место доступа становится важным идентифицирующим атрибутом самого доступа (в противоположность тому, когда оно является атрибутом процедуры введения).

7.3.3.2 Access.targetSiteCode:: CD (0..1)

Словарный домен: ActSite (CWE)

Определение: кодированное указание анатомической локализации места или части тела, куда обеспечивается доступ, т. е. часть, в которую вводится материал, или часть, из которой он выводится.

Примеры: катетер легочной артерии будет иметь целевым местом «легочную артерию».

Ограничения: система кодирования значений атрибута Access.targetSiteCode та же, что у атрибута Procedure.targetSiteCode; действительно, атрибут targetSiteCode был скопирован из класса Procedure в класс Access. Значение атрибута Access.targetSiteCode должно быть идентично значению атрибута Procedure.targetSiteCode соответствующей процедуры введения устройства.

Обоснование: поскольку устройства доступа обычно помещаются в тело на значительный период времени и используются как ресурс для многих действий, то целевое место доступа становится важным идентифицирующим атрибутом самого доступа (в противоположность тому, когда оно является атрибутом процедуры введения). Целевое место является важной информацией, по которой можно установить вид субстанций, которые могут или не могут быть применены в этом месте (например, должны быть предприняты специальные меры во избежание инъекции лекарства в устройство доступа, находящееся в артерии).

7.3.3.3 Access.gaugeQuantity:: PQ (0..1)

Определение: мера внутреннего диаметра устройства доступа (например, просвет трубки).

7.3.4 Класс Employee (в предметной области Roles)

Атрибуты класса Employee:

- jobCode:: CE
- jobTitleName:: SC
- jobClassCode:: CE
- salaryTypeCode:: CE
- salaryQuantity:: MO
- hazardExposureText:: ED
- protectiveEquipmentText:: ED

Класс Employee является специализацией класса Role.

Определение класса Employee: специализация класса Role, описывающая роль лица, связанного с организацией (работодателем в роли контролера) для получения заработной платы рабочего или служащего.

Обсуждение: класс Employee предназначен для описания трудовых отношений между работником и работодателем, а не для описания характера фактически выполненной работы (в противоположность классу AssignedEntity).

В следующих подпунктах описаны атрибуты класса Employee.

7.3.4.1 Employee.jobCode:: CE (0..1)

Словарный домен: EmployeeJob (CWE)

Определение: код, указывающий вид работы, выполняемой работником для работодателя. Например, бухгалтер, аналитик-программист, сиделка, штатная медсестра и т. д.

7.3.4.2 Employee.jobTitleName:: SC (0..1)

Словарный домен: JobTitleName (CWE)

Определение: название занимаемой должности, например, вице-президент, старший технический аналитик.

7.3.4.3 Employee.jobClassCode:: CE (0..1)

Словарный домен: EmployeeJobClass (CWE)

Определение: код, указывающий разные варианты занятости, например, полная ставка или совместитель.

7.3.4.4 Employee.salaryTypeCode:: CE (0..1)

Словарный домен: EmployeeSalaryType (CWE)

Определение: код, указывающий метод начисления заработной платы рабочему или служащему. Например, почасовая ставка, годовая ставка, комиссионные.

7.3.4.5 Employee.salaryQuantity:: MO (0..1)

Определение: сумма заработной платы рабочего или служащего, начисляемая в соответствии с методом, указанным в значении атрибута salaryTypeCode. Например, если атрибут salaryTypeCode имеет значение «почасовая», то в атрибуте salaryQuantity передается сумма почасовой ставки.

7.3.4.6 Employee.hazardExposureText:: ED (0..1)

Определение: тип вредности выполняемой работы. Например, асбест, инфекционные агенты.

7.3.4.7 Employee.protectiveEquipmentText:: ED (0..1)

Определение: средства защиты, необходимые для выполнения работы. Например, защитные очки, каска.

7.3.5 Класс LicensedEntity (в предметной области Roles)

Код класса: LIC.

Атрибуты класса Employee:

- recertificationTime:: TS

Класс LicensedEntity является специализацией класса Role.

Определение класса LicensedEntity: специализация класса Role, описывающая роль сущности (исполнителя), обладающей лицензией или квалификацией (дипломом), подтверждающей право данной сущности выполнять специфичные функции.

Примеры:

1) диплом фельдшера скорой помощи;

2) сертификат безопасности оборудования;

3) сертификат специалиста или лицензия на право осуществления медицинской деятельности.

Ограничения: контролером является организация, выпускающая диплом, сертификат или лицензию.

В следующих подпунктах описаны атрибуты класса LicensedEntity.

7.3.5.1 LicensedEntity.recertificationTime:: TS (0..1)

Определение: требуемая дата повторной сертификации.

7.3.6 Класс Patient (в предметной области Roles)

Код класса: PAT.

Атрибуты класса Patient:

- confidentialityCode:: CE
- veryImportantPersonCode:: CE

Класс Patient является специализацией класса Role.

Определение класса Patient: специализация класса Role, описывающая роль живого существа (исполнителя роли), получающего медицинскую помощь от ее поставщика (контролера роли).

В следующих подпунктах описаны атрибуты класса Patient.

7.3.6.1 Patient.confidentialityCode:: CE (0..1)

Словарный домен: Confidentiality (CWE)

Определение: код, указывающий политику защиты персональных данных, применяемую для данного пациента.

7.3.6.2 Patient.veryImportantPersonCode:: CE (0..1)

Словарный домен: PatientImportance (CWE)

Определение: код, указывающий особый статус пациента, предоставленный ему контролирующей организацией, часто обеспечивающий приоритетное лечение и особые условия. Например, член правления, дипломат и т. д.

8 Ассоциации в ЭИМ, выпуск 1

8.1 (0..*) ActRelationship :: source :: (1..1) Act :: outboundRelationship
8.2 (0..*) ActRelationship :: target :: (1..1) Act :: inboundRelationship
8.3 (1..1) Entity :: languageCommunication :: (0..*) LanguageCommunication :: entity
8.4 (0..*) Participation :: act :: (1..1) Act :: participation
8.5 (0..*) Participation :: role :: (1..1) Role :: participation
8.6 (0..*) Role :: player :: (0..1) Entity :: playedRole
К этой ассоциации применяется следующее ограничение:
Invariant (Role x) { not(x.player.equals(null)) or not(x.scoper.equals(null)) }
8.7 (0..*) Role :: scoper :: (0..1) Entity :: scopedRole
К этой ассоциации применяется следующее ограничение:
Invariant (Role x) { not(x.player.equals(null)) or not(x.scoper.equals(null)) }
8.8 (0..*) RoleLink :: source :: (1..1) Role :: outboundLink
8.9 (0..*) RoleLink :: target :: (1..1) Role :: inboundLink

9 Содержание нормативного словаря

9.1 Структурированный словарь ЭИМ

Таблицы словаря, определенные в стандарте HL7, содержат термины, управляющие структурными атрибутами ЭИМ. В этих таблицах определен полный набор специализаций классов Act (действие), Role (роль) и Entity (сущность), а также типы ассоциаций, представленных классами ActRelationship (связь действий), Participation (участие) и RoleLink (связь ролей). В других таблицах приведены значения кодов состояний и других контролируемых понятий, используемых в нормативных классах ЭИМ.

Представленные далее таблицы организованы в алфавитном порядке по именам доменов таблицы или именам доменов. Для каждого кодированного значения в них указан мнемокод, отображаемое имя и определение/описание. (Абстрактным доменам код не присваивается.) В них используются следующие правила представления информации:

- первый столбец содержит целое число, указывающее уровень элемента, описанного в этой строке, в иерархии элементов;

- второй столбец содержит составной элемент, характеризующий иерархию понятий. Первый символ этого элемента указывает, является ли понятие специализируемым (S), т.е. имеет собственный код и содержит дочерние понятия, или абстрактным (A), которое не имеет собственного кода, но может содержать дочерние понятия, или листом (Leaf), имеющим код, но не содержащим дочерних понятий. Имена понятий абстрактного и специализируемого доменов следуют за первым символом составного элемента.

- третий столбец содержит мнемокод неабстрактного понятия.

- четвертый столбец содержит отображаемое имя неабстрактного понятия.

- последний столбец содержит определение понятия.

Строки упорядочены в алфавитной последовательности доменных имен для каждого уровня вложенности, после которых следуют листовые понятия (упорядоченные по кодам) на том же самом уровне.

Эта спецификация содержит информацию для следующих словарных доменов:

ActClass	ActStatus	ParticipationType
ActMood	ContextControl	RelationshipConjunction
ActRelationshipCheckpoint	EntityClass	RoleClass
ActRelationshipJoin	EntityDeterminer	RoleLinkType
ActRelationshipSplit	EntityStatus	RoleStatus
ActRelationshipType	ManagedParticipationStatus	

Таблица 1 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 1 — Домен ActClass (используется в атрибуте Act.classCode)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
	S: ActClassRoot	ACT	act (действие)	(См. также определение класса Act ЭИМ). Действие, которое произошло, может произойти, происходит, планируется или потребовано. Это понятие описывает намеренное действие в предметной области стандарта HL7. Здравоохранение (и любая профессия или бизнес) состоит из намеренных действий. Экземпляр класса Act представляет собой информацию о таком намеренном действии. Любое намеренное действие может существовать в различных «наклонениях». Наклонения описывают цикл деятельности от определения деятельности к ее планированию, от запланированной или потребованной деятельности к завершенной. Любой экземпляр класса Act имеет одно и только одно наклонение и не меняет его на протяжении своего жизненного цикла. Наклонения — определение, намерение, требование, событие — описывают разные моменты жизненного цикла деятельности. Однако участники каждого такого момента деятельности различны, как и данные, которыми они оперируют. Поэтому наклонение, описанное в экземпляре класса Act, является статическим. Актуализацию процесса деятельности (т. е. движение от ее определения к планированию, требованию и выполнению) называют «циклом деятельности», чтобы отличить ее от «жизненного цикла» единственного экземпляра действия. Экземпляры класса Act, формирующие такой «цикл деятельности», связаны между собой с помощью экземпляров класса ActRelationship. Примеры действий в здравоохранении: клинический тест, оценка состояния здоровья (например, жалобы и диагнозы), выбор цели оказания медицинской помощи, оказание медицинской помощи (например, применение лекарств, хирургическая операция, физиотерапия и психотерапия), ассистирование, мониторинг, ведение пациента; санитарное просвещение пациентов и их близких лиц; услуги нотариуса (например, упреждающие указания или отказ от искусственного поддержания жизни). У действий есть участники, которые могут быть исполнителями или целями действий. Примеры исполнителей — медсестры, врачи, члены семьи, нотариусы и обслуживающие организации — каждый человек или организация, способные к независимым решениям и принимающие ответственность (и отвечающие) за выполняемые действия. Целевыми участниками действия могут быть пациент, супруг пациента, семья или сообщество, биоматериал, взятый у пациента или у любого исследуемого объекта. Поскольку пациенты принимают

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				активную роль в оказании ему медицинской помощи, пациент может быть и активным, и целевым участником одновременно (самолечение или рефлексивные услуги). У действия может быть много активных и много целевых участников, их конкретная роль описывается атрибутом typeCode соответствующего экземпляра класса Participation. В частности, действие координации ухода за пациентом может иметь двух или более активных участников, выполняющих разные роли: действующих от имени пациента или семьи или объединенных ролью целевого участника. Например, медсестра (активный участник) вызывает службу доставки обедов (активный участник) от имени пациента (целевой участник). К действиям относятся «результаты», «ответы» или информационные «продукты процедур», полученные во время действия. В данной модели «результаты» не существуют без действия, и каждый клинический результат, включая полученный случайно, предоставляется через действие. В других наклонениях, например, «определение» или «намерение», под результатами понимаются возможные результаты, ожидаемые результаты или цели получения результатов, полученные результаты
	S: ActClassContract	CNTRCT	contract (контракт)	Соглашение об обязательстве между двумя или более участниками, которое является субъектом гражданского права и правоприменения
	S: ActClassFinancialContract	FCNTRCT	financial contract (финансовый контракт)	(См. также определение класса FinancialContract (финансовый контракт) из ЭИМ). Контракт, значение которого определено в денежном выражении
	Leaf	COV	coverage (полис)	Страховой медицинский полис или план лечения, который по контракту связывает между собой двух или более участников
	S: ActClassControlAct	CACT	control act (управляющее действие)	Действие, представляющее системное действие, такое как изменение состояния другого действия или инициирования запроса. Все управляющие действия представляют собой триггерные события в контексте HL7. Управляющее действие может иметь различные наклонения
	Leaf	ACTN	action (воздействие)	Отправитель просит, чтобы адресат сделал что-то в зависимости от информации о действии, переданной в сообщении. Например, «выполни направление». Адресат должен или отклонить сообщение, или действовать надлежащим способом (определенным специфическими обязанностями получателя сообщения)

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
	Leaf	INFO	Information (информация)	Отправитель отправляет адресату информационное сообщение. У адресата нет обязанности выполнения каких-либо действий вне сферы его собственных интересов (например, «прочтите и запомните, если сочтете нужным»). Это эквивалентно резолюции «для сведения» на письме
	Leaf	STC	state transition control (управление изменением состояния)	Отправитель передает информацию об изменении состояния действия, указанного в сообщении. Эта информация описывает результирующее состояние. Информация может быть передана в форме запроса или команды в зависимости от наклонения действия управления
	S: ActClassObservation	OBS	observation (исследование)	(См. также определение класса Observation (исследование) ЭИМ). Под исследованием понимаются действия, выполняемые для определения ответа на запрос или значения результата. Значения результата обследования (передаваемые в атрибуте Observation.value) включают определенную информацию об исследуемом объекте. Тип и ограничения значений результата зависят от вида выполняемого действия. В клинических документов обычно есть разделы «субъективных» и «объективных» данных, каждый из которых является видом исследования. Кроме того, клинические документы обычно содержат разделы «заключений», которые также являются видами исследований. Таким образом, постановка диагноза — это тоже исследование
	S: ActClassObservationSeries	OBSSER	observation series (серия исследований)	Оболочка (контейнер) для коррелируемых серий исследований, совместно использующих общую систему отсчета. Все исследования с тем же самым кодом должны быть сопоставимыми и относиться к общей системе отсчета. Например, 3-канальный кардиограф записывает ЭКГ с 12 отведениями за 4 шага (3 отведения за один шаг). Записи этих трех отведений были бы объединены с помощью связи типа OBSCOR. И все 4 группы записей, объединенных с помощью связи типа OBSCOR, были бы объединены с помощью связи типа OBSSER, потому что все времена указаны по отношению к одному и тому же моменту (началу регистрации ЭКГ) и все сигналы ЭКГ сняты с фиксированного набора электродов
	Leaf	OBSCOR	correlated observation sequences (коррелируемые исследования)	Хранилище (контейнер) для комплексов связанных исследований (результаты которых имеют тип данных LIST <>), имеющих коррелируемые значения. Все указанные в нем результаты типа LIST <> должны иметь одинаковую длину. Значения типов данных LIST <> связаны по индексу. Например, вторые по порядку значения во всех LIST <> коррелируются. Это похоже на таблицу, где каждый столбец содержит последовательность

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				результатов исследований типа LIST<>, а каждая строка таблицы описывает корреляцию между столбцами. Например, ЭКГ с 12 отведениями содержала бы 13 последовательностей: одна последовательность значений времени и 12 последовательностей для значений, снятых для каждого отведения
	S: ActClassPublic HealthCase	CASE	public health (общественное здоровье)	(См. также определение класса PublicHealthCase ЭИМ.) Описание события, имеющего определенное значение для общественного здоровья, представляется как специализация класса Observation. Обычно это информация об одном или нескольких случаях инфекционного заболевания или другого события, подлежащего регистрации. Описание события общественного здоровья может включать в себя сведения о состоянии здоровья одного лица или сведения о нескольких случаях того же самого заболевания либо условия, которые рассматриваются как угроза общественному здоровью. Типичным примером может служить вспышка заболевания. Определение события общественного здоровья (передаваемое в экземпляре класса PublicHealthCase, у которого атрибут moodCode имеет значение DEF) включает в себя описание клинических, лабораторных и эпидемиологических показателей, связанных с заболеванием или условием, влияющим на общественное здоровье. Есть определения событий для условий, которые подлежат регистрации, а также для тех, которые не подлежат. Есть также определения событий вспышек. Определение события общественного здоровья используется здравоохранением для статистики событий и не должно использоваться в качестве клинических рекомендаций, предназначенных для лечения. Примерами служат СПИД, синдром токсического шока, сальмонеллез и связанные с ними симптомы, используемые при определении угрозы
	Leaf	OUTB	outbreak (вспышка)	Вспышка представляет собой ряд событий общественного здоровья. Датой начала вспышки является дата начала самого раннего случая среди случаев, относящихся к вспышке, а датой ее окончания — дата окончания случая, последнего среди случаев, относящихся к вспышке
	A: ActClassROI			Область интереса (ОИ) в информации, переданной в экземпляре класса Act, описывающем исследование субъекта. В основном используется для вторичного исследования подмножества исследований субъекта. Связь между ОИ и экземпляром класса Act, к которому она относится, задается с помощью экземпляра класса ActRelationship, у которого атрибут typeCode имеет значение SUBJ (предмет), который обязательно должен присутствовать

