ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 59167— 2020 (ИСО/МЭК 19987:2017)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Стандарт информационных сервисов EPC (EPCIS)

(ISO/IEC 19987:2017, MOD)

Издание официальное

Москва
Стандартинформ
2021

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Ассоциацией автоматической идентификации «ЮНИСКАН/ГС1 РУС» на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 355 «Технологии автоматической идентификации и сбора данных» совместно с ТК 022 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 22 декабря 2020 г. № 1358-ст

4 Настоящий стандарт является модифицированным по отношению к международному стандарту ИСО/МЭК 19987:2017 «Информационные технологии. Стандарт информационных сервисов EPC (EPCIS)» (ISO/IEC 19987:2017 «Information technology — EPC Information Services (EPCIS) Standard», MOD) путем включения раздела «Введение», переименования раздела 1 в «Область применения», перевода раздела «13 Ссылки» в «Библиографию», введения нумерации таблиц и рисунков, а также ссылок на пронумерованные таблицы и рисунки, замены ссылок на международные документы на порядковые номера, приведенные в «Библиографии». В дополнительном приложении ДА приведен перечень сокращений, использованных в настоящем стандарте. Изменения, включенные в текст стандарта, выделены в тексте курсивом и подчеркиванием. Сноски в тексте настоящего стандарта, выделенные курсивом, приведены для пояснения его содержания. Подробная информация об изменениях и пояснения по необходимости их внесения приведены в дополнительном приложении ДБ

5 ВВЕДЕН ВПЕРВЫЕ

6 Некоторые положения настоящего стандарта могут быть объектами патентных прав. Федеральное агентство по техническому регулированию и метрологии не несет ответственности за идентификацию подобных патентных прав

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации. Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ближайшем выпуске ежегодного информационного указателя «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© ISO, 2017 — Все права сохраняются
© IEC, 2017 — Все права сохраняются
© Стандартинформ, оформление, 2021

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

Введение

Настоящее введение содержит следующие структурные элементы, приведенные в [1]:
- аннотацию к документу [2],
- протокол изменений в [2],
- отказ от ответственности.

Аннотация к документу [2]

Параметр документа	Текущее значение
Наименование документа	Стандарт информационных сервисов EPC (EPCIS)
Дата публикации документа	Сентябрь 2016
Версия документа	1.2
Издание документа	
Статус документа	Утвержден
Описание документа	Предоставляет возможность различным приложениям создавать данные о событиях, обеспечивающих прозрачность, и обмениваться ими как внутри отдельных предприятий, так и между ними

Протокол изменений в [2]

Выпуск	Дата внесения изменения	Ответственный за внесение изменения	Краткое изложение изменения
1.0 *(См. [3])*			Начальная версия
1.1 *(См. [4])*	май 2014		Версия EPCIS 1.1 *по [4]* обеспечивает полную обратную совместимость с версией EPCIS 1.0.1 по *[3]*. Версия EPCIS 1.1 *по [4]* включает следующие новые или усовершенствованные функции: Добавлена поддержка идентификации на уровне класса для событий `ObjectEvent (Событие_с_Объектом)`, `AggregationEvent (Событие_с_Агрегацией)` и `TransformationEvent (Событие_с_Преобразованием)` посредством добавления количественных списков. Новый тип события – `TransformationEvent (Событие_с_Преобразованием)` позволяет описывать события, в рамках которых исходные объекты полностью трансформируются и на выходе возникают новые объекты. Параметр «на каком основании» ("why") всех типов событий усовершенствован с тем, чтобы обеспечить возможность включения информации о начальном и конечном пунктах процесса хозяйственной деятельности. Параметр «на каком основании» ("why") некоторых типов событий усовершенствован с тем, чтобы обеспечить возможность включения мастер-данных экземпляра/серии. `SimpleEventQuery (Простой_Запрос_События)` усовершенствован таким образом, чтобы учесть вышеуказанные изменения в типах событий. Вводные материалы пересмотрены с тем, чтобы обеспечить соответствие системной архитектуре GS1. Объяснение механизма расширения XML стало более полным. Использование типа события `QuantityEvent (Событие_с_Количеством_Объектов)` объявлено нежелательным, поскольку его функционал полностью покрывается типом событий `ObjectEvent (Событие_с_Объектом)` с добавлением количественных списков

Окончание таблицы

Выпуск	Дата внесения изменения	Ответственный за внесение изменения	Краткое изложение изменения
1.2 *(См. [1])*	сентябрь 2016		Версия EPCIS 1.2 *по [2]* обеспечивает полную обратную совместимость с версиями EPCIS 1.1 *по [4]* и 1.0.1 *по [3]*. Версия EPCIS 1.2 *по [2]* включает следующие новые или усовершенствованные функции: Введен механизм, позволяющий объявлять предшествующее событие EPCIS ошибочным и предназначенный для использования в ситуациях, когда отсутствует возможность корректировки истории с помощью обычных событий EPCIS. Данный механизм включает структуру `errorDeclaration` (`Заявление_Об_Ошибке`) в событии EPCIS и связанных параметрах запроса. Ко всем событиям EPCIS добавлен необязательный `eventID` (`Идентификатор_события`). Его основное предназначение состоит в обеспечении возможности для события заявления об ошибке (при необходимости) ссылаться на одно или несколько корректирующих событий. Простой запрос события усовершенствован с тем, чтобы прояснить, что запросы для полей расширения или мастер-данных на уровне экземпляра/серии (МДЭС) относятся исключительно к элементам XML высокого уровня, при этом также введен новый набор параметров запроса для запросов в отношении элементов XML, вложенных внутрь элементов высокого уровня. Уточнена роль документа EPCIS как средства передачи данных о событиях из пункта в пункт. Заголовок EPCIS в схемах XML усовершенствован с тем, чтобы при необходимости обеспечить возможность включения мастер-данных. Использование элементов расширения в полях `<readPoint>` (`<место_Считывания>`) и `<bizLocation>` (`<производственное_Место_Назначения>`) объявлено нежелательным. Добавлен раздел 12, касающийся соответствия

Отказ от ответственности

В соответствии с Положением о политике в сфере интеллектуальной собственности GS1 стремится избежать неопределенности в отношении претензий, связанных с интеллектуальной собственностью. С этой целью от участников Рабочей группы, разработавшей настоящий **стандарт информационных сервисов EPC (EPCIS)**, требуется согласие на предоставление членам GS1 безвозмездной лицензии или лицензии на обоснованных и недискриминационных условиях (RAND) в отношении необходимых притязаний в соответствии с определением данного термина, приведенным в Положении GS1 о политике в сфере интеллектуальной собственности. Кроме того, обращается внимание на возможность возникновения ситуации, при которой применение одного или нескольких элементов настоящего документа может оказаться в сфере действия патента или иного права интеллектуальной собственности, не предусматривающего необходимого притязания. Подобный патент или право интеллектуальной собственности не является предметом данных лицензионных обязательств GS1. Согласие на предоставление лицензий, предоставляемых в соответствии с Положением GS1 о политике в сфере интеллектуальной собственности, также не распространяется на права интеллектуальной собственности и любые притязания третьих сторон, которые не являлись участниками настоящей Рабочей группы.

В этой связи GS1 рекомендует организациям, разрабатывающим применения, соответствующие настоящему документу, установить наличие патентов, которые могут затрагивать конкретное применение, разрабатываемое данной организацией в соответствии с настоящим документом, а также определить необходимость получения лицензии в соответствии с патентом или иным правом интеллектуальной собственности. Определение необходимости получения лицензии должно производиться с учетом особенностей конкретной системы, разрабатываемой организацией, с привлечением ее патентного поверенного.

НАСТОЯЩИЙ ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ» БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ВКЛЮЧАЯ ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, ПАТЕНТНОЙ ЧИСТОТЫ, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ, ИЛИ КАКИХ-ЛИБО ИНЫХ ГАРАНТИЙ, ПРОИСТЕКАЮЩИХ ИЗ НАСТОЯЩЕГО ДОКУМЕНТА. GS1 снимает с себя всякую ответственность за любые убытки, проистекающие из использования или ненадлежащего использования настоящего стандарта, включая фактические, непрямые, косвенные и компенсаторные убытки, а также включая ответственность за нарушение каких-либо прав интеллектуальной собственности, связанное с использованием информации, содержащейся в настоящем документе, или основанное на использовании настоящего документа.

GS1 оставляет за собой право в любое время и без уведомления вносить изменения в настоящий документ. GS1 не предоставляет никаких гарантий в отношении использования настоящего документа и отказывается от ответственности за какие-либо ошибки, которые могут обнаружиться в настоящем документе, а также от обязательства по актуализации содержащейся в нем информации.

GS1 и логотип GS1 являются зарегистрированными товарными знаками GS1 AISBL.

Содержание

1 *Область применения* ... 1
2 Взаимосвязь с архитектурой системы GS1 ... 2
2.1 Обзор стандартов GS1 ... 2
2.2 EPCIS во взаимосвязи с уровнями «сбора данных» и «обмена данными» ... 3
2.3 Стандарт EPCIS во взаимосвязи с торговыми партнерами ... 5
2.4 Стандарт EPCIS во взаимосвязи с другими компонентами архитектуры системы GS1 ... 7
3 Принципы спецификации EPCIS ... 9
4 Терминология и условные обозначения ... 9
5 Структура спецификации EPCIS ... 10
5.1 Уровни ... 10
5.2 Расширяемость ... 11
5.3 Модульность ... 12
6 Уровень абстрактной модели данных ... 12
6.1 Данные о событии и мастер-данные ... 12
6.2 Категории лексики ... 15
6.3 Механизмы расширения ... 16
6.4 Представление идентификаторов ... 17
6.5 Иерархические лексиконы ... 18
7 Уровень определения данных ... 18
7.1 Общие правила формирования спецификации модулей уровня определения данных ... 18
7.2 Модуль типов основных событий — обзор ... 21
7.3 Модуль типов основных событий — составные части ... 25
7.4 Модуль типов основных событий. События ... 36
8 Сервисный уровень ... 57
8.1 Модуль основных операций сбора данных ... 58
8.2 Модуль основных операций с запросами ... 60
9 Привязки XML для модулей определения данных ... 88
9.1 Механизм расширяемости ... 88
9.2 Типовой заголовок бизнес-документа ... 90
9.3 Базовая схема EPCglobal ... 91
9.4 Мастер-данные в привязке XML ... 92
9.5 Схема для типов основных событий ... 94
9.6 Типы основных событий. Примеры (справочная информация) ... 109
9.7 Схема для мастер-данных ... 114
9.8 Мастер-данные. Пример (справочная информация) ... 117
10 Привязки для модуля основных операций сбора данных ... 118
10.1 Привязка очереди сообщений ... 118
10.2 Привязка к протоколу HTTP ... 119
11 Привязки для модуля основных операций с запросами ... 120
11.1 Схема XML для модуля основных операций с запросами ... 120
11.2 Привязка к протоколам SOAP/HTTP интерфейса управления запросами ... 129
11.3 Привязка к протоколу AS2 интерфейса управления запросами ... 139
11.4 Привязки для интерфейса обратных вызовов по запросам ... 141
12 Соответствие ... 143
12.1 Соответствие данных EPCIS ... 143
12.2 Соответствие клиентов интерфейса EPCIS сбора данных ... 144
12.3 Соответствие серверов интерфейса EPCIS сбора данных ... 144
12.4 Соответствие клиентов интерфейса EPCIS запросов ... 144
12.5 Соответствие серверов интерфейса EPCIS запросов ... 144
12.6 Соответствие применений интерфейса EPCIS обратных вызовов по запросам ... 145
Приложение ДА (справочное) Перечень сокращений ... 146
Приложение ДБ (справочное) Перечень изменений в настоящем стандарте относительно ИСО/МЭК 19987:2017 и пояснения по необходимости их внесения ... 148
Библиография ... 152

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Стандарт информационных сервисов EPC (EPCIS)

Information technology. EPC Information Services (EPCIS) Standard

Дата введения — 2022—01—01

1 *Область применения*

Настоящий документ соответствует стандарту GS1 [2], который определяет версию 1.2 информационных сервисов EPC (EPC Information Services, EPCIS). Целью EPCIS является предоставление возможности различным приложениям создавать данные о событиях, обеспечивающих прозрачность, и обмениваться ими как внутри предприятий, так и между ними. В конечном счете создание этого общего ресурса направлено на то, чтобы дать возможность пользователям получать общий доступ к физическим и цифровым объектам в рамках соответствующего контекста ведения хозяйственной деятельности.

В контексте EPCIS понятие «объекты» обычно соответствует физическим объектам, которые идентифицируются либо на уровне класса, либо на уровне экземпляра и которые рассматриваются на этапах физической обработки в рамках всего бизнес-процесса, охватывающего одну или несколько организаций. Примерами таких физических объектов являются предметы торговли, логистические единицы, возвратные активы, материальные активы, физические документы и т. д. «Объекты» также могут относиться к цифровым объектам, идентифицируемым либо на уровне класса, либо на уровне экземпляра и участвующим на сопоставимых этапах бизнес-процессов. Примерами таких цифровых объектов являются цифровые предметы торговли (загрузки музыкальных произведений, электронные книги и т. п.), цифровые документы (электронные купоны и т. п.) и тому подобное. В настоящем стандарте слово «объект» используется для обозначения физического или цифрового объекта, являющегося предметом этапа бизнес-процесса и идентифицируемого на уровне класса или на уровне экземпляра. Данные в EPCIS состоят из «событий, обеспечивающих прозрачность», каждое из которых является записью о завершении конкретного этапа бизнес-процесса, воздействующего на один или несколько объектов.

Изначально стандарт EPCIS задумывался как часть более широких попыток расширения сотрудничества между торговыми партнерами путем предоставления общего доступа к детализированной информации о физических или цифровых объектах. Название EPCIS отражает начальные усилия в развитии электронного кода продукции (Electronic Product Code — EPC). Однако следует заметить, что EPCIS не требуют использования номеров электронного кода продукции, так же как и носителей данных радиочастотной идентификации (Radio-Frequency Identification — RFID). EPCIS версии 1.2 даже не требует идентификации на уровне экземпляра (для которого и был изначально разработан электронный код продукции). Стандарт EPCIS применяется для всех ситуаций, в которых данные о событиях, обеспечивающих прозрачность, должны быть получены и размещены в общий доступ, а наличие аббревиатуры «EPC» в наименовании стандарта имеет только историческое значение.

Сервисы EPCIS предоставляют открытые, типовые интерфейсы, которые служат «бесшовной» интеграции четко определенных сервисов как в средах взаимодействия между предприятиями, так и внутри самих предприятий. Типовые интерфейсы, определенные в EPCIS, позволяют запрашивать и получать данные о событиях, обеспечивающих прозрачность, с помощью определенного набора служебных

Издание официальное

операций и связанных с ними стандартов данных в сочетании с соответствующими механизмами безопасности, которые удовлетворяют потребностям предприятий-пользователей. В большинстве случаев это влечет применение одной или более баз данных долговременного хранения данных о событиях, обеспечивающих прозрачность, хотя для прямого, от приложения к приложению, доступа к данным без применения баз данных долговременного хранения могут быть использованы элементы сервисов.

С базами данных долговременного хранения или без них спецификация стандарта EPCIS определяет только типовой интерфейс обмена данными между теми приложениями, которые осуществляют сбор данных о событиях, обеспечивающих прозрачность, и теми приложениями, которым необходим доступ к этим данным. Однако настоящий стандарт не определяет, каким образом должны быть реализованы эти сервисные операции или сами базы данных. Также не определяется, каким образом сервисы EPCIS должны получать и/или обрабатывать необходимые данные, за исключением случаев, когда данные собираются с использованием типовых операций сбора данных EPCIS. Интерфейсы необходимы для функциональной совместимости, в то время как отличия в реализациях позволяют конкурировать тем, кто предоставляет технологию и внедряет настоящий стандарт.

Настоящий стандарт EPCIS предназначен для совместного использования со стандартом базовой деловой лексики GS1 по *ГОСТ Р 59168*[1)], содержащим определения значений данных, которые могут быть использованы для наполнения структур данных, определенных настоящим стандартом EPCIS. Использование стандартизованной лексики, предоставляемой стандартом БДЛ по *ГОСТ Р 59168*, имеет решающее значение для взаимодействия и для обеспечения выполнения запросов данных путем сокращения числа вариантов того, каким образом различные предприятия выражают общее намерение. Таким образом, при построении данных EPCIS приложения должны в максимально возможной степени использовать стандарт БДЛ по *ГОСТ Р 59168*.

Справочное Руководство по внедрению EPCIS и CBV (БДЛ) *[7]* предоставляет дополнительную информацию по построению систем, обеспечивающих прозрачность, с применением EPCIS и CBV (БДЛ), включая подробные сведения о моделировании конкретных деловых ситуаций с использованием данных EPCIS/CBV (БДЛ) и способах обмена такими данными между торговыми партнерами.

2 Взаимосвязь с архитектурой системы GS1

Настоящий раздел в основном состоит из выдержек из документов *[8]* и *[9]* и показывает взаимосвязь стандарта EPCIS с другими стандартами GS1.

2.1 Обзор стандартов GS1

Стандарты GS1 удовлетворяют информационные потребности конечных пользователей, взаимодействующих друг с другом в цепи поставок, в частности предоставляют информацию, необходимую для поддержки бизнес-процессов, посредством которых взаимодействуют участники цепи поставок. Субъектами такой информации являются объекты реального мира, которые являются частью этих бизнес-процессов. Объекты реального мира включают в себя предметы, которыми предприятия обмениваются между собой, такие как продукция, детали, сырье, упаковка и т. д. Другие объекты реального мира, имеющие отношение к торговым партнерам, включают оборудование и материалы, необходимые для выполнения бизнес-процессов, связанных с торговлей, такие как контейнеры, транспорт, машины и механизмы; объекты, относящиеся к физическим местам нахождения выполняемых бизнес-процессов; юридические лица, такие как предприятия, подразделения; служебные отношения; бизнес-транзакции и документы, а также многое другое.

Объекты реального мира могут существовать в материальном мире, а могут быть цифровыми или понятийными. Примерами физических объектов являются продукция бытовой электроники, транспортный контейнер, а также производственный участок (объект в месте нахождения). Примерами цифровых объектов являются загруженное музыкальное произведение, электронная книга, а также электронный купон. Примерами понятийных объектов являются класс предметов торговли, категория продукции, а также тип юридического лица.

Стандарты GS1 могут быть разделены на следующие группы в соответствии с их ролью в поддержке информационных потребностей, связанных с объектами реального мира, в бизнес-процессах цепи поставок:

[1)] *ГОСТ Р 59168—2020 (ИСО/МЭК 19988:2017) «Информационные технологии. Стандарт базовой деловой лексики» является модифицированным по отношению к ИСО/МЭК 19988:2017 [5]. В тексте ИСО/МЭК 19987:2017 [1] вместо ссылок на [5] используется фраза «стандарт базовой деловой лексики GS1» (см. [6]), в то же время [5] соответствует [6] (стандарту базовой деловой лексики GS1 версии 1.2).*

- стандарты, которые предоставляют средства для **идентификации («identify»)** объектов реального мира, так что они могут становиться объектами электронной информации, которая сохраняется и/или передается конечными пользователями. Стандарты GS1 для идентификации включают стандарты, которые определяют уникальные идентификационные кодовые обозначения (так называемые идентификационные ключи GS1);

- стандарты, которые предоставляют средства автоматического **сбора данных («capture»)**, нанесенные непосредственно на физические объекты, перекидывая мост из мира физических объектов в мир электронной информации. Стандарты сбора данных GS1 включают определения носителей данных штрихового кода и радиочастотной идентификации, которые позволяют прикреплять идентификаторы непосредственно к физическому объекту, а также стандарты, определяющие совместимые интерфейсы устройств считывания, печати и других компонентов аппаратного и программного обеспечения, которые связывают носители данных с приложениями для ведения хозяйственной деятельности;

- стандарты, которые предоставляют средства для **обмена данными («share»)**, как между торговыми партнерами, так и внутри предприятий, обеспечивая основу для электронных бизнес-транзакций, электронной прозрачности физического или цифрового мира, а также других информационных приложений. Стандарты GS1 для обмена информацией включают в себя настоящий стандарт EPCIS, который является стандартом для данных о событиях, обеспечивающих прозрачность. Другими стандартами в группе «обмена данными» являются стандарты для мастер-данных и данных бизнес-транзакций, так же как и стандарты поиска, которые помогают определить, где находятся соответствующие данные по всей цепи поставок, а также доверительные стандарты, которые помогают создать условия для обмена данными с надлежащей безопасностью.

Настоящий стандарт EPCIS вписывается в группу «обмена данными», предоставляя стандарт для данных о событиях, обеспечивающих прозрачность, и стандарты интерфейса для сбора такой информации от инфраструктуры для сбора данных (которая использует стандарты из группы «сбора данных»), а также для обмена такой информацией с приложениями для ведения хозяйственной деятельности и торговыми партнерами.

2.2 EPCIS во взаимосвязи с уровнями «сбора данных» и «обмена данными»

На *рисунке 1* показана взаимосвязь между настоящим стандартом EPCIS и другими стандартами GS1 в группах «сбор данных» и «обмен данными». (Группа стандартов «идентификация» охватывает данные на всех уровнях этой архитектуры и поэтому не показана.)

Как показано на *рисунке 1*, интерфейс сбора данных EPCIS (EPCIS Capture Interface) служит мостом между стандартами «сбора данных» и «обмена данными». Интерфейс запросов EPCIS (EPCIS Query Interface) предоставляет данные о событиях, обеспечивающих прозрачность, как для внутренних приложений, так и для обмена с торговыми партнерами.

В центре приложения по сбору данных (Data Capture Application) находится рабочий процесс сбора данных (Data Capture Workflow), контролирующий этап бизнес-процесса, в рамках которого происходит сбор данных. Этот процесс, как правило, имеет собственную логику, специфичную для конкретного приложения. Под рабочим процессом сбора данных на *рисунке 1* показан канал передачи данных между рабочим процессом и носителями данных GS1: символами штрихового кода и радиочастотными метками. Зеленые прямоугольные блоки на *рисунке 1* обозначают стандарты GS1, положения которых могут быть использованы в качестве требований к интерфейсам носителей данных. В верхней части *рисунка 1* расположены интерфейсы между рабочим процессом сбора данных и крупномасштабными корпоративными приложениями. Многие из этих интерфейсов являются специфичными для конкретного приложения или конкретного предприятия, даже с использованием данных GS1 в качестве компоновочных блоков. Тем не менее интерфейс EPCIS является одним из стандартов GS1. Следует обратить внимание, что интерфейсы в верхней части схемы, в том числе и EPCIS, не зависят от носителя данных, используемого в нижней части *рисунка 1*.

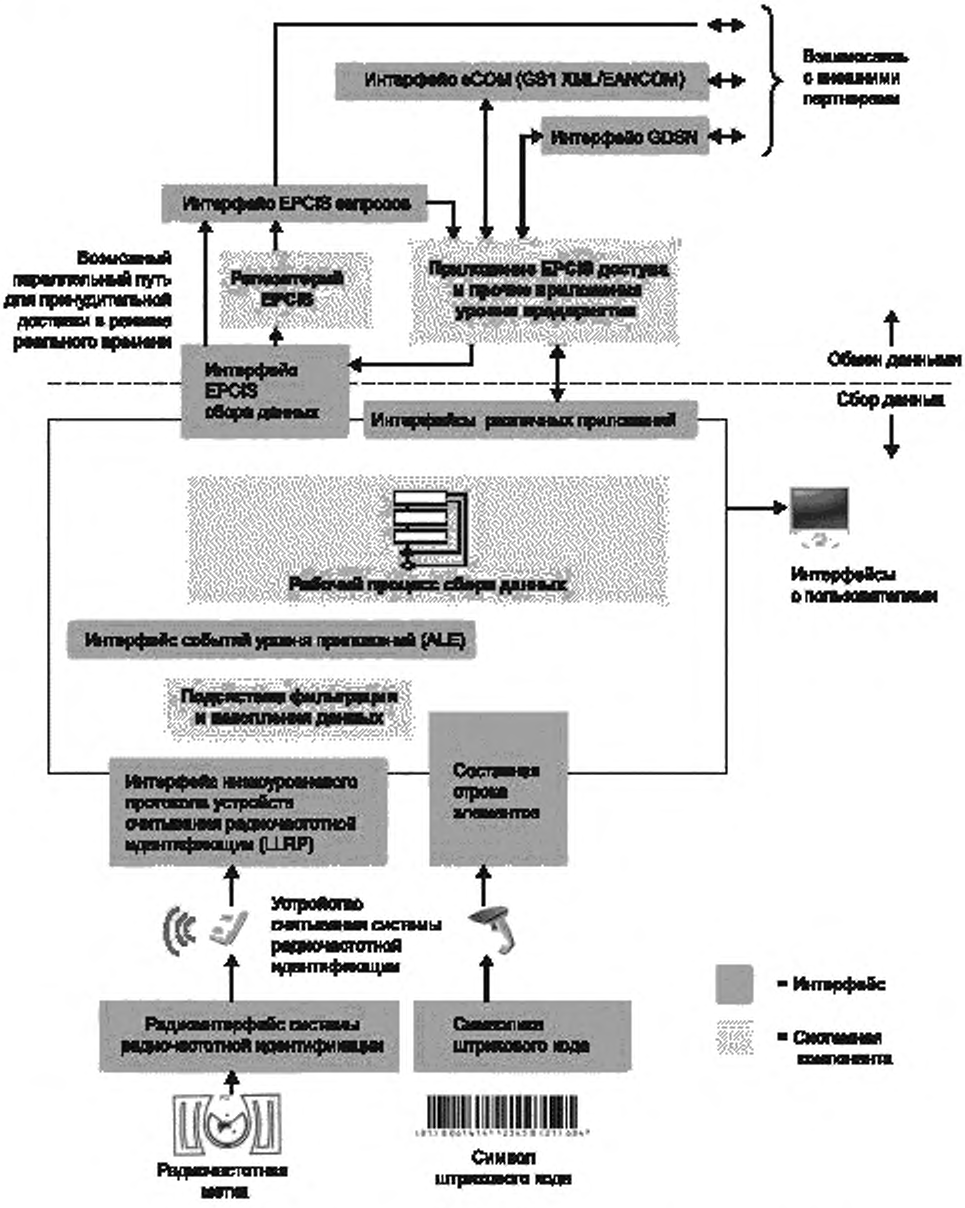


Рисунок 1

Целью данных интерфейсов, а также причиной использования многоуровневой архитектуры для сбора данных является обеспечение разграничения между различными уровнями абстракции. С позиции корпоративного приложения (т. е. верхнего синего прямоугольного блока на *рисунке 1*) приложение сбора данных целиком защищает корпоративное приложение (Enterprise-level Application) от детализации того, как именно происходит сбор данных. Через интерфейсы прикладного уровня (зеленые прямоугольные блоки в верхней части *рисунка 1*) корпоративное приложение взаимодействует с рабочим процессом сбора данных с помощью данных, которые не зависят от носителя данных и в которых все детали взаимодействия между компонентами сбора данных включены в эти данные. На более низком уровне рабочий процесс сбора данных способен различать взаимодействие со сканерами штрихового кода, устройствами

опроса радиочастотных меток, с процессом ручного ввода и т. д., но интерфейсы передачи (зеленые прямоугольные блоки в средней части *рисунка 1*) защищают рабочий процесс сбора данных от детализации работы низкоуровневого аппаратного обеспечения, сведений о том, как именно функционируют носители данных. Низкоуровневые интерфейсы (зеленые прямоугольные блоки внизу *рисунка 1*) отображают внутренние подробные сведения о носителе данных. Стандарт EPCIS и уровень «обмена данными» в общем случае отличаются от элементов на уровне сбора данных в трех ключевых аспектах:

1) Сервисы EPCIS имеют дело с накопленными данными (в дополнение к текущим данным). Уровень сбора данных, напротив, ориентирован исключительно на обработку собранных данных в реальном времени.

2) Сервисам EPCIS часто приходится иметь дело не только с исходными данными, полученными от носителей данных, таких как символы штрихового кода и радиочастотные метки, но также с ситуациями, когда процесс сбора данных приобретает дополнительный смысл, связанный с физическим или цифровым миром, а также с конкретными этапами в операционных или аналитических бизнес-процессах. Уровни сбора данных по своей сути носят чисто наблюдательный характер. Несмотря на то, что событие EPCIS содержит большую часть тех же данных группы стандартов «идентификация», что и событие «фильтрации и накопления данных» (Filtering & Collection) или сканирование символа штрихового кода, оно находится на семантически более высоком уровне, поскольку включает в себя понимание контекста ведения хозяйственной деятельности, в котором были получены данные с идентификаторами. Кроме того, не существует никаких требований в отношении непосредственной связи события EPCIS с конкретным физическим носителем данных. Например, событие EPCIS может свидетельствовать о том, что скоропортящийся предмет торговли только что превысил срок годности. Такое событие может целиком генерироваться программным обеспечением.

3) EPCIS задействуют в корпоративных средах информационных технологий на уровне, который является гораздо более разнообразным и многоцелевым, чем уровень сбора данных, где, как правило, системы автономны и служат единственной цели ведения хозяйственной деятельности. Отчасти, и что самое важное, это связано с желанием разделить данные EPCIS между предприятиями, которые, вероятно, имеют различные решения, развернутые для выполнения аналогичных задач. Отчасти это также связано с постоянной природой данных EPCIS. И наконец, это связано с тем, что настоящий стандарт EPCIS находится на самом высоком уровне общей архитектуры, и, следовательно, в естественной точке входа в другие корпоративные системы, которые широко варьируются от одного предприятия к другому (или даже в пределах частей одного и того же предприятия).

2.3 Стандарт EPCIS во взаимосвязи с торговыми партнерами

Стандарты GS1 на уровне «обмена данными» относятся к трем категориям данных, которые разделены между конечными пользователями (*см. таблицу 1*).

Таблица 1

Данные	Описание	Стандарты GS1
Мастер-данные	Данные, используемые одним торговым партнером совместно с многими торговыми партнерами, которые предоставляют описательные атрибуты объектов реального мира, определяемые идентификационными ключами GS1, включая предметы торговли, партии товаров и физические места нахождения	GDSN
Данные транзакции	Торговые транзакции, запускающие или подтверждающие выполнение функции в рамках бизнес-процесса, как это определено явным (например, договором на поставку) или неявным (например, таможенное оформление) соглашением о ведении хозяйственной деятельности, с самого начала бизнес-процесса (например, заказ продукции) до его завершения (например, окончательный расчет), а также использующие идентификационные ключи GS1	GS1 XML EANCOM
Данные о событиях, обеспечивающих прозрачность	Подробные сведения о физической или цифровой деятельности в цепи поставок продукции и других активов, идентифицированных с помощью ключей, в которых детализируется, где эти объекты находятся в данный момент времени и на каком основании. Не только в пределах одной организации, но также и между различными организациями	EPCIS

Данные документов транзакции и данные о событии, обеспечивающем прозрачность, имеют особенность, заключающуюся в том, что новые документы этих типов создаются постоянно в процессе хозяйственной деятельности в цепи поставок, даже если не создаются никакие новые объекты реального мира. Мастер-данные, напротив, более статичны: мастер-данные для данного объекта изменяются очень медленно (если вообще изменяются), а количество мастер-данных возрастает только по мере создания новых объектов, и не только потому, что существующие объекты участвуют в бизнес-процессах. Например, по мере того, как экземпляр определенного предмета торговли перемещается по цепи поставок, новые данные документов транзакции и данные о событиях, обеспечивающих прозрачность, генерируются в те моменты, когда экземпляр подвергается бизнес-транзакциям (таким, как покупка и продажа) и физическим процессам обработки (упаковывание, комплектация, складирование, и т. д.). Однако новые мастер-данные создаются только тогда, когда к цепи поставки добавляется новый предмет торговли или место нахождения.

Рисунок 2 иллюстрирует поток данных между торговыми партнерами, выделяя части настоящего стандарта EPCIS, участвующие в потоке данных о событиях, обеспечивающих прозрачность.

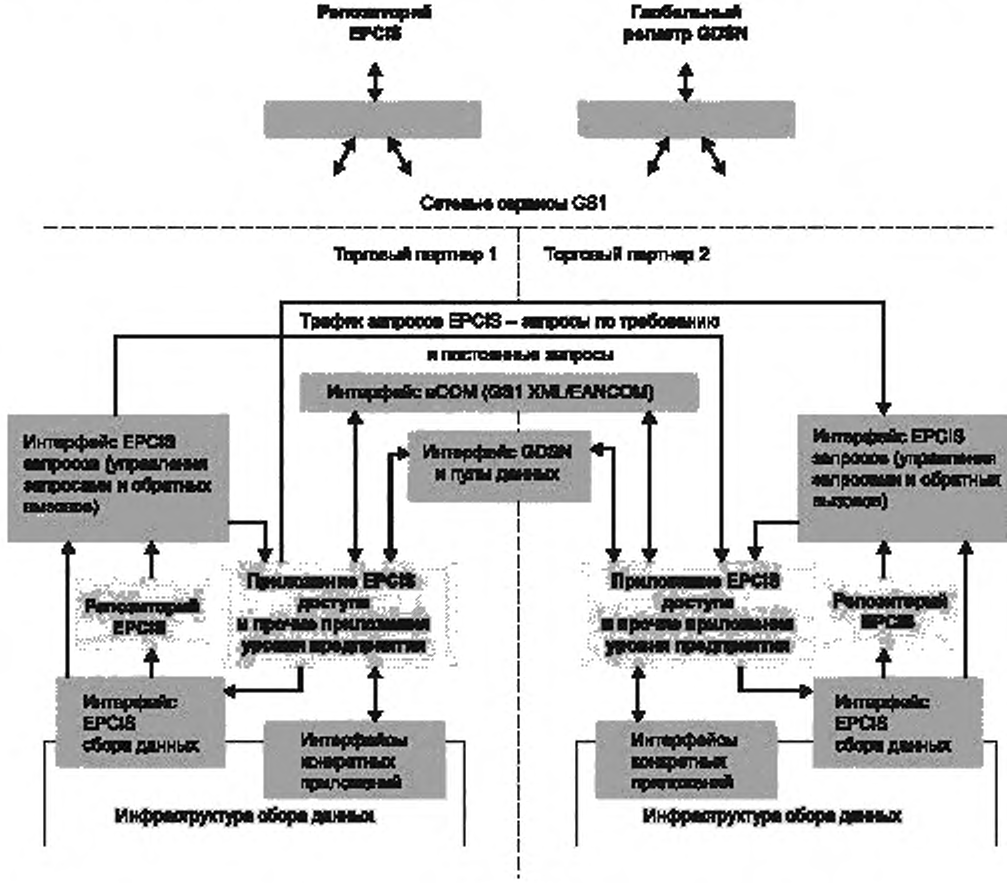


Рисунок 2

В дополнение к описанному выше использованию интерфейса запросов EPCIS торговые партнеры могут по взаимному согласию также использовать структуру документа EPCIS, определенную в подразделе 9.3, в качестве средства передачи набора событий EPCIS, при необходимости дополненного соответствующими мастер-данными, в виде единого электронного документа.

2.4 Стандарт EPCIS во взаимосвязи с другими компонентами архитектуры системы GS1

Ниже приведено описание того, за что ответственен каждый элемент архитектуры системы GS1, проиллюстрированной на рисунках в предыдущих разделах. Дополнительную информацию можно найти в документе [9], из которого заимствованы вышеприведенная схема и большая часть текста, а также в документе [8], из которого заимствована большая часть нижеследующего текста подраздела 3.4:

- *Устройства считывания штрихового кода и радиочастотных меток.* Производят опрос радиочастотных меток в момент их нахождения в зоне считывания, а также сканирование символов штрихового кода после запуска процесса считывания.

- *Интерфейс низкоуровневого протокола устройства считывания радиочастотной идентификации (Low-Level RFID Reader Protocol (LLRP) Interface).* Определяет порядок контроля и передачи первичных данных радиочастотных меток от устройств считывания радиочастотной идентификации к устройству, выполняющему роль фильтрации и накопления данных. События в этом интерфейсе имеют следующий формат: «устройство считывания А обнаружило номер EPC X в момент времени Т».

- *Фильтрация и накопление.* Эта функция фильтрует и накапливает сведения о первичных актах считывания радиочастотных меток через интервалы времени, ограниченные событиями, определяемыми приложением EPCIS сбора данных (например, срабатывание датчика движения). Для считывания символов штрихового кода подобной функции, как правило, не существует, так как устройства считывания символов штрихового кода обычно производят единичное считывание при запуске.

- *Интерфейс фильтрации и накопления данных (событий уровня приложения, Application-Level Events–ALE).* Определяет порядок контроля и передачи отфильтрованных и накопленных данных радиочастотных меток от функции фильтрации и накопления данных к функции рабочего процесса сбора данных. События в этом интерфейсе имеют следующий формат: «на логическом устройстве считывания L, в интервале времени между Т1 и Т2, были обнаружены следующие номера EPC», где список номеров EPC не имеет дубликатов и отфильтрован по критерию, определенному приложением EPCIS сбора данных. В случае символов штрихового кода подобные данные направляются к функции рабочего процесса сбора данных непосредственно от устройства считывания символов штрихового кода в форме строки элемента (Element String) GS1.

- *Рабочий процесс сбора данных.* Управляет функционированием элементов архитектуры нижнего уровня, а также предоставляет контекст хозяйственной деятельности путем координации с другими источниками информации, участвующими в выполнении конкретного этапа бизнес-процесса. Рабочий процесс сбора данных может, например, координировать конвейерную систему с событиями типа «фильтрация и накопление данных» и актами считывания символов штрихового кода, может проверить наличие исключительных условий и принять меры по исправлению ситуации (например, перенаправление дефектных объектов в область доработки), может представлять информацию для человека-оператора, и т. д. Рабочий процесс сбора данных воспринимает этап или этапы бизнес-процесса, во время которых происходит сбор данных о событиях EPCIS. Эта функция может быть сложной, предполагающей объединение нескольких событий фильтрации и накопления данных и/или актов считывания символов штрихового кода с одним или несколькими событиями хозяйственной деятельности, как это происходит при погрузке партии товара, или может быть такой же простой, как в бизнес-процессе инвентаризации, где могут применяться устройства считывания, которые сканируют объекты, поступающие на полку или покидающие ее. Здесь событие уровня фильтрации и накопления данных, или акты считывания символов штрихового кода, и событие уровня EPCIS могут быть настолько похожи, что на уровне рабочего процесса сбора данных требуется незначительная обработка, и рабочий процесс сбора данных просто конфигурирует и маршрутизирует события от интерфейса фильтрации и накопления данных и/или устройств считывания символов штрихового кода непосредственно через интерфейс сбора данных EPCIS в репозиторий EPCIS или в приложение для хозяйственной деятельности. В рамках настоящего стандарта рабочий процесс сбора данных, основные выходные данные которого состоят из событий EPCIS, называют «приложением EPCIS сбора данных» (EPCIS Capturing Application).

- *Интерфейс EPCIS.* Представляет собой интерфейс, через который производится передача данных EPCIS к функциям на уровне предприятий, включая репозитории EPCIS (EPCIS Repositories), приложения EPCIS доступа (EPCIS Accessing Applications), а также обмен данными с партнерами. События в таких интерфейсах выражают, например, таким образом: «В месте нахождения X, в момент времени Т, были проверены следующие объекты (коробки), содержащиеся в объединяющем их объекте (поддоне)». В этом случае присутствуют три интерфейса EPCIS. Интерфейс сбора данных определяет доставку событий EPCIS от приложений сбора данных к другим функциям, которые используют данные в

реальном времени, включая репозитории EPCIS, а также производит принудительную передачу данных (режим «push» («принудительная доставка»)) в режиме реального времени к приложениям доступа и к торговым партнерам. Интерфейс управления запросами EPCIS (EPCIS Query Control Interface) определяет средства получения последовательности данных EPCIS для приложений доступа и торговых партнеров, как правило, путем взаимодействия с репозиторием EPCIS. Интерфейс управления запросами EPCIS предоставляет два режима взаимодействия. В режиме «по требованию» или в синхронном режиме клиент делает запрос через интерфейс управления запросами EPCIS и сразу же получает ответ. В режиме «длительный запрос» или асинхронном режиме клиент устанавливает подписку для периодического запроса. Каждый раз после выполнения периодического запроса его результаты асинхронно (или принудительно, режим «push» («принудительная доставка»)) доставляются получателю через интерфейс обратных вызовов по запросам EPCIS» (EPCIS Query Callback). Интерфейс обратных вызовов по запросам EPCIS может также использоваться для немедленной передачи информации. Это соответствует стрелке «возможный параллельный путь для принудительной доставки в режиме реального времени» на *рисунке 1*. Обязательные требования к трем интерфейсам EPCIS определены в настоящем стандарте.

- *Приложение EPCIS доступа.* Отвечает за выполнение всех бизнес-процессов предприятия, таких как управление складом, отгрузка и приемка, временной анализ пропускной способности и так далее, опираясь на данные, связанные с номером EPC.

- *Репозиторий EPCIS.* Фиксирует события уровня EPCIS, генерируемые одним или несколькими приложениями EPCIS сбора данных, а также делает их доступными для дальнейших запросов с помощью приложений EPCIS доступа.

- *Партнерское приложение.* Системы торгового партнера, которые выполняют те же функции, что и приложения EPCIS доступа, но извне по отношению к сети партнерской стороны. Партнерским приложениям может быть предоставлен доступ к той части информации, которая доступна из приложения EPCIS сбора данных или внутри репозитория EPCIS.

Интерфейсы внутри данного множества разработаны для изолирования высоких уровней архитектуры от необязательных подробных сведений о том, как реализованы более низкие уровни. Один из способов понять это — рассмотреть, что происходит, если произведены определенные изменения:

- *Низкоуровневый протокол устройства считывания радиочастотной идентификации (LLRP) и строка элемента GS1* изолируют более высокие уровни от необходимости знать, какие протоколы радиочастотной идентификации или символики штрихового кода используются, а также какие марки/модели устройств считывания были выбраны. Если вместо них устанавливаются другие устройства считывания, то информация, посылаемая через их интерфейсы, остается той же самой.

- В ситуациях, когда используется радиочастотная идентификация, интерфейс фильтрации и накопления данных изолирует более высокие уровни от выбора реализации физического уровня, выполненной в отношении того, как радиочастотные метки обнаруживаются и накапливаются и как определяются временные границы событий. Так, если одно устройство считывания радиочастотной идентификации с четырьмя антеннами заменяется на группу из пяти устройств считывания типа «smart antenna» («умная антенна»), оснащенных единственной антенной, события на уровне фильтрации и накопления данных остаются теми же.

- Сервисы EPCIS защищают корпоративные приложения от необходимости понимания деталей выполнения отдельных этапов бизнес-процесса. Например, типовым событием EPCIS является следующее: «В месте нахождения X, в момент времени T, были проверены следующие коробки, находившиеся на следующем поддоне». В реализации хозяйственной деятельности конвейерного типа это может соответствовать одному событию уровня фильтрации и накопления данных, в котором данные актов считывания накапливаются в течение интервала времени, начало и конец которого регулируются событием пересечения объектом зоны контроля электронных оптических датчиков, установленных на конвейере и объединенных с устройством считывания. В то же время в другой реализации могут быть привлечены три физически крепких человека, которые будут перемещать объекты и использовать мобильные ручные устройства считывания для считывания радиочастотных меток. На уровне фильтрации и накопления данных эти события выглядят иначе (каждое срабатывание ручного мобильного устройства считывания похоже на отдельное событие фильтрации и накопления данных), и обработка, выполненная приложением EPCIS сбора данных, также отличается (возможно привлечение интерактивного пульта управления, который персонал использует для проверки работы). Однако событие EPCIS остается все тем же для обеих реализаций.

Таким образом, данные на уровне EPCIS отличаются от данных, используемых на уровне сбора данных (Capture layer) в архитектуре системы GS1, за счет включения семантической информации о бизнес-процессе, в рамках которого осуществляется сбор данных, а также предоставления истории наблюдений. При этом сервисы EPCIS изолируют приложения, которые потребляют эту информацию, от понимания подробных сведений, связанных с выполнением конкретного этапа бизнес-процесса на низком уровне.

3 Принципы спецификации EPCIS

Факторы, приведенные в двух предыдущих разделах, показывают, что требования, предъявляемые к стандартам на уровне EPCIS, значительно сложнее, чем на уровне сбора данных архитектуры системы GS1. Историческая природа данных EPCIS предполагает, что для интерфейсов EPCIS требуется более богатый набор методов доступа, чем для интерфейсов событий уровня приложения (ALE) или радиочастотной идентификации (RFID) и устройств считывания символов штрихового кода. Включение эксплуатационного контекста или контекста бизнес-процесса в EPCIS подразумевает то, что спецификации EPCIS имеют дело с более богатым набором типов данных, и, кроме того, они должны быть гораздо более открытыми для расширения с учетом широкого спектра бизнес-процессов, существующих в мире. Функционирование сервисов EPCIS в разнообразной среде подразумевает, что в стандарте EPCIS уровни проработаны настолько тщательно, что даже тогда, когда сервисы EPCIS используются между внешними системами, которые значительно отличаются в деталях работы, должна существовать согласованность и функциональная совместимость на уровне представления абстрактных структур данных, а также значений этих данных.

В ответ на эти требования стандарт EPCIS представляет собой рамочную спецификацию и более узкие и детальные спецификации, которые ее наполняют. Указанная рамочная спецификация должна быть:

- *Многоуровневой*. В частности, структура и значение данных в абстрактном смысле задаются отдельно от конкретных деталей услуг доступа к данным и привязок к конкретным протоколам интерфейса. Это допускает вариации в конкретных деталях с течением времени и между предприятиями, сохраняя при этом общий смысл самих данных. Это также допускает повторное использование спецификаций данных EPCIS в других подходах, отличных от ориентированного на сервисы подхода настоящей спецификации. Например, определения данных могут быть повторно использованы в рамках электронного обмена данными.

- *Расширяемой*. Основные спецификации предоставляют не только базовый набор типов данных и операций, но также предусматривают некоторые средства, с помощью которых базовый набор может быть расширен для целей, характерных для данной отрасли или области применения. Расширения отвечают не только собственным требованиям, которые предусматривают использование, насколько это возможно, большего числа стандартных структур, но и позволяют стандартам развиваться и расширяться с течением времени естественным путем.

- *Модульной*. Механизмы образования уровней и механизмы расширяемости позволяют устанавливать требования к различным частям общей структуры EPCIS в разных документах, а также обеспечивать согласованность в рамках всей структуры. Это способствует расширению процесса стандартизации (так же, как и процесса реализации).

Остальная часть настоящего стандарта определяет структуру EPCIS. Она также описывает базовый набор типов данных и интерфейсов данных, заполняющих данную структуру. Сопровождающий стандарт базовой деловой лексики GS1 по *ГОСТ Р 59168* предоставляет дополнительные определения данных в дополнение к предусмотренным настоящим стандартом.

4 Терминология и условные обозначения

В рамках настоящего стандарта термины ДОЛЖЕН (SHALL), НЕ ДОЛЖЕН (SHALL NOT), СЛЕДУЕТ (рекомендуется) (SHOULD), НЕ СЛЕДУЕТ (не рекомендуется) (SHOULD NOT), МОЖЕТ (разрешается) (MAY), НЕ ОБЯЗАТЕЛЬНО ДОЛЖЕН (не требуется) (NEED NOT), МОЖЕТ (способен) (CAN) и НЕ МОЖЕТ (не способен) (CANNOT) будут интерпретироваться согласно [*10*, приложение G]. В случае использования вышеуказанным образом данные термины будут всегда приведены в записи ПРОПИСНЫМИ БУКВАМИ; в случае записи строчными буквами их следует понимать в обычном значении слов.

Все разделы настоящего документа, за исключением разделов 2, 3[1)] и 4, являются нормативными, кроме случаев, когда в тексте документа прямо указано иное.

В настоящем документе используются следующие соглашения о типографских обозначениях:

- записи ПРОПИСНЫМИ БУКВАМИ используются для специальных терминов, приведенных из вышеуказанной спецификации *[10]*;

- шрифт `Courier` используется для выделения идентификаторов на языке программирования, унифицированном языке моделирования (UML) и расширяемом языке разметки (XML), а также в тексте XML-документов.

➢ Места для внесения изменений, которые необходимо произвести в настоящем документе до достижения им завершающей стадии, соответствующей одобренному стандарту GS1, предваряются знаком в виде обращенной вправо стрелки, аналогичной указанной в начале настоящего абзаца.

5 Структура спецификации EPCIS

Спецификация EPCIS является многоуровневой, расширяемой и модульной.

5.1 Уровни

Структура спецификации EPCIS представлена на нескольких уровнях, как показано *на рисунке 3*.

Уровни (*представленные на рисунке 3*) описаны ниже:

- *Уровень абстрактной модели данных (Abstract Data Model Layer)*. Уровень абстрактной модели данных определяет общую структуру данных EPCIS. Это единственный уровень, который не расширяется никакими другими механизмами, кроме как пересмотром самой спецификации EPCIS. Уровень абстрактной модели данных определяет общие требования к созданию определений данных в рамках уровня определения данных.

- *Уровень определения данных (Data Definition Layer)*. Уровень определения данных указывает, какими данными обмениваются через EPCIS, какова их абстрактная структура и что она означает. В рамках данной спецификации представлен один модуль определения данных, называемый модулем типов основных событий (Core Event Types Module). Описания данных на уровне определения данных представлены абстрактно, следуя правилам, установленным на уровне абстрактной модели данных.

- *Сервисный уровень (Service Layer)*. Сервисный уровень определяет служебные интерфейсы, через которые взаимодействуют клиенты EPCIS. В данной спецификации описаны два модуля сервисного уровня. Модуль основных операций сбора данных (Core Capture Operations Module) определяет служебный интерфейс (интерфейс сбора данных EPCIS), через который приложения сбора данных доставляют типы основных событий (Core Event Types) заинтересованным сторонам. Модуль основных операций с запросами (Core Query Operations Module) определяет служебные интерфейсы: интерфейс EPCIS управления запросами и интерфейс EPCIS обратных вызовов по запросам, которые используют приложения EPCIS доступа для получения предварительно собранных данных. Определения интерфейсов сервисного уровня даны абстрактно с использованием языка UML.

- *Привязки (Bindings)*. Привязки определяют конкретные реализации уровня определения данных и сервисного уровня. Для любого определения данных или сервисного модуля может быть определено множество привязок. В данной спецификации для трех модулей, указанных в определении данных и сервисных уровнях, заданы в общей сложности девять привязок. Определения данных в модуле определения данных типов основных событий получают привязку к схеме XML. Интерфейс EPCIS для сбора данных в модуле основных операций сбора данных получает привязки к очереди сообщений (Message Queue) и HTTP. Интерфейс EPCIS для управления запросами в модуле основных операций с запросами (Core Query Operations Module) получает привязку к протоколу SOAP через HTTP с помощью языка описания веб-сервисов WSDL, а также вторую привязку для протокола AS2. Интерфейс обратных вызовов по запросам в модуле основных операций с запросами получает привязки к протоколам HTTP, HTTPS и AS2.

- *Стандарт базовой деловой лексики GS1 (GS1 Core Business Vocabulary Standard)*. Стандарт базовой деловой лексики (БДЛ) по *ГОСТ Р 59168* является дополнением к настоящему стандарту EPCIS. Он определяет конкретные лексические элементы, которые могут быть использованы для заполнения определений данных, указанных на уровне определения данных EPCIS. В то время как EPCIS может применяться без БДЛ, используя только конфиденциальные или частные значения данных, для приложений EPCIS гораздо выгоднее максимально использовать стандарт БДЛ *по ГОСТ 59168*.

[1)] *В [1] приведена ошибочная ссылка на «раздел 33».*

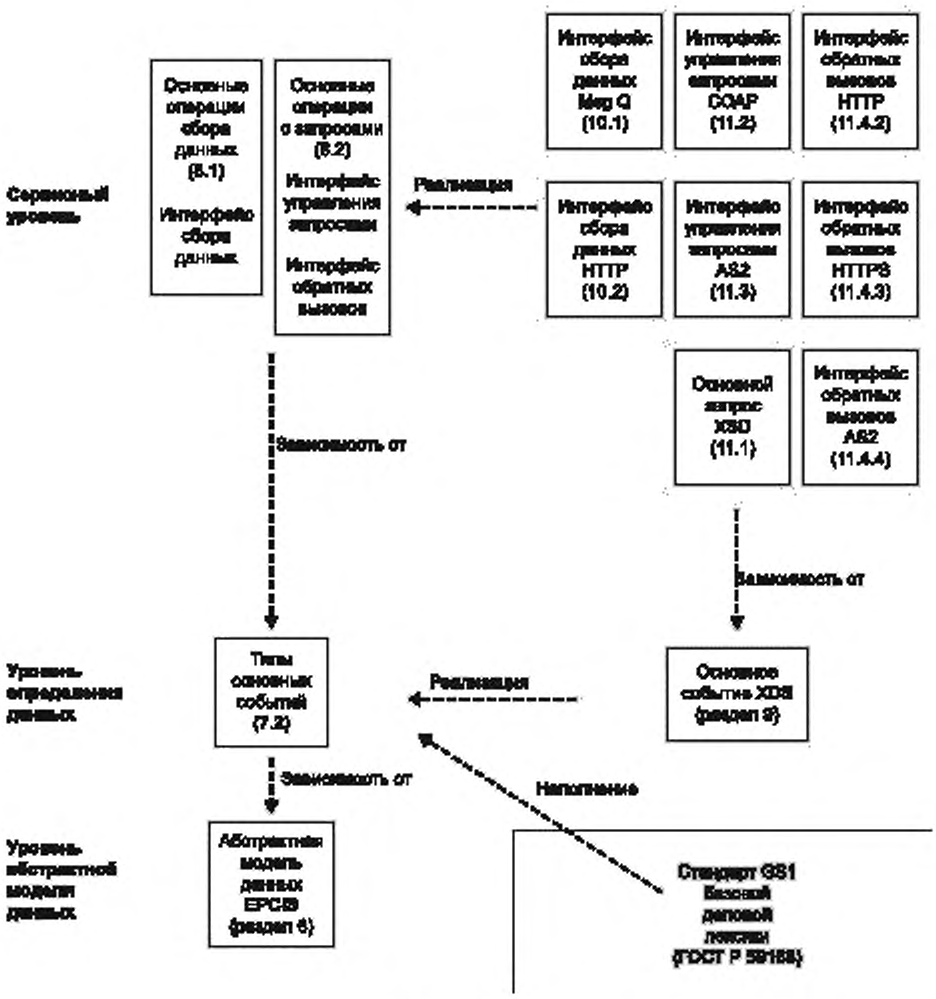


Рисунок 3

5.2 Расширяемость

Многоуровневая методика спецификации способствует расширяемости, поскольку один уровень может быть повторно использован более чем в одном варианте реализации на другом уровне. Например, в то время, как данная спецификация включает в себя привязку XML к модулю определения данных типов основных событий, другая спецификация определяет привязку к такому же модулю с другим синтаксисом, например к файлу CSV[1)].

Помимо расширяемости, присущей многоуровневым структурам, спецификация EPCIS содержит несколько особых механизмов расширяемости:

- создание производного класса (Subclassing). Данные на уровне определения данных задают с использованием языка UML, который позволяет представить определение новых данных с помощью создания подкласса существующего определения. Подкласс — это новый тип данных, который включает в себя все поля существующего типа, дополняя его новыми полями. Экземпляр подкласса может использоваться в любом контексте, в котором предполагалось использовать экземпляр порождающего класса;

[1)] *CSV (от англ. Comma-Separated Values — значения, разделенные запятыми) — текстовый формат, предназначенный для представления табличных данных.*

- точки расширения (Extension Points). Определения данных и служебные спецификации включают также точки расширения, которые поставщики могут использовать для расширения функциональности без создания подклассов.

5.3 Модульность

Структура спецификации EPCIS разрабатывалась как модульная. То есть она состоит не из единственной спецификации, а скорее из набора отдельных спецификаций, которые связаны между собой. Это позволяет стандарту EPCIS расширяться и совершенствоваться в распределенном режиме. Многоуровневая структура и механизмы расширения обеспечивают необходимые составляющие для достижения модульности, как это происходит с группировкой в модули.

Поскольку спецификации EPCIS являются модульными, требований в отношении того, что границы модуля спецификаций в рамках реализаций EPCIS должны быть фиксируемыми или явными, не существует. Примером может служить конкретный программный продукт, который представляет собой основанную на протоколах SOAP/HTTP реализацию ассоциативного сервиса коробка — поддон и сервиса каталога продукции, передающего данные, указанные в соответствующих модулях определения данных. Этот продукт может соответствовать целым шести различным модулям из стандарта EPCIS: модулю определения данных, описывающему данные каталога продукции, модулю определения данных, который задает объединение коробка — поддон, спецификации для соответствующих сервисов, а также соответствующих привязок SOAP/HTTP. Но исходное кодовое значение изделия может не обеспечивать прослеживания этих границ, и на самом деле конкретная схема базы данных, используемая для изделия, может внести дезорганизацию в данные таким образом, что данные каталога продукции и данные объединения коробка — поддон оказываются неразрывно переплетены. Однако, пока конечный результат соответствует спецификациям, данная реализация допускается.

6 Уровень абстрактной модели данных

В настоящем разделе дается нормативное описание абстрактной модели данных, которая лежит в основе стандарта EPCIS.

6.1 Данные о событии и мастер-данные

В целом стандарт EPCIS имеет дело с двумя видами данных: данные о событиях (event data) и мастер-данные (master data). Данные о событиях возникают в ходе выполнения бизнес-процессов, собираются через интерфейс EPCIS сбора данных (EPCIS Capture Interface) и становятся доступными для запроса через интерфейсы EPCIS запросов (EPCIS Query Interfaces). Мастер-данные являются дополнительными данными, которые обеспечивают необходимый контекст для интерпретации данных о событиях. Они доступны для запроса через интерфейс EPCIS управления запросами (EPCIS Query Control Interface), но средства, с помощью которых мастер-данные поступают в систему, не указаны в стандарте EPCIS.

Уровень абстрактной модели данных не пытается определить значение терминов «данные о событиях» или «мастер-данные», а лишь предоставляет точные определения структур данных, которые используются спецификацией стандарта EPCIS. Ответственность за моделирование информации о хозяйственной деятельности, связанной с объектами реального мира, такими как данные о событиях и мастер-данные, несет уровень определения данных, а также отраслевые соглашения и соглашения конечных пользователей, которые основываются на данном документе.

Примечание — В то время как для целей настоящей спецификации термины «данные о событиях» и «мастер-данные» означают не более чем «данные, которые соответствуют представленной здесь структуре», структуры, определенные на уровне абстрактной модели данных, предназначены для обеспечения надлежащего представления для данных, обычно требующих обмена через EPCIS. Неформально эти два типа данных можно понимать следующим образом. Объем данных о событиях растет по мере роста количества бизнес-транзакций, а также соотносится с ситуациями, которые происходят в конкретные моменты времени. Примером данных о событиях является: «В 13:23 15 марта 2004 года номер EPC X был обнаружен в месте нахождения L». Объем мастер-данных обычно не увеличивается только лишь с ростом числа бизнес-транзакций (хотя мастер-данные имеют тенденцию возрастать одновременно с увеличением размера организации), как правило, они не привязаны к конкретным моментам времени (хотя мастер-данные могут медленно изменяться с течением времени), а также обеспечивают интерпретацию элементов данных о событиях. Примером мастер-данных является: «Место нахождения L относится к распределительному центру, расположенному по адресу 123 Elm Street, Anytown, USA». Все данные в наборе вариантов использования, рассмотренных при создании стандарта EPCIS, могут быть смоделированы как совокупность данных о событиях и мастер-данных такого рода.

Структура данных о событиях и мастер-данных EPCIS показана на *рисунке 4*. (Следует обратить внимание, что это только пример: специфические лексические элементы и имена атрибутов мастер-данных, показанных на этом рисунке, не определены в настоящей спецификации.)

