Процедура 1 раздела 15 Руководства Международного комитета  
 по регистрации животных (ICAR) — Методология

Процедура 1 — Методология Утверждено к публикации

Дата выпуска версии: февраль 2019 г. Утверждено к публикации

Имя файла: 15 Procedure 1 Methodology v19.01.docx

1 Введение

Существует множество технологий для обмена данными в электронном виде; кроме того, в этой области постоянно появляются новые технологии. Однако с целью установления стандартов обмена данными мы сосредоточимся на определенном наборе признанных и давно существующих технологий, применение которых будет подробно описано.

Данная процедура, входящая в Раздел 15 Руководства, необходима для введения в общую методологию и технологию отдельных бизнес-процессов, описанных в процедурах 2, 3 и 4:

а. **Общие требования к передаче данных**

описывает применяемые транспортные протоколы веб-сервисов и протоколы данных

б. **Доступ к сервису**

описывает некоторые аспекты, которые необходимо принимать во внимание при внедрении и установке сервисов

в. **Рабочие процессы коммуникации**

описываются аспекты рабочих процессов коммуникации в сессии бизнес-процесса (например, запрос, ответ и обработка данных.

г. **Работа с локальными требованиями**

подробно описывается процедура добавления в интерфейсы обязательных (на местном уровне) элементов данных, которые не входят в объем ICAR.

д. **Общие компоненты**

описывает все обычные компоненты протоколов, которые постоянно используются в специальных бизнес-процессах

2 Определения и терминология

Таблица 1 содержит список определений терминов и аббревиатур, используемых в настоящем руководстве.

Таблица 1. Определения терминов и аббревиатур, используемых в настоящем руководстве.

| **Термин** | **Определение** |
| --- | --- |
| TCP/IP | Протокол контроля передачи |
| HTTP | Протокол передачи гипертекста |
| XML | Расширяемый язык разметки |
| XSD | Определение схемы XML |
| SOAP | Протокол доступа к простому объекту |
| WSDL | Язык описания веб-сервиса |
| REST | Передача состояния представления |
| ADIS | Синтаксис обмена сельскохозяйственными данными |
| ADED | Словарь элементов сельскохозяйственных данных |
| JSON | Объектная нотация JavaScript |
| UNCEFACT | UN/CEFACT — Центр ООН по упрощению процедур и практики в управлении, торговле и на транспорте |
| ISO | Международная организация по стандартизации |
| IANA | Управление по присвоению интернет-номеров |

3 Область применения

дает наглядную схему основных элементов настоящей процедуры. Числа на данном рисунке относятся к номерам разделов настоящей процедуры.



Рисунок 1. Область применения процедуры 1 Раздела 15 Руководства ICAR

4 Общие требования к передаче данных

4.1 Веб-сервисы SOAP/XML

В качестве основного метода передачи данных был выбран стандарт W3C SOAP/XML версии 1.2. Подробное описание данного стандарта можно найти по следующей ссылке:

[http://www.w3.org/TR/2007/REC-soap12-parto-20070427/](http://www.w3.org/TR/2oo7/REC-soap12-parto-2oo7o427/)

Так как был принят нисходящий принцип, ICAR ADE предоставляет WSDL файлы и набор XSD файлов, образующих полный набор машиночитаемых файлов определений, с помощью которых можно работать с веб-сервисами. Большинство современных языков программирования

(например, Java, .NET и прочие) предоставляют инструменты, позволяющие легко задавать классы интерфейсов, которые будут связывать слой передачи данных и слой бизнес-процессов.

Для каждого релиза спецификаций ICAR ADE можно загрузить специальный набор файлов WSDL/XSD с веб-сайта ICAR ADE ([ссылка](https://icar-web.atlassian.net/wiki/spaces/ADE/overview)). См. Таблицу 2. Файлы схемы ICAR ADE для описания файлов. Файлы будут предоставлены в zip-архиве, например “ADE\_Schema20150309\_1.8.zip” для версии 1.8, доступной по [ссылке](https://icar-web.atlassian.net/wiki/spaces/ADE/pages/1835044/WSDL-files+1.8).

Файлы WSDL/XSD описывают различные аспекты транспортного протокола:

е. Оконечные точки сервиса

ж. Сообщения ввода и вывода

з. Определения базовых типов данных (типы XSD, типы UNCEFACT и т. д.)

е. Определения сложных типов данных  
 (метки E-Tag, теги элементов, композиции, атрибуты, использование базовых типов данных)

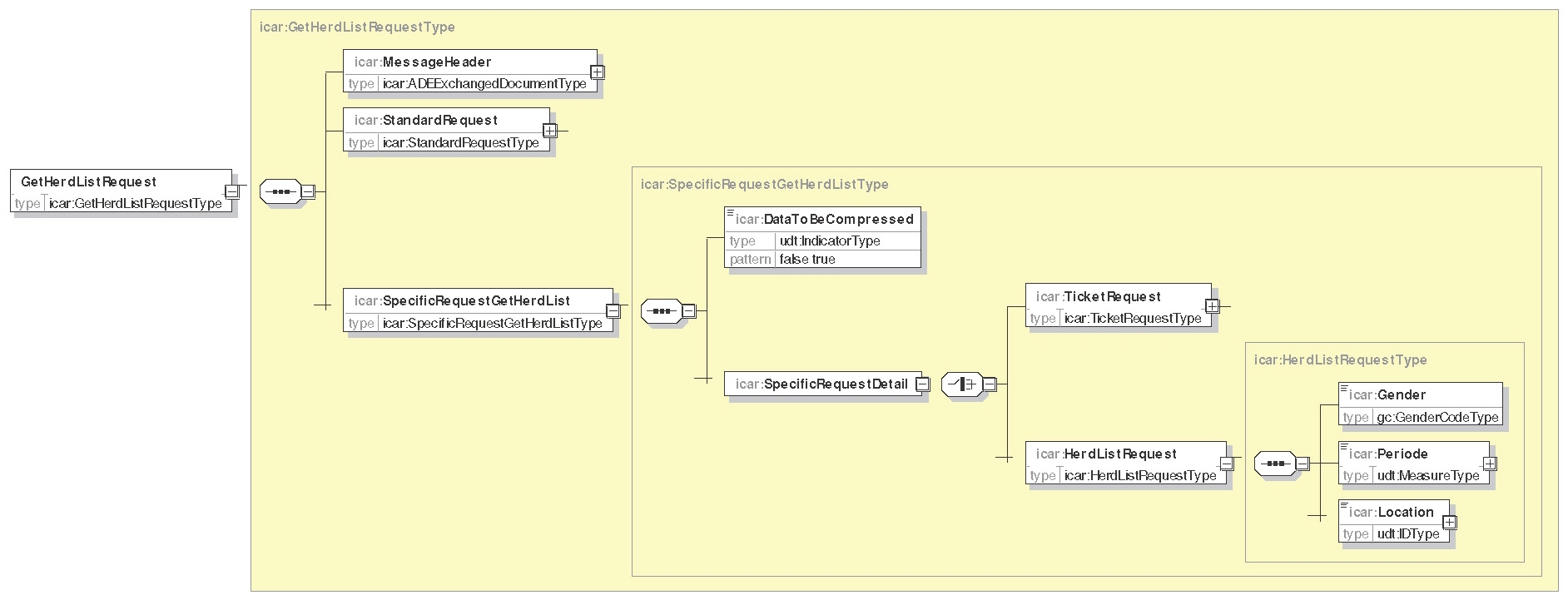
к. Требования (ограничения для типов данных и диапазона значений)

Таблица 2. Файлы схемы ICAR ADE

| **Имя файла /**  **Схема имени файла** | **Описание файла** |
| --- | --- |
| \*.XSD | Файлы определения схемы XML, машиночитаемое описание ICAR ADE  Структуры данных XML |
| ICAR\_\*.XSD | Определения элементов данных, созданные группой ICAR ADE |
| ISO\_\*.XSD | Определения элементов файлов, выведенные из стандартов ISO |
| UNECE\_\*.XSD | Файлы XSD, предоставляемые проектом UN/CEFACT:  (см. <http://www.unece.org/cefact/xml_schemas/index.html>) |
| UnqualifiedDataType\_13p0.XSD | Файл XSD, определяющий неквалифицированные типы данных UNCEFACT. Эти типы данных расширяют стандартные типы данных XSD с помощью дополнительных атрибутов. Они используются в определениях ICAR и UNCEFACT вместо стандартных типов XSD. |
| \*\_CODE\_\* | Файлы XSD, определяющие списки кодов. |
| wsMrAde.wsdl | Описание сервиса ICAR ADE SOAP (WSDL = Язык описания веб-сервиса)  Он определяет набор пар сообщений, состоящий из запроса и ответа. |
| ADE\_vlp8.xsd | Данный файл предоставляет точку входа в определения структуры данных сообщений ICAR ADE |

В настоящем документе сообщения и определения данных иллюстрируются с помощью графического представления структур XSD, созданного с помощью коммерческого ПО XMLSPY компании ALTOVA.