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
	Leaf	ROIBND	bounded ROI (ограниченная ОИ)	Область интереса (ОИ), определенная для многомерного исследования, такого как серия исследований (OBSSER). ОИ указывается в форме ряда критериев исследования, каждый из которых линеаризует границу области в одной из размерностей многомерного исследования. Связь между ОИ и экземпляром класса Observation, к которому она относится, задается с помощью экземпляра класса ActRelationship, у которого атрибут typeCode имеет значение SUBJ (предмет), который обязательно должен присутствовать. Каждый из ограничивающих критериев исследования связан с ОИ с помощью экземпляра класса ActRelationship, у которого атрибут typeCode имеет значение COMP (компонент). В каждом экземпляре класса Observation, описывающем ограничивающий критерий, атрибут code указывает размерность, а атрибут value задает диапазон значений, включаемых в область. Обычно ограниченная размерность является непрерывной, поэтому значение атрибута value должно иметь тип данных интервала (IVL). Это значение не должно быть указано, если соответствующая размерность только названа, но не ограничивается. Например, ОИ для интервала QT на втором отведении ЭКГ содержала бы два ограничивающих критерия: один ограничивает интервал времени, а второй — имя интервала (QT) на втором отведении ЭКГ (только имя, без ограничений)
	Leaf	ROIOVL	overlay ROI (наложенная ОИ)	Область интереса (ОИ), определенная для изображения с помощью наложения. Обычно используется для указания определенной области изображения, например, указать место изображения, описываемое специалистом, или указать физическое место исследования в виде «кружка» на схематическом изображении человеческого тела. Значения координат измеряются в пикселах. Начало координат находится в верхнем левом углу, положительные значения по оси X отсчитываются направо, а положительные значения по оси Y отсчитываются вниз. Связь между ОИ и экземпляром класса Observation, к которому она относится, задается с помощью экземпляра класса ActRelationship, у которого атрибут typeCode имеет значение SUBJ (предмет), который обязательно должен присутствовать
	Leaf	ALRT	alert (сигнал тревоги)	Исследование, идентифицирующее потенциально отрицательное происшествие, возникшее в результате действия или комбинации действий
	Leaf	CLNTRL	clinical trial (клиническое испытание)	Набор действий, которые определяют эксперимент, проводящийся для оценки эффективности и/или безопасности биофармацевтического продукта (еды, препарата, устройства и т. д.). В наклонении определения действия этот набор нередко включается в протокол клинического испытания;

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				при наклонении, описывающем свершившийся факт, этот набор описывает совокупное действие применения воздействий к одному или нескольким субъектам
	Leaf	CNOD	Condition Node (момент состояния)	Экземпляр класса Observation, описывающий состояние в момент времени, которое включает в себя любые исследования или процедуры, связанные с этим состоянием, а также связи с предыдущими экземплярами моментов состояния, относящимися к одному и тому же состоянию
	Leaf	COND	Condition (состояние)	Наблюдаемое значение или состояние, сохраняющееся в течение длительного времени и потенциально требующее вмешательства или контроля. Тем самым оно отличается от исследования, сделанного в момент времени; оно может существовать до того как оно было выявлено с помощью исследования, или после вмешательства в целях управления состоянием здоровья. Примеры: состояние ремонта устройства, состояние перезагрузки устройства, фактор риска заболевания, финансовый риск, угроза общественному здоровью, беременность, управление здоровьем, хроническое заболевание
	Leaf	DGIMG	diagnostic image (диагностическое изображение)	(См. также определение класса DiagnosticImage ЭИМ.) Класс, предназначенный для хранения атрибутов, уникальных для диагностических изображений
	Leaf	MPROT	monitoring program (программа мониторинга)	Официальная или неофициальная программа отслеживания действий определенного типа либо определенной классификации
	Leaf	SPCOBS	specimen observation (исследование биоматериала)	Исследование биоматериала на лабораторном оборудовании, которое может влиять на обработку, анализ или интерпретацию результата
	S: ActClassSupply	SPLY	Supply (предоставление материала)	(См. также определение класса Supply ЭИМ.) Заказы на представление материала представляют собой простые действия, которые фокусируются на предоставляемом материале. Материал связан с экземпляром класса Supply с помощью экземпляра класса Participation, у которого атрибут typeCode имеет значение PRD (product — товар). Для общих действий предоставления материала важна его точная идентификация (производитель, серийные номера и т. д.). Большая часть подробной информации о предоставлении материала должна быть передана в экземпляре класса Material. Если требуется отдельно описать планирование доставки, доставку и оплату материала,

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				то с экземпляром класса Supply можно связать экземпляр класса Transportation. Для описания услуги отпуска лекарственного средства используется экземпляр класса Supply, связанный с экземпляром класса SubstanceAdministration. В этом случае экземпляр класса SubstanceAdministration описывает введение лекарства, а экземпляр класса Supply — отпуск
	Leaf	DIET	diet (диета)	(См. также определение класса Diet ЭИМ.) Службы питания можно рассматривать как службы представления материала, с некоторыми нюансами, напоминающими службы введения лекарств: детали диеты передаются в экземпляре класса Material, связанном с экземпляром класса Diet помощью экземпляра класса Participation, у которого атрибут typeCode имеет значение PRD (product — товар). Номер диеты, имеющий медицинское значение, можно передать в атрибуте Diet.code, значения которого берутся из словарного домена ActDietCode, однако детали пищи, входящей в диету, и различные сочетания блюд должны передаваться в форме экземпляров класса Material
	A: ActDocument StructureClass			Структура — это контейнер данных внутри документа. У структур есть заголовки, которые могут быть кодированными. Структуры могут быть вложенными и могут содержать разделы
	S: ActClassDocument	DOC	document (документ)	Специализация класса Act, имеющая характеристики, уникальные для служб управления документооборотом
	S: ActClinicalDocument	DOCCLIN	clinical document (медицинский документ)	Медицинский документ представляет собой результат документирования медицинских исследований и услуг, обладающий следующими характеристиками: (1) Постоянство — медицинский документ продолжает существовать в неизменном состоянии в течение времени, определенного местными и нормативными требованиями; (2) ответственность — ответственность за медицинский документ возложена на медицинскую лицо или организацию, оказывающую пациенту медицинскую помощь; (3) Потенциальная юридическая значимость — медицинский документ представляет собой совокупность сведений, предназначенную для придания им юридической значимости; (4) Целостность — медицинский документ аутентифицируется как единое целое, а не отдельными частями без полного содержания документа; (5) Восприятие человеком — медицинский документ должен восприниматься человеком

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
	Leaf	CDALVLONE	CDA Level One clinical document (медицинский документ, представленный на первом уровне архитектуры клинических документов)	Содержание медицинского документа, соответствующее первому уровню архитектуры клинических документов (CDA) HL7
	Leaf	ACCM	accommodation (размещение)	Размещение является услугой лицу или другому живому организму, состоящей в предоставлении места, в котором этот субъект может находиться в течение некоторого времени. Обычно используется для регистрации предоставления пациенту общей палаты, одноместной палаты, двух- или трехместной палаты
	Leaf	ACCT	account (счет)	(См. также определение класса Account.) Бухгалтерский счет, на котором отражаются результаты финансовых операций
	Leaf	ACSN	accession (единица работы)	Единица работы или группа единиц, определенная системой, выполняющей эту работу. Обычно некоторые исполнители направлений на лабораторные анализы передают ссылки на единицы работы при передаче информации об обработке полученных направлений. Нередко с единицей работы связаны один или несколько образцов биоматериала, и в некоторых лабораториях номер единицы работы используется в качестве идентификатора биоматериала (группы биоматериалов)
	Leaf	ADJUD	financial adjudication (согласование оплаты)	Процесс преобразования, при котором выставленный счет преобразуется в согласованный счет. Представляет процесс согласования счета (требования страхового возмещения). Результатами согласования могут быть решения принять счет как есть, внести поправки или отказать в приемке счета. Результаты согласования содержат два компонента: результаты процесса согласования и скорректированный (или согласованный) счет или требование страхового возмещения
	Leaf	CONS	consent (согласие)	Класс Consent представляет информированные согласия и все аналогичные медико-юридические процедуры между пациентом (или его законным представителем) и поставщиком медицинской помощи. Примерами служат информированное согласие на операцию, информированное согласие на участие в клиническом испытании, уведомление о выгодоприобретателе, отказ от услуги вопреки медицинским показаниям, соглашение о раскрытии информации и т. д.

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				Детали согласий различны. Нередко в учреждении имеется несколько разных форм документов согласия, используемых для разных целей, включая напоминание врачу о том, что должно быть упомянуто в документе. Такие формы могут также содержать информацию, предназначенную для санитарного просвещения пациентов. При обмене данными электронной медицинской карты согласия сами по себе являются актами создания информации, которыми надо управлять аналогично медицинской деятельности. Поэтому класс Consent моделируется как специальный класс Act. «Подписи» документа согласия представляются в электронном виде как экземпляры класса Participation, связанные с объектом согласия. Обычно у экземпляра класса Participation атрибут typeCode имеет значение PRF (performer — исполнитель) для медицинского работника, получающего согласие от пациента, и значение CNS (consenter — согласный) для пациента или его законного представителя. Для некоторых видов согласия может требоваться участие свидетеля или нотариуса (например, отказ от искусственного поддержания жизни, упреждающие указания). В согласиях, где медицинский работник не требуется (например, отказ от искусственного поддержания жизни), исполнителем может быть сам пациент или нотариус. В некоторых видах согласия предусмотрена минимально необходимая задержка между согласием и предоставлением услуги, позволяющая пациенту заново обдумать свое решение. Эта минимальная задержка может быть выражена в определении услуги с помощью атрибута ActRelationship.pauseQuantity, который указывает время, которое должно пройти с момента получения согласия до начала предоставления услуги
	Leaf	CONTREG	container registration (регистрация контейнера)	Действие регистрации контейнера с помощью автоматизированного датчика, например, считывателя штрих-кода, или вручную по его маркировке
	Leaf	CTTEVENT	clinical trial timepoint event (момент события клинического испытания)	Идентифицированный момент клинического испытания, в котором одно или несколько запланированных действий должны быть выполнены (наклонение определения) или фактически выполнены (наклонение события). К этим действиям может относиться, а может и не относиться контакт медицинского специалиста с пациентом
	Leaf	ENC	encounter (обращение)	Взаимодействие между пациентом и участником (участниками) процесса оказания медицинской помощи в целях получения или оценки состояния здоровья. Например, посещение амбулаторным пациентом нескольких отделений, медицинская помощь на дому (включая физиотерапию),

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				стационарная помощь, обращение в отделение неотложной помощи, помощь в полевых условиях (например, при дорожном происшествии), посещение участкового терапевта, производственная терапия, телефонный звонок
	Leaf	INC	incident (инцидент)	Событие, произошедшее за пределами контроля одной или более вовлеченных в него сторон. Включает понятие несчастного случая
	Leaf	INFRM	inform (информирование)	Действие передачи информации субъекту и понимания ее содержания. Обсуждение: это действие может использоваться, чтобы передать запрос предоставления пациенту или поставщику медицинской помощи информации о другом действии или чтобы указать, что лицу сообщили о конкретном действии
	Leaf	INVE	invoice element (элемент счета–фактуры)	(См. также определение класса InvoiceElement ЭИМ), Представляет понятия, связанные с обработкой счетов на оплату лечения
	Leaf	LIST	working list (рабочий лист)	(См. также определение класса WorkingList). С помощью экземпляра класса WorkingList, связанного с несколькими экземплярами класса Act с помощью отношений ActRelationship, формируется динамический список действий, отражающий потребность отдельного работника, бригады или организации в списках действий, выполняемых по различным клиническим и административным причинам. Примерами рабочих листов служат списки жалоб, списки целей медицинской помощи, списки аллергий и списки текущих дел
	Leaf	PCPR	patient care provision (предоставление медицинской помощи)	Предоставление медицинской помощи означает принятие исполнителем ответственности за медицинскую помощь пациенту или группе пациентов. Обсуждение: событие оказания медицинской помощи может иметь место без выполнения каких-либо медицинских действий. Рамки медицинской помощи определяются значением атрибута Act.code. Примеры: 1) предпочтительное оказание первичной медицинской помощи: участковый терапевт принимает в ней участие как основной исполнитель, а пациент — как автор; 2) направление пациента к специалисту, выданное врачом общей практики (код PCPR в действии, имеющем наклонение запроса; автором является врач общей практики, а основным исполнителем — специалист); 3) куратор пациента или группы пациентов; 4) закрепление медсестер за пациентами в каждой смене

Продолжение таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
	Leaf	PROC	procedure (процедура)	(См. также определение класса Procedure.) Действие, непосредственным и основным результатом которого (постусловием) является изменение физического состояния субъекта. Примеры: к процедурам можно отнести нарушение целостности некоторой поверхности тела (например, разрез при хирургической операции), выполнение консервативных процедур, например, вправление вывихнутого сустава, физиотерапию, манипуляции хиропрактика, массаж, бальнеотерапия, иглоукалывание, шиатсу и т. д. За пределами клинической медицины можно привести такие примеры процедур как изменение окружающей среды (например, спрямление рек, осушение болот, возведение дамб), ремонт или модернизация машин и т. д. Обсуждение: применительно к клинической медицине, процедура — один из нескольких типов клинических действий, например, исследование, лекарственные назначения и коммуникативные воздействия (к примеру, обучение, консультация, психотерапия, представленные в форме экземпляров класса Act без специальных атрибутов). Понятие процедуры не поглощает эти другие понятия и, в свою очередь, не поглощается ими. Примечательно, что оно не включает в себя все действия, целью которых является вмешательство или лечение. Будет ли физическое изменение полезным субъекту или предполагается полезным, не имеет значения, существенно лишь то, что целью действия является изменение физического состояния субъекта. Выбор между способами представления реальных действий в форме экземпляров классов основан на том, применимы ли специфические свойства процедуры и будет ли физическое изменение необходимым постусловием деятельности или ее части. Например, получение рентгеновского изображения иногда называется «процедурой», но в ЭИМ оно не отображается на экземпляр класс Procedure, поскольку целью рентгеновского снимка не является изменение физического состояния тела
	Leaf	REG	registration (регистрация)	Описывает действие управления в регистре информацией о сущности или роли. Этот класс является весьма общим, рассчитанным на представление разнообразных регистров лиц, пациентов, медицинских работников, оборудования и т. д. При необходимости можно определить конкретные регистры как специализацию этого класса
	Leaf	SBADM	substance administration (лекарственное назначение)	(См. также определение класса SubstanceAdministration). Действие введения или иного применения субстанции к субъекту. Обсуждение: эффект применения субстанции обычно вызывается биохимическими процессами,

Окончание таблицы 1

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				однако это не обязательно. Например, лучевую терапию в целом можно описать аналогичным образом, особенно соматическую терапию, к примеру, радиоактивным йодом. К этому же классу действий относится применение химического лечения к определенной области. Примеры: протокол химиотерапии, рецепт, запись о вакцинации
	Leaf	SPCTRT	specimen treatment (обработка биоматериала)	Процедура или обработка, применяемая к биоматериалу в целях его подготовки к анализу
	Leaf	TRNS	transportation (транспортировка)	Транспортировкой является доставка груза (людей или материалов) из пункта отправки в пункт назначения. Таким образом, в дополнение к целям, общим для любой услуги (например, исполнитель, устройство и т. д.) каждая услуга транспортировки имеет три дополнительных целевых экземпляра, имеющих типы груза, пункта отправки, пункта назначения
	Leaf	XACT	financial transaction (финансовая операция)	(См. также определение класса FinancialTransaction). Подкласс класса Act, представляющий движение денежных сумм между двумя счетами. Если у экземпляра класса FinancialTransaction атрибут moodCode имеет значение ORD, то этот экземпляр трактуется как требование выполнения операции или перевода суммы с одного счета на другой. Если атрибут moodCode имеет значение EVN, то этот экземпляр класса FinancialTransaction содержит информацию об уже выполненной операции.

Таблица 2 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 2 — Домен ActMood (используется в атрибуте Act.moodCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	A: ActMoodCompletionTrack			Наклонения описывают цикл деятельности от определения деятельности к ее планированию, от запланированной или потребованной деятельности к завершенной
2	S: ActMoodIntent	INT	intent (намерение)	Намерение или план выполнения услуги. Экскурс в историю: в предыдущих версиях ЭИМ наклонение намерения представлялось в виде отдельной иерархии классов, называвшейся Service_intent_or_order

Продолжение таблицы 2

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
3	Leaf	APT	appointment (запланированное действие)	Запланированное действие, которое должно быть выполнено в определенное время в определенном месте
3	Leaf	ARQ	appointment request (запрос планирования)	Запрос планирования действия
3	Leaf	PRMS	promise (обязательство)	Намерение выполнить услугу, имеющее силу обязательства, т.е. другие стороны могут полагаться на то, что сторона, взявшая на себя обязательство выполнения услуги, намерена ее выполнять. Обязательство может быть инициировано другой стороной (solicited) или взято по собственной инициативе (unsolicited)
3	Leaf	PRP	proposal (предложение)	Предложение выполнить действие, не имеющее обязательной силы. Используется при передаче информации о предложении, которое не является ни направлением, ни заказом. Профессиональная ответственность за выполнение «предложения» может иметь, а может и не иметь место
3	Leaf	RQO	request (требование)	Требование или заказ услуги направляется заказчиком (автором требования) исполнителю (сущности, выполняющей требование). Обоснование: понятия «требование» и «заказ» рассматриваются как разные, поскольку у приказа, ассоциированного с заказом, есть определенные последствия. На практике, однако, это различие не имеет общего функционального значения при взаимодействии медицинских информационных систем. «Заказ» может быть отклонен по разным клиническим и деловым причинам, а понятие «требование» подразумевает, что получатель требования (исполнитель) должен дать ответ его отправителю (автору). Действительно, во многих регионах, в том числе в Австралии и Европе, общеупотребительным термином является «требование». Таким образом, данный код охватывает оба этих понятия, поскольку между ними не проведено какое-либо полезное различие. Если указание выполнить действие должно быть ассоциировано с требованием, то оно должно выполняться в соответствии с «местными» правилами выполнения транзакций. Если в будущем комитету HL7 понадобится провести различие между двумя этими понятиями, то отдельные варианты могут быть добавлены как специализации этого кода. Критичным моментом является различие между этим понятием и «намерением», для которого оно является специализацией. Намерение является решением одной стороны, а именно, его