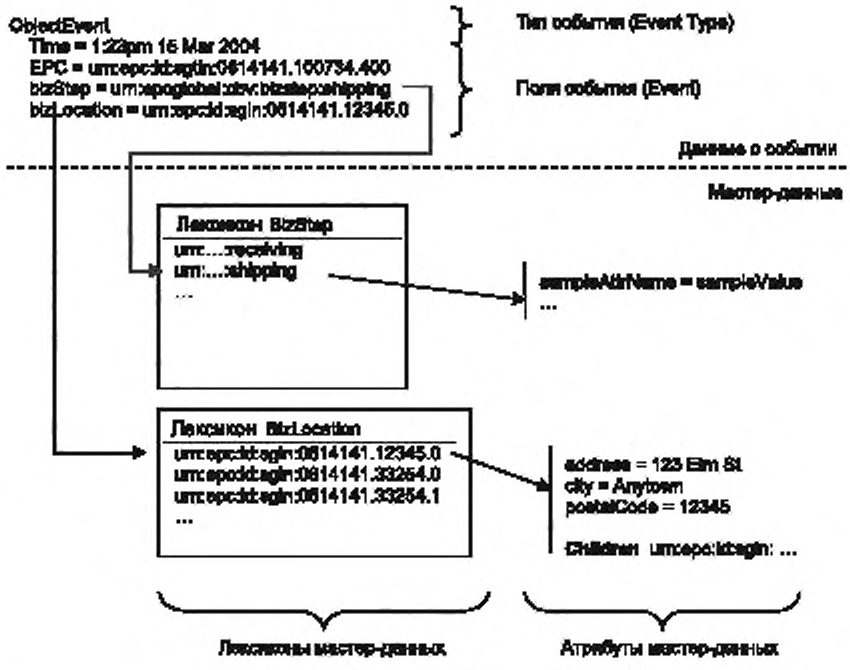


Рисунок 4

Составные части абстрактной модели данных EPCIS определены ниже:

- *Данные о событии (Event Data)*: набор событий.

- *Событие (Event)*: структура, состоящая из типа события и одного или нескольких упомянутых полей события.

- *Тип события (Event Type)*: имя, ограниченное пространством имен qname (*qualified name — составное имя*) и показывающее, которой из нескольких возможных структур (заданных на уровне определения данных) соответствует данное событие.

- *Поле события (Event Field)*: поименованное поле в рамках события. Имя этого поля задается с помощью пространства имен qname со ссылкой либо на имя поля, заданного на уровне определения данных, либо на имя поля, заданного как расширение к данной спецификации. Значение этого поля может представлять собой простой тип (такой, как целый (integer) или метка времени (timestamp)), лексический элемент или список простых типов лексических элементов.

- *Мастер-данные (Master data)*: набор лексиконов вместе с атрибутами мастер-данных, связанных с элементами этих лексиконов.

- *Лексикон (Vocabulary)*: поименованный набор идентификаторов. Имя лексикона принадлежит пространству имен qname и может использоваться как имя типа для поля события. Идентификаторы внутри лексикона называются лексическими элементами. Лексикон представляет собой набор альтернативных значений, которые могут отображаться в виде значений конкретных полей события. Лексиконы в EPCIS используются для моделирования таких наборов, как набор имен доступных мест нахождения, набор доступных имен этапов бизнес-процессов и так далее.

- *Лексический элемент (Vocabulary Element)*: идентификатор, который именует одну из альтернатив, моделируемых с помощью лексикона. Значение поля события может быть лексическим элементом. Лексические элементы представлены как унифицированные идентификаторы ресурсов (URIs). Каждый лексический элемент может быть связан с атрибутами мастер-данных.

- *Атрибуты мастер-данных (Master data attributes)*: неупорядоченный набор пар имя/значение, связанный с индивидуальным лексическим элементом. Имя, как часть этой пары, принадлежит пространству имен qname. Значение, как часть той же пары, может быть значением произвольного типа. Специальным атрибутом (возможно, пустым) является список потомков (children), причем каждый из потомков (child) является еще одним лексическим элементом этого же лексикона (см. 6.5).

Новые события EPCIS генерируются на границе и попадают в инфраструктуру EPCIS через интерфейс EPCIS сбора данных, где они могут быть впоследствии доставлены заинтересованным приложениям через интерфейсы EPCIS запросов. Здесь нет механизма, предусмотренного в любом интерфейсе, с помощью которого приложение может удалить или изменить событие EPCIS. Единственным способом «отменить» или «исправить» событие EPCIS является генерация последующего события, смыслом в отношении хозяйственной деятельности которого является аннулирование или исправление действия предшествующего события (в 7.4.1.3 указано, как это может быть реализовано).

В то время как интерфейс EPCIS сбора данных и интерфейсы EPCIS запросов не предоставляют приложению никаких средств для явного запроса на удаление события, репозитории EPCIS МОГУТ применять правила хранения данных, при которых по истечении некоторого периода времени устаревшие события EPCIS становятся недоступными.

Мастер-данные, напротив, могут изменяться с течением времени, хотя такие изменения, как предполагается, будут нечастыми по отношению к скорости, с которой генерируются новые данные о событии. Текущая версия настоящей спецификации не определяет, каким образом изменяются мастер-данные (а также, как было замечено выше, не определяет, как первоначально мастер-данные вводятся).

6.1.1 Передача мастер-данных в EPCIS

Интерфейсы EPCIS сбора данных и интерфейсы EPCIS запросов главным образом относятся к передаче событий EPCIS. В стандарте EPCIS не указаны средства, с помощью которых мастер-данные вводятся в систему, реализующую данные интерфейсы. Однако стандарт EPCIS предоставляет механизмы передачи мастер-данных, используя которые приложение обеспечивает получателю данных о событии EPCIS доступ к мастер-данным, необходимым для интерпретации этих данных о событии. Мастер-данные могут также передаваться иными способами, абсолютно не входящими в область применения настоящего стандарта EPCIS. Стандарт EPCIS не устанавливает каких-либо требований относительно того, должны ли данные о событии EPCIS сопровождаться мастер-данными, кроме требования в отношении того, что мастер-данные, сопровождающие данные о событиях, должны соответствовать любым мастер-данным в разделах МДЭС.

Настоящий стандарт EPCIS устанавливает четыре механизма передачи мастер-данных, описание которых приведено в таблице 2:

Таблица 2

Механизм	Структурный элемент	Описание	Ограничения
Запрос мастер-данных	8.2.7.2	Клиент запроса EPCIS может запросить у приложения интерфейса EPCIS запросов мастер-данные, соответствующие конкретным критериям	Мастер-данные, полученные в ответ на запрос, ДОЛЖНЫ отражать текущие значения атрибутов мастер-данных, известные стороне, отвечающей на запрос, на момент создания/формирования ответа на запрос
Мастер-данные на уровне экземпляра/серии (МДЭС)	7.3.6	Событие EPCIS, обозначающее начало жизненного цикла идентификатора на уровне экземпляра или на уровне серии, может содержать соответствующие мастер-данные непосредственно в самом событии	Мастер-данные в событии ДОЛЖНЫ отражать текущие значения атрибутов мастер-данных, известные стороне, являющейся создателем события, на момент осуществления события. Следует заметить, что, поскольку эти данные встроены непосредственно в само событие, они являются постоянной частью этого события и будут всегда включаться в информацию о событии, направляемую в ответ на запросы о событии (предмет редактирования, как описано в 8.2.2)

Окончание таблицы 2

Механизм	Структурный элемент	Описание	Ограничения
Заголовок документа EPCIS	9.5	Документ EPCIS, используемый для передачи набора событий EPCIS из пункта в пункт вне интерфейса EPCIS запросов, может включать соответствующие мастер-данные в заголовке документа	Мастер-данные в заголовке документа ДОЛЖНЫ отражать текущие значения атрибутов мастер-данных, известные стороне, являющейся создателем документа, на момент создания документа. Мастер-данные в заголовке документа EPCIS НЕ ДОЛЖНЫ указывать значения атрибутов, противоречащие разделу МДЭС какого-либо из событий, содержащихся в основной части документа EPCIS
Документ мастер-данных EPCIS	9.7	Для передачи мастер-данных из пункта в пункт вне интерфейса EPCIS запросов может быть использован документ мастер-данных EPCIS	Мастер-данные в документе ДОЛЖНЫ отражать текущие значения атрибутов мастер-данных, известные стороне, являющейся создателем документа, на момент создания документа

6.2 Категории лексики

В стандарте EPCIS лексиконы широко используются для моделирования физических, цифровых, а также понятийных объектов, которые существуют в реальном мире. Примерами лексиконов, установленных на основном уровне определения данных EPCIS, являются названия мест нахождения, наименования классов объектов (наименование класса объекта — это наименование типа «изделие Acme Deluxe Widget» в отличие от номера EPC, который обозначает конкретные экземпляры изделия Acme Deluxe Widget), а также наименования бизнес-этапов. Лексикон всегда представляет собой конечный (но тем не менее допускающий изменения) набор альтернатив, которые могут появиться в конкретных полях событий.

Рекомендуется различать две категории лексики, которые представляют собой различные модели того, как они определяются и расширяются с течением времени:

- *Типовая лексика (Standard Vocabulary):* типовая лексика представляет собой набор лексических элементов, определения и значения которых должны быть согласованы торговыми партнерами, которые будут обмениваться данными о событиях с использованием этой лексики. Например, уровень определения мастер-данных EPCIS (EPCIS Core Data Definition Layer) определяет лексикон, называемый «бизнес-этапы» (business step), элементами которого являются идентификаторы, обозначающие такие понятия, как «отгрузка», «приемка» и так далее. Один из торговых партнеров может создать событие с бизнес-этапом «отгрузка», а другой партнер, получив это событие посредством запроса, сможет интерпретировать его в силу предшествующего соглашения о смысловом значении лексического элемента «отгрузка».

Элементы типовой лексики, как правило, определяются организациями, объединяющими множество конечных пользователей, такими как Ассоциация GS1, отраслевые сообщества вне GS1, частные объединения торговых партнеров и т. д. Мастер-данные, связанные с типовыми лексическими элементами (в случае, если такие мастер-данные вообще определяются), устанавливаются теми же организациями и, как правило, доводятся до пользователей в качестве части стандарта или другими подобными способами. Новые лексические элементы в рамках какого-либо типового лексикона обычно вводятся посредством специальной регламентированной процедуры, такой как ратификация новой версии стандарта или голосование в рамках отраслевого объединения. Несмотря на то, что отдельная организация конечных пользователей может самостоятельно ввести новый типовой лексический элемент, такой лексический элемент будет иметь ограниченное применение в условиях обмена данными и, вероятно, будет использоваться только в пределах этой организации.

- *Пользовательская лексика (User Vocabulary)*: пользовательская лексика представляет собой набор лексических элементов, определение и значение которых находятся под контролем одной организации. Например, уровень определения мастер-данных EPCIS определяет лексикон, называемый «производственное место назначения» («business location»), элементами которого являются идентификаторы, обозначающие такие объекты, как «организация Acme Corp., распределительный центр № 3».

Организация Acme Corp. может генерировать событие с местом нахождения, где проводится хозяйственная деятельность «организация Acme Corp., распределительный центр № 3», а другой партнер, получив это событие посредством запроса, может интерпретировать его, либо соотнеся с другими событиями, имеющими то же место нахождения, либо сравнивая с атрибутами мастер-данных, связанных с этим местом нахождения, либо используя оба указанных способа.

Элементы пользовательской лексики главным образом определяются отдельными организациями конечных пользователей, действующими независимо. Мастер-данные, связанные с элементами пользовательской лексики, как правило, определяют те же организации и обычно распространяют среди торговых партнеров посредством интерфейса EPCIS для запросов или других механизмов обмена/синхронизации данных. Новые лексические элементы в рамках какого-либо пользовательского лексикона вводятся исключительно по усмотрению конечного пользователя, и торговые партнеры должны быть готовы реагировать соответствующим образом. Обычно правила построения новых пользовательских лексических элементов устанавливаются организациями с большим числом конечных пользователей, и в любом случае они должны следовать правилам, указанным в 6.4.

Границы между этими двумя категориями лексики носят несколько субъективный характер. Однако механизмы, определенные в спецификации EPCIS, не делают абсолютно никакого различия между этими категориями лексики, и поэтому нет необходимости идентифицировать конкретный лексикон как принадлежащий одной или другой категории. Термины «типовая лексика» и «пользовательская лексика» введены только потому, что они полезны как подсказка при выборе способа определения и расширения конкретного лексикона.

Стандарт базовой деловой лексики GS1 (БДЛ) *ГОСТ Р 59168* предоставляет типовые лексические элементы для многих видов лексиконов, используемых в типах событий EPCIS. В частности, *ГОСТ Р 59168* определяет лексические элементы для следующих видов типовых лексиконов EPCIS: бизнес-этап (Business Step), состояние (Disposition), тип документа бизнес-транзакции (Business Transaction Type), а также тип начального/конечного пункта (Source/Destination Type). *ГОСТ Р 59168* также определяет шаблоны для построения лексических элементов для следующих видов пользовательской лексики EPCIS: объектов (EPC), объектов на уровне классов (EPCClass), мест нахождения (Location) [мест считывания (Read Point) и производственных мест назначения (Business Location)], идентификаторов документов бизнес-транзакции (Business Transaction ID), идентификаторов (ID) начальных/конечных пунктов (Source/Destination ID) и идентификаторов преобразований (Transformation ID).

6.3 Механизмы расширения

Ключевой особенностью стандарта EPCIS является способность к расширению различными организациями для адаптации к конкретным ситуациям, связанным с хозяйственной деятельностью. В целом уровень абстрактной модели данных предоставляет пять методов, которые перечислены здесь, начиная с наиболее агрессивного типа расширения до наименее агрессивного, с помощью которых могут быть расширены данные, обрабатываемые посредством сервисов EPCIS (кроме того, сервисный уровень предоставляет механизмы для добавления дополнительных услуг):

- *Новый тип события*. Новый тип события может быть добавлен на уровне определения данных. Добавление нового типа события требует расширения каждой привязки определения данных, а также может потребовать расширения интерфейсов для сбора данных и интерфейсов для запросов и их привязок.

- *Новое поле события*. Новое поле может быть добавлено к существующему типу события на уровне определения данных. Привязки, интерфейс для сбора данных и интерфейсы для запроса, определенные в настоящей спецификации, предназначены для обеспечения такого типа расширения без необходимости внесения изменений в саму спецификацию. (То же самое может быть неверно для других привязок или языков запросов, определенных вне указанной спецификации.)

- *Новый вид лексики*. Новый вид лексики *(лексикон)* может быть добавлен в набор существующих видов лексики. Никаких изменений в привязках или интерфейсах не требуется.

- *Новый атрибут мастер-данных*. Имя нового атрибута может быть определено для существующего лексикона. Никаких изменений в привязках или интерфейсах не требуется.

- *Новый атрибут мастер-данных на уровне экземпляра/серии* (МДЭС от Instance/Lot Master Data, ILMD). Имя нового атрибута может быть определено для использования в мастер-данных на уровне экземпляра/серии (см. 7.3.6). Никаких изменений в привязках или интерфейсах не требуется.

- *Новый лексический элемент*. Новый элемент может быть добавлен к существующему лексикону.

Уровень абстрактной модели данных был разработан таким образом, что большинство расширений, возникающих в связи с их принятием различными отраслями или в связи с углубленным понимани-

ем в рамках отдельной отрасли, могут быть размещены с помощью последних методов в приведенном выше списке, которые не требуют пересмотра самой спецификации. Тем не менее во главе списка приведены более агрессивные методы для случая возникновения ситуации, которая не может быть решена завершающими список методами.

Предполагается, что будет предложено несколько различных способов расширить спецификацию EPCIS, как это представлено ниже (*см. таблицу 3*):

Таблица 3

Как распространяется расширение	Ответственная организация	Метод расширения				
		Новый тип события	Новое поле события	Новый вид лексики	Новые мастер-данные или атрибут МДЭС (см. 7.3.6)	Новый лексический элемент
Новая версия стандарта EPCIS	Рабочая группа GS1 по EPCIS	Да	Да	Да	Случайным образом	Редкий
Новая версия стандарта БДЛ	Рабочая группа GS1 по БДЛ (CBV)	Нет	Нет	Нет	Да	Да (шаблон типовой лексики, пользовательской лексики)
Прикладной стандарт GS1 для конкретной отрасли	Рабочая группа по прикладным стандартам GS1 для конкретной отрасли	Редкий	Редкое	Случайным образом	Да	Да (типовой лексикон)
Местный рекомендательный документ национальной организации GS1 для конкретной отрасли в конкретном географическом регионе	Национальная организация — член GS1	Редкий	Редкое	Случайным образом	Да	Да (типовой лексикон)
Спецификация по взаимодействию отдельной группы	Отраслевой консорциум или узкая группа конечных пользователей вне рамок GS1	Редкий	Редкое	Случайным образом	Да	Да (типовой лексикон)
Обновление мастер-данных с помощью запроса EPCIS или другого способа синхронизации данных	Индивидуальный конечный пользователь	Редкий	Редкое	Редкий	Редкие	Да (пользовательский лексикон)

6.4 Представление идентификаторов

Уровень абстрактной модели данных (Abstract Data Model Layer) представляет несколько видов идентификаторов, включая имена типов событий (Event Type), имена полей событий (Event Field), имена лексиконов (Vocabulary), лексических элементов (Vocabulary Elements), имена атрибутов мастер-данных (master data Attribute). Поскольку все эти пространства имен открыты для расширения, данная спецификация устанавливает некоторые правила, касающиеся конструкции этих имен, с тем чтобы независимые организации могли создавать расширения, не опасаясь коллизии имен.

Лексические элементы являются субъектом следующих правил. Во всех случаях лексический элемент представлен в виде унифицированного идентификатора ресурса (URI), общий синтаксис которого определен в *[11]*. Типами идентификаторов URI, допустимыми в качестве лексических элементов, являются следующие идентификаторы URI, у которых есть организация-владелец:

- представления идентификатора URI для номеров EPC согласно [*12*, раздел 6]. Организацией — владельцем конкретного идентификатора EPC URI является организация, которой был присвоен префикс предприятия GS1[1)] (или иное агентство выдачи в зависимости от конкретной схемы EPC);

[1)] *Единственной организацией, уполномоченной GS1 на присвоение префиксов предприятий GS1 на территории Российской Федерации, является Ассоциация автоматической идентификации «ЮНИСКАН/ГС1 РУС» (ГС1 РУС).*

- абсолютные унифицированные указатели ресурсов (Uniform Resource Locator, URL) согласно *[13]*. Организацией — владельцем конкретного указателя URL является организация, которая владеет доменным именем в сети Интернет в рамках принадлежащей ей части указателя URL;

- унифицированные имена ресурсов (Uniform Resource Names, URN) согласно *[14]* в пространстве имен `oid` (object identifier, идентификатор объекта), которые начинаются с номера частного предприятия (Private Enterprise Number, PEN). Организацией — владельцем для идентификаторов OID-URN является организация, которой был присвоен номер PEN;

- унифицированные имена ресурсов согласно *[14]* в пространстве имен `epc` или `epcglobal`, кроме тех идентификаторов URI, которые используются для представления номеров электронного кода продукции (EPC) согласно *[12]*. Организацией-владельцем этих имен URN является GS1.

Имена типов событий и имена полей событий представлены как составные имена из пространства имен (qname), состоящие из идентификатора URI и имени. Это имеет прямое представление в привязках XML, что удобно для расширения.

6.5 Иерархические лексиконы

Некоторые лексиконы имеют иерархическую или мульти-иерархическую структуру. Например, лексикон имен мест нахождения может иметь элемент, который означает «организация Acme Corp. розничный магазин № 3», а также другие элементы, которые означают «организация Acme Corp. розничный магазин № 3 подсобное помещение» и «организация Acme Corp. розничный магазин № 3 торговый этаж». В этом примере показаны естественные иерархические связи, в которых первый идентификатор является идентификатором-родителем, а последние два идентификатора являются идентификаторами-потомками.

Иерархические связи между лексическими элементами представляются через мастер-данные. В частности, идентификатор элемента-родителя, помимо его атрибутов мастер-данных, несет список его идентификаторов элементов-потомков. Каждый идентификатор элемента-потомка ДОЛЖЕН принадлежать тому же лексикону, что и идентификатор элемента-родителя. В приведенном выше примере элемент, называемый «организация Acme Corp. розничный магазин № 3», будет иметь список элементов-потомков, включающий в себя элемент «организация Acme Corp. розничный магазин № 3 дверь № 5».

В настоящей спецификации термин «прямой или косвенный потомок» используется для обозначения множества лексических элементов, включая элементы-потомки данного лексического элемента, элементы-потомки этих элементов-потомков и т. д. То есть «прямые или косвенные потомки» лексического элемента являются набором элементов, полученных путем транзитивного замыкания отношения «потомок», начиная с данного лексического элемента.

Данный элемент МОЖЕТ быть элементом-потомком более чем одного элемента-родителя. Это позволяет использовать несколько способов группировки лексических элементов. Например, места нахождения могут группироваться как по географическому признаку, так и по функциональному. Однако элемент НЕ ДОЛЖЕН приходиться элементом-потомком самому себе, прямо или косвенно.

Примечание — В данной версии спецификации предусмотрена только одна иерархическая связь, а именно отношение, закодированное в специальном списке «потомки». В будущих версиях настоящей спецификации это может быть обобщено для того, чтобы позволить использование более одной связи, возможно; кодируя каждую связь через различные атрибуты мастер-данных.

Иерархические отношения дают возможность специальной обработки в запросах (8.2), а также могут играть определенную роль в осуществлении политики авторизации (8.2.2), но не должны усложнять или добавлять какой-либо механизм к уровню абстрактной модели данных.

7 Уровень определения данных

Настоящий раздел содержит нормативные спецификации модулей на уровне определения данных (Data Definition Layer).

7.1 Общие правила формирования спецификации модулей уровня определения данных

В настоящем подразделе приводятся общие правила формирования спецификации на уровне определения данных. Эти правила в дальнейшем применяются в 7.2 для определения модуля типов основных событий (Core Event Types Module). Эти правила могут также применяться организациями, намеревающимися разместить свою спецификацию поверх указанной.

7.1.1 Содержимое

В общем случае спецификация модуля определения данных имеет следующие компоненты, которые наполняют структуру абстрактной модели данных, указанной в разделе 6:

- *типы значений (Value Types):* определения типов данных, которые используют для описания значений полей событий (Event Fields) и атрибутов мастер-данных (master data attributes). Модуль типов основных событий определяет простые типы значений, которые доступны для использования всеми модулями определения данных. Каждый определяемый лексикон также неявным образом является типом значений;

- *типы событий (Event Types):* определения типов событий, каждое из которых задает название типу события (которое должно быть уникальным для всех типов событий), а также список типовых полей событий для этого типа. Тип события может быть определен как подкласс существующего типа события, а это означает, что новый тип события включает все поля событий существующего типа событий плюс любые дополнительные поля событий, предусмотренных в рамках его спецификации;

- *поля событий (Event Fields):* определения полей событий внутри типов событий, каждое из которых указывает имя для поля (которое должно быть уникальным среди всех полей данного типа событий) и тип данных для значений этого поля. Определения полей событий внутри модуля определения данных могут быть частью новых типов событий, внесенных этим модулем, или могут расширить типы событий, определенные в других модулях;

- *виды лексики (Vocabulary Types):* определения видов лексики, каждое из которых задает имя лексикона (которое должно быть уникальным среди всех лексиконов), список типовых атрибутов мастер-данных для элементов этого лексикона и правила построения новых лексических элементов для этого лексикона. (Любые правила, установленные для построения лексических элементов в виде лексикона, должны соответствовать общим правилами, приведенным в 6.4.);

- *атрибуты мастер-данных (master data attributes):* определения атрибутов мастер-данных для видов лексики, каждое из которых указывает имя атрибута (которое должно быть уникальным для всех атрибутов покрывающего его вида лексики) и тип данных для значений этого атрибута. Определения мастер-данных внутри модуля определения данных могут принадлежать новым видам лексики, введенным данным модулем, или могут расширить виды лексики, определенные в других модулях;

- *лексические элементы (Vocabulary Elements):* Определения лексических элементов, каждое из которых указывает имя (которое должно быть уникальным среди всех элементов в рамках лексикона и соответствовать общим правилам для лексических элементов, указанным в 6.4, так же, как какие-либо конкретные правила, указанные в определении вида лексики), а также дополнительно определяют мастер-данные (конкретные значения атрибутов) для этого элемента.

Примечание — Как объяснено в 6.3, модули определения данных, указанные в настоящей и сопутствующих спецификациях, разработанных рабочей группой по EPCIS, будут стремиться включать определения типов значений, типов событий, полей событий и видов лексики, в то время как модули, определенные другими группами, будут стремиться включать определения полей событий, которые расширят существующие типы событий, атрибуты мастер-данных, которые расширят существующие виды лексики, а также лексические элементы, наполняющие существующие лексиконы. Другие группы могут также иногда определять виды лексики.

Слово «лексикон» используется неформально для ссылки на вид лексики и набор всех лексических элементов, которые его наполняют.

7.1.2 Описание

В нижеуказанных разделах типы событий и поля событий определены с использованием ограниченной формы схемы классов в нотации языка UML. Схемы классов языка UML, используемые для этой цели, могут содержать классы, которые имеют атрибуты (поля) и ассоциации (связи), но не операции. *На рисунке 5* приведен пример:

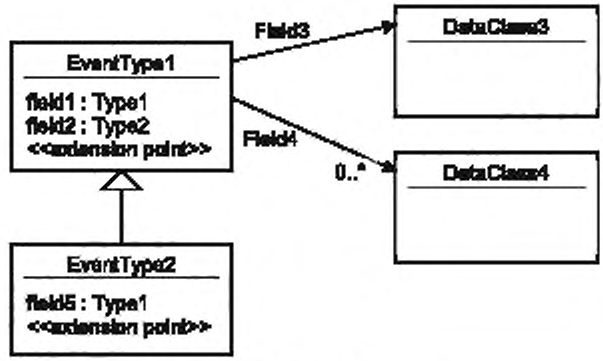


Рисунок 5

В схеме *на рисунке 5* приведено определение данных для двух типов событий, `EventType1` и `EventType2 (Тип_Событий_2)`. Эти типы событий используют четыре типа значений: `Type1 (Тип_1)`, `Type2 (Тип_2)`, `DataClass3 (Класс_Данных_3)` и `DataClass4 (Класс_Данных_4)`. Типы `Type1 (Тип_1)` и `Type2 (Тип_2)` являются простыми типами, в то время как типы `DataClass3 (Класс_Данных_3)` и `DataClass4 (Класс_Данных_4)` являются сложными типами, структура которых также установлена в UML.

Тип события `EventType1` в этом примере имеет четыре поля. Поля `Field1 (Поле_1)` и `Field2 (Поле_2)` имеют простые типы `Type1 (Тип_1)` и `Type2 (Тип_2)` соответственно. Тип `EventType1 (Тип_Событий_1)` имеет другое поле `Field3 (Поле_3)` с типом `DataClass3 (Класс_Данных_3)`. Наконец, тип `EventType1 (Тип_Событий_1)` имеет другое поле `Field4 (Поле_4)`, которое содержит список из 0 или более экземпляров типа `DataClass4 (Класс_Данных_4)` (нотация «0..*» означает «больше или равно нулю»).

Данная схема также показывает определение данных для `EventType2 (Тип_Событий_2)`. Стрелка с открытым треугольным наконечником показывает, что тип `EventType2 (Тип_Событий_2)` является подклассом типа `EventType1 (Тип_Событий_1)`. Это означает, что тип `EventType2 (Тип_Событий_2)` на самом деле имеет пять полей: четыре поля, наследуемые от типа `EventType1 (Тип_Событий_1)`, плюс пятое поле `field5 (Поле_5)` типа `Type1 (Тип_1)`.

В рамках описаний языка UML обозначение `<<extension point>>` (`<<точка расширения>>`) определяет место, где реализации ДОЛЖНЫ предусматривать расширяемость за счет добавления новых элементов данных. (Когда один тип имеет точку расширения, а другой тип определен как подкласс первого типа и также имеет точку расширения, то это не означает, что второй тип имеет две точки расширения, скорее это просто указывает на то, что второй тип также открыт для расширения.) Механизмы расширяемости ДОЛЖНЫ предусматриваться как для собственных расширений поставщиками продуктов, совместимых с EPCIS, так и для расширений, определенных стандартом GS1, посредством будущих версий настоящей спецификации или посредством новых спецификаций.

В случае стандартных привязок точки расширения XML реализованы в схеме XML в соответствии с методологией, описанной в 9.1.

Все определения типов событий ДОЛЖНЫ включать точку расширения, чтобы обеспечить расширяемость, описанную в 6.3 («новые поля событий»). Типы значений МОГУТ включать точку расширения.

7.1.3 Семантика

Каждое событие (экземпляр типа события) кодирует несколько утверждений, которые в совокупности определяют семантику события. Некоторые из этих утверждений говорят о том, что было верно в момент времени, когда было зафиксировано событие. Другие утверждения говорят о том, что, как предполагается, будет верным после наступления события до тех пор, пока оно не будет аннулировано последующим событием. Они называются соответственно *ретроспективная семантика (retrospective semantics)* события и *перспективная семантика (prospective semantics)* события. Например, если предмет (widget) № 23 перемещается в здание № 5 через дверь № 6 в 23:23, то ретроспективное утверждение звучит так: «предмет № 23 зафиксирован у двери № 6 в 23:23», тогда как перспективное утверждение звучит так: «предмет № 23 находится в здании № 5». Ключевым отличием является то, что ретроспективное утверждение ссылается на конкретный момент времени в прошлом («предмет № 23 зафиксирован…»),

в то время как перспективное утверждение является утверждением о текущем состоянии объекта («предмет № 23 находится в ...»). Перспективное утверждение предполагает, что если предмет № 23 когда-либо покинет здание № 5, то будет записано другое событие EPCIS, чтобы заменить предшествовавшее событие.

В общем случае ретроспективная семантика — это вещи, о которых неопровержимо известно, что они верны в момент сбора данных о событии и, как правило, могут полагаться на приложения EPCIS доступа как на точную констатацию исторического факта. Поскольку перспективная семантика пытается сказать, что есть истина, после того, как событие произошло, она должна рассматриваться в лучшем случае как утверждение того, «что должно быть» скорее, чем того, «что есть». Перспективное утверждение может оказаться неправдой в том случае, если оборудование сбора данных о событиях несовершенно, или если бизнес-процесс или архитектура системы для сбора данных о событиях EPCIS не были разработаны с возможностью работы в любых обстоятельствах. Кроме того, для того, чтобы сделать использование перспективного утверждения в событии неявным, приложение EPCIS доступа должно быть уверено, что будет иметь доступ к любому последующему событию, которое может заменить собой рассматриваемое событие.

Раздвоение на ретроспективное/перспективное играет важную роль в определении места нахождения по EPCIS в 7.3.4.

Возможны ситуации, когда по прошествии времени устанавливают ошибочность ранее произошедшего события (устанавливают некорректность утверждений его семантики), и эту ошибку невозможно исправить с помощью записи нового события, которое добавляет дополнительные утверждения посредством собственной семантики. Для таких случаев предусмотрен механизм для записи события, семантика которого утверждает, что утверждения, вытекающие из ошибочного события, являются некорректными (см. 7.4.1.2).

7.2 Модуль типов основных событий — обзор

Модуль определения данных типов основных событий (*Core Event Types*) определяет типы событий, которые представляют собой события сбора данных EPCIS (EPCIS data capture events). Эти события, как правило, генерируются приложением EPCIS сбора данных и предоставляются инфраструктуре EPCIS посредством операций, указанных в 8.1. Эти события возвращаются также в ответ на операции запроса, которые извлекают события в соответствии с критерием запроса.

Следуя основным положениям, приведенным в 7.1.1, компоненты этого модуля представляют собой следующее:

- *типы значений*: это простые типы, определенные в 7.3.1 и 7.3.2;
- *типы событий*: это типы событий, как показано ниже на схеме на языке UML и определено в 7.4.1—7.4.6;
- *поля событий*: включены как часть в определения типов событий,
- *типы лексики*: это типы, определенные в 7.3.3—7.3.5 и обобщенные в 7.2;
- *атрибуты мастер-данных:* включены как часть определений типов лексики. Предполагается, что отраслевые вертикальные рабочие группы будут определять дополнительные атрибуты мастер-данных для лексиконов, определенных в настоящем стандарте;
- *лексические элементы:* не предусмотрены в рамках данной спецификации. Предполагается, что отраслевые вертикальные рабочие группы будут определять лексические элементы для лексикона `BusinessStep (Бизнес-этап)` (см. 7.3.5), лексикона `Disposition (Состояние)` (см. 7.3.5.2), а также лексикона `BusinessTransactionType (Тип_Документа_Бизнес-Транзакции)` (см. 7.3.5.3.1).

Данный модуль определяет шесть типов событий: одно общее базовое событие и пять подклассов (один из которых является нежелательным в стандарте EPCIS 1.1 *[4]*, которые могут представлять собой события, возникающие в работе цепи поставок в широком спектре отраслей:

- `EPCISEvent (Событие_EPCIS)` (см. 7.4.1) является общим базовым классом для всех типов событий как в этом модуле, так и в других;
- `ObjectEvent (Событие_с_Объектом)` (см. 7.4.1.2) представляет собой событие, которое произошло для одного или нескольких физических или цифровых объектов;
- `AggregationEvent (Событие_с_Агрегацией_Объектов)` (см. 7.4.3) представляет собой событие, произошедшее для одного или нескольких объектов, которые физически собраны вместе (физически соединены, чтобы быть в одном месте и в одно время, как, например, коробки, собранные на одном поддоне);

- QuantityEvent (Событие_с_Количеством_Объектов) (см. 7.4.4) представляет собой событие, связанное с определенным количеством объектов, разделяющих общий класс EPC, но в котором не указаны ключевые идентификаторы отдельных объектов. Согласно *[4]* (версии 1.1 стандарта EPCIS) это событие является устаревшим. Вместо него следует использовать событие типа ObjectEvent (Событие_с_Объектом) (см. 7.4.1.2) с одним или несколькими элементами QuantityElements (Элементы_Количества) (см. 7.3.3.3);

- TransactionEvent (Событие_с_Документом_Транзакции) (см. 7.4.5) представляет собой событие, в котором один или несколько объектов становятся связанными или разъединенными с одним или несколькими указанными документами бизнес-транзакций;

- TransformationEvent (Событие_с_Преобразованием) (см. 7.4.6) представляет собой событие, в котором входные объекты потребляются полностью или частично, а выходные объекты производятся таким образом, чтобы любой из входных объектов мог способствовать созданию всех выходных объектов.

Схема на языке UML, представляющая указанные типы событий следующим образом, приведена *на рисунке 6*.

Примечание — На схеме *на рисунке 6* некоторые имена сокращены из-за пространственных ограничений; например, на схеме используется имя типа BizLocationID, тогда как фактически данный тип называется BusinessLocationID (Идентификатор_Производственного_Места_Назначения). Нормативные названия полей и их типов см. в тексте настоящего стандарта.

Каждый из типов основных событий (не считая общий тип EPCISEvent (Событие_EPCIS)) имеет поля, которые представляют собой четыре ключевых параметра любого события EPCIS: (1) объект(ы) или другие сущности, которые являются субъектом события; (2) дата и время; (3) место нахождения, в котором произошло событие; (4) контекст хозяйственной деятельности. Эти четыре параметра можно легко запомнить, как «что», «когда», «где/куда» и «на каком основании» (соответственно). Параметр «что» меняется в зависимости от типа событий (например, для типа ObjectEvent (Событие_с_Объектом) параметр «что» представляет собой один или несколько номеров EPC; для типа AggregationEvent (Событие_с_Агрегацией) параметр «что» представляет собой идентификатор-родитель (ID) или список номеров-потомков EPC). Параметры «где/куда» и «на каком основании» имеют как ретроспективный, так и перспективный аспекты (см. 7.1.3), представленные различными полями.

В таблице *4* приведены поля типов событий, которые относятся к четырем основным параметрам:

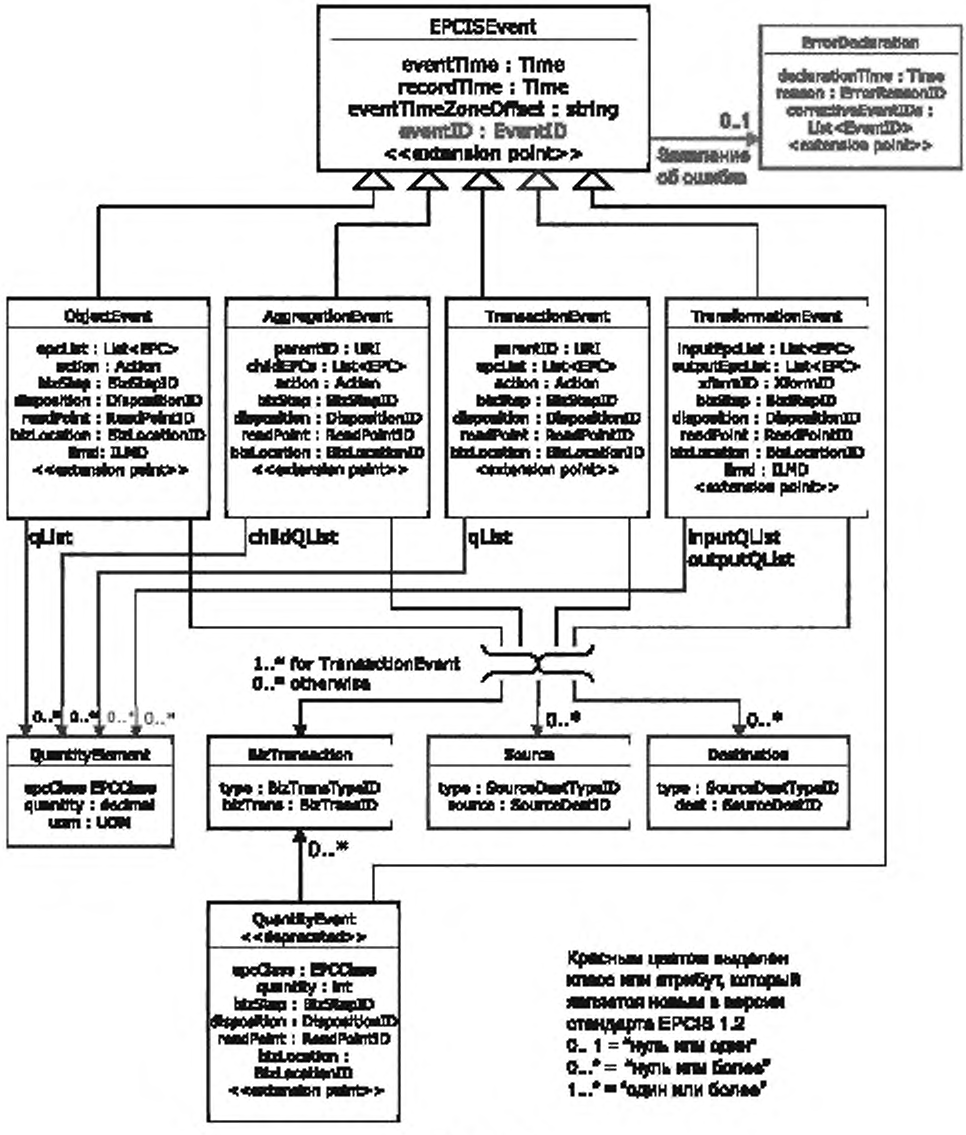


Рисунок 6

Таблица 4

	Ретроспективный аспект (верно в момент наступления события)	Перспективный аспект (верно, пока не опровергнуто последующим событием)
«Что»	EPC EPCClass (EPC_на_уровне_Класса) + количество	
«Когда»	Time (Время)	
«Где/куда»	ReadPointID (Идентификатор_Места_Считывания)	BusinessLocationID (Идентификатор_Производственного_Места Назначения)

Окончание таблицы 4

	Ретроспективный аспект (верно в момент наступления события)	Перспективный аспект (верно, пока не опровергнуто последующим событием)
«На каком основании» (контекст хозяйственной деятельности)	BusinessStepID (Идентификатор_Бизнес-этапа)	DispositionID (Идентификатор_Состояния)
	BusinessTransactionList (Список_Документов_Бизнес-транзакций) Source/Destination (Начальный_Пункт/Конечный_Пункт) ILMD (МДЭС)	

В дополнение к полям, принадлежащим четырем ключевым параметрам, события могут нести дополнительную описательную информацию в других полях. Предполагается, что большинство дополнительных описательных информационных полей будет определяться конкретными отраслевыми спецификациями, базирующимися на данной спецификации.

В таблице 5 приведены виды лексики, определенные в этом модуле. Столбец «Идентификатор URI» предоставляет формальное имя для лексикона, которое используется тогда, когда необходимо ссылаться на имя лексикона через интерфейс EPCIS.

Таблица 5

Вид лексики (*лексикон*)	Пункт стандарта	Пользовательская/Типовая лексика	Идентификатор URI
ReadPointID (Идентификатор_Места_Считывания)	7.3.4	Пользовательская	urn:epcglobal:epcis:vtype:ReadPoint
BusinessLocationID (Идентификатор_Производственного_Места_Назначения)	7.3.4	Пользовательская	urn:epcglobal:epcis:vtype:BusinessLocation
BusinessStepID (Идентификатор_Бизнес-этапа)	7.3.5	Типовая	urn:epcglobal:epcis:vtype:BusinessStep
DispositionID (Идентификатор_Состояния)	7.3.5.2	Типовая	urn:epcglobal:epcis:vtype:Disposition
BusinessTransaction (Документ_Бизнес_Транзакции)	7.3.5.3.2	Пользовательская	urn:epcglobal:epcis:vtype:BusinessTransaction
BusinessTrasactionTypeID (Идентификатор_Типа_Документа_Бизнес-транзакции)	7.3.5.3.1	Типовая	urn:epcglobal:epcis:vtype:BusinessTransactionType
EPCClass (EPC_на_уровне_Класса)	7.3.5.4	Пользовательская	urn:epcglobal:epcis:vtype:EPCClass
SourceDestTypeID (Идентификатор_Типа_Начального_Пункта_Конечного_Пункта)	7.3.5.4.1	Типовая	urn:epcglobal:epcis:vtype:SourceDestType
SourceDestID (Идентификатор_Начального_Пункта_Конечного_Пункта)	7.3.5.4.2	Пользовательская	urn:epcglobal:epcis:vtype:SourceDest
LocationID (Идентификатор_Места_Назначения)	*	Пользовательская	urn:epcglobal:epcis:vtype:Location
ErrorReasonID (Идентификатор_Причины_Ошибки)	7.4.1.2	Типовая	urn:epcglobal:epcis:vtype:ErrorReason

* Вид лексики LocationID (Идентификатор_Места_Назначения) является обобщающим видом для ReadPointID (Идентификатор_Места_Считывания), BusinessStepID (Идентификатор_Бизнес-этапа) и SourceDestID (Идентификатор_ Начального_Пункта_Конечного_Пункта). В документе мастер-данных EPCIS (либо в заголовке мастер-данных в документе EPCIS или в документе запроса EPCIS) идентификатор urn:epcglobal:epcis:vtype:Location URI может быть использован для обозначения единой лексики, содержащей идентификаторы, которые могут появляться в полях read point (место считывания), business step (бизнес-этап), source (начальный пункт) или destination (конечный пункт) соответствующих событий EPCIS.

7.3 Модуль типов основных событий — составные части

Данный подраздел определяет составные части для типов событий, указанных в 7.3.5.4.

7.3.1 Простые типы

В рамках модуля типов основных событий используются следующие простые типы значений, *приведенные в таблице 6*.

Таблица 6

Тип значений	Описание
int (целочисленное значение)	Целочисленное значение (integer). Ограничения диапазона упоминаются, когда это применимо
Time (Время)	Временная метка, дает дату и время в независимой от часового пояса форме. Для привязок, в которых поля этого типа представлены текстуально, СЛЕДУЕТ применять совместимые представления по [15]
EPC	Номера электронного кода продукции, как это определено в [12]. Если не указано иное, номера EPC представлены в форме «чистого ключевого идентификатора» URI, как определено в [12, раздел 7]

Тип значений EPC определяется как простой тип значений для использования в событиях при обращении к номерам EPC, которые не являются частью вида лексики. Например, номер EPC в формате номера SGTIN, используемый для обозначения экземпляра предмета торговли в поле epcList (список_EPC) события типа ObjectEvent (Событие_с_Объектом), является экземпляром простого типа значений EPC. Но номер EPC в формате номера SGLN[1)], используемый как идентификатор места считывания (см. 7.3.4) в поле ReadPoint (Место_Считывания) события типа ObjectEvent (Событие_с_Объектом), является лексическим элементом, а не экземпляром простого типа значений EPC.

Примечание — Это отражает проектное решение не рассматривать экземпляры отдельного предмета торговли как лексические элементы, имеющие мастер-данные, в связи с тем, что экземпляры предмета торговли создаются постоянно и, следовательно, постоянно задействуются новые номера EPC, представляющие предметы торговли. Отчасти данное проектное решение отражает последовательную обработку мастер-данных как исключительных данных, количество которых растет по мере ведения хозяйственной деятельности (см. примечание в 6.1), а отчасти отражает прагматическую реальность, заключающуюся в том, что данные об экземплярах предметов торговли скорее всего будут обрабатываться в большей степени как данные о событии, чем как мастер-данные, когда речь идет об устаревании, структуре базы данных и т. д.

7.3.2 Тип Action (действие)

Тип Action (Действие) свидетельствует о том, как событие относится к жизненному циклу описываемого объекта. Например, тип событий AggregationEvent (Событие_с_Агрегацией) (см. 7.4.3) используется для событий сбора данных, относящихся к группам объектов, таких как коробки, собранные на поддоне. На протяжении своей жизни поддон с грузом участвует во многих этапах бизнес-процессов, каждый из которых может генерировать событие EPCIS. Поле action (действие) каждого события свидетельствует о том, как сама группа объектов изменилась во время события: были ли к группе добавлены объекты, были ли объекты удалены из группы, или группа была зафиксирована без изменений в ее составе. Поле action (действие) не зависит от значения поля bizStep (бизнес-этап) (типа BusinessStepID (Идентификатор_Бизнес-Этапа)), которое идентифицирует конкретный этап бизнес-процесса, в котором произошло данное действие.

Тип Action (Действие) является перечисляемым типом, имеющим три возможных значения (*см. таблицу 7*).

Таблица 7

Значение Action (Действие)	Смысловое содержание
ADD (ДОБАВЛЕНИЕ)	Рассматриваемый объект был создан или дополнен
OBSERVE (ОТСЛЕЖИВАНИЕ)	Рассматриваемый объект не был изменен: он не был создан, добавлен, уничтожен или удален
DELETE (УДАЛЕНИЕ)	Рассматриваемый объект был удален или уничтожен полностью

[1)] *SGLN — структура данных GS1, соответствующая глобальному номеру места нахождения GLN, с включением или без опционального расширения, используемая для идентификации физического места нахождения.*

Приведенное ниже описание для каждого типа события, которое включает значение Action (Действие), говорит более точно, как интерпретируется значение Action (Действие) в контексте этого события.

Следует отметить, что три приведенных выше значения являются единственно возможными значениями для типа Action (Действие). В отличие от других типов, описанных ниже, тип Action (Действие) *не* является видом лексики, а также НЕ ДОЛЖЕН расширяться отраслевыми группами.

7.3.3 Параметр «что»

Данный подраздел описывает типы данных, используемые в параметре «что» («What») типов событий, указанных в 7.3.5.4.

7.3.3.1 Идентификация на уровне экземпляра в сравнении с идентификацией на уровне класса

Параметр «что» события EPCIS определяет, какие физические или цифровые объекты участвовали в событии. EPCIS предусматривает идентификацию объектов двумя способами:

- *уровень экземпляра*: идентификатор называют идентификатором на уровне экземпляра, если такие идентификаторы присвоены таким образом, что каждый является уникальным для одного-единственного объекта. То есть не существует двух объектов, имеющих один и тот же идентификатор на уровне экземпляра;
- *уровень класса*. Идентификатор называют идентификатором на уровне класса, если несколько объектов могут иметь один и тот же идентификатор.

В общем случае идентификаторы на уровне экземпляра позволяют событиям EPCIS передавать больше информации, поскольку существует возможность соотнести несколько событий EPCIS, параметр «что» которых содержит одни и те же идентификаторы на уровне экземпляра. Например, если событие EPCIS содержит указанный идентификатор на уровне экземпляра, а последующее событие EPCIS содержит такой же идентификатор, то в обоих событиях определенно участвовал один и тот же объект. Напротив, если оба события содержат идентификаторы на уровне класса, то не ясно, участвовал ли в обоих событиях один и тот же объект, поскольку второе событие могло быть другим экземпляром того же класса (т. е. другой объект, имеющий тот же идентификатор на уровне класса, что и первый объект). Идентификаторы на уровне класса, как правило, используются только тогда, когда непрактично присваивать уникальные идентификаторы на уровне экземпляра каждому объекту.

Пример — В системе GS1 примерами идентификаторов на уровне экземпляра являются номер GTIN+серийный номер, номер SSCC, идентификатор GRAI с серийным номером, идентификатор GIAI, номер GSRN, и идентификатор GDTI с серийным номером. Примерами идентификаторов на уровне класса являются номер GTIN, номер GTIN+номер партии, идентификатор GRAI без серийного номера и идентификатор GDTI без серийного номера.

7.3.3.2 Номер EPC

Номер электронного кода продукции (Electronic Product Code - EPC) является структурой идентификатора на уровне экземпляра, определенной в стандарте данных радиочастотных меток EPC *[12]*. В параметре «что» события EPCIS значение элемента epc ДОЛЖНО быть в формате идентификатора URI согласно *[11]*, обозначающем уникальный идентификатор на уровне экземпляра для объекта. Если уникальным идентификатором является номер электронного кода продукции, то элементом списка ДОЛЖЕН быть универсальный идентификатор ресурса (URI) с «чистым ключевым идентификатором» для EPC, как указано в [*12*, раздел 6]. Реализации МОГУТ принимать идентификаторы в формате URI, кроме номеров EPC, в качестве значения элемента epc.

7.3.3.3 Структура QuantityElement (Элемент_Количество)

Структура QuantityElement (Элемент_Количество) определяет объекты, идентифицируемые специальным идентификатором на уровне класса, либо с заданным полем quantity, либо с незаданным полем quantity (количество). Структура QuantityElement (Элемент_Количество) приведена *в таблице 8*.

Таблица 8

Поле	Тип	Описание
epcClass (epc_на_уровне_Класса)	EPCClass (EPC_на_уровне_Класса)	Идентификатор на уровне класса, которому принадлежит определенное количество объектов
quantity (количество)	Decimal (Десятичное_Значение)	(Необязательное) Число, которое определяет, какое число или какое количество указанного EPCClass (EPC_на_уровне_Класса) обозначается QuantityElement (Элемент_Количество).

Окончание таблицы 8

Поле	Тип	Описание
		Поле quantity (количество) может быть опущено, чтобы показать, что количество неизвестно или не указано. Если поле quantity (количество) не задано, то поле uom (единица_измерения) также ДОЛЖНО быть опущено. В противном случае, если поле quantity (количество) задано: Если в структуре QuantityElement (Элемент_Количество) отсутствует поле uom (единица_измерения) (см. ниже), то поле quantity (количество) ДОЛЖНО иметь положительное целое значение и означает подсчет числа экземпляров указанного EPCClass (EPC_на_уровне_Класса), которые обозначаются QuantityElement (Элемент_Количество). Если в структуре QuantityElement (Элемент_Количество) присутствует поле uom (единица_измерения), то количество (quantity) ДОЛЖНО иметь положительное значение (но не обязательно целое значение) и обозначать величину физической меры, которая определяет, какое количество указанного EPCClass (EPC_на_уровне_Класса) обозначается QuantityElement (Элемент_Количество)
uom (единица_измерения)	UOM (Единица_Измерения)	(Необязательное) Если поле присутствует, то оно устанавливает единицу измерения, с помощью которой указанное количество будет интерпретировано как физическая мера, которая определяет, какое количество указанного EPCClass (EPC_на_уровне_Класса) обозначается структурой QuantityElement (Элемент_Количество). Поле uom (единица_измерения) ДОЛЖНО быть опущено, если опущено поле quantity (количество)

EPCClass (EPC_на_уровне_Класса) является лексиконом, элементы которого обозначают классы объектов. EPCClass (EPC_на_уровне_Класса) является пользовательской лексикой, как это указано в 6.2. На любой номер EPC, структура которого включает в себя понятие класса объектов, можно ссылаться как на EPCClass (номер EPC на уровне класса объектов). Требования для номеров EPC в формате номера SGTIN подробно изложены ниже.

EPCClass (EPC_на_уровне_Класса) может относиться к классу для фиксированной или переменной величины. Класс для фиксированной величины имеет дело с экземплярами, которые могут быть подсчитаны. Например, номер GTIN, который относится к коробкам с продукцией фиксированного размера. Класс для переменной величины имеет дело с экземплярами, которые не могут быть подсчитаны, и потому их количество определяется как физическая мера. Например, номер GTIN, относящийся к медной проволоке, которая продается по длине, ковровому покрытию, которое продается по площади, маслу, которое продается по объему, или свежей плодоовощной продукции, которая продается на вес. В таблице 9 указано, как поля quantity (количество) и uom (единица_измерения) используются в каждом случае:

Таблица 9

EPCClass (EPC_на_уровне_Класса)	поле quantity (количество)	поле uom (единица_измерения)	Значение
Фиксированная величина	Целое положительное	Не задано	Поле quantity (количество) определяет количество экземпляров указанного класса
Фиксированная величина	Целое положительное	Не задано	Поле quantity (количество) определяет количество экземпляров указанного класса
Переменная величина	Положительное число, не обязательно целое	Присутствует	Поле quantity (количество) определяет величину, а поле uom (единица_измерения) — физическую единицу измерения, описывающую количество указанного класса
Фиксированная или переменная величина	Не задано	Не задано	Количество неизвестно или не определено

Атрибуты мастер-данных для лексикона EPCClass (EPC_на_уровне_Класса) содержат все мастер-данные для объектов, на которые дается ссылка, независимо от EPCIS (например, данные каталога продукции). Их определения выходят за рамки настоящей спецификации.

7.3.3.3.1 UOM (единица измерения)

Как указано выше, поле uom (единица_измерения) структуры QuantityElement (Элемент_Количество) присутствует тогда, когда QuantityElement (Элемент_Количество) использует физическую величину для определения количества указанного EPCClass (EPC_на_уровне_Класса). Если поле uom (единица_измерения) присутствует, его значение ДОЛЖНО быть двух- или трехзначным кодовым значением для физической единицы, указанной в столбце «Общее кодовое значение» («Common Code») Рекомендации 20 СЕФАКТ ООН (UN/CEFACT), см. [16]. Кроме того, это кодовое значение ДОЛЖНО соответствовать кодовому значению, содержащемуся в строке в [3], удовлетворяющей следующим критериям:

- столбец «Количество» («Quantity») содержит одно из следующих значений: *длина (length)*, *площадь (area)*, *объем (volume)* или *масса (mass)*;
- столбец «Статус» («Status») *не содержит* знак "X" (значение удалено) или знак "D" (значение устарело).

Для оценки по первому критерию значение количества должно появиться как законченная фраза. Например: значение «метр» (MTR, metre) допустимо, поскольку диапазон значений количества включает *длину (length)* (среди других значений, таких как *ширина (breadth)*, *высота (height)* и т. д.). *Однако значение «килограмм на метр (kilogram per metre)» (KL) недопустимо, поскольку здесь приведено значение в килограммах, поделенное на длину в метрах, а не просто длина (length).*

7.3.3.3.2 Значения EPCClass (номера EPC на уровне классов объектов) для номеров GTIN

Если лексический элемент в лексиконе EPCClass (EPC_на_уровне_Класса) представляет класс номеров SGTIN EPC, обозначаемый конкретным номером GTIN, то он ДОЛЖЕН представлять собой идентификатор URI в форме, указанной в [12], а также в более ранних версиях этого документа, начиная с версии 1.3:

urn:epc:idpat:sgtin:CompanyPrefix.ItemRefAndIndicator.*,

где составляющая CompanyPrefix (Префикс_Предприятия) является префиксом предприятия GS1 в номере GTIN (включая ведущие нули), а составляющая ItemRefAndIndicator (Ссылочный_Номер_Предмета_и_Индикатор) состоит из цифры индикатора, за которой следуют цифры ссылочного номера предмета номера GTIN.

Лексический элемент лексикона EPCClass (EPC_на_уровне_Класса) в этой форме обозначает класс объектов, номера EPC которых являются номерами SGTIN (urn:epc:id:sgtin:…), имеющими те же поля — CompanyPrefix (Префикс_Предприятия) и ItemRefAndIndicator (Ссылочный_Номер_Предмета _и_Индикатор), а также имеющими какой-либо серийный номер (или вообще не имеющими серийного номера).

7.3.3.3.3 Значения EPCClass (номера EPC на уровне классов объектов) для номера GTIN + номера партии/серии

Если лексический элемент в лексиконе EPCClass (EPC_на_уровне_Класса) представляет класс номеров SGTIN EPC, обозначаемый конкретным номером GTIN вместе с номером партии/серии, он ДОЛЖЕН представлять собой идентификатор URI в форме, указанной в [12, раздел 6]:

urn:epc:class:lgtin:CompanyPrefix.ItemRefAndIndicator.Lot,

где составляющая CompanyPrefix (Префикс_Предприятия) является префиксом предприятия GS1 в номере GTIN (включая ведущие нули), а составляющая ItemRefAndIndicator (Ссылочный_Номер_Предмета_и_Индикатор) состоит из цифры индикатора номера GTIN, за которым следуют цифры ссылочного номера предмета номера GTIN, и Lot (партия/серия) является номером конкретной партии/серии.

Лексический элемент EPCClass (EPC_на_уровне_Класса) в этой форме обозначает класс объектов, номера EPC которых являются номерами SGTIN (urn:epc:id:sgtin:…), имеющими те же поля CompanyPrefix (Префикс_Предприятия) и ItemRefAndIndicator (Ссылочный_Номер_Предмета _и_Индикатор) и принадлежащими к конкретной партии/серии, независимо от серийного номера (если таковой имеется).

7.3.3.4 Краткий перечень типов идентификаторов (справочная информация)

В данном подразделе приводится краткий перечень идентификаторов, которые могут быть использованы в параметре «что» событий EPCIS. Требования к идентификаторам приведены в стандарте данных радиочастотных меток EPC [12] и базовой деловой лексики по ГОСТ Р 59168 (см. таблицу 10).

Таблица 10

Тип идентификатора	Уровень экземпляра (EPC)	Уровень класса (EPCClass)	Префикс URI	Ссылка на нормативные документы
GTIN		+	urn:epc:idpat:sgtin:	[12, раздел 8]
GTIN +партия/серия		+	urn:epc:class:lgtin:	[12, раздел 6]
GTIN + серийный номер	+		urn:epc:id:sgtin:	[12, пункт 6.3.1]
SSCC	+		urn:epc:id:sscc:	[12, пункт 6.3.2]
GRAI (без серийного номера)		+	urn:epc:idpat:grai:	[12, раздел 8]
GRAI (с серийным номером)	+		urn:epc:id:grai:	[12, пункт 6.3.4]
GIAI	+		urn:epc:id:giai:	[12, пункт 6.3.5]
GDTI (без серийного номера)		+	urn:epc:idpat:gdti:	[12, раздел 8]
GDTI (с серийным номером)	+		urn:epc:id:gdti:	[12, пункт 6.3.7]
GSRN (Поставщик)	+		urn:epc:id:gsrn:	[12, пункт 6.3.6]
GSRN (Провайдер)	+		urn:epc:id:gsrnp:	[12, пункт 6.3.6]
GCN (без серийного номера)		+	urn:epc:idpat:sgcn:	[12, раздел 8]
GCN (с серийным номером)	+		urn:epc:id:sgcn:	[12, раздел 6]
CPI		+	urn:epc:idpat:cpi:	[12, раздел 8]
CPI + серийный номер	+		urn:epc:id:cpi:	[12, пункт 6.3.11]
GID	+		urn:epc:id:gid:	[12, пункт 6.3.8]
USDoD	+		urn:epc:id:usdod:	[12, пункт 6.3.9]
ADI	+		urn:epc:id:adi:	[12, пункт 6.3.10]
Идентификатор, не являющийся идентификатором GS1	+	+	Любой идентификатор URI — для рекомендаций	по *ГОСТ Р 59168*, подраздел 8.2

7.3.4 Параметр «где/куда» — место считывания и производственное место назначения

Настоящий подраздел определяет четыре типа, имеющих отношение к понятию информации о *месте нахождения* в том виде, как это понятие используется в стандарте EPCIS. Два типа из этих четырех являются способами обращения к «устройствам считывания», или устройствам, которые воспринимают наличие оснащенных радиочастотными метками EPC объектов, с использованием технологии радиочастотной идентификации (RFID) или других средств. Эти типы в действительности не считаются типами «мест нахождения» для целей EPCIS. Они включены в настоящую спецификацию в основном для их противопоставления истинным типам места нахождения (хотя некоторые приложения могут ориентироваться на их использование как полей расширения в процессе наблюдения с целью аудита). Оставшиеся два типа являются действительными типами места нахождения и определены как виды лексики (лексиконы) EPCIS.