См., например, Рисунок 2 (Пример “GetHerdListRequest”)

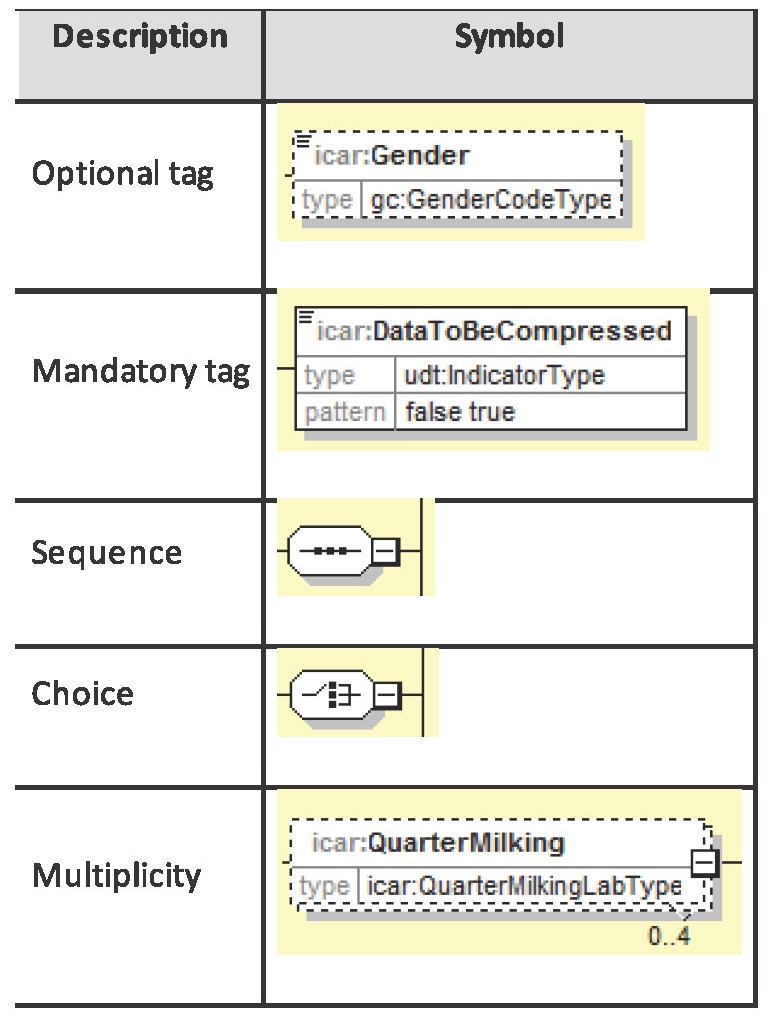


|  |  |
| --- | --- |
| icar :Get HerdListRequestType | icar: Get HerdListRequestType |
| icar:MessageHeader | icar: MessageHeader |
| icar:ADEExchangedDocumentType | icar: ADEExchangedDocumentType |
| icar:StandardRequest | icar: StandardRequest |
| icar:SiatidardlRequastType | icar: SiatidardlRequastType |
| icar:SpecificRequestGetHerdListType | icar: SpecificRequestGetHerdListType |
| icar:Data To Be Compressed | icar: Data To Be Compressed |
| udt :lndicatorType | udt: lndicatorType |
| false true | неверно верно |
| Get He r d List Request | Get Herd List Request |
| icar:GetHerdLstRequestType | icar: GetHerdLstRequestType |
| icar:SpecificRequestDetail | icar: SpecificRequestDetail |
| icar:TicketRequest | icar: TicketRequest |
| icar:TicketRequestType | icar: TicketRequestType |
| icar:HerdListRequestType | icar: HerdListRequestType |
| icar:Gender | icar: Gender |
| icar:SpecificRequestGetHerdList | icar: SpecificRequestGetHerdList |
| icar:SpecificRequestGetHerdListType | icar: SpecificRequestGetHerdListType |
| gc:GenderCodeType | gc: GenderCodeType |
| icar:HerdListRequest | icar: HerdListRequest |
| icar:HerdListRequestType | icar: HerdListRequestType |
| icar:Periode | icar: Periode |
| udt:MeasureType | udt: MeasureType |
| icar:Location | icar: Location |
| udt:IDType | udt: IDType |

*Рисунок 2. Пример “GetHerdListRequest”*

Символы ниже используются для представления схем XSD на следующих страницах:

Таблица 3. Символы схемы XSD



|  |  |
| --- | --- |
| Description | Описание |
| Symbol | Символ |
| Optional tag | Опциональный тег |
| icar:Gender | icar: Gender |
| gc:GenderCodeType | gc: GenderCodeType |
| icar:DataToBeCompressed | icar: DataToBeCompressed |
| udt:IndicaterType | udt: IndicaterType |
| Mandatory tag | Обязательный тег |
| false true | неверно верно |
| Sequence | Последовательность |
| Choice | Выбор |
| Multiplicity | Кратность |
| icar:QuarterMilking | icar: QuarterMilking |
| icar:QuarterMilkingLabType | icar: QuarterMilkingLabType |

4.2 Альтернативные транспортные протоколы

Наряду с SOAP существуют иные международные транспортные протоколы. Например, стандарт ISO ADIS/ADED и его последние расширения, описанные в ISO 17532(2007). С другой стороны, в последние годы широкое распространение в области передачи данных через интернет получил протокол REST на основе транспортного протокола HTTP с использованием XML, JSON и прочих технологий шифрования данных.

ADE-WG работает над созданием стандартизированного описания REST-API.

4.3 Типы и списки кодов ICAR

Элементы, создаваемые в рамках процесса спецификации ICAR ADE, можно идентифицировать по префиксу **"icar: "** в файлах WSDL/XSD и описаниях ниже.

Все элементы данных основываются на типах данных UNCEFACT (см. раздел «Типы данных UNCEFACT»).

Ниже представлено подробное описание элементов данных.

4.4 Внешние элементы

Целью процесса стандартизации ICAR ADE является максимально полное использование элементов, уже определенных в рамках существующих структур стандартизации, таких как ISO, IANA и UNCEFACT.

4.4.1. Типы данных UNCEFACT

Определения WSDL/XSD ICAR ADE широко используют базовые типы UNCEFACT.

Базовые типы UNCEFACT — это типы, производные от базовых типов XSD. Концепция в основном предусматривает добавление к типам XSD дополнительных атрибутов и ограничителей. Они определены в файле UnqualifiedDataType\_13po.xsd.

Эти элементы в файлах WSDL/XSD и описаниях ниже можно идентифицировать по префиксу **“udt:”**

В Таблице 4 «Базовые типы данных UNCEFACT» перечислены типы UNCEFACT, которые в настоящее время используются для типов элементов данных.

Таблица 4. Базовые типы данных UNCEFACT

| Наименование | Описание | Комментарий | Тип XSD |
| --- | --- | --- | --- |
| udt: CodeType | Строка символов, которая может использоваться для представления или замены определяемого значения. | Код относится к нумерации. | Cтрока |
| udt: DateType | Конкретная точка в массиве дат | Представление, как оно определено в транспортном протоколе | Cтрока |
| udt: DateTimeType | Конкретная точка в потоке времени | Представление, как оно определено в транспортном протоколе | Cтрока |
| udt: IDType | Уникальная строка символов, используемая для идентификации и отличения некоего объекта в схеме идентификации от всех прочих объектов в той же схеме. |  | Cтрока |
| udt: MeasureType | Численный показатель, определяемый при помощи измерения объекта | Измерения дополняются атрибутом единицы, содержащим в себе единицу измерения, разрядность и, когда это требуется, минимальное и максимальное значения. | Десятичное число |
| udt: NameType | Слово, которое представляет собой четкое определение лица, места, вещи или концепции |  | Cтрока |
| udt: BinaryObjectType | Двоичные данные, кодированные BASE64 | Используется, например, для сжатых сообщений | Cтрока |
| udt: TextType | Общий тип текста |  | Cтрока |
| udt: IndicatorType | Булев показатель или верно/неверно |  | Булев атрибут |

4.4.2 Списки кодов UNCEFACT

UNCEFACT предоставляет расширенный набор списков кодов. В Таблице 5 «Списки кодов UNCEFACT» указаны списки кодов, используемые в спецификациях ICAR ADE.