Окончание таблицы 2

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				автора. Напротив, требование охватывает решения двух сторон, автора и исполнителя, с обязательством исполнителя дать ответ, будет ли он действительно выполнять это требование
3	Leaf	SLOT	resource slot (ячейка ресурса)	Периоды времени в расписании ресурса. Назначения занимают одну или несколько ячеек расписания. Ячейка, открытая для назначения, является доступной, а ячейка, которая предназначена для административных целей, является заблокированной. Ячейка ресурса, которая «предварительно» занята, называется зарезервированной
2	Leaf	DEF	definition (определение)	Определение услуги (в справочнике). Исторический экскурс: в предыдущих версиях ЭИМ наклонение определения выделялось в отдельную иерархию классов, а именно, Master_service
2	Leaf	EVN	event (occurrence) (событие, происшествие)	Услуга, действительно имеющая место. Это может быть текущая услуга или документирование ранее предоставленной услуги. Исторический экскурс: в предыдущих версиях ЭИМ наклонение события выделялось в отдельную иерархию классов, которая вначале имела имя Patient_service_event, а затем Service_event
1	A: ActMoodPredicate			Любое из указанных ранее наклонений услуги (например, событие, намерение или цель) может быть превращено в предикат, используемый в качестве критерия в условиях (или запросах). Однако в настоящее время допускаются только критерии для событий услуги
2	Leaf	EVN.CRT	event criterion (критерий события)	Критерий или условие, применяемые к ассоциированной услуге, которую надо рассмотреть
2	Leaf	GOL	Goal (Цель)	Ожидание желательного результата исследования в определенный будущий момент времени
2	Leaf	OPT	option (дополнительная возможность)	Дополнительная возможность представляет собой альтернативное множество связанных значений свойств. Такие альтернативные множества обычно используются в определениях услуги или в направлениях, чтобы описать дополнительные возможности. Дополнительная возможность может быть использована только в составе группы, т. е. все возможные значения должны быть указаны вместе. Исторический экскурс: в стандартах HL7 v2.x дополнительная возможность имелась в специальном случае альтернативных способов введения лекарств (которые могли быть указаны в сегменте RXR)

Таблица 3 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 3 — Домен ActRelationshipCheckpoint (используется в атрибуте ActRelationship.checkpointCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	Leaf	B	beginning (начало)	Условие проверяется перед каждым выполнением услуги (WHILE условие DO услуга).
1	Leaf	E	end (конец)	Условие проверяется в конце цикла повторения услуги. Услуга повторяется только в том случае, если условие выполнено (DO услуга WHILE условие)
1	Leaf	S	entry (вход)	Условие проверяется однократно перед выполнением услуги (IF условие THEN услуга)
1	Leaf	T	through (в течение)	Условие должно быть выполнено в течение всего процесса выполнения услуги. Как только условие перестает выполняться, услуга должна прерваться (асинхронно) (асинхронно WHILE цикл). Услуга должна допускать прерывание
1	Leaf	X	exit (выход)	Условие проверяется в контрольной точке цикла, например, такая проверка является одним из шагов плана действий, и если она привела к отрицательному результату, осуществляется выход из цикла

Таблица 4 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 4 — Домен ActRelationshipJoin (используется в атрибуте ActRelationship.joinCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	Leaf	D	detached (отделенный)	Отделить данную ветвь от других ветвей, чтобы не надо было восстанавливать ее синхронизацию с ними
1	Leaf	K	kill (прекращение)	Как только выполнение всех других параллельных ветвей завершится, прервать и прекратить выполнение данной ветви
1	Leaf	W	wait (ожидание)	Ждать завершения данной ветви
1	Leaf	X	exclusive wait (исключающее ожидание)	Ждать завершения хотя бы одной ветви из совокупности ветвей с исключающим ожиданием, затем прекратить выполнение всех остальных ветвей этой совокупности

Таблица 5 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 5 — Домен ActRelationshipSplit (используется в атрибуте ActRelationship.splitCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	Leaf	E1	exclusive try once (исключающее ветвление с однократной проверкой)	Предусловие, ассоциированное с ветвью, однократно вычисляется. Если оно выполнено, то к ветви можно перейти. Все другие исключающие ветви конкурируют друг с другом и лишь одна из них может быть выбрана. Таким способом реализуются операторы COND, IF и CASE условие или «XOR-ветвление». Порядок рассмотрения ветвей может быть указан в атрибуте Service_relationship.priority_nmb
1	Leaf	EW	exclusive wait (исключающее ветвление с ожиданием)	Ветвь выбирается, как только выполнится предусловие, ассоциированное с этой ветвью. Если условие не выполнено, то переход к ветви может быть осуществлен позже, когда оно станет выполненным. Все другие исключающие ветви конкурируют друг с другом и лишь одна из них может быть выбрана. Каждая ожидающая ветвь выполняется параллельно, при этом по умолчанию используется код соединения wait (см. далее). Порядок рассмотрения ветвей может быть указан в атрибуте Service_relationship.priority_nmb
1	Leaf	I1	inclusive try (включающее ветвление с однократной проверкой)	Ветвь выполняется, если это разрешено ассоциированными с ней предусловиями. Если они не разрешают выполнение, то ветвь отбрасывается. Включающие ветви не подавляются другими ветвями и не подавляют другие ветви
1	Leaf	IW	inclusive wait (включающее ветвление с ожиданием)	Ветвь выполняется, если это разрешено ассоциированными с ней предусловиями. Если они не разрешают выполнение, то переход к ветви может быть осуществлен позже, когда условие станет выполненным. Включающие ветви не подавляются другими ветвями и не подавляют другие ветви. Каждая ожидающая ветвь выполняется параллельно, при этом по умолчанию используется код соединения wait (см. ниже)

Таблица 6 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 6 — Домен ActRelationshipType (используется в атрибуте ActRelationship.typeCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	A: ActRelationship Conditional			Указывает, при каких обстоятельствах (описанных в целевом экземпляре класса Act) действие, описанное в экземпляре класса Act (источнике связи) может произойти, должно произойти, не должно произойти или произошло

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	S: ActRelationship Reason	RSON	has reason (имеет причину)	Причина или обоснование услуги. Причинно-следственная связь слабее триггера, она только предполагает, что некоторая услуга может быть, а может и не быть причиной некоторого действия. Из нее не следует, что эта причина требует, чтобы действие было выполнено, или что эта причина необходима для выполнения действия. Кроме того, в противоположность триггеру, между причиной и действием нет строгой зависимости во времени. Обсуждение: в предыдущих версиях код SUGG (suggests — предполагает) был описан как «обращение причинно-следственной связи». Этот код был отменен в пользу индикатора inversionInd, являющегося атрибутом класса ActRelationship
3	Leaf	MITGT	mitigates (облегчение)	Действие-источник удаляет целевое действие или понижает его эффект
2	Leaf	CIND	has contra-indication (имеет противо-показания)	Противопоказание является противоположностью показания (причины). Оно указывает условие, при котором действие не должно выполняться. Источник и цель связи могут быть любыми видами услуг, при этом у целевого экземпляра класса Act атрибут moodCode должен иметь значение EVN.CRT (event-criterion — критерий события). Каким образом выражается категоричность противопоказаний (например, относительная, абсолютная), остается открытым вопросом. Атрибут priority_nmb не должен использоваться для этих целей
2	Leaf	PRCN	has pre-condition (имеет предусловие)	Требование, которое должно быть выполнено до начала выполнения услуги. Целью предусловия может быть экземпляр класса Act, у которого атрибут moodCode имеет значение EVN.CRT (event-criterion — критерий события). При наличии нескольких предусловий должны использоваться атрибуты союзов (AND, OR, XOR)
2	Leaf	TRIG	has trigger (имеет триггер)	Предусловие, при выполнении которого должно быть инициировано выполнение действия-источника. Целью обычно является экземпляр класса Act, у которого атрибут moodCode имеет значение EVN.CRT (event-criterion — критерий события). Если этот экземпляр содержит отчет о факте выполнения (т.е. критерий был выполнен), то атрибут moodCode может иметь значение EVN (event — событие). Может быть указана задержка между срабатыванием триггера и выполнением действия Обсуждение: это понятие включает в себя действие, необходимое для выполнения услуги или для финансового инструмента, например, программы или полиса медицинского страхования. В этом случае триггером может быть появление определенного события, например, выход затрат за лимит страхового покрытия

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: ActRelationship HasComponent	COMP	has component (имеет компонент)	Совокупность услуг, являющихся шагами или подзадачами деятельности, описанной в экземпляре класса Act, являющемся источником связи. Эти услуги могут выполняться последовательно или параллельно
2	Leaf	ARR	arrival (прибытие)	Связь между экземпляром класса Transportation (цель связи) и действием-источником, указывающая, что субъект действия-источника прибывает для выполнения этого действия способом, описанным в экземпляре класса Transportation
2	Leaf	CTRLV	has control variable (имеет управляющий параметр)	Связь между экземпляром класса Act (источник) и управляющим параметром. Например, если исследование, представленное экземпляром класса Observation, выполняется с помощью устройства, представленного экземпляром класса Device, то экземпляр класса Observation может иметь связь этого типа с экземпляром класса Device в целях документирования параметров устройства, которые могли влиять на результат исследования
2	Leaf	DEP	departure (убытие)	Связь между экземпляром класса Transportation (цель связи) и действием-источником, указывающая, что субъект действия-источника убыл после выполнения этого действия способом, описанным в экземпляре класса Transportation
1	S: ActRelationship Outcome	OUTC	has outcome (имеет последствие)	Исследование, которое должно произойти или фактически произошло в результате или вследствие условия или действия (иногда называемом постусловием). Целевой экземпляр класса Observation должен описывать цель, риск или иной критерий. Для сложных условий должен использоваться атрибут соединения conjunctionCode
2	Leaf	GOAL	has goal (имеет цель)	Цель, определенная для состояния здоровья пациента. Запланированная медицинская помощь должна быть направлена на достижение этой цели. Источником связи является исследование или описание условия, целью должен быть экземпляр класса Observation, у которого атрибут moodCode имеет значение GOL (goal — цель)
2	Leaf	OBJC	has continuing objective (имеет непрерывную цель)	Желаемое состояние, которое должно поддерживать действие услуги. Например, поддерживать систолическое давление крови между 90 и 110 мм рт.ст. Источником связи является услуга вмешательства. Целью должен быть экземпляр класса Observation, у которого атрибут moodCode имеет значение EVN.CRT (event-criterion — критерий события)

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	OBJF	has final objective (имеет финальные требования)	Желаемый конечный результат действия услуги. Источником является любая услуга (обычно вмешательство). Целью должен быть экземпляр класса Observation, у которого атрибут moodCode имеет значение EVN.CRT (event-criterion — критерий события)
2	Leaf	RISK	has risk (имеет риск)	Требующее внимания нежелательное развитие состояния пациента, которое вероятно может стать заболеванием или даже иметь опасные последствия для его здоровья
1	S: ActRelationship Pertains	PERT	has pertinent information (имеет связанную информацию)	Весьма неспецифичная связь между двумя элементами клинической информации. Оно не позволяет судить о роли связанной информации
2	A: ActRelationship Posts			Коды, описывающие связь между действием и финансовым инструментом, например, финансовой операцией, счетом или элементом счета-фактуры
3	Leaf	CHRG	has charge (требует оплаты)	Связь, обеспечивающая возможность ассоциировать финансовую операцию (цель связи) с оплатой клинического действия (источник связи), обеспечивающего выполнение или предоставление услуги. Финансовая операция (счет) определяет сумму оплаты, причитающуюся за выполненную или предоставленную услугу. Сумма оплаты и сумма затрат являются разными терминами. Сумма оплаты определяет, сколько должна заплатить другая организация или другая структурная единица данной организации. Сумма затрат указывает, во что обошлось данной организации выполнение услуги или предоставление продукта
3	Leaf	COST	has cost (имеет затраты)	Связь, обеспечивающая возможность ассоциировать финансовую операцию (цель связи) с затратами на клиническое действие (источник связи), обеспечивающее выполнение или предоставление услуги. Финансовая операция определяет затраты на выполненную или предоставленную услугу. Сумма оплаты и сумма затрат являются разными терминами. Сумма оплаты определяет, сколько должна заплатить другая организация или другая структурная единица данной организации. Сумма затрат указывает, во что обошлось данной организации выполнение услуги или представление продукта

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
3	Leaf	CREDIT	has credit (имеет кредит)	Отношение кредита связывает финансовую операцию со счетом. Выполнение операции кредита может увеличить или уменьшить баланс счета в зависимости от его типа. По активным счетам кредит уменьшает баланс счета, по пассивным — увеличивает
3	Leaf	DEBIT	has debit (имеет дебет)	Отношение дебета связывает финансовую операцию (цель связи) со счетом (источник связи). Выполнение операции дебета может увеличить или уменьшить баланс счета в зависимости от его типа. По активным счетам дебет увеличивает баланс счета, по пассивным — уменьшает
2	S: has support	SPRT	has support (имеет обоснование)	Используется для указания, что существующая услуга является обоснованием для нового исследования. Это обоснование относится к тому же действующему лицу, которое объявляет новое исследование. Источником связи должно быть исследование, целью — любая услуга (например, чтобы указать состояние, вызванное вмешательством)
3	Leaf	SPRTBND	has bounded support (имеет обоснование с указанием границ)	Специализация кода SPRT (has support — имеет обоснование). Используется, чтобы связать вторичное исследование с областью интереса для многомерного исследования, при условии, что эта область указывает точные границы вторичного исследования, а не просто отмечает примерное расположение. Например, если на ЭКГ видны начало и конец подъема сегмента ST, то такая связь укажет область интереса, связанную с исследованием «подъем сегмента ST» и определяющую точные начало и конец случая подъема. Напротив, если область интереса содержит эпизод подъема сегмента ST, но не определяет его границы, то надо использовать более общий тип связи SPRT (has support — имеет обоснование). Аналогичным образом, если область интереса на изображении определяет точные границы «ожога первой степени», то используется связь с кодом SPRTBND, а если она только указывает примерное место ожога, то используется связь с более общим кодом SPRT
2	Leaf	AUTH	authorized by (авторизован чем-либо)	Связь, в которой целевой экземпляр класса Act описывает авторизацию или сертификацию экземпляра класса Act, являющегося источником связи
2	Leaf	CAUS	is etiology for (есть этиология чего-либо)	Утверждение, что результат нового исследования предположительно выявил причину другого существующего исследования. Это предположение относится к тому же действующему лицу, которое объявило исследование. Эта связь сильнее и специфичнее связи обоснования. Например, рост золотистого стафиллококка при анализе экссудата может указывать, что этот микроорганизм является причиной абсцесса.

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				Источником связи (причиной) обычно является исследование, но в принципе источником может быть любая услуга. Целью связи обязано быть исследование
2	Leaf	COVBY	covered by (покрывается чем-либо)	Связь, в которой целевой действие обеспечивает финансовое покрытие или иное административное воздействие на действие-источник. Например, оплата элемента счета-фактуры покрывается одним или несколькими экземплярами страховых полисов
2	Leaf	DRIV	is derived from (является производным от)	Эта связь обеспечивает явную ассоциацию производного исследования с его входными параметрами. Источником и целью связи должны быть экземпляры класса Observation, обычно имеющие числовые результаты. Например, исследование анионного провала (anion-gap) является производным от конкретных исследований содержания кальция (калия), хлоридов и бикарбоната и может быть ассоциировано с ними
2	Leaf	EXPL	has explanation (имеет объяснение)	Это отношение, обратное к обоснованию. Оно используется для указания, что результат данного исследования объясняется другим результатом исследования или условием
2	Leaf	LIMIT	limited by (ограничен чем-либо)	Отношение, при котором на содержание действия-источника накладываются ограничения (пределы), описанные в элементах целевого действия. Например, авторизация может иметь денежный предел (до 500 долл. США). У целевого экземпляра класса Act атрибут moodCode должен иметь значение EVN.CRT (event-criterion — критерий события)
2	Leaf	MFST	is manifestation of (является проявлением чего-либо)	Утверждение, что результат нового исследования указывает на проявление того, что было обнаружено в предыдущем исследовании или действии. Это утверждение относится к тому же действующему лицу, которое установило факт проявления. Эта связь сильнее и специфичнее обращения связи обоснования. Например, повышенная агрессивность может рассматриваться как проявление (эффект) имеющегося у пациента гипертиреоза. Это означает, что трудно распознать симптом, если он не является общим проявлением известного состояния. Целью связи (причиной) может быть любая услуга, а источник (проявление) должен быть описан в экземпляре класса Observation
2	Leaf	NAME	assigns name (присваивает имя)	Используется для присваивания «имени» потоку состояний. Источником может быть узел условия, целью может быть любая услуга

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	PREV	has previous instance (имеет предшественника)	Отношение, в котором целевое действие является предшественником действия-источника. Обычно экземпляры класса Act, описывающие эти действия, похожи, но не идентичны. При покрытии расходов на медицинскую помощь это отношение используется, чтобы связать элемент счета-фактуры с предшествующим элементом, который мог относиться к той же самой группе услуг
2	Leaf	REFR	refers to (ссылается на что-либо)	Отношение, в котором действие-источник ссылается на целевое действие. С его помощью можно задать простое отношение ссылки, в котором проводится различие между ссылкой и тем, на что она ссылается
2	Leaf	REFV	has reference values (имеет референтные пределы)	Референтные пределы являются по своей сути описателями класса значений результатов исследований, которые объявляются «нормой», «патологией» или «критичными значениями». Эти пределы могут зависеть от пола, возраста и других критериев. Источником и целью связи должны быть экземпляры класса Observation, при этом у целевого экземпляра атрибут moodCode должен иметь значение EVN.CRT (event-criterion — критерий события). Этот тип связи может использоваться для описания триггеров, инициирующих тревожные сигналы при выходе результатов за референтные пределы
2	Leaf	SUBJ	has subject (имеет предмет действия)	Связывает один экземпляр класса Act с другим экземпляром класса Act, описывающим основной предмет действия, указанного в первом экземпляре. Примеры: 1. Первый экземпляр класса Act может описывать управление действием, указанным во втором экземпляре. 2. Первый экземпляр описывает область интереса, указывающую регион в результатах исследования, содержащихся во втором экземпляре. 3. Первый экземпляр описывает действие отчета или уведомления, в котором содержание второго экземпляра отражается для конкретной новой цели. Ограничения: у одного действия может быть несколько предметов действия. Обоснование: тип связи между действиями SUBJ (has subject — имеет предмет действия), передаваемый в атрибуте ActRelationship. typeCode похож на тип участия SBJ (subject — субъект), передаваемый в атрибуте Participation.typeCode. Действия, которые, в основном, оперируют физическими предметами, связаны между собой экземплярами класса Participation, действия, которые в основном оперируют другими действиями (другой информацией) связаны между собой экземплярами класса ActRelationship