В *таблице 11* приведены следующие типы устройств считывания/мест нахождения:

Таблица 11

Тип	Описание
Простые типы устройства считывания — отсутствуют типы мест нахождения для EPCIS	
PhysicalReaderID (Идентификатор_Физического_Устройства_Считывания)	Это уникальный ключевой идентификатор или имя конкретного источника информации (например, физического устройства считывания радиочастотных меток), который передает результаты события считывания номера EPC. Идентификатор физического устройства считывания определен в *[17]*

Окончание таблицы 11

Тип	Описание
LogicalReaderID (Идентификатор_Логического_Устройства_Считывания)	Это ключевой идентификатор или имя, данное источнику информации о событии считывания номера EPC, которое не зависит от физического устройства или устройств, которые используются для выполнения считывания. Идентификатор логического устройства считывания определен в [17]. Существует несколько причин для введения понятия логического устройства считывания, как это описано в [17], включая разрешение заменять физические устройства считывания без запроса изменений у приложения сбора данных EPCIS, разрешение давать одно имя нескольким физическим устройствам считывания, когда они уже используются одновременно для покрытия одного места нахождения, а также (напротив) разрешение сопоставить единственному физическому устройству считывания несколько логических устройств, когда физическое устройство считывания имеет несколько антенн, используемых независимо для покрытия различных мест нахождения
Действительные типы мест нахождения	
ReadPointID (Идентификатор_Места_Считывания)	Место считывания является отдельно зафиксированным местом нахождения, которое предназначено для выявления наиболее точного места, в котором имело место событие EPCIS. В случае использования стационарных устройств считывания места считывания определяются приложением EPCIS сбора данных, возможно, введением функции логического устройства считывания, в случае использования мобильного устройства считывания места считывания, возможно, определяются непосредственно считыванием радиочастотной метки места нахождения или в общем случае определяются любыми другими средствами, которые выбирает для использования приложение EPCIS сбора данных. Теоретически место считывания предназначено для идентификации того, «где находились объекты в момент совершения события EPCIS»
BusinessLocationID (Идентификатор _Производственного_Места_Назначения)	Производственное место назначения является уникальным образом идентифицированным и отдельно зафиксированным местом нахождения, которое предназначено для обозначения конкретного места, откуда, как предполагается, объект будет следовать вслед за событием EPCIS до тех пор, пока посредством последующего события не поступит сообщение о том, что он находится в другом производственном месте назначения. Как и в случае с местом считывания, приложение EPCIS сбора данных определяет производственное место назначения любыми выбранными им средствами. Теоретически производственное месте назначения предназначено для идентификации того, «куда объекты следуют после события EPCIS»

Как указано в 6.2, ReadPointID (Идентификатор_Места_Считывания) и BusinessLocationID (Идентификатор_Производственного_Места _Назначения) являются лексиконами пользовательской лексики. Некоторые отрасли, возможно, отдадут предпочтение использованию номеров EPC в качестве лексических элементов. В этом случае ДОЛЖНЫ использоваться чистые ключевые идентификаторы в формате идентификатора URI, как указано в [12].

Пример — В отраслях, которые руководствуются Общими спецификациями GS1, типы readPointID (Идентификатор_Места_Считывания) и businessLocationID (Идентификатор_ Производственного_Места_Назначения) могут иметь вид идентификаторов SGLN-URI по [12, пункт 6.3.3], а тип physicalReaderID (Идентификатор_Физического_ Устройства_Считывания) может иметь вид идентификатора SGTIN-URI по [12, пункт 6.3.1].

Однако во всех этих случаях не требуется, чтобы лексические элементы мест нахождения являлись номерами EPC.

Примечание — Разрешение использовать идентификаторы URI не в формате EPC для мест нахождения дает организациям большую свободу для повторного использования существующих способов именования мест нахождения.

Для всех типов событий EPCIS, указанных в подразделе 7.2, события сбора данных включают отдельные поля для места считывания и производственного места назначения. В большинстве случаев оба поля не обязательны, так что для приложения EPCIS сбора данных событие может включать неполную информацию, если информация для обоих этих полей не известна.

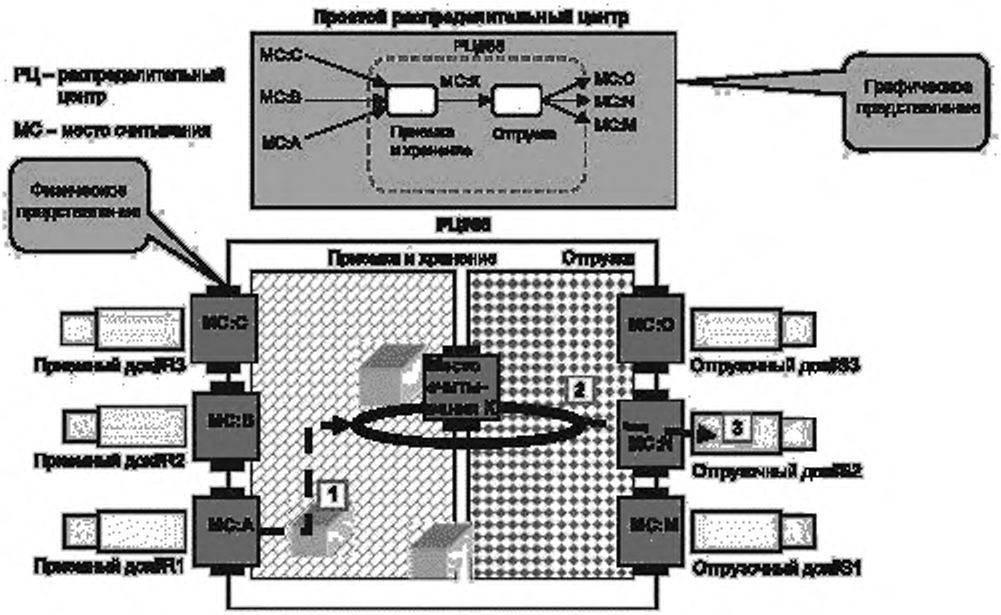
Примечание — В настоящей спецификации логические и физические устройства считывания исключены из описаний событий EPCIS. Информация о физическом устройстве считывания в общем случае не является полезной информацией для обмена между партнерами. Например, если устройство считывания неисправно и

меняется на другое устройство считывания идентичной марки и модели, идентификатор физического устройства считывания изменяется. Эта информация не представляет большого интереса для торговых партнеров. Аналогично идентификатор логического устройства считывания может измениться, если организация, ответственная за сбор данных, вносит изменения в порядок выполнения конкретного бизнес-процесса, что, опять-таки, зачастую не представляет интерес для партнера.

Различие между местом считывания и производственным местом назначения в большей степени связано с раздвоением на ретроспективную и перспективную семантику, описанную выше. В общем случае, пункты считывания играют роль в ретроспективной семантике, тогда как производственные места назначения участвуют в перспективных утверждениях. Это делает понятным то, каким образом каждый тип места нахождения входит в семантические описания, приведенные ниже, в конце каждого раздела, и определяющие событие EPCIS сбора данных.

7.3.4.1 Пример различия между местом считывания и производственным местом назначения (для справки)

На рисунке 7 приведен пример, представляющий различия между местом считывания и производственным местом назначения.



Радиочастотная метка	Время	Место считывания	Производственное место назначения	Комментарий
№ 123	7:00	"МС-ЦД № 88-А"	ЦД № 88. Приемка и хранение	Продукция поступила в распределительный центр через дверной проем дока № R1
№ 123	9:00	"МС-ЦД № 88-К"	ЦД № 88. Отгрузка	Продукция размещена на конвейере для отгрузки
№ 123	9:30	"МС-ЦД № 88-N"	[не задано]	Продукция отгружена через дверной проем дока № S2

Рисунок 7

На рисунке 7 показан типичный пример использования, в рамках которого имеются помещения с фиксированными дверными проемами на границах помещений. В данном случае места считывания соответствуют дверным проемам (с устройствами радиочастотной идентификации), а производственные места назначения соответствуют помещениям. Следует обратить внимание, что места считывания и производственные места назначения не находятся во взаимно-однозначном соответствии. Единственной ситуацией, когда места считывания и производственные места назначения могут иметь соответствие один к одному, является нетипичный случай комнаты с единственной дверью, например маленькая кладовая.

Снова обращаясь к примеру с помещениями и дверями, следует отметить, что производственное место назначения обычно является типом места нахождения, представляющим наибольший интерес для приложений для ведения хозяйственной деятельности, поскольку он сообщает, в каком помещении находится объект. Таким образом, имеет смысл сделать запрос на инвентаризацию производственного места назначения, такого как подсобное помещение. В отличие от этого место считывания указывает на дверной проем, через который объект попал в помещение. Делать запрос на инвентаризацию дверных проемов не имеет смысла. Иногда не имеющее отношения к приложениям для ведения хозяйственной деятельности место считывания тем не менее представляет значительный интерес для программного обеспечения высокого уровня для понимания бизнес-процесса и конечного статуса объекта, в особенности при наличии менее чем 100%-ного коэффициента считывания. Следует обратить внимание, что корректное обозначение производственного места назначения требует, чтобы объект, снабженный радиочастотной меткой, был зафиксирован в месте считывания, а также чтобы было корректно определено направление перемещения. К тому же сообщение о событии в месте считывания будет очень важным для программного обеспечения высокого уровня.

Цепь поставок, такая как в примере с помещениями и дверями, может быть представлена в виде графа, где каждый узел графа представляет собой помещение, в котором могут быть найдены объекты, а каждая дуга представляет собой дверной проем, который соединяет два помещения. Производственные места назначения, следовательно, соответствуют узлам этого графа, места считывания соответствуют дугам. Если граф являлся прямой однонаправленной цепью, то дуги, пройденные данным объектом, могут быть перестроены по информации об узлах, то есть информация из мест считывания была бы избыточной по отношению к информации, полученной из производственных мест назначения. Однако в более реальных ситуациях объекты могут проходить несколькими путями и перемещаться «в обратном направлении» по цепи поставок. В этих реальных ситуациях информация, предоставляемая местом считывания, в дополнение к информации о производственном месте назначения, является важной для высокоуровневого программного обеспечения.

7.3.4.2 Различие между типами устройств считывания и типами мест нахождения (для справки)

В контексте EPC термин «место нахождения» используется для обозначения множества различных понятий, что приводит к неоднозначности понимания значения термина и его применения, особенно исходя из перспектив ведения хозяйственной деятельности. Эта неоднозначность имеет ряд причин:

1) в ситуациях, когда устройства считывания EPC являются стационарными, возникает естественная тенденция отождествлять устройство считывания с местом нахождения, хотя такой подход может не всегда быть правильным в случаях, когда в одном месте нахождения находится несколько устройств считывания;

2) возможны ситуации, при которых стационарные устройства считывания размещают между различными, с точки зрения персонала, ответственного за ведение хозяйственной деятельности, местами нахождения (например, на воротах между подсобным помещением и торговым залом розничного магазина), тем самым устройства считывания сами по себе не определяют место нахождения без дополнительного указания направления движения объекта, снабженного радиочастотной меткой;

3) устройство считывания, имеющее несколько антенн с независимыми адресами, может быть использовано для обнаружения объектов, снабженных радиочастотными метками, находящихся в нескольких различных, с точки зрения персонала, ответственного за ведение хозяйственной деятельности, местах нахождения;

4) два или несколько устройств считывания могут быть использованы для обнаружения объектов, снабженных радиочастотными метками, находящихся в одном, с точки зрения персонала, ответственного за ведение хозяйственной деятельности, месте нахождения;

5) в случае мобильных устройств считывания данное устройство может считывать радиочастотные метки, которыми снабжены объекты, находящиеся в нескольких местах нахождения, при этом для определения конкретного места нахождения, связанного с данным событием считывания, могут использоваться «позиционированные» радиочастотные метки или иные средства;

6) места нахождения, представляющие интерес для одной стороны (торгового партнера или приложения), могут не представлять интерес для другой стороны или могут быть закрыты для просмотра другой стороной, что приводит к возникновению заинтересованности в поиске способов дифференциации мест нахождения.

Удовлетворение этих на первый взгляд противоречивых требований возможно при условии определения различных типов мест нахождения и соотношения между ними с последующим обеспечением корректной их записи для каждого события сбора данных с помощью приложения EPCIS сбора данных. Именно для этих целей события EPCIS содержат как идентификатор места считывания ReadPointID, так и идентификатор производственного места назначения BusinessLocationID (оба простых типа мест нахождения).

Кроме того, во избежание традиционной путаницы следует провести различие между «местом нахождения» как понятием, используемым приложениями на уровне EPCIS, и ключевыми идентификаторами устройств считывания. Настоящая спецификация EPCIS устанавливает четкое различие между местом нахождения и ключевыми идентификаторами устройств считывания. Для обеспечения полной ясности типы ключевых идентификаторов устройств считывания определены выше отлично от определений действительных типов мест нахождения EPCIS. Также типы ключевых идентификаторов устройств считывания могут вноситься в EPCIS в форме «атрибутов наблюдения» в тех случаях, когда приложению требуется хранить информацию о том, какие устройства считывания участвовали в процессе наблюдения, например, для целей аудита. (Сбор данных по «атрибутам наблюдения» и обмен этими данными требуют использования полей расширения, не определяемых в настоящей спецификации.)

7.3.5 Параметр «на каком основании»

Данный подраздел определяет типы данных, используемые в параметре «на каком основании» («Why») типов событий, указанных в 7.3.5.4.

7.3.5.1 Бизнес-этап (Business Step)

Вид лексики `BusinessStepID` (`Идентификатор_Бизнес-Этапа`) является лексиконом, элементы которого обозначают этапы бизнес-процесса. Примером служит идентификатор, обозначающий «отгрузку» («shipping»). Поле бизнес-этапа события определяет контекст ведения хозяйственной деятельности в рамках события: во время выполнения какого этапа бизнес-процесса было вызвано событие, данные о котором подлежат сбору. Лексикон `BusinessStepID` (`Идентификатор_Бизнес-Этапа`) является примером типовой лексики, как указано в 6.2.

Примечание — Использование расширяемого лексикона для идентификаторов бизнес-этапов позволяет стандартам GS1 (включая базовую деловую лексику GS1 по *ГОСТ Р 59168*) определять некоторые общие термины, такие как «отгрузка» («shipping») или «приемка» («receiving»), в то же время позволяя отраслевым группам и индивидуальным конечным пользователям определять свои собственные термины. Мастер-данные предоставляют дополнительную информацию.

Данная спецификация определяет атрибуты мастер-данных для идентификаторов бизнес-этапов.

7.3.5.2 Состояние (Disposition)

Вид лексики `DispositionID` (`Идентификатор_Состояния`) — это лексикон, элементы которого обозначают состояние объекта при проведении хозяйственной деятельности. Примером является идентификатор, который обозначает состояние «отозван» («recalled»). Поле состояния какого-либо события определяет условие ведения хозяйственной деятельности объектов события, следующих за событием. Предполагается, что состояние останется действительным до тех пор, пока другое событие не укажет на изменение состояния. Промежуточные события, которые не определяют поле состояния, не оказывают никакого влияния на предполагаемое состояние объекта. Лексикон `DispositionID` (`Идентификатор_Состояния`) является примером типовой лексики, как указано в 6.2.

Примечание — Использование расширяемого лексикона для идентификаторов состояния позволяет стандартам GS1 (включая в большей степени базовую деловую лексику GS1 по *ГОСТ Р 59168*) устанавливать некоторые общие термины, такие как «отозван» («recalled») или «в транзите» («in transit»), в то же время позволяя отраслевым группам и индивидуальным конечным пользователям определять свои собственные термины. Мастер-данные могут предоставлять дополнительную информацию.

Данная спецификация не определяет атрибуты мастер-данных для идентификаторов состояния.

7.3.5.3 Документ бизнес-транзакции (Business transaction)

Вид лексики `BusinessTransaction` (`Документ_Бизнес_Транзакции`) определяет конкретный документ бизнес-транзакции. Примером документа бизнес-транзакции является конкретный заказ на поставку. Информация о документе бизнес-транзакции может быть включена в события EPCIS для фиксирования приобщения события в конкретном документе бизнес-транзакции.

Документ бизнес-транзакции описан в стандарте EPCIS с помощью структурированного типа, состоящего из пары идентификаторов, как приведено *в таблице 12*.

Таблица 12

Поле	Вид лексики	Описание
type (тип)	BusinessTransactionTypeID (Идентификатор_Типа_Документа_Бизнес-транзакции)	(Необязательное) Идентификатор, который указывает, какой вид документа бизнес-транзакции указывает этот лексикон BusinessTransaction (Документ_Бизнес_Транзакции). Если этот параметр не задан, информация о типе документа бизнес-транзакции не доступна, кроме той, что вытекает из самого значения поля bizTransaction (документ_ Бизнес_Транзакции)
bizTransaction (документ_Бизнес-Транзакции)	BusinessTransactionID (Идентификатор_Документа_Бизнес-Транзакции)	Идентификатор, который обозначает конкретный документ бизнес-транзакции

В следующих подразделах определены два вида лексики: BusinessTransactionTypeID (Идентификатор_Типа_Документа_Бизнес-транзакции) и BusinessTransactionID (Идентификатор_Документа_Бизнес-Транзакции).

7.3.5.3.1 Тип документа бизнес-транзакции (Business transaction type)

Вид лексики BusinessTransactionTypeID (Идентификатор_Типа_ Документа_Бизнес-Транзакции) — это лексикон, элементы которого обозначают конкретные типы документов бизнес-транзакций. Примером является идентификатор, обозначающий «заказ на поставку» («purchase order»).

Лексикон BusinessTransactionTypeID (Идентификатор_Типа_Документа _Бизнес-Транзакции) является примером типовой лексики, как указано в 6.2.

Примечание — Использование расширяемого лексикона для идентификаторов типа документа бизнес-транзакции позволяет стандартам GS1 определять некоторые общие термины, такие как «заказ на поставку» («purchase order»), и в то же время предоставляет возможность отраслевым группам и индивидуальным конечным пользователям определять свои собственные термины. Мастер-данные могут предоставлять дополнительную информацию.

Данная спецификация не определяет атрибуты мастер-данных для идентификаторов типов документов бизнес-транзакций.

7.3.5.3.2 Идентификатор документа бизнес-транзакции (Business transaction ID)

Вид лексики BusinessTransactionID (Идентификатор_Документа_Бизнес-Транзакции) — это лексикон, элементы которого обозначают документы бизнес-транзакции. Примером является идентификатор, обозначающий «Acme Corp. purchase order number 12345678» (Заказ на поставку номер 12345678 организации Acme Corp.). Лексикон BusinessTransactionID (Идентификатор_Документа_ Бизнес-Транзакции) является пользовательской лексикой, как указано в 6.2.

Примечание — Для обеспечения расширяемости, а также в качестве удобного для организаций способа различения одного типа документа транзакции от другого, используются идентификаторы URI. Например, если организация Acme Corp. имеет заказы на поставку (один тип документа бизнес-транзакции), идентифицируемые с помощью восьмизначного номера, а также поставки (другой тип документа бизнес-транзакции), идентифицируемые с помощью шестизначной строки, и, кроме того, транспортное предприятие PostHaste использует 12-значные идентификаторы отслеживания, тогда с течением времени с конкретным номером EPC могут быть связаны следующие идентификаторы документов бизнес-транзакций:

Пример

gost.e-ecolog.ru

gost.e-ecolog.ru

urn:posthaste:tracking:123456789012

(В этом примере предполагается, что предприятие PostHaste зарегистрировало пространство имен URN «posthaste» в организации IANA (Internet Assigned Numbers Authority — «Администрация адресного пространства Интернет»). Приложение EPCIS доступа может запросить сервисы EPCIS и обнаружить все три идентификатора документов транзакций. Использование идентификаторов URI дает приложению путь к пониманию того, какой из идентификаторов представляет интерес.

7.3.5.4 Начальный пункт и конечный пункт (Source и Destination)

Source (Начальный_Пункт) или Destination (Конечный_Пункт) используются для обеспечения дополнительного контекста ведения хозяйственной деятельности, когда событие EPCIS является частью перемещения в процессе хозяйственной деятельности, то есть процесса, в котором происходит передача прав собственности, ответственности и/или владения физическими или цифровыми объектами.

Во многих случаях перемещение в процессе хозяйственной деятельности требует выполнения нескольких отдельных бизнес-этапов и поэтому нескольких событий EPCIS (например, доставка с последующим получением или более сложная последовательность этапов, такая как погрузка -> отправка -> транспортирование -> прибытие -> разгрузка -> приемка). ReadPoint (Идентификатор_Места_Считывания) и BusinessLocation (Идентификатор_Производственного_Места_Назначения) в параметре «где/куда» («where») этих событий EPCIS указывают известное физическое место нахождения каждого из этапов этого процесса. Source (Начальный_Пункт) и Destination (Конечный_Пункт), напротив, могут быть использованы для указания сторон и/или мест нахождения, которые являются предполагаемыми конечными пунктами перемещения в процессе хозяйственной деятельности. В многоэтапном перемещении в процессе хозяйственной деятельности некоторые или все события EPCIS могут иметь данные Source (Начальный_Пункт) и Destination (Конечный_Пункт), а информация будет одинаковой для всех событий в данном перемещении.

Source (Начальный_Пункт) и Destination (Конечный_Пункт) предоставляют стандартизированный способ представить участвующие стороны и/или физические места нахождения, вовлеченные в перемещение в процессе хозяйственной деятельности, дополняя информацию о документе бизнес-транзакции (например, заказы на поставку, счета-фактуры и т. п.), на которую могут ссылаться с помощью элементов лексикона BusinessTransaction (Документ_Бизнес_Транзакции).

Начальный пункт и конечный пункт описываются в EPCIS с помощью структурированного типа данных, состоящего из пары идентификаторов, следующим образом (*см. таблицу 13*).

Таблица 13

Поле	Вид лексики	Описание
type (тип)	SourceDestTypeID (Идентификатор_Типа_Начального_Пункта_Конечного_Пункта)	Идентификатор, который указывает, какой тип начального или конечного пункта этого Source (Начальный_Пункт) или Destination (Конечный_Пункт) (соответственно) обозначает
source (начальный_пункт) или destination (конечный_пункт)	SourceDestID (Идентификатор_Начального_Пункта_Конечного_Пункта)	Идентификатор, который обозначает конкретный начальный или конечный пункт

Два вида лексики SourceDestTypeID (Идентификатор_Типа_Начального_Пункта_Конечного_Пункта) и SourceDestID (Идентификатор_Начального_Пункта_Конечного_Пункта) определены ниже.

7.3.5.4.1 Тип начального пункта/конечного пункта (Source/Destination)

Вид лексики SourceDestTypeID (Идентификатор_Типа_Начального_Пункта_Конечного_Пункта) — это лексикон, элементы которого обозначают конкретные типы начального пункта и конечного пункта при перемещении в процессе хозяйственной деятельности. Примером является идентификатор, который обозначает «сторону-собственник» («owning party»). Лексикон SourceDestTypeID (Идентификатор_Типа_Начального_Пункта_Конечного_Пункта) является примером типовой лексики, как указано в 6.2.

Примечание — Использование расширяемого лексикона для идентификаторов типов начальных и конечных пунктов позволяет стандартам GS1 устанавливать некоторые общие термины, такие как «сторона-владелец», и в то же время предоставляет возможность отраслевым группам и индивидуальным конечным пользователям определять свои собственные термины. Мастер-данные могут предоставлять дополнительную информацию.

Данная спецификация определяет атрибуты мастер-данных для идентификаторов типов начальных и конечных пунктов.

7.3.5.4.2 Идентификатор начального пункта/конечного пункта (Source/Destination ID)

Вид лексики SourceDestID (Идентификатор_Начального_Пункта_Конечного_Пункта) — это лексикон, элементы которого обозначают конкретные начальные и конечные пункты. Примером

является идентификатор, который обозначает организацию «Acme Corporation (в качестве стороны владельца)». Лексикон SourceDestID (Идентификатор_Начального_Пункта_Конечного_ Пункта) является примером пользовательской лексики, как указано в 6.2.

Примечание — Для обеспечения расширяемости, а также удобного способа для организаций различить один вид идентификатора начального пункта и конечного пункта от другого, используются идентификаторы URI.

7.3.6 Мастер-данные экземпляра/партии (МДЭП)

Мастер-данные экземпляра/партии (МДЭП, ILMD) — это данные, описывающие конкретный экземпляр физического или цифрового объекта, или конкретной серии/партии объектов, которые производятся сериями/партиями. МДЭП состоят из набора описательных атрибутов, предоставляющих информацию об одном или нескольких конкретных объектах или партиях объектов. Они похожи на обычные мастер-данные, которые также состоят из набора описательных атрибутов, предоставляющих информацию об объектах. Однако в то время, как атрибуты мастер-данных имеют одинаковые значения для большого класса объектов (например, для всех объектов, имеющих данный номер GTIN), значения атрибутов МДЭП могут быть различными для гораздо меньших групп объектов (например, одной серии или партии) и могут быть различными для каждого объекта (то есть различными для каждого экземпляра объекта).

Примером атрибута мастер-данных являются вес и физические величины предметов торговли, идентифицируемых конкретным номером GTIN. Эти значения одинаковы для всей продукции, использующей данный номер GTIN. Примером МДЭП является дата истечения срока годности скоропортящегося предмета торговли. В отличие от мастер-данных дата истечения срока годности не одинакова для всех предметов торговли, имеющих один и тот же номер GTIN. В принципе каждый предмет торговли может иметь различную дату истечения срока годности, зависящую от даты его изготовления. Другие примеры МДЭП (ILMD) включают дату изготовления, место изготовления, вес и другие физические параметры предмета торговли переменной величины, информацию об урожае для сельскохозяйственной продукции и т. д.

МДЭП, как обычные мастер-данные, остаются статичными в течение всей жизни объекта. Например, дата истечения срока годности скоропортящегося предмета торговли или вес предмета переменной величины не изменяются в течение всей жизни этого предмета торговли, причем различные предметы торговли, имеющие один и тот же номер GTIN, могут иметь различные значения даты истечения срока годности и веса. МДЭП не подлежат использованию для представления информации, которая изменяется в течение жизни объекта, например текущая температура объекта в процессе его перемещения по цепи поставки.

В то время как стандарты для регистрации и распространения обычных мастер-данных в цепи поставок (такие, как номер GDSN) существуют, стандарты и системы для распространения МДЭП пока не созданы. По этой причине стандарт EPCIS позволяет непосредственно передавать данные МДЭП в определенных событиях EPCIS. Эта функция должна использоваться только тогда, когда не существует отдельной системы для распространения МДЭП.

Когда объект создается, для данного объекта определяются МДЭП. Поэтому МДЭП могут быть включены только в события типа ObjectEvent (Событие_с_Объектом) с полем ADD (ДОБАВЛЕНИЕ) (см. 7.4.1.2) и в события TransformationEvent (Событие_с_Преобразованием) (см. 7.4.6). В случае с TransformationEvent (Событие_с_Преобразованием) МДЭП применяется к выходным данным преобразования, а не к входным.

Структура МДЭП, определенная в данном стандарте EPCIS, состоит из набора поименованных атрибутов, со значениями любого типа. В привязках XML (см. 9.5) схема XML предоставляет неограниченный список элементов XML, имеющих любое имя элемента и содержание. Другие документы, дополняющие стандарт EPCIS, могут определять конкретные элементы данных МДЭП, см. 6.3. Таким образом, МДЭП похожи на расширения EPCIS на уровне события, но отличаются тем, что МДЭП применяются в течение всего времени жизни объектов, в то время как расширение EPCIS на уровне события применяется только к этому конкретному событию.

7.4 Модуль типов основных событий. События

7.4.1 Событие_EPCIS (EPCISEvent)

Тип событий EPCISEvent (Событие_EPCIS) — это общий базовый тип для всех событий EPCIS. Все более конкретные типы событий в следующих подразделах являются подклассами EPCISEvent (Событие_EPCIS).

Этот общий базовый тип имеет только следующие поля, *приведенные в таблице 14*.

Таблица 14

Поле	Тип данных	Описание
eventTime (время_События)	Time (Время)	Дата и время возникновения события, о котором заявляют приложения EPCIS сбора данных
recordTime (время_Записи)	Time (Время)	(Необязательное) Дата и время записи события репозиторием EPCIS. Это поле ДОЛЖНО игнорироваться, когда событие предъявлено интерфейсу EPCIS сбора данных, и ДОЛЖНО быть представлено, когда событие извлекается через интерфейсы EPCIS запросов. Поле recordTime (время_Записи) не содержит какого-либо описания события реального мира, это скорее механизм учета, который играет роль в интерпретации длительных запросов, как указано в 8.2.5.2
eventTimeZoneOffset (смещение_Часового_Пояса_События)	String (Строка)	Смещение часового пояса фактически в момент времени и в месте возникновения события, выражаемое как смещение относительно всемирного координированного времени UTC). Значением этого поля ДОЛЖНА быть строка, состоящая из знака «+» или знака «-», за которым следуют две цифры, значения которых лежат в пределах от 00 до 14 (включительно), затем следует знак «двоеточие» «:», затем — две цифры, значения которых находятся в пределах от 00 до 59 (включительно), за исключением случая, когда значения первых двух цифр 14: тогда значениями вторых двух цифр должно быть 00. Например, значение +05:30 означает, что в месте, где произошло событие, местное время было на пять часов 30 минут позже, чем по всемирному координированному времени UTC (то есть полночь по всемирному координированному времени UTC соответствует 5:30 утра местного времени)
eventID (идентификатор_События)	EventID (Идентификатор_События)	(Необязательное) Идентификатор для указанного события, определенный приложением сбора данных, являющийся уникальным в глобальном масштабе среди всех событий, кроме заявлений об ошибке. «Уникальный в глобальном масштабе» означает отличный от идентификаторов любых иных событий EPCIS в любом применении EPCIS, а не только среди событий, зафиксированных одним приложением сбора данных или одним сервером сбора данных. (Для этих целей в стандарте базовой деловой лексики по *ГОСТ Р 59168* предусмотрено использование идентификатора UUID URI.) Следует заметить, что в случае заявления об ошибке идентификатор события аналогичен идентификатору ошибочного события (или имеет нулевое значение, если значение идентификатора ошибочного события является нулевым) и в данном смысле не является уникальным (см. 7.4.1.2)
errorDeclaration (заявление_об_ошибке)	ErrorDeclaration (Заявление_об_Ошибке)	(Необязательное) Наличие такого поля означает, что указанное событие служит утверждением о том, что утверждения, сделанные предшествующим событием, являются ошибочными (см. 7.4.1.2)

7.4.1.1 Объяснение для поля eventTimeZoneOffset (смещение _Часового _Пояса_События) (справочная информация)

Наличие поля eventTimeZoneOffset (смещение _Часового _Пояса _События) не обязательно для понимания того, в какой момент времени произошло событие. Причина в том, что поле eventTime (время _События) имеет тип Time (Время), определяемый в 7.3 как «дата и время в формате, независимом от часового пояса». Например, в привязке XML (см. 9.5) поле eventTime (время _События) представляется как элемент типа xsd:dateTime, а в 9.5 далее оговаривается, что представление XML должно содержать спецификатор часового пояса. Поэтому представление XML для поля eventTime (время _События) однозначно идентифицирует момент времени в абсолютном формате, и отсутствует необходимость в информации о значении поля eventTimeZoneOffset (смещение _Часового _Пояса _События) для понимания истинного значения времени.

Назначением поля eventTimeZoneOffset (смещение _Часового _Пояса _События) является обеспечение дополнительного контекста ведения хозяйственной деятельности для события,

а именно идентификации того, каким фактически было смещение часового пояса в том месте и времени, когда было зафиксировано событие. Эта информация может быть полезной, чтобы определить, например, произошло ли событие в рабочие часы, и представить событие персоналу в формате, соответствующем местному времени, и т. д. Информация о смещении местного часового пояса не обязательно доступна из поля `eventTime (время _События)`, потому что не существует требований к тому, что спецификатор часового пояса в представлении XML поля `eventTime (время _События)` должен быть смещением местного часового пояса, в котором было зафиксировано событие. Например, событие, происходящее в 8:00 утра по восточному стандартному времени США (US Eastern Standard Time), может иметь поле XML `eventTime (время _События)`, имеющее вид 08:00-05:00 (используя восточное стандартное время США), или 13:00Z (по всемирному координированному времени UTC), или даже 07:00-06:00 (используя центральное стандартное поясное время США (US Central Standard Time). Более того, документ *[18]* не требует от процессоров XML сохранять и представлять приложениям спецификатор часового пояса, который был частью поля `xsd:dateTime`, и, таким образом, спецификатор часового пояса в поле `eventTime (время _События)` может быть и вовсе не доступен приложениям. Аналогичные соображения будут применяться для других (не XML) привязок модуля типов основных событий. Например, гипотетическая двоичная привязка может представлять значения `Time (Время)` как миллисекундное смещение относительно полуночи по всемирному координированному времени UTC на 1 января 1970 года — опять же, однозначно идентифицирующее момент в абсолютном времени, но не предоставляющее никакой информации о местном часовом поясе. По этим причинам поле `eventTimeZoneOffset (смещение _Часового _Пояса _События)` предусмотрено как дополнительное поле события.

7.4.1.2 Элемент ErrorDeclaration (Заявление_об_Ошибке)

Наличие в событии элемента `ErrorDeclaration (Заявление_об_Ошибке)` указывает на то, что данное событие имеет особую семантику: в отличие от нормальной семантики, которая содержит утверждение о том, что какие-либо события произошли, или о том, что какие-либо факты являются верными после того, как событие произошло, семантика данного события содержит утверждение о том, что предшествующие утверждения являются ошибочными. Событие, содержащее элемент `ErrorDeclaration (Заявление_об_Ошибке)`, ДОЛЖНО быть во всех остальных отношениях идентичным предшествующему событию, при этом «во всех остальных отношениях» означает, что все поля события, за исключением элемента `ErrorDeclaration (Заявление_об_Ошибке)` и значения поля `recordTime (время_Записи)`, в точности совпадают с предшествующим событием. (Следует заметить, что включает поле `eventID (идентификатор_События)`: значение поля `eventID (идентификатор_События)` заявления об ошибке будет равно значению поля `eventID (идентификатор_События)` предшествующего события или будет нулевым, если значение поля `eventID (идентификатор_События)` предшествующего события является нулевым. Это представляет собой единственный случай, когда одно и то же отличное от нуля значение поля `eventID (идентификатор_События)` может присутствовать в двух событиях.) Семантика события, содержащего элемент `ErrorDeclaration (Заявление_об_Ошибке)`, заявляет о том, что все утверждения, подразумеваемые предшествующим событием, считаются ошибочными с момента, указанного в поле `declarationTime`. Предшествующее событие никаким образом не модифицируется, и в ответ на последующие запросы будет поступать как предшествующее событие, так и заявление об ошибке.

Элемент `ErrorDeclaration (Время_Заявления)` содержит следующие поля (*см. таблицу 15*):

Таблица 15

Поле	Тип данных	Описание
`declarationTime (время_Заявления)`	`Timestamp (Отметка_Времени)`	Дата и время заявления об ошибке. (Следует заметить, что значение поля eventTime (время_ События) данного события должно соответствовать значению поля eventTime (время_События) предшествующего события, объявляемого ошибочным, поэтому в поле declarationTime должно быть указано заявленное время этого события)
`reason (причина_Ошибки)`	`ErrorReasonID (Идентификатор_Причины_Ошибки)`	(Необязательное) Элемент типовой лексики, указывающий причину, по которой предшествующее событие считается ошибочным

Окончание таблицы 15

Поле	Тип данных	Описание
correctiveEventIDs (идентификаторы_Корректирующих_Событий)	List<EventID> (Список<Идентификаторов_Событий>)	(Необязательное) Наличие данного поля указывает на то, что события, имеющие приведенные идентификаторы URI в качестве значения полей eventID (идентификатор_События), должны считаться «корректирующими» в отношении события, объявленного данным событием ошибочным. Оно предоставляет способ привязывания события заявления об ошибке к одному или нескольким событиям, которые предназначены для замены ошибочного события

Элемент ErrorDeclaration (Заявление_об_Ошибке) НЕ ДОЛЖЕН использоваться, если имеется возможность смоделировать действительную ситуацию в виде обычного события (то есть с использованием события, которое не содержит элемент ErrorDeclaration (Заявление_об_Ошибке).

7.4.1.3 Примеры использования заявлений об ошибках (справочная информация)

Событие EPCIS фиксирует завершение этапа бизнес-процесса. Бизнес-процесс моделируется посредством разделения его на ряд этапов, каждый из которых моделируется в виде события EPCIS. В результате набор всех событий, относящихся к определенному объекту (часто называемый «прослеживание»), должен корректно отображать историю и текущее состояние данного объекта посредством интерпретации событий в соответствии с семантикой, описанной в настоящем стандарте и соответствующих лексических стандартах.

Иногда оказывается, что ранее зафиксированное событие некорректно отображает то, что произошло в действительности. В таких случаях в соответствии с 6.1 ранее произошедшие события не удаляются и не модифицируются. Вместо этого фиксируются новые события, в результате чего полная запись (включая новые события и все предшествующие события, включая некорректное) точно отображает историю и текущее состояние в соответствии с вышеизложенным принципом.

Предпочтительным способом отражения наступления дополнительных событий является признание того факта, что обнаружение ошибочного события и его исправление также представляют собой бизнес-процесс, который можно смоделировать путем создания соответствующих событий EPCIS. В большинстве случаев это осуществляется с использованием событий EPCIS из модуля типов основных событий в соответствии с пунктами 7.4.1—7.4.6 с применением соответствующего лексикона.

Пример 1: Предприятие X фиксирует событие EPCIS, утверждающее, что продукция с серийными номерами 101, 102, 103 была отгружена предприятию Y. Предприятие Y по получении груза обнаружило в дополнение к продукции с серийными номерами 101, 102, 103 продукцию с серийным номером 104. В процессе взаимодействия с Предприятием X было установлено, что продукция с серийным номером 104 действительно была отгружена и что событие отгрузки было ошибочным. Корректирующие действия: Предприятие X фиксирует новое событие EPCIS, утверждающее, что продукция с серийным номером 104 была отгружена, с контекстной информацией, как у изначального события.

Пример 2: Предприятие X фиксирует событие EPCIS, утверждающее, что продукция с серийными номерами 101, 102, 103 была отгружена предприятию Y. Предприятие Y по получении груза обнаружило только продукцию с серийными номерами 101, 102. В процессе взаимодействия с предприятием X было установлено, что продукция с серийным номером 103 не была отгружена и осталась в наличии у предприятия X. Достигается соглашение об отмене счета, выставленного за продукцию с серийным номером 103. Корректирующие действия: Предприятие X фиксирует новое событие EPCIS, признающее недействительной поставку продукции с серийным номером 103.

В примере 1 в дополнительном событии используется тот же самый деловой лексикон, что и в первоначальном событии. В примере 2 используется лексикон, ассоциированный с процессом признания поставки недействительной, но при этом он не выходит за рамки «обычной» семантики EPCIS в том смысле, что он моделирует завершение четко определенного этапа бизнес-процесса. Тем самым подтверждается факт того, что корректирующее действие представляет собой бизнес-процесс, который может быть смоделирован в виде события EPCIS.

В некоторых случаях представляется невозможным (или крайне нежелательным) корректировать историю объекта посредством создания нового события EPCIS с обычной семантикой (то есть с семантикой, описанной в 7.4.1—7.4.6).

Пример 3: Предприятие X фиксирует событие EPCIS, утверждающее, что продукция X с серийным номером 101 была уничтожена. Данное событие является событием с объектом (Object Event) в соответствии с 7.4.2 со значением действия = DELETE (УДАЛЕНИЕ). Впоследствии обнаруживается, что продукция с серийным номером 101 все еще находится на хранении, а не уничтожена. Для исправления истории нельзя использовать обычное событие, поскольку семантика значения действия DELETE (УДАЛЕНИЕ) для события объекта говорит о том, что «данный объект не может появляться в последующих событиях».

Пример 4: Предприятие X фиксирует событие EPCIS, утверждающее, что были отгружены несколько изделий, и при этом указывающее в параметре «на каком основании» заказ № 123. Предприятие Y получает товар и фиксирует событие получения. Уже после этого выясняется, что ссылка на заказ в событии отгрузки неверна: там указан заказ № 456 вместо № 123. Эту ошибку можно было бы исправить посредством обычных событий EPCIS: Предприятие X может зафиксировать событие «отмены поставки» и создать новое событие «поставки» с указанием правильного номера заказа. Но такой способ коррекции является нежелательным с точки зрения всеобъемлющего прослеживания, особенно с учетом того, что уже имело место событие получения.

Для решения подобных ситуаций в модуле типов основных событий предусмотрен механизм, с помощью которого декларируется, что утверждения, вытекающие из предшествующего события, являются ошибочными. При использовании данной семантики в событии обязательно должны быть указаны те же самые условия, что и в предшествующем обычном событии, с тем чтобы утверждение об ошибке можно было соотнести с предшествующим событием. Такое событие называют «событием заявления об ошибке».

В примере 3 событие заявления об ошибке подразумевает, что продукция X с серийным номером 101 не была уничтожена. В примере 4 событие заявления об ошибке подразумевает, что поставка в соответствии с заказом № 123 не имела места, а дополнительное («корректирующее») событие будет утверждать, что произошла поставка в соответствии с заказом № 456. Этот способ подобен моделированию примера 4 с использованием события «отмены поставки», но вместо утверждения о том, что отгрузка была осуществлена в соответствии с заказом № 123 и потом аннулирована, событие заявления об ошибке просто заявляет, что утверждение по заказу № 123 было ошибочным.

Событие заявления об ошибке формируется посредством добавления части `ErrorDeclaration (Заявление_об_Ошибке)`. Так, по отношению к событию С1 структура события заявления об ошибке С2, целью которого является утверждение об ошибочности утверждений, вытекающих из С1, будет аналогична по содержанию С1, но с добавлением элемента `ErrorDeclaration`. Например, заявление об ошибке в отношении события «уничтожения» в примере 3 тоже является событием с объектом (Object Event) со значением действия = DELETE (УДАЛЕНИЕ), но с добавлением элемента `ErrorDeclaration (Заявление_об_Ошибке)`. В целом для того, чтобы объявить событие С ошибочным, фиксируют новое событие, идентичное событию С, но с добавлением элемента `ErrorDeclaration (Заявление_об_Ошибке)` (и отличающимся от С временем фиксирования события).

Существуют три причины, по которым события заявления об ошибке фиксируются в EPCIS указанным образом. Во-первых, идентификатор события не требуется для указания на ошибочное событие, а следовательно, не обязательно включать идентификатор события в каждое событие, чтобы обеспечить возможность заявления об ошибке в будущем. Идентификаторы событий могут использоваться для привязки события заявления об ошибке к корректирующему событию, но использование идентификаторов событий не является обязательным. Во-вторых, любой запрос EPCIS, соответствующий некоему событию, также будет соответствовать заявлению об ошибке в отношении этого события, если таковое существует. Благодаря этому приложениям EPCIS доступа не требуется специальная логика для обнаружения существующих заявлений об ошибках. В-третьих, если приложение EPCIS доступа получает событие заявления об ошибке, но при этом по какой-либо причине не имеет копии первоначального (ошибочного) события, нет необходимости в восстановлении первоначального события, так как вся информация по данному событию также имеется в событии заявления об ошибке.

7.4.1.4 Установление соответствия заявления об ошибке первоначальному событию (справочная информация)

Как уже было указано в 7.4.1.2, событие заявления об ошибке по содержанию идентично первоначальному (ошибочному) событию, отличаясь лишь наличием самого элемента `ErrorDeclaration (Заявление_об_Ошибке)` и временем фиксирования. Одно из преимуществ такого подхода состоит в том, что, когда приложение EPCIS доступа обнаруживает заявления об ошибке, нет необходимости в восстановлении первоначального (ошибочного) события, так как вся информация, содержащаяся в том событии, также присутствует в событии заявления об ошибке, которое уже имеется в наличии у приложения EPCIS доступа.

Тем не менее возможны ситуации, при которых приложению EPCIS доступа или приложению EPCIS сбора данных требуется подтвердить существование первоначального (ошибочного) события посредством запроса. Единственным способом подтверждения того, что некоторое событие является первоначальным событием, соответствующим заявлению об ошибке, является подтверждение соответствия всех элементов данных этих событий (кроме элемента ErrorDeclaration (Заявление_об_Ошибке) и времени фиксирования). Рекомендации по формированию запросов в таких ситуациях содержатся в [7].

7.4.2 Событие_с_Объектом (ObjectEvent) (подкласс EPCISEvent)

Тип событий ObjectEvent (Событие_с_Объектом) предназначен для сбора информации о событии, имеющем отношение к одному или нескольким физическим или цифровым объектам, обозначенным идентификаторами на уровне экземпляра (EPC) или на уровне класса (EPC Class). Большинство значений ObjectEvent (Событие_с_Объектом) предусмотрено для представления фактических результатов отслеживания объектов, но, строго говоря, их можно использовать для любого события, о котором приложение сбора данных намеревается заявить, что событие касалось указанных объектов, включая, например, фиксирование факта того, что ожидаемое отслеживание не произошло.

В то время как в ObjectEvent (Событие_с_Объектом) может присутствовать более одного идентификатора EPC на уровне экземпляра и/или на уровне класса EPC Class, отношений или связей между этими объектами не подразумевается, кроме случая, когда в реальном мире произошли одинаковые события.

Поле action (действие) для ObjectEvent (Событие_с_Объектом) описывает отношения событий к жизненному циклу объектов и их идентификаторов, указанных в событии (*см. таблицу 16*).

Таблица 16

Значение поля action (действие)	Содержание
ADD (ДОБАВЛЕНИЕ)	Объекты, идентифицированные в рамках события, были введены как часть этого события. Для объектов, идентифицированных с помощью идентификаторов EPC на уровне экземпляра, номер(а) EPC был(и) выпущен(ы) и связан(ы) с объектом(объектами) в первый раз. Для объектов, идентифицированных с помощью идентификаторов на уровне класса EPC Class, были созданы указанные количества объектов на уровне класса, идентифицированных в событии (хотя другие экземпляры тех же объектов на уровне классов, возможно, существовали до этого события, а дополнительные экземпляры могут быть созданы после этого события)
OBSERVE (ОТСЛЕЖИВАНИЕ)	Событие представляет собой простое отслеживание объектов, идентифицированных в событии, а не их введение или выведение
DELETE (УДАЛЕНИЕ)	Объекты, идентифицированные в этом событии, списаны, как часть этого события. Для объектов, идентифицированных с помощью идентификаторов EPC на уровне экземпляра, номера EPC не существуют после события и не смогут быть зарегистрированы снова. Для объектов, идентифицированных с помощью идентификаторов на уровне класса EPC Class, указанные количества объектов на уровне класса EPC Class, определенных в событии, прекратили существование (хотя другие экземпляры тех же объектов на уровне классов могли продолжить существование после этого события, а дополнительные экземпляры могли прекратить существование до этого события)

Поля определены в *таблице 17*:

Таблица 17

Поле	Тип	Описание
eventTime (время_События) recordTime (время_Записи) eventTimeZoneOffset (смещение_Часового_Пояса_События)	(унаследован от общего базового типа EPCISEvent (Событие_EPCIS); см. 7.4.1)	

Окончание таблицы 17

Поле	Тип	Описание
epcList (список_EPC)	List<EPC> (Список<EPC>)	(Необязательное) Неупорядоченный список из одного или нескольких номеров EPC, именующих конкретные объекты, к которым имело отношение событие. См. 7.3.3.2. Элемент ObjectEvent (Событие_с_ Объектом) должен содержать либо непустое поле epcList (список_EPC), либо непустое поле quantityList (список_ Количество), или оба этих непустых поля
quantityList (список_Количество)	ListQuantityElement (Список<Элементы_Количество>)	(Необязательное) Неупорядоченный список из одного или нескольких элементов QuantityElements (Элемент_ Количество), идентифицирующих (на уровне класса) объекты, к которым имело отношение событие. Элемент ObjectEvent (Событие_с_ Объектом) должен содержать либо непустое поле epcList (список_EPC), либо непустое поле quantityList (список_ Количество), или оба этих непустых поля
action (действие)	Action (Действие)	Как данное событие относится к жизненному циклу номеров EPC, участвующих в данном событии. См. выше для более подробной информации
bizStep (бизнес-этап)	BusinessStepID (Идентификатор_Бизнес-Этапа)	(Необязательное) Бизнес-этап, частью которого было данное событие
disposition (состояние	DispositionID (Идентификатор_Состояния)	(Необязательное) Условие ведения хозяйственной деятельности для объектов, связанных с номерами EPC, значение которого считается верным до тех пор, пока оно не будет противоречить последующему событию
readPoint (место_Считывания)	ReadPointID (Идентификатор_Места_Считывания)	(Необязательное) Место считывания, в котором произошло событие
bizLocation (производственное_Место_Назначения)	BusinessLocationID (Идентификатор_Производственного_Места_Назначения)	(Необязательное) Производственное место назначения, в котором могут быть обнаружены объекты, связанные с номерами EPC, до тех пор, пока оно не будет противоречить последующему событию
bizTransactionList (список_Документов_ Бизнес-транзакций)	Перечисленный список, состоящий из 0 или более элементов BusinessTransaction (Документ_ Бизнес_ Транзакции)	(Необязательное) Неупорядоченный список документов бизнес-транзакций, который определяет контекст данного события
sourceList (список_ Начальных_Пунктов)	List<Source> (Список<Начальных_ Пунктов>)	(Необязательное) Неупорядоченный список элементов Source (Начальный_Пункт) (7.3.5.4), который предоставляет контекст об исходной точке перемещения в процессе хозяйственной деятельности, частью которого является данное событие
destinationList (список_Конечных_Пунктов)	List<Destination> (Список< Конечных_ Пунктов>)	(Необязательное) Неупорядоченный список элементов Destination (Конечный_ Пункт) (7.3.5.4), который предоставляет контекст о конечной точке перемещения в процессе хозяйственной деятельности, частью которого является данное событие
ilmd (мдэс)	ILMD (МДЭС)	(Необязательное) Мастер-данные экземпляра/партии (7.3.6), которые определяют объекты, созданные во время данного события. Тип события ObjectEvent (Событие_с_Объектом) НЕ ДОЛЖЕН содержать поле ilmd (мдэс), если поле action (действие) принимает значение OBSERVE или DELETE (УДАЛЕНИЕ)

Следует обратить внимание, что в привязке XML (см. 9.3) поля quantityList (список_Количество), sourceList (список_Начальных_Пунктов), destinationList (список_Конечных_Пунктов) и ilmd (мдэс) появляются в стандартной области расширения, чтобы поддержать совместимость с [3].

Ретроспективная семантика:

- событие, описанное с помощью поля bizStep (бизнес-этап) (и любых других полей), имело место в отношении объектов, идентифицированных полями epcList (список_EPC) и quantityList (список_Количество) в момент времени, указанный в поле eventTime (время_События), в месте нахождениия, указанном в поле readPoint (место_Считывания);
- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) номера EPC в поле epcList (список_EPC) были введены для использования (присвоены в первый раз);
- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) в поле quantityList (список_Количество) было создано определенное количество экземпляров на уровне класса EPC (или неизвестное количество для каждого элемента QuantityElement (Элемент_Количество), в котором значение поля quantity (количество) не задано);
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ)) номера EPC в поле epcList (список_EPC) были выведены из эксплуатации (удалены для будущего использования);
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ)) определенное число экземпляров на уровне класса EPC в поле quantityList (список_Количество) прекратило существование (или неизвестное количество для каждого элемента QuantityElement (Элемент_Количество), в котором значение поля quantity (количество) не задано);
- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ) и задано непустое значение поля bizTransactionList (список_Документов_Бизнес-транзакций)) существует связь между документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), и объектами, указанными в поле epcList (список_EPC) и поле quantityList (список_Количество);
- (если значением поля action (действие) является OBSERVE (ОТСЛЕЖИВАНИЕ) и задано непустое значение поля bizTransactionList (список_Документов_Бизнес-транзакций) данное событие произошло в контексте документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ) и задано непустое значение поля bizTransactionList (список_Документов_Бизнес-транзакций)) данное событие произошло в контексте документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);
- (если значение поля sourceList (список_Начальных_Пунктов) не пустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, начальная точка которого определяется источниками, перечисленными в поле sourceList (список_Начальных_Пунктов);
- (если значение поля destinationList (список_Конечных_Пунктов) не пустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, конечная точка которого определяется конечными пунктами, перечисленными в поле destinationList (список_Конечных_Пунктов).

Перспективная семантика:

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) объекты, обозначенные идентификаторами на уровне экземпляра в поле epcList (список_EPC), могут появиться в последующих событиях;
- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) объекты, обозначенные идентификаторами на уровне класса в поле quantityList (список_Количество), могут появиться в последующих событиях;
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ)) объекты, обозначенные идентификаторами на уровне экземпляра в поле epcList (список_EPC), не должны появиться в последующих событиях;
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ)) общее число объектов, обозначенных идентификаторами на уровне класса в поле quantityList (список_Количество), которые могут появиться в последующих событиях, было уменьшено с помощью значений, заданных в поле quantityList (список_Количество) (или с помощью неизвестного значения количества для каждого элемента QuantityElement (Элемент_Количество), в котором значение поля quantity (количество) не задано);

- (если значение поля disposition (состояние) задано) условие ведения хозяйственной деятельности для объектов, определенных полями epcList (список_EPC) и quantityList (список_Количество), является таким, каким оно задано полем disposition (состояние);

- (если значение поля disposition (состояние) не задано) условие ведения хозяйственной деятельности для объектов, связанных с объектами, определенными в поле epcList (список_EPC) или поле quantityList (список_Количество), не изменяется;

- (если значение поля bizLocation (производственное_Место_ Назначения) задано) объекты, определенные полями epcList (список_EPC) и quantityList (список_Количество), находятся в производственном месте назначения, указанном в поле bizLocation (производственное_Место_ Назначения);

- (если значение поля bizLocation (производственное_Место_ Назначения) не задано) производственное место назначения объектов, определенных полями epcList (список_EPC) и quantityList (список_ Количество), неизвестно;

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ) и значение поля ilmd (МДЭС) не пустое) объекты, определенные полями epcList (список_EPC) и quantityList (список_Количество), описываются атрибутами в поле ilmd (МДЭС);

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ) и задано непустое значение поля bizTransactionList (список_Документов_ Бизнес-транзакций)) существует связь между документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), и объектами, определенными в поле epcList (список_EPC) и поле quantityList (список_Количество).

Примечание — В случае, когда значением поля действия (action) является ADD (ДОБАВЛЕНИЕ) и задано не пустое значение поля bizTransactionList (список_Документов_Бизнес-транзакций), семантический эффект эквивалентен наличию события с объектом (ObjectEvent) без поля bizTransactionList (список_Документов_Бизнес-транзакций) вместе с событием с документом транзакции (TransactionEvent), имеющим поле bizTransactionList (список_Документов_Бизнес-транзакций), и те же значения полей, что и события с объектом (ObjectEvent). Следует заметить, что событие с объектом (ObjectEvent) с непустым полем bizTransactionList (список_Документов_Бизнес-транзакций) не вызывает возврата события с документом транзакции (TransactionEvent) из запроса.

7.4.3 Событие_с_Агрегацией (AggregationEvent) (подкласс EPCISEvent)

Тип событий AggregationEvent (Событие_с_Агрегацией) описывает события, которые применяются к объекту, объединенному с другим объектом. В таком событии имеется набор «содержащихся» объектов, которые были объединены внутри «содержащего» их объекта, что означает определение агрегации как таковой.

Данный тип событий предназначен для использования для «агрегаций», т. е. объединения, в котором существуют сильные физические взаимосвязи между содержащим и содержащимися объектами. При этом все они будут занимать одно и то же место нахождения в одно и то же время до тех пор, пока не будут разъединены. Примером агрегации является случай, когда коробки загружаются на поддон и перемещаются как единое целое. Тип событий AggregationEvent (Событие_с_Агрегацией) не предназначен для более слабых объединений, таких как два поддона, являющиеся частью одной и той же партии, но в которой эти поддоны не всегда могут находиться в одном и том же месте и в одно и то же время (в таких обстоятельствах может подойти тип событий TransactionEvent (Событие_с_Документом_Транзакции). В зависимости от бизнес-этапа может быть задано больше конкретных семантических значений.

Поле action (действие) события типа AggregationEvent (Событие_с_Агрегацией) описывает отношение события к жизненному циклу агрегации (*см. таблицу 18*).

Таблица 18

Значение поля action (действие)	Содержание
ADD (ДОБАВЛЕНИЕ)	Объекты, идентифицированные в списке потомков, были агрегированы объектом-родителем во время этого события. Сюда относятся ситуации, когда агрегация создается впервые, а также когда новые потомки добавляются в существующую агрегацию объектов

Окончание таблицы 18

Значение поля action (действие)	Содержание
OBSERVE (ОТСЛЕЖИВАНИЕ)	Событие не представляет собой ни добавление, ни удаление объектов-потомков из агрегации. Отслеживание может быть неполным: могут существовать потомки, которые являются частью агрегации, но не отслеживаются во время этого события и, следовательно, не включены в поле childEPC (потомок_EPC) или поле childQuantityList (список_Количества_Потомков) типа события AggregationEvent (Событие_с_Агрегацией). Аналогично идентификатор объекта-родителя может не отслеживаться или не быть известным в течение этого события, и поэтому поле parentID (идентификатор_Родителя) не должно быть пропущено в AggregationEvent (Событие_с_Агрегацией)
DELETE (УДАЛЕНИЕ)	Объекты, идентифицированные в списке потомков, были дезагрегированы из объекта-родителя во время этого события. Сюда относятся ситуации, когда подмножество объектов-потомков удаляется из агрегации, а также при разборке всей агрегации. И поле childEPC (потомок_EPC), и поле childQuantityList (список_Количества_Потомков) могут отсутствовать в типе события AggregationEvent (Событие_с_Агрегацией), и это означает, что все потомки дезагрегированы.

Тип событий AggregationEvent (Событие_с_Агрегацией) включает поля, которые ссылаются на единственный «объект-родитель» (часто это — «содержащий» объект) и один или несколько «объектов-потомков» (часто это — «содержащиеся» объекты). Если поле action (действие) имеет значение ADD (ДОБАВЛЕНИЕ) или DELETE (УДАЛЕНИЕ), то требуется родительский идентификатор, но он не обязателен, если поле action (действие) имеет значение OBSERVE (ОТСЛЕЖИВАНИЕ).