Таблица 5. Списки кодов UNCEFACT

| **Список кодов** | **Описание** |
| --- | --- |
| AgencyIdentificationCode | Список агентств, ответственных за поддержание списков кодов, которые используются в качестве значения атрибута **listAgencyID** в типе данных UNCEFACT **udt: CodeType** |
| CharacterSetEncodingCode | Список набора символов кодов, которые используются в качестве значения атрибута **encodingCode** в типе данных UNCEFACT **xsd:base64Binary** |
| MeasurementUnitCommonCode | Расширенный набор обычно используемых единиц измерения, которые используются в качестве значения атрибута **unitCode** в типе данных UNCEFACT **udt: MeasureType** |

4.4.3 Списки кодов ISO

В таблице 6 «Списки кодов ISO» указаны списки кодов ISO, используемые в спецификациях ICAR ADE.

Таблица 6. Списки кодов ISO

| **Список кодов** | **Описание** |
| --- | --- |
| ISO3AlphaCurrencyCode | Список кодов валют ISO, которые используются в качестве значения атрибута **currencyID** в типе данных UNCEFACT **udt: AmountType** |
| ISOTwoletterCountryCode | Двухбуквенные коды страны ISO, которые используются в качестве значения показателя **Страна (Country)** |

4.4.4 Списки кодов IANA

В Таблице 7 «Списки кодов IANA» указаны списки кодов IANA, используемые в спецификациях ICAR ADE.

Таблица 7. Списки кодов IANA

| **Список кодов** | **Описание** |
| --- | --- |
| CharacterSetCode | Список кодов IANA, кодирующих набор символов, которые используются в качестве значения атрибута **characterSetCode** в типе данных UNCEFACT **xsd: base64Binary** |
| MIMEMediaType | Список кодов типа медиа, кодирующих набор символов, которые используются в качестве значения атрибута **mimeCode** в типе данных UNCEFACT **xsd:base64Binary** |

5 Доступ к сервису

5.1 Параметры обмена

Оборудование должно предоставлять оператору интерфейс для получения параметров обмена соответствующего поставщика услуг, которые включают в себя следующее:

а. URL поставщика услуги

б. Информация для аутентификации (пользователь/пароль, электронный ключ безопасности)

в. Идентификационный номер отправителя

г. Имя отправителя

д. Страна отправителя

е. Идентификационный номер получателя

ж. Имя получателя

з. Страна получателя

е. Тип месторасположения

к. Тип идентификатора месторасположения

л. Идентификатор месторасположения

м. Название месторасположения

н. Страна месторасположения

о. Тип основного идентификатора животного

п. Тип вторичных идентификаторов животного (например, идентификатор животного на ферме, имя животного)

р. Прочие специальные параметры изготовителя или поставщика услуги

Для конкретного поставщика услуги параметры обмена должны быть одинаковыми в отношении всех потребляемых услуг.

5.2 Передача данных

Так как сервисы не являются доступными на постоянной основе, запрос никогда не должен отправляться на оборудование, если не был проверен доступ к следующим компонентам:

а. Сеть

б. Сервер поставщика

в. Сам сервис

Все новая информация должна в максимально короткие сроки регистрироваться информационной системой.

Пока обрабатывается предыдущий набор данных, запрещается посылать новые данные в информационную систему. Пользователь сервиса должен дождаться результата обработки или мандата в случае асинхронной обработки.

Запрещено посылать мандат в информационную систему ранее времени, указанного информационной системой.

Запрещено отправлять любые сообщения, синтаксис которых не соответствует требованиям к синтаксису.

После включения данных в запрос и получения отклика данные считаются отправленными в информационную систему.

Поставщики услуг и потребители должны быть готовы к обновлениям и удалениям. Поставщик также должен принимать во внимание, что отправляющая сторона заявляет обновления в качестве операции создания, не зная о том, что эти данные уже присутствуют в базе данных поставщика услуг (например, в случае повторной передачи, инициированной потребителем).

Подготовка запроса в соответствии с коммерческими требованиями должна осуществляться отдельно от подготовки сообщения, синтаксис которого зависит от типа технического маппинга (в настоящий момент — веб-сервис с SOAP).

5.3 Шифрование

По возможности, для обеспечения приватности и подлинности должно использоваться шифрование. Сам процесс шифрования здесь не описывается. Должны применять возможности шифрования применяемого транспортного протокола, например HTTPS для транспортного HTTP.

5.4 Сжатие

Для минимизации задержки передачи больших сообщений, особенно в случае кодирования в xml, необходимо применять сжатие. Некоторые транспортные протоколы предлагают возможности сжатия, например прозрачное сжатие gzip для откликов http. Однако поскольку сжатие http применимо только к отклику, оно не используется для сжатия загружаемых данных в запросах http.

Для избежания данного ограничения для обмена данными ICAR описан специальный механизм сжатия на основе XML. Отдельные сообщения можно определить как подлежащие zip-сжатию, которое является альтернативой использованию несжатых или zip-сжатых бинарно закодированных Base64 xml-структур, используемых как для запроса, так и для отклика. Более подробная информация представлена в главе «Структура сообщения»

5.5 Верификация прав пользователя

Право конкретного пользователя на использование конкретного сервиса верифицируется либо с помощью **«аутентификационного ключа»**, который не позволяет ему посылать имя пользователя и пароль при каждом запросе, либо с помощью **имени пользователя, аутентифицируемого паролем.**

*Аутентификацию с помощью* имени пользователя и пароля *необходимо использовать только с зашифрованным транспортным протоколом.*

6 Рабочие процессы коммуникации

6.1 Обработка данных

Любой сервис может предоставляться либо в рельном времени, либо с задержкой.

Отклик на любой запрос должен приходить в течение нескольких секунд.

Любой запрос и любой отклик должны сохраняться оборудованием и информационной системой.

Требования к обновлению базы данных информационной системы разнятся в зависимости от поставщика услуги.

Запрос не обрабатывается, если он не соответствует одному из следующих условий:

а. Соблюдение синтаксиса

б. Действительность аутентификационного ключа

в. Действительность мандата, при его наличии

6.2 Спецификация запроса

Любой запрос должен состоять из трех частей:

а. Одиночный **заголовок сообщения**

(см. объект Заголовок сообщения (MessageHeader) типа “ADEExchangedDocumentType” в разделе «Описание данных»)

Отправитель запроса создает и предоставляет свой собственный уникальный Идентификационный номер сообщения (MessageID)

(Элемент ADEExchangedDocumentType.Identifier).

См. главу «Оптимальные способы создания MessageID».

б. Одиночный **стандартный запрос**

(см. объект «Стандартный запрос» (StandardRequest) в разделе «Описание данных»)

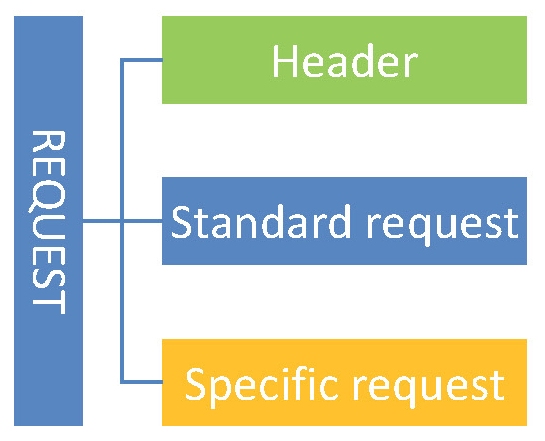
в. **Специальный запрос**, который может состоять из:

специального сообщения для определенного сервиса

или сжатая (zip) бинарная структура base64 специального сообщения

или мандат.

Мандат посылается с целью получения результатов обработки асинхронного сообщения (см. схемы ниже)



|  |  |
| --- | --- |
| Header | Заголовок |
| Standard request | Стандартный запрос |
| Specific request | Специальный запрос |
| REQUEST | ЗАПРОС |

Рисунок 3. Специальный запрос

6.3 Спецификация отклика

Любой отклик на сообщение должен состоять из трех частей:

а. Заголовок одиночного сообщения (см. объект Заголовок сообщения (MessageHeader) типа “ADEExchangedDocumentType” в разделе «Описание данных»)

Отправитель отклика создает и предоставляет свой собственный уникальный Идентификационный номер сообщения (MessageID).

(См. главу «Оптимальные способы создания MessageID»).

б. Одиночный стандартный отклик (см. объект «Стандартный отклик (StandardResponse)» в разделе «Описание данных») в стандартном отклике оригинальный MessageID от отправителя возвращается в элементе Идентификационный номер запроса (RequestID)

Стандартный отклик содержит элемент «Статус обработки запроса (RequestProcessingStatus)», который указывает статус обработки запроса на стороне поставщика.