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	SUMM	summarized by (является итоговой суммой)	Действие, содержащее итоговые суммы для списка или множества подчиненных действий. Например, итоговая сумма финансовых операций, имевших место в конкретном расчетном периоде
1	S: ActRelationship Sequel	SEQL	is sequel (является продолжением)	Связь между действиями, при которой источник связи является продолжением цели связи. В принципе действие-источник должно быть того же типа, что и целевое действие связи, но значения атрибутов наклонения moodCode у них могут различаться (что часто имеет место). Целевое действие такой связи называется предшественником. Примерами отношения продолжения служат пересмотр, преобразование, действие, являющееся производным от прототипа (подобно тому как специализация является производной от генерализации), следующий этап лечения, реализация интерфейса, создание экземпляра класса
2	S: ActRelationship Excerpt	XCRPT	Excerpts (является извлечением)	Содержание источника связи извлечено из содержания цели связи
3	Leaf	VRXCRPT	Excerpt verbatim (дословное извлечение)	Содержание источника связи является прямой цитатой из содержания цели связи
2	S: ActRelationship Fulfills	FLFS	fulfills	Экземпляр класса Act источника связи описывает полное или частичное выполнение действия, описанного в целевом экземпляре класса Act. Значение атрибута moodCode первого экземпляра должно совпадать со значением атрибута moodCode второго экземпляра или отвечать более актуальному наклонению действия
3	Leaf	OCCR	occurrence (случай)	Действие-источник описывает однократный случай циклического целевого действия. Атрибуты moodCode экземпляров класса Act, описывающих источник и цель, могут быть любыми среди последовательности наклонений от планирования до завершения, но при этом атрибут moodCode экземпляра класса Act, являющегося источником связи, должен быть тем же самым или более близким к концу этой последовательности (например, случай намерения может быть событием, но не наоборот)
3	Leaf	OREF	references order (ссылается на направление)	Связывает запрос записи на выполнение услуги или запись расписания на ее выполнение с направлением на планируемую услугу
3	Leaf	SCH	schedules request (планирует выполнение запроса в расписании)	Связывает определенное время (и определенные ресурсы) с запросом на планирование услуги или другим намерением

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	S: ActRelationship Replacement	RPLC	replaces (заменяет)	Действие-источник заменяет существующее целевое действие. Код состояния целевого действия изменяется на obsolete (устаревшее), но информация о целевом действии обычно оставляется в системе для хранения истории изменений. Действие-источник и целевое действие должны иметь одинаковый тип
3	Leaf	SUCC	succeeds (является преемником)	Новое направление, которое добавляется к своему предшественнику, но не заменяет его полностью
2	Leaf	APND	is appendage (является дополнением)	Дополнение (источник связи) к существующему объекту услуги (цель связи), содержащее дополняющую информацию. Дополнение само по себе является отдельным объектом услуги, связанным с дополняемым объектом услуги. Дополняемый объект услуги остается таким, каким был, его содержание и состояние не изменяются
2	Leaf	DOC	documents (документирует)	Действие-источник документирует целевое действие
2	Leaf	ELNK	episodeLink (связывает части эпизода)	Представляет ассоциацию, связывающую два экземпляра одного и того же действия во времени, указывая, что эти экземпляры являются частью одного и того же эпизода, например, привязывает два момента состояния к эпизоду заболевания или два посещения пациента к эпизоду медицинской помощи (обращению пациента)
2	Leaf	GEN	has generalization (имеет обобщение)	Отношение обобщения может использоваться для представления качественного знания об услугах (например, общим обобщением для амилорида, триамтерена и спиронолактона является термин «калийсберегающий диуретик»)
2	Leaf	GEVL	evaluates (goal) (оценивает соответствие цели)	Оценка соответствия цели связывает исследование (планируемое или выполненное) с целью, чтобы указать, что исследование является оценкой соответствия цели. При наличии заданной цели и конкретного результата исследования можно вычислить «расстояние до цели» (т.е. разность между целевым и наблюдаемым значением) и не посылать его как отдельный результат
2	Leaf	INST	instantiates (master) (реализует услугу, описанную в справочнике)	Используется для указания связи между потенциальной услугой (описанной в «справочнике услуг» или запланированной) и фактически выполненной услугой, которую можно рассматривать как реализацию потенциальной услуги. При реализации могут перекрываться значения по умолчанию, описанные в справочнике

Продолжение таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	MTCH	matches (trigger) (совпадает с условием триггера)	Совпадение с условием триггера связывает фактическую услугу (например, выполненное исследование или процедуру) с экземпляром класса Act, у которого атрибут moodCode имеет значение EVN.CRT (event-criterion — критерий события). Например, если триггером является «наличие боли» и боль действительно имеет место, и если наличие боли приводит к срабатыванию триггера, то исследование наличия боли может быть связано с триггером
2	Leaf	OPTN	has option (имеет дополнительную возможность)	Связь, описывающая более детальные свойства действия-источника по сравнению с целевым действием. Таким образом, действие-источник является специализацией целевого действия, но вместо указания всех наследуемых свойств содержит только информацию о новых свойствах или уточнение унаследованных свойств. Типичным примером служит указание определенных альтернативных вариантов одного вида действия. Атрибут priorityNumber используется, чтобы приписать веса уточнениям, определяющие их приоритеты. Пример: несколько способов применения лекарства можно описать как одно действие лекарственного назначения для общего случая, дополненное несколькими уточнениями, описывающими другие возможные способы применения
2	Leaf	REV	reverses (обращает)	Отношение между действием-источником и целевым действием, при котором действие-источник обращает или отменяет ранее выполненное целевое действие. Пример: финансовая операция (например, операция с дебетом счета) выполнена по ошибке и должна быть обращена (например, используя операцию с кредитом счета). Такая обратная операция идентифицируется как отмена (или обращение) ранее выполненной целевой операции. Ограничения: атрибут moodCode экземпляра класса Act, являющегося целью связи, должен быть тем же самым или более близким к концу последовательности наклонений от планирования до завершения. Например, если атрибут moodCode экземпляра класса Act, являющегося целью связи, имеет значение EVN, то у экземпляра класса-источника этот атрибут может иметь значение EVN или INT. Если атрибут moodCode целевого экземпляра класса имеет значение EVN, то у экземпляра класса-источника этот атрибут может иметь значение INT

Окончание таблицы 6

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	UPDT	updates (condition) (изменяет состояние)	Это отношение является специфичной связью моментов состояния, объединяющей их в поток состояний. Источником связи является новый момент состояния, а целью связи — самый последний по времени момент состояния существующего потока
2	Leaf	XFRM	transformation (преобразование)	Используется, когда целевой экземпляр класса Act является преобразованием экземпляра класса Act — источника связи. (Например, используется, чтобы показать, что документ в формате CDA является преобразованием документа в формате DICOM SR)

Таблица 7 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 7 — Домен ActStatus (используется в атрибуте Act.statusCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: ActStatusNormal	normal	normal (нормальное)	Охватывает все ожидаемые состояния объекта услуги, за исключением nullified и obsolete, которые представляют необычные терминальные состояния жизненного цикла
2	Leaf	aborted	aborted (прервано)	Действие было прекращено до его ожидаемого завершения
2	Leaf	active	active (активно)	Действие может выполняться или выполняется
2	Leaf	cancelled	cancelled (отменено)	Действие было отменено до того, как начало выполняться
2	Leaf	completed	completed (завершено)	Действие нормально завершилось после выполнения всех его составляющих
2	Leaf	held	held (отложено)	Действие, находящееся на подготовительной стадии, отложено. Оно не может стать активным, пока не будет выведено из этого состояния
2	Leaf	new	new (новое)	Действие находится на подготовительной стадии и его выполнение еще не начато
2	Leaf	suspended	suspended (приостановлено)	Действие стало активным (какие-то его составляющие могли быть или были выполнены), но затем было временно приостановлено. Никакие его составляющие не будут выполняться, пока действие не будет выведено из этого состояния
1	Leaf	nullified	nullified (аннулировано)	Экземпляр класса Act был создан по ошибке, был «удален» и трактуется, как если бы никогда не существовал. Информация о нем сохраняется только для целей аудита
1	Leaf	obsolete	obsolete (устарело)	Данный экземпляр класса Act заменен другим экземпляром

Таблица 8 контролирует значения структурных элементов эталонной информационной модели HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 8 — Домен ContextControl (используется в атрибуте ActRelationship.contextControlCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	A: ContextControl Additive			Ассоциация, которая вносит дополнение к существующему контексту экземпляра класса Act. Эта ассоциация и все ассоциации, распространенные от родительского экземпляра класса Act, будут интерпретироваться, как относящиеся к данному экземпляру
2	Leaf	AN	additive, non-propagating (аддитивная, не распространяемая)	Ассоциация, которая вносит дополнение к существующему контексту экземпляра класса Act, но не распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение AN (additive, non-propagating — аддитивная, не распространяемая). Это означает, что автор, описываемый этим экземпляром, будет добавлен к совокупности авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act. Однако в контекст его дочерних экземпляров, разрешающих распространение контекста, будут добавлены только унаследованные авторы
2	Leaf	AP	additive, propagating (аддитивная, распространяемая)	Ассоциация, которая вносит дополнение к существующему контексту экземпляра класса Act и распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение AP (additive, propagating — аддитивная, распространяемая). Это означает, что автор, описываемый этим экземпляром, будет добавлен к совокупности авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act, и вместе с унаследованными авторами будет распространяться в контекст его дочерних экземпляров, разрешающих распространение контекста
1	A: ContextControlNon Propagating			Ассоциация применяется только к данному экземпляру класса Act и не распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd)

Продолжение таблицы 8

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	AN	additive, non-propagating (аддитивная, не распространяемая)	Ассоциация, которая вносит дополнение к существующему контексту экземпляра класса Act, но не распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение AN (additive, non-propagating — аддитивная, не распространяемая). Это означает, что автор, описываемый этим экземпляром, будет добавлен к совокупности авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act. Однако в контекст его дочерних экземпляров, разрешающих распространение контекста, будут добавлены только унаследованные авторы
2	Leaf	ON	overriding, non-propagating (замещающая, не распространяемая)	Ассоциация, добавляемая к существующему контексту экземпляра класса Act и замещающая ассоциации с тем же или более специфичным значением атрибута typeCode. Однако эта замещающая ассоциация не распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение ON (overriding, non-propagating — замещающая, не распространяемая). Это означает, что автор, описываемый этим экземпляром, заменит всю совокупность авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act. Далее, в контекст его дочерних экземпляров, разрешающих распространение контекста, не будет добавлен ни один автор
2	Leaf	OP	overriding, propagating (замещающая, распространяемая)	Ассоциация, добавляемая к существующему контексту экземпляра класса Act и замещающая ассоциации с тем же или более специфичным значением атрибута typeCode. Эта замещающая ассоциация будет распространяться на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение OP (overriding, propagating — замещающая, распространяемая). Это означает, что автор, описываемый этим экземпляром, заменит всю совокупность

Окончание таблицы 8

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act. Далее, в контекст его дочерних экземпляров, разрешающих распространение контекста, будет добавлен только этот автор
1	A: ContextControl Propagating			Ассоциация распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd)
2	Leaf	AP	additive, propagating (аддитивная, распространяемая)	Ассоциация, которая вносит дополнение к существующему контексту экземпляра класса Act и распространяется на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение AP (additive, propagating — аддитивная, распространяемая). Это означает, что автор, описываемый этим экземпляром, будет добавлен к совокупности авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act, и вместе с унаследованными авторами будет распространяться в контекст его дочерних экземпляров, разрешающих распространение контекста
2	Leaf	OP	overriding, propagating (замещающая, распространяемая)	Ассоциация, добавляемая к существующему контексту экземпляра класса Act и замещающая ассоциации с тем же или более специфичным значением атрибута typeCode. Эта замещающая ассоциация будет распространяться на действия-потомки, которые связаны с данным экземпляром класса Act с помощью экземпляра класса ActRelationship (см. описание атрибута ActRelationship.contextConductionInd). Примеры: пусть у экземпляра класса Participation атрибут typeCode имеет значение AUT (author — автор), а атрибут contextControlCode имеет значение OP (overriding, propagating — замещающая, распространяемая). Это означает, что автор, описываемый этим экземпляром, заменит всю совокупность авторов, которая была распространена от родительских экземпляров класса Act в контекст данного экземпляра класса Act. Далее, в контекст его дочерних экземпляров, разрешающих распространение контекста, будет добавлен только этот автор

Таблица 9 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 9 — Домен EntityClass (используется в атрибуте Entity.classCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: EntityClassRoot	ENT	entity (сущность)	(См. также определение класса Entity.) Соответствует классу Entity
2	A: EntityClass Document Receiving			
3	A: EntityClassPerson OrOrgReceiving			
4	S: EntityClass Organization	ORG	organization (организация)	(См. также определение класса Entity.) Социальная структура или юридическое лицо, созданное людьми
5	Leaf	PUB	public institution (публичный институт)	Учреждение, созданное определенным сообществом, нередко обладающее определенными полномочиями в некоторой сфере. Примерами служат правительства, государственные органы, ассоциации
5	Leaf	STATE	state (сообщество)	Политически организованное сообщество, ограниченное территориальными, культурными или этническими рамками, получившее суверенитет (до определенной степени) от других сообществ (охватывающих или соседних). К таким сообществам относятся страны (нации), провинции (например, штат в США, департамент во Франции), графства или муниципальные образования. Может быть проведена дальнейшая детализация, например, используя коды стран ISO, коды штатов FIPS-PUB и т. д.
4	Leaf	PSN	person (лицо)	(См. также определение класса Person.) Живой организм вида homo sapiens
3	Leaf	HCE	health chart entity (медицинская карта)	Медицинская карта, рассматриваемая как хранилище поступающих медицинских записей
2	S: EntityClassLiving Subject	LIV	living subject (живой организм)	(См. также определение класса LivingSubject.) Все сущее, имеющее свойство жизни, независимо от текущего состояния (труп человека все еще рассматривается как живой организм)
3	S: EntityClassNon PersonLiving Subject	NLIV	non-person living subject	
4	Leaf	ANM	animal (животное)	Живой организм царства животных

Продолжение таблицы 9

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
4	Leaf	MIC	microorganism (микроорганизм)	Все одноклеточные живые организмы, включая простейшие, бактерии, дрожжи, вирусы и т. д.
4	Leaf	PLNT	plant (растение)	Живой организм из порядка растений
3	Leaf	PSN	person (лицо)	(См. также определение класса Person.) Живой организм вида homo sapiens
2	S: EntityClassMaterial	MAT	material (материал)	(См. также определение класса Material.) Любой предмет, имеющий протяженность и массу. Может принадлежать к живой или к неживой природе
3	S: EntityClass Manufactured Material	MMAT	manufactured material (изготовленный материал)	(См. также определение класса ManufacturedMaterial.) Соответствует классу ManufacturedMaterial
4	S: EntityClass Container	CONT	container (контейнер)	(См. также определение класса Container.) Контейнер для других сущностей
5	Leaf	HOLD	holder (штатив)	Разновидность контейнера, предназначенная для помещения в нее других контейнеров или других штативов
4	S: EntityClassDevice	DEV	device (устройство)	(См. также определение класса Device)
5	Leaf	CER	certificate representation (представление сертификата)	Физический артефакт, хранящий информацию о наделении полномочиями
5	Leaf	MODDV	imaging modality (тип устройства лучевой диагностики)	Класс, содержащий уникальные атрибуты устройства лучевой диагностики
3	Leaf	CHEM	chemical substance (химическая субстанция)	Субстанция, полностью определенная формулой органической или неорганической химии, включая смеси других химических субстанций. Может быть сделана дальнейшая детализация, например, с помощью кодов IUPAC
3	Leaf	FOOD	food (пища)	Натуральные, обработанные или изготовленные сущности, которые, в основном, используются как пища людей и животных
2	S: EntityClass Organization	ORG	organization (организация)	(См. также определение класса Entity.) Социальная структура или юридическое лицо, созданное людьми

Окончание таблицы 9

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
3	Leaf	PUB	public institution (публичный институт)	Учреждение, созданное определенным сообществом, нередко обладающее определенными полномочиями в некоторой сфере. Примерами служат правительства, государственные органы, ассоциации
3	Leaf	STATE	state (сообщество)	Политически организованное сообщество, ограниченное территориальными, культурными или этническими рамками, получившее суверенитет (до определенной степени) от других сообществ (охватывающих или соседних). К таким сообществам относятся страны (нации), провинции (например, штат в США, департамент во Франции), графства или муниципальные образования. Может быть проведена дальнейшая детализация, например, используя коды стран ISO, коды штатов FIPS-PUB и т. д.
2	S: EntityClassPlace	PLC	place (место)	(См. также определение класса Place.) Физическое место или участок с содержащимися в них структурами, если таковые имеются. Место может быть естественным или созданным человеком. Географическое положение места может быть непостоянным
3	Leaf	CITY	city or town (город или населенный пункт)	Территория города, населенного пункта или другого муниципального образования
3	Leaf	COUNTRY	country (страна)	Суверенная национальная территория
3	Leaf	COUNTY	county or parish (графство или округ)	Территория графства, округа или другого административного деления штата или провинции
3	Leaf	PROVINCE	state or province (штат или провинция)	Территория штата, провинции, департамента или другой административной территории суверенного государства
2	Leaf	RGRP	group (группа)	Группа ресурсов (персонал, материал, помещения), планируемая в расписании как единое целое. Может быть пулом однотипных ресурсов, бригадой или сочетанием персонала, материала и помещений

Таблица 10 контролирует значения структурных элементов эталонной информационной модели HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 10 — Домен EntityDeterminer (используется в атрибуте Entity.determinerCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: EntityDeterminer Determined	KIND	described (описывает)	Определитель KIND указывает, что данный экземпляр класса Entity представляет общее описание вида сущностей, которые могут рассматриваться в целом, по частям или по отдельным элементам