Примечание — Если поле action (действие) имеет значение ADD (ДОБАВЛЕНИЕ), то используется идентификатор объекта-родителя, и поэтому существует способ сослаться на объединение в последующих событиях, если значением поля action (действие) является DELETE (УДАЛЕНИЕ). Идентификатор объекта-родителя является необязательным, если поле action (действие) имеет значение OBSERVE (ОТСЛЕЖИВАНИЕ), потому что объект-родитель не всегда известен во время непосредственного отслеживания. Например, процесс приемки поддона для определения номеров EPC может опираться на радиочастотные метки, нанесенные на коробки, собранные на поддоне, но на самом поддоне радиочастотная метка может отсутствовать (или если она есть, но не считывается).

Тип событий AggregationEvent (Событие_с_Агрегацией) предназначен для указания агрегаций среди объектов, поэтому потомки идентифицируются с помощью номеров EPC и/или классов EPC. Объект-родитель, тем не менее, необязательно является физическим или цифровым объектом, отделенным от самой агрегации, и поэтому объект-родитель идентифицируется с помощью произвольного идентификатора в формате URI, который МОЖЕТ быть номером EPC, но МОЖЕТ быть и другим идентификатором, заимствованным из подходящего частного лексикона.

Примечание — Например, во многих производственных операциях обычно груз проходит несколько этапов до присвоения ему номера EPC . В таких ситуациях в момент формирования груза ему присваивается внутренний контрольный номер (часто называемый «номерным знаком» («license plate number», LPN), который используется вплоть до момента отгрузки. В момент отгрузки присваивается серийный номер транспортной упаковки (Serial Shipping Container Code – SSCC), который является номером EPC. В стандарте EPCIS это моделируется с помощью (a) AggregationEvent (Событие_с_Агрегацией) со значением поля action (действие), соответствующим ADD (ДОБАВЛЕНИЕ), в момент формирования груза, и (b) второго события типа AggregationEvent со значением action (действие), соответствующим ADD (ДОБАВЛЕНИЕ), в момент присвоения номера SSCC (первое объединение может быть также аннулировано через событие типа AggregationEvent со значением action (действие), соответствующим DELETE (УДАЛЕНИЕ)). Первое событие типа AggregationEvent (Событие_с_Агрегацией) будет использовать номер LPN как идентификатор родителя (выраженный в подходящем представлении идентификатора URI; см. 6.4), тогда как второе событие типа AggregationEvent (Событие_с_Агрегацией) будет использовать номер SSCC (который является разновидностью EPC) как идентификатор родителя, таким образом *изменяя* поле parentID (идентификатор_Родителя).

Тип событий AggregationEvent (Событие_с_Агрегацией) имеет следующие поля (*см. таблицу 19*):

Таблица 19

Поле	Тип поля	Описание
eventTime (время_События) recordTime (время_Записи) eventTimeZoneOffset (смещение_Часового_Пояса_События)	(Наследуется от типа EPCISEvent (Событие_EPCIS); см. 7.4.1)	
parentID (идентификатор_Родителя)	URI	(Необязательное, если поле action (действие) имеет значение OBSERVE (ОТСЛЕЖИВАНИЕ), в иных случаях обяза-тельное) Идентификатор объекта-родителя объединения. Если идентификатор родителя является номером EPC, это поле ДОЛЖНО содержать «чистый ключевой идентификатор» URI для номера EPC, как указано в [12, раздел 7]
childEPCs (номера_EPC_Потомков)	List<EPC> (Список<EPC>)	(Необязательное) Неупорядоченный список номеров EPC «содержащихся» объектов, обозначенных с помощью идентификаторов на уровне экземпляра (см. 7.3.3.2). Событие типа AggregationEvent (Событие_ с_Агрегацией) ДОЛЖНО содержать либо непустое поле childEPCs (номера_EPC_Потомков), либо непустое поле childQuantityList (список_Количества_Потомков), или оба непустых поля, за исключением случая, когда оба поля, childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), могут быть пустыми, если поле action (действие) имеет значение DELETE (УДАЛЕНИЕ), показывая, что все объекты-потомки дезагрегированы от объекта-родителя
childQuantityList (список_Количества_Потомков)	List<QuantityElement> (Список< Элемент_Количество>)	(Необязательное) Неупорядоченный список из одного или нескольких элементов QuantityElements (Элемент_Количество), обозначающих (на уровне класса) «содержащиеся» объекты (см. 7.3.3.2). Событие типа AggregationEvent (Событие_ с_Агрегацией) ДОЛЖНО содержать либо непустое поле childEPCs (номера_EPC_Потомков), либо непустое поле childQuantityList (список_Количества_Потомков), или оба непустых поля, за исключением случая, когда оба поля, childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), могут быть пустыми, если поле action (действие) имеет значение DELETE (УДАЛЕНИЕ), показывая, что все объекты-потомки дезагрегированы от объекта-родителя
action (действие)	Action (Действие)	Как данное событие относится к жизненному циклу агрегации, упомянутой в данном событии. См. выше для более подробной информации
bizStep (бизнес-этап)	BusinessStepID (Идентификатор_Бизнес-Этапа)	(Необязательное) Бизнес-этап, частью которого было данное событие
disposition (состояние)	DispositionID (Идентификатор_Состояния)	(Необязательное) Условие ведения хозяйственной деятельности для объектов, связанных с номерами EPC, значение которого считается верным до тех пор, пока оно не будет противоречить последующему событию

Окончание таблицы 19

Поле	Тип поля	Описание
readPoint (место_Считывания)	ReadPointID (Идентификатор_Места_Считывания)	(Необязательное) Место считывания, в котором произошло событие
bizLocation (производственное_Место_Назначения)	BusinessLocationID (Идентификатор_Производственного_Места_Назначения)	(Необязательное) Производственное место назначения, где могут быть обнаружены объекты, связанные с «содержащими» и «содержащимися» номерами EPC, до тех пор, пока оно не будет противоречить последующему событию
bizTransactionList (список_Документов_Бизнес-транзакций)	Неупорядоченный список, состоящий из 0 или более экземпляров BusinessTransaction (Документ_Бизнес_Транзакции)	(Необязательное) Неупорядоченный список документов бизнес-транзакций, которые определяют контекст данного события
sourceList (список_Начальных_Пунктов)	List<Source> (Список<Начальных_Пунктов>)	(Необязательное) Неупорядоченный список элементов Source (7.3.5.4), который предоставляет контекст относительно исходной точки перемещения в процессе хозяйственной деятельности, частью которого является данное событие
destinationList (список_Конечных_Пунктов)	List<Destination> (Список< Конечных_Пунктов>)	(Необязательное) Неупорядоченный список элементов Destination (Конечный_Пункт) (см. 7.3.5.4), который предоставляет контекст относительно конечной точки перемещения в процессе хозяйственной деятельности, частью которого является данное событие

Следует обратить внимание, что в привязке XML (см. 9.3) поля childQuantityList (список_Количества_Потомков), sourceList (список_ Начальных_Пунктов) и destinationList (список_Конечных_Пунктов) появляются в стандартной области расширения, чтобы поддержать совместимость с EPCIS 1.0.

Ретроспективная семантика:

- событие, описанное с помощью поля bizStep (бизнес-этап) (и любых других полей), имело место с привлечением «содержащего» объекта, указанного в поле parentID (идентификатор_Родителя), и «содержащихся» объектов, указанных в поле childEPCs (номера_EPC_Потомков) и поле childQuantityList (список_Количества_Потомков), в момент времени, указанный в поле eventTime (время_События), и в месте нахождения, указанном в поле readPoint (место_Считывания).

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) объекты, определенные в поле childEPCs (номера_EPC_Потомков) и поле childQuantityList (список_Количества_Потомков), были агрегированы в содержащий их объект, указанный в поле parentID (идентификатор_Родителя);

- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), а значения полей childEPCs(номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков) не пустые) объекты, определенные в поле childEPCs (номера_EPC_Потомков) и поле childQuantityList (список_ Количества_Потомков), были дезагрегированы из объекта, указанного в поле parentID (идентификатор_Родителя);

- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), а оба поля, childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), пустые) все содержащиеся объекты были дезагрегированы из содержащего их объекта, указанного в поле parentID (идентификатор_Родителя);

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ) и задано непустое поле bizTransactionList (список_Документов_Бизнес-транзакций)) существует связь между документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), объектами, идентифицированными в поле childEPCs (номера_EPC_Потомков) и поле childQuantityList (список_Количества _Потомков), и «содержащим» их объектом, указанным в поле parentID (идентификатор_Родителя);

- (если значением поля action (действие) является OBSERVE (ОТСЛЕЖИВАНИЕ) и задано непустое поле bizTransactionList (список_Документов_Бизнес-транзакций)) данное событие произошло в контексте документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);

- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ) и задано непустое поле bizTransactionList (список_Документов_Бизнес-транзакций)) данное событие произошло в контексте документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);

- (если значение поля sourceList (список_Начальных_Пунктов) не пустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, начальная точка которого описывается с помощью начальных пунктов, перечисленных в поле sourceList (список_Начальных_Пунктов);

- (если значение поля destinationList (список_Конечных_Пунктов) не пустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, конечная точка которого описывается с помощью конечных пунктов, перечисленных в поле destinationList (список_Конечных_Пунктов).

Перспективная семантика:

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) существует агрегирование между «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя), и «содержащимися» объектами, указанными в полях childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков);

- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), а поля childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков) не пустые) агрегирование между «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя), и «содержащимися» объектами, указанными в полях childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), больше не существует;

- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), и оба поля, childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), пустые) агрегирование между «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя), и любыми «содержащимися» объектами больше не существует;

- (если задано поле disposition (состояние)) Условия ведения хозяйственной деятельности для объектов, связанных с объектами, указанными в полях parentID (идентификатор_Родителя), childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества _Потомков), заданы значением поля disposition (состояние);

- (если значение поля disposition (состояние) не задано) условия ведения хозяйственной деятельности для объектов, связанных с объектами, указанными в полях parentID (идентификатор_Родителя), childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), не меняются;

- (если задано поле bizLocation (производственное_Место_ Назначения)) объекты, связанные с объектами, указанными в полях parentID (идентификатор_Родителя), childEPCs (номера_EPC_Потомков) и childQuantityList, находятся в производственном месте назначения, указанном в поле bizLocation (производственное_Место_Назначения);

- (если значение поля bizLocation (производственное_Место_Назначения) не задано) производственное место назначения объектов, связанных с объектами, указанными в полях parentID (идентификатор_Родителя), childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), неизвестно;

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ) и задано непустое значение поля bizTransactionList (список_Документов _Бизнес-транзакций)) существует связь между документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), объектами, указанными в полях childEPCs (номера_EPC_Потомков) и childQuantityList (список_Количества_Потомков), и «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан).

Примечание — В случае, когда значением поля action (действие) является ADD (ДОБАВЛЕНИЕ) и задано не пустое значение поля bizTransactionList (список_Документов_Бизнес-транзакций), семантический эффект эквивалентен наличию события типа AggregationEvent (Событие_с_Агрегацией) без поля bizTransactionList (список_Документов_Бизнес-транзакций) вместе с событием типа TransactionEvent

(Событие_с_Документом_Транзакции), имеющим поле bizTransactionList (список_Документов_Бизнес-транзакций), и те же значения полей, что и событие типа AggregationEvent (Событие_с_Агрегацией). Следует обратить внимание, что событие типа AggregationEvent (Событие_с_Агрегацией) с непустым полем bizTransactionList (список_Документов_Бизнес-транзакций) не вызывает возврат события типа TransactionEvent (Событие_с_Документом_Транзакции) из запроса.

Примечание — Многие семантически несостоятельные ситуации могут представлять собой неправильное использование агрегации. Например, одним и тем же объектам может быть предоставлено несколько объектов-родителей в течение одного и того же периода времени с помощью различных операций ADD (ДОБАВЛЕНИЕ) без промежуточного удаления (Delete). Аналогично объект может быть указан как потомок своего прародителя (объекта-родителя от объекта-родителя) или даже себя самого. Несуществующая агрегация может иметь статус DELETED (УДАЛЕНА). Эти ситуации не могут быть обнаружены синтаксически, и в общем случае отдельный репозиторий EPCIS может не иметь достаточной информации, чтобы их обнаружить. Таким образом, настоящая спецификация не учитывает эти ошибочные обстоятельства.

7.4.4 Событие_с_Количеством_Объектов (QuantityEvent) (подкласс EPCISEvent) УСТАРЕВШИЙ

Тип событий QuantityEvent (Событие_с_Количеством_Объектов) фиксирует событие, которое имеет место в отношении определенного количества объектов на уровне класса. Данный тип события может быть использован, например, для отчета об уровнях запасов продукции.

Что касается *[24]*, версии EPCIS 1.1, подкласс QuantityEvent (Событие_с_Количеством_Объектов) в ней не используется. Вместо этого приложения должен использоваться подкласс ObjectEvent (Событие_с_Объектом), содержащий один или несколько элементов QuantityListElement (Элемент_Списка_Количество). Подкласс QuantityEvent (Событие_с_Количеством_Объектов) эквивалентен подклассу ObjectEvent (Событие_с_Объектом), содержащему пустой список EPCList и единственный элемент QuantityListElement (Элемент_Списка_Количество), содержащий поле quantity (количество), без поля uom (единица_измерения).

Таблица 20

Поле	Тип	Описание
eventTime (время_События) recordTime (время_Записи) eventTimeZoneOffset (смещение_Часового_Пояса_События)	(Наследуется от класса EPCISEvent (Событие_EPCIS); см. 7.4.1)	
epcClass (epc_на_уровне_Класса)	EPCClass (EPC_на_уровне_Класса)	Идентификатор, определяющий класс объекта, к которому относится данное событие
quantity (количество)	Int (Целочисленное_Значение)	Число объектов внутри класса, описанного этим событием
bizStep (бизнес-этап)	BusinessStepID (Идентификатор_Бизнес-Этапа)	(Необязательное) Бизнес-этап, частью которого было данное событие
disposition (состояние)	DispositionID (Идентификатор_Состояния)	(Необязательное) Условие ведения хозяйственной деятельности для объектов, связанных с номерами EPC, значение которых считается верным до тех пор, пока оно не будет противоречить последующему событию
readPoint (место_Считывания)	ReadPointID (Идентификатор_Места_Считывания)	(Необязательное) Место считывания, в котором произошло событие
bizLocation (производственное_Место_Назначения)	BusinessLocationID (Идентификатор_Производственного_Места_Назначения)	(Необязательное) Производственное место назначения, в котором могут быть обнаружены объекты, до тех пор, пока оно не будет противоречить последующему событию
bizTransactionList (список_Документов_Бизнес-транзакций)	Ненумерованный список, состоящий из 0 или более экземпляров вида лексики BusinessTransaction (Документ_Бизнес_Транзакции)	(Необязательное) Неупорядоченный список документов бизнес-транзакций, который определяет контекст данного события

Следует обратить внимание, что поскольку экземпляр на уровне класса EPC (EPCClass) всегда обозначает конкретную упаковочную единицу (например, коробку с 12 предметами), необходимость

в явном поле «unit of measure» («единица измерения») отсутствует. Единица измерения всегда является классом объектов, обозначаемым согласно epcClass (epc_на_уровне_Класса), как определено в мастер-данных для данного класса объектов.

Ретроспективная семантика:

- событие, описанное с помощью поля bizStep (бизнес-этап) (и любых других полей), имело место в отношении объектов, указанных в поле quantity (количество), EPC-класса, указанного в поле epcClass (epc_на_уровне _Класса), в момент времени, указанный в поле eventTime (время_События), в месте нахождения, указанном в поле readPoint (место_Считывания);
- (если заданы непустые значения поля bizTransactionList (список_Документов_Бизнес-транзакций)) данное событие произошло внутри контекста документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций).

Перспективная семантика:

- (если задано значение поля disposition (состояние)) условия ведения хозяйственной деятельности для объектов описаны с помощью поля disposition (состояние);
- (если значение поля disposition (состояние) не задано) условия ведения хозяйственной деятельности для объектов не изменяются,
- (если задано значение поля bizLocation (производственное_Место_Назначения)) объекты находятся в производственном месте назначения, указанном в поле bizLocation (производственное_Место_Назначения);
- (если значение поля bizLocation (производственное_Место _Назначения) не задано) место нахождения объектов неизвестно.

7.4.5 Событие_с_Документом_Транзакции (TransactionEvent) (подкласс EPCISEvent)

Тип события TransactionEvent (Событие_с_Документом_Транзакции) описывает объединение или разъединение физических или цифровых объектов с одним или несколькими документами бизнес-транзакций. В то время как в других типах событий есть необязательное поле bizTransactionList (список_ Документов_Бизнес-транзакций), которое может использоваться для определения контекста события, тип события TransactionEvent (Событие_ с_Документом_Транзакции) используется для однозначного обозначения того, что определенные объекты связаны с одним или несколькими документами бизнес-транзакций, являющимися частью события, или отделены от них.

Поле action события типа TransactionEvent (Событие_с_Документом_Транзакции) описывает отношение события к жизненному циклу документа транзакции (*см. таблицу 21*).

Таблица 21

Значение поля action (действие)	Смысл
ADD (ДОБАВЛЕНИЕ)	Объекты, идентифицированные во время события, были связаны с документом (документами) бизнес-транзакции(й) в течение данного события. Это относится к ситуациям, когда документы транзакции(й) создаются в первый раз, а также когда новые объекты добавляются к существующему(им) документу (документам) транзакции(й)
OBSERVE (ОТСЛЕЖИВАНИЕ)	Было подтверждено, что объекты, названные в событии, остаются связанными с документами бизнес-транзакций во время этого события. Примечание — Событие типа TransactionEvent (Событие_с_Документом_Транзакции) со значением действия OBSERVE (ОТСЛЕЖИВАНИЕ) очень похоже на событие типа ObjectEvent (Событие_с_Объектом), которое включает в себя непустое поле bizTransactionList (список_Документов_Бизнес-транзакций). Если группа конечных пользователей согласится использовать оба вида событий, эта группа должна будет четко определить, когда должен использоваться каждый вид. Примером, когда событие типа TransactionEvent (Событие_с_Документом _Транзакции) со значением действия OBSERVE (ОТСЛЕЖИВАНИЕ) может быть уместно, является международная отгрузка с идентификатором документа транзакции xxx, проходящая через порт, и при этом есть намерение записать номера EPC, которые были замечены в этой точке при обработке этого документа транзакции. Последующие запросы будут сконцентрированы на запросе идентификатора документа транзакции, чтобы найти номера EPC, а не на номерах EPC, чтобы найти идентификатор документа транзакции

Окончание таблицы 21

Значение поля action (действие)	Смысл
DELETE (УДАЛЕНИЕ)	Объекты, названные в событии, были отделены от бизнес-транзакции(й) во время этого события. Это относится к ситуациям, когда подмножество объектов отделяется от бизнес-транзакции(й), а также когда бизнес-транзакция(и) в целом завершена(ы). Как список номеров EPC, так и элементы QuantityElements (Элемент _Количество) для удобства могут быть исключены из события типа TransactionEvent (Событие_с_ Документом _Транзакции), означая то, что все объекты были отделены

Тип событий TransactionEvent (Событие_с_ Документом_Транзакции) имеет следующие поля (см. таблицу 22):

Таблица 22

Поле	Тип	Описание
eventTime (время_События) recordTime (время_Записи) eventTimeZoneOffset (смещение_Часового_Пояса_События)	(Наследуется от класса EPCISEvent (Событие_EPCIS); см. 7.4.1)	
bizTransactionList (список_Документов_Бизнес-транзакций)	Неупорядоченный список, состоящий из одного или более экземпляров BusinessTransaction (Документ_Бизнес_Транзакции)	Документы бизнес-транзакции(й)
parentID (идентификатор_Родителя)	URI	(Необязательное) Идентификатор объекта-родителя объектов, указанных в поле epcList (список_EPC) и поле quantityList (список_Количество). Если идентификатор объекта-родителя является номером EPC, это поле ДОЛЖНО содержать «чистый ключевой идентификатор» URI для номера EPC, как указано в [12, раздел 7]. См. также примечание, следующее за таблицей
epcList (список_EPC)	List<EPC> (Список<EPC>)	(Необязательное) Неупорядоченный список номеров EPC объектов, обозначенных с помощью идентификаторов на уровне экземпляра, ассоциируется с документами бизнес-транзакции (см. 7.3.3.2). Событие типа TransactionEvent (Событие_ с_ Документом_Транзакции) ДОЛЖНО содержать либо непустое поле epcList (список_EPC), либо непустое поле quantityList (список_Количество), или оба непустых поля, за исключением случая, когда оба поля, epcList (список_EPC) и quantityList (список_Количество), МОГУТ быть пустыми, если поле action (действие) имеет значение DELETE (УДАЛЕНИЕ), показывая, что все объекты отделены от документов бизнес-транзакции(-й).
quantityList (список_Количество)	List<QuantityElement> (Список<Элементы Количество>)	(Необязательное) Неупорядоченный список из одного или нескольких элементов QuantityElements (Элемент_Количество), обозначающих объекты (на уровне класса), к которым относится данное событие (см. 7.3.3.2). Событие типа TransactionEvent (Событие_ с_ Документом_Транзакции) ДОЛЖНО содержать либо непустое поле epcList (список_EPC), либо непустое поле quantityList

Окончание таблицы 22

Поле	Тип	Описание
		(список_ Количество), или оба непустых поля, за исключением случая, когда оба поля, epcList (список_EPC) и quantityList (список_Количество), могут быть пустыми, если поле action (действие) имеет значение DELETE (УДАЛЕНИЕ), показывая, что все объекты отделены от документов бизнес-транзакции(й)
action (действие)	Action (Действие)	Как данное событие относится к жизненному циклу документа бизнес-транзакции, упомянутому в данном событии. См. выше для более подробной информации
bizStep (бизнес-этап)	BusinessStepID (Идентификатор_ Бизнес-Этапа)	(Необязательное) Бизнес-этап, частью которого было данное событие
disposition (состояние)	DispositionID (Идентификатор_ Состояния)	(Необязательное) Условие ведения хозяйственной деятельности для объектов, связанных с объектами, значение которого считается верным, пока оно не будет противоречить последующему событию
readPoint (место_ Считывания)	ReadPointID (Идентификатор_Места_ Считывания)	(Необязательное) Место считывания, в котором произошло событие
bizLocation (производственное_Место_ Назначения)	BusinessLocationID (Идентификатор_ Производственного_ Места_Назначения)	(Необязательное) Производственное место назначения, где могут быть обнаружены объекты, связанные с «содержащими» и «содержащимися» объектами, до тех пор, пока оно не будет противоречить последующему событию
sourceList (список_ Начальных_Пунктов)	List<Source> (Список<Начальных_ Пунктов>)	(Необязательное) Неупорядоченный список элементов Source (Начальный_ Пункт) (см. 7.3.5.4), который предоставляет контекст об исходной точке перемещения в процессе хозяйственной деятельности а, частью которого является данное событие
destinationList (список_Конечных_ Пунктов)	List<Destination> (Список<Конечных_ Пунктов>)	(Необязательное) Неупорядоченный список элементов Destination (Конечный_Пункт) (см. 7.3.5.4), который предоставляет контекст о конечной точке перемещения в процессе хозяйственной деятельности, частью которого является данное событие

Следует обратить внимание, что в привязке XML (см. 9.3) поля quantityList (список_Количество), sourceList (список_Начальных_Пунктов) и destinationList (список_Конечных_Пунктов) появляются в стандартной области расширения, чтобы поддержать совместимость с [3] (EPCIS 1.0).

Примечания

1 Использование имени поля parentID (идентификатор _Родителя) в обоих событиях, типа TransactionEvent (Событие_с_ Документом_Транзакции) и типа AggregationEvent (Событие_с_ Агрегацией) (см. 7.2.10), не показывает схожесть функции или семантики. В общем случае тип события TransactionEvent (Событие_с_ Документом_Транзакции) несет ту же информацию об идентификации объекта, что и тип события ObjectEvent (Событие_с_Объектом), то есть список номеров EPC и/или элементов QuantityElements (Элемент_Количество). Все другие информационные поля (bizTransactionList (список_Документов_Бизнес-транзакций), bizStep (бизнес-этап), bizLocation (производственное_ Место_Назначения) и т. д.) применяются одинаково и единообразно ко всем указанным объектам, независимо от того, указаны ли объекты только в поле epcList (список_EPC) и поле quantityList (список_Количество), или указано дополнительное поле parentID (идентификатор_Родителя).

2 Тип события TransactionEvent (Событие_с_Документом_Транзакции) предоставляет способ описания объединения или разукрупнения бизнес-транзакций с объектами. Поле parentID (идентификатор _Родителя) в типе события TransactionEvent (Событие_с_Документом_Транзакции) выделяет конкретный номер EPC или другой идентификатор как предпочтительный или первичный объект, но не подразумевает физическую связь какого-либо рода, а также не является каким-либо видом вложенности или наследования, подразумеваемым самим типом события TransactionEvent (Событие_с_Документом_Транзакции). Только экземпляры типа события AggregationEvent (Событие_с_Агрегацией) описывают фактические отношения родитель —

потомок и вложенные отношения родитель — потомок. Это можно увидеть, сравнив семантику типа сообщения AggregationEvent (Событие_с_Агрегацией) в 7.2.10 с семантикой типа события TransactionEvent (Событие_с_Документом_Транзакции) ниже.

Ретроспективная семантика:

- событие, описанное с помощью поля bizStep (бизнес-этап) (и любых других полей), имело место с привлечением документов бизнес-транзакций, указанных в поле bizTransactionList (список_Документов_Бизнес-транзакций), объектов, приведенных в поле epcList (список_EPC) и в поле quantityList (список_Количество), и объекта, содержащего экземпляры, указанного в поле parentID (идентификатор_Родителя) (если задан), в момент времени, указанный в поле eventTime (время_События), и в месте нахождения, указанном в поле readPoint (место_Считывания);
- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) объекты, определенные в поле epcList (список_EPC) и поле quantityList (список_Количество), и «содержащий» объект, указанный в поле parentID (идентификатор_Родителя) (если задан), были связаны с документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций);
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), а поля epcList (список_EPC) и quantityList (список_Количество) не пустые) объекты, определенные в полях epcList (список_EPC), quantityList (список_Количество) и «содержащем» объекте, указанном в поле parentID (идентификатор_Родителя) (если задан), были отделены от документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), оба поля, epcList (список_EPC) и quantityList (список_Количество), пустые, а поле parentID (идентификатор_Родителя) не задано) все объекты были отделены от документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);
- (если значение sourceList (список_Начальных_Пунктов) непустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, исходная точка которого описывается с помощью начальных пунктов, перечисленных в поле sourceList (список_Начальных_Пунктов);
- (если значение поля destinationList (список_Конечных_Пунктов) непустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, конечная точка которого описывается с помощью конечных пунктов, перечисленных в поле destinationList (список_Конечных_Пунктов).

Перспективная семантика:

- (если значением поля action (действие) является ADD (ДОБАВЛЕНИЕ)) существует связь между документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), объектами, указанными в поле epcList (список_EPC) и в поле quantityList (список_Количество), и «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан);
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ), а поля epcList (список_EPC) и quantityList (список_Количество) не пустые) связи между бизнес-транзакциями, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), объектами, указанными в поле epcList (список_EPC) и в поле quantityList (список_Количество), и «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан), больше не существует;
- (если значением поля action (действие) является DELETE (УДАЛЕНИЕ) и оба поля, epcList (список_EPC) и quantityList (список_Количество), пустые и поле parentID (идентификатор_Родителя) не задано) связи между документами бизнес-транзакций, перечисленными в поле bizTransactionList (список_Документов_Бизнес-транзакций), и любыми объектами больше не существует;
- (если задано поле disposition (состояние)) условия ведения хозяйственной деятельности для объектов, связанных с объектами, указанными в поле epcList (список_EPC) и поле quantityList (список_Количество), а также «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан), заданы значением поля disposition (состояние);
- (если значение поля disposition (состояние) не задано) условия ведения хозяйственной деятельности для объектов, связанных с объектами, указанными в поле epcList (список_EPC) и в поле quantityList (список_Количество), и «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан), не меняются;

- (если задано поле bizLocation (производственное_Место _Назначения)) объекты, связанные с объектами, указанными в полях epcList (список_EPC), quantityList (список_Количество), и «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан), находятся в производственном месте назначения, указанном в поле bizLocation (производственное_Место_Назначения);

- (если значение поля bizLocation (производственное_Место_ Назначения) не задано) производственное место назначения объектов, связанных с объектами, указанными в полях epcList (список_EPC), quantityList (список_Количество), и «содержащим» объектом, указанным в поле parentID (идентификатор_Родителя) (если задан), неизвестно.

7.4.6 Событие_с_Преобразованием (TransformationEvent) (подкласс EPCISEvent)

Тип события TransformationEvent (Событие_с_Преобразованием) собирает информацию о событии, в котором один или несколько физических или цифровых объектов, обозначенных идентификаторами на уровне экземпляра (EPC) или класса (EPC Class), полностью или частично потребляются в качестве входных данных, а также один или несколько объектов, обозначенных идентификаторами на уровне экземпляра (EPC) или на уровне класса (EPC Class), производятся как выходные данные. Тип события TransformationEvent (Событие_с_Преобразованием) фиксирует взаимосвязь между входными данными и выходными данными, при которой любые входные данные могут каким-либо образом включаться в выходные данные.

Некоторые бизнес-процессы с преобразованием происходят в течение длительного периода времени, и поэтому более целесообразно представлять их в виде серии событий EPCIS. Для связи между собой двух и более событий типа TransformationEvent (Событие_с_Преобразованием) в них может включаться значение TransformationID (Идентификатор_Преобразования). Когда события имеют одинаковое значение TransformationID (Идентификатор_ Преобразования), то это означает, что входные данные для *любого* из этих событий могут каким-то образом включаться в выходные данные для *любого* из этих событий.

Поля приведены в *таблице 23*.

Таблица 23

Поле	Тип	Описание
eventTime (время_События) recordTime (время_Записи) eventTimeZoneOffset (смещение_ Часового_ Пояса_События)	(Наследуется от класса EPCISEvent (Событие_EPCIS); см. 7.4.1)	
inputEPCList (список_ EPC_на_Входе)	List<EPC> (Список<EPC>)	(Необязательное) Неупорядоченный список одного или нескольких номеров EPC, обозначающих (на уровне экземпляра) объекты, которые являлись входными данными для преобразования (см. 7.3.3.2). См. ниже ограничения на случаи, когда поле inputEPCList (список_ EPC_на _Входе) может быть не задано
inputQuantityList (список_Количество_на_Входе)	List<QuantityElement> (Список<Элемент_ Количество>)	(Необязательное) Неупорядоченный список из одного или нескольких элементов QuantityElements (Элемент_ Количество), обозначающих объекты (на уровне класса), которые являлись входной информацией для преобразования. См. ниже ограничения на случаи, когда поле inputQuantityList (список_ Количество_на_Входе) может быть не задано
outputEPCList (список_ EPC_на_Выходе)	List<EPC> (Список<EPC>)	(Необязательное) Неупорядоченный список из одного или нескольких номеров EPC, обозначающих объекты (на уровне экземпляра), которые являлись выходной информацией после преобразования (см. 7.3.3.2). См. ниже ограничения на случаи, когда поле outputEPCList (список_EPC_на _Выходе) может быть не задано

Окончание таблицы 23

Поле	Тип	Описание
outputQuantityList (список_Количество_на_Выходе)	List<QuantityElement> (Список<Элемент_Количество>)	(Необязательное) Неупорядоченный список из одного или нескольких элементов QuantityElements, обозначающих объекты (на уровне класса), которые являлись выходной информацией после преобразования. См. ниже ограничения на случаи, когда поле outputQuantityList (список_ Количество_на_ Выходе) может быть не задано
transformationID (идентификатор_Преобразования)	TransformationID (Идентификатор_Преобразования)	(Необязательное) Уникальный идентификатор, который связывает данное событие с другими событиями типа TransformationEvent (Событие_с_Преобразованием), имеющими идентичное значение поля transformationID (идентификатор _Преобразования). Если оно задано, все входные данные для всех событий, имеющих одинаковое значение поля transformationID (идентификатор_ Преобразования), могут влиять на все выходные данные всех событий, разделяющих это значение поля transformationID (идентификатор_Преобразования). Если значение поля transformationID (идентификатор_Преобразования) не задано, входные данные этого события могут влиять на выходные данные этого события, но входные и выходные данные других событий не связаны с этим событием
bizStep (бизнес-этап)	BusinessStepID (Идентификатор_Бизнес-Этапа)	(Необязательное) Бизнес-этап, частью которого было данное событие
disposition (состояние)	DispositionID (Идентификатор_Состояния)	(Необязательное) Условия ведения хозяйственной деятельности для объектов, связанных с выходными объектами, значение которых считается верным, пока оно не будет противоречить последующему событию
readPoint (место_Считывания)	ReadPointID (Идентификатор_Места_Считывания)	(Необязательное) Место считывания, в котором произошло событие
bizLocation (производственное_Место_Назначения)	BusinessLocationID (Идентификатор_Производственного_Места_Назначения)	(Необязательное) Производственное место назначения, где могут быть обнаружены выходные объекты данного события, до тех пор, пока оно не будет противоречить последующему событию
bizTransactionList (список_Документов_Бизнес-транзакций)	Неупорядоченный список от 0 или более экземпляров BusinessTransaction (Документ_Бизнес_Транзакции)	(Необязательное) Неупорядоченный список документов бизнес-транзакций, которые определяют контекст этого события
sourceList (список_Начальных_Пунктов)	List<Source> (Список<Начальных_Пунктов>)	(Необязательное) Неупорядоченный список элементов Source (Начальный_Пункт) (см. 7.3.5.4), который предоставляет контекст об исходной точке перемещения в процессе хозяйственной деятельности, частью которого является данное событие
destinationList (список_Конечных_Пунктов)	List<Destination> (Список< Конечных_Пунктов>)	(Необязательное) Неупорядоченный список элементов Destination (Конечный _Пункт) (см. 7.3.5.4), который предоставляет контекст о конечной точке перемещения в процессе хозяйственной деятельности, частью которого является данное событие
ilmd (мдэс)	ILMD (МДЭС)	(Необязательное) Мастер-данные экземпляра/партии (см. 7.3.6), которые описывают выходные объекты, созданные во время этого события

Если значение поля transformationID (идентификатор_Преобразования) не задано, то событие типа TransformationEvent (Событие_с _Преобразованием) ДОЛЖНО включать по крайней мере одно поле входных данных (т. е. по меньшей мере одно значение поля inputEPCList (список_ EPC_на_Входе) и поля inputQuantityList (список_Количество_на_Входе) должно быть непустым) и по крайней мере одно поле выходных данных (т. е. по меньшей мере одно значение поля outputEPCList (список_EPC_на_Выходе) и поля outputQuantityList (список_Количество_на_Выходе) должно быть непустым). Если значение поля transformationID (идентификатор_ Преобразования) присутствует, то событие типа TransformationEvent (Событие_с_Преобразованием) ДОЛЖНО включать по крайней мере одно поле входных данных ИЛИ по крайней мере одно поле выходных данных (или оба поля). Последнее обеспечивает возможность того, что при преобразовании, описываемом несколькими событиями, связанными с общим полем transformationID (идентификатор_Преобразования), любое из этих событий может только добавлять входные данные или только извлекать выходные данные.

Ретроспективная семантика:

- преобразование, описанное с помощью поля bizStep (бизнес-этап) (и любых других полей), имело место с привлечением входных объектов, указанных в поле inputEPCList (список_EPC_на_Входе) и поле inputQuantityList (список_Количество_на_Входе), и выходных объектов, указанных в поле outputEPCList (список_EPC_на_Выходе) и поле outputQuantityList (список_Количество_на_Выходе), в момент времени, указанный в поле eventTime (время_События), и в месте нахождения, указанном в поле readPoint (место_Считывания);
- данное событие произошло в контексте документов бизнес-транзакций, перечисленных в поле bizTransactionList (список_Документов_Бизнес-транзакций);
- (если значение поля transformationID (идентификатор_ Преобразования) не задано) любой из входных объектов, указанных в поле inputEPCList (список_EPC_на_Входе) и поле inputQuantityList (список_Количество_на_Входе) данного события, может включаться в каждый из выходных объектов, указанных в поле outputEPCList (список_EPC_на_Выходе) и поле outputQuantityList (список_Количество_на_Выходе) этого события;
- (если значение поля transformationID (идентификатор_ Преобразования) задано) любой из входных объектов, указанных в поле inputEPCList (список_EPC_на_Входе) и поле inputQuantityList (список_Количество_на_Входе) данного события, совместно с входными объектами, указанными в поле inputEPCList (список_EPC_на_Входе) и поле inputQuantityList (список_Количество_на_Входе) других событий, имеющих то же самое значение поля transformationID (идентификатор_ Преобразования), может включаться в каждый из выходных объектов, указанных в поле outputEPCList (список_EPC_на_Выходе) и поле outputQuantityList (список_Количество_на_Выходе) этого события, так же как в каждый из выходных объектов, указанных в поле outputEPCList (список_EPC_на_Выходе) и поле outputQuantityList (список_ Количество_на_Выходе) других событий, имеющих то же самое значение поля transformationID (идентификатор_Преобразования),
- (если значение поля sourceList (список_Начальных_Пунктов) не пустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, начальная точка которого описывается с помощью источников, перечисленных в поле sourceList (список_Начальных_Пунктов);
- (если значение поля destinationList (список_Конечных_Пунктов) не пустое) данное событие произошло в контексте перемещения в процессе хозяйственной деятельности, конечная точка которого описывается с помощью конечных пунктов, перечисленных в поле destinationList (список_Конечных_Пунктов).

Перспективная семантика:

- объекты, обозначенные идентификаторами на уровне экземпляра в поле outputEPCList (список_EPC_на_Выходе), могут появиться в последующих событиях;
- объекты, обозначенные идентификаторами на уровне класса в поле outputQuantityList (список_Количество_на_Выходе), могут появиться в последующих событиях;
- (если задано поле disposition (состояние)) условие ведения хозяйственной деятельности для объектов, определенных полями outputEPCList (список_EPC_на_Выходе) и outputQuantityList (список_Количество_на_ Выходе), задано значением поля disposition (состояние);

- (если значение поля disposition (состояние) не задано) условие ведения хозяйственной деятельности для объектов, определенных полями outputEPCList (список_EPC_на_Выходе) и outputQuantityList (список_Количество_на_Выходе), неизвестно;

- (если задано поле bizLocation) объекты, определенные полями outputEPCList (список_EPC_на_Выходе) и outputQuantityList (список_Количество_на_Выходе), находятся в месте нахождения, указанном в поле bizLocation;

- (если значение поля bizLocation не задано) производственное место назначения объектов, указанных в поле outputEPCList (список_EPC_на_Выходе) и поле outputQuantityList (список_Количество_на_Выходе), неизвестно;

- (если значение поля ilmd (мдэс) непустое) объекты, указанные в поле outputEPCList (список_EPC_на_Выходе) и поле outputQuantityList (список_Количество_на_Выходе), описываются с помощью атрибутов в поле ilmd (мдэс).

8 Сервисный уровень

Данный раздел включает нормативные спецификации модулей на сервисном уровне. Вместе взятые, эти модули определяют три интерфейса: интерфейс EPCIS сбора данных, интерфейс EPCIS управления запросами, интерфейс EPCIS обратных вызовов по запросам. (Два последних интерфейса совместно относятся к интерфейсам EPCIS запросов.) На схеме (*рисунок 8*) показаны взаимосвязи между этими интерфейсами, расширяя схему из раздела 2 (эта схема является справочной):

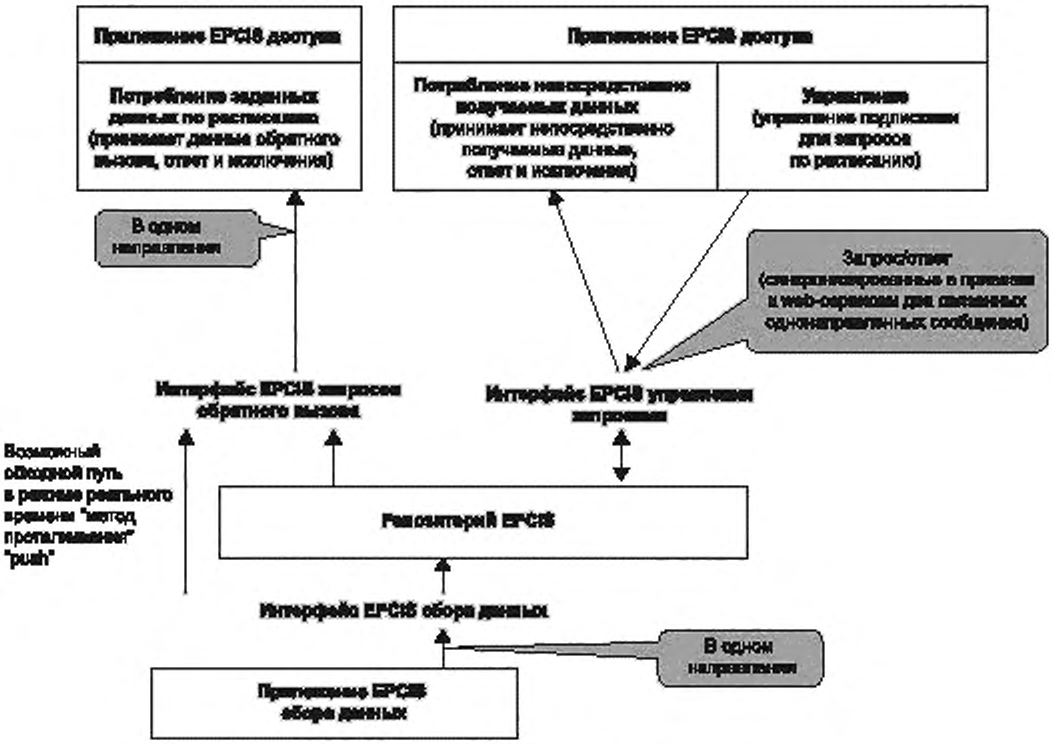


Рисунок 8

В приведенных ниже подразделах сервисы определяются с помощью нотации схемы классов UML. Схемы классов UML, используемые для этой цели, могут содержать интерфейсы, имеющие операции, а не поля или связи (*см. рисунок 9*).

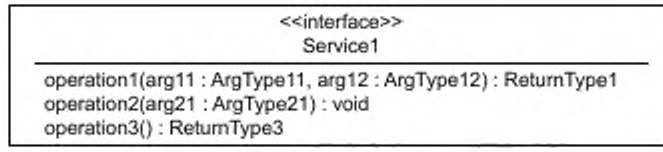


Рисунок 9

Эта схема показывает определение сервиса для интерфейса `Service1`, который предусматривает три операции. Операция `Operation1` принимает два аргумента `arg11` и `arg12`, имеющих типы `ArgType11` и `ArgType12` соответственно, и возвращает значение типа `ReturnType1`. Операция `Operation2` принимает один аргумент, но не возвращает результат. Операция `Operation3` не принимает никаких аргументов, но возвращает значение типа `ReturnType3`.

В описаниях UML обозначение `<<extension point>>` идентифицирует место, где реализации ДОЛЖНЫ предусматривать расширяемость посредством добавления новых операций. Механизмы расширяемости ДОЛЖНЫ предоставлять поставщикам продукции как частные варианты расширения, совместимые с EPCIS, так и расширения, определяемые GS1 в будущих версиях данной спецификации или в новых спецификациях.

В случае стандартных привязок языка WSDL точки расширения реализуются за счет добавления дополнительных операций.

8.1 Модуль основных операций сбора данных

Модуль основных операций сбора данных (Core Capture Operations Module) предоставляет операции, с помощью которых основные события могут доставляться от приложения EPCIS сбора данных. В данном подразделе слово «клиент» (client) относится к приложению EPCIS сбора данных, а «сервис EPCIS» (EPCIS Service) относится к системе, которая реализует интерфейс EPCIS сбора данных.

8.1.1 Аутентификация и авторизация

Некоторые привязки интерфейса EPCIS сбора данных предоставляют сервису EPCIS средства для аутентификации подлинности клиента, а клиенту — средства для аутентификации подлинности сервиса EPCIS либо и то и другое. Спецификация средств аутентификации включена в спецификацию каждой привязки. Если сервис EPCIS аутентифицирует подлинность клиента, реализация МОЖЕТ использовать ключевой идентификатор клиента, чтобы принять решение об авторизации, как описано ниже. Кроме того, реализация МОЖЕТ сохранить ключевой идентификатор клиента с собранными данными для использования в последующих решениях по авторизации путем внедрения интерфейсов EPCIS запросов, как описано в 8.2.2.

Благодаря простоте интерфейса EPCIS сбора данных механизмы авторизации очень просты. Это означает, что реализация МОЖЕТ использовать аутентифицированный ключевой идентификатор клиента, чтобы решить, разрешена операция сбора данных или нет.

Примечание — Предполагается, что торговые партнеры всегда будут использовать привязки, которые обеспечивают аутентификацию подлинности клиента или множественную аутентификацию при использовании интерфейсов EPCIS для обмена данными между организациями. Предполагается, что привязки, которые не обеспечивают аутентификацию, будут использоваться только в рамках одной организации в ситуациях, когда для удовлетворения внутренних требований безопасности аутентификация не требуется.

8.1.2 Сервис сбора данных

На рисунке 10 приведена схема, представляющая сервис сбора данных.

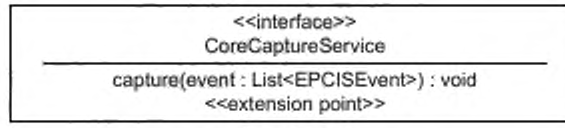


Рисунок 10

Интерфейс сбора данных содержит только один метод, `capture (сбор_данных)`, который принимает единственный аргумент и не возвращает результат. Реализации интерфейса EPCIS сбора данных ДОЛЖНЫ принимать каждый элемент перечня аргументов, который является действительным ти-

пом EPCISEvent (Событие_EPCIS) или его подтипом, в соответствии с настоящей спецификацией. Реализации МОГУТ принимать другие типы событий через расширение, предоставленное поставщиком. Простота данного интерфейса допускает широкое разнообразие привязок, включая простые привязки типа очереди сообщений.

Примечание — «Привязки типа очереди сообщений» («Message-queue type bindings») означают следующее. Предприятия обычно используют технологию «шины сообщений» («message bus») для соединения разных компонентов распределенной системы. Шина сообщений обеспечивает надежный канал для последовательной доставки сообщений по заказу от отправителя к получателю. (Отношение между отправителем и получателем может быть точка-точка (сообщение типа «очередь» («queue»)) или «один-ко-многим» через механизм публикации/подписки (сообщение типа «тема» («topic»)). «Привязка типа очередь сообщений» («message-queue type binding») интерфейса EPCIS сбора данных просто будет обозначением особого канала шины сообщений для доставки событий EPCIS из приложения EPCIS сбора данных в репозиторий EPCIS или в приложение EPCIS доступа посредством интерфейса EPCIS обратных вызовов по запросам. Каждое сообщение будет иметь полезную нагрузку, содержащую одно или несколько событий EPCIS (сериализованных посредством некоторой привязки на уровне определения данных, например привязки XML). Поэтому в такой привязке каждая передача/доставка сообщения соответствует одной операции «сбора данных» («capture»).

Операция capture (сбор_данных) записывает один или несколько событий EPCIS любого типа. Аргументы (*см. таблицу 24*):

Таблица 24

Аргумент	Тип	Описание
event (событие)	List of EPCISEvent (Перечень Событий _EPCIS)	Событие(я) для сбора данных. Вся необходимая информация, такая как время события, номер EPC и т. п., содержится внутри каждого события. Исключение: значение поля recordTime (время_Записи) МОЖЕТ быть не задано. Вне зависимости от наличия значения recordTime (время_Записи) во входных данных, после операции сбора данных значение поля recordTime (время_Записи) события, записанного репозиторием EPCIS или приложением EPCIS доступа, является временем события. Следует отметить, что необязательное поле eventID (идентификатор_События) обрабатывается отличным способом от поля recordTime (время_ Записи); как и любое поле в EPCIS, поле eventID (идентификатор_События) должно считываться интерфейсом сбора данных без модификаций. Разъяснение (справочное): эта обработка поля recordTime (время_Записи) необходима для того, чтобы длительные запросы обрабатывались правильно (см. 8.2.5.2)

Возвращаемое значение:

(отсутствует).

Конкретные привязки интерфейса EPCIS сбора данных в разделе 10, используя структуру документа EPCIS, определенную в 9.5, несут список событий EPCIS, подлежащих считыванию. Документ EPCIS может содержать мастер-данные в заголовке. Применение интерфейса EPCIS сбора данных, соответствующее настоящему стандарту, МОЖЕТ выбрать запись этих мастер-данных или проигнорировать их: стандарт EPCIS не регламентирует обращение с мастер-данными, полученными через интерфейс EPCIS сбора данных. Аналогичным образом в системе, реализующей интерфейс EPCIS сбора данных, могут быть предусмотрены средства по сбору мастер-данных путем получения документа мастер-данных EPCIS (см. 9.7), но в стандарте EPCIS не содержится требований по поддержке такого решения.

С другой стороны, все мастер-данные на уровне экземпляра/серии, содержащиеся в разделе МДЭП (ILMD) события, ДОЛЖНЫ быть считаны как составная часть события, аналогично любому элементу данных в рамках события EPCIS. Такие мастер-данные ДОЛЖНЫ подлежать запросу с использованием параметров запроса SimpleEventQuery (Простой_Запрос_События), указанных в 8.2.7.1, но не существует требования к применению обеспечить доступность мастер-данных в разделе МДЭП (ILMD) события по запросу SimpleMasterDataQuery (Простой_Запрос_Мастер_Данных), указанному в 8.2.7.2.

Реализация интерфейса сбора данных ДОЛЖНА либо фиксировать все события, указанные в конкретной операции сбора данных, или быть неспособной фиксировать ни одно из событий в указанной операции. То есть реализация НЕ ДОЛЖНА предоставлять возможность частичного фиксирования, при котором некоторые события в списке фиксируются, а другие нет.

Причины неудачи операции сбора данных зависят от конкретной реализации. Ниже перечислены некоторые из возможных причин неудачи:

- вводные данные к операции сбора данных сформированы неудачно или не соответствуют требованиям синтаксиса используемых конкретных привязок, включая валидацию схемы для конкретных привязок, использующих XML-схемы, определенные в разделе 9;
- клиент не имеет права проводить операцию считывания;
- превышение установленных для данной реализации ограничений по количеству событий в одной операции считывания, общему количеству хранимых событий, частоте считывания и т. п.;
- нарушение установленных для данной реализации правил в отношении содержимого событий, либо самих по себе, либо в привязке к ранее считанным событиям. Следует отметить, что такие правила могут быть оправданы в закрытой системе, где использование EPCIS регламентируется конкретным стандартом по применению, но они могут быть непригодны для открытой системы, предназначенной для обработки любых данных EPCIS. Если заданные параметры правил слишком узки, они могут ограничивать возможности функциональной совместимости;
- временные технические проблемы, например временная недоступность сервера или сети.

8.2 Модуль основных операций с запросами

Модуль основных операций с запросами (Core Query Operations Module) предоставляет два интерфейса, называемые интерфейс EPCIS управления запросами (EPCIS Query Control Interface) и интерфейс EPCIS обратных вызовов по запросам (EPCIS Query Callback Interface), с помощью которых данные EPCIS могут быть извлечены приложением EPCIS доступа (EPCIS Accessing Application). Интерфейс EPCIS управления запросами предоставляет средства для приложений EPCIS доступа и торговых партнеров для получения данных EPCIS после их сбора из любого источника, как правило, посредством взаимодействия с репозиторием EPCIS. Это предоставляет средство для приложения EPCIS доступа средства для извлечения данных по требованию, а также вводит подписки на длительные запросы (standing queries). Результаты длительных запросов доставляются приложениям EPCIS доступа через интерфейс EPCIS обратных вызовов по запросам.

8.2.1 Аутентификация

Некоторые привязки интерфейса EPCIS управления запросами обеспечивают сервисам EPCIS средства для аутентификации подлинности клиента, а клиенту — средства для аутентификации подлинности сервисов EPCIS либо и то, и другое. Спецификация средств аутентификации включена в спецификацию каждой привязки. Если сервис EPCIS аутентифицирует подлинность клиента, реализация МОЖЕТ использовать ключевой идентификатор клиента, чтобы принять решение об авторизации, как описано в *8.2.2*.

Примечание — Предполагается, что торговые партнеры всегда будут использовать привязки, которые обеспечивают аутентификацию подлинности клиента или множественную аутентификацию при использовании интерфейсов EPCIS для обмена данными между организациями. Привязки, не предлагающие проверку подлинности, предполагается использовать только в рамках одной организации в ситуациях, когда проверка подлинности не требуется для удовлетворения внутренних требований безопасности.

8.2.2 Авторизация

Сервис EPCIS, в зависимости от ключевого идентификатора запрашивающего клиента, может предоставлять доступ только к подмножеству информации. Такая ситуация обычно возникает в сценариях между предприятиями, когда запрашивающий клиент принадлежит иной организации, нежели оператор службы EPCIS, но также может возникать в сценариях внутри предприятия.

Получив запрос EPCIS, сервис EPCIS при обработке запроса на основе аутентифицированной подлинности клиента МОЖЕТ предпринять любое из следующих действий:

- сервис МОЖЕТ отказаться от выполнения запроса в целом, ответив с помощью `SecurityException (Исключение_по_Безопасности)`, как определено ниже;
- сервис МОЖЕТ отреагировать меньшим количеством данных, чем было у него запрошено. Например, если клиент запрашивает все экземпляры событий типа `ObjectEvent (Событие_с_Объектом)` в рамках определенного интервала времени и сервису известно о 100 совпадающих событиях, сервис может выбрать ответ менее чем из 100 событий (например, возвратить только те события, чьи номера EPC представляют собой номер SGTIN с префиксом предприятия, который назначен клиенту);
- сервис МОЖЕТ ответить более укрупненной информацией. В частности, если ответ на запрос включает тип места нахождения (как указано в 7.3.4), сервис может подставить вместо простого места нахождения агрегированное место нахождения;

- сервис МОЖЕТ скрыть информацию. Например, если клиент запрашивает экземпляры событий типа ObjectEvent (Событие_с_Объектом), сервис может в ответ выбрать удаление полей bizTransactionList (список_Документов _Бизнес-транзакций). Возвращенная информация ДОЛЖНА быть правильно сформированными событиями EPCIS, соответствующими данной спецификации и отраслевым руководствам. Кроме того, если сокрытие информации влечет за собой неоднозначную или вводящую в заблуждение информацию, то все событие ДОЛЖНО быть скрыто. Это правило применяется независимо от того, была ли исходная информация собрана через интерфейс EPCIS сбора данных или предоставлена каким-либо другим способом. Например, в событии типа AggregationEvent (Событие_с_Агрегацией) с действием (action), значение которого равно ADD (ДОБАВЛЕНИЕ), попытка скрыть поле parentID (идентификатор_ Родителя) привела бы к событию, сформированному неправильно, потому что присутствие поля parentID (идентификатор_Родителя) необходимо, когда значением действия (action) является ADD (ДОБАВЛЕНИЕ). Следовательно, в этом случае все событие целиком должно быть скрыто;

- сервис МОЖЕТ ограничить область запроса данными, которые первоначально были собраны с помощью определенного ключевого идентификатора клиента. Это позволяет «разделить» единственный сервис EPCIS для использования группами несвязанных пользователей, чьи данные следует хранить отдельно.

Реализация EPCIS обладает свободой в определении того, какое из этих действий нужно выполнить при обработке любого запроса, используя любые выбранные ею средства. Спецификация правил авторизации находится вне рамок рассмотрения данной спецификации.

Примечание — Поскольку спецификация EPCIS касается интерфейсов запросов, а не какой-либо конкретной реализации, спецификация EPCIS не обозначает позицию относительно того, как принимаются решения по авторизации. Частные реализации EPCIS могут иметь произвольно сложные правила авторизации в процессе хозяйственной деятельности. При этом спецификация EPCIS может содержать типовые данные, которые необходимы для авторизации, независимо от того, предназначены они исключительно для этой цели или нет.

8.2.3 Запросы для большого объема данных

Многие из описанных ниже операций запроса позволяют клиенту делать запрос на потенциально неограниченный объем данных. Например, ответ на запрос, который запрашивает все экземпляры класса ObjectEvent (Событие_с_Объектом) в течение заданного интервала времени, может возвратить одно, тысячу, миллион или миллиард событий в зависимости от временного интервала и количества зафиксированных событий. Это может представлять проблемы с производительностью для реализаций сервиса.

Для решения этой проблемы сервис EPCIS МОЖЕТ отклонить любой запрос, ответив с помощью исключения (exception) QueryTooLarge (Запрос_с_Превышением_Объема). Это исключение указывает, что объем запрашиваемых данных больше, чем сервис в состоянии предоставить клиенту. Исключение QueryTooLarge (Запрос_с_Превышением_Объема) — это подсказка клиенту о том, что клиент может получить успешный результат, если сузит область исходного запроса или представит запрос в другое время (например, если служба принимает или отклоняет запросы на основе текущей расчетной нагрузки на сервис).

Примечание — Предполагается, что будущие версии данной спецификации будут предоставлять более сложные способы решения проблемы запросов с превышением, таких как поддержка многостраничного режима (paging), курсоринг (cursoring) и т. п. Из соображений целесообразности в этой версии не было согласовано ничего более сложного.

8.2.4 Слишком сложные запросы

Реализации сервиса EPCIS могут ограничивать виды обрабатываемых запросов, чтобы избежать обработки запросов, которые будут потреблять больше ресурсов, чем сервис в силах предоставить. Например, запрос, по которому производится поиск событий, имеющих конкретное значение в определенном поле события, может потребовать больше или меньше ресурсов для обработки в зависимости от того, предусматривала ли реализация поиск по этому полю (например, в зависимости от того, индексируется или нет колонка базы данных, соответствующая этому полю). Как и в случае запросов для слишком большого объема данных (см. 8.2.3), это может представлять проблемы с производительностью для реализаций сервиса.

Для решения этой проблемы сервис EPCIS МОЖЕТ отклонить любой запрос, ответив с помощью исключения QueryTooComplex (Запрос_с_Чрезмерной_ Сложностью). Это исключение указывает, что структура запроса такова, что сервис не в состоянии выполнить его для клиента.

В отличие от исключения QueryTooLarge (Запрос_с_Превышением_Объема) (см. 8.2.3) исключение QueryTooComplex (Запрос_с_Чрезмерной_Сложностью) указывает, что простое сужение области запроса (например, при запросе событий за одну неделю вместо одного месяца) вряд ли приведет к успеху при ответе на запрос.

Особый язык запросов может определять условия, при которых сервису EPCIS не разрешается отклонять запрос с исключением QueryTooComplex (Запрос_с_Чрезмерной_Сложностью). Это обеспечивает минимальный уровень совместимости.

8.2.5 Структура запроса (интерфейс EPCIS управления запросами)

Интерфейс EPCIS управления запросами предоставляет общую структуру, с помощью которой клиентские приложения могут запрашивать данные EPCIS. Интерфейс обеспечивает как запросы по требованию, в которых определенное требование клиента вызывает запрос на исполнение и ответ, возвращающий результаты, так и длительные запросы, в которых клиент проявляет постоянный интерес к запросу и после этого получает периодическую доставку результатов через интерфейс EPCIS обратных вызовов по запросам без дополнительных запросов. Эти два режима неофициально называются «pull» («доставка по запросу») и «push» («принудительная доставка») соответственно.

Интерфейс EPCIS управления запросами определен ниже. Реализация интерфейса EPCIS управления запросами ДОЛЖНА применять все методы, приведенные ниже:

```
<<interface>>
EPCISQueryControlInterface
---
subscribe(queryName : String, params : QueryParams, dest : URI,
controls : SubscriptionControls, subscriptionID : String)
unsubscribe(subscriptionID : String)
poll(queryName : String, params : QueryParams) : QueryResults
getQueryNames() : List // of names
getSubscriptionIDs(queryName : String) : List // of Strings
getStandardVersion() : string
getVendorVersion() : string
<<extension point>>
```

Длительные запросы формируются с помощью создания одного или нескольких подписок на ранее определенный, с использованием метода subscribe (подписка), запрос. Результаты будут периодически доставляться через интерфейс обратных вызовов по запросам к конкретному получателю до тех пор, пока подписка не будет отменена с использованием метода unsubscribe (отмена_подписки). Запросы по требованию формируются с помощью выполнения ранее определенного, с использованием метода poll (опрос), запроса. Каждый вызов метода poll (опрос) возвращает результат непосредственно инициатору вызова. В любом случае, если запрос параметризован, конкретные значения для параметров могут быть предоставлены в качестве аргументов для методов subscribe (подписка) или poll (опрос).