Таблица 8. RequestProcessingStatusCodeType (base xsd:string)

| **Ключ** | **Описание** |
| --- | --- |
| О | Обработано без ошибок |
| P | Данные приняты для асинхронной обработки (клиент может позднее получить результат обработки при помощи полученного мандата) |
| E | Обработано и отказано по причине ошибок |
| W | Обработано с предупреждениями |

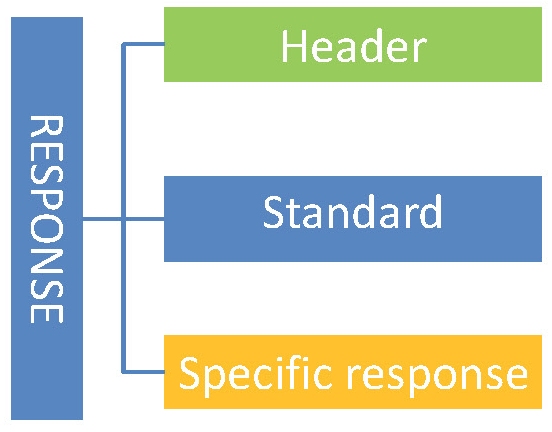
в. Специальный отклик, который может состоять либо из мандата, либо из специального сообщения конкретного сервиса, либо из zip-сжатой бинарной структуры base64 отдельного сообщения.

Существование и содержание конкретного отклика зависит от определения сервиса и от содержания элемента «Статус обработки запроса (RequestProcessingStatus)».

В случае если RequestProcessingStatus = ‘P’, то специальный отклик содержит мандат.

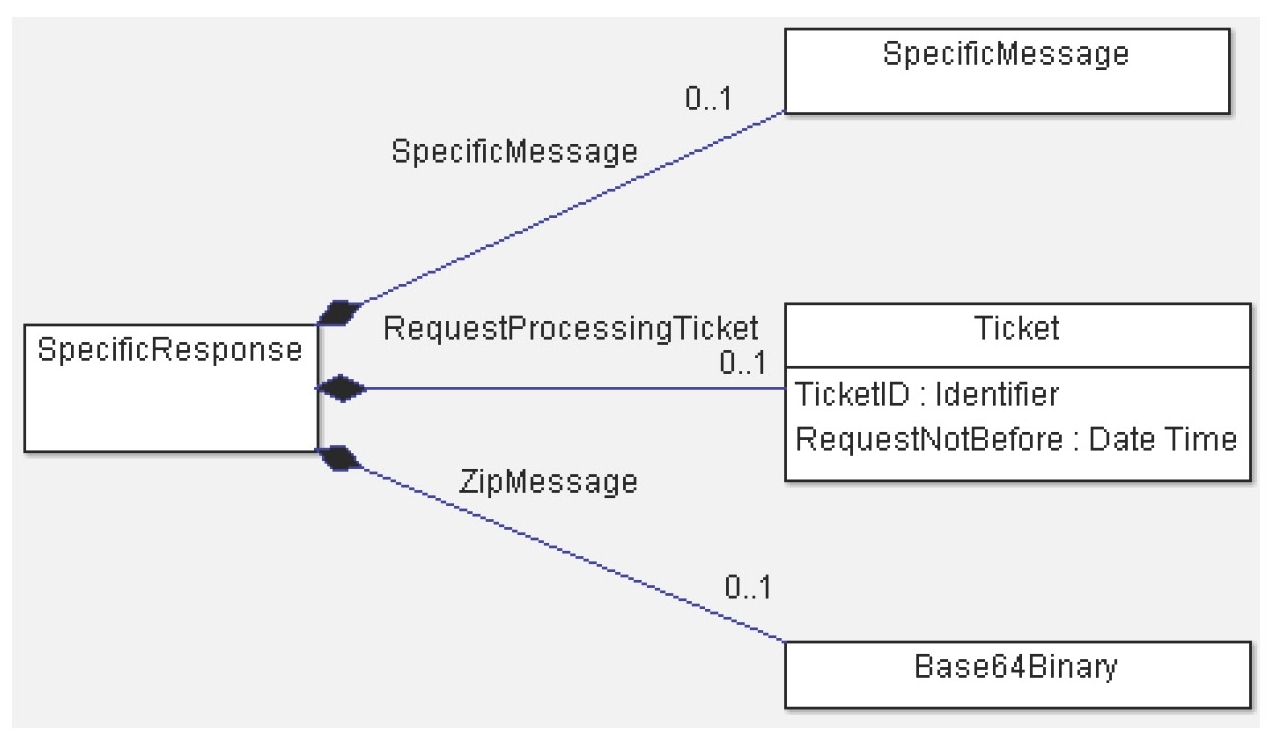
Если определено конкретное сообщение-отклик, оно может быть представлено в несжатом виде, либо в виде сжатого zip-сообщения типа Base64binary.

Если Статус обработки сообщения (RequestProcessingStatus) = ‘P’, ‘E’ или ‘W’, конкретный отклик может отсутствовать в зависимости от определения сообщения.



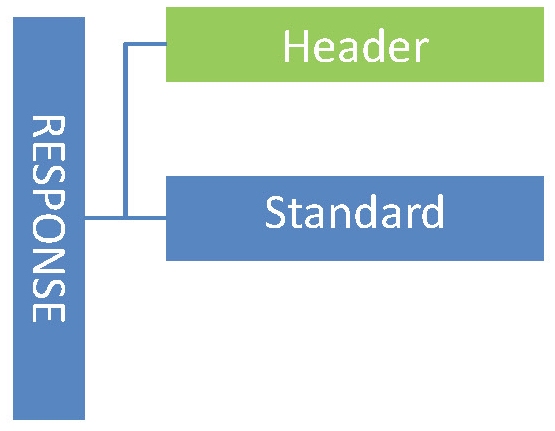
|  |  |
| --- | --- |
| Header | Заголовок |
| Standard | Стандарт |
| Specific response | Специальный отклик |
| RESPONSE | ОТКЛИК |

Рисунок 4. Специальный отклик



|  |  |
| --- | --- |
| SpecificMessage | Специальный отклик (SpecificMessage) |
| SpecificMessage | Специальный отклик (SpecificMessage) |
| RequestProcessingTicket | Мандат обработки запроса (RequestProcessingTicket) |
| Ticket | Мандат (Ticket) |
| TicketID : Identifier RequestNotBefore : Date Time | TicketID : Identifier RequestNotBefore : Дата / время |
| SpecificResponse | Специальный отклик (SpecificResponse) |
| ZipMessage | Zip-сообщение (ZipMessage) |
| Base64Binary | Base64Binary |

Рисунок 5. Схема специального отклика



|  |  |
| --- | --- |
| Header | Заголовок |
| Standard | Стандарт |
| RESPONSE | ОТКЛИК |

Рисунок 6. Стандартный отклик

6.4 Оптимальные способы создания MessageID

Идентификационный номер сообщения, создаваемый потребителем и поставщиком сервиса, должен быть глобально уникальным, чтобы обеспечивать простую идентификацию каждого сообщения (например, для целей отслеживания или отладки).

Идентификационный номер сообщения (MessageID) должен основываться на уникальном идентификаторе оборудования, уникальном идентификаторе программного обеспечения и на дополнительном отличительном компоненте, который является достаточно большим, чтобы обеспечить уникальную идентификацию сообщений, создаваемых с очень высокой скоростью. В виде такого динамического компонента может выступать последовательность или метка времени с точностью до миллисекунды. Уникальным идентификатором оборудования может быть глобально уникальный MAC-адрес сетевого интерфейса. Уникальный идентификатор программного обеспечения должен быть уникальным в системе, в которой запускается данное ПО.

В различных языках программирования имеются инструменты, предоставляющие простые инструменты для создания уникального идентификационного номера сообщения, например Guid.NewGuid() в C#

7 Работа с локальными требованиями

Процесс стандартизации ICAR ADE не может учитывать все локальные требования. В первую очередь он предлагает некую основу, на которой можно выстраивать процесс.

Понятие «локальный» прежде всего относится к национальным стандартам. Однако также возможно включение локальных расширений на основе договоренности между потребителем и поставщиком сервиса (например, для специальных проектов, прототипов и т. д.).

Для того, чтобы позволить существовать локальным расширениям и заполнить пробелы в стандартизации, которые невозможно предусмотреть в ICAR, спецификации ADE предоставляют два способа для локальных вариантов и расширений:

а. **Местные списки кодов:**

Местные списки кодов для всех элементов типа **udt: CodeType**, для которых ICAR в данный момент не в состоянии предоставить универсально принятый список кодов.

б. **Дополнительные локальные данные:**

Опциональный список ключевых пар значений в каждом отдельном объекте, который используется для элементов данных, необходимых на локальном уровне

Локальные вариации должны сводиться к минимуму, так как это противоречит цели ICAR по установлению глобальных стандартов, ведет к увеличению сложности, осложняет разработку программного обеспечения и приводит к дополнительным расходам.