Окончание таблицы 10

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	QUANTIFIED_KIND	described quantified (описывает перечислимую группу)	Определитель QUANTIFIED_KIND указывает, что данный экземпляр класса Entity представляет общее описание перечислимой группы сущностей. Например, если экземпляр класса Entity имеет атрибут code со значением «шприц» и при этом атрибут determinerCode имеет значение QUANTIFIED_KIND, а атрибут quantity имеет значение 3, то этот экземпляр класса Entity описывает набор из трех шприцев
2	Leaf	INSTANCE	specific (специфичный)	Определитель INSTANCE указывает, что данный экземпляр класса Entity представляет ровно один объект. Например, если экземпляр класса Entity имеет атрибут code со значением «человек» и при этом атрибут determinerCode имеет значение INSTANCE, а атрибут quantity имеет значение 1, то этот экземпляр класса Entity описывает ровно одного человека

Таблица 11 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 11 — Домен EntityStatus (используется в атрибуте Entity.statusCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: EntityStatusNormal	normal	normal (нормальное)	«Типичное» состояние. Исключает состояние nullified, представляющее терминальное состояние экземпляра класса Entity, созданного по ошибке
2	Leaf	active	active (активно)	Это состояние отражает тот факт, что экземпляр класса Entity в настоящее время активен
2	Leaf	terminated	terminated (завершено)	Это состояние представляет нормальное завершение экземпляра класса Entity
1	Leaf	nullified	nullified (аннулировано)	Это состояние представляет завершение экземпляра класса Entity, созданного по ошибке

Таблица 12 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 12 — Домен ManagedParticipationStatus (используется в атрибуте ManagedParticipation.statusCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: Managed Participation StatusNormal	normal	normal (нормальное)	«Типичное» состояние. Исключает состояние nullified, представляющее терминальное состояние экземпляра класса ManagedParticipation, созданного по ошибке

Окончание таблицы 12

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	active	active (активно)	Это состояние отражает тот факт, что участие продолжается
2	Leaf	cancelled	cancelled (отменено)	Терминальное состояние, означающее, что участие было отменено до того, как стало активным
2	Leaf	completed	completed (завершено)	Терминальное состояние, означающее, что участие успешно завершено
2	Leaf	pending	pending (готовящееся)	Это состояние отражает тот факт, что участие еще не стало активным
1	Leaf	nullified	nullified (аннулировано)	Это состояние является терминальным состоянием экземпляра участия, созданного по ошибке

Таблица 13 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 13 — Домен ParticipationType (используется в атрибуте Participation.typeCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	A: Participation Ancillary			Коды, относящиеся в основном к участию, а не к действию. Направляющие, принимающие и выписывающие врачи должны быть теми же, что являются авторами соответствующих событий, информация о которых передается в экземплярах класса ControlAct
2	Leaf	ADM	admitter (врач приемного отделения)	Врач приемного отделения, отвечающий за поступление пациента в стационар
2	Leaf	ATND	attender (лечащий врач)	Врач, отвечающий за лечение пациента во время его пребывания в стационаре
2	Leaf	CON	consultant (консультант)	Врач, оценивающий состояние пациента и дающий рекомендации лечащему врачу
2	Leaf	DIS	discharger (выписывающий врач)	Врач, отвечающий за выписку пациента из стационара
2	Leaf	ESC	escort (сопровождающее лицо)	Только для услуг перевозки пациента. Лицо, сопровождающее пациента
2	Leaf	REF	referrer (направляющий врач)	Лицо, направляющее пациента к другому врачу для оказания медицинской помощи. Обычно направляющий врач получает отчет об оказанной помощи от принимающего врача

Продолжение таблицы 13

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: Participation IndirectTarget	IND	indirect target (косвенная целевая сущность)	Целевая сущность, роль которой в действии невелика и которая непосредственно не затрагивается действием, но при этом эта сущность должна быть упомянута в записи или в документе, регистрирующем это действие
2	Leaf	BEN	beneficiary (выгодоприобретатель)	Целевая сущность, с ведома которой совершается действие, но сама она может не принимать непосредственного участия в этом действии. Может быть указана вместе с непосредственной целью, чтобы отметить, что обе участвуют в действии. Примером служит выгодоприобретатель данного действия, например, застрахованное лицо
2	Leaf	COV	coverage target (целевое лицо страхового покрытия)	Целевое участие лица в действии страхового покрытия медицинской помощи, в котором целевая роль принадлежит или владельцу страхового полиса, или лицу, застрахованному по этому полису
2	Leaf	HLD	holder (владелец)	Участник, владеющий финансовым инструментом, например, контрактом (страховым полисом), обычно в силу некоторого соглашения с автором
2	Leaf	RCT	record target (целевая медицинская карта)	Целевой медицинской картой является та, которая содержит документацию о данном действии. Это особенно важно, если сам пациент не является субъектом услуги
2	Leaf	RCV	receiver (получатель)	Лицо (или организация), получающее результат действия
1	A: Participation Information Generator			Стороны, которые могут, должны вносить или внесли свой вклад в содержание информации о действии. Таким вкладом может быть информация, которая привела к решению о выполнении действия и о способе его выполнения (например, информация, полученная от консультанта), информация, которая должна быть выявлена в данном действии (например, сведения об истории заболевания, полученные от информатора), а также информация о том, что действие выполнено (например, сведения от свидетеля действия)
2	Leaf	AUT	author (originator) (автор или инициатор)	Сторона, которая инициирует выполнение действия и, следовательно, несет ответственность за информацию об этом действии и является ее обладателем. Примеры: составитель протокола исследования, лицо, составляющее описание действия, автор методических рекомендаций, лицо, составившее направление на исследование, кардиограф (устройство), снявший ЭКГ и т. д. Каждое действие должно иметь автора. Независимо от наклонения действия авторство всегда фактическое. Автор (или авторы) является обладателем действий, которые они инициировали. Это означает, что сторона, не являющаяся данным автором (или

Продолжение таблицы 13

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
				данными авторами), не может отменить, прекратить, завершить или каким-либо иным способом модифицировать состояние или содержание данного действия. Сторона, не являющаяся автором, может лишь выполнить другое действие, которое исправляет, обращает, перекрывает, заменяет или иным образом изменяет результаты данного действия, которое остается неизменным и приобретает связь с действием, выполненным этой стороной
2	Leaf	ENT	data entry person (оператор по вводу данных)	Лицо, обеспечившее ввод данных в исходную систему. Регистрация оператора по вводу данных не обязательна и осуществляется для внутренних целей контроля качества информации. Примером служит оператор, который ввёл данные в информационную систему с диктофонной записи
2	Leaf	INF	informant (информатор)	Источник информации, включенной в содержание действия (например, близкое лицо, отвечающее на вопросы об истории заболевания пациента). При сборе сведений об истории заболевания информатором логически является пациент, однако лицо, отвечающее на такие вопросы, является косвенным субъектом медицинской помощи
2	Leaf	WIT	witness (свидетель)	Используется только для событий оказания медицинской помощи. Лицо, являющееся свидетелем действия, не принимает в нем участия. Свидетель не обязательно заинтересован в этом действии и в еще меньшей степени выказывает свое отношение к этому действию (например, одобряет). Примерами свидетелей служат студент, наблюдающий за ходом операции, или лицо, свидетельствующее подписание упреждающих указаний
1	A: ParticipationInformationRecipient			Сторона, которая может или должна получить либо получила информацию о содержании данного действия либо дальнейшую или производную информацию об этом действии. Получатель информации инертен, т. е. не зависит от наклонения действия. Обоснование: это понятие является обобщением весьма разнородного семейства понятий, поэтому определение не может быть более конкретным. Это понятие является абстрактным и вместо него должна использоваться одна из специализаций
2	Leaf	NOT	urgent notification contact (экстренный контакт)	Получатель информации, который должен быть уведомлен о возникновении чрезвычайной ситуации в данном действии (например, если лабораторный анализ имеет критичные результаты, то получателем является контактное лицо, кому надо об этом позвонить; другим примером служит близкое лицо, которое надо уведомить о критическом состоянии госпитализированного пациента)

Продолжение таблицы 13

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	Leaf	PRCP	primary information recipient (основной получатель)	Получатель, которому в основном предназначена информация о действии, например, участковый терапевт, получающий выписной эпикриз стационара, санитарно-эпидемиологическая служба, получающая информацию о подозрении на инфекционное заболевание. У одного и того же действия может быть несколько таких участников, не подразделяемых по приоритету, например, первый, второй
2	Leaf	TRC	tracker (получатель копии)	Получатель копии информации, например, участковый терапевт, получающий копию результата исследования, выполненного по направлению специалиста
1	S: Participation PhysicalPerformer	PRF	performer (исполнитель)	Лицо, которое действительно выполнило действие, в основном, своими усилиями. На нем не обязательно лежит полная ответственность за это действие, например, хирург-ординатор выполнил операцию под присмотром ведущего хирурга. Этим лицом может быть и пациент, когда прокалывает палец для измерения сахара крови
2	Leaf	DIST	distributer (распределитель)	Лицо, распределяющее материал, используемый или созданный при данном действии
2	Leaf	PPRF	primary performer (основной исполнитель)	Главный или основной исполнитель действия
2	Leaf	SPRF	secondary performer (ассистент)	Лицо, существенно содействующее выполнению действия своим присутствием и помощью. Примерами служат ассистенты, техники, помощники и другие аналогичные должности
1	S: Participation TargetDirect	DIR	direct target (непосредственная цель)	Целевая сущность, существенным образом участвующая в действии и непосредственно изменяемая этим действием (включая расходные материалы, устройства и т. д.)
2	S: Participation Consumable	CSM	consumable (расходный материал)	Целевая сущность, которая берется для выполнения действия, расходуется и исчезает в процессе его выполнения
3	Leaf	TPA	therapeutic agent (терапевтический агент)	Нечто, применяемое к субъекту терапевтической услуги для достижения физиологического эффекта (например, заживление, облегчение, провоцирование реакции и т. д.). При применении лекарства терапевтический агент является расходным материалом, при изготовлении или отпуске лекарства — товаром. Таким образом, роль расходного материала или товара должна соответствовать виду услуги

Продолжение таблицы 13

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
2	S: Participation TargetDevice	DEV	device (устройство)	(См. также определение класса Device.) Нечто, используемое при оказании услуги и существенно не изменяющееся в процессе ее оказания (т.е. прочное или инертное по отношению к данной услуге). Примером служат устройства мониторинга, инструменты, а также устройства доступа в тело, дренаж, протезы, водитель ритма и т. д.
3	Leaf	NRD	non-reusable device (изделие, повторно не используемое)	Изделие, меняющее владельца при оказании услуги, например, водитель ритма, протез, устройство для инъекции инсулина (шприц-ручка) и т. д. После выполнения услуги может понадобиться пополнение склада таких изделий
3	Leaf	RDV	reusable device (повторно используемое изделие)	Изделие, не меняющее владельца при оказании услуги (например, хирургический инструмент, иной инструмент, эндоскоп). Различие между повторно используемым и повторно не используемым изделием должно проводиться, чтобы знать, надо ли пополнять склад изделий
2	Leaf	BBY	baby (ребенок)	Ребенок (при родовспоможении)
2	Leaf	DON	donor (донор)	При некоторых услугах трансплантации органов и изредка при переливании крови донор может быть целевым участником услуги. Однако в большинстве случаев трансплантация подразделяется на три услуги: изъятие органа, транспортировка и имплантация. Для многих услуг изъятия органов или имплантации идентичность донора или реципиента не существенна
2	Leaf	PRD	product (продукт)	Целевой материал, который предоставляется (производится) при выполнении услуги (например, биоматериал при услуге сбора биоматериала, устройство доступа или дренажа при выполнении услуги помещения в тело, упаковка лекарства при услуге отпуска из аптеки. Не имеет значения, существовал ли материал до услуги или появился в процессе ее выполнения (например, при услуге поставки материал берется со склада)
2	Leaf	SBJ	subject (субъект)	Основная целевая сущность, для которой оказывается услуга. Например, пациент в физикальном исследовании, биоматериал при лабораторном анализе. Целевой сущностью может быть член семьи пациента (санитарное просвещение), устройство или помещение (уборка, дезинфекция, ведение хозяйства). Примечание — Не все непосредственные цели являются субъектами. Например, расходные материалы или устройства, используемые как средство выполнения действия, субъектами не являются. Однако устройство может быть субъектом технического обслуживания

Продолжение таблицы 13

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: Participation TargetLocation	LOC	location (место)	Место оказания медицинской помощи. Может быть неподвижным (здание или кабинет в нем) или передвижным (например, машина скорой помощи, вертолет, самолет, поезд, грузовик, судно и т. д.)
2	Leaf	DST	destination (место назначения)	Место назначения для услуги. Может быть неподвижным (здание или кабинет в нем) или передвижным (например, судно)
2	Leaf	ELOC	entry location (место ввода)	Место ввода данных о действии
2	Leaf	ORG	origin (место начала)	Место начала услуг. Может быть неподвижным (здание или кабинет в нем) или передвижным (например, судно)
2	Leaf	RML	remote (дистанционно удаленное место)	Некоторые услуги могут осуществляться одновременно в нескольких местах (например, телемедицинская или телефонная консультация). Место, где находится основной исполнитель услуги, считается основным местом услуги (код LOC), а остальные места рассматриваются как дистанционно удаленные
2	Leaf	VIA	via (через)	Для услуг — промежуточное место на пути от начального пункта до места назначения
1	S: Participation Verifier	VRF	verifier (контролер)	Лицо, которое проверяет правильность и пригодность услуги (плана, направления, события и т. д.) и тем самым обеспечивает подотчетность
2	Leaf	AUTHEN	authenticator (аутентификатор)	Контролер, который заверяет точность описания действия, но не наделен этим правом юридически. Примером служит ординатор, который осматривает пациента, диктует дневниковую запись, а затем подписывает ее распечатку. Его подпись заверяет аутентичность записи, но содержание записи не имеет юридического значения
2	Leaf	LA	legal authenticator (юридический аутентификатор)	Контролер, который юридически заверяет точность описания действия. Примером служит врач-специалист, который осматривает пациента, диктует дневниковую запись, а затем подписывает ее распечатку. Его подпись заверяет аутентичность записи, содержание которой может быть использовано в юридических актах
1	Leaf	CST	custodian (ответственный)	Лицо (или организация), которое несет юридическую ответственность за составление и хранение информации о данном объекте услуги (например, отвечает за составление отчета или за ведение каталога товаров и т. д.)

Окончание таблицы 13

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	Leaf	RESP	responsible party (ответственная сторона)	Поставщик медицинской помощи (физическое лицо или организация), на которого возложена основная ответственность за действие. Ответственный поставщик не обязательно принимает непосредственное участие в действии, но в этом случае отчитывается за делегирование своих полномочий и обязан проконтролировать результат выполненного действия (например, заведующий биохимической лабораторией)

Таблица 14 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 14 — Домен RelationshipConjunction (используется в атрибуте ActRelationship.conjunctionCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	Leaf	AND	and (и)	Каждое условие, соединенное союзом AND, должно быть истинным
1	Leaf	OR	or (или)	Хотя бы одно из условий, соединенных союзами OR, должно быть истинным
1	Leaf	XOR	exclusive or (исключающее или)	Одно и только одно из условий, соединенных союзами XOR, должно быть истинным

Таблица 15 содержит коды иерархии класса Role. Значения, принадлежащие этой иерархии, описывают роль, представляющую собой ассоциацию или связь двух сущностей — исполнителя роли и контролера роли. Имена ролей произведены от имен исполнителей роли.

Иерархия ролей образована из трех базовых понятий, или абстрактных доменов:

- в абстрактный домен RoleClassOntological входят роли исполнителя, определенные или заданные контролером;

- в абстрактный домен RoleClassPartitive входят роли исполнителя, который в каком-то смысле является «частью» контролера;

- в абстрактный домен RoleClassAssociative включены все остальные формы ассоциаций между исполнителем и контролером. Эта совокупность ролей подразделяется дальше на следующие группы:

- домен RoleClassPassive включает в себя роли, исполнитель которых используется, известен, лечится, обрабатывается, строится, уничтожается и т. п. под надзором контролера. Исполнитель таких ролей пассивен в том смысле, что роль существует без соглашения с ее исполнителем;

- домен RoleClassMutualRelationship включает в себя отношения, основанные на взаимном поведении двух сущностей. Такое отношение может быть построено на формальном соглашении сторон либо существовать де-факто. Этот поддомен подразделяется дальше на следующие группы:

- домен RoleClassRelationshipFormal содержит коды формальных отношений, часто возникающих по контракту или соглашению сторон;

- личные отношения, характеризующие личные связи двух лиц.

Описанная иерархия представлена в таблице 15 как совокупность абстрактных доменов. Исключения составляют «личные отношения», которые являются листовым понятием.