Реализация МОЖЕТ предоставить один или несколько «предопределенных» запросов. Предопределенный запрос доступен для использования методами subscribe (подписка) или poll (опрос) и возвращается в списке имен запросов с помощью метода getQueryNames (получение_Имен_Запросов) без предварительного выполнения клиентом каких-либо действий по определению запроса. В частности, EPCIS версии 1.0 не поддерживает механизм, с помощью которого клиент может определить новый запрос, и, таким образом, предопределенные запросы являются единственно доступными. Информацию в отношении конкретных предопределенных запросов, которые ДОЛЖНЫ быть предоставлены реализацией интерфейса запросов для *[3]* (EPCIS версии 1.0), см. также в 8.2.7.

Реализация МОЖЕТ разрешить использовать данный запрос с помощью метода poll (опрос), но не с помощью метода subscribe (подписка). В общем случае запросы для данных о событиях могут использоваться как с методом poll (опрос), так и с методом subscribe (подписка), однако запросы для мастер-данных могут использоваться только с методом poll (опрос). Это происходит из-за того, что метод subscribe (подписка) устанавливает периодическое расписание для многократного запуска запроса, каждый раз ограничивая внимание новыми событиями, записанными с момента последнего запуска запроса. Этот механизм не может применяться к запросам для мастер-данных, так как считается, что мастер-данные являются квазистатическими и не имеют ничего, что соответствует времени записи.

Спецификация этих методов приведена в *таблице 25*.

Таблица 25

Метод	Описание
subscribe (подписка)	Регистрирует подписчика для ранее определенного запроса, имеющего конкретное имя. Аргумент params (параметры) предоставляет значения, которые будут использоваться для любых именованных параметров, определенных запросом. Параметр dest (получатель) определяет получателя, которому будут доставлены результаты запроса через интерфейс обратных вызовов по запросам. Параметр dest (получатель) является идентификатором URI, который и определяет конкретную привязку интерфейса обратных вызовов по запросам для использования, и определяет адресную информацию. Параметр controls (управление_обработкой) управляет тем, как будет обрабатываться подписка. В частности, он определяет условия, при которых будет вызываться запрос (например, определение периодического расписания). subscriptionID (удостоверение_подписки) — это произвольная строка, которая копируется в каждый ответ, доставленный указанному получателю, и не интерпретируется сервисом EPCIS иным образом. Клиент может использовать параметр subscriptionID (удостоверение_подписки), чтобы определить, из какой подписки был сгенерирован данный результат, особенно когда для одного и того же получателя созданы несколько подписок. Аргумент dest (получатель) МОЖЕТ быть нулевым или пустым, и в этом случае результаты доставляются предварительно назначенному получателю на основании аутентифицированной подлинности инициатора запроса. Если реализация EPCIS не имеет получателя, предварительно назначенного для инициатора запроса, или не разрешает его использование, она ДОЛЖНА ответить с помощью исключения InvalidURIException (Исключение_при_Недействительном_URI)
unsubscribe (отмена_подписки)	Удаляет ранее зарегистрированную подписку, имеющую определенный параметр subscriptionID (удостоверение_подписки)
poll (опрос)	Вызывает предварительно определенный запрос с указанным именем и возвращает результаты. Аргумент params (параметры) предоставляет значения, которые будут использоваться для любых именованных параметров, определенных данных запросом
getQueryNames (получение_Имен_Запросов)	Возвращает список всех имен запросов, доступных для использования с методами subscribe (подписка) и poll (опрос). Он включает все предопределенные запросы, предоставляемые реализацией, включая и те, что указаны в 8.2.7
getSubscriptionIDs (получение_Удостоверений_Подписки)	Возвращает список всех значений subscriptionID (удостоверение_подписки), подписанных в настоящий момент на указанный именованный запрос
getStandardVersion (получение_Версии_Стандарта)	Возвращает строку с версией спецификации, которой соответствует данная реализация. Возможные значения для этой строки определены GS1. Реализация ДОЛЖНА возвратить строку с версией данной спецификации, которой реализация полностью отвечает, либо она ДОЛЖНА возвратить строку, соответствующую последней версии, которой отвечает данная реализация. Чтобы показать соответствие настоящему стандарту и [2], реализация ДОЛЖНА возвратить строку 1.2
getVendorVersion (получение_Версии_Поставщика)	Возвращает строку, которая определяет, какие расширения поставщика предусматривает данная реализация. Возможные значения этой строки и их смысл определяются поставщиком, за исключением того, что пустая строка ДОЛЖНА показывать, что реализация предоставляет только обычную функциональность без расширений поставщика. Если реализация возвращает непустую строку, возвращаемое значение ДОЛЖНО быть в формате URI, где владеющей организацией является поставщик. Например, это может быть URL адрес в протоколе HTTP, полномочная часть которого является доменным именем, принадлежащим поставщику, URN адрес с идентификатором из пространства имен URN, выданный поставщику организацией IANA, адрес в формате OID URN, начальным путем которого является номер частного предприятия (Private Enterprise Number), присвоенный поставщику, и т. д.

Данная структура применяется вне зависимости от содержания запроса. Подробное содержание запроса, а также результаты, возвращаемые методом poll (опрос) или доставляемые подписчику (абоненту) через интерфейс обратных вызовов по запросам, определены в последующих разделах

данного документа. Эта структура предназначена для облегчения расширяемости, поскольку новые типы запросов могут быть указаны и вписаны в эту общую структуру.

Реализация МОЖЕТ ограничить функционирование любого метода в соответствии с решениями по авторизации на основании аутентифицированной подлинности клиента, делающего запрос. Например, реализация может ограничить множество идентификаторов, возвращаемое методом getSubscriptionIDs (получение_Удостоверений_Подписки) и распознаваемое методом unsubscribe (отмена_подписки), только теми абонентами, которые были подписаны ранее одним и тем же клиентом на получение результатов. Это позволяет «разделить» единый сервис EPCIS для использования группами несвязанных пользователей, данные которых должны храниться отдельно.

Если предопределенный запрос определяет именованные параметры, значения этих параметров могут предоставляться, когда запрос впоследствии обращается к методам poll (опрос) или subscribe (подписка). Экземпляр QueryParams представляет собой просто множество пар имя/значение, где имена соответствуют именам параметров, определенных запросом, а значения являются конкретными значениями, которые будут использоваться для вызова запроса (метод poll (опрос)) или подписки на запрос (метод subscribe (подписка)). Если экземпляр QueryParams включает пару имя/значение, в которой значение — пусто, он ДОЛЖЕН интерпретироваться так, как будто этот параметр запроса полностью опущен.

Метод poll (опрос) или метод subscribe (подписка) ДОЛЖНЫ ответить с помощью исключения QueryParameterException (Исключение_по_ Параметру_Запроса) в любом из следующих обстоятельств:

- параметр, требуемый указанным запросом, не был задан или был предоставлен с пустым значением;
- был предоставлен параметр, имя которого не соответствует ни одному из имен параметров, определенных указанным запросом;
- были предоставлены два параметра с одним и тем же именем;
- любое другое ограничение, налагаемое указанным запросом, нарушается. Такие ограничения могут включать ограничения на диапазон значений, разрешенных для данного параметра, требования, чтобы два или более параметров были взаимоисключающими или должны быть предоставлены вместе, и так далее. Конкретные ограничения, налагаемые указанным запросом, определены в документации для этого запроса.

8.2.5.1 Элементы управления подпиской

Подписка на длительные запросы производится через метод subscribe (подписка). Для каждой подписки экземпляр SubscriptionControls (Управление_Подпиской) определяет, как будет обрабатываться запрос.

```
SubscriptionControls
---
schedule : QuerySchedule // см. 8.2.5.3
trigger : URI // определяет запускающее событие, известное сервису
initialRecordTime : Time // см. 8.2.5.2
reportIfEmpty : Boolean
<<extension point>>
```

Поля экземпляра SubscriptionControls (Управление_Подпиской) определены в *таблице 26*.

Таблица 26

Аргумент	Тип	Описание
schedule (расписание)	QuerySchedule (Расписание_ Запросов)	(Необязательное) Определяет периодическое расписание, по которому будет выполняться запрос (см. 8.2.5.3). Должен быть задан один аргумент: schedule (расписание) или trigger (запуск). Если заданы оба аргумента или оба не заданы, реализация ДОЛЖНА ответить с помощью исключения SubscriptionControlsException (Исключение_Управления_Подпиской)

Окончание таблицы 26

Аргумент	Тип	Описание
trigger (запуск)	URI	(Необязательное) Определяет запускающее событие, известное сервису EPCIS, которое будет служить для запуска выполнения этого запроса. Идентификаторы URI запускающего события available (доступно) независимы от сервиса. Должен быть задан один аргумент: schedule (расписание) или trigger (запуск). Если заданы оба аргумента или оба не заданы, реализация ДОЛЖНА ответить с помощью исключения SubscriptionControlsException (Исключение_Управления_Подпиской)
initialRecordTime (время_Первой_Записи)	Time	(Необязательное) Определяет время, используемое для ограничения того, какие события учитываются при обработке запроса, когда он выполняется в первый раз (см. 8.2.5.2). Если аргумент не задан, по умолчанию устанавливается время, в которое создается подписка
reportIfEmpty (пустой_ли_Отчет)	boolean	Если значение аргумента — истина (true), то при выполнении запроса экземпляр QueryResults (Результаты_Запроса) всегда отсылается подписчику. Если — ложь (false), экземпляр QueryResults (Результаты_Запроса) отсылается подписчику, только когда результаты не пустые

8.2.5.2 Автоматическое ограничение на основе времени записи события

Каждая подписка на запрос приводит к выполнению запроса несколько раз подряд, причем время каждого выполнения контролируется заданным аргументом schedule (расписание) или запускается условием запуска, заданным аргументом trigger (запуск). Наличие нескольких выполнений одного запроса является разумным только в том случае, если каждое выполнение ограничено объемом данных новых событий, сгенерированных с момента последнего выполнения. В противном случае одни и те же события будут возвращаться более одного раза. Однако временные ограничения не могут быть явно указаны в запросе или параметрах запроса, поскольку они не меняются от одного выполнения к другому.

По этой причине сервис EPCIS ДОЛЖЕН ограничить объем выполнения каждого запроса для запроса по подписке следующим образом. В первый раз, когда запрос выполняется для данной подписки, рассматриваются только те события, значение поля recordTime (время_Записи) которых больше или равно значению параметра initialRecordTime (время_Первой_Записи), указанному при создании подписки. При каждом выполнении запроса, следующего за первым, рассматриваются только те события, значение поля recordTime (время_Записи) которых больше или равно значению времени выполнения последнего запроса. Реализация зависит от того, насколько отказ от доставки результатов запроса подписчику влияет на этот расчет. Реализациям СЛЕДУЕТ предпринимать все усилия для обеспечения надежной доставки результатов запроса, чтобы подписчик не пропустил каких-либо данных. Запрос или параметры запроса могут определять дополнительные ограничения во время записи. Они применяются после ограничения совокупности событий, как описано выше.

Примечание — Одно возможное выполнение этого требования состоит в том, что сервис EPCIS поддерживает значение minRecordTime (минимальное_Время_Записи) для каждой существующей подписки. Значение minRecordTime (минимальное_Время_Записи) для данной подписки изначально устанавливается в initialRecordTime (время_Первой_Записи) и обновляется до значения текущего времени каждый раз, когда выполняется запрос для этой подписки. Каждый раз, когда выполняется запрос, рассматриваются только те события, значение поля recordTime (время_Записи) которых больше или равно значению minRecordTime (минимальное_Время_Записи) для этой подписки.

8.2.5.3 Расписание запросов (Query Schedule)

Тип QuerySchedule (Расписание_Запросов) служит для определения расписания периодического выполнения запроса для конкретной подписки. Каждое поле типа QuerySchedule (Расписание_Запросов) — это строка, задающая шаблон для сопоставления с некоторым интервалом текущего времени. Запрос будет выполняться каждый раз, когда текущие дата и время совпадают с данными в поле QuerySchedule (Расписание_Запросов).

Каждое поле типа QuerySchedule (Расписание_Запросов) — это строка, значение которой должно соответствовать следующим грамматическим правилам:

```
QueryScheduleField ::= Element ( "," Element )*
Element ::= Number   Range
Range ::= "[" Number "-" Number "]"
Number ::= Digit+
Digit ::= "0" | "1" | "2" | "3" | "4"
   | "5" | "6" | "7" | "8" | "9"
```

Каждое значение Number (Число), которое является частью значения поля расписания запросов (QuerySchedule), должно попадать в разрешенный для этого поля диапазон, как указано в таблице 27. Реализация EPCIS ДОЛЖНА ответить с помощью исключения SubscriptionControlsException (Исключение_Управления_Подпиской), если любое значение поля расписания запросов (QuerySchedule) не соответствует грамматическим правилам, приведенным выше, или содержит значение Number (Число), которое выпадает из разрешенного диапазона, или включает значение диапазона Range (Диапазон), в котором значение первого параметра Number (Число) больше, чем значение второго параметра Number (Число).

Тип QuerySchedule (Расписание_Запросов) определяет периодическую последовательность временных значений («время запроса»). Время запроса — это любое временно́е значение, которое соответствует типу QuerySchedule (Расписание_Запросов), удовлетворяя при этом следующему правилу:

- получив значение времени, извлекают секунду, минуту, час (от 0 до 23 включительно), день месяца (от 1 до 31 включительно) и день недели (от 1 до 7 включительно, обозначая дни недели с понедельника по воскресенье). Данный расчет будет выполняться в соответствии с часовым поясом, выбранным сервисом EPCIS;

- временно́е значение соответствует типу QuerySchedule (Расписание_Запросов), если каждое из извлеченных значений подходит (как указано ниже) соответствующему полю типа QuerySchedule (Расписание_Запросов) для всех заданных полей QuerySchedule (Расписание_Запросов);

- значение, извлеченное из временно́го значения, совпадает с полем типа QuerySchedule (Расписание_Запросов), если оно соответствует любому из разделенных запятыми элементов Element (Элемент) поля типа QuerySchedule (Расписание_Запросов);

- значение, извлеченное из временно́го значения, совпадает с элементом Element (Элемент) поля QuerySchedule (Расписание_Запросов), если

- Element (Элемент) является числом Number (Число) и значение, извлеченное из временно́го значения, равно значению Number (Число);

или

- Element (Элемент) является Range (Диапазон) и значение, извлеченное из временно́го значения, больше или равно первому значению Number (Число) в диапазоне значений Range (Диапазон) и меньше или равно второму значению Number (Число) в диапазоне значений Range (Диапазон).

См. примеры после таблицы 27.

Реализация EPCIS ДОЛЖНА интерпретировать тип QuerySchedule (Расписание_Запросов) как заявление клиента о том, когда он хотел бы выполнить запрос, и ДОЛЖНА предпринять разумные усилия, чтобы придерживаться этого графика. Однако реализация EPCIS МОЖЕТ отклоняться от запрошенного расписания, согласуя свои действия с собственными правилами относительно загрузки сервера, авторизации или по любой другой причине. Если во время вызова метода subscribe (подписка) реализация EPCIS знает, что она не сможет соблюдать указанное в поле QuerySchedule (Расписание_Запросов) расписание без существенного отклонения от запроса, реализация EPCIS ДОЛЖНА ответить с помощью исключения SubscriptionControlsException (Исключение_Управления_Подпиской).

Примечание — Расписание в поле QuerySchedule (Расписание_Запросов), принятое буквально, указывает точное время выполнения запроса с точностью до секунды. На практике реализация может не пожелать или быть неспособной точно выполнить запрос, но может выполнить общее намерение. Например, расписание в поле QuerySchedule (Расписание_Запросов) может задать выполнение запроса в момент точного значения каждого часа, в то время как реализация может выбрать выполнение запроса каждый час плюс-минус пять минут от момента точного значения часа. Вышеприведенный параграф предназначен для предоставления свободы реализации для такого вида отклонений.

В любом случае автоматическая обработка поля recordTime (время_Записи), как было указано ранее, ДОЛЖНА быть основана на фактическом времени выполнения запроса, независимо от того, насколько точно оно соответствует типу QuerySchedule (Расписание_Запросов).

Значения поля экземпляра QuerySchedule (Расписание_Запросов) выглядят следующим образом *(см. таблицу 27)*.

Таблица 27

Аргумент	Тип	Описание
second (секунда)	String (Строка)	(Необязательное) Указывает, что время запроса должно иметь соответствующее значение секунд. Диапазон значений этого параметра — в пределах от 0 до 59 включительно
minute (минута)	String (Строка)	(Необязательное) Указывает, что время запроса должно иметь соответствующее значение минут. Диапазон значений этого параметра — в пределах от 0 до 59 включительно
hour (час)	String (Строка)	(Необязательное) Указывает, что время запроса должно иметь соответствующее значение часов. Диапазон значений этого параметра — в пределах от 0 до 23 включительно, причем 0 означает, что час начинается в полночь, и 23 означает, что час заканчивается в полночь
dayOfMonth (день_месяца)	String (Строка)	(Необязательное) Указывает, что время запроса должно иметь соответствующее значение дня месяца. (Значение этого параметра, равное 29, 30 и 31, будет правильным только для тех месяцев, которые имеют как минимум такое количество дней)
month (месяц)	String (Строка)	(Необязательное) Указывает, что время запроса должно иметь соответствующее значение месяца. Диапазон значений этого параметра — в пределах от 1 до 12 включительно
dayOfWeek (день_недели)	String (Строка)	(Необязательное) Указывает, что время запроса должно иметь соответствующее значение дня недели. Диапазон значений этого параметра — в пределах от 1 до 7 включительно, причем 1 означает понедельник, 2 означает вторник, и так далее до 7, означающего воскресенье. Примечание — Данная схема нумерации совместима с [15]

8.2.5.3.1 Примеры расписания запросов (справочная информация)

Ниже приведено несколько примеров типа QuerySchedule (Расписание_Запросов) и объяснение того, что они означают.

Пример 1
QuerySchedule
second = "0"
minute = "0"
все другие поля не заданы

Это означает «запускать запрос один раз в час, в начале часа». Если значение аргумента reportIfEmpty (пустой_ли_Отчет) для метода subscribe (подписка) — false (ложно), то отсутствует необходимость запускать отчет для отсылки каждый час: отчет будет отправлен в течение часа после появления новых данных о событиях, соответствующих запросу.

Пример 2
QuerySchedule
second = "0"
minute = "30"
hour = "2"
все другие поля не заданы

Это означает «запускать запрос один раз в день, в 2:30 ночи.»

Пример 3
QuerySchedule
second = "0"
minute = "0"
dayOfWeek = "[1-5]"

Это означает «запускать запрос один раз в час в начале каждого часа, но только в рабочие дни».

Пример 4
QuerySchedule
hour = "2"
другие поля не заданы

Это означает «запускать запрос один раз в секунду в диапазоне между 2:00:00 и 2:59:59 ежедневно». Этот пример показывает, что обычно не желательно пропускать поле с более тонкой детализацией, чем указанные поля.

8.2.5.4 QueryResults (Результаты запроса)

Экземпляр `QueryResults (Результаты_Запроса)` синхронно возвращается методом `poll (опрос)` интерфейса EPCIS управления запросами, а также синхронно направляется подписчику длительного запроса через интерфейс EPCIS обратных вызовов по запросам.

```
QueryResults
---
queryName : string
subscriptionID : string
resultsBody : QueryResultsBody
<<extension point>>
```

Поля экземпляра QueryResults (Результаты_Запроса) определены в *таблице 28*.

Таблица 28

Поле	Тип	Описание
`queryName (имя_Запроса)`	`string (строка)`	Поле ДОЛЖНО содержать имя запроса (аргумент `queryName (имя_Запроса)`, который был указан в вызове метода `poll (опрос)` или метода `subscribe (подписка)`)
`subscriptionID (удостоверение_Подписки)`	`string (строка)`	(Условное) Если экземпляр `QueryResults (Результаты_Запроса)` доставляется подписчику как результат длительного запроса, поле `subscriptionID (удостоверение_Подписки)` ДОЛЖНО содержать ту же строку, что и аргумент `subscriptionID (удостоверение_подписки)` при вызове метода `subscribe (подписка)`. Если экземпляр `QueryResults (Результаты_Запроса)` возвращается как результат метода `poll (опрос)`, это поле ДОЛЖНО быть опущено
`resultsBody (тело_Результата)`	`QueryResultsBody (Тело_Результата_Запроса)`	Информация, возвращаемая как результат запроса. Точный тип этого поля зависит от того, какой запрос выполняется. Каждый из предопределенных запросов в 8.2.7 определяет соответствующий тип этого поля

8.2.6 Условия возникновения ошибок

Методы прикладного программного интерфейса (API — application programming interface) интерфейса EPCIS управления запросами сигнализируют клиенту об ошибках с помощью исключений. Определены следующие исключения. Все типы исключений, приведенные в таблице *29*, являются общим базовым типом `EPCISException (Исключение_EPCIS)`, содержащим один обязательный элемент строки, указывающий причину исключения.

Таблица 29

Имя исключения	Значение
`SecurityException (Исключение_по_Безопасности)`	Операция не была разрешена из-за нарушения контроля доступа или других проблем безопасности. Сюда относится случай, когда сервис хочет отклонить авторизацию для выполнения конкретной операции на основе аутентифицированной подлинности клиента. Конкретные обстоятельства, которые могут вызывать это исключение, зависят от реализации и выходят за рамки данной спецификации
`DuplicateNameException (Исключение_при_Дублировании_Имени)`	(Не реализовано в настоящем стандарте и [2]). Указанное имя запроса уже существует

Окончание таблицы 29

Имя исключения	Значение
QueryValidationException (Исключение_по_Валидации_Запроса)	(Не реализовано в настоящем стандарте и [2]). Указанный запрос недействителен; например, он содержит синтаксическую ошибку
QueryParameterException (Исключение_по_Параметру_Запроса)	Один или несколько параметров запроса неверны, включая любую из следующих ситуаций: - имя параметра не является распознаваемым параметром для указанного запроса; - значение параметра имеет неверный тип или выходит за пределы диапазона; - два или более параметра запроса имеют одно и то же имя
QueryTooLargeException (Исключение_при_Превышении_Объёма_Запроса)	Попытка выполнить запрос привела к большему объему данных, чем сервис готов предоставить
QueryTooComplexException (Исключение_при_Чрезмерной_Сложности_Запроса)	Указанные параметры запроса, в другом случае вполне допустимые, предполагали выполнение запроса, который оказался более сложным, чем сервис готов был выполнить
InvalidURIException (Исключение_при_Недействительном_URI)	Идентификатор URI, указанный для подписчика, не проходит грамматический анализ, не именует схему, распознанную реализацией, или нарушает правила, наложенные определенной схемой
SubscriptionControlsException (Исключение_Управления_Подпиской)	Указанные элементы управления подпиской были недействительны. Например, параметры расписания были вне диапазона, идентификатор триггера в формате идентификатора URI не прошел грамматический анализ или не именует распознанный триггер и т. п.
NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса)	Указанное имя запроса не существует
NoSuchSubscriptionException (Исключение_при_Несуществующем_Удостоверении_Подписки)	Указанный параметр subscriptionID (удостоверение_подписки) не существует
DuplicateSubscriptionException (Исключение_при_Дублировании_Подписки)	Указанный параметр subscriptionID (удостоверение_подписки) идентичен предыдущей подписке, которая была создана и все еще не отписана
SubscribeNotPermittedException (Исключение_при_Недопустимой_Подписке)	Указанное имя запроса не может быть использовано с методом subscribe (подписка), только с методом poll (опрос)
ValidationException (Исключение_по_Валидации)	Входные данные для операции были синтаксически неверны, в соответствии с синтаксисом, определенным привязкой. Каждая привязка определяет особые обстоятельства, при которых используется данное исключение
ImplementationException (Исключение_по_Реализации)	Общее исключение, созданное реализацией по причинам, специфичным для данной реализации. Это исключение содержит один дополнительный элемент: элемент severity (критичность), значениями которого являются либо ERROR (ОШИБКА), либо SEVERE (НЕОПРЕДЕЛЕННОСТЬ). ERROR показывает, что реализация EPCIS остается в том же состоянии, которое оно имело до попытки выполнения операции. SEVERE (НЕОПРЕДЕЛЕННОСТЬ) показывает, что реализация EPCIS остается в неопределенном состоянии

Исключения, которые могут быть выбраны каждым методом интерфейса EPCIS управления запросами, указаны в таблице *30*:

Таблица 30

Метод EPCIS	Исключения
getQueryNames (получение_Имен_Запросов)	SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)

Окончание таблицы 30

Метод EPCIS	Исключения
subscribe (подписка)	NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса) InvalidURIException (Исключение_при_Недействительном_URI) DuplicateSubscriptionException (Исключение_при_Дублировании_Подписки) QueryParameterException (Исключение_по_Параметру_Запроса) QueryTooComplexException (Исключение_при_Чрезмерной_Сложности_Запроса) SubscriptionControlsException (Исключение_Управления_Подпиской) SubscribeNotPermittedException (Исключение_при_Недопустимой_Подписке) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
unsubscribe (отмена_подписки)	NoSuchSubscriptionException (Исключение_при_Несуществующем_Удостоверении_Подписки) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
poll (опрос)	NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса) QueryParameterException (Исключение_по_Параметру_Запроса QueryTooComplexException (Исключение_при_Чрезмерной_Сложности_Запроса) QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
getSubscriptionIDs (получение_Удостоверений_Подписки)	NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
getStandardVersion (получение_Версии_Стандарта)	SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
getVendorVersion (получение_Версии_Поставщика)	SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)

В дополнение к исключениям, полученным из методов интерфейса EPCIS управления запросами, перечисленным выше, попытка выполнения длительного запроса может привести к тому, что вместо обычного результата запроса подписчику через интерфейс EPCIS обратных вызовов по запросам отправляются исключения QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) или ImplementationException (Исключение_по_Реализации). В этом случае исключения QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) или ImplementationException (Исключение_по_Реализации) ДОЛЖНЫ включать, помимо строки с причиной, имя запроса и параметр subscriptionID (удостоверение_подписки), как указано в вызове метода subscribe (подписка), который создал длительный запрос.

8.2.7 Предопределенные запросы для EPCIS

В стандарте EPCIS не предусмотрен язык запросов, с помощью которого клиент может выразить произвольный запрос данных. Вместо этого реализация EPCIS ДОЛЖНА предоставлять следующие предопределенные запросы, которые клиент может вызывать, используя методы poll (опрос) и subscribe (подписка) в интерфейсе EPCIS управления запросами. Каждый вызов метода poll (опрос) и subscribe (подписка) может включать параметры через аргумент params (параметры). Предопределенные запросы, определенные в этом подразделе, имеют большое количество необязательных параметров. При соответствующем выборе параметров клиент может получить различные результаты.

Параметры для каждого предопределенного запроса, а также то, какие результаты он возвращает, определены в этом подразделе. Реализация EPCIS может использовать любое внутреннее представление данных, которое она пожелает, и реализовывать эти предопределенные запросы с использованием

любой базы данных или технологии запросов, которые она выбирает, если результаты, предоставленные клиенту, соответствуют данной спецификации.

8.2.7.1 Запрос SimpleEventQuery (Простой_Запрос_События)

Этот запрос вызывается путем указания строки SimpleEventQuery (Простой_Запрос_События) в качестве аргумента queryName (имя_Запроса) для методов poll (опрос) или subscribe (подписка). Результатом является экземпляр QueryResults (Результаты_Запроса), основа которого содержит (возможно, пустой) список экземпляров EPCISEvent (Событие_EPCIS). При отсутствии ограничений параметром eventType (тип_События) каждый элемент списка результатов может иметь любой тип события, т. е. ObjectEvent (Событие_с_Объектом), AggregationEvent (Событие_с_Агрегацией), QuantityEvent (Событие_с_Количеством_Объектов), TransactionEvent (Событие_с_Документом_Транзакции) или любой тип события расширения, которое является подклассом EPCISEvent (Событие_EPCIS).

Запрос SimpleEventQuery (Простой_Запрос_События) ДОЛЖЕН быть доступен как через метод poll (опрос), так и через метод subscribe (подписка), то есть реализация НЕ ДОЛЖНА использовать исключение SubscribeNotPermittedException (Исключение_при_Недопустимой_Подписке), если запрос SimpleEventQuery (Простой_Запрос_События) указывается как аргумент queryName (имя_Запроса) для метода subscribe (подписка).

Запрос SimpleEventQuery (Простой_Запрос_События) определен для того, чтобы возвращать набор событий, которые удовлетворяют критериям, указанным в параметрах запроса (как указано ниже в *таблице 31*). При возврате событий, зафиксированных через интерфейс EPCIS сбора данных, каждое событие, выбранное для возврата, должно быть идентично первоначально зафиксированному событию, с учетом положений по авторизации (см. 8.2.2), включения поля recordTime (время_Записи) и любых необходимых преобразований в абстрактное внутреннее представление и из него. Тем не менее для любого поля события, определенного для хранения неупорядоченного списка, реализация EPCIS НЕ ДОЛЖНА сохранять порядок.

Параметры для этого запроса приведены *в таблице 31*. Ни один из этих параметров не является обязательным (хотя в большинстве случаев запрос будет содержать хотя бы один параметр).

Таблица 31

Имя параметра	Тип значения параметра	Смысл
eventType (тип_События)	List of String (Список значений строки)	Если параметр указан, результат будет содержать только те события, тип которых соответствует одному из типов, указанных в значении параметра. Каждый элемент значения параметра может быть одной из следующих строк: ObjectEvent (Событие_с_Объектом), AggregationEvent (Событие_с_Агрегацией), QuantityEvent (Событие_с_Количеством_Объектов), TransactionEvent (Событие_с_Документом_Транзакции) или TransformationEvent (Событие_с_Преобразованием). Элемент значения параметра может быть также именем типа события расширения. Если параметр не задан, то для включения в результат будут рассмотрены все типы событий
GE_eventTime (время_Последующего_События)	Time (Время)	Если параметр указан, то в результат будут включены только те события, у которых значение поля eventTime (время_События) больше или равно указанному значению. Если параметр не задан, то события включаются независимо от значения их поля eventTime (время_События) (если не ограничено параметром LT_eventTime (время_Предыдущего_События))
LT_eventTime (время_Предыдущего_События)	Time (Время)	Если параметр указан, то в результат будут включены только те события, у которых значение поля eventTime (время_События) меньше, чем указанное значение. Если параметр не задан, то события включаются независимо от значения их поля eventTime (время_События) (если не ограничено параметром GE_eventTime (время_Последующего_События))

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
GE_recordTime (время_ Последующей_Записи)	Time (Время)	Если параметр предусмотрен, то будут возвращены только те события, у которых значение поля recordTime (время_Записи) больше или равно указанному значению. Автоматическое ограничение на основе времени записи события (см. 8.2.5.2) может неявно обеспечивать ограничение, подобное этому параметру. Если параметр не задан, то события включаются независимо от значения их поля recordTime (время_ Записи), кроме автоматического ограничения, основанного на времени записи события (см. 8.2.5.2)
LT_recordTime (время_ Предыдущей_Записи)	Время	Если параметр предусмотрен, то будут возвращены только те события, у которых значение поля recordTime (время_Записи) меньше, чем указанное значение. Если параметр не задан, то события включаются независимо от значения их поля recordTime (время_Записи) (если не ограничено параметром GE_recordTime (время_Последующей_Записи) или автоматическим ограничением, основанным на времени записи события)
EQ_action (соответствие_по_Действию)	List of String (Список значений строки)	Если параметр указан, то результат будет содержать только те события, которые (a) имеют поле action (действие) и где (b) значение поля action (действие) совпадает с одним из указанных значений. Элементы значения этого параметра должны быть одной из строк: ADD (ДОБАВЛЕНИЕ), OBSERVE (ОТСЛЕЖИВАНИЕ) или DELETE (УДАЛЕНИЕ); если нет, то реализация ДОЛЖНА ответить с помощью исключения QueryParameterException (Исключение_по_Параметру_ Запроса). Если параметр не задан, то события включаются независимо от значения их поля action (действие)
EQ_bizStep (соответствие_по_Бизнес-этапу)	List of String (Список значений строки)	Если параметр указан, то результат будет включать только те события, которые (a) имеют ненулевое поле bizStep (бизнес-этап) и (b) где значение поля bizStep (бизнес-этап) совпадает с одним из указанных значений. Если этот параметр не задан, то события возвращаются независимо от значения поля bizStep (бизнес-этап), если поле bizStep (бизнес-этап) вообще существует
EQ_disposition (соответствие_по_Состоянию)	List of String (Список значений строки)	Аналогично условиям для параметра EQ_bizStep (соответствие_по_Бизнес-этапу), но применительно к полю disposition (состояние)
EQ_readPoint (соответствие_по_Месту_Считывания)	List of String (Список значений строки)	Если параметр указан, то результат будет включать только те события, которые (a) имеют ненулевое значение поля readPoint (место_Считывания) и (b) где значение поля readPoint (место_ Считывания) совпадает с одним из указанных значений. Если этот параметр и параметр WD_readPoint (место_ Считывания_с_ Потомками) оба не заданы, то события возвращаются независимо от значения поля readPoint (место_ Считывания), если поле readPoint (место_Считывания) вообще существует

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
WD_readPoint (место_Считывания_с_Потомками)	List of String (Список значений строки)	Если параметр указан, то результат будет включать только те события, которые (a) имеют ненулевое поле readPoint (место_Считывания) и (b) где значение поля readPoint (место_Считывания) совпадает с одним из указанных значений либо является прямым или косвенным потомком одного из указанных значений. Значение «прямого или косвенного потомка» определяется мастер-данными, как описано в подразделе 6.5 (WD — аббревиатура для «with descendants» («с потомками»)). Если этот параметр и параметр EQ_readPoint (соответствие_по_Месту_Считывания) оба не заданы, то события возвращаются независимо от значения поля readPoint (место_Считывания), если поле readPoint (место_Считывания) вообще существует
EQ_bizLocation (соответствие_по_Производственному_Месту_Назначения)	List of String (Список значений строки)	Аналогично условиям для параметра EQ_readPoint (совпадающее_Место _Считывания), но применительно к полю bizLocation (производственное_Место _Назначения)
WD_bizLocation (производственное_Место_Назначения_с_Потомками)	List of String (Список значений строки)	Аналогично условиям для параметра WD_readPoint (место_Считывания_с _Потомками), но применительно к полю bizLocation (производственное_Место _Назначения)
EQ_bizTransaction_type (соответствие_по_Типу_Документов_Бизнес-транзакций)	List of String (Список значений строки)	Это не единственный параметр, а семейство параметров. Если параметр этой формы указан, то результат будет содержать только те события, которые (a) включают поле bizTransactionList (список_Документов_Бизнес-транзакций), (b) где список документов бизнес-транзакций включает запись, подполе type (тип) которой соответствует type (тип), извлеченному из имени этого параметра, и (c) где значение подполя bizTransaction (Документ_Бизнес-Транзакции) этой записи равно одному из значений, указанных в этом параметре
EQ_source_type (соответствие_по_Типу_Начальных_Пунктов)	List of String (Список значений строки)	Это не единственный параметр, а семейство параметров. Если параметр этой формы указан, то результат будет содержать только те события, которые (a) включают поле sourceList (список_Начальных_Пунктов), (b) где список начальных пунктов включает запись, подполе type (тип) которой соответствует type (тип), извлеченному из имени этого параметра, и (c) где значение подполя source (начальный_пункт) этой записи равно одному из значений, указанных в этом параметре
EQ_destination_type (соответствие_по_Типу_Конечных_Пунктов)	List of String (Список значений строки)	Это не единственный параметр, а семейство параметров. Если параметр этой формы указан, то результат будет содержать только те события, которые (a) включают поле destinationList (список_Конечных_Пунктов), (b) где список конечных пунктов включает запись, подполе type которой соответствует type, извлеченному из имени этого параметра, и (c) где значение подполя destination (конечный_пункт) этой записи равно одному из значений, указанных в этом параметре
EQ_transformationID (соответствие_по_Идентификатору_Преобразования)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле transformationID (идентификатор_Преобразования) (то есть типы TransformationEvent (Событие_с_Преобразованием) или тип события расширения TransformationEvent (Событие_с_Преобразованием)), а также (b) где значение поля transformationID (идентификатор_Преобразования) равно одному из значений, указанных в этом параметре

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
MATCH_epc (совпадение_по_EPC)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле epcList (список_EPC) или поле childEPCs (номера_EPC_Потомков) (то есть типы ObjectEvent (Событие_с_Объектом), AggregationEvent (Событие_с_Агрегацией), TransactionEvent (Событие_с_Документом_Транзакции) или типы события расширения, которые расширяют один из этих трех типов) и (b) где один из номеров EPC, перечисленных в поле epcList (список_EPC) или поле childEPCs (номера_EPC_Потомков) (в зависимости от типа события), совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре, где смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1. Если этот параметр не задан, события включаются независимо от значения поля epcList (список_ EPC) или поля childEPCs (номера_EPC _Потомков), если поле epcList (список_EPC) или поле childEPCs (номера_EPC_Потомков) вообще существуют
MATCH_parentID (совпадение_по_Идентификатору_Родителя)	List of String (Список значений строки)	Аналогично условиям для параметра MATCH_epc (совпадение_по_EPC), но совпадает со значением поля parentID (идентификатор_Родителя) события типа AggregationEvent (Событие_с_ Агрегацией), поля parentID (идентификатор_Родителя) события типа TransactionEvent (Событие_с_Документом_Транзакции) и типами события расширения, которые расширяют либо тип AggregationEvent (Событие_с_Агрегацией), либо TransactionEvent (Событие_с_Документом_Транзакции). Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1
MATCH_inputEPC (совпадение_по_EPC_на_Входе)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле inputEPCList (список_EPC_на_Входе) (то есть тип TransformationEvent (Событие_с_ Преобразованием) или типы события расширения, которые расширяют тип TransformationEvent (Событие_с_Преобразованием)), и (b) где один из номеров EPC, перечисленных в поле inputEPCList (список_EPC_на_Входе), совпадает с одним из шаблонов номера EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1. Если этот параметр не задан, то события включаются независимо от значения их поля inputEPCList (список_EPC_на_Входе), если поле inputEPCList (список_EPC_на_Входе) существует
MATCH_outputEPC (совпадение_по_EPC_на_Выходе)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле outputEPCList (список_EPC_на_Выходе) (то есть тип TransformationEvent (Событие_с_ Преобразованием) или тип события расширения, который расширяет тип TransformationEvent (Событие_с_Преобразованием)), и (b) где один из номеров EPC, перечисленных в поле outputEPCList (список_EPC_на_Выходе), совпадает с одним из шаблонов номера EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1. Если этот параметр не задан, то события включаются независимо от значения их поля outputEPCList (список_ EPC_на_Выходе), если поле outputEPCList (список_EPC_на_Выходе) существует

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
MATCH_anyEPC (совпадение_по_какому-либо_EPC)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле epcList (список_EPC), поле childEPCs (номера_EPC_Потомков), поле parentID (идентификатор_Родителя), поле inputEPCList (список_EPC_на_Входе) или поле outputEPCList (список_EPC_на_Выходе) (то есть типы ObjectEvent (Событие_с_Объектом), AggregationEvent (Событие_с_Агрегацией), TransactionEvent (Событие_с_Документом_Транзакции), TransformationEvent (Событие_с_Преобразованием) или типы события расширения, которые расширяют один из этих четырех типов), и где (b) поле parentID (идентификатор_Родителя) или один из номеров EPC, перечисленных в поле epcList (список_EPC), поле childEPCs (номера_EPC_Потомков), поле inputEPCList (список_EPC_на_Входе) или поле outputEPCList (список_EPC_на_Выходе) (зависит от типа события), совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1
MATCH_epcClass (совпадение_по_Классу_EPC)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле quantityList (список_Количество) или childQuantityList (список_Количества_Потомков) (то есть типы ObjectEvent (Событие_с_Объектом), AggregationEvent (Событие_с_Агрегацией), TransactionEvent (Событие_с_Документом_Транзакции) или типы события расширения, которые расширяют эти три типа), и (b) где один из классов EPC, перечисленных в поле quantityList (список_Количество) или поле childQuantityList (список_Количества_Потомков) (зависит от типа события), совпадает с одним из шаблонов номера EPC или идентификаторов URI, указанных в этом параметре. Результат будет также включать события типа QuantityEvent (Событие_с_Количеством_Объектов), поле epcClass (epc_на_уровне_Класса) которых совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1
MATCH_inputEPCClass (совпадение_по_Классу_EPC_на_Входе)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле inputQuantityList (список_Количество_на_Входе) (то есть тип TransformationEvent (Событие_с_Преобразованием) или типы события расширения, которые расширяют его), и (b) где один из классов EPC, перечисленных в поле inputQuantityList (список_Количество_на_Входе) (зависит от типа события), совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1
MATCH_outputEPCClass (совпадение_по_Классу_EPC_на_Выходе)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле outputQuantityList (список_Количество_на_Выходе) (то есть TransformationEvent (Событие_с_Преобразованием) или типы события расширения, которые расширяют его), и (b) где один из классов EPC, перечисленных в поле outputQuantityList (список_Количество_на_Выходе) (зависит от типа события), совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
MATCH_anyEPCClass (совпадение_по_какому-либо_Классу_EPC)	List of String (Список значений строки)	Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле quantityList (список_Количество), childQuantityList (список_Количества_Потомков), inputQuantityList (список_Количество_на_Входе) или outputQuantityList (список_Количество_на_Выходе) (то есть, типы ObjectEvent (Событие_с_Объектом), AggregationEvent (Событие_с_Агрегацией), TransactionEvent (Событие_с_Документом_Транзакции), TransformationEvent (Событие_с_Преобразованием) или типы события расширения, которые расширяют один их этих четырех типов), и (b) где один из классов EPC, перечисленных в любом из этих полей, совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре. Результат включает также QuantityEvents (Событие_с_Количеством_Объектов), поле epcClass (epc_на_уровне_Класса) которого совпадает с одним из шаблонов номеров EPC или идентификаторов URI, указанных в этом параметре. Смысл понятия «совпадает» соответствует описанному в 8.2.7.1.1
EQ_quantity (соответствие_по_Количеству)	Int (целочисленное значение)	(УСТАРЕЛ в EPCIS версии 1.1) Если этот параметр указан, то результат будет содержать только те события, которые (a) имеют поле quantity (количество) (то есть события типа QuantityEvent (Событие_с_Количеством_Объектов) или типы события расширения, которые расширяют тип QuantityEvent (Событие_с_Количеством_Объектов)), и (b) где значение поля quantity (количество) соответствует указанному параметру
GT_quantity (количество_с_Превышением)	Int (целочисленное значение)	(УСТАРЕЛ в EPCIS версии 1.1) Аналогичен параметру EQ_quantity (соответствие_по_Количеству), но включает события, значение поля quantity (количество) которых больше, чем указанный параметр
GE_quantity (количество_с_Соответствием_и_Превышением)	Int (целочисленное значение)	(УСТАРЕЛ в EPCIS версии 1.1) Аналогичен параметру EQ_quantity (соответствие_по_Количеству), но включает события, значение поля quantity (количество) которых больше или равно указанному параметру
LT_quantity (количество_с_Недостачей)	Int (целочисленное значение)	(УСТАРЕЛ в EPCIS версии 1.1) Аналогичен параметру EQ_quantity (соответствие_по_Количеству), но включает события, значение поля quantity (количество) которых меньше, чем указанный параметр
LE_quantity (количество_с_Соответствием_и_Недостачей)	Int (Целочисленное значение)	(УСТАРЕЛ в EPCIS версии 1.1) Аналогичен параметру EQ_quantity (соответствие_по_Количеству), но включает события, значение поля quantity (количество) которых меньше или равно указанному параметру
EQ_fieldname (соответствие_по_*имени_поля*)	List of String (Список значений строки)	Это не единственный параметр, а семейство параметров. Если параметр этой формы указан, то результат будет содержать только те события, которые (a) имеют расширение высокого уровня поля с именем *fieldname (имя_поля)*, типом которого является либо строка String, либо вид лексики, и (b) где значение этого поля соответствует одному из значений, указанных в этом параметре. Обозначение *fieldname (имя_поля)* является полным составным именем расширения высокого уровня поля. Имя расширения поля — это XML qname; т. е. пара, состоящая из идентификатора пространства имен XML в формате идентификатора URI и собственно имени. Имя соответствующего параметра запроса

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
		строится посредством конкатенации следующих составляющих: строки EQ_, пространства имен URI для расширения поля, знака «номер» (#) и имени расширения поля. Словосочетание "высокий уровень" означает, что соответствующий элемент расширения должен являться непосредственным потомком содержащего события EPCIS, а не элемента, расположенного внутри элемента расширения высокого уровня. Для запроса внутренних элементов расширения см. параметр EQ_INNER_fieldname (внутреннее_соответствие_по_имени_поля)
EQ_fieldname (соответствие_по_*имени_поля*)	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Аналогичен параметру EQ_*fieldname* (соответствие_по_*Имени_Поля*), описанному выше, но может быть применен к полю типа Int (целочисленное значение), Float (Число с плавающей точкой) или Time (Время). Результат будет содержать события, которые (a) имеют поле с именем *fieldname (имя_поля)*, (b) где тип поля совпадает с типом этого параметра (Int, Float или Time) и (c) где значение поля равно указанному значению. Имя поля *fieldname (имя_поля)* строится так же, как и параметр EQ_*fieldname* (соответствие_по_*имени_поля*)
GT_fieldname (больше_ Значение_*по_имени_поля*))	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Как и параметр EQ_*fieldname* (соответствие_по _*имени_поля*), описанный выше, но может быть применен к полю типа Int, Float или Time. Результат будет содержать события, которые (a) имеют поле с именем *fieldname (имя_поля)*, где (b) тип поля совпадает с типом этого параметра (Int, Float или Time) и где (c) значение поля больше, чем указанное значение. Имя поля *fieldname (имя_поля)* строится так же, как и параметр EQ_*fieldname* (соответствие_по_*имени_поля*)
GE_fieldname (соответствующее_и_Большее_Значение_*по_имени_поля*) LT_*fieldname* (меньшее_Значение_*по_имени_поля*) LE_*fieldname* (соответствующее_и_Меньшее_Значение_*по_имени_поля*)	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Аналогичны параметру GT_*fieldname* (большее_Значение_*по_имени_поля*)
EQ_ILMD_*fieldname* (соответствие_по_*имени_поля*_для_МДЭП)	List of String (Список значений строки)	Аналогичен параметру EQ_*fieldname* (*имя_поля_с_Большим_Значением*), но соответствует событиям, область МДЭП (ILMD) которых (см. 7.3.6) содержит поле высокого уровня, имеющее указанное поле *fieldname (имя_поля)*, значение которого совпадает с одним из установленных значений. «Высокий уровень» означает, что соответствующий элемент МДЭП (ILMD) должен являться непосредственным потомком элемента <ilmd>(<мдэп>), а не элемента, расположенного внутри такого элемента. Для запроса внутренних элементов расширения см. параметр EQ_INNER_ILMD_*fieldname*

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
EQ_ILMD_*fieldname* (соответствие_по_имени_поля_для_МДЭП) GT_ILMD_*fieldname* (большее_Значение_по_имени_поля_для_МДЭП) GE_ILMD_*fieldname* (соответствующее_и_Большее_Значение_по_имени_поля_для_МДЭП) LT_ILMD_*fieldname* (меньшее_Значение_по_имени_поля_для_МДЭП) LE_ILMD_*fieldname* (соответствующее_и_Меньшее_Значение_по_имени_поля_для_МДЭП)	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Аналогичен параметрам EQ_*fieldname* (соответствие_по_имени_поля), GT_*fieldname* (большее_Значение_по_имени_поля), GE_*fieldname*, GE_*fieldname* (соответствующее_и_Большее_Значение_по_имени_поля), LT_*fieldname* (меньшее_Значение_по_имени_поля) и LE_*fieldname* (соответствующее_и_Меньшее_Значение_по_имени_поля) соответственно, но соответствует событиям, область МДЭП (ILMD) которых (см. 7.3.6) содержит поле, имеющее указанное поле *fieldname (имя_поля)*, значение integer (целочисленное значение), float (число с плавающей точкой) или time (время) которого совпадает с установленным значением в соответствии с определенным реляционным оператором
EQ_INNER_*fieldname* (внутреннее_соответствие_по_имени_поля)	List of String (Список значений строки)	Аналогичен параметру EQ_*fieldname* (соответствие_по_имени_поля), но соответствует внутренним элементам расширения, то есть любому элементу XML, расположенному внутри элемента расширения высокого уровня. Следует отметить, что соответствующий внутренний элемент может существовать в двух или нескольких элементах высокого уровня или может встречаться дважды или несколько раз в рамках одного элемента высокого уровня; этот параметр совпадает при условии обнаружения хотя бы одного совпадения где-либо в событии (кроме высокого уровня). Следует отметить, что в отличие от элемента расширения высокого уровня внутренний элемент расширения может иметь нулевое пространство имени XML. Для обеспечения соответствия такому внутреннему элементу при формировании имени параметра запроса вместо пространства имени XML используется пустая строка. Например, для обеспечения соответствия внутреннему элементу <elt1> <элемент1> без пространства имени XML необходим параметр запроса EQ_INNER_*elt1* (внутреннее_соответствие_по_элементу1)
EQ_INNER_*fieldname* (*имя_поля*_с_Внутренним_Соответствующим_и_Большим_Значением) GT_INNER_*fieldname* (внутреннее_Большее_Значение_по_имени_поля) GE_INNER_*fieldname* (внутреннее_Соответствующее_и_Большее_Значение_по_имени_поля) LT_INNER_*fieldname* (внутреннее_Меньшее_Значение_по_имени_поля) LE_INNER_*fieldname* (внутреннее_Соответствующее_и_Меньшее_Значение_по_имени_поля)	Int (Целочис-ленное значение) Float (Число с плавающей точкой) Time (Время)	Подобен параметру EQ_INNER_*fieldname* (внутреннее_соответствие_по_имени_поля), описанному выше, но может быть применен к полю типа Int (целое число), Float (число с плавающей точкой) или Time (время)

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
EQ_INNER_ILMD_*fieldname* (внутреннее_соответствие_по_имени_поля_для_МДЭП)	List of String (Список значений строки)	Подобен параметру EQ_ILMD_*fieldname* (соответствие_по_имени_поля_для_МДЭП), но соответствует внутренним элементам МДЭП (ILMD), то есть любому элементу XML, расположенному на любом уровне внутри элемента МДЭП (ILMD) высокого уровня. Следует отметить, что соответствующий внутренний элемент может существовать в двух или нескольких элементах высокого уровня или может встречаться дважды или несколько раз в рамках одного элемента высокого уровня; этот параметр совпадает при условии обнаружения хотя бы одного совпадения где-либо в разделе МДЭП (ILMD) (кроме высокого уровня)
EQ_INNER_ILMD_*fieldname* (внутреннее_соответствие_по_имени_поля_для_МДЭП) GT_INNER_ILMD_*fieldname* (внутреннее_Большее_Значение_по_имени_поля_для_МДЭП) GE_INNER_ILMD_*fieldname* (внутреннее_Соответствующее_и_Большее_Значение_по_имени_поля_для_МДЭП) LT_INNER_ILMD_*fieldname* (внутреннее_Меньшее_Значение_по_имени_поля_для_МДЭП) LE_INNER_ILMD_*fieldname* (внутреннее_Соответствующее_и_Меньшее_Значение_по_имени_поля_для_МДЭП)	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Подобен параметру EQ_INNER_ILMD_*fieldname* (внутреннее_соответствие_по_имени_поля_для_МДЭП), описанному выше, но может быть применен к полю типа Int (целочисленное значение), Float (число с плавающей точкой) или Time (время)
EXISTS_*fieldname* (действительно_е_Соответствие_по_имени_поля)	Void (Пустой тип данных)	Подобен параметру *EQ_fieldname (соответствие_по_имени_поля)*, описанному выше, но может быть применен к полю любого типа (включая комплексные типы). Результат будет включать события, которые имеют непустое поле с именем *fieldname (имя_поля)*. Имя поля *fieldname (имя_поля)* строится так же, как для параметра EQ_*fieldname* (соответствие_по_имени_поля). Следует обратить внимание, что значение для этого параметра запроса игнорируется
EXISTS_INNER_*fieldname* (действительное_Внутреннее_Соответствие_по_имени_поля)	Void (Пустой тип данных)	Подобен параметру EXISTS_*fieldname* (действительное_Соответствие_по_имени_поля), описанному выше, но включает события, имеющие непустое поле внутреннего расширения с именем *fieldname (имя_поля)*. Следует обратить внимание, что значение этого параметра запроса игнорируется
EXISTS_ILMD_*fieldname* (действительное_Соответствие_по_имени_поля_для_МДЭП)	Void (Пустой тип данных)	Подобен параметру EXISTS_*fieldname* (действительное_Соответствие_по_имени_поля), описанному выше, кроме событий, имеющих непустое поле с именем *fieldname (имя_поля)* в области МДЭП (ILMD) (7.3.6). Имя поля *fieldname (имя_поля)* строится так же, как для параметра EQ_ILMD_*fieldname* (соответствие_по_имени_поля_для_МДЭП). Следует обратить внимание, что значение для этого параметра запроса игнорируется

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
EXISTS_INNER_ILMD_*fieldname* (действительное_Внутреннее_соответствие_по_*имени_поля_для_* МДЭП)	Void (Пустой тип данных)	Подобен параметру EXISTS_ILMD_*fieldname*, описанному выше, но включает события, имеющие непустое поле внутреннего расширения с именем *fieldname (имя_поля)* в области МДЭП (ILMD). Следует обратить внимание, что значение этого параметра запроса игнорируется
HASATTR_*fieldname* (наличие_Ненулевого_Атрибута_по_*имени_поля*)	List of String (Список значений строки)	Это не единственный параметр, а семейство параметров. Если параметр этой формы установлен, результат будет содержать только те события, которые (a) имеют поле с именем *fieldname (имя_поля)*, типом которого является вид лексики, и (b) где значением этого поля является лексический элемент, для которого доступны мастер-данные, и (c) мастер-данные имеют отличный от нуля атрибут, имя которого совпадает с одним из значений, установленных в этом параметре. Имя *fieldname (имя_поля)* является полностью определенным именем (полным именем) поля. Для обычного поля это просто имя поля, например bizLocation. (производственное_Место_Назначения). Для расширения поля именем расширения поля является составное имя XML qname, то есть пара, состоящая из идентификатора формата URI из пространства имен XML и собственно имени. Имя соответствующего параметра запроса строится посредством конкатенации следующих составляющих: строки HASATTR_ (наличие_Ненулевого_Атрибута), идентификатора пространства имен URI для поля расширения, знака «номер» (#) и имени поля расширения
EQATTR_*fieldname_attrname* (соответствие_атрибута_по_*имени_поля_и_имени_атрибута*)	String (Список значений строки)	Это не единственный параметр, а семейство параметров. Если параметр этой формы указан, результат будет содержать только те события, которые (a) имеют поле с именем *fieldname (имя_поля)*, типом которого является вид лексики, и (b) где значением этого поля является лексический элемент, для которого доступны мастер-данные, и (c) мастер-данные имеют ненулевой атрибут с именем *attrname (имя_атрибута)*, и (d) где значение этого атрибута совпадает с одним из значений, указанных в этом параметре. Имя поля *fieldname (имя_поля)* строится так же, как и HASATTR_*fieldname* (наличие_Ненулевого_Атрибута_Мастер-Данных_по_*имени_поля*). Реализация МОЖЕТ ответить с помощью исключения QueryParameterException (Исключение_по_Параметру_Запроса), если поле *fieldname (имя_поля)* или поле *attrname (имя_атрибута)* включают знак «подчеркивание». Примечание — поскольку наличие знака «подчеркивание» в поле fieldname (имя_поля) или поле attrname (имя_атрибута) представляет неоднозначность в отношении того, где лежит разделение между fieldname (имя_поля) и attrname (имя_атрибута), реализация может отклонить параметр запроса, если она не может устранить двусмысленность
EQ_eventID (соответствие_по_Идентификатору_События)	List of String (Список значений строки)	Если параметр этой формы указан, результат будет включать только те события, которые (a) имеют поле eventID (идентификатор_События) со значением, отличным от нулевого, и (b) где поле eventID (идентификатор_События) имеет значение, равное одному из значений, установленных в этом параметре. Если данный параметр опущен, события возвращаются вне зависимости от значения поля eventID (идентификатор_События) или даже от наличия поля eventID (идентификатор_События)

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
EXISTS_errorDeclaration (действительные_Заявления_об_Ошибке)	Void (Пустой тип данных)	Если параметр этой формы указан, результат будет включать только те события, которые содержат ErrorDeclaration (Заявление_об_Ошибке). Если данный параметр опущен, события возвращаются вне зависимости от наличия у них ErrorDeclaration (Заявление_об_Ошибке)
GE_errorDeclarationTime (время_Соответствующих_и_Последующих_Заявлений_об_Ошибках)	Time (Время)	Если параметр этой формы указан, результат будет включать только те события, которые (a) содержат ErrorDeclaration (Заявление_об_Ошибке) и (b) где значение поля errorDeclarationTime больше или равно указанному значению. Если данный параметр опущен, события возвращаются вне зависимости от наличия у них ErrorDeclaration (Заявление_об_Ошибке) и значения поля errorDeclarationTime (время_Заявлений_об_Ошибках)
LT_errorDeclarationTime (время_Соответствующих_и_Последующих_Заявлений_об_Ошибках)	Time (Время)	Если данный параметр этой формы указан, результат будет включать только те события, которые (a) содержат ErrorDeclaration (Заявление_об_Ошибке) и (b) где значение поля errorDeclarationTime меньше установленного значения. Если данный параметр опущен, события возвращаются вне зависимости от наличия у них ErrorDeclaration (Заявление_об_Ошибке) и значения поля errorDeclarationTime
EQ_errorReason (соответствие_Причине_Ошибки)	List of String (Список значений строки)	Если этот параметр указан, результат будет включать только те события, которые (a) содержат ErrorDeclaration (Заявление_об_Ошибке), (b) где заявление об ошибке содержит поле reason (причина_Ошибки) со значением, отличным от нулевого, и (c) где поле reason (причина_Ошибки) равно одному из значений, указанных в настоящем параметре. Если данный параметр опущен, события возвращаются вне зависимости от наличия у них ErrorDeclaration (Заявление_об_Ошибке) и значения поля reason (причина_Ошибки)
EQ_correctiveEventID (соответствие_Идентификатору_Корректирующего_События)	List of String (Список значений строки)	Если параметр этой формы указан, результат будет включать только те события, которые (a) содержат ErrorDeclaration (Заявление_об_Ошибке) и (b) где один из элементов списка correctiveEventIDs (идентификаторы_Корректирующих_Событий) равен одному из значений, указанных в настоящем параметре. Если данный параметр опущен, события возвращаются вне зависимости от наличия у них ErrorDeclaration (Заявление_об_Ошибке) и от содержимого списка correctiveEventIDs (идентификаторы_Корректирующих_Событий)
EQ_ERROR_DECLARATION_*fieldname* (соответствие_Заявления_об_Ошибке_по_*имени_поля*)	List of String (Список значений строки)	Аналогичен параметру EQ_*fieldname* (соответствие_по_*имени_поля*), но соответствует событиям, содержащим ErrorDeclaration (Заявление_об_Ошибке), и где ErrorDeclaration (Заявление_об_Ошибке) содержит поле, имеющее указанное поле *fieldname (имя_поля)*, значение которого соответствует одному из установленных значений
EQ_ERROR_DECLARATION_*fieldname* (соответствие_Заявления_об_Ошибке_по_*имени_поля*) GT_ERROR_DECLARATION_*fieldname* (большее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке)	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Аналогичен параметрам EQ_*fieldname* (соответствие_по_*имени_поля*), GT_*fieldname* (большее_Значение_*по_имени_поля*), GE_*fieldname* (соответствующее_и_Большее_Значение_*по_имени_поля*), GE_*fieldname*, LT_*fieldname* (меньшее_Значение_*по_имени_поля*) и LE_*fieldname* (соответствующее_и_Меньшее_Значение_*по_имени_поля*) соответственно, но соответствует событиям, содержащим ErrorDeclaration (Заявление_об_Ошибке), и где ErrorDeclaration (Заявление_об_Ошибке)

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
GE_ERROR_DECLARATION_*fieldname* (соответствующее_и_Большее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке) LT_ERROR_DECLARATION_*fieldname* (меньшее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке) LE_ERROR_DECLARATION_*fieldname* (соответствующее_и_Меньшее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке)		содержит поле, имеющее установленное поле *fieldname (имя_поля)*, значение integer (целочисленное значение), float (число с плавающей точкой) или time (время) которого совпадает с указанным значением согласно установленному реляционному оператору
EQ_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Соответствие_*по_имени_поля*_для_Заявления_об_Ошибке)	List of String (Список значений строки)	Аналогичен параметру EQ_ERROR_DECLARATION_*fieldname* (соответствие_Заявления_об_Ошибке_*по_имени_поля*), но соответствует внутренним элементам расширения, то есть любому элементу XML, расположенному внутри элемента расширения высокого уровня. Следует отметить, что соответствующий внутренний элемент может существовать в двух или нескольких элементах высокого уровня или может встречаться дважды или несколько раз в рамках одного элемента высокого уровня; этот параметр совпадает при условии обнаружения хотя бы одного совпадения где-либо в событии (кроме высокого уровня)
EQ_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Соответствие_*по_имени_поля*_для_Заявления_об_Ошибке) GT_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Большее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке) GE_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Соответствующее_и_Большее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке) LT_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Меньшее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке) LE_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Соответствующее_и_Меньшее_Значение_*по_имени_поля*_для_Заявления_об_Ошибке)	Int (Целочисленное значение) Float (Число с плавающей точкой) Time (Время)	Подобен параметру EQ_INNER_ERROR_DECLARATION_*fieldname* (внутреннее_Соответствие_*по_имени_поля*_для_Заявления_об_Ошибке), описанному выше, но может быть применен к полю типа Int (целочисленное значение), Float (число с плавающей точкой) или Time (время)

Продолжение таблицы 31

Имя параметра	Тип значения параметра	Смысл
EXISTS_ERROR_DECLARATION_*fieldname* (действительное_Соответствие_по_*имени_поля*)	Void (Пустой тип данных)	Подобен параметру EXISTS_*fieldname* (действительное_Соответствие_по_*имени_поля*), описанному выше, но соответствует событиям, имеющим заявление об ошибке с включением непустого поля расширения с именем *fieldname (имя_поля)*. Поле *Fieldname (имя_поля)* формируется аналогично полю EQ_ERROR_DECLARATION_*fieldname* (соответствие_Заявления_об_Ошибке_по *имени_поля*). Следует отметить, что значение этого параметра запроса игнорируется
EXISTS_INNER_ERROR_DECLARATION_*fieldname* (действительное _Внутреннее_Соответствие_по_*имени_поля*)	Void (Пустой тип данных)	Подобен параметру EXISTS_ERROR_DECLARATION_*fieldname* (действительное_Соответствие_по_*имени_поля*), описанному выше, но включает события, имеющие заявления об ошибках с включением непустого поля внутреннего расширения с именем *fieldname (имя_поля)*. Следует отметить, что значение этого параметра запроса игнорируется
orderBy (упорядоченность_по_Полю)	String (Строка)	Если параметр указан, то он именует единственное поле, которое будет использоваться для упорядочения результатов. Поле orderDirection (порядок_Следования) указывает, находится ли порядок следования в восходящей или убывающей последовательности. События, включенные в результат, которые вообще не имеют указанного поля, могут встречаться в любой позиции в списке результатов. Значение этого параметра ДОЛЖНО быть одним из следующих: eventTime (время_События), recordTime (время_Записи) или полное составное имя поля расширения, тип которого — Int (целочисленное значение), Float (число с плавающей точкой), или Time (время), или String (строка). Полное составное имя fieldname (имя_поля) строится так же, как и для параметра EQ_*fieldname* (соответствие_по_*имени_поля*). В случае поля с типом String (строка), порядок следования ДОЛЖЕН соответствовать лексикографическому порядку на основе кодирования строк согласно набору знаков Unicode или в какой-либо иной систематизирующей последовательности, соответствующей месту действия. Если параметр не задан, то порядок не указан. Реализация МОЖЕТ систематизировать результаты в любом выбранном ею порядке, и этот порядок МОЖЕТ отличаться даже тогда, когда один и тот же запрос выполняется дважды для одних и тех же данных. (В *[3]* (EPCIS версии 1.0) также было разрешено значение quantity (количество), но его использование устарело в *[4]* (EPCIS версии 1.1))
orderDirection (порядок_Следования)	String (Строка)	Если параметр указан и также указан параметр orderBy (упорядоченность_по_Полю), то он показывает, находится ли порядок следования в восходящей или убывающей последовательности в соответствии с ключом, определенным параметром orderBy (упорядоченность_по_Полю). Значение этого параметра должно быть либо ASC (ВОЗРАСТАНИЕ) (для восходящего порядка), либо DESC (УБЫВАНИЕ) (для убывающего порядка). Если условие не выполнено, реализация ДОЛЖНА ответить с помощью исключения QueryParameterException (Исключение_по_Параметру_Запроса). Если параметр не задан, то его значением по умолчанию должно быть DESC (УБЫВАНИЕ)

Окончание таблицы 31

Имя параметра	Тип значения параметра	Смысл
eventCountLimit (ограничение_по_Числу_Событий)	Int (целочисленное значение)	Если параметр указан, то результаты будут включать только первые N событий, которые соответствуют другим критериям, где N — значение этого параметра. Порядок следования, заданный параметрами orderBy (упорядоченность_по_Полю) и orderDirection (порядок_Следования), определяет смысл слова «первые». Если параметр не задан, все события, удовлетворяющие заданному критерию, будут включены в результаты. Указанный параметр и параметр maxEventCount (максимальное_Число_Событий) являются взаимоисключающими. Если заданы оба параметра, то ДОЛЖНО применяться исключение QueryParameterException (Исключение_по_ Параметру_Запроса). Этот параметр может быть использован, только если задан параметр orderBy (упорядоченность _по_Полю). Если параметр orderBy (упорядоченность_по_Полю) не задан и задан параметр eventCountLimit (ограничение_по_Числу_Событий), должно применяться исключение QueryParameterException (Исключение_по_Параметру_Запроса). Данный параметр отличается от параметра maxEventCount (максимальное_Число _Событий) в том, что он ограничивает количество возвращаемых данных, в то время как параметр maxEventCount (максимальное_Число _Событий) вызывает исключение, если предел превышен. Примечание — Общепринятое использование параметров orderBy (упорядоченность_по_Полю), orderDirection и eventCountLimit (ограничение_по_Числу _Событий) служит для экстремальных запросов. Например, для того, чтобы выбрать самое последнее событие, удовлетворяющее некоторому критерию, запрос будет включать параметры, которые отбирают события, удовлетворяющие желаемому критерию, и устанавливают значения параметра orderBy (упорядоченность_по_Полю) в eventTime (время_События), параметра orderDirection (порядок_Следования) в DESC (УБЫВАНИЕ) и параметра eventCountLimit (ограничение_по_Числу_Событий) в единицу
maxEventCount (максимальное_Число_Событий)	Int (целочисленное значение)	Если параметр указан, то в результат запроса будет включено событий не больше, чем значение данного параметра. В противном случае, если запрос возвращал бы больше указанного числа событий, вместо обычного результата запроса ДОЛЖНО применяться исключение QueryTooLargeException (Исключение_при_ Превышении_Объема_Запроса). Данный параметр и параметр eventCountLimit (ограничение_по_Числу_Событий) — взаимоисключающие. Если заданы оба параметра, ДОЛЖНО применяться исключение QueryParameterException (Исключение_по_Параметру_ Запроса). Если этот параметр не задан, в результат запроса может быть включено любое число событий. Следует обратить внимание, что реализация EPCIS может применить исключение QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) независимо от значения этого параметра (см. 8.2.3)

Как следует из приведенных выше описаний, если указано несколько параметров, событие должно удовлетворять всем критериям для включения в результирующий набор. Другими словами, если каждый параметр рассматривается как логическое утверждение, все такие логические утверждения неявно соединяются как бы с помощью оператора И (AND). Например, если данный вызов метода poll (опрос) определяет значение для обоих параметров, EQ_bizStep (соответствие_по_Бизнес-этапу) и EQ_disposition (соответствие_ по_Состоянию), тогда для включения в результат событие должно соответствовать одному из установленных значений поля bizStep (бизнес-этап) И (AND) совпадать с одним из установленных значений поля disposition (состояние).

С другой стороны, для тех параметров, значением которых является список, событие должно соответствовать *по крайней мере одному* из элементов списка, чтобы быть включенным в результирующий набор. Другими словами, если каждый элемент списка рассматривается как логическое утверждение, все такие логические утверждения для данного списка явно разъединяются как бы оператором ИЛИ (OR). Например, если значением параметра EQ_bizStep (соответствие_по_Бизнес-этапу) является список из двух элементов ("bs1", "bs2"), то событие включается, если его поле bizStep (бизнес-этап) содержит значение bs1 ИЛИ (OR) его поле bizStep (бизнес-этап) содержит значение bs2.

В другом примере, если значение параметра EQ_bizStep (соответствие_по_Бизнес-этапу) является списком из двух элементов ("bs1", "bs2"), а параметр EQ_disposition (соответствие_ по_Состоянию) представляет собой список из двух элементов ("d1", "d2"), то результат должен включать события, удовлетворяющие следующему логическому утверждению:

((bizStep = "bs1" OR bizStep = "bs2")
AND (disposition = "d1" OR disposition = "d2"))

8.2.7.1.1 Обработка параметров запроса MATCH (совпадение_по)

Список параметров для MATCH_epc (совпадение_по_EPC), MATCH_parentID (совпадение_ по_Идентификатору_Родителя), MATCH_inputEPC (совпадение_ по_EPC_на_Входе), MATCH_ outputEPC (совпадение_по_EPC_на_Выходе) и MATCH_anyEPC (совпадение_по_какому-либо_EPC)_ ДОЛЖЕН обрабатываться следующим образом. Каждый элемент списка параметров может быть шаблоном чистого ключевого идентификатора (pure identity pattern), указанным в [12], или любым другим идентификатором URI. Если элемент является шаблоном чистого ключевого идентификатора, он сопоставляется с значениями поля события, используя процедуру сопоставления шаблонов ключевого идентификатора, указанную в [12, раздел 8]. Если этот элемент является любым другим идентификатором URI, он сопоставляется со значениями полей события посредством проверки равенства строки.

Список параметров для MATCH_epcClass (совпадение_по_Классу_EPC), MATCH_ inputEPCClass (совпадение_по_Классу_EPC_на_Входе), MATCH_outputEPCClass (совпадение_по_Классу_EPC_на_Выходе) и MATCH_anyEPCClass (совпадение_по_какому-либо_Классу_EPC) ДОЛЖЕН обрабатываться следующим образом. Пусть *P* — один из шаблонов, указанных в значении этого параметра, и пусть *C* будет значением поля epcClass (epc_на_уровне_Класса) в соответствующем списке значений количества, рассматриваемого для включения в результат. Тогда событие включается в результат, если каждый компонент *Pi* из *P* совпадает с соответствующей компонентой *Ci* из *C*, где понятие «совпадает» определено в [12, раздел 8].

Примечание — Различие между параметрами MATCH_epcClass (совпадение_по_Классу_EPC) и MATCH_epc (совпадение_по_EPC), а также похожими параметрами, заключается в том, что для параметра MATCH_epcClass (совпадение_по_Классу_EPC) значение в событии (поле epcClass (epc_на_уровне_Класса) в списке значений количества) может само по себе являться шаблоном, как указано в 7.3.3.3). Это означает, что значение в событии может содержать в параметре запроса компоненту «*». Например, если поле epcClass (epc_на_уровне_Класса) внутри события является urn: epc:idpat:sgtin:0614141.112345.*, то этому событию будет соответствовать параметр запроса urn: epc:idpat:sgtin:0614141.*.* или urn:epc:idpat:sgtin:0614141.112345.*, но не urn:epc:idpat:sgtin:0614141.112345.400.

8.2.7.2 Запрос SimpleMasterDataQuery (Простой_Запрос_Мастер_Данных)

Этот запрос вызывается путем указания строки SimpleMasterDataQuery (Простой_Запрос_ Мастер_Данных) в качестве аргумента queryName (имя_Запроса) для метода poll (опрос). Результатом является экземпляр QueryResults (Результаты_Запроса), тело которого содержит (возможно, пустой) список лексических элементов вместе с выбранными атрибутами.

Запрос SimpleMasterDataQuery (Простой_Запрос_Мастер_Данных) ДОЛЖЕН быть доступен через метод poll (опрос), но не через метод subscribe (подписка); то есть реализация ДОЛЖНА ответить с помощью исключения SubscribeNotPermittedException (Исключение_при_Недопустимой _Подписке), если запрос SimpleMasterDataQuery (Простой_Запрос_ Мастер_Данных) задан как аргумент queryName (имя_Запроса) для subscribe (подписка).

Параметры для этого запроса приведены в *таблице 32*:

Таблица 32

Имя параметра	Тип значения параметра	Обязательность параметра	Смысл
vocabularyName (название_Лексикона)	List of String (Список значений строки)	Нет	Если параметр указан, то в результаты будут включены только лексические элементы, взятые из одного из установленных лексиконов. Каждый элемент установленного списка является официальным именем идентификатора URI для лексикона. Например, один из идентификаторов URI, указанных в таблице 5. Если параметр не задан, то рассматриваются все лексиконы
includeAttributes (включение_Атрибутов)	Boolean (логическое выражение)	Да	Если значение параметра — true (истина), то результаты будут включать имена атрибутов и значения для соответствия лексическим элементам. Если false (ложь) — имена и значения атрибутов не будут включены в результат
includeChildren (включение_Потомков)	Boolean (логическое выражение)	Да	Если значение параметра — true (истина), то результаты будут включать список потомков для соответствия лексическим элементам. Если false (ложь) — списки потомков не будут включены в результат
attributeNames (имена_Атрибутов)	List of String (Список значений строки)	Нет	Если параметр указан, то в результаты будут включены только те атрибуты, имена которых совпадают с одним из указанных имен. Если параметр не задан, то будут включены все атрибуты для каждого соответствующего лексического элемента. (Чтобы получить список имен лексических элементов с атрибутами, указывают false (ложь) для параметра includeAttributes (включение_Атрибутов).) Значение этого параметра ДОЛЖНО быть проигнорировано, если значение параметра includeAttributes (включение_Атрибутов) — false (ложь). Следует обратить внимание, что этот параметр не влияет на то, какие лексические элементы включаются в результат запроса. Он только ограничивает то, какие атрибуты будут включены с каждым лексическим элементом
EQ_name (соответствие_по_Имени)	List of String (Список значений строки)	Нет	Если параметр указан, то результат будет включать только лексические элементы, имена которых соответствуют одному из указанных значений. Если этот параметр и параметр WD_name (соответствие_по_Имени_с_учетом_Потомков) оба не заданы, то лексические элементы включаются независимо от их имен
WD_name (соответствие_по_Имени_с_учетом_Потомков)	List of String (Список значений строки)	Нет	Если параметр указан, то результат будет включать только те лексические элементы, которые либо соответствуют одному из указанных имен, либо являются прямыми или косвенными потомками лексического элемента, который соответствует одному из указанных имен. Смысл понятия «прямой или косвенный потомок» описывается в 6.5. (WD является аббревиатурой для фразы «с потомками» («with descendants».) Если этот параметр и EQ_name (соответствие_по_Имени) оба не заданы, то лексические элементы включаются независимо от их имен
HASATTR (наличие_Ненулевого_Атрибута)	List of String (Список значений строки)	Нет	Если параметр указан, то результат будет включать только те лексические элементы, которые имеют отличный от нуля атрибут, имя которого совпадает с одним из значений, указанных в этом параметре

Окончание таблицы 32

Имя параметра	Тип значения параметра	Обязательность параметра	Смысл
EQATTR_*attrname* (соответствие_по_*имени_атрибута*)	List of String (Список значений строки)	Нет	Это не единственный параметр, а семейство параметров. Если параметр в данной форме задан, то результат будет включать только те лексические элементы, которые имеют отличный от нуля атрибут с именем *attrname (имя_атрибута)* и где значение этого атрибута совпадает с одним из значений, указанных в данном параметре
maxElementCount (максимальное_Число_Элементов)	Int (целочисленное значение)	Нет	Если параметр указан, то в результат запроса будет включено лексических элементов не больше, чем значение данного параметра. В противном случае, если бы запрос возвращал больше указанного числа лексических элементов, вместо обычного результата запроса ДОЛЖНО применяться исключение QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса). Если этот параметр не задан, в результат запроса может быть включено любое число лексических элементов. Следует обратить внимание, что независимо от значения этого параметра реализация EPCIS может применять исключение QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) (см. 8.2.3)

Как следует из приведенных выше описаний, если указано несколько параметров, то для включения в результирующий набор лексический элемент должен удовлетворять всем критериям. Другими словами, если каждый параметр рассматривается как логическое утверждение, то все такие логические утверждения неявно соединяются как бы с помощью оператора AND (И). Например, если данный вызов метода poll (опрос) определяет значение для обоих параметров, WD_name (соответствие_по_Имени _с_учетом_Потомков) и HASATTR (наличие_Ненулевого_Атрибута), то тогда для включения в результат лексический элемент должен быть потомком установленного элемента И владеть одним из указанных атрибутов.

С другой стороны, для тех параметров, значением которых является список, лексический элемент должен соответствовать *по крайней мере одному* из элементов списка для включения в результирующий набор. Другими словами, если каждый элемент списка рассматривается как логическое утверждение, то все такие логические утверждения для данного списка явно разъединяются как бы оператором OR (ИЛИ). Например, если значением параметра EQATTR_sample (соответствие_по_Атрибуту_Образец) является список из двух элементов ("s1", "s2"), то лексический элемент включается, если он имеет атрибут sample (образец), значение которого соответствует s1 ИЛИ равно s2.

В другом примере, если значение параметра EQ_name (соответствие_по_Имени) является списком из двух элементов ("ve1", "ve2"), а параметр EQATTR_sample (соответствие_по_Атрибуту_Образец) представляет собой список из двух элементов ("s1", "s2"), то результат должен включать события, удовлетворяющие следующему логическому утверждению:

((name = "ve1" OR name = "ve2")
AND (sample = "s1" OR sample = "s2"))

где name неформально ссылается на имя лексического элемента, а sample неформально ссылается на значение атрибута sample (образец).

8.2.8 Интерфейс обратных вызовов по запросам

Интерфейс обратных вызовов по запросам (*Query Callback Interface*) — это путь, по которому сервис EPCIS доставляет результаты длительного запроса клиенту.

```
<<interface>>
EPCISQueryCallbackInterface
---
callbackResults(resultData : QueryResults)   :  void
callbackQueryTooLargeException(e : QueryTooLargeException)   :  void
callbackImplementationException(e : ImplementationException)   :  void
```

Каждый раз, когда сервис EPCIS выполняет длительный запрос в соответствии с расписанием QuerySchedule (Расписание_Запросов), он ДОЛЖЕН попытаться доставить результаты подписчику посредством вызова одного из трех методов интерфейса обратных вызовов по запросам. Если запрос выполнен нормально, сервис EPCIS ДОЛЖЕН вызвать метод callbackResults (результаты_Обратного_Вызова). Если в результате выполнения запроса было применено исключение QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) или ImplementationException (Исключение_по_Реализации), сервис EPCIS ДОЛЖЕН вызвать соответствующий метод интерфейса обратных вызовов по запросам.

Следует обратить внимание, что «исключения» в интерфейсе обратных вызовов по запросам не являются исключениями в привычном смысле, как исключения прикладного программного интерфейса API, поскольку они не возникают как следствие вызова метода клиентом. Вместо этого исключение доставляется получателю тем же способом, что и нормальный результат, как аргумент метода интерфейса.

9 Привязки XML для модулей определения данных

Настоящий раздел определяет типовую привязку XML для модуля определения данных типов основных событий с использованием языка определения схемы для документов XML (XML Schema language) стандарта W3C (World Wide Web Consortium) см. *[19]*, *[18]*. Также приводятся примеры.

Схема, приведенная ниже, согласуется с правилами проектирования типовых схем GS1. Приведенная ниже схема импортирует типовую базовую схему EPCglobal в соответствии с правилами проектирования по *[20]*.

9.1 Механизм расширяемости

Схема XML в данном разделе реализует точку расширения <<extension point>>, приведенную на языке UML в разделе 6, с использованием методологии, описанной в [21]. Эта методология предусматривает как расширение поставщика/пользователя, так и расширение в соответствии с правилами GS1 в будущих версиях данной спецификации или в дополнительных спецификациях. Расширения, внедренные через этот механизм, будут иметь совместимость снизу вверх, так как документы, соответствующие старым версиям схемы, будут также соответствовать более новым версиям типовой схемы и схеме, содержащей расширения для конкретных поставщиков. Расширения также будут совместимы снизу вверх с будущими версиями, поскольку документы, содержащие расширения поставщиков/пользователей или соответствующие более новым версиям типовой схемы, также будут соответствовать более старым версиям схемы.

Если документ содержит расширения (для конкретных поставщиков/пользователей или стандартизированных в более новых версиях схемы), он может соответствовать более чем одной схеме. Например, документ, содержащий расширения поставщиков до схемы GS1 версии 1.0 *по* [3], будет соответствовать как схеме GS1 версии 1.0 *по* [3], так и схеме конкретного поставщика, которая включает расширения поставщика. В этом примере при анализе документа с использованием типовой схемы не будет проверки элементов и атрибутов расширения, но при анализе документа с использованием схемы, зависящей от поставщика, расширения должны быть проверены. Аналогично документ, содержащий новые функции, представленные в схеме *согласно настоящему стандарту* и [2], будет соответствовать как схеме *по* [3], так и схеме *по* [4] и схеме *согласно настоящему стандарту* и [2], но проверка новых функций будет доступна только при использовании схемы *согласно настоящему стандарту* и [2].

Правила проектирования для этого шаблона расширяемости приведены в документе *[21]*. В итоге они сводятся к следующему:

- Для каждого типа, в котором встречается точка расширения <<extension point>>, включают объявление xsd:anyAttribute. Это объявление предусматривает добавление новых атрибутов XML либо в последующих версиях типовой схемы, либо в схеме поставщика/пользователя.

- Для каждого типа, в котором встречается точка расширения <<extension point>>, включают необязательный элемент (minOccurs = 0) с именем extension (расширение). Тип, объявленный для элемента extension (расширение), будет всегда выглядеть так:

```
<xsd:sequence>
<xsd:any processContents="lax" minOccurs="1" maxOccurs="unbounded"
namespace="##local"/>
</xsd:sequence>
<xsd:anyAttribute processContents="lax"/>
```

Это объявление обеспечивает совместимость снизу вверх с новыми элементами, введенными в последующие версии типовой схемы.

- Для каждого типа, в котором встречается точка расширения <<extension point>>, включают в конце элемента список объявлений

```
<xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"
namespace="##other"/>
```

Это объявление обеспечивает совместимость снизу вверх с новыми элементами, введенными в схему конкретного поставщика/пользователя.

Правила добавления расширений конкретных поставщиков/пользователей в схему следующие:

- атрибуты конкретного поставщика/пользователя могут быть добавлены к любому типу, в котором встречается точка расширения <<extension point>>. Атрибуты конкретного поставщика/пользователя НЕ ДОЛЖНЫ находиться в пространстве имен EPCglobal EPCIS (urn:epcglobal:epcis:xsd:1) или в пустом пространстве имен. Атрибуты конкретного поставщика/пользователя ДОЛЖНЫ находиться в пространстве имен, идентификатор формата URI которого содержит имя поставщика в качестве организации-владельца. (На языке схемы это означает, что все атрибуты, зависящие от поставщика/пользователя, должны быть указаны в форме qualified (составное_имя). Например, пространству имен URI может принадлежать адрес URL протокола HTTP, полномочная часть которого является доменным именем, принадлежащим поставщику/пользователю; имя URN, имеющее идентификатор пространства имен URN, выданный поставщику/пользователю организацией IANA; имя OID URN, начальным путем которого является номер частного предприятия (Private Enterprise Number), присвоенный поставщику/пользователю, и т. д. Объявления атрибутов конкретного поставщика/пользователя ДОЛЖНЫ устанавливать use=»optional»;
- элементы конкретного поставщика/пользователя могут быть добавлены к любому типу, в котором присутствует точка расширения <<extension point>>. Элементы конкретного поставщика/пользователя НЕ ДОЛЖНЫ находиться в пространстве имен EPCglobal EPCIS (urn:epcglobal:epcis:xsd:1) или в пустом пространстве имен. Элементы конкретного поставщика/пользователя ДОЛЖНЫ находиться в пространстве имен, идентификатор URI которого содержит имя поставщика в качестве организации-владельца (как описано выше). (На языке схемы это означает, что все элементы, зависящие от поставщика/пользователя, должны быть указаны в форме qualified (составное_имя).

Для создания схемы, которая содержит расширения поставщика/пользователя, производят замену объявления <xsd:any … namespace="##other"/> ссылкой на группу контента для группы, определенной в пространстве имен поставщика/пользователя. Например, <xsd:group ref="vendor:VendorExtension">. В данной схеме файл, определяющий элементы для пространства имен поставщика/пользователя, определяет группу контента, используя объявление в следующей форме:

```
<xsd:group name="VendorExtension">
  <xsd:sequence>
    <!--
      Определения или ссылки на элементы поставщика
      следуют здесь. Каждое ДОЛЖНО задавать minOccurs="0".
    -->
    <xsd:any processContents="lax"
        minOccurs="0" maxOccurs="unbounded"
        namespace="##other"/>
  </xsd:sequence>
</xsd:group>
```

(В последующих иллюстрациях vendor (поставщик) и VendorExtension (Расширение_поставщика) могут быть любыми строками, которые выберет поставщик/пользователь).

Примечание — Поскольку элементы, зависящие от конкретных поставщиков/пользователей, должны быть необязательны, включение ссылок на их определения непосредственно в схему EPCIS будет нарушать ограничение присвоения уникальной частицы (Unique Particle Attribution) схемы XML, так как элемент <xsd:any...> в схеме EPCIS также может соответствовать элементам, зависящим от поставщика/пользователя. Перемещение <xsd:any...> в схему поставщика/пользователя устраняет эту проблему, потому что ##other в этой схеме означает «совпадение с элементом, который имеет пространство имен, отличное от пространства имен поставщика/пользователя». Это не вступает в конфликт с типовыми элементами, так как форма элемента по умолчанию для типовой схемы EPCIS является unqualified (неуточненный), и, следовательно, ##other в схеме поставщика/пользователя также не соответствует типовым элементам EPCIS.

Правила для добавления атрибутов или элементов в будущие версии типовой схемы GS1 следующие:

- Типовые атрибуты могут быть добавлены к любому типу, в котором встречается точка расширения <<extension point>>. Типовые атрибуты НЕ ДОЛЖНЫ принадлежать ни одному пространству имен (т. е. ДОЛЖНЫ находиться в пустом пространстве имен) и НЕ ДОЛЖНЫ конфликтовать с любым существующим именем типового атрибута.

- Типовые элементы могут быть добавлены к любому типу, в котором встречается точка расширения <<extension point>>. Новые элементы добавляются по следующим правилам:

- Находят самый глубокий вложенный тип элемента extension (расширение).

- Заменяют объявление <xsd:any ... namespace="##local"/> на (a) новые элементы (которые НЕ ДОЛЖНЫ принадлежать ни одному пространству имен, другими словами, которые ДОЛЖНЫ находиться в пустом пространстве имен), за которыми следует (b) новый элемент extension, тип которого строится, как описано выше. В последующих версиях типовой схемы новые типовые элементы будут скорее добавлены в этот новый элемент extension (расширение)..

Примечание — Причиной того, что новые типовые атрибуты и элементы, указанные выше, не должны находиться в каком-либо пространстве имен, является то, что они должны быть совместимы с атрибутом схемы EPCIS и с формой элемента по умолчанию unqualified.

Применительно к схеме XML *настоящего стандарта* (EPCIS *версия* 1.2) для основных событий (см. 9.5) это приводит к следующему:

Типы событий, определенные в *[3]* (EPCIS *версия* 1.0), появляются в элементе <EventList> (<Список_Событий>).

Каждый из типов событий, определенных в *[4]* (EPCIS *версия* 1.1) (т. е. TransformationEvent (Событие_с_Преобразованием)), появляется внутри элемента <extension> (<расширение>) в элементе <EventList> (<Список_Событий>).

Для типов событий, определенных в *[3]* (EPCIS *версия* 1.0), новые поля, добавленные в *[4]* (EPCIS *версия* 1.1), появляются в элементе <extension> (<расширение>), следующем за полями *[3]* (EPCIS *версия* 1.0). Если в будущую версию EPCIS будут добавлены дополнительные поля, они появятся во втором элементе <extension> (<расширение>), вложенном в первый элемент <extension> (<расширение>), следующий за полями стандарта *[4]* (EPCIS *версия* 1.1). Новые поля, добавленные в настоящем стандарте и *[2]* (EPCIS *версия* 1.2) в местах, где в *[4]* (EPCIS *версия* 1.1) не было добавлено новых полей (т. е. errorDeclaration (Заявление_об_Ошибке)), появляются в элементе <extension> (<расширение>) после полей, определенных в *[3]* (EPCIS *версия* 1.0).

Для типов событий, определенных в *[4]* (EPCIS *версия* 1.1), нет элемента <extension> (<расширение>), поскольку в *[4]* (EPCIS *версия* 1.1) весь тип события является новым. Если в будущей версии стандарта EPCIS будут добавлены дополнительные поля, они появятся в элементе <extension> (<расширение>) после полей, определенных в *[4]* (EPCIS *версия* 1.1).

Расширения на уровне событий поставщика/пользователя всегда появляются непосредственно перед закрывающим тэгом события (т. е. после любых типовых полей и любого элемента <extension> (<расширение>)) и всегда находятся в непустом пространстве имен XML. Ни при каких обстоятельствах расширения поставщика/пользователя не появляются внутри элемента <extension> (<расширение>). Элемент <extension> (<расширение>) зарезервирован для полей, определенных в самом стандарте EPCIS.

Примеры см. в подразделе 9.6.

9.2 Типовой заголовок бизнес-документа

Привязки XML для модуля определения данных типов основных событий включают необязательный элемент EPCISHeader (Заголовок_EPCIS), который может использоваться отраслевыми группами для включения дополнительной информации, требуемой для обработки в рамках этой отрасли. Основная схема включает «типовой заголовок бизнес-документа» (Standard Business Document Header — SBDH), определенный в *[22]* как необходимый компонент элемента EPCISHeader (Заголовок_EPCIS). Отраслевым группам МОГУТ также потребоваться некоторые другие виды заголовка внутри элемента EPCISHeader (Заголовок_EPCIS) дополнительно к SBDH.

Схему XSD для типового заголовка бизнес-документа можно получить с веб-сайта СЕФАКТ ООН (UN/CEFACT), см. *[22]*. Эта схема включена сюда с помощью ссылки.

Если включен типовой заголовок бизнес-документа, то для элементов схемы SBDH по *[22]* ДОЛЖНЫ быть использованы указанные ниже значения, *приведенные в таблице 33*.

Таблица 33

Поле SBDH (XPath)	Значение
HeaderVersion (Версия_Заголовка)	1.0
DocumentIdentification/Standard (Идентификация_Документа/Стандарт)	EPCglobal
DocumentIdentification/TypeVersion (Идентификация_Документа/Версия_Типа)	1.0
DocumentIdentification/Type (Идентификация_Документа/Тип)	*См. таблицу 34*

Значение для поля DocumentIdentification/Type (Идентификация_Документа/Тип) ДОЛЖНО быть установлено в соответствии со следующей таблицей *34*, устанавливающей значение для этого поля, основываясь на виде документа EPCIS и контексте, в котором он используется.

Таблица 34

Тип документа и контекст	Значение для поля DocumentIdentification/Type (Идентификация_Документа/Тип)
EPCISDocument (Документ_EPCIS), используемый в любом контексте	Events (События)
EPCISMasterData (Мастер-Данные_EPCIS), используемый в любом контексте	MasterData (Мастер-Данные)
EPCISQueryDocument (Документ_Запроса _EPCIS), используемый в качестве документа запрашивающей стороны для привязки по 11.3	QueryControl-Request (Управление_Запросами_Запрос)
EPCISQueryDocument (Документ_Запроса _EPCIS), используемый в качестве документа отвечающей стороны для привязки в соответствии с 11.3	QueryControl-Response (Управление_Запросами_Ответ)
EPCISQueryDocument (Документ_Запроса _EPCIS), используемый в любой привязке XML интерфейса обратных вызовов (11.4.2 - 11.4.4)	QueryCallback (Обратный_ Вызов_по_ Запросу)
EPCISQueryDocument (Документ_Запроса _EPCIS), используемый в любом другом контексте	Query (Запрос)

Привязка протокола AS2 для интерфейса управления запросами (см. 11.3) также задает дополнительные поля типового заголовка бизнес-документа, которые должны быть представлены в экземпляре EPCISQueryDocument (Документ_Запроса_EPCIS), используемом в качестве ответного сообщения интерфейса управления запросами. Подробности см. в 11.3.

В дополнение к полям, указанным выше, типовой заголовок бизнес-документа ДОЛЖЕН включать все другие поля, которые требуются схемой SBDH, и МОЖЕТ включать дополнительные поля SBDH. Во всех случаях значения для этих полей ДОЛЖНЫ быть установлены в соответствии с *[22]*. Отраслевая группа МОЖЕТ указать дополнительные ограничения на контенты SBDH, которые будут использоваться внутри этой отраслевой группы, однако такие ограничения ДОЛЖНЫ соответствовать указанным здесь спецификациям.

9.3 Базовая схема EPCglobal

Привязка XML для модуля определения данных типов основных событий, так же как и другие привязки XML в этой спецификации, ссылаются на базовую схему EPCglobal. Эта схема воспроизводится ниже.

```
<xsd:schema targetNamespace="urn:epcglobal:xsd:1"
            xmlns:epcglobal="urn:epcglobal:xsd:1"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="unqualified"
            attributeFormDefault="unqualified"
            version="1.0">
  <xsd:annotation>
```

gost.e-ecolog.ru

```
        <xsd:documentation>
           <epcglobal:copyright>Copyright (C) 2004 Epcglobal Inc., All Rights
Reserved.</epcglobal:copyright>
      <epcglobal:disclaimer>EPCglobal Inc., его члены, должностные лица,
директора, сотрудники или агенты не несут ответственности за любой вред,
убытки, ущерб, финансовые или иные последствия, возникающие из, связанные
с, или вызванные использованием этого документа. Использование указанного
документа означает ваше явное согласие с вышеизложенным освобождением от
ответственности.</epcglobal:disclaimer>
      <epcglobal:specification>EPCglobal common components Version 1.0</
epcglobal:specification>
        </xsd:documentation>
     </xsd:annotation>
     <xsd:complexType name="Document" abstract="true">
        <xsd:annotation>
           <xsd:documentation xml:lang="en">
                Свойства документа EPCglobal для всех сообщений.
           </xsd:documentation>
        </xsd:annotation>
        <xsd:attribute name="schemaVersion" type="xsd:decimal" use="required">
           <xsd:annotation>
              <xsd:documentation xml:lang="en">
                   Версия схемы, которой соответствует экземпляр.
              </xsd:documentation>
           </xsd:annotation>
        </xsd:attribute>
        <xsd:attribute name="creationDate" type="xsd:dateTime" use="required">
           <xsd:annotation>
              <xsd:documentation xml:lang="en">
                  Дата создания сообщения. Используется для аудита и ведения журнала.
              </xsd:documentation>
           </xsd:annotation>
        </xsd:attribute>
     </xsd:complexType>
     <xsd:complexType name="EPC">
        <xsd:annotation>
           <xsd:documentation xml:lang="en">
                EPC представляет собой номер электронного кода продукции.
           </xsd:documentation>
        </xsd:annotation>
     <xsd:simpleContent>
        <xsd:extension base="xsd:string"/>
     </xsd:simpleContent>
   </xsd:complexType>
 </xsd:schema>
```

9.4 Мастер-данные в привязке XML

Как было отмечено в 6.1.1, EPCIS предусматривает четыре способа передачи мастер-данных. Эти четыре способа поддерживаются различными частями XML схемы, указанной в настоящем разделе, как описано в таблице *35*:

Таблица 35

Механизм передачи	Поддержка схемы
Запрос мастер-данных	`VocabularyElement` (Лексический_Элемент) внутри `VocabularyList` (Список_из_Лексикона), содержащийся внутри `epcisq:QueryResults` (Результаты_Запроса)

Окончание таблицы 35

Механизм передачи	Поддержка схемы
МДЭП (ILMD)	Элемент XML, содержащийся внутри элемента ILMD (МДЭП)
Заголовок документа EPCIS	VocabularyElement (Лексический_Элемент) внутри VocabularyList (Список_из_Лексикона), содержащийся внутри EPCISHeader (Заголовок_EPCIS)
Документ мастер-данных EPCIS	VocabularyElement (Лексический_Элемент) внутри VocabularyList (Список_из_Лексикона), содержащийся внутри EPCISBody (Тело_EPCIS) внутри epcismd:EPCISMasterDataDocument (Документ_Мастер-Данных_EPCIS)

Каждый атрибут мастер-данных представляет собой пару имя/значение, в которой имя как часть этой пары является составным именем, состоящим из идентификатора URI пространства имен и локального имени, а значение как другая часть пары представляет собой любой тип данных, выражаемый в формате XML. Вне зависимости от использования любого из четырех вышеуказанных механизмов передачи мастер-данных, передаваемые данные ДОЛЖНЫ всегда использовать для определенного атрибута одни и те же идентификатор URI пространства имен и локальное имя. Однако способы, при помощи которых осуществляется кодирование идентификатора URI пространства имен и локального имени в XML, отличаются в зависимости от выбранного механизма:

- для элементов МДЭП (ILMD) атрибутом мастер-данных ДОЛЖЕН являться элемент XML, именем элемента которого является составное имя с префиксом, привязанным к идентификатору URI пространства имени атрибута мастер-данных, а локальным именем составного имени является локальное имя атрибута мастер-данных. Содержанием элемента ДОЛЖНО быть значение атрибута мастер-данных;
- для механизмов, использующих VocabularyElement (Лексический_Элемент), идентификационным id атрибутом VocabularyElement (Лексический_Элемент) ДОЛЖНА быть строка, состоящая из идентификатора URI пространства имени, знака «номер» (#) и локального имени. Содержанием VocabularyElement (Лексический_Элемент) ДОЛЖНО быть значение атрибута мастер-данных.

Пример: Рассматривают атрибут мастер-данных с идентификатором URI пространства имени gost.e-ecolog.ru ***, локальным именем myAttrName и значением в виде цепочки myAttrValue. Ниже показано представление данного атрибута в разделе МДЭП (ILMD):***

```
<epcis:EPCISDocument
    xmlns:epcis="urn:epcglobal:epcis:xsd:1"
   xmlns:example="gost.e-ecolog.ru                    ...>
...
    <ObjectEvent>
...
      <ILMD>
         <example:myAttrName>myAttrValue</example:myAttrName>
...
    </ObjectEvent>
...
</epcis:EPCISDocument>
```

Представление данного атрибута в VocabularyElement (Лексический_Элемент):

```
<VocabularyElement
id="http://epcis.example.com/ns/md#myAttrName">
myAttrValue
</VocabularyElement>
```

(Знаки «перевод строки» и «пробел» добавлены с каждой из сторон myAttrValue для большей ясности, но в реальном сообщении XML они бы не присутствовали.)

НЕ РЕКОМЕНДУЕТСЯ: Связка XML для модуля определения данных типов основных событий предусматривает механизм для включения дополнительной информации в поля readPoint (место_Считывания) и bizLocation (производственное_Место_Назначения) всех типов событий посредством включения дополнительных подэлементов внутри этих полей после необходимого идентификационного подэлемента. Этот механизм был изначально задуман как средство передачи мастер-данных для идентификаторов места нахождения. Однако настоящим стандартом и [2] (EPCIS *версия* 1.2) **НЕ РЕКОМЕНДУЕТСЯ** его использование, и он НЕ ДОЛЖЕН использоваться в данных EPCIS, соответствующих настоящему стандарту и [2], а также более поздним версиям. Вместо него следует использовать один или несколько прочих механизмов передачи мастер-данных.

9.5 Схема для типов основных событий

Ниже приведена схема XML (XSD) для модуля определения данных типов основных событий. Эта схема импортирует дополнительные схемы, как показано в таблице 36.

Таблица 36

Пространство имен	Ссылка на место нахождения	Источник
urn:epcglobal:xsd:1	EPCglobal.xsd	Подраздел 9.3
gost.e-ecolog.ru StandardBusiness DocumentHeader	StandardBusinessDocumentHeader.xsd	веб-сайт СЕФАКТ ООН (UN/ CEFACT); см. 9.2

В дополнение к ограничениям, которые предполагает данная схема, значение типа `xsd:dateTime` в экземпляре документа ДОЛЖНО включать идентификатор часового пояса (либо «Z» для всемирного координированного времени, UTC, либо явный временной сдвиг относительно всемирного координированного времени, UTC).

Для любого элемента XML, который определяет значение `minOccurs="0"` типа `xsd:anyURI`, `xsd:string` или типа, производного от одного из этих двух, реализация EPCIS ДОЛЖНА обрабатывать экземпляр, имеющий пустую строку в качестве своего значения, точно тем же способом, как если бы элемент отсутствовал вовсе. То же самое верно для любого атрибута XML подобного типа, который определяет значение `use="optional"`.

Эта схема также включает привязку XML мастер-данных для модуля определения данных типов основных событий. Сегменты схемы, содержащие мастер-данные, используются для (a) возврата результатов по запросу типа `SimpleMasterDataQuery (Простой_Запрос_Мастер_Данных)` (см. 8.2.7.2), (b) формирования основной части документа мастер-данных, как определено в подразделе 9.7, и (c) для предоставления необязательного раздела мастер-данных заголовка EPCIS, который может использоваться в документе EPCIS или документе запроса EPCIS.

Элемент высокого уровня `EPCISDocument (Документ_EPCIS)`, определенный в схеме, используется конкретными привязками интерфейса EPCIS сбора данных, указанными в разделе 10. Торговые партнеры могут также по взаимному согласию дополнительно использовать документ EPCIS как средство передачи набора событий EPCIS, при необходимости дополненного соответствующими мастер-данными, в виде единого электронного документа.

Документ EPCIS МОЖЕТ включать мастер-данные в заголовке. Это позволяет создателю документа EPCIS включить мастер-данные, которые могут потребоваться получателю документа для осуществления запроса с использованием интерфейса запросов EPCIS. Не существует требований в отношении включения мастер-данных в заголовок документа EPCIS, включения в мастер-данные в заголовке мастер-данных для каждого идентификатора, используемого в основной части документа EPCIS, или ограничения мастер-данных в заголовке исключительно теми идентификаторами, которые используются в основной части документа EPCIS. Однако, если мастер-данные в заголовке все же относятся к идентификатору в основной части, они ДОЛЖНЫ представлять собой актуальные мастер-данные для этого идентификатора на момент создания документа EPCIS. Получатель документа EPCIS, в том числе система, реализующая применение интерфейса EPCIS сбора данных , может по своему усмотрению использовать или игнорировать такие мастер-данные. Мастер-данные в заголовке документа EPCIS НЕ ДОЛЖНЫ указывать значения атрибутов, противоречащие разделу МДЭП (ILMD) любого из событий, содержащихся в основной части документа EPCIS.

Схема XML (XSD) для модуля определения данных типов основных событий приводится ниже:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:epcis="urn:epcglobal:epcis:xsd:1"
xmlns:sbdh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
xmlns:epcglobal="urn:epcglobal:xsd:1" xmlns:xsd="gost.e-ecolog.ru                    "
targetNamespace="urn:epcglobal:epcis:xsd:1" elementFormDefault="unqualified"
attributeFormDefault="unqualified" version="1.2">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      <epcglobal:copyright>Copyright (C) 2006-2016 GS1 AISBL, All Rights
Reserved.</epcglobal:copyright>
      <epcglobal:disclaimer>
```

```
THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY
WARRANTY OF MERCHANTABILITY, NONINFRINGMENT, FITNESS FOR PARTICULAR PURPOSE,
OR ANY WARRANTY OTHER WISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims
all liability for any damages arising from use or misuse of this Standard,
whether special, indirect, consequential, or compensatory damages, and including
liability for infringement of any intellectual property rights, relating to use
of information in or reliance upon this document.

GS1 retains the right to make changes to this document at any time, without
notice.
GS1 makes no warranty for the use of this document and assumes no responsibility
for any errors which may appear in the document, nor does it make a commitment
to update the information contained herein.

(НАСТОЯЩИЙ ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ВКЛЮЧАЯ
ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, ПАТЕНТНОЙ ЧИСТОТЫ, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ
ЦЕЛИ, ИЛИ КАКИХ-ЛИБО ИНЫХ ГАРАНТИЙ, ВЫТЕКАЮЩИХ ИЗ НАСТОЯЩЕГО ДОКУМЕНТА.
GS1 снимает с себя всякую ответственность за любые убытки, проистекающие из
использования или ненадлежащего использования настоящего стандарта, включая
фактические, непрямые, косвенные и компенсаторные убытки, а также включая
ответственность за нарушение каких-либо прав интеллектуальной собственности,
связанное с использованием информации, содержащейся в настоящем документе, или
основанное на использовании настоящего документа.

GS1 оставляет за собой право в любое время и без уведомления вносить изменения
в настоящий документ. GS1 не предоставляет никаких гарантий в отношении
использования настоящего документа и отказывается от ответственности за какие-
либо ошибки, которые могут обнаружиться в настоящем документе, а также от
обязательства по актуализации содержащейся в нем информации).
                </epcglobal:disclaimer>
                <epcglobal:specification>EPC INFORMATION SERVICE (EPCIS) Version
    1.2</epcglobal:specification>
            </xsd:documentation>
        </xsd:annotation>
        <xsd:import namespace="urn:epcglobal:xsd:1" schemaLocation="./EPCglobal.
    xsd"/>
        <xsd:import
    namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
    schemaLocation="./StandardBusinessDocumentHeader.xsd"/>
        <!-- EPCIS CORE ELEMENTS -->
        <xsd:element name="EPCISDocument" type="epcis:EPCISDocumentType"/>
        <xsd:complexType name="EPCISDocumentType">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                document that contains a Header and a Body.
                    </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
            <xsd:extension base="epcglobal:Document">
                <xsd:sequence>
                    <xsd:element name="EPCISHeader" type="epcis:EPCISHeaderType"
    minOccurs="0"/>
                    <xsd:element name="EPCISBody" type="epcis:EPCISBodyType"/>
                    <xsd:element name="extension" type="epcis:EPCISDocumentExten-
                sionType"
    minOccurs="0"/>
```

```
                <xsd:any namespace="##other" processContents="lax" minOc-
            curs="0"
maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:anyAttribute processContents="lax"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="EPCISDocumentExtensionType">
        <xsd:sequence>
            <xsd:any namespace="##local" processContents="lax" maxOc-
        curs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <xsd:complexType name="EPCISHeaderType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            specific header(s) including the Standard Business Document Header
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element ref="sbdh:StandardBusinessDocumentHeader"/>
            <xsd:element name="extension" type="epcis:EPCISHeaderExtension-
        Type"
minOccurs="0"/>
        <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <xsd:complexType name="EPCISHeaderExtensionType">
        <xsd:sequence>
            <xsd:element name="EPCISMasterData" type="epcis:EPCISMaster-
        DataType"
minOccurs="0"/>
            <xsd:element name="extension" type="epcis:EPCISHeaderExtension-
        2Type"
minOccurs="0"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <xsd:complexType name="EPCISHeaderExtension2Type">
        <xsd:sequence>
            <xsd:any namespace="##local" processContents="lax" maxOc-
        curs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <!-- Since 1.2 -->
    <xsd:complexType name="EPCISMasterDataType">
        <xsd:sequence>
            <xsd:element name="VocabularyList" type="epcis:VocabularyList-
        Type" />
```

```
            <xsd:element name="extension" type="epcis:EPCISMasterDataExten-
        sionType"
minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="EPCISMasterDataExtensionType">
        <xsd:sequence>
            <xsd:any   namespace="##local"   processContents="lax"   maxOc-
        curs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- MasterData CORE ELEMENT TYPES -->
    <xsd:complexType name="VocabularyListType">
        <xsd:sequence>
            <xsd:element name="Vocabulary" type="epcis:VocabularyType" min-
        Occurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="VocabularyType">
        <xsd:sequence>
            <xsd:element name="VocabularyElementList"
type="epcis:VocabularyElementListType" minOccurs="0"/>
        <xsd:element name="extension" type="epcis:VocabularyExtensionType"
minOccurs="0"/>
        <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="type" type="xsd:anyURI" use="required"/>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="VocabularyElementListType">
        <xsd:sequence>
            <xsd:element  name="VocabularyElement"  type="epcis:VocabularyE-
        lementType"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <!-- Implementations SHALL treat a <children list containing zero elements
    in the same way as if the <children> element were omitted altogether.
    -->
    <xsd:complexType name="VocabularyElementType">
        <xsd:sequence>
            <xsd:element name="attribute" type="epcis:AttributeType" minOc-
        curs="0"
maxOccurs="unbounded"/>
            <xsd:element   name="children"   type="epcis:IDListType"   minOc-
        curs="0"/>
            <xsd:element  name="extension"  type="epcis:VocabularyElementEx-
        tensionType"
minOccurs="0"/>
            <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
```

```
            </xsd:sequence>
            <xsd:attribute name="id" type="xsd:anyURI" use="required"/>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <xsd:complexType name="AttributeType">
            <xsd:complexContent mixed="true">
                <xsd:extension base="xsd:anyType">
                    <xsd:attribute name="id" type="xsd:anyURI" use="required"/>
                    <xsd:anyAttribute processContents="lax"/>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>

        <xsd:complexType name="IDListType">
            <xsd:sequence>
                 <xsd:element name="id" type="xsd:anyURI" minOccurs="0" maxOc-
curs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <xsd:complexType name="VocabularyExtensionType">
            <xsd:sequence>
                <xsd:any   namespace="##local"   processContents="lax"   maxOc-
            curs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <xsd:complexType name="VocabularyElementExtensionType">
            <xsd:sequence>
                <xsd:any   namespace="##local"   processContents="lax"   maxOc-
            curs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <xsd:complexType name="EPCISBodyType">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                specific body that contains EPCIS related Events.
                    </xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                <xsd:element name="EventList" type="epcis:EventListType" minOccurs="0"/>
                <xsd:element name="extension" type="epcis:EPCISBodyExtensionType"
    minOccurs="0"/>
                <xsd:any namespace="##other" processContents="lax" minOccurs="0"
    maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>
        <xsd:complexType name="EPCISBodyExtensionType">
            <xsd:sequence>
                <xsd:any        namespace="##local"        processContents="lax"
            maxOccurs="unbounded"/>
```

```
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <!-- EPCIS CORE ELEMENT TYPES -->
        <xsd:complexType name="EventListType">
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element  name="ObjectEvent"  type="epcis:ObjectEventType"
            minOccurs="0"
    maxOccurs="unbounded"/>
                <xsd:element name="AggregationEvent" type="epcis:AggregationEve
            ntType"
    minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="QuantityEvent" type="epcis:QuantityEventType"
    minOccurs="0"
    maxOccurs="unbounded"/>
            <xsd:element name="TransactionEvent" type="epcis:TransactionEventType"
    minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="extension" type="epcis:EPCISEventListExtensionType"/>
            <xsd:any namespace="##other" processContents="lax"/>
        </xsd:choice>
        <!-- Note: the use of «unbounded" in both the xsd:choice element
                    and the enclosed xsd:element elements is, strictly speaking,
                    redundant. However, this was found to avoid problems with
                    certain XML processing tools, and so is retained here.
                -->
        </xsd:complexType>
        <!-- Modified in 1.1 -->
        <xsd:complexType name="EPCISEventListExtensionType">
            <xsd:choice>
                <xsd:element name="TransformationEvent" type="epcis:Transformat
ionEventType"/>
                <xsd:element name="extension" type="epcis:EPCISEventListExtensi
on2Type"/>
            </xsd:choice>
        </xsd:complexType>
        <!-- Since 1.1 -->
        <xsd:complexType name="EPCISEventListExtension2Type">
            <xsd:sequence>
                <xsd:any        namespace="##local"        processContents="lax"
            maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>
        <xsd:complexType name="EPCListType">
            <xsd:sequence>
                <xsd:element name="epc" type="epcglobal:EPC" minOccurs="0"
    maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
        <xsd:simpleType name="ActionType">
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="ADD"/>
                <xsd:enumeration value="OBSERVE"/>
                <xsd:enumeration value="DELETE"/>
            </xsd:restriction>
```

```
        </xsd:simpleType>
        <xsd:simpleType name="ParentIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <!-- Standard Vocabulary -->
        <xsd:simpleType name="BusinessStepIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <!-- Standard Vocabulary -->
        <xsd:simpleType name="DispositionIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <!-- User Vocabulary -->
        <xsd:simpleType name="EPCClassType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <!-- Standard Vocabulary -->
        <!-- Since 1.1 -->
        <xsd:simpleType name="UOMType">
            <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
        <!-- Since 1.1 -->
        <xsd:complexType name="QuantityElementType">
            <xsd:sequence>
                <xsd:element name="epcClass" type="epcis:EPCClassType"/>
                <xsd:sequence minOccurs="0">
                    <xsd:element name="quantity" type="xsd:decimal"/>
                    <xsd:element name="uom" type="epcis:UOMType" minOccurs="0"/>
                </xsd:sequence>
            </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="QuantityListType">
            <xsd:sequence>
                <xsd:element name="quantityElement" type="epcis:QuantityElement
            Type"
    minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>

        <!-- User Vocabulary -->
        <xsd:simpleType name="ReadPointIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <xsd:complexType name="ReadPointType">
            <xsd:sequence>
                <xsd:element name="id" type="epcis:ReadPointIDType"/>
                <xsd:element name="extension" type="epcis:ReadPointExtensionType"
    minOccurs="0"/>
                <!-- The wildcard below provides the extension mechanism described
            in Section
    9.4 -->
            <xsd:any namespace="##other" processContents="lax" minOccurs="0"
    maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="ReadPointExtensionType">
```

```
        <xsd:sequence>
            <xsd:any        namespace="##local"        processContents="lax"
        maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <!-- User Vocabulary -->
    <xsd:simpleType name="BusinessLocationIDType">
        <xsd:restriction base="xsd:anyURI"/>
    </xsd:simpleType>
    <xsd:complexType name="BusinessLocationType">
        <xsd:sequence>
            <xsd:element name="id" type="epcis:BusinessLocationIDType"/>
            <xsd:element name="extension" type="epcis:BusinessLocationExten
        sionType"
minOccurs="0"/>
            <!-- The wildcard below provides the extension mechanism described
        in Section
9.4 -->
            <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="BusinessLocationExtensionType">
        <xsd:sequence>
            <xsd:any        namespace="##local"        processContents="lax"
        maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <!-- User Vocabulary -->
    <xsd:simpleType name="BusinessTransactionIDType">
        <xsd:restriction base="xsd:anyURI"/>
    </xsd:simpleType>
    <!-- Standard Vocabulary -->
    <xsd:simpleType name="BusinessTransactionTypeIDType">
        <xsd:restriction base="xsd:anyURI"/>
    </xsd:simpleType>
    <xsd:complexType name="BusinessTransactionType">
        <xsd:simpleContent>
            <xsd:extension base="epcis:BusinessTransactionIDType">
                <xsd:attribute name="type" type="epcis:BusinessTransactionT
            ypeIDType" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="BusinessTransactionListType">
        <xsd:sequence>
            <xsd:element name="bizTransaction" type="epcis:BusinessTransact
        ionType"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- User Vocabulary -->
    <!-- Since 1.1 -->
```

```
        <xsd:simpleType name="SourceDestIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <!-- Standard Vocabulary -->
        <!-- Since 1.1 -->
        <xsd:simpleType name="SourceDestTypeIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>
        <!-- Since 1.1 -->
        <xsd:complexType name="SourceDestType">
            <xsd:simpleContent>
                <xsd:extension base="epcis:SourceDestIDType">
                    <xsd:attribute name="type" type="epcis:SourceDestTypeIDType"
use="required"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
        <xsd:complexType name="SourceListType">
            <xsd:sequence>
                    <xsd:element  name="source"  type="epcis:SourceDestType"
        maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="DestinationListType">
            <xsd:sequence>
                <xsd:element name="destination" type="epcis:SourceDestType"
maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>

        <!-- User Vocabulary -->
        <!-- Since 1.1 -->
        <xsd:simpleType name="TransformationIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>

        <!-- Since 1.1 -->
        <xsd:complexType name="ILMDType">
            <xsd:sequence>
                <xsd:element  name="extension"  type="epcis:ILMDExtensionType"
            minOccurs="0"/>
                <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>
        <xsd:complexType name="ILMDExtensionType">
            <xsd:sequence>
                <xsd:any       namespace="##local"       processContents="lax"
            maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <!-- User Vocabulary -->
        <!-- Since 1.2 -->
```

```
        <xsd:simpleType name="EventIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>

        <!-- Standard Vocabulary -->
        <!-- Since 1.2 -->
        <xsd:simpleType name="ErrorReasonIDType">
            <xsd:restriction base="xsd:anyURI"/>
        </xsd:simpleType>

        <!-- Since 1.2 -->
        <xsd:complexType name="CorrectiveEventIDsType">
            <xsd:sequence>
                <xsd:element  name="correctiveEventID"  type="epcis:EventIDType"
            minOccurs="0"
    maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>

        <!-- Since 1.2 -->
        <xsd:complexType name="ErrorDeclarationType">
            <xsd:sequence>
                <xsd:element name="declarationTime" type="xsd:dateTime"/>
                <xsd:element    name="reason"    type="epcis:ErrorReasonIDType"
            minOccurs="0"/>
                <xsd:element name="correctiveEventIDs" type="epcis:CorrectiveEv
            entIDsType"
    minOccurs="0"/>
                <xsd:element name="extension" type="epcis:ErrorDeclarationExten
            sionType"
    minOccurs="0"/>
                <xsd:any namespace="##other" processContents="lax" minOccurs="0"
    maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>

        <xsd:complexType name="ErrorDeclarationExtensionType">
            <xsd:sequence>
                <xsd:any        namespace="##local"        processContents="lax"
maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute processContents="lax"/>
        </xsd:complexType>


        <!-- items listed alphabetically by name -->
        <!-- Some element types accommodate extensibility in the manner of
            «Versioning XML Vocabularies» by David Orchard (see
            gost.e-ecolog.ru                                                    ).

            In this approach, an optional <extension> element is defined
            for each extensible element type, where an <extension> element
            may contain future elements defined in the target namespace.

            In addition to the optional <extension> element, extensible element
```

```
        types are declared with a final xsd:any wildcard to accommodate
        future elements defined by third parties (as denoted by the ##other
        namespace).

        Finally, the xsd:anyAttribute facility is used to allow arbitrary
        attributes to be added to extensible element types. -->
    <xsd:complexType name="EPCISEventType" abstract="true">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            base type for all EPCIS events.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="eventTime" type="xsd:dateTime"/>
            <xsd:element name="recordTime" type="xsd:dateTime" minOccurs="0"/>
            <xsd:element name="eventTimeZoneOffset" type="xsd:string"/>
            <xsd:element name="baseExtension" type="epcis:EPCISEventExtensi
        onType"
minOccurs="0"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="EPCISEventExtensionType">
        <xsd:sequence>
            <xsd:element name="eventID" type="epcis:EventIDType" minOccurs="0"/>
            <xsd:element name="errorDeclaration" type="epcis:ErrorDeclarati
        onType"
minOccurs="0"/>
            <xsd:element name="extension" type="epcis:EPCISEventExtension2Type"
minOccurs="0"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="EPCISEventExtension2Type">
        <xsd:sequence>
            <xsd:any        namespace="##local"        processContents="lax"
        maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="ObjectEventType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
        Object Event captures information about an event pertaining to one or
    more objects identified by EPCs.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
            <xsd:extension base="epcis:EPCISEventType">
                <xsd:sequence>
                    <xsd:element name="epcList" type="epcis:EPCListType"/>
                    <xsd:element name="action" type="epcis:ActionType"/>
```

```
                    <xsd:element name="bizStep" type="epcis:BusinessStepIDType"
minOccurs="0"/>
                    <xsd:element name="disposition" type="epcis:DispositionIDType"
minOccurs="0"/>
                <xsd:element  name="readPoint"  type="epcis:ReadPointType"
            minOccurs="0"/>
                <xsd:element name="bizLocation" type="epcis:BusinessLocationType"
minOccurs="0"/>
                    <xsd:element name="bizTransactionList"
type="epcis:BusinessTransactionListType" minOccurs="0"/>
                    <xsd:element name="extension" type="epcis:ObjectEventE
                xtensionType"
minOccurs="0"/>
                    <xsd:any   namespace="##other"   processContents="lax"
                minOccurs="0"
maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:anyAttribute processContents="lax"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- Modified in 1.1 -->
    <xsd:complexType name="ObjectEventExtensionType">
        <xsd:sequence>
            <xsd:element name="quantityList" type="epcis:QuantityListType"
    minOccurs="0"/>
            <xsd:element  name="sourceList"  type="epcis:SourceListType"
    minOccurs="0"/>
            <xsd:element name="destinationList" type="epcis:DestinationListType"
minOccurs="0"/>
            <xsd:element name="ilmd" type="epcis:ILMDType" minOccurs="0"/>
            <xsd:element name="extension" type="epcis:ObjectEventExtension2
        Type"
minOccurs="0"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <!-- Since 1.1 -->
    <xsd:complexType name="ObjectEventExtension2Type">
        <xsd:sequence>
            <xsd:any       namespace="##local"       processContents="lax"
        maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="AggregationEventType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Aggregation Event captures an event that applies to objects that
            have a physical association with one another.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
            <xsd:extension base="epcis:EPCISEventType">
                <xsd:sequence>
```

```
                    <xsd:element name="parentID" type="epcis:ParentIDType"
            minOccurs="0"/>
                    <xsd:element name="childEPCs" type="epcis:EPCListType"/>
                    <xsd:element name="action" type="epcis:ActionType"/>
                    <xsd:element name="bizStep" type="epcis:BusinessStepIDType"
minOccurs="0"/>
                    <xsd:element name="disposition" type="epcis:DispositionIDType"
minOccurs="0"/>
                    <xsd:element name="readPoint" type="epcis:ReadPointType"
            minOccurs="0"/>
                    <xsd:element name="bizLocation" type="epcis:BusinessLoca
            tionType"
minOccurs="0"/>
                    <xsd:element name="bizTransactionList"
type="epcis:BusinessTransactionListType" minOccurs="0"/>
                    <xsd:element name="extension" type="epcis:AggregationEven
            tExtensionType"
minOccurs="0"/>
                      <xsd:any namespace="##other" processContents="lax"
            minOccurs="0"
maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:anyAttribute processContents="lax"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- Modified in 1.1 -->
    <xsd:complexType name="AggregationEventExtensionType">
        <xsd:sequence>
            <xsd:element name="childQuantityList" type="epcis:QuantityListType"
minOccurs="0"/>
            <xsd:element name="sourceList" type="epcis:SourceListType"
        minOccurs="0"/>
            <xsd:element name="destinationList" type="epcis:DestinationList
        Type"
minOccurs="0"/>
            <xsd:element name="extension" type="epcis:AggregationEventExten
        sion2Type"
minOccurs="0"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <!-- Since 1.1 -->
    <xsd:complexType name="AggregationEventExtension2Type">
        <xsd:sequence>
            <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="QuantityEventType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Quantity Event captures an event that takes place with respect
        to a specified quantity of object class.
            </xsd:documentation>
```

```
        </xsd:annotation>
        <xsd:complexContent>
            <xsd:extension base="epcis:EPCISEventType">
                <xsd:sequence>
                    <xsd:element name="epcClass" type="epcis:EPCClassType"/>
                    <xsd:element name="quantity" type="xsd:int"/>
                    <xsd:element name="bizStep" type="epcis:BusinessStepIDType"
minOccurs="0"/>
                    <xsd:element name="disposition" type="epcis:DispositionIDType"
minOccurs="0"/>
                    <xsd:element name="readPoint" type="epcis:ReadPointType"
                minOccurs="0"/>
                    <xsd:element name="bizLocation" type="epcis:BusinessLo
                cationType"
minOccurs="0"/>
                    <xsd:element name="bizTransactionList"
type="epcis:BusinessTransactionListType" minOccurs="0"/>
                    <xsd:element name="extension" type="epcis:QuantityEven
                tExtensionType"
minOccurs="0"/>
                    <xsd:any   namespace="##other"   processContents="lax"
                minOccurs="0"
maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:anyAttribute processContents="lax"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="QuantityEventExtensionType">
        <xsd:sequence>
            <xsd:any       namespace="##local"       processContents="lax"
        maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <xsd:complexType name="TransactionEventType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Transaction Event describes the association or disassociation of
        physical
objects to one or more business
            transactions.
                </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
            <xsd:extension base="epcis:EPCISEventType">
                <xsd:sequence>
                    <xsd:element name="bizTransactionList"
type="epcis:BusinessTransactionListType"/>
                    <xsd:element name="parentID" type="epcis:ParentIDType"
                minOccurs="0"/>
                    <xsd:element name="epcList" type="epcis:EPCListType"/>
                    <xsd:element name="action" type="epcis:ActionType"/>
                    <xsd:element name="bizStep" type="epcis:BusinessStepIDType"
minOccurs="0"/>
```

```
                    <xsd:element name="disposition" type="epcis:DispositionIDType"
minOccurs="0"/>
                    <xsd:element name="readPoint" type="epcis:ReadPointType"
                minOccurs="0"/>
                    <xsd:element name="bizLocation" type="epcis:BusinessLo
                cationType"
minOccurs="0"/>
                    <xsd:element name="extension" type="epcis:TransactionEv
                entExtensionType"
minOccurs="0"/>
                    <xsd:any namespace="##other" processContents="lax"
                minOccurs="0"
maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:anyAttribute processContents="lax"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- Modified in 1.1 -->
    <xsd:complexType name="TransactionEventExtensionType">
        <xsd:sequence>
            <xsd:element name="quantityList" type="epcis:QuantityListType"
        minOccurs="0"/>
            <xsd:element name="sourceList" type="epcis:SourceListType"
        minOccurs="0"/>
            <xsd:element name="destinationList" type="epcis:DestinationListType"
minOccurs="0"/>
            <xsd:element name="extension" type="epcis:TransactionEventExten
        sion2Type"
minOccurs="0"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
    <!-- Since 1.1 -->
    <xsd:complexType name="TransactionEventExtension2Type">
        <xsd:sequence>
            <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>

    <!-- Since 1.1 -->
    <xsd:complexType name="TransformationEventType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
            Transformation Event captures an event in which inputs are
        consumed
            and outputs are produced
                </xsd:documentation>
        </xsd:annotation>
        <xsd:complexContent>
            <xsd:extension base="epcis:EPCISEventType">
                <xsd:sequence>
                    <xsd:element name="inputEPCList" type="epcis:EPCListType"
                minOccurs="0"/>
```

```
                    <xsd:element                    name="inputQuantityList"
                type="epcis:QuantityListType"
minOccurs="0"/>
                    <xsd:element name="outputEPCList" type="epcis:EPCListType"
                minOccurs="0"/>
                    <xsd:element                    name="outputQuantityList"
                type="epcis:QuantityListType"
minOccurs="0"/>
                    <xsd:element  name="transformationID"  type="epcis:Trans
                formationIDType"
minOccurs="0"/>
                    <xsd:element name="bizStep" type="epcis:BusinessStepIDType"
minOccurs="0"/>
                    <xsd:element name="disposition" type="epcis:DispositionIDType"
minOccurs="0"/>
                    <xsd:element name="readPoint" type="epcis:ReadPointType"
                minOccurs="0"/>
                    <xsd:element  name="bizLocation"  type="epcis:BusinessLo
                cationType"
minOccurs="0"/>
                    <xsd:element name="bizTransactionList"
type="epcis:BusinessTransactionListType" minOccurs="0"/>
                    <xsd:element name="sourceList" type="epcis:SourceListType"
                minOccurs="0"/>
                    <xsd:element  name="destinationList"  type="epcis:Destin
                ationListType"
minOccurs="0"/>
                    <xsd:element name="ilmd" type="epcis:ILMDType" minOccurs="0"/>
                    <xsd:element name="extension"
type="epcis:TransformationEventExtensionType" minOccurs="0"/>
                    <xsd:any   namespace="##other"   processContents="lax"
                minOccurs="0"
maxOccurs="unbounded"/>
                </xsd:sequence>
                <xsd:anyAttribute processContents="lax"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- Since 1.1 -->
    <xsd:complexType name="TransformationEventExtensionType">
        <xsd:sequence>
            <xsd:any   namespace="##local"   processContents="lax"   maxOc-
        curs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
    </xsd:complexType>
</xsd:schema>
```

9.6 Типы основных событий. Примеры (справочная информация)

В настоящем подразделе приведены примеры типов `EPCISDocuments (Документы_EPCIS)`, представленных в XML по *[23]*.

9.6.1 Пример 1 — События типа Object Events (События с Объектами) с идентификацией на уровне экземпляра

Пример в этом подразделе содержит два события типа `ObjectEvent (Событие_с_Объектом)`, каждое из которых содержит идентификацию на уровне экземпляра. В этом примере используются только функции из *[3]* (EPCIS *версия* 1.0) и лексикон из *[24]* (CBV *версия* 1.1). Второе событие пока-

зывает элемент расширения поставщика/пользователя на уровне события с именем myField, следуя методу для расширений поставщика/пользователя, указанному в подразделе 9.1.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcis:EPCISDocument
      xmlns:epcis="urn:epcglobal:epcis:xsd:1"
      xmlns:example="http://ns.example.com/epcis"
      xmlns:xsi="                                                    "
      creationDate="2005-07-11T11:30:47.0Z"
      schemaVersion="1.2">
   <EPCISBody>
      <EventList>
         <ObjectEvent>
            <eventTime>2005-04-03T20:33:31.116-06:00</eventTime>
            <eventTimeZoneOffset>-06:00</eventTimeZoneOffset>
            <epcList>
               <epc>urn:epc:id:sgtin:0614141.107346.2017</epc>
               <epc>urn:epc:id:sgtin:0614141.107346.2018</epc>
            </epcList>
            <action>OBSERVE</action>
            <bizStep>urn:epcglobal:cbv:bizstep:shipping</bizStep>
            <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
            <readPoint>
                 <id>urn:epc:id:sgln:0614141.07346.1234</id>
             </readPoint>
            <bizTransactionList>
       <bizTransaction
type="urn:epcglobal:cbv:btt:po">
bizTransaction>
            </bizTransactionList>
         </ObjectEvent>
         <ObjectEvent>
            <eventTime>2005-04-04T20:33:31.116-06:00</eventTime>
            <eventTimeZoneOffset>-06:00</eventTimeZoneOffset>
            <epcList>
               <epc>urn:epc:id:sgtin:0614141.107346.2018</epc>
            </epcList>
            <action>OBSERVE</action>
            <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
            <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
            <readPoint>
                 <id>urn:epc:id:sgln:0012345.11111.400</id>
            </readPoint>
            <bizLocation>
                 <id>urn:epc:id:sgln:0012345.11111.0</id>
            </bizLocation>
            <bizTransactionList>
                   <bizTransaction
type="urn:epcglobal:cbv:btt:po">
bizTransaction>
                  <bizTransaction
type="urn:epcglobal:cbv:btt:desadv">urn:epcglobal:cbv:bt:0614141073467:1152</
bizTransaction>
            </bizTransactionList>
            <example:myField>Example of a vendor/user extension</
example:myField>
         </ObjectEvent>
```

gost.e-ecolog.ru

```
      </EventList>
   </EPCISBody>
</epcis:EPCISDocument>
```

9.6.2 Пример 2 — Событие типа Object Event с идентификацией на уровне класса

Пример в этом подразделе содержит одно событие типа `ObjectEvent (Событие_с_Объектом)`, содержащее идентификацию только на уровне класса. Следует обратить внимание на то, что элемент **<epcList>** (`<список_EPC>`) все же представлен, хотя и пустой, как того требует схема XML для обеспечения обратной совместимости с *[3]* (EPCIS *версия* 1.0) . Элемент `QuantityList (Список_Количество)` наряду с другими элементами, новыми для *[4]* (EPCIS *версия* 1.1), находится в области **<extension>** (`<расширение>`), которая зарезервирована для новых функций в [4] (EPCIS *версия* 1.1) (см. подраздел 9.1). Расширение поставщика/пользователя с именем `myField` также включено.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcis:EPCISDocument
      xmlns:epcis="urn:epcglobal:epcis:xsd:1"
      xmlns:example="gost.e-ecolog.ru                    "
      xmlns:xsi="http://gost.e-ecolog.ru                          "
      creationDate="2005-07-11T11:30:47.0Z"
      schemaVersion="1.2">
   <EPCISBody>
      <EventList>
         <ObjectEvent>
     <eventTime>2013-06-08T14:58:56.591Z</eventTime>
     <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
     <epcList/>
     <action>OBSERVE</action>
     <bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
     <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
     <readPoint>
        <id>urn:epc:id:sgln:0614141.00777.0</id>
     </readPoint>
     <bizLocation>
        <id>urn:epc:id:sgln:0614141.00888.0</id>
     </bizLocation>
     <extension>
        <quantityList>
           <quantityElement>
       <epcClass>urn:epc:class:lgtin:4012345.012345.998877</epcClass>
       <quantity>200</quantity>
       <uom>KGM</uom>
       <!-- Значение: груз весом 200 кг с номером GTIN '04012345123456',
принадлежащий партии '998877'-->
           </quantityElement>
        </quantityList>
        <sourceList>
           <source
type="urn:epcglobal:cbv:sdt:possessing_party">urn:epc:id:sgln:4012345.00001.0</
source>
           <!-- Сторона - физический владелец в начальной точке перемещения в
процессе хозяйственной деятельности, например экспедитор -->
           <source
type="urn:epcglobal:cbv:sdt:location">urn:epc:id:sgln:4012345.00225.0</source>
           <!-- Физическое место нахождения начальной точки, например
распределительного центра экспедитора -->
        </sourceList>
        <destinationList>
           <destination
```

```
type="urn:epcglobal:cbv:sdt:owning_party">urn:epc:id:sgln:0614141.00001.0</
destination>
          <!-- Сторона, которой принадлежат физические объекты в конечной
точке, например предприятие розничной торговли -->
                  <destination
type="urn:epcglobal:cbv:sdt:location">urn:epc:id:sgln:0614141.00777.0</destination>
          <!-- Физическое место нахождения конечной точки, например, склада
предприятия розничной торговли -->
        </destinationList>
    </extension>
    <example:myField>Example of a vendor/user extension</example:myField>
         </ObjectEvent>
      </EventList>
   </EPCISBody>
</epcis:EPCISDocument>
```

9.6.3 Пример 3. Событие типа Aggregation Event со смешанной идентификацией

Пример в этом подразделе содержит одно событие типа `AggregationEvent` (Событие_с_Агрегацией), содержащее события-потомки, имеющие идентификацию как на уровне экземпляра, так и на уровне класса. `ChildQuantityList` (список_Количества_Потомков) находится в области **<extension>** (<расширение>), которая зарезервирована для новых функций в *[4]* (EPCIS *версия* 1.1) (см. подраздел 9.1). Расширение поставщика/пользователя с именем `myField` также включено.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcis:EPCISDocument
      xmlns:epcis="urn:epcglobal:epcis:xsd:1"
      xmlns:example="http://ns.example.com/epcis"
      xmlns:xsi="gost.e-ecolog.ru                                    "
      creationDate="2005-07-11T11:30:47.0Z"
      schemaVersion="1.2">
   <EPCISBody>
      <EventList>
         <AggregationEvent>
<eventTime>2013-06-08T14:58:56.591Z</eventTime>
<eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
<parentID>urn:epc:id:sscc:0614141.1234567890</parentID>
<childEPCs>
   <epc>urn:epc:id:sgtin:0614141.107346.2017</epc>
        <epc>urn:epc:id:sgtin:0614141.107346.2018</epc>
</childEPCs>
<action>OBSERVE</action>
<bizStep>urn:epcglobal:cbv:bizstep:receiving</bizStep>
<disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
<readPoint>
        <id>urn:epc:id:sgln:0614141.00777.0</id>
</readPoint>
<bizLocation>
         <id>urn:epc:id:sgln:0614141.00888.0</id>
</bizLocation>
<extension>
   <childQuantityList>
      <quantityElement>
<epcClass>urn:epc:idpat:sgtin:4012345.098765.*</epcClass>
<quantity>10</quantity>
<!-- Значение: 10 единиц груза с номером GTIN '04012345987652' -->
            </quantityElement>
          <quantityElement>
<epcClass>urn:epc:class:lgtin:4012345.012345.998877</epcClass>
```

```
<quantity>200.5</quantity>
<uom>KGM</uom>
<!-- Значение: груз весом 200.5 кг с номером GTIN '04012345123456',
принадлежащий партии '998877'-->
            </quantityElement>
       </childQuantityList>
      </extension>
   <example:myField>Example of a vendor/user extension</example:myField>
      </AggregationEvent>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

9.6.4 Пример 4. Событие типа Transformation Event

Пример в этом подразделе содержит одно событие типа `TransformationEvent (Событие_с_Преобразованием)`, содержащее потомков, имеющих идентификацию как на уровне экземпляра, так и на уровне класса. Мастер-данные Экземпляра/Партии (МДЭП, ILMD) также включены и описывают выходные данные преобразования. Расширение поставщика/пользователя с именем `myField` также включено. Событие целиком включено в элемент `<extension>` (`<расширение>`) для EventList (Список_Событий), который зарезервирован для новых типов событий в стандарте *[4]* (EPCIS *версия* 1.1) (см. подраздел 9.1).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcis:EPCISDocument schemaVersion="1.2" creationDate="2013-06-
04T14:59:02.099+02:00"
xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:example="
epcis">
   <EPCISBody>
      <EventList>
       <extension>
        <TransformationEvent>
           <eventTime>2013-10-31T14:58:56.591Z</eventTime>
           <eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
           <inputEPCList>
              <epc>urn:epc:id:sgtin:4012345.011122.25</epc>
              <epc>urn:epc:id:sgtin:4000001.065432.99886655</epc>
           </inputEPCList>
           <inputQuantityList>
              <quantityElement>
                 <epcClass>urn:epc:class:lgtin:4012345.011111.4444</epcClass>
                 <quantity>10</quantity>
                 <uom>KGM</uom>
              </quantityElement>
              <quantityElement>
                 <epcClass>urn:epc:class:lgtin:0614141.077777.987</epcClass>
                 <quantity>30</quantity>
                 <!-- Поскольку поле uom (единица_измерения) не задано, было
использовано 30 экземпляров предмета торговли (например, штук) с номером GTIN
'00614141777778', принадлежащих партии с номером '987'. -->
              </quantityElement>
              <quantityElement>
                 <epcClass>urn:epc:idpat:sgtin:4012345.066666.*</epcClass>
                 <quantity>220</quantity>
                 <!-- Поскольку поле uom (единица_измерения) не задано и указан
шаблон EPC, были использованы 220 экземпляров предмета торговли(например, штук)
с номером GTIN '04012345666663'. -->
              </quantityElement>
           </inputQuantityList>
```

gost.e-ecolog.ru

```
            <outputEPCList>
               <epc>urn:epc:id:sgtin:4012345.077889.25</epc>
               <epc>urn:epc:id:sgtin:4012345.077889.26</epc>
               <epc>urn:epc:id:sgtin:4012345.077889.27</epc>
               <epc>urn:epc:id:sgtin:4012345.077889.28</epc>
            </outputEPCList>
            <bizStep>urn:epcglobal:cbv:bizstep:transforming</bizStep>
            <disposition>urn:epcglobal:cbv:disp:in_progress</disposition>
            <readPoint>
                <id>urn:epc:id:sgln:4012345.00001.0</id>
            </readPoint>
            <ilmd>
                <!-- Раздел, в котором задаются мастер-данные на уровне
экземпляра/партии, относящиеся к объектам, указанным в outputEPCList.-->
                <example:bestBeforeDate>2014-12-10</example:bestBeforeDate>
                <!-- Пространство имен 'example' является просто заменителем
домена, в котором определены атрибуты мастер-данных на уровне экземпляра/
серии (например, рабочей группой GS1). Смысл: срок хранения предмета торговли с
указанными номером GTIN + номером партии - 10 декабря 2014 г. -->
                <example:batch>XYZ</example:batch>
            </ilmd>
            <example:myField>Example of a vendor/user extension</example:myField>
         </TransformationEvent>
      </extension>
    </EventList>
  </EPCISBody>
</epcis:EPCISDocument>
```

9.7 Схема для мастер-данных

Ниже приведена схема XML (XSD), определяющая документ мастер-данных EPCIS . Документ мастер-данных EPCIS может быть использован по взаимному согласию для передачи мастер-данных. Схема импортирует дополнительные схемы, как показано в таблице *37*.

Таблица 37

Пространство имен	Ссылка на место нахождения	Источник
urn:epcglobal:xsd:1	EPCglobal.xsd	подраздел 9.3
http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader	StandardBusinessDocument Header.xsd	веб-сайт СЕФАКТ ООН (UN/ CEFACT); см. подраздел 9.2
urn:epcglobal:epcis:xsd :1	EPCglobal-epcis-1_2.xsd	подраздел 9.5

В дополнение к ограничениям, налагаемым схемой, любое значение типа `xsd:dateTime` в документе экземпляра ДОЛЖНО включать спецификатор часового пояса (либо "Z" для всемирного координированного времени, UTC, либо явный временной сдвиг относительно всемирного координированного времени UTC).

Для любого элемента XML типа `xsd:anyURI` или типа `xsd:string`, который задает значение `minOccurs="0"`, реализация EPCIS ДОЛЖНА обрабатывать экземпляр, имеющий пустую строку в качестве значения, точно так же, как если бы этот элемент вообще отсутствовал.

Эта схема включает заголовок EPCIS из схемы типов основных событий, указанной в подразделе 9.5. Такой заголовок предоставляет возможность включения мастер-данных при необходимости. Однако документ мастер-данных EPCIS (документ XML, корневым элементом которого является `EPCISMasterDataDocument (Документ_Мастер-Данных_EPCIS)`, определенный в приведенной ниже схеме) НЕ ДОЛЖЕН включать необязательный элемент `EPCISMasterData (Мастер-Данные_EPCIS)` в своем заголовке EPCIS.

Ниже приводится схема XML (XSD) для мастер-данных:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:epcismd="urn:epcglobal:epcis-masterdata:xsd:1"

xmlns:sbdh="http://www.unece.org/cefact/namespaces/
StandardBusinessDocumentHeader"
   xmlns:epcglobal="urn:epcglobal:xsd:1"
   xmlns:epcis="urn:epcglobal:epcis:xsd:1"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:epcglobal:epcis-masterdata:xsd:1"
   elementFormDefault="unqualified"
   attributeFormDefault="unqualified"
   version="1.2">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      <epcglobal:copyright> Copyright (C) 2006-2016 GS1 AISBL, All Rights
Reserved.</epcglobal:copyright>
      <epcglobal:disclaimer>
THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY
WARRANTY OF MERCHANTABILITY, NONINFRINGMENT, FITNESS FOR PARTICULAR PURPOSE, OR
ANY WARRANTY OTHER WISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims all
liability for any damages arising from use or misuse of this Standard, whether
special, indirect, consequential, or compensatory damages, and including
liability for infringement of any intellectual property rights, relating to use
of information in or reliance upon this document.
GS1 retains the right to make changes to this document at any time, without
notice.
GS1 makes no warranty for the use of this document and assumes no responsibility
for any errors which may appear in the document, nor does it make a commitment
to update the information contained herein.
(НАСТОЯЩИЙ ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ» БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ВКЛЮЧАЯ
ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, ПАТЕНТНОЙ ЧИСТОТЫ, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ
ЦЕЛИ, ИЛИ КАКИХ-ЛИБО ИНЫХ ГАРАНТИЙ, ПРОИСТЕКАЮЩИХ ИЗ НАСТОЯЩЕГО ДОКУМЕНТА.
GS1 снимает с себя всякую ответственность за любые убытки, проистекающие из
использования или ненадлежащего использования настоящего стандарта, включая
фактические, непрямые, косвенные и компенсаторные убытки, а также включая
ответственность за нарушение каких-либо прав интеллектуальной собственности,
связанное с использованием информации, содержащейся в настоящем документе, или
основанное на использовании настоящего документа.
GS1 оставляет за собой право в любое время и без уведомления вносить изменения
в настоящий документ. GS1 не предоставляет никаких гарантий в отношении
использования настоящего документа и отказывается от ответственности за какие-
либо ошибки, которые могут обнаружиться в настоящем документе, а также от
обязательства по актуализации содержащейся в нем информации).
      </epcglobal:disclaimer>
      <epcglobal:specification>EPC INFORMATION SERVICE (EPCIS) Version
1.2</epcglobal:specification>
    </xsd:documentation>
  </xsd:annotation>
  <xsd:import namespace="urn:epcglobal:xsd:1" schemaLocation="./EPCglobal.xsd"/>
  <xsd:import

namespace="http://gost.e-ecolog.ru
StandardBusinessDocumentHeader"
     schemaLocation="./StandardBusinessDocumentHeader.xsd"/>
  <xsd:import
```

```
    namespace="urn:epcglobal:epcis:xsd:1"
    schemaLocation="./EPCglobal-epcis-1_2.xsd"/>

  <!-- MasterData CORE ELEMENTS -->
  <xsd:element name="EPCISMasterDataDocument"
type="epcismd:EPCISMasterDataDocumentType"/>
  <xsd:complexType name="EPCISMasterDataDocumentType">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
      MasterData document that contains a Header and a Body.
            </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="epcglobal:Document">
        <xsd:sequence>
          <xsd:element name="EPCISHeader" type="epcis:EPCISHeaderType"
minOccurs="0"/>
          <xsd:element name="EPCISBody" type="epcismd:EPCISMasterDataBodyType"/>
          <xsd:element name="extension"
type="epcismd:EPCISMasterDataDocumentExtensionType" minOccurs="0"/>
          <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="EPCISMasterDataBodyType">
    <xsd:annotation>
      <xsd:documentation xml:lang="en">
      MasterData specific body that contains Vocabularies.
            </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="VocabularyList" type="epcis:VocabularyListType"
minOccurs="0"/>
      <xsd:element name="extension" type="epcismd:EPCISMasterDataBodyExtensionType"
minOccurs="0"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="EPCISMasterDataDocumentExtensionType">
    <xsd:sequence>
      <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="EPCISMasterDataHeaderExtensionType">
    <xsd:sequence>
      <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
    </xsd:sequence>
```

```
    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="EPCISMasterDataBodyExtensionType">
    <xsd:sequence>
      <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

</xsd:schema>
```

9.8 Мастер-данные. Пример (справочная информация)

Ниже приводится пример элемента EPCISMasterDataDocument (Документ_Мастер-Данных_EPCIS), содержащего мастер-данные для лексиконов BusinessLocation (Производственное_Место_Назначения) и ReadPoint (место_Считывания), представленных в XML [23]:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <epcismd:EPCISMasterDataDocument
    xmlns:epcismd="urn:epcglobal:epcis-masterdata:xsd:1"
    xmlns:xsi="gost.e-ecolog.ru                                    "
    schemaVersion="1.0"
    creationDate="2005-07-11T11:30:47.0Z">
   <EPCISBody>
     <VocabularyList>
       <Vocabulary type="urn:epcglobal:epcis:vtype:BusinessLocation">
         <VocabularyElementList>
           <VocabularyElement id="urn:epc:id:sgln:0037000.00729.0">
             <attribute
id="gost.e-ecolog.ru                                                    >
             <attribute id="gost.e-ecolog.ru
gost.e-ecolog.ru            >
             <attribute id="gost.e-ecolog.ru                                    ">
               <example:Address xmlns:example="gost.e-ecolog.ru                          ">
                 <Street>100 Nowhere Street</Street>
                 <City>Fancy</City>
                 <State>DC</State>
                 <Zip>99999</Zip>
               </example:Address>
             </attribute>
             <children>
               <id>urn:epc:id:sgln:0037000.00729.8201</id>
               <id>urn:epc:id:sgln:0037000.00729.8202</id>
               <id>urn:epc:id:sgln:0037000.00729.8203</id>
             </children>
           </VocabularyElement>
           <VocabularyElement id="urn:epc:id:sgln:0037000.00729.8201">
             <attribute id="urn:epcglobal:cbv:mda:sst">201</attribute>
           </VocabularyElement>
           <VocabularyElement id="urn:epc:id:sgln:0037000.00729.8202">
             <attribute id="urn:epcglobal:cbv:mda:sst">202</attribute>
             <children>
               <id>urn:epc:id:sgln:0037000.00729.8203</id>
             </children>
           </VocabularyElement>
           <VocabularyElement id="urn:epc:id:sgln:0037000.00729.8203">
```

```
            <attribute id="urn:epcglobal:cbv:mda:sst">202</attribute>
            <attribute id="urn:epcglobal:cbv:mda:ssa">402</attribute>
          </VocabularyElement>
        </VocabularyElementList>
      </Vocabulary>
      <Vocabulary type="urn:epcglobal:epcis:vtype:ReadPoint">
        <VocabularyElementList>
          <VocabularyElement id="urn:epc:id:sgln:0037000.00729.8201">
            <attribute id="urn:epcglobal:cbv:mda:site">0037000007296</attribute>
            <attribute id="urn:epcglobal:cbv:mda:sst">201</attribute>
          </VocabularyElement>
          <VocabularyElement id="urn:epc:id:sgln:0037000.00729.8202">
            <attribute id="urn:epcglobal:cbv:mda:site">0037000007296</attribute>
            <attribute id="urn:epcglobal:cbv:mda:sst">202</attribute>
          </VocabularyElement>
          <VocabularyElement id="urn:epc:id:sgln:0037000.00729.8203">
            <attribute id="urn:epcglobal:cbv:mda:site">0037000007296</attribute>
            <attribute id="urn:epcglobal:cbv:mda:sst">203</attribute>
          </VocabularyElement>
        </VocabularyElementList>
      </Vocabulary>
    </VocabularyList>
  </EPCISBody>
</epcismd:EPCISMasterDataDocument>
```

10 Привязки для модуля основных операций сбора данных

Данный раздел определяет привязки для модуля основных операций сбора данных (Core Capture Operations Module). Все привязки, указанные здесь, основываются на представлении событий XML, определенных в подразделе 9.5. Как указано ниже, реализация EPCIS МОЖЕТ обеспечить поддержку одной или более привязок модуля основных операций сбора данных.

10.1 Привязка очереди сообщений

В данном подразделе определяется привязка модуля основных операций сбора данных к системе очереди сообщений, как это обычно реализуется на крупных предприятиях. Система очереди сообщений определена для данного подраздела как любая система, которая позволяет одному приложению отправлять сообщение XML другому приложению. Системы очереди сообщений обычно поддерживают как доставку сообщения типа точка-точка, так и доставку сообщений типа публикации/подписки. Системы очереди сообщений часто включают в себя функции гарантированной надежной доставки, а также другие гарантии качества обслуживания (quality-of-service — QoS).

Поскольку общепринятой типовой системы очереди сообщений не существует, эта спецификация предназначена для применения к любой такой системе. Поэтому многие детали реализации неизбежно выходят за рамки данной спецификации. Такие детали включают в себя систему очереди сообщений, адресацию, протоколы, использование QoS или других параметров конкретной системы и так далее.

Реализация EPCIS МОЖЕТ обеспечить привязку очереди сообщений к модулю основных операций сбора данных следующим образом. Для этой привязки «клиентом сбора данных» является приложение EPCIS по сбору данных (EPCISCapture Application), которое намеревается доставить Событие EPCIS через интерфейс EPCIS сбора данных (EPCIS Capture Interface), а «сервер сбора данных» — это репозиторий EPCIS (EPCIS Repository) или приложение EPCIS доступа (EPCIS Accessing Application), которое получает событие от клиента сбора данных.

Сервер сбора данных ДОЛЖЕН предоставить одну или несколько конечных точек очереди сообщений, через которые клиент сбора данных может доставить одно или несколько событий EPCIS. Каждая конечная точка очереди сообщений МОЖЕТ быть очередью типа точка-точка, заголовком публикации/подписки или каким-либо другим соответствующим адресуемым каналом, предоставляемым системой очереди сообщений. Подробные сведения выходят за рамки данной спецификации.

Клиент сбора данных ДОЛЖЕН выполнить операцию capture (сбор_данных), описанную в 8.1.2, с помощью доставки сообщения в конечную точку, предоставленную сервером сбора данных. Сообщение ДОЛЖНО быть одним из следующих:

- документ XML, корневой элемент которого соответствует элементу EPCISDocument (Документ_EPCIS), как определено схемой подраздела 9.5; или
- документ XML, корневой элемент которого соответствует элементу EPCISQueryDocument (Документ_Запроса_EPCIS), как определено схемой подраздела 11.1, где элемент, непосредственно вложенный в элемент EPCISBody (Тело_EPCIS), является элементом QueryResults (Результаты_Запроса), а элемент resultsBody (тело_Результата) внутри элемента QueryResults (Результаты_Запроса) содержит элемент EventList (Список_Событий).

Реализация интерфейса сбора данных ДОЛЖНА принять форму элемента EPCISDocument (Документ_EPCIS) и форму элемента EPCISQueryDocument (Документ_Запроса_EPCIS). Реализация интерфейса сбора данных НЕ ДОЛЖНА принимать документы, которые не имеют силы, как это указано выше. Успешное принятие этого сообщения сервером ДОЛЖНО представлять собой фиксирование всех событий EPCIS, включенных в сообщение.

Системы очереди сообщений различаются по своей способности предоставлять положительные и отрицательные подтверждения отправителям сообщений. Если в системе очереди сообщений имеется функция положительного подтверждения, положительное подтверждение МОЖЕТ использоваться для указания успешного фиксирования сервером сбора данных. Если в системе очереди сообщений имеется функция отрицательного подтверждения, отрицательное подтверждение МОЖЕТ использоваться для указания на отказ выполнить операцию сбора данных. Отказ может быть вызван недействительным документом, сбоем авторизации, как описано в 8.1.1, или по какой-либо другой причине. Конкретные обстоятельства, при которых указывается положительное или отрицательное подтверждение, зависят от реализации. Тем не менее все реализации ДОЛЖНЫ либо принимать все события в сообщении, либо отклонять все события.

10.2 Привязка к протоколу HTTP

В данном подразделе определяется привязка модуля основных операций сбора данных к протоколу HTTP *[25]*.

Реализация EPCIS МОЖЕТ обеспечить HTTP привязку к модулю основных операций сбора данных следующим образом. Для этой привязки «клиентом сбора данных» является приложение EPCIS сбора данных, которое хочет доставить событие EPCIS через интерфейс EPCIS сбора данных, а «сервер сбора данных» — это репозиторий EPCIS или приложение EPCIS по организации доступа, которое получает событие от клиента сбора данных.

Сервер сбора данных ДОЛЖЕН предоставить URL адрес протокола HTTP, через который клиент сбора данных может доставить одно или несколько событий EPCIS.

Клиент сбора данных ДОЛЖЕН выполнить операцию capture (сбор_данных), описанную в 8.1.2, с помощью вызова операции POST протокола HTTP по URL адресу, предоставленному сервером сбора данных. Содержание сообщения ДОЛЖНО быть одним из следующих:

- документ XML, корневой элемент которого соответствует элементу EPCISDocument (Документ_EPCIS), как определено схемой подраздела 9.5, или
- документ XML, корневой элемент которого соответствует элементу EPCISQueryDocument (Документ_Запроса_EPCIS), как определено схемой подраздела 11.1, где элемент, непосредственно вложенный в элемент EPCISBody (Тело_EPCIS), является элементом QueryResults (Результаты_Запроса), а элемент resultsBody (тело_Результата) внутри элемента QueryResults (Результаты_Запроса) содержит элемент EventList (Список_Событий).

Реализация интерфейса сбора данных ДОЛЖНА принять форму элемента EPCISDocument (Документ_EPCIS) и форму элемента EPCISQueryDocument (Документ_Запроса_EPCIS). Реализация интерфейса сбора данных НЕ ДОЛЖНА принимать документы, которые не имеют силы, как это указано выше. Успешное принятие этого сообщения сервером ДОЛЖНО представлять собой фиксирование всех событий EPCIS, включенных в сообщение.

Кодовые значения состояния, возвращаемые сервером сбора данных, ДОЛЖНЫ соответствовать [25, раздел 10]. В частности, сервер сбора данных ДОЛЖЕН возвращать кодовое значение состояния 200, чтобы показать успешное завершение операции сбора данных, а любое кодовое значение состояния: 3xx, 4xx или 5xx — ДОЛЖНО показывать, что операция сбора данных не была успешно завершена. Обстоятельства возврата соответствующих кодовых значений состояния определяются конкретным применением. Однако любое применение ДОЛЖНО принять или отвергнуть все события в сообщении.

11 Привязки для модуля основных операций с запросами

В данном разделе определяются привязки для модуля основных операций с запросами (Core Query Operations Module), как показано *в таблице 38*:

Таблица 38

Интерфейс	Привязка	Структурный элемент
Интерфейс управления запросами (Query Control Interface)	SOAP через HTTP (WSDL)	Подраздел 11.2
	XML через AS2	Подраздел 11.3
Интерфейс обратных вызовов по запросам (Query Callback Interface)	XML через HTTP	Пункт 11.4.2
	XML через HTTP+TLS (HTTPS)	Пункт 11.4.3
	XML через AS2	Пункт 11.4.4

Все эти привязки используют общий синтаксис XML, определенный в подразделе 11.1. Схема XML имеет следующие составные части:

- элементы XML для аргумента и возвращаемой сигнатуры каждого метода в интерфейсе управления запросами, как определено в 8.2.5;
- типы XML для каждого типа данных, используемого в этих аргументах, и возвращаемых сигнатур;
- элементы XML для каждого исключения, определенного в 8.2.6;
- элементы XML для интерфейса обратных вызовов по запросам, как определено в 8.2.8 (это, на самом деле, просто подмножество предыдущих трех пунктов);
- элемент `EPCISQueryDocument (Документ_Запроса_EPCIS)`, который используется привязками в качестве «конверта», базовая технология которых не предоставляет собственный конверт или механизм заголовка (а именно всеми привязками, за исключением привязки SOAP). Привязка протокола AS2 использует этот элемент для того, чтобы предоставить заголовок для соответствия запросам и ответам. Элемент `EPCISQueryDocument (Документ_Запроса_EPCIS)` имеет общий тип `EPCISHeader (Заголовок_EPCIS)`, определенный в подразделе 9.5. Каждая привязка определяет собственные правила использования этого заголовка, если это применимо.

11.1 Схема XML для модуля основных операций с запросами

Следующая схема определяет XML представления типов данных, запросов, ответов и исключений, используемых интерфейсом EPCIS управления запросами и интерфейсом EPCIS обратных вызовов по запросам в модуле основных операций с запросами. Эта схема включена в виде ссылки во все привязки для этих двух интерфейсов, указанных в настоящем разделе 11. Эта схема ДОЛЖНА использоваться любой новой привязкой в рамках модуля основных операций с запросами, которая использует XML как основной формат сообщения.

Тип `QueryParam (Параметр_Запроса)`, определенный в нижеприведенной схеме, используется для представления параметра запроса в методах `poll (опрос)` и `subscribe (подписка)` интерфейса запросов, определенных в 8.2.5. Параметр запроса состоит из имени и значения. Для значения схема XML задает `xsd:anyType`, таким образом может быть представлено значение параметра любого типа. При создании экземпляра документа фактическое значение ДОЛЖНО соответствовать типу, подходящему для параметра запроса, как определено в таблице *39*:

Таблица 39

Тип параметра	XML-тип для элемента `value (значение)`
Int (Целочисленное значение)	`xsd:integer`
Float (Число с плавающей точкой)	`xsd:double`
Time (Время)	`xsd:dateTime`
String (Строка)	`xsd:string`
List of String (Список строк)	`epcisq:ArrayOfString`
Void (Недействительное значение)	`epcisq:VoidHolder`

В частности, таблица *39* ДОЛЖНА использоваться для сопоставления типов параметров, указанных в 8.2.7 для предопределенных запросов, и соответствующих типов XML.

Как указано в *[19]*, каждый элемент <value> (<значение>), задающий значение параметра запроса в документе экземпляра, МОЖЕТ включать атрибут xsi:type. Следующие правила определяют, каким образом обрабатываются значения параметра запроса:

- если элемент <value> (<значение>) не включает атрибут xsi:type, то в том случае, если указанное значение имеет недопустимый синтаксис для типа, требуемого параметром запроса, метод subscribe (подписка) или метод poll (опрос) интерфейса управления запросами ДОЛЖЕН ответить с помощью исключения QueryParameterException (Исключение_по_Параметру_Запроса);
- если элемент <value> (<значение>) включает атрибут xsi:type, то применяются следующие правила:

1) если тело элемента <value> (<значение>) имеет недопустимый синтаксис для типа, заданного атрибутом xsi:type, то запрос EPCISQueryDocument (Документ_Запроса_EPCIS) или запрос протокола SOAP МОЖЕТ быть отклонен синтаксическим анализатором XML данной реализации,

2) если значение атрибута xsi:type не является корректным типом для параметра запроса, как указано во второй колонке приведенной выше таблицы, то метод subscribe (подписка) или метод poll (опрос) интерфейса управления запросами МОЖЕТ применить исключение QueryParameterException (Исключение_по_Параметру_Запроса), даже если тело элемента <value> (<значение>) имеет допустимый синтаксис для типа, требуемого параметром запроса;

3) если тело элемента <value> (<значение>) имеет недопустимый синтаксис для типа, требуемого параметром запроса, то метод интерфейса управления запросами subscribe (подписка) или poll (опрос) ДОЛЖЕН применить исключение QueryParameterException (Исключение_по_Параметру_Запроса) в том случае, если запрос EPCISQueryDocument (Документ_Запроса_EPCIS) или запрос протокола SOAP не был отклонен синтаксическим анализатором XML данной реализации в соответствии с вышеприведенным правилом.

Указанная схема импортирует дополнительные схемы, как показано в таблице *40*:

Таблица 40

Пространство имен	Ссылка на место нахождения	Источник
urn:epcglobal:xsd:1	EPCglobal.xsd	Подраздел 9.3
gost.e-ecolog.ru namespaces/StandardBusiness DocumentHeader	StandardBusinessDocumentHeader.xsd	Веб-сайт СЕФАКТ ООН (UN/ CEFACT); см. подраздел 9.2
urn:epcglobal:epcis:xsd:1	EPCglobal-epcis-1_0.xsd	Подраздел 9.5
urn:epcglobal:epcis masterdata: xsd:1	EPCglobal-epcis-masterdata-1_0.xsd	Подраздел 9.7

В дополнение к ограничениям, налагаемым схемой, любое значение типа xsd:dateTime в документе экземпляра ДОЛЖНО включать спецификатор часового пояса (либо "Z" для всемирного координированного времени UTC, либо явный временной сдвиг относительно всемирного координированного времени UTC).

Для любого элемента XML типа xsd:anyURI или xsd:string, который задает значение minOccurs="0", реализация EPCIS ДОЛЖНА обрабатывать экземпляр, имеющий пустую строку в качестве значения, точно так же, как если бы этот элемент вообще отсутствовал.

Схема XML (XSD) для модуля основных операций с запросами приведена ниже:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="urn:epcglobal:epcis-query:xsd:1"
   xmlns:epcis="urn:epcglobal:epcis:xsd:1"
   xmlns:epcismd="urn:epcglobal:epcis-masterdata:xsd:1"
   xmlns:epcisq="urn:epcglobal:epcis-query:xsd:1"
   xmlns:epcglobal="urn:epcglobal:xsd:1"
   xmlns:xsd="http://gost.e-ecolog.ru                    "
   elementFormDefault="unqualified"
```

```
    attributeFormDefault="unqualified"
    version="1.2">

  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      <epcglobal:copyright>
        Copyright (C) 2006-2016 GS1 AISBL, All Rights Reserved
      </epcglobal:copyright>
      <epcglobal:disclaimer>
THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY
WARRANTY OF MERCHANTABILITY, NONINFRINGMENT, FITNESS FOR PARTICULAR PURPOSE, OR ANY
WARRANTY OTHER WISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims all liability
for any damages arising from use or misuse of this Standard, whether special,
indirect, consequential, or compensatory damages, and including liability for
infringement of any intellectual property rights, relating to use of information
in or reliance upon this document.
GS1 retains the right to make changes to this document at any time, without notice.
GS1 makes no warranty for the use of this document and assumes no responsibility
for any errors which may appear in the document, nor does it make a commitment to
update the information contained herein.
(НАСТОЯЩИЙ ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ» БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ВКЛЮЧАЯ
ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, ПАТЕНТНОЙ ЧИСТОТЫ, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ,
ИЛИ КАКИХ-ЛИБО ИНЫХ ГАРАНТИЙ, ПРОИСТЕКАЮЩИХ ИЗ НАСТОЯЩЕГО ДОКУМЕНТА. GS1 снимает
с себя всякую ответственность за любые убытки, проистекающие из использования или
ненадлежащего использования настоящего стандарта, включая фактические, непрямые,
косвенные и компенсаторные убытки, а также включая ответственность за нарушение
каких-либо прав интеллектуальной собственности, связанное с использованием
информации, содержащейся в настоящем документе, или основанное на использовании
настоящего документа.

GS1 оставляет за собой право в любое время и без уведомления вносить изменения в
настоящий документ. GS1 не предоставляет никаких гарантий в отношении использования
настоящего документа и отказывается от ответственности за какие-либо ошибки,
которые могут обнаружиться в настоящем документе, а также от обязательства по
актуализации содержащейся в нем информации).
      </epcglobal:disclaimer>
      <epcglobal:specification>
         EPCIS Query 1.2
      </epcglobal:specification>
    </xsd:documentation>
  </xsd:annotation>


  <xsd:import namespace="urn:epcglobal:xsd:1" schemaLocation="./EPCglobal.xsd"/>
  <xsd:import namespace="urn:epcglobal:epcis:xsd:1" schemaLocation="./
EPCglobalepcis-1_2.xsd"/>

  <xsd:element name="EPCISQueryDocument" type="epcisq:EPCISQueryDocumentType"/>
  <xsd:complexType name="EPCISQueryDocumentType">
    <xsd:complexContent>
      <xsd:extension base="epcglobal:Document">
        <xsd:sequence>
          <xsd:element name="EPCISHeader" type="epcis:EPCISHeaderType"
minOccurs="0"/>
          <xsd:element name="EPCISBody" type="epcisq:EPCISQueryBodyType"/>
          <xsd:element name="extension"
```

```
type="epcisq:EPCISQueryDocumentExtensionType" minOccurs="0"/>
          <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute processContents="lax"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

     <xsd:complexType name="EPCISQueryDocumentExtensionType">
           <xsd:sequence>
                 <xsd:any namespace="##local" processContents="lax"
maxOccurs="unbounded"/>
           </xsd:sequence>
           <xsd:anyAttribute processContents="lax"/>
     </xsd:complexType>
     <xsd:complexType name="EPCISQueryBodyType">
        <xsd:choice>
           <xsd:element ref="epcisq:GetQueryNames"/>
           <xsd:element ref="epcisq:GetQueryNamesResult"/>
           <xsd:element ref="epcisq:Subscribe"/>
           <xsd:element ref="epcisq:SubscribeResult"/>
           <xsd:element ref="epcisq:Unsubscribe"/>
           <xsd:element ref="epcisq:UnsubscribeResult"/>
           <xsd:element ref="epcisq:GetSubscriptionIDs"/>
           <xsd:element ref="epcisq:GetSubscriptionIDsResult"/>
           <xsd:element ref="epcisq:Poll"/>
           <xsd:element ref="epcisq:GetStandardVersion"/>
           <xsd:element ref="epcisq:GetStandardVersionResult"/>
           <xsd:element ref="epcisq:GetVendorVersion"/>
           <xsd:element ref="epcisq:GetVendorVersionResult"/>
           <xsd:element ref="epcisq:DuplicateNameException"/>
         <!-- queryValidationException unimplemented in EPCIS 1.0
           <xsd:element ref="epcisq:QueryValidationException"/>
         -->
           <xsd:element ref="epcisq:InvalidURIException"/>
           <xsd:element ref="epcisq:NoSuchNameException"/>
           <xsd:element ref="epcisq:NoSuchSubscriptionException"/>
           <xsd:element ref="epcisq:DuplicateSubscriptionException"/>
           <xsd:element ref="epcisq:QueryParameterException"/>
           <xsd:element ref="epcisq:QueryTooLargeException"/>
           <xsd:element ref="epcisq:QueryTooComplexException"/>
           <xsd:element ref="epcisq:SubscriptionControlsException"/>
           <xsd:element ref="epcisq:SubscribeNotPermittedException"/>
           <xsd:element ref="epcisq:SecurityException"/>
           <xsd:element ref="epcisq:ValidationException"/>
           <xsd:element ref="epcisq:ImplementationException"/>
           <xsd:element ref="epcisq:QueryResults"/>
        </xsd:choice>
     </xsd:complexType>

     <!-- EPCISSERVICE MESSAGE WRAPPERS -->

     <xsd:element name="GetQueryNames" type="epcisq:EmptyParms"/>
     <xsd:element name="GetQueryNamesResult" type="epcisq:ArrayOfString"/>
```

```
      <xsd:element name="Subscribe" type="epcisq:Subscribe"/>
      <xsd:complexType name="Subscribe">
         <xsd:sequence>
            <xsd:element name="queryName" type="xsd:string"/>
            <xsd:element name="params" type="epcisq:QueryParams"/>
            <xsd:element name="dest" type="xsd:anyURI"/>
            <xsd:element name="controls" type="epcisq:SubscriptionControls"/>
            <xsd:element name="subscriptionID" type="xsd:string"/>
         </xsd:sequence>
      </xsd:complexType>
   <xsd:element name="SubscribeResult" type="epcisq:VoidHolder"/>

   <xsd:element name="Unsubscribe" type="epcisq:Unsubscribe"/>
   <xsd:complexType name="Unsubscribe">
      <xsd:sequence>
   <xsd:element name="subscriptionID" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:element name="UnsubscribeResult" type="epcisq:VoidHolder"/>

   <xsd:element name="GetSubscriptionIDs" type="epcisq:GetSubscriptionIDs"/>
   <xsd:complexType name="GetSubscriptionIDs">
      <xsd:sequence>
         <xsd:element name="queryName" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:element name="GetSubscriptionIDsResult" type="epcisq:ArrayOfString"/>

   <xsd:element name="Poll" type="epcisq:Poll"/>
   <xsd:complexType name="Poll">
      <xsd:sequence>
         <xsd:element name="queryName" type="xsd:string"/>
         <xsd:element name="params" type="epcisq:QueryParams"/>
      </xsd:sequence>
   </xsd:complexType>
   <!-- The response from a Poll method is the QueryResults element, defined below.
        The QueryResults element is also used to deliver standing query results
        through the Query Callback Interface -->

   <xsd:element name="GetStandardVersion" type="epcisq:EmptyParms"/>
   <xsd:element name="GetStandardVersionResult" type="xsd:string"/>

   <xsd:element name="GetVendorVersion" type="epcisq:EmptyParms"/>
   <xsd:element name="GetVendorVersionResult" type="xsd:string"/>

   <xsd:element name="VoidHolder" type="epcisq:VoidHolder"/>
   <xsd:complexType name="VoidHolder">
<xsd:sequence>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="EmptyParms"/>

   <xsd:complexType name="ArrayOfString">
      <xsd:sequence>
         <xsd:element name="string" type="xsd:string" minOccurs="0"
```

```
maxOccurs="unbounded"/>
      </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="SubscriptionControls">
     <xsd:sequence>
       <xsd:element name="schedule" type="epcisq:QuerySchedule" minOccurs="0"/>
       <xsd:element name="trigger" type="xsd:anyURI" minOccurs="0"/>
       <xsd:element name="initialRecordTime" type="xsd:dateTime" minOccurs="0"/>
       <xsd:element name="reportIfEmpty" type="xsd:boolean"/>
       <xsd:element name="extension" type="epcisq:SubscriptionControlsExtensionType"
minOccurs="0"/>
       <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
     </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="SubscriptionControlsExtensionType">
     <xsd:sequence>
       <xsd:any namespace="##local" processContents="lax"
maxOccurs="unbounded"/>
     </xsd:sequence>
     <xsd:anyAttribute processContents="lax"/>
   </xsd:complexType>

   <xsd:complexType name="QuerySchedule">
     <xsd:sequence>
        <xsd:element name="second" type="xsd:string" minOccurs="0"/>
        <xsd:element name="minute" type="xsd:string" minOccurs="0"/>
        <xsd:element name="hour" type="xsd:string" minOccurs="0"/>
        <xsd:element name="dayOfMonth" type="xsd:string" minOccurs="0"/>
        <xsd:element name="month" type="xsd:string" minOccurs="0"/>
        <xsd:element name="dayOfWeek" type="xsd:string" minOccurs="0"/>
        <xsd:element name="extension" type="epcisq:QueryScheduleExtensionType"
minOccurs="0"/>
  <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="QueryScheduleExtensionType">
    <xsd:sequence>
      <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="QueryParams">
    <xsd:sequence>
      <xsd:element name="param" type="epcisq:QueryParam" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

  <xsd:complexType name="QueryParam">
    <xsd:sequence>
```

```
      <xsd:element name="name" type="xsd:string"/>
      <!-- See note in EPCIS spec text regarding the value for this element -->
      <xsd:element name="value" type="xsd:anyType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="QueryResults" type="epcisq:QueryResults"/>
  <xsd:complexType name="QueryResults">
    <xsd:sequence>
      <xsd:element name="queryName" type="xsd:string"/>
      <xsd:element name="subscriptionID" type="xsd:string" minOccurs="0"/>
      <xsd:element name="resultsBody" type="epcisq:QueryResultsBody"/>
      <xsd:element name="extension" type="epcisq:QueryResultsExtensionType"
minOccurs="0"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="QueryResultsExtensionType">
    <xsd:sequence>
      <xsd:any namespace="##local" processContents="lax" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="QueryResultsBody">
    <xsd:choice>
      <xsd:element name="EventList" type="epcis:EventListType"/>
      <xsd:element name="VocabularyList" type="epcis:VocabularyListType"/>
    </xsd:choice>
  </xsd:complexType>

  <!-- EPCIS EXCEPTIONS -->

  <xsd:element name="EPCISException" type="epcisq:EPCISException"/>
  <xsd:complexType name="EPCISException">
    <xsd:sequence>
      <xsd:element name="reason" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="DuplicateNameException" type="epcisq:DuplicateNameExcepti
on"/>
  <xsd:complexType name="DuplicateNameException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <!-- QueryValidationException not implemented in EPCIS 1.0
  <xsd:element name="QueryValidationException"
type="epcisq:QueryValidationException"/>
  <xsd:complexType name="QueryValidationException">
```

```
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  -->

  <xsd:element name="InvalidURIException" type="epcisq:InvalidURIException"/>
  <xsd:complexType name="InvalidURIException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="NoSuchNameException" type="epcisq:NoSuchNameException"/>
  <xsd:complexType name="NoSuchNameException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="NoSuchSubscriptionException"
type="epcisq:NoSuchSubscriptionException"/>
  <xsd:complexType name="NoSuchSubscriptionException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="DuplicateSubscriptionException"
type="epcisq:DuplicateSubscriptionException"/>
  <xsd:complexType name="DuplicateSubscriptionException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:element name="QueryParameterException"
type="epcisq:QueryParameterException"/>
  <xsd:complexType name="QueryParameterException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

```
  <xsd:element name="QueryTooLargeException" type="epcisq:QueryTooLargeException"/>
  <xsd:complexType name="QueryTooLargeException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence>
          <xsd:element name="queryName" type="xsd:string" minOccurs="0"/>
          <xsd:element name="subscriptionID" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:element name="QueryTooComplexException"
type="epcisq:QueryTooComplexException"/>
..<xsd:complexType name="QueryTooComplexException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:element name="SubscriptionControlsException"
type="epcisq:SubscriptionControlsException"/>
..<xsd:complexType name="SubscriptionControlsException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:element name="SubscribeNotPermittedException"
type="epcisq:SubscribeNotPermittedException"/>
..<xsd:complexType name="SubscribeNotPermittedException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:element name="SecurityException" type="epcisq:SecurityException"/>
..<xsd:complexType name="SecurityException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:element name="ValidationException" type="epcisq:ValidationException"/>
..<xsd:complexType name="ValidationException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence/>
```

```
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:element name="ImplementationException"
               type="epcisq:ImplementationException"/>
..<xsd:complexType name="ImplementationException">
    <xsd:complexContent>
      <xsd:extension base="epcisq:EPCISException">
        <xsd:sequence>
<xsd:element name="severity"
             type="epcisq:ImplementationExceptionSeverity"/>
          <xsd:element name="queryName" type="xsd:string" minOccurs="0"/>
          <xsd:element name="subscriptionID" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
..</xsd:complexType>

..<xsd:simpleType name="ImplementationExceptionSeverity">
    <xsd:restriction base="xsd:NCName">
        <xsd:enumeration value="ERROR"/>
        <xsd:enumeration value="SEVERE"/>
    </xsd:restriction>
..</xsd:simpleType>

</xsd:schema>
```

11.2 Привязка к протоколам SOAP/HTTP интерфейса управления запросами

Ниже приведена спецификация языка описания веб-сервисов (Web Service Description Language, WSDL) версии 1.1 по *[26]*, определяющая типовую привязку к протоколам SOAP/HTTP интерфейса EPCIS управления запросами. Если предоставляется привязка к протоколам SOAP/HTTP, она должна соответствовать WSDL. Эта привязка SOAP/HTTP соответствует версии 1.0 базового профиля WS-I (*аббревиатура от Web Services Interoperability industry consortium*) по *[27]*. Эта привязка строится на схеме, определенной в подразделе 11.1.