7.1 Локальные списки кодов

Локальные списки кодов необходимо предоставлять для элементов типа **udt: CodeType** в контексте локального процесса стандартизации (например, коды пород).

Их содержание нельзя проверить с помощью процедуры валидации SOAP/XML, а необходимо оценивать на дополнительном шаге программы, прежде чем заполнять структуры данных XML. Разработчики программного обеспечения должны убедиться, что обмен данными соответствует спецификациям ICAR и локальным спецификациям.

Чтобы помочь производителям в поддержании и интеграции локальных списков кодов, эти спецификации определяют сервис **Получить локальный список кодов (GetLocalCodeList)**, который можно использовать для запроса локальных списков кодов у локальных поставщиков сервисов. Более подробная информация приведена в разделе «Технические сервисы».

7.2 Дополнительные локальные данные:

Иногда для локального применения сообщений ICAR к отдельным сообщениям необходимо добавить элементы локальных данных, которые неизвестны или не указаны в процессе спецификации ICAR.

Для таких сценариев к каждому конкретному объекту было добавлено простое и гибкое расширение (Дополнительные локальные данные (**LocalAdditionalData**)).

Оно состоит из опционального списка пар кодов/значений типа udt: textType, который можно заполнять в соответствии с местными определениями.

Владельцы локальных бизнес-процессов отвечают за ведение документации и обеспечение правильной работы с такими парами кодов/значений. Определения wsdl/xsd ICAR проверяют лишь правильность использования структуры XML, а не правильность ее содержания.

Подробнее это обсуждается в файлах wsdl/xsd и в разделе

«Общие компоненты» — «Типы дополнительных локальных данных (LocalAdditionalDataType)»

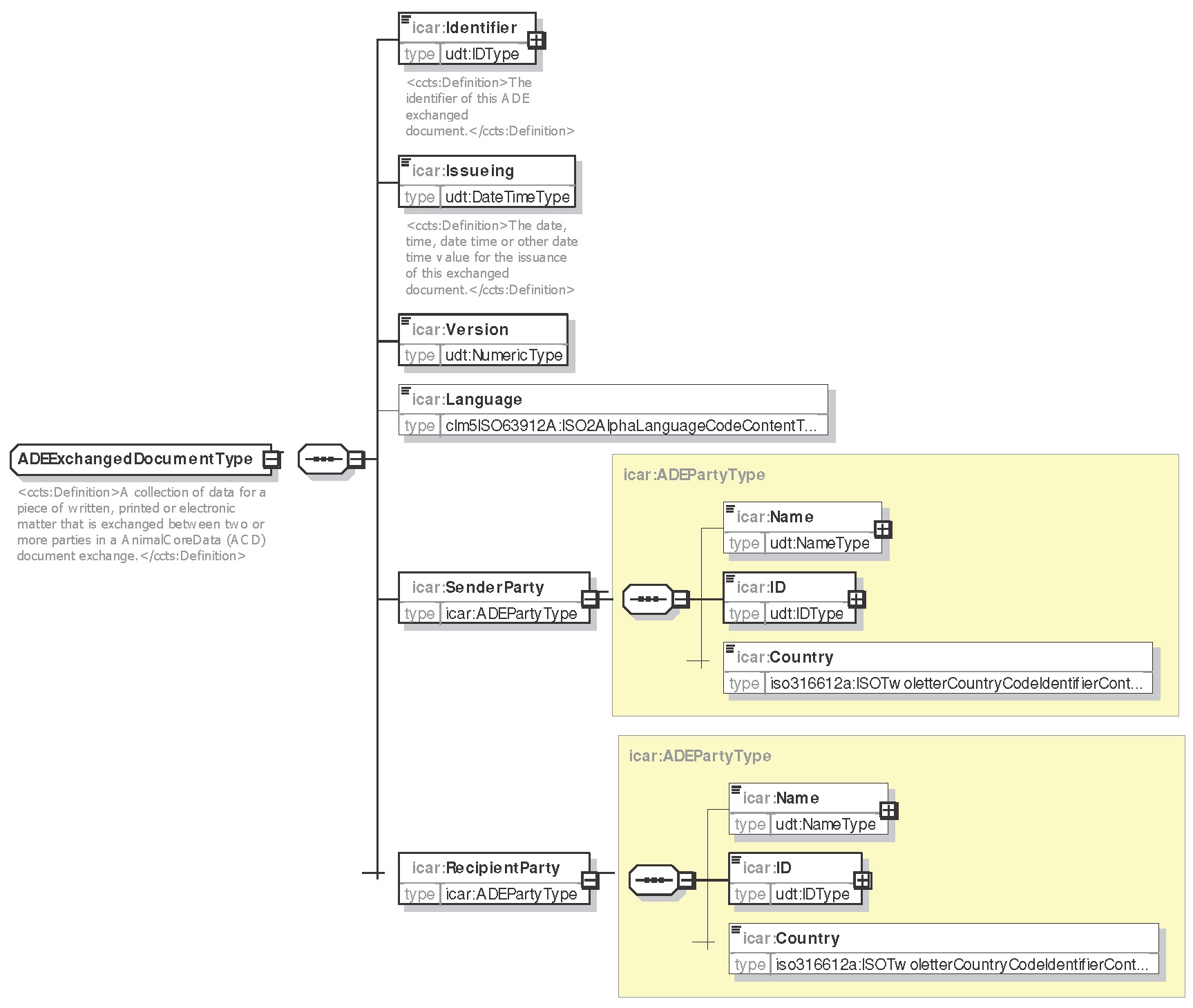
8 Общие компоненты

В данном разделе описываются общие компоненты, используемые во всех специальных сервисах, описанных в приложениях B, C и D.

8.1 Тип документа обмена ADE

Объект **«Тип документа обмена ADE (ADEExchangedDocumentType)»** (Рисунок 7 «Тип документа обмена ADE (ADEExchangedDocumentType)») дает информацию, описывающую общие аспекты сообщения.

Используется элементом данных: **Заголовок сообщения (MessageHeader)**



|  |  |
| --- | --- |
| icar:Identifier | icar: Identifier |
| udt:IDType | udt: IDType |
| <ccts:Definition>The identifier of this ADE exchanged document. </ccts:Definition> | <ccts:Definition>Идентификатор данного документа обмена с ADE. </ccts:Definition> |
| icar:lssueing | icar: lssueing |
| udt:DateTimeType | udt: DateTimeType |
| <ccts:Definition>The date, time, date time or other date time value for the issuance of this exchanged document. </ccts:Definition> | <ccts:Definition>Дата, время, дата/время или иной показатель даты/времени для выдачи документа обмена. </ccts:Definition> |
| icar:Version | icar: Version |
| udt: Numeric Type | udt: Numeric Type |
| icar:Language | icar: Language |
| clm5ISQ63912A:ISO2AlphaLanguageCodeContentT... | clm5ISQ63912A:ISO2AlphaLanguageCodeContentT... |
| ADEExchanged Document Type | ADEExchanged Document Type |
| <ccts:Definition>A collection of data fora piece of written, printed or electronic matter that is exchanged between two or more parties in a A nimalC ore Data (ACD) document exchange. < /ccts: Definition > | <ccts:Definition>Набор данных для письменного, печатного или электронного носителя информации, которым обмениваются две или более сторон в документе Основные данные о животных (Anival Core Data, ACD). < /ccts: Definition > |
| icar:ADEPartyType | icar: ADEPartyType |
| icar: Name | icar: Name |
| icar:SenderParty | icar: SenderParty |
| icar:ADEPartyType | icar: ADEPartyType |
| udt:NameType | udt: NameType |
| icar:ID | icar: ID |
| udt:IDType | udt: IDType |
| icar:Country | icar: Country |
| iso316612a:IS0TwoletterCountryCodeldentifierCont... | iso316612a:IS0TwoletterCountryCodeldentifierCont... |
| icar:ADEPartyType | icar: ADEPartyType |
| icar:Name | icar: Name |
| udt:NameType | udt: NameType |
| icar:RecipientParty | icar: RecipientParty |
| icar:ADEPartyType | icar: ADEPartyType |
| icar:ID | icar: ID |
| udt:IDType | udt: IDType |
| icar:Country | icar: Country |
| iso316612a:ISOTwoletterCountryCodeldentifierCont... | iso316612a:ISOTwoletterCountryCodeldentifierCont... |

Рисунок 7. Тип документа обмена ADE (ADEExchangedDocumentType)

а. **Идентификатор (Identifier):** Уникальный идентификатор сообщения, присвоенный отправителем

б. **Формирование (Issueing):** Дата и время формирования сообщения.