Таблица 15 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 15 — Домен RoleClass (используется в атрибуте Role.classCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: RoleClassRoot	ROL	role (роль)	(См. также определение класса Role.) Соответствует классу Role
2	A: RoleClass Associative			Общая ассоциация между двумя сущностями, которая не является ни отношением целое-часть, ни онтологией.
3	A: RoleClassMutual Relationship			Отношение, основанное на взаимном поведении двух связанных сущностей. Основой такого отношения могут быть соглашения (например, брачный союз, двусторонний договор) или произвольные связи (например, участие сторон в диспуте, братья или сестры, дети)
4	A: RoleClass RelationshipFormal			Официально признаваемые отношения двух сторон, часто по контракту или аналогичному соглашению
5	S: LicensedEntityRole	LIC	licensed entity (лицензированная сущность)	(См. также определение класса LicensedEntity.) Отношение, в котором контролер сертифицирует право исполнителя (например, медицинского работника, медицинского устройства или медицинской организации) на выполнение определенной деятельности, находящейся под юрисдикцией контролера (например, орган управления здравоохранением лицензирует медицинскую организацию, квалификационная комиссия выдает специалисту сертификат)
6	Leaf	DSDLOC	dedicated service delivery location (выделенное место оказания медицинской помощи)	Место (исполнитель роли), в котором разрешено оказание медицинской помощи. Контролером является орган санитарно-эпидемиологического надзора
6	Leaf	NOT	notary public (нотариус)	
6	Leaf	PROV	healthcare provider (поставщик медицинской помощи)	Сущность (исполнитель роли), получившая от уполномоченного органа (контролера) разрешение на право ведения медицинской деятельности
5	S: RoleClassAgent	AGNT	agent (агент)	Сущность (исполнитель роли), которая действует или уполномочена действовать от имени другой сущности (контролера)

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
6	S: RoleClass AssignedEntity	ASSIGNED	assigned entity (полномочный представитель)	Роль агента, в которой работник выступает как представитель организации. Акцент делается на функциональной роли представительства, в отличие от роли работника, в которой акцент делается на трудовые отношения работника и работодателя
7	S: RoleClassContact	CON	contact (контакт)	Лицо или организация (исполнитель роли), предоставляющая информацию другой сущности (контролеру) или получающая от нее информацию. Примеры: близкое лицо пациента и контакт со службой скорой помощи; контакт с гарантом оплаты медицинской помощи, контакт с работодателем
8	Leaf	ECON	emergency contact (контакт со службой скорой помощи)	Сущность, к которой надо обратиться при необходимости скорой помощи
8	Leaf	NOK	next of kin (близкое лицо)	Лицо, которое должно быть уведомлено в экстренной ситуации как близкое лицо данной сущности
7	Leaf	SGNOFF	signing authority or officer (руководитель или лицо с правом подписи)	Роль лица (исполнителя), являющегося руководителем организации (контролера) или имеющим право подписи от ее имени
6	Leaf	GUARD	guardian (постовая сестра)	Постовая сестра
5	S: RoleClassEmployee	EMP	employee (работник)	(См. также определение класса Employee.) Отношение между лицом или организацией и другим лицом или организацией, состоящее в выполнении работы за вознаграждение. Эта роль идентифицирует тип отношения между работником и работодателем, а не характер выполняемой работы. (В противоположность типу роли ASSIGNED — полномочный представитель)
6	Leaf	MIL	military person (военнослужащий)	Роль, выполняемая лицом, находящимся на военной службе. Контролером является военная организация (например, армия, военно-морской флот, военно-воздушные силы) или, более конкретно, воинская часть (например, отряд С, 3-й батальон, 4-й дивизион и т. д.)
5	Leaf	CIT	citizen (гражданин)	Гражданин аполитичной сущности

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
5	Leaf	COVPTY	covered party (застрахованное лицо)	Роль лица (исполнителя роли), получающего услуги на условиях конкретного страхового полиса. Страховщик этого полиса является контролером. Застрахованное лицо получает услуги в силу договорных или иных отношений со страхователем (владельцем полиса). Эта причина передается в атрибуте Role.code, а между экземпляром класса Role, описывающим роль застрахованного лица, и экземпляром класса Role, описывающим роль страхователя, устанавливается связь с помощью экземпляра класса RoleRelationship, у которого атрибут typeCode имеет значение INDAUTH (has indirect authority over — имеет косвенные полномочия над кем-либо), при этом роль владельца полиса является источником связи, а роль застрахованного лица — целью связи. Учтите, что конкретный полис может страховать несколько лиц, одним из которых может быть, а может и не быть владелец полиса. Поэтому роль застрахованного лица отделяется от роли владельца полиса
5	Leaf	CRINV	clinical research investigator (исследователь в клиническом испытании)	Роль поставщика медицинской помощи, обязательно физического лица, которое имеет полномочия от спонсора клинического испытания непосредственно руководить проведением клинического испытания или обрабатывать результаты испытания в интересах спонсора
5	Leaf	CRSPNSR	clinical research sponsor (спонсор клинического испытания)	Роль сущности, обычно организации, являющейся спонсором клинического исследования или испытания. Спонсор заказывает испытание, оплачивает издержки на его проведение и несет ответственность за выполнение всех юридических требований, касающихся безопасности субъекта испытания и конфиденциальности его персональных данных. Кроме того, спонсор несет ответственность за сбор, хранение и анализ информации о клиническом испытании. В отдельном контролере нет необходимости, поскольку спонсор клинического испытания выполняет эту роль на свой страх и риск. Спонсорами клинических испытаний обычно являются образовательные или научные медицинские организации, государственные органы или биофармацевтические предприятия
5	Leaf	GUAR	guarantor (гарант)	Лицо или организация (исполнитель роли), являющиеся финансовым гарантом для другого лица или организации (контролера)
5	Leaf	PAT	patient (пациент)	(См. также определение класса Patient.) Роль пациента контролируется поставщиком медицинской помощи

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
5	Leaf	PAYEE	payee (получатель оплаты)	Роль организации или лица, которое должно получить оплату по счету за оказанную медицинскую помощь. Контролером является организация, которая выставляет данный счет
5	Leaf	PAYOR	invoice payor (плательщик по счету)	Роль организации, которая берет на себя обязательства принимать счета, определять размер страхового покрытия или оплаты по этим счетам и перечислять оплату получателям, указанным в этих счетах. Эту роль может выполнять страховщик или третья сторона, уполномоченная страховщиком. Контролером является организация, осуществляющая страховое покрытие
5	Leaf	POLHOLD	policy holder (владелец полиса)	Роль, выполняемая сущностью, обычно физическим лицом, которое владеет страховым полисом. Страховщик этого полиса является контролером. Эквивалентными терминами являются страховщик и получатель полиса. Идентификатор полиса передается в атрибуте Role.id экземпляра класса Role, описывающего роль владельца полиса. Конкретный полис может страховать несколько лиц, одним из которых может быть, а может и не быть владелец полиса. Поэтому роль застрахованного лица отделяется от роли владельца полиса
5	Leaf	QUAL	qualified entity (квалифицированная сущность)	Сущность (исполнитель роли), имеющая определенную подготовку (опыт) или другие характеристики, которые позволяют рассматривать ее как подходящего исполнителя определенной деятельности. Контролером является организация, которая осуществила подготовку или присвоила квалификацию
5	Leaf	RESBJ	research subject (субъект исследования)	Живое существо, к которому применяются экспериментальные процедуры (в его интересах)
5	Leaf	SPNSR	sponsor (страхователь)	Роль сущности (обычно организации), которая является спонсором страховой программы. Страховщик этой программы является контролирующей сущностью. Примеры включают ситуацию, когда предприятие оплачивает страховую программу своим работникам, но индивидуальный страховой полис определяет договорные отношения между страхователем и работником. В общем случае в функции страхователя входят ведение переговоров со страховщиком, согласование условий страховой программы и передача списка лиц, которые могут стать владельцами полиса, обеспечивающего эту программу
5	Leaf	STD	student (учащийся)	Роль учащегося в образовательном учреждении, которое является контролирующей организацией

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
5	Leaf	UNDWRT	underwriter (страховщик)	Роль, выполняемая организацией, которая предлагает страховую программу или принимает фискальную ответственность за эту программу и за полисы страхования по этой программе
4	Leaf	PRS	personal relationship (личные отношения)	Личные отношения между двумя людьми. Характер отношений должен быть определен в атрибуте Role.code, значения которого должны быть взяты из словарного домена PersonalRelationshipRoleType. Значение этого атрибута определяет также, у кого роль исполнителя, а у кого — контролера
3	A: RoleClassPassive			Ассоциация к сущности (исполнителю роли), которая используется, известна, лечится, обрабатывается, строится, уничтожается и т.п. под надзором контролера. Исполнитель пассивен в такой роли (хотя может быть активным в других ролях) в том смысле, что все происходящее с ним в этой роли осуществляется без соглашения с ним
4	S: RoleClass Distributed Material	DST	distributed material (оптовый товар)	Товар (исполнитель роли), поставляемый оптовым торговцем (контролером), который служит посредником между производителем и покупателем (или розничным торговцем)
5	Leaf	RET	retailed material (розничный товар)	Товар (исполнитель роли), поставляемый розничным торговцем (контролером), который также может консультировать потенциальных покупателей
4	S: RoleClass Manufactured Product	MANU	manufactured product (изготовленный материал)	Контролируется производителем
5	Leaf	THER	therapeutic agent (терапевтический агент)	Изготовленный материал (исполнитель роли), обладающий терапевтическими свойствами. Производитель является контролирующей сущностью
4	S: RoleClassService DeliveryLocation	SDLOC	service delivery location (место оказания медицинской помощи)	Роль места, в котором может быть оказана медицинская помощь
5	Leaf	DSDLOC	dedicated service delivery location (выделенное место оказания медицинской помощи)	Место (исполнитель роли), в котором разрешено предоставление медицинской помощи. Контролером является организация, выдавшая разрешение

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
5	Leaf	ISDLOC	incidental service delivery location (произвольное место оказания медицинской помощи)	Роль места, в котором медицинская помощь может быть оказана без предварительного выделения места или разрешения на его использование
4	Leaf	ACCESS	access (доступ)	(См. также определение класса Access.) Роль материала или устройства, используемого для проникновения в другую сущность. Основным примером служат средства внутривенного введения (или другого внедрения в тело пациента), которые уже существуют и должны упоминаться в указаниях по применению лекарственных средств
4	Leaf	BIRTHPL	birthplace (место рождения)	Связывает экземпляр класса Place, в котором указано место рождения (исполнитель роли) с экземпляром класса LivingSubject, описывающим живой организм (контролирующую сущность)
4	Leaf	EXPR	exposed entity (контактная сущность)	Роль сущности, контактировавшей с лицом или животным, страдающим инфекционным заболеванием, или находившейся в месте, зараженном токсином. Контролирующей сущностью является источник заражения, и вполне вероятно, что исполнитель роли, т.е. лицо, имевшее с ним контакт, станет контролирующей сущностью по отношению к другому лицу. Другими словами, если лицо инфицировано кем-то, то оно, в свою очередь, может инфицировать другие лица. Контроль источников заражения и выявление контактировавших с ними лиц являются основной задачей санитарно-эпидемиологической службы, и в контексте общественного здоровья стороны, имевшие контакты с источниками заражения, называются «контакными лицами»
4	Leaf	HLD	held entity (предмет владения)	Сущность, находящаяся во владении другой сущности (контролера), которая владеет ею или использует ее, обычно по некоторому соглашению с ее собственником
4	Leaf	HLTHCHRT	health chart (таблица показателей здоровья)	Роль материала (исполнителя), являющегося физической таблицей (или графиком) показателей здоровья, принадлежащей организации (контролирующей сущности)
4	Leaf	IDENT	identified entity (идентифицированная сущность)	Роль, выполняемая сущностями и контролируемая сущностями, которые идентифицируют их для различных целей
4	Leaf	MNT	maintained entity (управляемая сущность)	Сущность (исполнитель роли), которая управляется другой сущностью (контролером). Это типичная роль прочного оборудования. На контролера возложена ответственность за правильное функционирование, качество и безопасность

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
4	Leaf	OWN	owned entity (предмет собственности)	Сущность (материал, исполнитель роли), которую некто (контролер) на законном основании называет своей собственностью. Это позволяет контролеру принимать решения о распоряжении этим материалом
4	Leaf	RGPR	regulated product (регулируемый материал)	Материал, обращение которого регулируется некоторой государственной организацией. Роль исполняется материалом и контролируется этой организацией. Обоснование: для обеспечения клона сущности, используемого для идентификации номера NDC (National Drug Code — национального кода лекарства), присвоенного данному лекарству
4	Leaf	TERR	territorial authority (территориальный орган)	Сущность, обычно организация (контролер), которая имеет определенные полномочия (юрисдикцию) на некоторой территории или в некотором регионе (исполнитель роли). Например, Региональный орган управления здравоохранением Калгари (Calgary Regional Health Authority) имеет определенные полномочия над Регионом 4 Альберты (Region 4 of Alberta), являющимся контролируемым местом
4	Leaf	WRTE	warranted product (гарантийный товар)	Роль товара (исполнителя), на который покупателю дается гарантия от продавца (контролирующая сущность), объявляющая, что товар надежен и свободен от известных недостатков, и что продавец починит или заменит дефектные детали, если дефект выявлен в течение определенного времени и при определенных условиях
2	A: RoleClass Ontological			Отношение, в котором контролирующая сущность определяет или указывает, что представляет собой сущность, выполняющая роль. Таким образом, указывается «бытие» (гр. (ontos) сущее) исполнителя роли
3	S: RoleClassIs SpeciesEntity	GEN	has generalization (имеет генерализацию)	Связывает специализированное материальное понятие (исполнителя роли) с его обобщением (контролирующей сущностью)
4	Leaf	GRIC	has generic (имеет непатентованный синоним)	Специальный вид связи между лекарственными препаратами, означающий, что цель связи (контролирующая сущность) является непатентованным синонимом источника связи (исполнителя роли)
3	Leaf	INST	instance (экземпляр)	Отдельный материал (исполнитель роли), являющийся экземпляром класса материалов (контролера)

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
3	Leaf	SUBS	subsumer (охватывающее понятие)	Сущность, которая охватывает идентичность другой сущности. Используется в контексте объединения документированных экземпляров сущностей. Как исполнитель роли, так и контролирующая сущность должны иметь одно и то же значение атрибута classCode
2	A: RoleClassPartitive			Ассоциация между двумя сущностями, в которой сущность, выполняющая роль, является в некотором смысле «частью» сущности, контролирующей роль, например, членом, компонентом, ингредиентом или содержанием. Быть «частью» в самом широком смысле этого слова может означать что угодно от интегрального структурного компонента до более произвольной временной ассоциации сущности, исполняющей роль, с (обычно более объемлющей) сущностью, контролирующей роль
3	S: RoleClass IngredientEntity	INGR	ingredient (ингредиент)	Связывает компонент (исполнитель роли) с микстурой (контролирующей сущностью). Например, глюкоза и вода являются ингредиентами 5 % раствора глюкозы (D5W), латекс может быть ингредиентом трахеальной трубки
4	S: RoleClass InactiveIngredient	IACT	inactive ingredient (неактивный ингредиент)	
5	Leaf	COLR	color additive (краситель)	Субстанция (исполнитель роли), влияющая на оптические свойства материала (контролирующей сущности)
5	Leaf	FLVR	flavor additive (вкусовая добавка)	Субстанция (исполнитель роли), добавляемая к микстуре (контролирующей сущности) для придания определенного вкуса. Обычно используется в пище, но может использоваться при изготовлении лекарства, чтобы скрыть неприятный вкус активного ингредиента (что важно при лечении детей)
5	Leaf	PRSV	preservative (консервант)	Субстанция (исполнитель роли), добавляемая к микстуре (контролирующей сущности) для предотвращения ее порчи микроорганизмами (грибками, бактериями)
5	Leaf	STBL	stabilizer (стабилизатор)	Стабилизатор (исполнитель роли), добавляемый к микстуре (контролирующей сущности) для предотвращения молекулярной дезинтеграции основной субстанции
4	Leaf	ACTI	active ingredient (активный ингредиент)	Терапевтически активный ингредиент (исполнитель роли) микстуры (контролирующей сущности), которая обычно является готовой лекарственной формой
4	Leaf	ACTM	active moiety (активная часть)	

Продолжение таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
4	Leaf	ADTV	additive (добавка)	Ингредиент (исполнитель роли), добавляемый к основе (контролирующей сущности) и составляющий малую часть всей микстуры
4	Leaf	BASE	base (основа)	Основной ингредиент (исполнитель роли), составляющий большую часть микстуры. Например, вода в растворах для внутривенного введения или вазелин в мазях. Среди всех ингредиентов материала только один должен быть основой. Основная субстанция, в свою очередь, может быть микстурой
3	S: RoleClass LocatedEntity	LOCE	located entity (помещенная сущность)	Связывает сущность (исполнитель роли) с местом (контролирующей сущностью), в котором эта сущность каким-либо образом представлена. Эта связь может быть ограничена во времени
4	Leaf	STOR	stored entity (хранящаяся сущность)	Связывает сущность (исполнителя роли), например, устройство, с местом (контролирующей сущностью), в котором обычно ее можно найти или в котором она хранится, когда не используется
3	S: RoleClassSpecimen	SPEC	specimen (биоматериал)	Роль экземпляра класса Material, который содержит информацию о биоматериале, участвующем в определенном действии. Эта роль контролируется источником биоматериала
4	Leaf	ALQT	aliquot (аликвота)	Порция (исполнитель роли) исходного биоматериала или биоматериала-источника (контролирующей сущности), используемая для анализа или транспортировки
4	Leaf	ISLT	isolate (изолят)	Микроорганизм, изолированный от других микроорганизмов или от исходного матрикса
3	Leaf	CONT	content (содержимое)	Связывает материал (исполнителя роли) с содержащим его контейнером (контролирующей сущностью). В отличие от ингредиентов содержимое и контейнер остаются разделенными (не смешиваются) и содержимое может быть удалено из контейнера. Содержимое не является частью пустого контейнера
3	Leaf	MBR	member (член)	Роль, сущности, являющейся членом группы. Контролирующей сущностью для этой роли является группа. Среди других группировок выделятся группы, используемые в страховании (группа лиц, застрахованных по одному полису) и в планировании ресурсов, которые могут группироваться для целей планирования и логистики

Окончание таблицы 15

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
3	Leaf	PART	part (часть)	Ассоциация между двумя сущностями, в которой сущность, исполняющая роль, является в некотором смысле «частью» сущности, контролирующей роль, например, членом, компонентом, ингредиентом или содержанием. Быть «частью» в самом широком смысле этого слова может означать что угодно от интегрального структурного компонента до более произвольной временной ассоциации сущности, исполняющей роль, с (обычно более объемлющей) сущностью, контролирующей роль

Таблица 16 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 16 — Домен RoleLinkType (используется в атрибуте RoleLink.typeCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	Leaf	DIRAUTH	has direct authority over (имеет прямые полномочия управления кем-либо или чем-либо)	Роль-источник имеет прямые полномочия управления целевой ролью в иерархии подчинения
1	Leaf	INDAUTH	has indirect authority over (имеет косвенные полномочия управления кем-либо или чем-либо)	Роль-источник имеет косвенные полномочия управления целевой ролью в иерархии подчинения
1	Leaf	PART	has part (имеет частью)	Целевая роль является частью роли-источника
1	Leaf	REPL	replaces (заменяет)	Эта связь указывает, что роль-источник заменяет (или охватывает) целевую роль. С ее помощью можно задать новые идентификаторы и/или новое эффективное время регистрационной записи или сертификации и т. д.