Если реализация EPCIS, предоставляющая привязку SOAP, получает входные данные, которые синтаксически не соответствуют языку WSDL, реализация ДОЛЖНА показать это одним из двух следующих способов: реализация МОЖЕТ применить исключение `ValidationException (Исключение_по_Валидации)` или МОЖЕТ применить более общее исключение, которое предоставляется используемым процессором протокола SOAP.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- EPCIS QUERY SERVICE DEFINITIONS -->
<wsdl:definitions
    targetNamespace="urn:epcglobal:epcis:wsdl:1"
    xmlns="gost.e-ecolog.ru                              "
    xmlns:apachesoap="gost.e-ecolog.ru                                  "
    xmlns:epcis="urn:epcglobal:epcis:xsd:1"
    xmlns:epcisq="urn:epcglobal:epcis-query:xsd:1"
    xmlns:epcglobal="urn:epcglobal:xsd:1"
    xmlns:impl="urn:epcglobal:epcis:wsdl:1"
    xmlns:soapenc="gost.e-ecolog.ru          mlsoap.org/soap/encoding/"
    xmlns:wsdl="gost.e-ecolog.ru                                    "
    xmlns:wsdlsoap="gost.e-ecolog.ru                                        "
    xmlns:xsd="gost.e-ecolog.ru                                    ">
```

```
  <wsdl:documentation>
    <epcglobal:copyright>
       Copyright (C) 2006-2016 GS1 AISBL, All Rights Reserved
    </epcglobal:copyright>
    <epcglobal:disclaimer>
THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY
WARRANTY OF MERCHANTABILITY, NONINFRINGMENT, FITNESS FOR PARTICULAR PURPOSE,
OR ANY WARRANTY OTHER WISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims
all liability for any damages arising from use or misuse of this Standard,
whether special, indirect, consequential, or compensatory damages, and including
liability for infringement of any intellectual property rights, relating to use
of information in or reliance upon this document.
GS1 retains the right to make changes to this document at any time, without
notice.
GS1 makes no warranty for the use of this document and assumes no responsibility
for any errors which may appear in the document, nor does it make a commitment
to update the information contained herein.
(НАСТОЯЩИЙ ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ВКЛЮЧАЯ
ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, ПАТЕНТНОЙ ЧИСТОТЫ, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ
ЦЕЛИ, ИЛИ КАКИХ-ЛИБО ИНЫХ ГАРАНТИЙ, ПРОИСТЕКАЮЩИХ ИЗ НАСТОЯЩЕГО ДОКУМЕНТА.
GS1 снимает с себя всякую ответственность за любые убытки, проистекающие из
использования или ненадлежащего использования настоящего стандарта, включая
фактические, непрямые, косвенные и компенсаторные убытки, а также включая
ответственность за нарушение каких-либо прав интеллектуальной собственности,
связанное с использованием информации, содержащейся в настоящем документе, или
основанное на использовании настоящего документа.

GS1 оставляет за собой право в любое время и без уведомления вносить изменения
в настоящий документ. GS1 не предоставляет никаких гарантий в отношении
использования настоящего документа и отказывается от ответственности за какие-
либо ошибки, которые могут обнаружиться в настоящем документе, а также от
обязательства по актуализации содержащейся в нем информации).
    </epcglobal:disclaimer>
    <epcglobal:specification>
    </epcglobal:specification>
  </wsdl:documentation>

  <!-- EPCISSERVICE TYPES -->
  <wsdl:types>
     <xsd:schema targetNamespace="urn:epcglobal:epcis:wsdl:1"
                 xmlns:impl="urn:epcglobal:epcis:wsdl:1"
                 xmlns:xsd="gost.e-ecolog.ru

       <xsd:import
            namespace="urn:epcglobal:xsd:1"
            schemaLocation="EPCglobal.xsd"/>
       <xsd:import
            namespace="urn:epcglobal:epcis:xsd:1"
            schemaLocation="EPCglobal-epcis-1_2.xsd"/>
       <xsd:import
            namespace="urn:epcglobal:epcis-query:xsd:1"
            schemaLocation="EPCglobal-epcis-query-1_2.xsd"/>
    </xsd:schema>
  </wsdl:types>

  <!-- EPCIS QUERY SERVICE MESSAGES -->
```

```
  <wsdl:message name="getQueryNamesRequest">
      <wsdl:part name="parms" element="epcisq:GetQueryNames"/>
  </wsdl:message>
  <wsdl:message name="getQueryNamesResponse">
      <wsdl:part name="getQueryNamesReturn" element="epcisq:GetQueryNamesResu
lt"/>
  </wsdl:message>

  <wsdl:message name="subscribeRequest">
      <wsdl:part name="parms" element="epcisq:Subscribe"/>
  </wsdl:message>
  <wsdl:message name="subscribeResponse">
      <wsdl:part name="subscribeReturn" element="epcisq:SubscribeResult"/>
  </wsdl:message>

  <wsdl:message name="unsubscribeRequest">
      <wsdl:part name="parms" element="epcisq:Unsubscribe"/>
  </wsdl:message>
  <wsdl:message name="unsubscribeResponse">
      <wsdl:part name="unsubscribeReturn" element="epcisq:UnsubscribeResult"/>
  </wsdl:message>

  <wsdl:message name="getSubscriptionIDsRequest">
       <wsdl:part name="parms" element="epcisq:GetSubscriptionIDs"/>
  </wsdl:message>
  <wsdl:message name="getSubscriptionIDsResponse">
       <wsdl:part name="getSubscriptionIDsReturn"
element="epcisq:GetSubscriptionIDsResult"/>
  </wsdl:message>

  <wsdl:message name="pollRequest">
      <wsdl:part name="parms" element="epcisq:Poll"/>
  </wsdl:message>
  <wsdl:message name="pollResponse">
      <wsdl:part name="pollReturn" element="epcisq:QueryResults"/>
  </wsdl:message>

  <wsdl:message name="getStandardVersionRequest">
      <wsdl:part name="parms" element="epcisq:GetStandardVersion"/>
  </wsdl:message>
  <wsdl:message name="getStandardVersionResponse">
      <wsdl:part name="getStandardVersionReturn"
element="epcisq:GetStandardVersionResult"/>
  </wsdl:message>
  <wsdl:message name="getVendorVersionRequest">
      <wsdl:part name="parms" element="epcisq:GetVendorVersion"/>
  </wsdl:message>
  <wsdl:message name="getVendorVersionResponse">
      <wsdl:part name="getVendorVersionReturn"
element="epcisq:GetVendorVersionResult"/>
  </wsdl:message>

  <!-- EPCISSERVICE FAULT EXCEPTIONS -->
  <wsdl:message name="DuplicateNameExceptionResponse">
      <wsdl:part name="fault" element="epcisq:DuplicateNameException"/>
  </wsdl:message>
```

```
      <!-- QueryValidationException not implemented in EPCIS 1.0
  <wsdl:message name="QueryValidationExceptionResponse">
      <wsdl:part name="fault" element="epcisq:QueryValidationException"/>
  </wsdl:message>
      -->
  <wsdl:message name="InvalidURIExceptionResponse">
      <wsdl:part name="fault" element="epcisq:InvalidURIException"/>
  </wsdl:message>
  <wsdl:message name="NoSuchNameExceptionResponse">
      <wsdl:part name="fault" element="epcisq:NoSuchNameException"/>
  </wsdl:message>
  <wsdl:message name="NoSuchSubscriptionExceptionResponse">
      <wsdl:part name="fault" element="epcisq:NoSuchSubscriptionException"/>
  </wsdl:message>
  <wsdl:message name="DuplicateSubscriptionExceptionResponse">
      <wsdl:part name="fault" element="epcisq:DuplicateSubscriptionException"/>
  </wsdl:message>
  <wsdl:message name="QueryParameterExceptionResponse">
      <wsdl:part name="fault" element="epcisq:QueryParameterException"/>
  </wsdl:message>
  <wsdl:message name="QueryTooLargeExceptionResponse">
      <wsdl:part name="fault" element="epcisq:QueryTooLargeException"/>
  </wsdl:message>
  <wsdl:message name="QueryTooComplexExceptionResponse">
      <wsdl:part name="fault" element="epcisq:QueryTooComplexException"/>
  </wsdl:message>
  <wsdl:message name="SubscriptionControlsExceptionResponse">
      <wsdl:part name="fault" element="epcisq:SubscriptionControlsException"/>
  </wsdl:message>
  <wsdl:message name="SubscribeNotPermittedExceptionResponse">
      <wsdl:part name="fault" element="epcisq:SubscribeNotPermittedException"/>
  </wsdl:message>
  <wsdl:message name="SecurityExceptionResponse">
      <wsdl:part name="fault" element="epcisq:SecurityException"/>
  </wsdl:message>
  <wsdl:message name="ValidationExceptionResponse">
      <wsdl:part name="fault" element="epcisq:ValidationException"/>
  </wsdl:message>
  <wsdl:message name="ImplementationExceptionResponse">
      <wsdl:part name="fault" element="epcisq:ImplementationException"/>
  </wsdl:message>

  <!-- EPCISSERVICE PORTTYPE -->
  <wsdl:portType name="EPCISServicePortType">

    <wsdl:operation name="getQueryNames">
      <wsdl:input message="impl:getQueryNamesRequest"
name="getQueryNamesRequest"/>
      <wsdl:output message="impl:getQueryNamesResponse"
name="getQueryNamesResponse"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>
```

```
    <wsdl:operation name="subscribe">
      <wsdl:input message="impl:subscribeRequest" name="subscribeRequest"/>
      <wsdl:output message="impl:subscribeResponse" name="subscribeResponse"/>
      <wsdl:fault message="impl:NoSuchNameExceptionResponse"
name="NoSuchNameExceptionFault"/>
      <wsdl:fault message="impl:InvalidURIExceptionResponse"
name="InvalidURIExceptionFault"/>
      <wsdl:fault message="impl:DuplicateSubscriptionExceptionResponse"
name="DuplicateSubscriptionExceptionFault"/>
      <wsdl:fault message="impl:QueryParameterExceptionResponse"
name="QueryParameterExceptionFault"/>
      <wsdl:fault message="impl:QueryTooComplexExceptionResponse"
name="QueryTooComplexExceptionFault"/>
      <wsdl:fault message="impl:SubscriptionControlsExceptionResponse"
name="SubscriptionControlsExceptionFault"/>
      <wsdl:fault message="impl:SubscribeNotPermittedExceptionResponse"
name="SubscribeNotPermittedExceptionFault"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>

    <wsdl:operation name="unsubscribe">
      <wsdl:input message="impl:unsubscribeRequest" name="unsubscribeRequest"/>
      <wsdl:output message="impl:unsubscribeResponse"
name="unsubscribeResponse"/>
      <wsdl:fault message="impl:NoSuchSubscriptionExceptionResponse"
name="NoSuchSubscriptionExceptionFault"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>

    <wsdl:operation name="getSubscriptionIDs">
      <wsdl:input message="impl:getSubscriptionIDsRequest"
name="getSubscriptionIDsRequest"/>
      <wsdl:output message="impl:getSubscriptionIDsResponse"
name="getSubscriptionIDsResponse"/>
      <wsdl:fault message="impl:NoSuchNameExceptionResponse"
name="NoSuchNameExceptionFault"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>

    <wsdl:operation name="poll">
      <wsdl:input message="impl:pollRequest" name="pollRequest"/>
```

```
      <wsdl:output message="impl:pollResponse" name="pollResponse"/>
      <wsdl:fault message="impl:QueryParameterExceptionResponse"
name="QueryParameterExceptionFault"/>
      <wsdl:fault message="impl:QueryTooLargeExceptionResponse"
name="QueryTooLargeExceptionFault"/>
      <wsdl:fault message="impl:QueryTooComplexExceptionResponse"
name="QueryTooComplexExceptionFault"/>
      <wsdl:fault message="impl:NoSuchNameExceptionResponse"
name="NoSuchNameExceptionFault"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>

    <wsdl:operation name="getStandardVersion">
      <wsdl:input message="impl:getStandardVersionRequest"
name="getStandardVersionRequest"/>
      <wsdl:output message="impl:getStandardVersionResponse"
name="getStandardVersionResponse"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>

    <wsdl:operation name="getVendorVersion">
      <wsdl:input message="impl:getVendorVersionRequest"
name="getVendorVersionRequest"/>
      <wsdl:output message="impl:getVendorVersionResponse"
name="getVendorVersionResponse"/>
      <wsdl:fault message="impl:SecurityExceptionResponse"
name="SecurityExceptionFault"/>
      <wsdl:fault message="impl:ValidationExceptionResponse"
name="ValidationExceptionFault"/>
      <wsdl:fault message="impl:ImplementationExceptionResponse"
name="ImplementationExceptionFault"/>
    </wsdl:operation>
  </wsdl:portType>

  <!-- EPCISSERVICE BINDING -->
  <wsdl:binding name="EPCISServiceBinding" type="impl:EPCISServicePortType">
    <wsdlsoap:binding style="document"
transport="gost.e-ecolog.ru                                    >

    <wsdl:operation name="getQueryNames">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getQueryNamesRequest">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:input>
        <wsdl:output name="getQueryNamesResponse">
```

```
            <wsdlsoap:body
                use="literal"/>
        </wsdl:output>
        <wsdl:fault name="SecurityExceptionFault">
            <wsdlsoap:fault
                name="SecurityExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ValidationExceptionFault">
            <wsdlsoap:fault
                name="ValidationExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ImplementationExceptionFault">
            <wsdlsoap:fault
                name="ImplementationExceptionFault"
                use="literal"/>
        </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="subscribe">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="subscribeRequest">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:input>
        <wsdl:output name="subscribeResponse">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:output>
        <wsdl:fault name="NoSuchNameExceptionFault">
            <wsdlsoap:fault
                name="NoSuchNameExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="InvalidURIExceptionFault">
            <wsdlsoap:fault
                name="InvalidURIExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="DuplicateSubscriptionExceptionFault">
            <wsdlsoap:fault
                name="DuplicateSubscriptionExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="QueryParameterExceptionFault">
            <wsdlsoap:fault
                name="QueryParameterExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="QueryTooComplexExceptionFault">
            <wsdlsoap:fault
                name="QueryTooComplexExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="SubscribeNotPermittedExceptionFault">
```

```
            <wsdlsoap:fault
                name="SubscribeNotPermittedExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="SubscriptionControlsExceptionFault">
            <wsdlsoap:fault
                name="SubscriptionControlsExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="SecurityExceptionFault">
            <wsdlsoap:fault
                name="SecurityExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ValidationExceptionFault">
            <wsdlsoap:fault
                name="ValidationExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ImplementationExceptionFault">
            <wsdlsoap:fault
                name="ImplementationExceptionFault"
                use="literal"/>
        </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="unsubscribe">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="unsubscribeRequest">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:input>
        <wsdl:output name="unsubscribeResponse">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:output>
        <wsdl:fault name="NoSuchSubscriptionExceptionFault">
            <wsdlsoap:fault
                name="NoSuchSubscriptionExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="SecurityExceptionFault">
            <wsdlsoap:fault
                name="SecurityExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ValidationExceptionFault">
            <wsdlsoap:fault
                name="ValidationExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ImplementationExceptionFault">
            <wsdlsoap:fault
                name="ImplementationExceptionFault"
                use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
```

```
<wsdl:operation name="getSubscriptionIDs">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getSubscriptionIDsRequest">
        <wsdlsoap:body
            use="literal"/>
    </wsdl:input>
    <wsdl:output name="getSubscriptionIDsResponse">
        <wsdlsoap:body
            use="literal"/>
    </wsdl:output>
    <wsdl:fault name="NoSuchNameExceptionFault">
        <wsdlsoap:fault
            name="NoSuchNameExceptionFault"
            use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="SecurityExceptionFault">
        <wsdlsoap:fault
            name="SecurityExceptionFault"
            use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ValidationExceptionFault">
        <wsdlsoap:fault
            name="ValidationExceptionFault"
            use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ImplementationExceptionFault">
        <wsdlsoap:fault
            name="ImplementationExceptionFault"
            use="literal"/>
    </wsdl:fault>
</wsdl:operation>

<wsdl:operation name="poll">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="pollRequest">
        <wsdlsoap:body
            use="literal"/>
    </wsdl:input>
    <wsdl:output name="pollResponse">
        <wsdlsoap:body
            use="literal"/>
    </wsdl:output>
    <wsdl:fault name="QueryParameterExceptionFault">
        <wsdlsoap:fault
            name="QueryParameterExceptionFault"
            use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="QueryTooComplexExceptionFault">
        <wsdlsoap:fault
            name="QueryTooComplexExceptionFault"
            use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="QueryTooLargeExceptionFault">
        <wsdlsoap:fault
            name="QueryTooLargeExceptionFault"
            use="literal"/>
    </wsdl:fault>
```

```
        <wsdl:fault name="NoSuchNameExceptionFault">
            <wsdlsoap:fault
                name="NoSuchNameExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="SecurityExceptionFault">
            <wsdlsoap:fault
                name="SecurityExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ValidationExceptionFault">
            <wsdlsoap:fault
                name="ValidationExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ImplementationExceptionFault">
            <wsdlsoap:fault
                name="ImplementationExceptionFault"
                use="literal"/>
        </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="getStandardVersion">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getStandardVersionRequest">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:input>
        <wsdl:output name="getStandardVersionResponse">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:output>
        <wsdl:fault name="SecurityExceptionFault">
            <wsdlsoap:fault
                name="SecurityExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ValidationExceptionFault">
            <wsdlsoap:fault
                name="ValidationExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ImplementationExceptionFault">
            <wsdlsoap:fault
                name="ImplementationExceptionFault"
                use="literal"/>
        </wsdl:fault>
    </wsdl:operation>

    <wsdl:operation name="getVendorVersion">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getVendorVersionRequest">
            <wsdlsoap:body
                use="literal"/>
        </wsdl:input>
        <wsdl:output name="getVendorVersionResponse">
            <wsdlsoap:body
```

```
                use="literal"/>
        </wsdl:output>
        <wsdl:fault name="SecurityExceptionFault">
            <wsdlsoap:fault
                name="SecurityExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ValidationExceptionFault">
            <wsdlsoap:fault
                name="ValidationExceptionFault"
                use="literal"/>
        </wsdl:fault>
        <wsdl:fault name="ImplementationExceptionFault">
            <wsdlsoap:fault
                name="ImplementationExceptionFault"
                use="literal"/>
        </wsdl:fault>
    </wsdl:operation>

  </wsdl:binding>

  <!-- EPCISSERVICE -->
  <wsdl:service name="EPCglobalEPCISService">
    <wsdl:port binding="impl:EPCISServiceBinding" name="EPCglobalEPCISServicePort">
    <!-- The address shown below is an example; an implementation MAY specify
         any port it wishes
    -->
      <wsdlsoap:address
         location="
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>
```

gost.e-ecolog.ru

11.3 Привязка к протоколу AS2 интерфейса управления запросами

В данном подразделе определяется привязка интерфейса EPCIS управления запросами к протоколу AS2 по [28]. Реализация EPCIS МОЖЕТ предоставить привязку к протоколу AS2 интерфейса EPCIS управления запросами. Если привязка AS2 предоставлена, то она ДОЛЖНА соответствовать положениям этого подраздела. Для данной привязки «клиентом запроса» является приложение EPCIS доступа, которое намеревается выполнить операции запроса EPCIS, как определено в 8.2.5, а «сервером запросов» является репозиторий EPCIS или другая система, которая выполняет такие операции в интересах клиента запроса.

Сервер запроса ДОЛЖЕН предоставить адрес URL протокола HTTP, по которому он получает сообщения от клиента запросов в соответствии с [28]. Сообщение, отправленное клиентом запросов серверу запросов, ДОЛЖНО быть документом XML, коренной элемент которого соответствует элементу `EPCISQueryDocument (Документ_Запроса_EPCIS)`, как определено схемой в подразделе 11.1. Элемент, непосредственно вложенный в элемент `EPCISBody (Тело_EPCIS)`, ДОЛЖЕН быть одним из элементов, относящихся к запросу метода интерфейса EPCIS управления запросами (т. е. одним из элементов: `Subscribe (Подписка)`, `Unsubscribe (Отмена_Подписки)`, `Poll (Опрос)` и т. д). Разрешенные элементы представлены в таблице 41. Если сообщение, отправленное клиентом запросов, не может удовлетворить названным выше требованиям, сервер запросов ДОЛЖЕН ответить исключением `ValidationException (Исключение_по_Валидации)` (то есть, возвратить экземпляр `EPCISQueryDocument (Документ_Запроса_EPCIS)`, в котором элемент, непосредственно вложенный в `EPCISBody (Тело_EPCIS)`, является исключением `ValidationException (Исключение_по_Валидации)`).

Клиент запроса ДОЛЖЕН предоставить URL адрес протокола HTTP, который сервер запросов будет использовать для доставки ответного сообщения. Этот адрес URL, как правило, сообщается дополнительно в рамках двусторонних договоренностей между торговыми партнерами (см. [28, подраздел 5.1]).

И клиент запроса, и сервер запросов ДОЛЖНЫ соответствовать требованиям и рекомендациям, приведенным в документе GS1 [29] («Рекомендации для транспортной коммуникации по протоколам EDIINT[1] AS1 и AS2»). Для справки: соответствующие части этого документа воспроизводятся ниже.

Клиент запроса ДОЛЖЕН включать типовой заголовок бизнес-документа в элемент EPCISHeader (Заголовок_EPCIS). Клиент запроса ДОЛЖЕН включать в типовой заголовок бизнес-документа уникальный идентификатор как значение элемента InstanceIdentifier (Идентификатор_Экземпляра). Клиент запроса МОЖЕТ включать другие элементы в типовой заголовок бизнес-документа, как это предусмотрено схемой. Идентификатор экземпляра, предоставленный клиентом запроса, ДОЛЖЕН быть уникальным в отношении всех других сообщений, для которых клиент запроса еще не получил соответствующий ответ. Как описано ниже, идентификатор экземпляра копируется в ответное сообщение, чтобы помочь клиенту сопоставить ответы с запросами.

Сервер запросов ДОЛЖЕН ответить на каждое сообщение, отправленное клиентом запроса, с помощью доставки ответного сообщения по адресу URL, предоставленному клиентом запроса, в соответствии с [28]. Ответное сообщение, отправленное сервером клиента, ДОЛЖНО быть документом XML, корневой элемент которого соответствует элементу EPCISQueryDocument (Документ_Запроса_EPCIS), как определено схемой в подразделе 11.1. Элемент, непосредственно вложенный в элемент EPCISBody (Тело_EPCIS), ДОЛЖЕН быть одним из элементов, приведенных в таблице 41, и соответствовать элементу, который был предоставлен в соответствующем запросе.

Таблица 41

Элемент запроса	Разрешенные возвращаемые элементы
GetQueryNames (Получение_Имен_ Запросов)	GetQueryNamesResult (Результат_Получения_Имен_запросов) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
Subscribe (Подписка)	SubscribeResult (Результат_Подписки) NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса) InvalidURIException (Исключение_при_Недействительном_URI) DuplicateSubscriptionException (Исключение_при_Дублировании_Подписки) QueryParameterException (Исключение_по_Параметру_Запроса) QueryTooComplexException (Исключение_при_Чрезмерной_Сложности_Запроса) SubscriptionControlsException (Исключение_Управления_Подпиской) SubscribeNotPermittedException (Исключение_при_Недопустимой_Подписке) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
Unsubscribe (Отмена_ Подписки)	UnsubscribeResult (Результат_Отмены_Подписки) NoSuchSubscriptionException (Исключение_при_Несуществующей_Подписке) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
GetSubscriptionIDs (Получение_ Удостоверений_ Подписки)	GetSubscriptionIDsResult (Результат_Получения_Удостоверений_ Подписки) NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
Poll (Опрос)	QueryResults (Результаты_Запроса) QueryParameterException (Исключение_по_Параметру_Запроса) QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса) QueryTooComplexException (Исключение_при_Чрезмерной_Сложности_Запроса) NoSuchNameException (Исключение_при_Несуществующем_Имени_Запроса) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)

[1] *EDIINT — сокращение от «EDIFACT over Internet».*

Окончание таблицы 41

Элемент запроса	Разрешенные возвращаемые элементы
GetStandardVersion (Получение_Версии_Стандарта)	GetStandardVersionResult (Результат_Получения_Версии_Стандарта) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)
GetVendorVersion (Получение_Версии_Поставщика)	GetVendorVersionResult (Результат_Получения_Версии_Поставщика) SecurityException (Исключение_по_Безопасности) ValidationException (Исключение_по_Валидации) ImplementationException (Исключение_по_Реализации)

Сервер запросов ДОЛЖЕН включать типовой заголовок бизнес-документа в элемент EPCISHeader (Заголовок_EPCIS). Сервер запросов ДОЛЖЕН включать в типовой заголовок бизнес-документа элемент BusinessScope (Область_Хозяйственной_Деятельности), содержащий элемент Scope (Область), содержащий элемент CorrelationInformation (Связанная_ Информация), содержащий элемент RequestingDocumentInstanceIdentifier (Идентификатор_Экземпляра_ Запрашиваемого_Документа). Значение последнего элемента ДОЛЖНО быть значением элемента InstanceIdentifier (Идентификатор_Экземпляра) из типового заголовка бизнес-документа соответствующего запроса. Вместе с элементом Scope (Область) в элемент EPCISQuery ДОЛЖЕН быть включен подэлемент Type (Тип), а элемент InstanceIdentifier (Идентификатор_Экземпляра) ДОЛЖЕН быть включен в элемент EPCIS. Сервер запросов МОЖЕТ включать другие элементы в типовой заголовок бизнес-документа, как предусмотрено схемой.

11.3.1 Требования по криптографической защите информации при применении протокола AS2

При применении протокола AS2 следует руководствоваться требованиями документов национальной системы стандартизации Российской Федерации в области криптографической защиты информации[1)].

11.4 Привязки для интерфейса обратных вызовов по запросам

В данном подразделе определяются привязки для интерфейса обратных вызовов по запросам (Query Callback Interface). Каждая привязка включает спецификацию адреса в формате идентификатора URI, которая может использоваться как параметр dest (получатель) для метода subscribe (подписка) (см. 8.2.5). Каждый нижеприведенный подраздел определяет требование соответствия (МОЖЕТ, СЛЕДУЕТ, ДОЛЖЕН) для каждой привязки.

Реализации МОГУТ поддерживать дополнительные привязки интерфейса обратных вызовов по запросам. Любая дополнительная привязка НЕ ДОЛЖНА использовать схему идентификатора URI, уже использованную одной из привязок, указанных в этом документе.

Все адреса конечных пунктов в формате идентификатора URI, независимо от того, стандартизированы ли они как часть этой спецификации или нет, ДОЛЖНЫ соответствовать общему синтаксису идентификаторов URI, как это указано в [*11*]. Каждая привязка интерфейса обратных вызовов по запросам может налагать дополнительные ограничения на синтаксис URI для использования с этой привязкой.

11.4.1 Общие соображения для всех привязок на основе XML

Нижеизложенное применяется ко всем привязкам интерфейса обратных вызовов по запросам на основе XML, указанным в 11.4.2, 11.4.3 и 11.4.4.

Полезная информация, доставленная получателю, ДОЛЖНА быть документом XML, соответствующим схеме, указанной в подразделе 11.1. В частности, полезная информация ДОЛЖНА быть экземпляром EPCISQueryDocument (Документ_Запроса_EPCIS), элемент EPCISBody (Тело_EPCIS) которого содержит один из трех элементов, приведенных в таблице *42*, в соответствии с вызываемым методом интерфейса обратных вызовов по запросам.

[1)] *Консультации по применению документов национальной системы стандартизации Российской Федерации в области криптографической защиты информации могут быть получены в ТК 026 «Криптографическая защита информации». Информация о ТК размещена на сайте Росстандарта* gost.e-ecolog.ru *ы/gost/home/activity/standardization/techcom.*

Таблица 42

Метод интерфейса обратных вызовов по запросам	Содержание полезной информации
callbackResults (результаты_Обратного_Вызова)	QueryResults (Результаты_Запроса)
callbackQueryTooLargeException (исключение_при_Превышении_ Объема_Запроса_Обратного_Вызова)	QueryTooLargeException (Исключение_при_Превышении_Объема_Запроса)
callbackImplementationException (исключение_по_Реализации_ Обратного_Вызова)	ImplementationException (Исключение_по_Реализации)

Во всех случаях поля queryName (имя_Запроса) и subscriptionID (удостоверение_Подписки) элемента с полезной информацией ДОЛЖНЫ содержать соответственно значения queryName (имя_Запроса) и subscriptionID, которые были указаны в вызове метода subscribe (подписка), создавшем длительный запрос.

11.4.2 Привязка интерфейса обратных вызовов по запросам к протоколу HTTP

Привязка к протоколу HTTP обеспечивает доставку результатов длительного запроса в XML через протокол HTTP, используя операцию POST. Реализации МОГУТ обеспечить поддержку этой привязки.

Синтаксис адресов назначения формата HTTP URI, используемых в стандарте EPCIS, ДОЛЖЕН быть таким, как определено в [*25*, пункт 3.2.2]. Адрес HTTP URI имеет одно из следующих двух неформальных представлений:

http://*host*:*port*/*remainder-of-URL*

http://*host*/*remainder-of-URL*

где *host* — это имя DNS или IP-адрес хоста, где приемник прослушивает входящие HTTP-соединения;

port — это порт TCP, на котором приемник прослушивает входящие HTTP-соединения. Порт и предшествующий знак «двоеточие» могут быть опущены, и в этом случае порт ДОЛЖЕН быть по умолчанию равен 80;

remainder-of-URL — это URL адрес, на который будет отправлена операция POST протокола HTTP (HTTP POST).

Реализация EPCIS ДОЛЖНА доставлять результаты запроса, отправляя запрос HTTP POST получателю, указанному в идентификаторе URI, где адрес *remainder-of-URL* включается в HTTP request-line (как определено в [*25*]) и где полезной информацией является документ XML, как указано в 11.4.1.

Интерпретация реализацией EPCIS кодового значения ответа, возвращенного получателем, находится вне рамок данной спецификации. Тем не менее все реализации ДОЛЖНЫ интерпретировать кодовое значение ответа 2xx (то есть любое кодовое значение ответа между 200 и 299 включительно) как нормальный ответ, не указывающий на какую-либо ошибку.

11.4.3 Привязка интерфейса обратных вызовов по запросам к протоколу HTTPS

Привязка к протоколу HTTPS обеспечивает доставку результатов длительного запроса в XML через протокол HTTP, используя операцию POST, защищенную через криптографический протокол TLS. Реализации МОГУТ обеспечить поддержку этой привязки.

Синтаксис адресов назначения формата URI в протоколе HTTPS (HTTPS URI), используемых в стандарте EPCIS, ДОЛЖЕН быть таким, как определено в [*30*, подраздел 2.4], который в свою очередь идентичен синтаксису, определенному в [*25*, подраздел 3.2.2], с заменой https на http. Адрес HTTPS URI имеет одно из следующих двух неформальных представлений:

https://*host*:*port*/*remainder-of-URL*

gost.e-ecolog.ru

где *host* — это имя DNS или IP-адрес хоста, где приемник прослушивает входящие HTTP-соединения;

port — это порт TCP, на котором приемник прослушивает входящие HTTP-соединения. Порт и предшествующий знак «двоеточие» могут быть опущены, и в этом случае порт ДОЛЖЕН быть по умолчанию равен 443;

remainder-of-URL — это URL адрес, на который будет отправлена операция POST протокола HTTP (HTTP POST).

Реализация EPCIS ДОЛЖНА доставлять результаты запроса, отправляя запрос HTTP POST получателю, указанному в идентификаторе URI, где адрес *remainder-of-URL* включается в HTTP request-line (как определено в [*25*]) и где полезной информацией является документ XML, как указано в 11.4.1.

Для привязки к HTTPS протокол HTTP ДОЛЖЕН использоваться поверх протокола TLS. С этой целью протокол TLS ДОЛЖЕН применяться, как это определено в [31] или [32]. Реализация ДОЛЖНА поддерживать хотя бы один из дополнительных наборов шифров, определенных в [31] и [32].

Интерпретация реализацией EPCIS кодового значения ответа, возвращенного получателем, находится вне рамок данной спецификации. Тем не менее все реализации ДОЛЖНЫ интерпретировать кодовое значение ответа 2xx (то есть любое кодовое значение ответа между 200 и 299 включительно) как нормальный ответ, не указывающий на какую-либо ошибку.

11.4.4 Привязка интерфейса обратных вызовов по запросам к протоколу AS2

Привязка к протоколу AS2 обеспечивает доставку результатов длительного запроса в XML через протокол AS2 по [28]. Реализации МОГУТ обеспечить поддержку этой привязки.

Синтаксис адресов назначения URI в протоколе AS2, используемых в стандарте EPCIS, ДОЛЖЕН быть следующим:

`as2:remainder-of-URI`

где адрес `remainder-of-URI` определяет конкретный коммуникационный профиль протокола AS2, который будет использоваться сервисом EPCIS для доставки информации подписчику. Синтаксис адреса `remainder-of-URI` специфичен для конкретного сервиса EPCIS, к которому выполняется подписка, при ограничении, что полный идентификатор URI ДОЛЖЕН соответствовать синтаксису URI, как определено документом [11].

Обычно значением адреса `remainder-of-URI` является строка, именующая конкретный коммуникационный профиль AS2, где профиль подразумевает такие положения, как URL адрес протокола HTTP, на который должны доставляться сообщения AS2, используемые сертификаты безопасности и т. д. Клиент интерфейса EPCIS запросов, желающий использовать протокол AS2 для доставки результатов длительного запроса, должен предварительно согласовать с провайдером Сервиса EPCIS конкретное значение адреса `remainder-of-URI` для использования.

Примечание — Использование протокола AS2 обычно требует предварительного согласования между взаимодействующими сторонами с целью замены сертификата, а также других нетипичных согласований как части двустороннего торгового партнерского соглашения (см. [28, подраздел 5.1]). Часть `remainder-of-URI` URI-адреса протокола AS2, по существу, является именем, относящимся к результату конкретного согласования такого рода.

Реализация EPCIS ДОЛЖНА доставлять результаты запроса отправкой сообщения AS2 в соответствии с документом [28]. Полезная информация сообщения AS2 ДОЛЖНА быть документом XML, как указано в 11.4.1.

Как сервис EPCIS, так и получатель результатов длительного запроса ДОЛЖНЫ соответствовать требованиям и ДОЛЖНЫ соответствовать рекомендациям, представленным в документе GS1 [29] («Рекомендации для транспортной коммуникации в протоколах EDIINT AS1 и AS2»). Для справки: соответствующие части этого документа воспроизводятся в подразделе 11.3.

12 Соответствие

В стандарте EPCIS определены как типовые данные событий, так и типовые интерфейсы между компонентами систем, передающими данные о событиях. И форматы данных и интерфейсы могут использоваться различными программными средствами и средствами работы с данными любой системы.

В настоящем разделе определяется понятие соответствия настоящему стандарту EPCIS. Ниже перечислены многочисленные системные компоненты, которые могут соответствовать различным частям настоящего стандарта EPCIS. Любая ссылка на раздел настоящего стандарта EPCIS, встречающаяся в нижеследующем тексте, понимается как относящаяся к данному разделу целиком, включая все его подразделы.

12.1 Соответствие данных EPCIS

Электронный документ считается соответствующим настоящему стандарту EPCIS при выполнении всех нижеперечисленных условий:

- он является правильно составленным документом XML, соответствующим [23];
- он соответствует XML схеме для `EPCISDocument (Документ_EPCIS)`, определенной в подразделе 9.5, XML схеме для `EPCISMasterDataDocument (Документ_Мастер-Данных_EPCIS)`, определенной в подразделе 9.7, или XML схеме для `EPCISQueryDocument (Документ_Запроса_EPCIS)`, определенной в подразделе 11.1, а также всем дополнительным условиям, определенным в соответствующем разделе;

- все данные событий EPCIS (если таковые содержатся в документе) соответствуют определениям данных событий EPCIS, указанным в разделе 7 и его подразделах;
- все мастер-данные (если таковые содержатся в документе) соответствуют ограничениям/условиям в отношении мастер-данных, определенным в 6.1.1, 6.5 и 9.4;
- все случаи использования механизма расширения (если таковые имеются) соответствуют условиям, определенным в подразделе 9.1;
- стандартный заголовок бизнес-документа (если таковой присутствует) соответствует условиям, определенным в подразделе 9.2.

Во многих случаях применение EPCIS будет требовать, помимо соответствия настоящему стандарту EPCIS, соответствия данных стандарту базовой деловой лексики по *ГОСТ Р 59168*, который определяет два уровня соответствия — «соответствующий БДЛ (CBV)» и «совместимый с БДЛ (CBV)». (Для более детальных пояснений см. стандарт БДЛ по *ГОСТ Р 59168*).

12.2 Соответствие клиентов интерфейса EPCIS сбора данных

Система соответствует настоящему стандарту EPCIS в качестве клиента интерфейса сбора данных при выполнении всех нижеперечисленных условий:

- она соответствует всем характеристикам «клиентов сбора данных», определенным в подразделах 10.1 или 10.2.

Система считается соответствующей определенной привязке интерфейса сбора данных (или нескольким таким привязкам), если она соответствует определенному подразделу раздела 10.

12.3 Соответствие серверов интерфейса EPCIS сбора данных

Система соответствует настоящему стандарту EPCIS в качестве сервера интерфейса сбора данных при выполнении всех нижеперечисленных условий:

- система соответствует всем характеристикам, определенным в подразделе 8.1;
- система соответствует всем характеристикам «серверов сбора данных», определенным в подразделах 10.1 или 10.2;
- система обрабатывает поле recordTime (время_Записи) в событиях EPCIS в соответствии с таблицей *14* в 7.4.1.

Система считается соответствующей определенной привязке интерфейса сбора данных (или нескольким таким привязкам), если она соответствует определенному подразделу раздела 10.

12.4 Соответствие клиентов интерфейса EPCIS запросов

Система соответствует настоящему стандарту EPCIS в качестве клиента интерфейса запросов при выполнении одного или обоих нижеперечисленных условий:

- система соответствует определению «отправителя», указанному в [*27*], и отправляет сообщения в соответствии со спецификацией WSDL *(по [26])* в подразделе 11.2;
- система соответствует всем характеристикам «клиента запроса», определенным в подразделе 11.3.

Система считается соответствующей определенной привязке интерфейса запросов (или нескольким таким привязкам) в зависимости от того, какому подразделу раздела 11 она соответствует.

12.5 Соответствие серверов интерфейса EPCIS запросов

Система соответствует настоящему стандарту EPCIS в качестве сервера интерфейса запросов при выполнении всех нижеперечисленных условий:

- система соответствует всем характеристикам, определенным в подразделе 8.2;
- мастер-данные, возвращаемые по сделанному системой запросу мастер-данных, соответствуют условиям, определенным в первой строке таблицы в 6.1.1;
- система включает поле `recordTime (время_Записи)` во все события EPCIS, возвращаемые в качестве результатов запроса, в соответствии с таблицей *14* в 7.4.1;
- из следующих утверждений одно или оба являются верными:

1) система соответствует определению «получателя», указанному в [*27*], получает сообщения в соответствии со спецификацией WSDL по [*26*] и соответствует дополнительным ограничениям/условиям (см. подраздел 11.2);

2) система соответствует всем характеристикам «сервера запроса», определенным в подразделе 11.3.

Система считается соответствующей определенной привязке интерфейса запросов (или нескольким таким привязкам) в зависимости от того, какому подразделу раздела 11 она соответствует.

12.6 Соответствие применений интерфейса EPCIS обратных вызовов по запросам

Система соответствует настоящему стандарту EPCIS в качестве применения интерфейса обратных вызовов по запросам, если она соответствует характеристикам, определенным в одном или нескольких пунктах подраздела 11.4. Система считается соответствующей определенной привязке интерфейса обратных вызовов по запросам (или нескольким таким привязкам), в зависимости от того, какому пункту *этого подраздела* она соответствует.

Приложение ДА
(справочное)

Перечень сокращений

Перечень сокращений, использованных в настоящем стандарте, приведен в таблице ДА.1.

Таблица ДА.1

Сокращение	Полная форма термина	Перевод на русский язык
ADI	Aerospace and Defense Identifier	Идентификатор для аэрокосмического и оборонного отраслевого сектора
ALE	Application-Level Event	Событие уровня приложения
CBV (БДЛ)	Core Business Vocabulary	Базовая деловая лексика
CPI	Component/Part Identifier	Идентификатор компонента/детали
DNS	Domain Name System	Доменная система именования
EPC	Electronic Product Code	Электронный код продукции
EPCIS	EPC Information Services	Информационные сервисы EPC
GIAI	Global Individual Asset Identifier	Глобальный идентификатор индивидуальных активов
GID	General Identifier	Общий идентификатор
GCN	Global Coupon Number	Глобальный номер купона
GDTI	Global Document Type Identifier	Глобальный идентификатор типа документа
GLN	Global Location Number	Глобальный номер места нахождения
GRAI	Global Individual Asset Identifier	Глобальный идентификатор индивидуальных активов
GSRN	Global Service Relation Number	Глобальный номер услуг
GTIN	Global Trade Item Number	Глобальный номер предмета торговли
IANA	Internet Assigned Numbers Authority	Администрация адресного пространства Интернет
IETF	Internet Engineering Task Force	Инженерный совет Интернета
ILMD (МДЭС)	Instance/Lot Master Data	Мастер-данные на уровне экземпляра/серии
LLRP	Low-Level Reader Protocol Interface	Низкоуровневый протокол устройства считывания радиочастотной идентификации
LPN	License plate number	Номерной знак
OID	Object identifier	Идентификатор объекта
PEN	Private Enterprise Number	Номер частного предприятия
QoS	Quality-of-service	Качество обслуживания
RFID	Radio-Frequency Identification	Радиочастотная идентификация
SBDH	Standard Business Document Header	Типовой заголовок бизнес-документа
SGTIN	Serialised Global Trade Item Number	Сериализованный глобальный номер предмета торговли
SSCC	Serial Shipping Container Code	Серийный номер транспортной упаковки
TCP	Transmission Control Protocol	Протокол управления передачей
TLS	Transport Layer Security	Протокол безопасности транспортного уровня
UML	Unified Modeling Language	Унифицированный язык моделирования
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business	Центр Организации Объединенных Наций по упрощению процедур торговли и электронным деловым операциям

Окончание таблицы ДА.1

Сокращение	Полная форма термина	Перевод на русский язык
URI	Uniform Resource Identifier	Универсальный идентификатор ресурса
URL	Uniform Resource Locator	Унифицированный указатель ресурса
URN	Uniform Resource Name	Унифицированное имя ресурса
USDoD	US Department of Defense Identifier	Идентификатор Министерства обороны США
UTC[1)]	Coordinated Universal Time	Всемирное координированное время
UUID	Universally Unique Identifier	Универсальный уникальный идентификатор
W3C	World Wide Web Consortium	Консорциум Всемирной паутины
WSDL	Web Service Description Language	Язык описания веб-сервисов
WS-I	Web Services Interoperability industry consortium	Организация функциональной совместимости веб-служб
XML	eXtensible Markup Language	Расширяемый язык разметки

[1)] *Принятое сокращение не соответствует полной форме термина.*

Приложение ДБ
(справочное)

Перечень изменений в настоящем стандарте относительно ИСО/МЭК 19987:2017 и пояснения по необходимости их внесения

В настоящем стандарте таблицам и рисункам присвоены порядковые номера, поскольку в [1] (ИСО/МЭК 19987:2017) отсутствует нумерация таблиц и рисунков. Кроме того, фрагменты текста [1], обозначенные «Справочная информация. Разъяснения», оформлены в виде примечаний. Сведения о соответствующих изменениях приведены в таблице ДБ.1.

Таблица ДБ.1

Структурный элемент настоящего стандарта	Характеристика технических отклонений и причин их внесения
Введение	В раздел «Введение» включены структурные элементы, приведенные в [1] (ИСО/МЭК 19988:2017): аннотация к документу [2], протокол изменений в [2], отказ от ответственности
Раздел 1, Область применения	Раздел «1 Введение» [1] (ИСО/МЭК 19987:2017) переименован в раздел «1 Область применения» для приведения в соответствие с требованиями ГОСТ Р 1.5 и в связи с отсутствием соответствующего раздела в [1] (ИСО/МЭК 19988:2017)
3.2, рисунок	Рисунку подраздела 3.2 присвоен номер 1
3.2, 1-й и 2-й абзацы	Приведена ссылка на рисунок 1
3.3, таблица	Таблице подраздела 3.3 присвоен номер 1
3.3, 1-й абзац	Приведена ссылка на таблицу 1
3.3, рисунок	Рисунку подраздела 3.3 присвоен номер 2
3.3, 3-й абзац	Приведена ссылка на рисунок 2
5.1, рисунок	Рисунку подраздела 5.1 присвоен номер 3
5.1, 1-й и 2-й абзацы	Приведена ссылка на рисунок 3
6.1, рисунок	Рисунку подраздела 6.1 присвоен номер 4
6.1, 3-й абзац	Приведена ссылка на рисунок 4
6.1.1, таблица	Таблице пункта 6.1.1 присвоен номер 2
6.1.1, 2-й абзац	Приведена ссылка на таблицу 2
6.3, таблица	Таблице подраздела 6.3 присвоен номер 3
6.3, 9-й абзац	Приведена ссылка на таблицу 3
7.1.2, рисунок	Рисунку пункта 7.1.2 присвоен номер 5
7.1.2, 1-й и 2-й абзацы	Приведена ссылка на рисунок 5
7.2, рисунок	Рисунку подраздела 7.2 присвоен номер 6
7.2, 16-й абзац и примечание	Приведена ссылка на рисунок 6
7.2, первая таблица	Первой таблице подраздела 7.2 присвоен номер 4
7.2, 19-й абзац	Приведена ссылка на таблицу .
7.2, вторая таблица	Второй таблице подраздела 7.2 присвоен номер 5
7.2, 21-й абзац	Приведена ссылка на таблицу 5
7.3.1, таблица	Таблице пункта 7.3.1 присвоен номер 6
7.3.1, 1-й абзац	Приведена ссылка на таблицу 6
7.3.2, таблица	Таблице пункта 7.3.2 присвоен номер .
7.3.2, 2-й абзац	Приведена ссылка на таблицу 7
7.3.3.3, первая таблица	Первой таблице подпункта 7.3.3.3 присвоен номер 8

Продолжение таблицы ДБ.1

Структурный элемент настоящего стандарта	Характеристика технических отклонений и причин их внесения
7.3.3.3, 1-й абзац	Приведена ссылка на таблицу 8
7.3.3.3, вторая таблица	Второй таблице подпункта 7.3.3.3 присвоен номер 9
7.3.3.3, 3-й абзац	Приведена ссылка на таблицу 9
7.3.3.3.1, последний абзац	Предложение в [1]: «Однако значение «фунт-сила на фут (pound-force per foot)» (F17) — недопустимо, поскольку здесь значение сила, поделенная на длину (Quantity — force divided by length), а не просто длина (length)» заменено на «Однако значение «килограмм на метр (kilogram per metre)» (KL) — недопустимо, поскольку здесь приведено значение в килограммах, поделенное на длину в метрах, а не просто длина (length)». В примере использованы не используемые в России единицы измерения.
7.3.3.4, таблица	Таблице подпункта 7.3.3.4 присвоен номер 10
7.3.3.4, 1-й абзац	Приведена ссылка на таблицу 10
7.3.4, таблица	Таблице пункта 7.3.4 присвоен номер 11
7.3.4, 2-й абзац	Приведена ссылка на таблицу 11
7.3.4.1, рисунок	Рисунку подпункта 7.3.4.1 присвоен номер 7
7.3.4.1, 1-й абзац	Для ссылки на рисунок 7 приведена фраза: «На рисунке 7 приведен пример, представляющий различия между местом считывания и производственным местом назначения»
7.3.4.1, 2-й абзац	Приведена ссылка на рисунок 7
7.3.5.3, таблица	Таблице подпункта 7.3.5.3 присвоен номер 12
7.3.5.3, 2-й абзац	Приведена ссылка на таблицу 12
7.3.5.4, таблица	Таблице подпункта 7.3.5.4 присвоен номер 13
7.3.5.4, 4-й абзац	Приведена ссылка на таблицу 13
7.4.1, таблица	Таблице пункта 7.4.1 присвоен номер 14
7.4.1, 2-й абзац	Приведена ссылка на таблицу 14
7.4.1.2, таблица	Таблице подпункта 7.4.1.2 присвоен номер 15
7.4.1.2, 2-й абзац	Приведена ссылка на таблицу 15
7.4.2, первая таблица	Первой таблице пункта 7.4.2 присвоен номер 16
7.4.2, 3-й абзац	Приведена ссылка на таблицу 16
7.4.2, вторая таблица	Второй таблице пункта 7.4.2 присвоен номер 17
7.4.2, 4-й абзац	Приведена ссылка на таблицу 17
7.4.3, первая таблица	Первой таблице пункта 7.4.3 присвоен номер 18
7.4.3, 3-й абзац	Приведена ссылка на таблицу 18
7.4.3, вторая таблица	Второй таблице пункта 7.4.3 присвоен номер 19
7.4.3, 9-й абзац	Приведена ссылка на таблицу 19
7.4.4, таблица	Таблице пункта 7.4.4 присвоен номер 20
7.4.5, первая таблица	Первой таблице пункта 7.4.5 присвоен номер 21
7.4.5, 2-й абзац	Приведена ссылка на таблицу 21
7.4.5, вторая таблица	Второй таблице пункта 7.4.5 присвоен номер 22
7.4.5, 3-й абзац	Приведена ссылка на таблицу 22
7.4.6, таблица	Таблице пункта 7.4.6 присвоен номер 23
7.4.6, 3-й абзац	Приведена ссылка на таблицу 23
8, первый рисунок	Первому рисунку раздела 8 присвоен номер 8

Продолжение таблицы ДБ.1

Структурный элемент настоящего стандарта	Характеристика технических отклонений и причин их внесения
8, 1-й абзац	Приведена ссылка на рисунок 8
8, второй рисунок	Второму рисунку раздела 8 присвоен номер 9
8, 2-й абзац	Приведена ссылка на рисунок 9
8.1.2, рисунок	Рисунку пункта 8.1.2 присвоен номер 10
8.1.2, 1-й абзац	Для ссылки на рисунок 10 приведена фраза: «На рисунке 10 приведена схема, представляющая сервис сбора данных»
8.1.2, таблица	Таблице пункта 8.1.2 присвоен номер 24
8.1.2, 5-й абзац	Приведена ссылка на таблицу 24
8.2.5, таблица	Таблице пункта 8.2.5 присвоен номер 25
8.2.5, 6-й абзац	Приведена ссылка на таблицу 25
8.2.5.1, таблица	Таблице подпункта 8.2.5.1 присвоен номер 26
8.2.5.1, 2-й абзац	Приведена ссылка на таблицу 26
8.2.5.3, таблица	Таблице подпункта 8.2.5.3 присвоен номер 27
8.2.5.3, 11-й и 15-й абзацы	Приведена ссылка на таблицу 27
8.2.5.4, таблица	Таблице подпункта 8.2.5.4 присвоен номер 28
8.2.5.4, 2-й абзац	Приведена ссылка на таблицу 28
8.2.6, первая таблица	Первой таблице пункта 8.2.6 присвоен номер 29
8.2.6, 1-й абзац	Приведена ссылка на таблицу 29
8.2.6, вторая таблица	Второй таблице пункта 8.2.6 присвоен номер 30
8.2.6, 2-й абзац	Приведена ссылка на таблицу 30
8.2.7.1, таблица	Таблице подпункта 8.2.7.1 присвоен номер 31
8.2.7.1, 4-й абзац	Приведена ссылка на таблицу 31
8.2.7.2, таблица	Таблице подпункта 8.2.7.2 присвоен номер 32
8.2.7.2, 3-й абзац	Приведена ссылка на таблицу 32
9.2, первая таблица	Первой таблице подраздела 9.2 присвоен номер 33
9.2, 3-й абзац	Приведена ссылка на таблицу 33
9.2, вторая таблица	Второй таблице подраздела 9.2 присвоен номер 34
9.2, 4-й абзац	Приведена ссылка на таблицу 34
9.4, таблица	Таблице подраздела 9.4 присвоен номер 35
9.4, 1-й абзац	Приведена ссылка на таблицу 35
9.5, таблица	Таблице подраздела 9.5 присвоен номер 36
9.5, 1-й абзац	Приведена ссылка на таблицу 36
9.7, таблица	Таблице подраздела 9.7 присвоен номер 37
9.7, 1-й абзац	Приведена ссылка на таблицу 37
11, таблица	Таблице раздела 11 присвоен номер 38
11, 1-й абзац	Приведена ссылка на таблицу 38
11.1, первая таблица	Первой таблице подраздела 11.1 присвоен номер 39
11.1, 2-й абзац	Приведена ссылка на таблицу 39
11.1, вторая таблица	Второй таблице подраздела 11.1 присвоен номер 40
11.1, 10-й абзац	Приведена ссылка на таблицу 40
11.3, таблица	Таблице подраздела 11.3 присвоен номер 41

Окончание таблицы ДБ.1

Структурный элемент настоящего стандарта	Характеристика технических отклонений и причин их внесения
11.3, 6-й абзац	Приведена ссылка на таблицу 41
11.3.1	Пункт переименован и содержит положения, соответствующие требованиям документов национальной системы стандартизации Российской Федерации в области криптографической защиты информации
11.4.1, таблица	Таблице пункта 11.4.1 присвоен номер 42
11.4.1, 6-й абзац	Приведена ссылка на таблицу 42
11.4.3, 4-й абзац	Абзац изменен с целью учета требований документов национальной системы стандартизации Российской Федерации в области криптографической защиты информации
Библиография	Раздел 13 «Ссылки» [1] (ИСО/МЭК 19987:2017) переименован в «Библиографию»
Библиография	Буквенно-цифровые обозначения документов заменены на порядковые цифровые обозначения документов в квадратных скобках. Библиография дополнена ссылками на документы, приведенными в тексте настоящего стандарта, но не указанными в разделе 12 «Ссылки» [1] (ИСО/МЭК 19987:2017)
Библиография	Из библиографии исключены ссылки, не используемые в тексте настоящего стандарта
Библиография	Библиография дополнена ссылками на рекомендации по стандартизации: [32] Р 1323565.1.020—2018 и [33] Р 1323565.1.030—2020
Стандарт в целом	Стандарт дополнен дополнительными справочными приложениями ДА и ДБ

Библиография

[1] *ИСО/МЭК 19987:2017 «Информационные технологии. Стандарт информационных сервисов EPC (EPCIS)» (ISO/IEC 19987:2017 «Information technology — EPC Information Services (EPCIS) Standard»)*[1)]

[2] *[EPCIS 1.2] EPC Information Services (EPCIS) Standard. Version 1.2. Sep 2016 (Стандарт информационных сервисов EPC (EPCIS) Версия 1.2 Сентябрь 2016)*[1)]

[3] *EPC Information Services (EPCIS) Version 1.0.1. Specification. Errata Approved by TSC on September 21, 2007*[1)]

[4] *[EPCIS 1.1] EPC Information Services (EPCIS). Specification. GS1 Standard.Version 1.1, May 2014*[1)]

[5] *ИСО/МЭК 19988:2017 «Информационные технологии. Стандарт базовой деловой лексики» (ISO/IEC 19988:2017 «Information technology — Core Business Vocabulary Standard»*[1)]

[6] [CBV1.2] GS1, "GS1 Core Business Vocabulary (CBV) Version 1.2 Standard," GS1 standard, June 2016, http://gost.e-ecolog.ru

[7] [EPCISGuideline] GS1, "EPCIS and CBV Implementation Guideline," GS1 Guideline, October 2015, http://www.gs1.org/docs/epc/EPCIS_Guideline.pdf

[8] [EPCAF] K. R. Traub et al, "EPCglobal Architecture Framework," EPCglobal technical document, July 2005, http://gost.e-ecolog.ru .pdf

[9] [GS1Arch] "The GS1 System Architecture," GS1 technical document, April 2014, gost.e-ecolog.ru gsmp/architecture/GS1_System_Architecture.pdf

[10] [ISODir2] ISO, "Rules for the structure and drafting of International Standards (ISO/IEC Directives, Part 2, 2001, 4th edition)," July 2002

[11] [RFC2396] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC2396, August 1998, gost.e-ecolog.ru

[12] [TDS1.9] GS1, "GS1 EPCglobal Tag Data Standards Version 1.9," GS1 Standard, June 2014, gost.e-ecolog.ru epc/tag-data-standard

[13] [RFC1738] T. Berners-Lee, L. Masinter, M. McCahill, "Uniform Resource Locators (URL)," RFC 1738, December 1994, gost.e-ecolog.ru

[14] [RFC2141] R. Moats, "URN Syntax," Internet Engineering Task Force Request for Comments RFC-2141, May 1997, gost.e-ecolog.ru .txt

[15] *ИСО 8601Data elements and interchange formats -- Information interchange -- Representation of dates and times (Элементы данных и форматы обмена. Обмен информацией. Представление дат и времени)*[1), 2)]

[16] [CEFACT20] United Nations Economic Commission for Europe, "Recommendation 20: Codes for Units of Measure Used in International Trade," September 2010, gost.e-ecolog.ru recommendations/rec20/rec20_Rev7e_2010.zip

[17] [ALE1.0] EPCglobal, "The Application Level Events (ALE) Specification, Version 1.0," EPCglobal Standard Specification, September 2005, http://gost.e-ecolog.ru

[18] [XSD2] P. Biron, A. Malhotra, "XML Schema Part 2: Datatypes," W3C Recommendation, May 2001, http://www.w3.org/TR/xmlschema-2/

[19] [XSD1] H. Thompson, D. Beech, M. Maloney, N. Mendelsohn, "XML Schema Part 1: Structures," W3C Recommendation, May 2001, gost.e-ecolog.ru

[20] [XMLDR] "XML Design Rules for EAN.UCC, Version 2.0," February 2004

[21] [XMLVersioning] D. Orchard, "Versioning XML Vocabularies," December 2003, gost.e-ecolog.ru 2003/12/03/versioning.html

[22] [SBDH] United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), "Standard Business Document Header Technical Specification, Version 1.3," June 2004, gost.e-ecolog.ru gsmp/kc/ecom/xml/xml_sbdh.html

[23] [XML1.0] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)," W3C Recommendation, February 2004, gost.e-ecolog.ru

[24] *GS1, «Core Business Vocabulary (CBV)», GS1 Standard, version 1.1, May 2014 (GS1, «Базовая деловая лексика (БДЛ)», стандарт GS1, версия 1.1, май 2014)*[1)]

[25] [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC2616, June 1999, gost.e-ecolog.ru

[26] [WSDL1.1] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C Note, March 2001, gost.e-ecolog.ru

[27] [WSI] K. Ballinger, D. Ehnebuske, M. Gudgin, M. Nottingham, P. Yendluri, "Basic Profile Version 1.0," WS-i Final Material, April 2004, gost.e-ecolog.ru

---

1) *Ссылка на данный документ включена в библиографию дополнительно в связи с ее отсутствием в разделе 13 «Ссылки» [1] (ИСО/МЭК 19988:2017).*

2) *На момент публикации ИСО/МЭК 19987:2017 действовал ISO 8601:2004. В Российской Федерации действует ГОСТ ИСО 8601—2001 «Система стандартов по информации, библиотечному и издательскому делу. Представление дат и времени. Общие требования», идентичный ИСО 8601:2000.*

[28] [RFC4130] D. Moberg and R. Drummond, "MIME-Based Secure Peer-to-Peer Business Data Interchange Using HTTP, Applicability Statement 2 (AS2)," RFC4130, July 2005, http://www.ietf.org/rfc/rfc4130

[29] [EDICG] GS1, "EDIINT AS1 and AS2 Transport Communications Guidelines," GS1 Technical Document, February 2006, gost.e-ecolog.ru)

[30] [RFC2818] E. Rescorla, "HTTP Over TLS," RFC2818, May 2000, http://www.ietf.org/rfc/rfc2818

[31] *Р 1323565.1.020—2018 «Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.2)»*

[32] *Р 1323565.1.030—2020 «Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3)»*

[1)] *См. также* gost.e-ecolog.ru *s/xml/EDIINT_AS1_AS2_Transport_Comm_Guide_i1.pdf.*

УДК 003.62:681.3.04:681.3.053:006.354 ОКС 35.200

Ключевые слова: информационные сервисы EPC, EPCIS, GS1

Редактор *П.К. Одинцов*
Технический редактор *В.Н. Прусакова*
Корректор *Л.С. Лысенко*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 23.12.2020. Подписано в печать 19.01.2021. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 18,60. Уч.-изд. л. 16,84.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении во ФГУП «СТАНДАРТИНФОРМ» для комплектования Федерального информационного фонда стандартов, 117418 Москва. Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru gost.e-ecolog.ru

gost.e-ecolog.ru