в. **Версия (Version):** Номер версии сообщения

г. **Язык (Language):** Язык, используемый в сообщении

д. **Дополнительные локальные данные (LocalAdditionalData):** Опциональный список пар ключей/значений, используемых в локальном контексте, как это описано в разделе «Общие компоненты — Локальные адаптации»

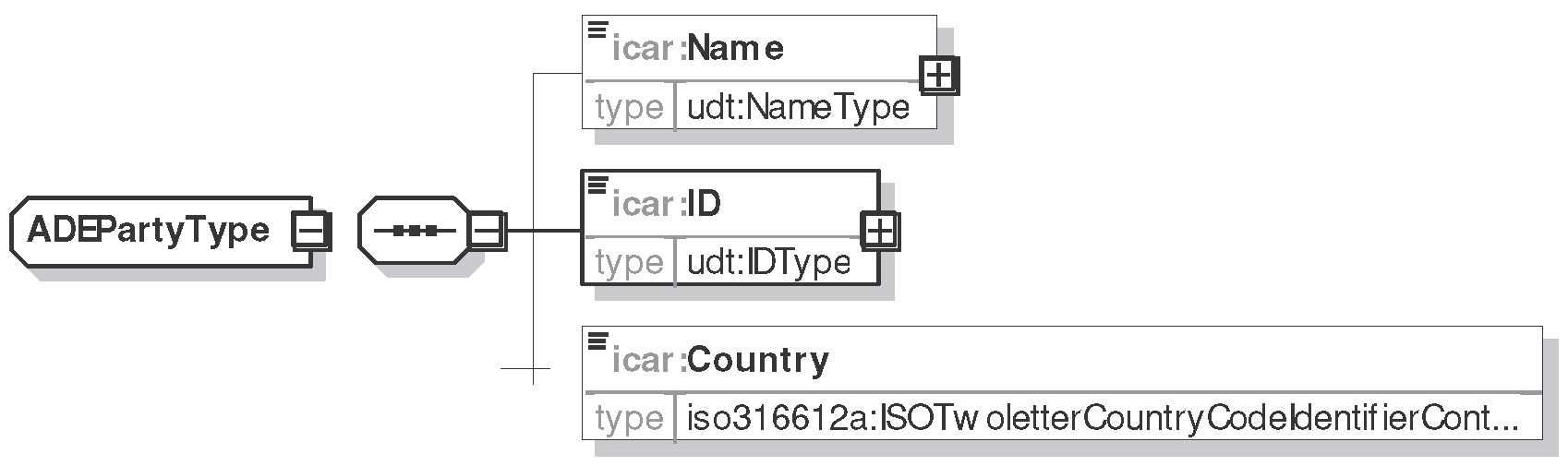
е. **Сторона-отправитель (SenderParty):** Организация или лицо, ответственные за содержание сообщения (не является идентификатором сервера).

ж. **Сторона-получатель (RecipientParty):** Организация или лицо, ответственные за обработку сообщения (не является идентификатором сервера).

8.2 Тип стороны ADE

Объект **«Тип стороны ADE (ADEPartyType)»** (Рисунок 8 «Тип стороны ADE (ADEPartyType)») предоставляет имя и идентификатор стороны, которая может быть либо отправителем, либо получателем сообщения.

Используется элементами данных: **Сторона-отправитель (SenderParty), Сторона-получатель (RecipientParty)**



|  |  |
| --- | --- |
| ADEPartyType | Тип стороны ADE (ADEPartyType) |
| icar:Name | icar: Name |
| udt:NameType | udt: NameType |
| icar:ID | icar: ID |
| udt:IDType | udt: IDType |
| icar:Country | icar: Country |
| iso316612a:ISOTwoletterCountryCodeldentif ierCont... | iso316612a:ISOTwoletterCountryCodeldentif ierCont... |

Рисунок 8. Тип стороны ADE (ADEPartyType)

**• Имя (Name):** Идентификатор указанной стороны

**• Идентификационный номер (ID):** Идентификатор указанной стороны

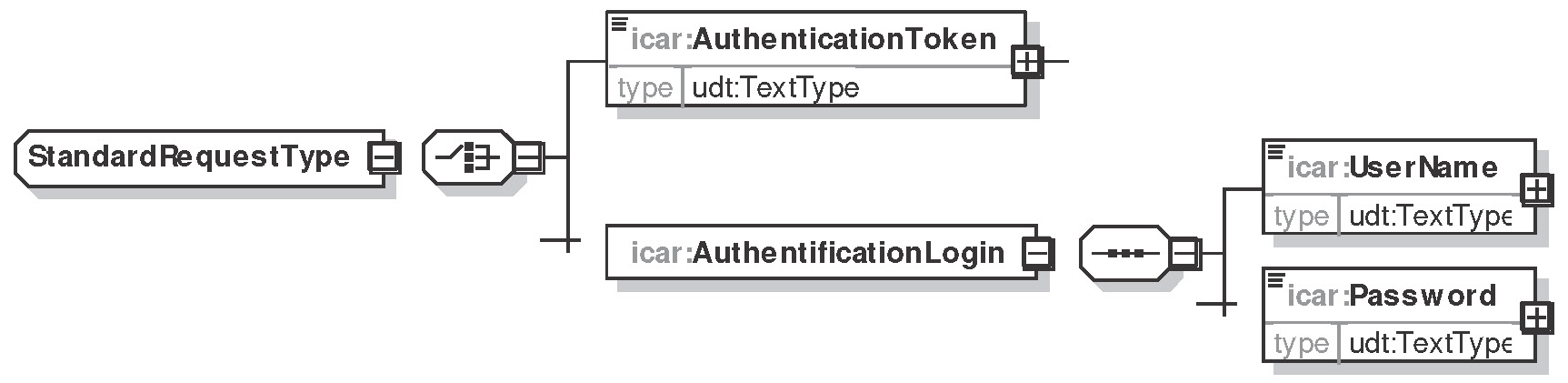
**• Страна (Country):** Страна указанной стороны в соответствии со списком кодов **ISOTwoletterCountryCodeIdentifierContent**

8.3 Тип стандартного запроса

Объект **Тип стандартного запроса (StandardRequestType)** (Рисунок 9 «Тип стандартного запроса (StandardRequestType)») представляет стандартное содержание запроса.

Могут использоваться различные типы аутентификации, на основе электронного ключа безопасности или на основе имени пользователя/пароля

Используется элементом данных: **Стандартный запрос (StandardRequest)**



|  |  |
| --- | --- |
| StandardRequestType | Стандартный тип запроса (StandardRequestType) |
| icar:AuthenticationToken | icar: AuthenticationToken |
| udt:TextType | udt: TextType |
| icar:AuthentificationLogin | icar: AuthentificationLogin |
| icar:UserName | icar: UserName |
| udt:TextTypc | udt: TextTypc |
| icar Password | icar Password |
| udt:TextType | udt: TextType |

Рисунок 9. Стандартный тип запроса (StandardRequestType)

а. **Аутентификационный электронный ключ безопасности (AuthenticationToken:):** временный код, используемый для верификации личности пользователя и его права пользоваться сервисом.

б. **Аутентификационные данные для входа (AuthentificationLogin):** Пара из имени пользователя и пароля, используемая для входа в систему

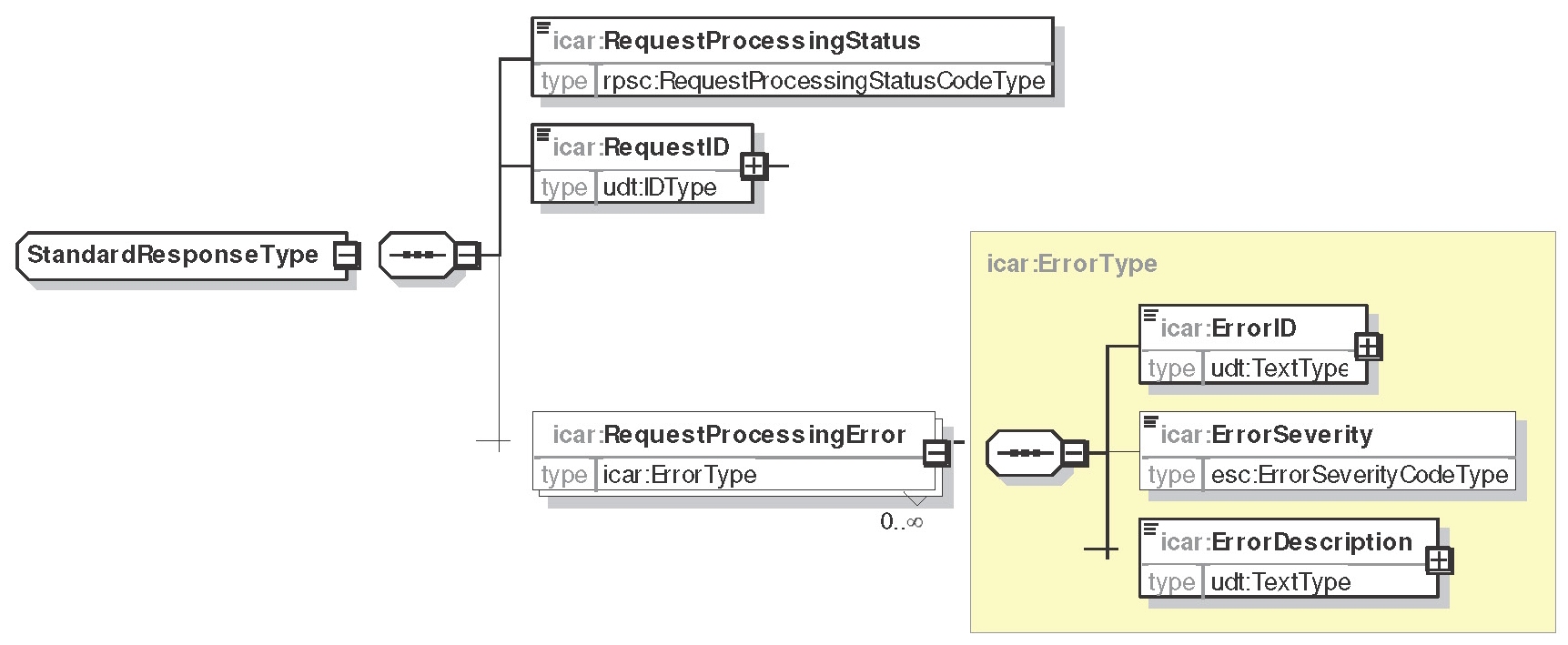
в. **Имя пользователя (UserName):** Имя пользователя, использующееся для аутентификации

г. **Пароль (Password):** Пароль, использующийся для аутентификации

8.4 Тип стандартного отклика

Объект **Тип стандартного отклика (StandardResponseType)** (Рисунок 10 «Тип стандартного отклика (StandardResponseType)») представляет стандартное содержание отклика.

Используется элементом данных: **Стандартный отклик (StandardResponse)**



|  |  |
| --- | --- |
| StandardResponseType | Тип стандартного отклика (StandardResponseType) |
| icar:RequestProcessingStatus | icar: RequestProcessingStatus |
| rpsc:RequestRocessingStatusCodeType | rpsc: RequestRocessingStatusCodeType |
| icar:RequestlD | icar: RequestlD |
| udt:IDType | udt: IDType |
| icar:ErrorType | icar: ErrorType |
| icar:ErrorlD | icar: ErrorlD |
| udt:TextType | udt: TextType |
| icar:RequestProcessingError | icar: RequestProcessingError |
| icar:ErrorType | icar: ErrorType |
| icar:ErrorSeverity | icar: ErrorSeverity |
| esc:ErrorSeverityCodeType | esc: ErrorSeverityCodeType |
| icar:ErrorDescription | icar: ErrorDescription |
| udt:TextType | udt: TextType |

Рисунок 10. Тип стандартного отклика (StandardResponseType)

а. **Статус обработки запроса (RequestProcessingStatus):** Статус обработки запроса, возвращенный сервером в соответствии со списком кодов **Код статуса обработки запроса (RequestProcessingStatusCode)**

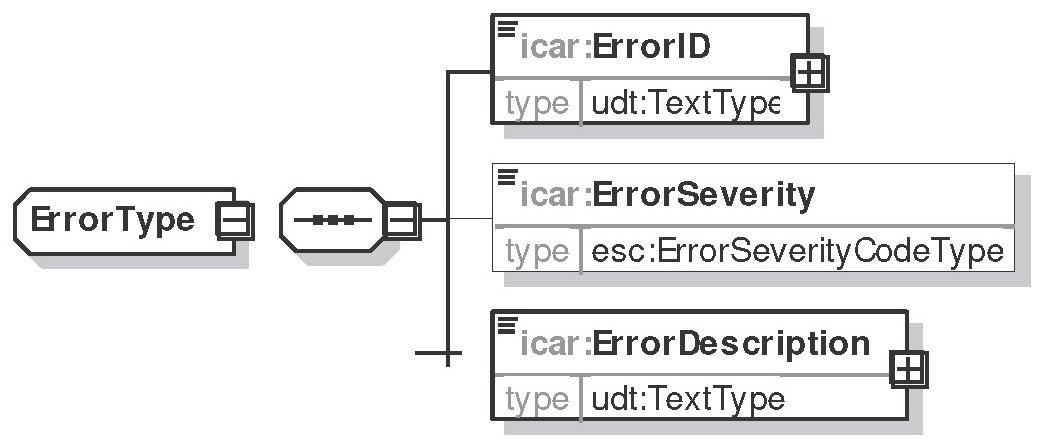
б. **Идентификационный номер запроса (RequestID)**: Копия идентификации сообщения в элементе **Идентификатор заголовка сообщения (MessageHeader.Identifier) запроса**

в. **Ошибка обработки запроса (RequestProcessingError):** В случае возникновения ошибки в процессе обработки запроса, описание ошибки содержится здесь

8.5 Тип ошибки (Error Type)

Объект **Тип ошибки (ErrorType)** отображает ошибки обработки запроса.

Используется элементом данных: **Ошибка обработки запроса (RequestProcessingError)**



|  |  |
| --- | --- |
| ErrorType | Тип ошибки (Error Type) |
| icar:ErrorlD | icar: ErrorlD |
| udt:TextType | udt: TextType |
| icar:ErrorSeverity | icar: ErrorSeverity |
| esc:ErrorSeverityCodeType | esc: ErrorSeverityCodeType |
| icar:ErrorDescription | icar: ErrorDescription |
| udt:TextType | udt: TextType |

*Рисунок 11. Тип ошибки (Error Type)*

а. **Идентификатор ошибки (ErrorID):** Идентификатор ошибки, например, текстовое сообщение ошибки, локальный код ошибки и т. д.

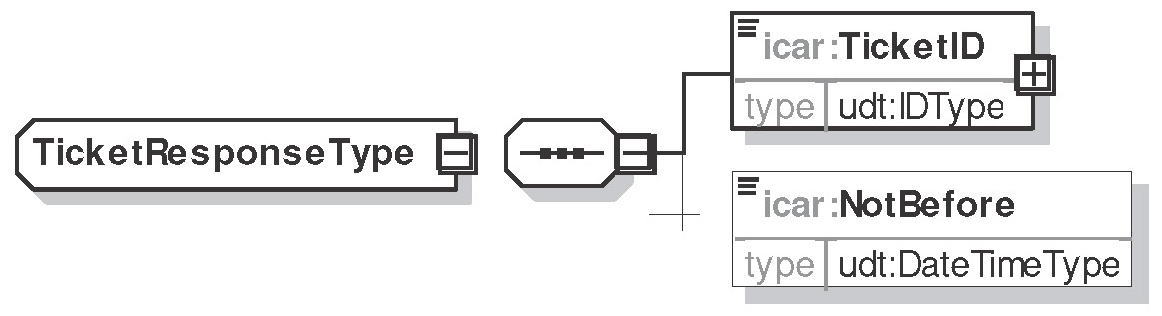
б. **Критичность ошибки (ErrorSeverity):** Критичность ошибки согласно списку кодов **Код серьезности ошибки (ErrorSeverityCode)**

в. **Описание ошибки (ErrorDescription):** Описание ошибки в текстовом виде, понятном человеку

8.6 Тип отклика мандата (Ticket Response Type)

Объект **Тип отклика мандата (Ticket Response Type)** (Рисунок 12 «Тип отклика мандата (Ticket Response Type)») передает мандат с целью получения отклика, обрабатываемого с задержкой (режим асинхронной обработки).

Используется в **Специальном запросе (SpecificRequest)** и **Специальном ответе (SpecificResponse)** элементом данных **Отклик мандата (TicketResponse).**



|  |  |
| --- | --- |
| TicketResponseType | Тип отклика мандата (TicketResponseType) |
| icar TicketID | icar: TicketID |
| udt:IDType | udt: IDType |
| icar:NotBefore | icar: NotBefore |
| udt:DateTimeType | udt: DateTimeType |

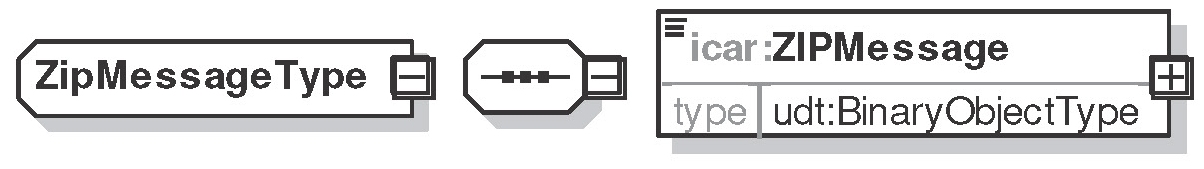
*Рисунок 12. Тип отклика мандата (TicketResponseType)*

а. **Идентификационная информация мандата (TicketID):** Идентификация отклика, который необходимо получить

б. **Не ранее (NotBefore):** Дата и время возможной доступности ответа

8.7 Тип ZIP-сообщения

**Тип ZIP-сообщения (ZipMessageType)** (Рисунок 13 «Тип ZIP-сообщения (ZipMessageType)») — тип оболочки, предназначенный для транспортировки сжатой zip-версии конкретной части данных сообщения. В отличие от метода сжатия http, который можно использовать только для части отклика, этот метод применим и к запросу, и к отклику. Он определяется как выбор для каждого отдельного запроса или отклика, который транспортирует объекты данных.



|  |  |
| --- | --- |
| ZipMessageType | Тип ZIP-сообщения (ZipMessageType) |
| icar:ZIPMessage | icar: ZIPMessage |
| udt:BinaryObiectType | udt: BinaryObiectType |

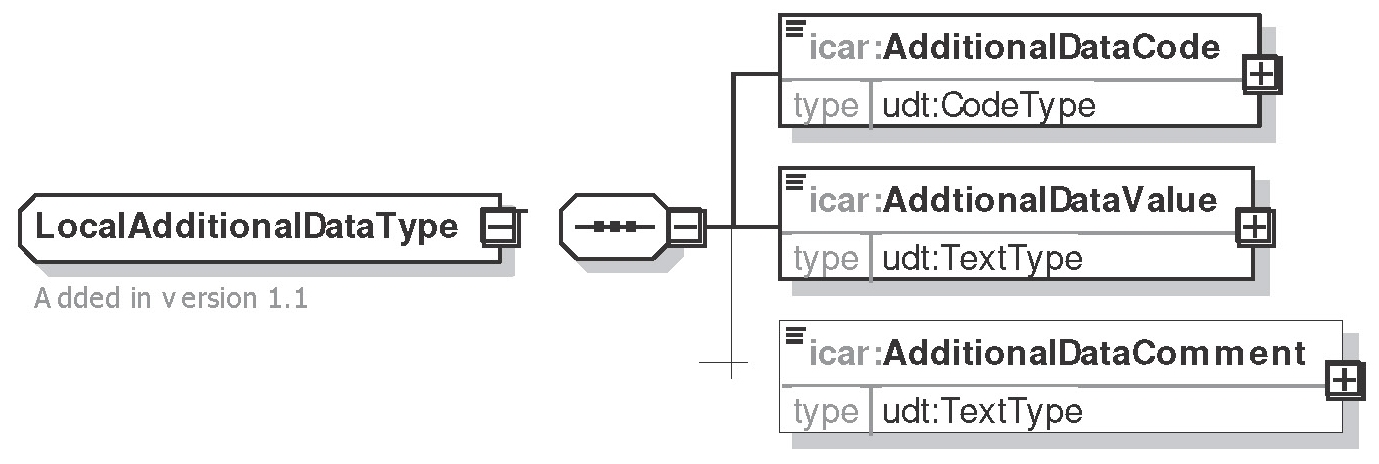
Рисунок 13. Тип ZIP-сообщения (ZipMessageType)

а. **Zip-ссобщение (ZIPMessage):** данные, сжатые при помощи zip

8.8 Тип дополнительных локальных данных

Объект **Тип дополнительных локальных данных (LocalAdditionalDataType)** (Рисунок 14 «Тип дополнительных локальных данных (LocalAdditionalDataType)») является контейнером для пар значений кода с целью локального применения параметров, не входящих в спецификации ICAR.

Используется в качестве элемента опционального списка **LocalAdditionalData** в каждом отдельном объекте.



|  |  |
| --- | --- |
| LocalAdditionalDataType | Тип дополнительных локальных данных (LocalAdditionalDataType) |
| Added in version 1.1 | Добавлено в версии 1.1 |
| icar:AdditionalDataCode | icar: AdditionalDataCode |
| udt:CodeType | udt: CodeType |
| icar:AddtionalDataValue | icar: AddtionalDataValue |
| udt:TextType | udt: TextType |
| icar:AdditionalDataComment | icar: AdditionalDataComment |
| udt:TextType | udt: TextType |

Рисунок 14. Тип дополнительных локальных данных (LocalAdditionalDataType)

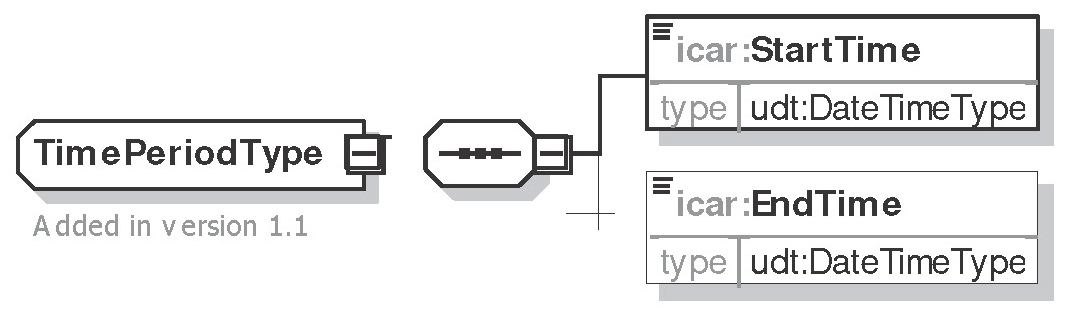
а. **Код дополнительных данных (AdditionalDataCode):** Имя пары код/значение

б. **Значение дополнительных данных (AdditionalDataValue):** Значение пары код/значение

в. **Комментарии к дополнительным данным (AdditionalDataComment):** Опциональное описание локальных данных

8.9 Тип периода времени

Утилита типа **Тип периода времени (TimePeriodType)** (Рисунок 15 «Тип периода времени (TimePeriodType)») — тип оболочки, предназначенный для транспортировки начала и конца периода времени.



|  |  |
| --- | --- |
| TimePeriodType | Тип периода времени (TimePeriodType) |
| icar:StartTime | icar: StartTime |
| udt:DateTimeType | udt: DateTimeType |
| icar:EndTime | icar: EndTime |
| udt:DateTimeType | udt: DateTimeType |
| Added in version 1.1 | Добавлено в версии 1.1 |

*Рисунок 15. Тип периода времени (TimePeriodType)*

а. **Время начала (StartTime):** Начало периода времени

б. **Время конца (EndTime):** Конец периода времени (опционально)

9 Правила наименования

В именах элементов данных, операций и сообщений используется Camel-регистр (слитное написание слов с заглавными буквами).

Имя сервиса состоит из:

а. Префикса, обозначающего тип операции в соответствии со следующими принципами:

- Создать (Create): Внести данные в базу данных

- Обновить (Update): обновить базу данных

- Получить (Get): получить данные из базы данных

- Удалить (Delete): удалить данные из базы данных

б. Название сервиса

Например, Обновить результат доения (UpdateMilkingResult) — это сервис, позволяющий обновить базу данных в результате обмена данными о результатах доения

Имя сообщения состоит из:

а. Названия сервиса

б. Суффикса, который указывает тип сообщения:

- Запрос

- Отклик

Например, Запрос на обновление результата доения (UpdateMilkingResult) — это запрос для вызова сервиса Обновление результата доения (UpdateMilkingResult).

10 Список литературы

1. Семантика интеллектуального управления молочной фермой: стандарт регистрации производства молока — SDF, июнь 2013 г.

2. Руководство пользователя UN/UNCEFACT по методологии моделирования (CEFACT/TMG/N093)

3. Шаблон документа спецификаций бизнес-требований UN/UNCEFACT (CEFECT/ICG/005)

4. ISO 11787: Электронный обмен данными между информационными системами в области сельского хозяйства — Синтаксис обмена сельскохозяйственными данными

5. ISO 11788: Электронный обмен данными между информационными системами в области сельского хозяйства — Словарь элементов сельскохозяйственных данных — Часть 1: Общее описание — Часть 2: Молочные фермы

6. ISO 17532: Стационарное сельскохозяйственное оборудование — Сеть для обмена данными животноводческих ферм

7. ISO 11784: Радиочастотная идентификация животных — структура кода.