Таблица 17 контролирует значения структурных элементов ЭИМ HL7. Поэтому она является нормативной частью утвержденной ЭИМ.

Таблица 17 — Домен RoleStatus (используется в атрибуте Role.statusCode ЭИМ)

Уровень	Тип, имя домена или мнемокод	Мнемокод	Отображаемое имя	Определение/описание
1	S: RoleStatusNormal	normal	normal (нормальное)	«Типичное» состояние. Исключает состояние nullified, которое указывает, что экземпляр роли был создан по ошибке
2	Leaf	active	active (активно)	Это состояние указывает, что сущность в настоящий момент активна в данной роли
2	Leaf	suspended	suspended (приостановлено)	Это состояние указывает, что выполнение роли сущностью, исполняющей эту роль, временно приостановлено. Переход в это состояние возможен из состояния active (активно)
2	Leaf	terminated	terminated (завершено)	Это состояние указывает, что выполнение роли успешно завершено
1	Leaf	nullified	nullified (аннулировано)	Это состояние является терминальным состоянием экземпляра роли, созданного по ошибке

Приложение А
(справочное)

История развития ЭИМ

Настоящее приложение является частью «введения» первичного, утвержденного стандарта ЭИМ HL7. Его содержание было перенесено в справочное приложение совместного документа ISO/HL7, чтобы оно не создавало путаницу, но при этом сохраняло текст, используемый комитетом HL7.

А.1 История ЭИМ

Развитие ЭИМ началось в апреле 1996 г. Первый выпуск ЭИМ был принят Техническим руководящим комитетом HL7 (Technical Steering Committee) на заседании рабочей группы в январе 1997 г. Этот выпуск был известен как проект ЭИМ версии 0.80.

На следующих двух заседаниях рабочей группы основное внимание уделялось ознакомлению с проектом ЭИМ и реализации процесса получения и согласования предложений по развитию модели. Процесс сопровождения ЭИМ стал известен как «гармонизация ЭИМ». Между 1998 г. и 2000 г. было девять заседаний, посвященных гармонизации, на которых рассматривались предложения по улучшению проекта ЭИМ. Кульминацией этой работы стала HL7 ЭИМ версии 1.0, обнародованная на заседании рабочей группы в январе 2001 г. На заседаниях по гармонизации основное внимание уделялось следующим пяти основным темам:

- гарантия охвата стандарта HL7 версии 2.х. Этот набор предложенных изменений привносил в проект модели все информационное содержание стандарта HL7 версии 2.х;
- устранение из модели содержания, не имеющего реализации. Этот набор предложенных изменений концентрировался на удалении той информации из проекта модели, которую проблемный технический комитет (steward technical committee) не инициировал и для сохранения которой в стандарте не мог найти достаточных оснований;
- объединенная модель обслуживания. Этот набор предложенных изменений представлял краткий, точно определенный набор структур и словарей, отвечавший информационным потребностям широкого круга клинических сценариев. Эта коллекция предложений, известная как USAM (Unified service action model), явилась плодом объединенных усилий ряда технических комитетов;
- обеспечения качества. Этот набор предложенных изменений был предназначен для устранения несогласованностей проекта модели и конфликтов между моделью и руководством по стилю моделирования. Он положил начало практике записи и отслеживания открытых проблем модели;
- внимание к «левой части» модели. Этот набор предложенных изменений был предназначен для определения развитых структур и словарей, рассчитанных на неклинические части модели (управление движением пациентов, финансы, ведение расписаний). Как и объединенная модель обслуживания, эти предложения явились плодом усилий ряда технических комитетов.

Процесс гармонизации и темы особого внимания позволили получить ЭИМ, которая является надежной, ясной, всеобъемлющей, лаконичной и непротиворечивой. Ее структура является и гибкой, и расширяемой, а ее содержание может быть достаточно легко отображено на стандарт передачи медицинских данных HL7 2.х и другие широко используемые спецификации структур данных в здравоохранении.

А.2 Значение утверждения ЭИМ

Утверждение ЭИМ в соответствии с принятыми процедурами голосования позволило вывести эту модель за рамки внутренних артефактов комитета HL7. Как утвержденный стандарт она будет представлена для принятия в качестве стандарта ANSI, а затем, возможно, для принятия в качестве стандарта ИСО. Другие организации-разработчики стандартов, аккредитованные в ANSI или ИСО, в том числе организация ASC X12 и ТК 251 «Медицинская информатика» Европейского комитета по стандартизации (CEN TC 251), выразили интерес в использовании ЭИМ или в ссылках на ЭИМ при разработке своих собственных стандартов. Появление ЭИМ в качестве стандарта значительно облегчает ее использование другими организациями-разработчиками стандартов (SDO — standards development organizations). Наличие общей эталонной информационной модели, используемой разными разработчиками стандартов информатизации здоровья, сулит значительные выгоды.

Утверждение ЭИМ имеет большое значение для местной стандартизации, осуществляемой национальными филиалами комитета HL7. Чтобы филиалы могли разрабатывать местные расширения стандартов HL7 в соответствии с процессом разработки стандарта HL7 Версии 3, сама ЭИМ должна быть стандартом. Это нередко требуется и местными юридическими нормами, например, местные стандарты должны быть расширениями уже принятых общих стандартов.

При проведении мероприятий, в которых применимы стандарты, органы государственного управления на федеральном уровне, уровне штата и муниципальном уровне в первую очередь рассматривают аккредитованные национальные стандарты. Поскольку ЭИМ стандартизует информационное содержание стандартов интероперабельности, разрабатываемых комитетом HL7, органы государственного управления США будут рассматривать ЭИМ в контексте решений, обеспечивающих интероперабельность информационных систем здравоохранения.

Приложение Б
(справочное)

Обзор принципов разработки ЭИМ

Б.1 Цель

Этот документ представляет собой краткий обзор базовых понятий и логических обоснований, положенных в основу разработки и развития ЭИМ HL7. Он не является руководством по применению ЭИМ, хотя в нем и представлено несколько частных примеров ее использования, полезных для объяснения фундаментальных понятий разработки и развития ЭИМ.

Б.2 Общие сведения

ЭИМ строится в целях согласованного совместного использования данных во многих «местных» контекстах. Вообще говоря, чем шире сфера интересов, тем важнее сделать явными все предположения, полагаемые в основу темы или области интереса. ЭИМ HL7 Версии 3 спроектирована как единая база и развитый источник всей информации, используемой в спецификации HL7. ЭИМ конкретно и однозначно формулирует точные определения понятий здравоохранения («предметы интереса»), используемые в сфере информационных систем здравоохранения, а также описывает отношения (они же «ассоциации») между этими понятиями.

Спецификации стандарта HL7 Версии 3 (например, сообщения HL7 Версии 3, структурированные документы и т. д.) позволяют взаимодействовать (например, обмениваться данными) слабо связанным информационным системам, используемым при разных условиях оказания медицинской помощи, предоставляемой разными поставщиками и на разных территориях. Поэтому область применения ЭИМ HL7 включает в себя всю информацию, которой должны обмениваться участвующие информационные системы здравоохранения в целях ее обработки. Кроме того, следует отметить, что ЭИМ не моделирует (и не должна моделировать) информацию, хранящуюся в конкретной информационной системе здравоохранения, но никогда не передаваемую другим системам.

ЭИМ представлена с помощью синтаксиса визуального моделирования, основанного на Унифицированном языке моделирования UML (Unified Modeling Language). (Определенные различия между «стандартным UML» и «UML HL7», использованным в ЭИМ, например, размещение названий ассоциаций, в настоящее время рассматриваются ТК по моделированию и методологии HL7 (Modeling and Methodology Technical Committee) в целях движения к максимальному согласованию обоих синтаксисов.) Комитет HL7 также ведет базу данных («хранилище ЭИМ»), содержащую детальную информацию о каждом понятии ЭИМ, каждом атрибуте и каждой ассоциации, включая логическое обоснование элемента, определение, ограничения и редактирование/изменение истории. В перспективе эта информация может быть формально опубликована (частично или целиком) в виде профиля UML, специфичного для стандарта HL7.

Важно отметить, что ЭИМ создается не для того, чтобы стать логической или физической моделью базы данных или иной структуры информационной системы конкретного производителя либо обслуживать интересы конкретной организации здравоохранения или сети организаций. В действительности ЭИМ даже не представляет конкретные группы сообщений HL7, она является обобщенной совокупностью данных и отношений между данными, из которых может быть сконструировано любое релевантное сообщение HL7. Ожидается, что по мере необходимости конкретные пользователи ЭИМ используют релевантные части ЭИМ, принимая их содержание при разработке и описании своих информационных моделей.

Б.3 Обоснование конструкции ЭИМ

Разветвленная структура ЭИМ основана на шести «базовых» классах: Act (действие), Entity (сущность), Role (роль), Participation (участие), ActRelationship (связь действий) и RoleLink (связь ролей). Определения этих классов приведены в разделе 1.4. Далее приводится обсуждение фундаментальных принципов, положенных в основу этих шести классов и их связей, а также базовые рекомендации по их использованию.

В ЭИМ HL7 выделены два основных «высокоуровневых» понятия, являющихся фундаментальными для понимания мира информации здравоохранения: намеренные «действия» или «услуги» (экземпляры класса Act), и «люди, места и предметы», представляющие интерес в мире здравоохранения (экземпляры класса Entity).

Класс Act (и его подклассы) представляет все намеренные действия, документируемые медицинским работником в клиническом или административном контексте. Появление класса Act в качестве одного из центральных классов ЭИМ отражает следующую позицию комитета HL7: с точки зрения информационного взаимодействия и обмена сообщениями оказание медицинской помощи состоит из последовательности атрибутированных намеренных действий. Таким образом, экземпляры класса Act описывают и клинические исследования (например, температуру пациента), и вмешательства (например, применение лекарств), и административные действия (например, госпитализацию пациента). Учтите, что с такой «действиецентричной» точки зрения на оказание медицинской помощи действие исследования имеет два очевидно противоположных значения: «действие установления и документирования конкретного факта» и «описание исследуемого предмета». Другими словами, экземп-

ляр класса Observation, являющегося специализацией класса Act, описывает как атрибутированное действие исследования, так и результаты исследования. Оба аспекта этого расширенного определения класса Observation отражены в специфических атрибутах класса Act или его подкласса Observation.

Класс Entity (и его подклассы) представляет все живые субъекты (например, люди, животные), организации (как формальные, так и неформальные), материалы (например, прочные и непрочные предметы, продукты, биоматериал, контейнеры) и места, которые могут представлять интерес в передаваемом контексте оказания медицинской помощи. Необходимо отметить, что понятие «коллекции информации» (например, медицинская карта), представляется не в виде экземпляра класса Entity или его подклассов, а моделируется как коллекция атрибутированных действий.

Между классами Act и Entity ЭИМ помещает два дополнительных класса — Role и Participation. Класс Role моделирует несколько важных понятий, превалирующих в предметной области оказания медицинской помощи. Во-первых, класс Role отражает тот факт, что в конкретном контексте медицинской помощи различные «статические» объекты могут «временно» выполнять одну или несколько «ролей» (например, пациент, врач общей практики, ответственная сторона, медицинская сестра и т. д.). Во-вторых, понятия «способности» (например, возможность оказания медицинской помощи в соответствии с рекомендациями Advanced Cardiac Life Support) и «сертификации» (например, диплом медицинской сестры) также могут моделироваться с помощью экземпляров класса Role. Наконец, тщательное изучение кратности (0..1) и имен (контролер, исполнитель) двух ассоциаций между классами Entity и Role показывает, что класс Role может использоваться для «группировки» экземпляров класса Entity.

Важно различать понятие сущности, имеющей определенную роль, от понятия «функциональной роли, выполняемой этой сущностью в контексте определенного действия», специфичной для конкретного действия. Функциональные роли моделируются с помощью экземпляров класса Participation. Например, анестезиолог-ординатор (сущность, имеющая роль) применяет анестезию (что моделируется с помощью экземпляра класса Participation, описывающего функциональную роль «поставщика» в действии «применить анестезию») к пациенту (что моделируется с помощью экземпляра класса Participation, описывающего функциональную роль «получателя» в действии «применить анестезию»). Учтите, что отсутствие прямой ассоциации между классами Participation и Entity вытекает из принципиального положения ЭИМ HL7, согласно которому все экземпляры класса Entity, вовлеченные в действие, указанное в экземпляре класса Act, описывают участие сущностей в этом действии в конкретной роли, указанной в экземплярах класса Role.

В целом классы Participation и Role необходимы для полного моделирования сложной семантики взаимоотношений экземпляров классов Entity и Act. Точное определение точки зрения на оказание медицинской помощи, положенной в основу ЭИМ, гласит: на самом верхнем уровне абстракции оказание медицинской помощи представляет собой ряд намеренных атрибутивных действий, в которых разными способами участвуют (выполняют, действуют по поручению, пользуются результатами и т. д.) несколько сущностей, имеющих определенные роли (например, «Джон Смит в роли пациента») и выполняющих определенные функции, описанные экземплярами класса Participaton (например, «участковый врач» и т.д.).

Два остальных базовых класса ЭИМ — ActRelationship и RoleLink — используются, чтобы «объединить» или «связать» экземпляры классов, с которыми они ассоциированы. Класс RoleLink используется, чтобы описать «связь на основе зависимости» (например, подотчетность, цепочку доверительных отношений и т.д.) между двумя экземплярами сущностей, выполняющих определенные роли. Семантика класса ActRelationship объясняется далее.

Б.4 Связывание действий: семантика класса ActRelationship

Понимание семантики и применения класса ActRelationship начинается с понимания «фрактальной» или «пошаговой» природы совокупности действий. В свою очередь, для этого лучше всего отправляться от разветвленной структуры классификации трех типов «связывания отношений», представленных экземплярами класса ActRelationship: «целое/часть» (например, панели лабораторных анализов (см. далее обсуждение «пошаговых действий»)); «связь на основе правил» (например, план лечения, протоколы и т. д.); «когнитивные действия» (например, суждение, переименование, замена, категоризация, обоснование, причина и т. д.).

В части уже упомянутой «фрактальной» или «пошаговой» природы экземпляры классов ActRelationship могут использоваться для моделирования понятий «фрактальных» или «пошаговых» отношений внутри иерархии «целое/часть». Рассмотрим хирургическую процедуру, например, лапароскопическую холецистэктомию. Она может быть представлена как единственный экземпляр класса Act или же в качестве альтернативы как «совокупность» (частично упорядоченных) экземпляров класса Act, каждый из которых описывает более тонкие детали, например, получить согласие на проведение операции, применить перед операцией определенные лекарства, управлять анестезией (в течение всей хирургической процедуры), сделать разрез и т. д. В свою очередь, каждое из этих более тонких действий может быть расчленено на еще более тонкие действия. Уровень степени детализации четко зависит от контекста действия или действий и степени интереса или намерений стороны, выполняющей (или не выполняющей) декомпозицию действий. На рис. 9 показана «точка зрения хирурга» на некоторые экземпляры классов Act и ActRelationship при моделировании действия «холецисэктомия». Прямоугольники с острыми углами представляют экземпляры класса Act (у каждого из них атрибут moodCode имеет значение DEF, что означает описание действия). Скругленные прямоугольники представляют экземпляры класса ActRelationship,

у которых атрибут typeCode имеет значение COMP (has component — имеет компонент). Значения атрибута sequenceNumber определяют порядок связи. Каждое действие, в свою очередь, может быть расщеплено на компоненты плана.

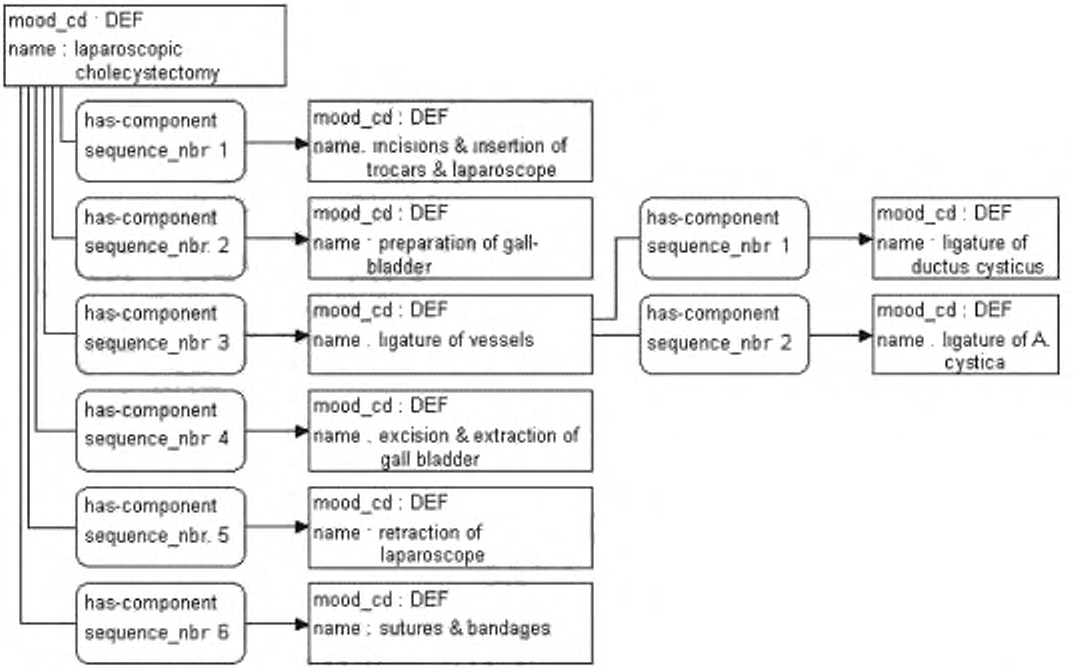

Рисунок 9 — Пример конструкции последовательного плана лапароскопической холецистэктомии

В целом классы Act и ActRealshionship необходимы для полного моделирования семантики каждого из трех упомянутых ранее типов «связывания отношений» с той степенью грубости или тонкости, которая необходима для конкретного контекста передачи данных. Кроме того, различные типы связывания и разные уровни детализации, представленные экземплярами класса ActRealshionship, могут (и будут) использоваться для коллективного формирования сложной семантики клинического мышления. Например, экземпляр класса ActRealshionship (описывающий связь «имеет обоснование») может использоваться для формирования связи экземпляра класса Observation, описывающего конкретный результат лабораторного анализа (например, скорость оседания эритроцитов равна 48), с экземпляром класса Observation, описывающим конкретный диагноз (например, системная красная волчанка).

Б.5 Определения шести базовых классов ЭИМ

Б.5.1 Класс Act (действие)

Класс Act описывает интересующее действие, которое произошло, может произойти, происходит, запланировано или затребовано. Оно должно быть намеренным действием в предметной области стандарта HL7. Оказание медицинской помощи (и любая профессиональная или деловая деятельность) состоит из намеренных действий. Экземпляр класса Act представляет собой запись о таком намеренном действии.

Б.5.2 Класс Entity (сущность)

Класс Entity описывает класс физических сущностей, или конкретный экземпляр такой сущности, или организацию либо группу физических сущностей, которые могут участвовать в действиях, артефакт. К ним относятся живые существа (включая людей), организации, материал и места. Иерархия специализаций класса Entity охватывает информацию о людях, организациях, живых организмах, устройствах, фармацевтических субстанциях и т. д. Она не включает в себя информацию о событиях, действиях, деятельности, определения предметов или о ролях, выполняемых сущностями (например, пациент, поставщик медицинской помощи).

Б.5.3 Класс Role (роль)

Класс Role описывает компетентность сущности, выполняющей роль в соответствии с определениями, данными другой сущностью, контролирующей эту роль.

Сущность, имеющая конкретную роль, может участвовать в действии. Следует отметить, что конкретная сущность в конкретной роли может участвовать в действии многими способами. Например, лицо в роли врача может участвовать в ведении пациента как дежурант или как лечащий врач. Роль определяет компетентность сущности, не зависящую от любого действия, в противоположность участию, которое ограничено рамками действия.

Каждая роль выполняется одной сущностью (той, что имеет эту роль) и обычно «контролируется» другой сущностью. Например, роль «пациент» может выполняться (обычно) некоторым лицом и контролироваться организацией, оказывающей пациенту медицинскую помощь. Аналогичным образом работодатель контролирует роль «работника».

Б.5.4 Класс Participation (участие)

Класс Participation моделирует ассоциацию между классом роли Role и классом действия Act. Экземпляр класса Participation описывает участие сущности, имеющей определенную роль, в ассоциированном действии. Один и тот же экземпляр класса Role может быть связан с помощью экземпляров класса Participation с несколькими экземплярами класса Act, а один и тот же экземпляр класса Act может быть связан с помощью экземпляров класса Participation с несколькими экземплярами класса Role. Один экземпляр класса Participation всегда является ассоциацией между конкретным экземпляром класса Role и конкретным экземпляром класса Act. Участие ограничено рамками действия, роль же, напротив, описывает компетенцию сущности, не зависящую от какого-либо действия.

Б.5.5 Класс ActRelationship (связь действий)

Класс ActRelationship моделирует ассоциацию между двумя экземплярами класса Act. Примерами могут служить такие отношения между действиями, как целое/часть, предшественник/последователь, причина/следствие.

Класс ActRelationship имеет две ассоциации с классом Act, одна из них имеет имя source (источник), вторая — target (цель). Связи, ассоциированные с экземпляром класса Act, должны рассматриваться как свойства экземпляра-источника. Это означает, что источник информации, передаваемой в экземпляре класса Act, несет ответственность не только за значения атрибутов этого экземпляра, но и за все связи с этим экземпляром, имеющие его в качестве источника.

Правило приписывания состоит в том, что все экземпляры класса ActRelationship приписаны ответственному действующему лицу того экземпляра класса Act, который является источником ассоциации с экземплярами класса ActRelationship (действием-источником).

Б.5.6 Класс RoleLink (связь ролей)

Класс RoleLink описывает связь между двумя ролями, выражающую зависимость между этими ролями.

Б. 5.7 Подклассы классов Act, Entity и Role

Классы Act, Entity и Role представляют собой классы «высокого уровня», хотя и не являются «абстрактными» в формальном смысле (т.е. значащие экземпляры этих классов вполне обычны). Поэтому потребовалось определить некоторое число более специализированных подклассов этих трех классов, чтобы указать дополнительные данные (атрибуты классов), требуемые в более конкретном контексте (например, класс Observation, являющийся подклассом класса Act, и классы LivingSubject и Material, являющиеся подклассами класса Entity). Атрибуты подкласса должны быть и полезны, и уникальны для этого подкласса. Подклассы наследуют все атрибуты родительского класса.

В каждой из этих иерархий есть значимые подклассы, для которых не нужны дополнительные атрибуты, поэтому они не представляются как отдельные классы в ЭИМ. Атрибут «classCode» в каждой из этих иерархий указывает, какой именно подкласс представлен данным классом. Система кодирования значений атрибута «classCode» строго контролируется комитетом HL7. Второй атрибут в каждой иерархии — атрибут «code» — используется для дальнейшей классификации подтипов каждого подкласса.

Б.5.8 Понятие наклонения

Класс Act описывает намеренные действия. Эти действия могут существовать в разных «наклонениях» («moods»). Наклонения описывают цикл деятельности от определения деятельности к ее планированию, от запланированной или потребованной деятельности к завершенной. Наклонение действия, описанного в экземпляре класса Act, определяется значением атрибута Act.moodCode.

Любой экземпляр класса Act имеет одно и только одно наклонение и не меняет его на протяжении своего жизненного цикла. Наклонения — определение, намерение, требование, событие — описывают разные моменты жизненного цикла деятельности. Однако участники каждого такого момента деятельности различны, как и данные, которыми они оперируют. Поэтому наклонение, описанное в экземпляре класса Act, является статическим. Актуализацию процесса деятельности (т. е. движение от ее определения к планированию, требованию и выполнению) называют «циклом деятельности», чтобы отличить ее от «жизненного цикла» единственного экземпляра действия. Экземпляры класса Act, формирующие такой «цикл деятельности», связаны между собой с помощью экземпляров класса ActRelationship.

Б.6 Типы данных и спецификации словарей

Определения классов ЭИМ, атрибутов и ассоциаций дают информацию о логическом значении этих элементов, но для полного определения необходимы типы данных и спецификации словарных доменов. Типы данных определяют допустимые значения атрибутов и «смысл» этих значений. Поэтому типы данных являются фундаментальными строительными блоками, образующими (и ограничивающими) всю семантику, которую в конечном счете можно выразить в ЭИМ. В стандарте HL7 Версии 3 спецификации типов данных описаны в документах Data Types Abstract Specification (абстрактная спецификация типов данных) и V3 Data Types Implementable Technology Specification for XML (технологическая спецификация реализации типов данных Версии 3 на языке XML). Сводный перечень типов данных приведен также в Приложении В.

Спецификации словарей значений атрибутов указывают словарный домен по умолчанию и «силу кодирования» («coding strength»). Словарные домены явно определены в разделе 9. Спецификации словарей документируют перекрестные ссылки и альтернативные представления систем кодирования, а также описывают логические понятия, выражаемые каждым кодом. Каждый кодированный атрибут может принимать значения только из соответствующего словарного домена.

Сила кодирования (Coding strength) является кодированным свойством атрибута, указывающим, может ли пользователь стандарта HL7 Версии 3 передавать кодированное значение, не входящее в словарный домен. Такое значение называется «исключением» для словарного домена. Сила кодирования может иметь одно из двух значений. Значение силы кодирования CWE (Coded with Exceptions — кодируемый с исключениями) указывает, что конечный пользователь может передавать кодированный термин, взятый из местной системы кодирования, а не из формально определенного словарного домена. Значение силы кодирования CNE (Coded, No Exceptions — кодируемый без исключений) указывает, что экземпляры сообщений могут содержать только коды из словарного домена, ограничивающего значения данного атрибута.

Б.7 Методология стандарта HL7 Версия 3 и ЭИМ

В целом основным мотивом разработки ЭИМ было желание точного определения различных элементов данных и отношений, образующих информационное пространство здравоохранения. Прямым результатом такого способа применения знаний явилась способность повторного использования того же самого понятия во многих сообщениях передачи медицинских данных. Полный процесс определения сообщения определен и обсужден в документе V3 Guide и в дополняющем его руководстве HL7 Message Definition Framework (MDF). Основными шагами этого процесса являются определение и документирование потребностей в обмене данными (например, комплекса сообщений для поддержки конкретного клинического или административного процесса) в форме сценариев (storyboard), выбор из ЭИМ тех классов и атрибутов, которые необходимы для составления данного сообщения, и последующее применение дополнительных ограничений на число повторений и допустимые значения каждого атрибута.

Дополнительную информацию см. в упомянутых ранее документах V3 Guide и MDF.

Приложение В
(справочное)

Сводный перечень типов данных Версии 3

В.1 Обзор типов данных

В таблице В.1 приведены определения типов данных, которые были приняты как часть стандарта HL7 Версии 3 по результатам голосования, проведенного 2 декабря 2002 года. Следующие версии ЭИМ будут содержать полную спецификацию типов данных, а тело ЭИМ будет содержать гиперссылки на определения типов данных.

Таблица В.1 — Типы данных ЭИМ

Имя	Символ	Описание
DataValue	ANY	Определяет основные свойства каждого типа данных. Это абстрактный тип, означающий, что никакое значение не может быть только значением данных, не принадлежащим никакому конкретному типу. Каждый конкретный тип является специализацией этого общего абстрактного типа DataValue
Boolean	BL	Булевский тип представляет значения двузначной логики. Булевское значение может быть или TRUE (истина), или FALSE (ложь), или, как и любое другое значение, может быть пустым (NULL)
Encapsulated Data	ED	Данные, в основном рассчитанные на чтение человеком или на дальнейшую машинную обработку данных за пределами области применения стандарта HL7. К ним относятся неформатированный или форматированный текст, мультимедийные данные, или структурированная информация, определенная другим стандартом (например, XML-подпись). Вместо самих данных значение типа ED может содержать только ссылку на данные (см. описание типа данных TEL). Заметьте, что тип данных ST — это специализация типа данных ED, у которой компонент типа среды ED имеет значение text/plain (только текст)
Character String	ST	Строковый тип данных, предназначенный для текстовых данных, используемых главным образом для машинной обработки (например, сортировка, запрос, индексирование и т. д.). Используется для имен, символов и формальных выражений
Concept Descriptor	CD	Описание понятия, предназначенное для представления любого вида понятий, обычно с помощью кода, определенного в некоторой системе кодирования. Описание понятия может содержать исходный текст или фразу, послужившую основой для кодирования, и один или несколько эквивалентов этого кода в других системах кодирования. Описание понятия может также содержать квалификатор, используемый, к примеру, для описания понятия «левая нога» как посткоординированного термина, образованного от исходного кода «нога» и квалификатора «левый». В исключительных случаях описание понятия может не содержать код, а только исходный текст, описывающий это понятие

Продолжение таблицы В.1

Имя	Символ	Описание
Coded Simple Value	CS	Кодированные данные в их простейшей форме, в которой указан только код без его значения, предназначенного для вывода на экран. Система кодирования и ее версия определяются контекстом, в котором передается значение типа CS. Тип данных CS используется для значений кодированных атрибутов, имеющих единственный словарный домен, определенный в стандарте HL7
Coded With Equivalents	CE	Кодированные данные, состоящие из кодированного значения (типа CV) и, необязательно, одного или нескольких кодированных значений, принадлежащих другим системам кодирования и описывающих то же самое понятие. Используется при наличии альтернативных кодов
Character String with Code	SC	Строка символов, к которой при необходимости может быть добавлен код. Если код присутствует, то должен присутствовать и текст. Код часто берется из местной системы кодирования
Instance Identifier	II	Идентификатор, который уникально идентифицирует предмет или объект. Примерами могут служить объектные идентификаторы (ОИД) объектов ЭИМ HL7, номер медицинской карты, идентификатор направления, идентификатор позиции прейскуранта, идентификационный номер транспортного средства и т. д. Идентификаторы экземпляров определяются на основе объектных идентификаторов ветви ISO
Telecommunication Address	TEL	Номер телефона (голосового или факса), адрес электронной почты или другой указатель ресурса, управляемого телекоммуникационным оборудованием. Адрес указывается в форме Единого указателя ресурсов URL (Uniform Resource Locator), к которому добавлена спецификация времени и коды использования, помогающие установить, какой адрес используется в конкретное время и для каких целей
Postal Address	AD	Почтовый, домашний или юридический адрес. Последовательность компонентов адреса, например, улица или почтовый ящик, город, почтовый индекс, страна и т. д.
Entity Name	EN	Фамилия, имя, отчество лица, название организации, места или предмета. Последовательность компонентов, например, имя или фамилия, приставка, суффикс и т. д. Примерами могут служить «Джим Боб Уолтон, мл.», «HL7, Inc.», «Озеро Tahoe» и т. д. Именование объекта может быть простой строкой символов или состоять из нескольких компонентов, например, «Джим», «Боб», «Уолтон» и «мл.»; «HL7» и «Inc.»; «Озеро» и «Tahoe»
Trivial Name	TN	Ограничение именования сущности в форме одной строки, используемой для простого названия предметов и мест
Person Name	PN	Фамилия, имя, отчество лица. Последовательность компонентов, например, имя, фамилия, приставка, суффикс и т. д.
Organization Name	ON	Название организации. Последовательность компонентов названия

Продолжение таблицы В.1

Имя	Символ	Описание
Integer Number	INT	Целые числа (–1,0,1,2,100,3398129 и т. д.) — точные числа, являющиеся результатами подсчета и нумерации. Целые числа дискретны, набор целых чисел бесконечен, но счетен. Диапазон значений целых чисел не ограничен никаким произвольным пределом. Для типа данных INT определены две причины пустоты (NULL flavor), описывающие положительную и отрицательную бесконечность
Real Number	REAL	Вещественные числа. Обычно используются в тех случаях, когда количества измеряются, оцениваются или вычисляются из других вещественных чисел. Типичным представлением вещественного числа служит десятичная запись, в которой число значащих десятичных цифр известно как точность
Ratio	RTO	Количество, представленное как отношение числителя к знаменателю. Общие множители в числителе и знаменателе автоматически не сокращаются. Тип данных RTO используется для титров (например, «1:128») и других величин в результатах лабораторных анализов, которые действительно представляют собой отношения. Отношения не являются просто «структурированными числовыми данными», поэтому измерения артериального давления (например, «120/60») не являются отношениями. Во многих случаях вместо типа данных RTO должен использоваться тип данных REAL
Physical Quantity	PQ	Размерная величина, выражающая результат измерения
Monetary Amount	MO	Денежная сумма, выражающая количество денег в некоторой валюте. Валюты — единицы, в которых указаны денежные суммы в разных экономических регионах. В то время как денежная сумма является единственным видом количества (денег), обменные курсы валют являются переменными. Это принципиальное отличие между физическим количеством и денежными количествами, а также причина того, почему единицы валюты не являются физическими единицами
Point in Time	TS	Величина, указывающая момент времени на оси естественного времени. Момент времени чаще всего представляется как календарное выражение
Set	SET	Значение, содержащее другие различные значения ни в каком конкретном порядке
Sequence	LIST	Значение, содержащее другие дискретные значения в определенной последовательности
Bag	BAG	Неупорядоченная коллекция значений, в которой каждое значение может содержаться более одного раза
Interval	IVL	Множество последовательных значений упорядоченного базового типа данных
History	HIST	Множество значений данных, которые соответствуют типу исторических данных (HXIT), т. е. имеют свойство действительного времени. Историческая информация не ограничена прошлым; в ней могут присутствовать ожидаемые будущие значения

Окончание таблицы В.1

Имя	Символ	Описание
Uncertain Value — Probabilistic	UVP	Общее расширение типа данных, используемое, чтобы указать вероятность, отражающую степень доверия поставщика данных к их значениям
Parametric Probability Distribution	PPD	Общее расширение типа данных, выражающее неопределенность количественных данных с помощью функции распределения и ее параметров. Помимо специфических параметров распределения всегда указывается среднее значение (математическое ожидание) и стандартное отклонение, чтобы обеспечить минимальный уровень интероперабельности на тот случай, если приложение-получатель не в состоянии работать с определенным распределением вероятности
General Timing Specification	GTS	Множество моментов времени, указывающее времена событий и действий, а также шаблоны циклов, которые могут применяться к некоторым видам информации, например, к телефонным номерам (утро, вечер), к адресам (для так называемых «перелетных птиц», которые живут зимой на юге, а летом на севере), а также к часам работы

Приложение ДА.1
(справочное)

Сведения о соответствии ссылочных межгосударственных стандартов международным стандартам (международным документам)

Таблица ДА.1

Обозначение ссылочного международного стандарта (международного документа)	Степень соответствия	Обозначение и наименование соответствующего межгосударственного стандарта
ISO 17113, Health informatics — Exchange of information between healthcare information systems — Method for development of messages. (ИСО 17113, Информатизация здоровья — Обмен информацией между медицинскими информационными системами — Метод разработки сообщений)	—	*
ISO/IEC 19501, Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2 (ИСО/МЭК 19501, Информационные технологии — Открытая распределенная обработка — Унифицированный язык моделирования (UML), Версия 1.4.2.).	—	*
ANSI/HL7 V3 DT, R1–2004, HL7 Version 3 Standard: Data Types — Abstract Specification, Release 1 (HL7 Версия 3 — Типы данных — Общая спецификация, выпуск 1).	—	*
* Соответствующий межгосударственный стандарт отсутствует.		

УДК 004:61:006.354 МКС 35.240.80 П85 ОКСТУ 4002

Ключевые слова: здравоохранение, информатизация здоровья, структура и определения информационных объектов ссылок

Редактор *В. В. Космин*
Технический редактор *Е. В. Беспрозванная*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *Т. Ф. Кузнецовой*

Сдано в набор 12.12.2014. Подписано в печать 29.01.2015. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 15,34. Уч.-изд. л. 13,50. Тираж 30 экз. Зак. 2279.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru