ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 57873— 2017

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ. СИСТЕМА TV-ANYTIME. ПРОТОКОЛЫ ПЕРЕДАЧИ МЕТАДАННЫХ ПО ДВУНАПРАВЛЕННОЙ СЕТИ

Основные параметры

[ETSI TS 102 822-6-1 V1.8.1 (2012-12), NEQ]

Издание официальное

Москва
Стандартинформ
2020

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Автономной некоммерческой организацией «Научно-технический центр информатики» (АНО «НТЦИ»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 480 «Связь»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 31 октября 2017 г. № 1587-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений стандарта Европейского института по стандартизации в области телекоммуникаций (ETSI) ЕТСИ ТС 102 822-6-1 V1.8.1 (2012-12) «Широковещательные и on-line услуги: поиск, выбор и использование контента в личных системах хранения («TV-Anytime»). Часть 6. Доставка метаданных по двунаправленной сети. Подраздел 1. Обслуживание и транспортирование» [ETSI TS 102 822-6-1 V1.8.1 (2012-12) «Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems («TV-Anytime»); Part 6: Delivery of metadata over a bi-directional network; Sub-part 1. Service and transport», NEQ]

5 ВВЕДЕН ВПЕРВЫЕ

6 ПЕРЕИЗДАНИЕ. Март 2020 г.

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

© Стандартинформ, оформление, 2017, 2020

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

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины, определения, сокращения и символы 1
4 Обмен метаданными системы TV-Anytime в двунаправленных сетях 4
5 Типы веб-служб метаданных 5
6 Транспортные протоколы 6
6.1 Вводная часть 6
6.2 Протокол SOAP 6
6.3 Протокол HTTP 10
6.4 Инкапсуляция метаданных 10
6.5 Кодирование метаданных 10
6.6 Службы безопасности метаданных 11
Приложение А (обязательное) Формальное определение служб метаданных 12
Библиография 17

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ. СИСТЕМА TV-ANYTIME. ПРОТОКОЛЫ ПЕРЕДАЧИ МЕТАДАННЫХ ПО ДВУНАПРАВЛЕННОЙ СЕТИ

Основные параметры

Digital video broadcasting. TV-Anytime system.
Metadata transmission protocols over a bi-directional network. Basic parameters

Дата введения — 2018—08—01

1 Область применения

Настоящий стандарт является одним из серии стандартов, нормирующих систему TV-Anytime, в том числе архитектуру и параметры составных частей этой системы.

Настояший стандарт нормирует параметры протоколов передачи метаданных между устройствами TV-Anytime по двунаправленной сети и описывает возможности служб метаданных системы TV-Anytime. Использование стандарта в оборудовании и программном обеспечении каналов и сетей TV-Anytime создает условия для эффективной эксплуатации служб системы TV-Anytime.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 52210 Телевидение вещательное цифровое. Термины и определения

ГОСТ Р 56476 Телевидение вещательное цифровое. Система TV-Anytime. Схемы метаданных. Основные параметры

ГОСТ Р 56952 Телевидение вещательное цифровое. Система TV-Anytime. Передача метаданных по двунаправленной сети. Основные параметры

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины, определения, сокращения и символы

3.1 В настоящем стандарте применены термины по ГОСТ Р 52210, а также следующие термины с соответствующими определениями:

3.1.1 **агрегатор** (aggregator): В контексте настоящего стандарта — объект (организация, программа или группа программ), собирающий и обрабатывающий информацию о контенте, службах и их поставщиках.

Издание официальное

3.1.2 **веб-служба** (web service): Программная система со стандартизированными интерфейсами, идентифицируемая веб-адресом.

3.1.3 **группа программ** (programme group): Одна или несколько программ, сгруппированные вместе по определенному признаку.

3.1.4 **двунаправленная сеть** (bi-directional network): Сеть, поддерживающая доставку данных по двум направлениям при топологиях «точка — точка», один ко многим и многие ко многим.

3.1.5 **идентификатор ссылки контента** (content reference identifier; CRID): Идентификатор контента, который не зависит от его местоположения.

3.1.6 **данные пользователя (телевизионной информации)** (user data): Данные, передаваемые по цифровому тракту вещательного телевидения вместе с видеоинформацией, звуковой и сервисной информацией и не зависящие от передаваемых телевизионных программ.

3.1.7 **интернет-протокол** (Internet Protocol; IP): межсетевой протокол.

3.1.8 **канал** (channel): Прикладное представление открытого подключения протокола управления передачей (TCP).

3.1.9 **контент** (content): Видео- и аудиофайлы, к которым клиент хотел бы получить доступ и которые могут быть сохранены на PDR.

3.1.10 **локация** (locator): Время и место, в котором элемент контента может быть приобретен.

3.1.11 **метаданные** (metadata): Данные о контенте (название, жанр и резюме телевизионной программы).

Примечание — В контексте TV-Anytime метаданные также включают данные профиля клиента и истории клиента.

3.1.12 **обратный канал** (return path): Канал в двунаправленной системе от клиента до сервера провайдера службы.

3.1.13 **орган** (authority): Организация, которая создает CRID.

3.1.14 **перечисленный тип** (enumerated type): В программировании тип данных, множество значений которых представляет собой ограниченный список идентификаторов.

3.1.15 **пользователь** (user): Оконечная система, которая может передавать или принимать информацию от других таких же оконечных систем с использованием сети. Пользователь может функционировать как клиент, сервер или как клиент и сервер одновременно.

3.1.16 **предикáт** (predicat): Функция утверждения или отрицания чего-либо о субъекте.

3.1.17 **приложение** (application): Специфический набор функций, работающих на PDR.

Примечание — Некоторые приложения используют метаданные либо автоматически, либо под контролем клиентов.

3.1.18 **провайдер** (provider): Поставщик интерактивных услуг для PDR.

3.1.19 **провайдер контента** (content provider): Объект, который выступает в качестве агента и является главным пользователем контента.

3.1.20 **провайдер службы** (service provider). Агрегатор и поставщик контента.

3.1.21 **программа** (programme): Отредактированная, логически целостная (связанная) часть контента.

Примечание — Как правило, программа принимается PDR в целом.

3.1.22 **профиль клиента** (consumer profile): Профиль клиента включает описание клиента (предпочтения клиента, история использованных просмотров), информацию о клиенте (например, биографическая информация) и описание среды клиента (например, информация об устройстве клиента, сетевой среде клиента).

3.1.23 **сервер** (server): Программный объект, экспортирующий ресурс имеющихся данных. Программный объект устанавливается на физическое устройство. Компьютер, подключенный к сети и предоставляющий услуги другим устройствам, работающим в этой сети.

3.1.24 **создатель контента** (content creator): Производитель контента.

3.1.25 **разрешение локации** (location resolution): Процесс установления адреса (местонахождения и времени) конкретного экземпляра контента по CRID.

3.1.26 **синтаксис** (syntax): Часть языка программирования, которая описывает структуру программ как наборов символов.

3.1.27 **система метаданных (**metadata system): Набор правил, описывающих синтаксис и семантику метаданных.

3.1.28 **служба метаданных** (metadata service): Служба, которая предоставляет данные TV-Anytime, используя сервер в двунаправленной сети.

Примечание — Форматы протоколов, используемых для доставки данных, определяются настоящим стандартом.

3.1.29 **ссылка контента** (content reference): Указатель конкретного элемента контента.

3.1.30 **схема метаданных** (metadata schema): Идентификатор, ассоциированный с набором схем расширяемого языка разметки (Extensible Markup Language; XML), который глобально идентифицирует эти схемы.

Примечание — Глобальное уникальное пространство имен гарантирует, что имена типов определяемых схем в этом пространстве имен не конфликтуют с такими же именами в других местах.

3.1.31 **схема XML** (XML schema): Язык описания структуры документа XML.

3.1.32 **транзакция** (transaction): Передача коротких сообщений в диалоговом режиме.

3.1.33 **формат текстового кодирования UTF-8** (Unicode Transformation Format-8): Формат кодирования текста, который позволяет хранить символы юникода, используя переменное количество байт.

3.1.34 **юникод** (Unicode): Стандарт кодирования символов, позволяющий представить знаки основных письменных языков.

3.1.35 **элемент контента** (content item): Объект, который может быть приобретен как единое целое, например, файл аудио-видео, аудиопоток.

3.2 В настоящем стандарте применены следующие сокращения:

CRID — Content Reference IDentifier — идентификатор ссылки контента;

DNS — Domain Name System — система доменных имен;

DS — Description Scheme — схема описания;

EPG — Electronic Programme Guide — электронный путеводитель программ;

ETSI — European Telecommunications Standards Institute — Европейский институт по стандартизации в области телекоммуникаций;

GET — Get-запрос — используется для запроса содержимого, указанного ресурса по протоколу http;

HTTP — HyperText Transfer Protocol — протокол передачи гипертекста;

HTTP/1.0 — HyperText Transfer Protocol 1.0 — протокол передачи гипертекста, версия 1.0;

HTTP/1.1 — HyperText Transfer Protocol 1.1 — протокол передачи гипертекста, версия 1.1;

HTTPS — Secure Hyper Text Transfer Protocol — защищенный протокол передачи гипертекста;

IP — Internet Protocol — интернет-протокол межсетевого взаимодействия.

Примечание — Общее наименование сетевых протоколов, применяемых в сети Интернет;

PDR — Personal Digital Recorder — персональный цифровой рекордер;

PID — Packet Identifier — идентификатор типа пакета;

POST — наименование запроса к серверу HTTP;

SOAP — Simple Object Access Protocol — простой протокол доступа к объектам;

SSL — Secure Sockets Layer — уровень защищенных разъемов (криптографический протокол, который обеспечивает безопасность связи. SSL использует асимметричную криптографию для аутентификации ключей обмена);

TCP — Transmission Control Protocol — протокол управления передачей (из стека протоколов TCP/IP);

TCP/IP — набор (стек) протоколов сетевого и транспортного уровня;

TLS — Transport Layer Security — безопасность транспортного уровня;

UDDI — Universal Description Discovery and Integration — универсальное описание, обнаружение и интеграция;

URL — Uniform Resource Locator — универсальный указатель (локация) ресурса;

W3C — World Wide Web Consortium — WWB консорциум;

WSDL — Web Services Description Language — язык описания веб-служб;

WS — Web Services Inspection language — система поиска веб-служб;

XML — Extensible Markup Language — расширяемый язык разметки;

XSD — XML Schema Definition — определение схемы XML.

3.3 В настоящем стандарте применены следующие основные обозначения:

«Accept-Encoding» — заголовок согласования формата сжатия между сервером HTTP и клиентом;

«Clear_Personal_Data» — наименование операции удаления информации о клиенте, исходящей из операции «Upload_Personal_Data» в двунаправленной сетевой среде;

«ContentReferencingTable» — экземпляр документа XML обеспечивает формат инкапсуляции для фрагментов контента;

«ErrorReport» — элемент, содержащий сообщение от сервера об ошибках прикладного уровня;

«ExtendedUserDescriptionType» — тип данных, содержащих расширенное описание метаданных пользователя, включая «UsageHistory» и «UserPreferences»;

«fault's Detail» — элемент, содержащий детализированные причины отказа SOAP;

«get_Data» — наименование службы, выполняющей операцию «get_Data» обнаружения данных,

«ProgramInformationTable» — элемент, содержащий таблицу записей информации о программе в соответствии с ГОСТ Р 56476;

«ProgramReviewTable» — элемент, содержащий таблицу записей анализов программы в соответствии с ГОСТ Р 56476;

«SOAPAction» — поле заголовка HTTP содержит имя вызываемой операции;

«SOAP Fault» — сообщение сервера об отклонении запроса содержащего атрибутом агента SOAP в заголовке SOAP и имя элемента;

«Submit_Data» — наименование службы, выполняющей операцию предоставления данных;

«Upload_Personal_Data» — обозначение операции, которая предназначена для предоставления любой персональной информации;

«UsageHistory» — обозначение схемы описания информации об истории использования контента пользователем за длительный период времени;

«UserPreferences» — обозначение схемы описания информации о предпочтениях клиента;

«TVAMain» — экземпляр документа XML, который обеспечивает формат инкапсуляции фрагментов контента.

Примечание — В тексте настоящего стандарта наименования операций, схем описания, служб, сообщений, типов данных, которыми обмениваются устройства TV-Anytime по двунаправленным сетям, для удобства чтения выделены кавычками «».

4 Обмен метаданными системы TV-Anytime в двунаправленных сетях

Устройства, работающие в системе TV-Anytime, могут обмениваться метаданными контента, информацией о поиске контента по ссылке, метаданными клиента.

Настоящий стандарт определяет параметры протоколов передачи данных, которые используются в процессе обмена данными между клиентами TV-Anytime и серверами метаданных по двунаправленной сети при использовании обратного канала. Клиента в системе TV-Anytime представляет PDR, однако его может представлять любое устройство, соединенное с сетью Интернет. У таких устройств могут отсутствовать возможности вывода контента на экран или сохранения контента. В этом случае они могут использовать службы метаданных TV-Anytime, например мобильный телефон, выводящий на экран EPG.

Метаданные контента и информация контента по ссылке могут доставляться в одном направлении по каналам традиционного вещания средствами многоадресной передачи IP с использованием оборудования двунаправленной сети.

Целесообразность поставки данных провайдером TV-Anytime через двунаправленную сеть определяется следующими факторами:

- поставка более полного набора метаданных при малых ограничениях пропускной способности;
- для источников данных TV-Anytime возможность поставки метаданных клиентам, не имеющим доступа к системе вещания;
- для провайдеров данных TV-Anytime возможность персонализировать метаданные, которые они предлагают источнику запроса;
- для оборудования клиентов, не обладающих возможностью получать данные вещания, обеспечивается возможность использовать данные системы TV-Anytime, например мобильный телефон или персональный органайзер, чтобы показать клиенту EPG;

- метаданные клиента могут доставляться от PDR к службе метаданных ТВ-Anytime в тех случаях, когда обратный канал доступен и клиент имеет полномочия на доступ к обратному каналу. Представление метаданных клиента позволяет службе метаданных создавать добавленную стоимость служб.

Типы служб метаданных, функциональные возможности служб метаданных, описания возможности и варианты обнаружения служб метаданных определены в ГОСТ Р 56952.

Настоящий стандарт описывает протоколы, которые позволяют описанные транзакции применять в интероперабельной (функционально совместимой) форме.

5 Типы веб-служб метаданных

Форум TV-Anytime определяет несколько типов метаданных веб-службы. Каждый тип веб-службы можно рассматривать как удаленную процедуру с четко определенным набором вводов и выводов и четко определенным поведением. В терминологии WSDL [1] эта удаленная процедура известна как операция (см. приложение А). Операция извлечения метаданных вызывается операцией «get_Data». Операция представления описания клиента вызывается операцией «submit_Data». Операция «Upload_ Personal_Data» определена для представления любой персональной информации и относится к типу данных «ExtendedUserDescriptionType». Операция «Clear _Personal _Data» является операцией удаления информации о клиенте, инициированной операцией «upload_Personal_Data» в двунаправленной сетевой среде.

Типы служб метаданных TV-Anytime, используемых в запросах и ответах, определены в целевом пространстве имен: «urn: tva:transport:2008». Это позволяет схеме XML использовать необходимые инструменты для проверки различных сообщений. Ссылки на типы схем служб метаданных приведены в ГОСТ Р 56476 и выполняются в транспортном пространстве имен схемы XML в соответствии с ГОСТ Р 56476.

Все фрагменты схем XML определены в пространстве имен в пунктах, следующих ниже. Соответствующим определителем имен, используемым в этих фрагментах схемы, является «tns:».

В этих фрагментах схемы XML используются классификаторы пространства имен, представленные в таблице 5.1.

Таблица 5.1 — Классификаторы пространства имен, используемые в фрагментах схемы XML

Классификатор пространства имен	Префикс пространства имен
tva2	urn:tva:metadata:extended:2012
p3p	http://www.w3.org/2002/01/p3pv1

Операция «get_Data» позволяет клиенту получить от сервера данные TV-Anytime о совокупности программ или о группах программ. Ниже приведены примеры типов функциональности, которые поставщик метаданных TV-Anytime может предложить с помощью операции «get_Data»:

- операция, которая принимает список CRID и возвращает ссылочные данные контента для этих CRID;
- операция, которая принимает список CRID и возвращает метаданные TV-Anytime для этих CRID;
- операция, которая принимает запрос программ с определенными атрибутами метаданных (например, с определенным жанром или определенным актером в главной роли и т. д.) и возвращает соответствующие программы;
- операция, которая принимает запрос для программ, транслируемых в определенное время или на определенном канале, и возвращает соответствующие программы.

Все эти типы запросов могут поддерживать операцию «get_data», а также более сложные запросы с участием ряда ограничений метаданных и логических комбинаций этих ограничений.

Примечание — Операция передачи персональной информации может потребовать безопасного и надежного подключения к провайдеру служб, используя такие протоколы, как HTTPS/SSL.

Технология обнаружения служб доставки (получения) метаданных выполнением запроса к серверу DNS представлена в ГОСТ Р 56952.

Технологии обнаружения служб метаданных при отсутствии предварительных данных о службе метаданных средствами UDDI и веб-служб WS-Inspection определены в ГОСТ Р 56952.

6 Транспортные протоколы

6.1 Вводная часть

Для доставки данных XML TV-Anytime поверх IP-сетей используются протоколы SOAP [1] и HTTP. На рисунке 1 представлен стек сетевых протоколов TV-Anytime.

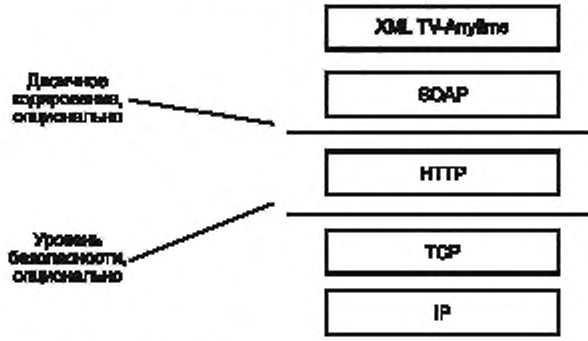

Рисунок 1 — Стек сетевых протоколов TV-Anytime

Архитектура стека протоколов, показанная на рисунке 1, обуславливает обмен сообщениями HTTP, примеры которых представлены на рисунках 2 и 3.

6.2 Протокол SOAP

В настоящем стандарте применение протокола SOAP описывается для следующих случаев:

- предоставление службам TV-Anytime привязки протокола HTTP и в случае необходимости поддержка других транспортных привязок;
- поддержка различных типов сообщений;
- поддержка удаленных вызовов процедуры. В общем случае TV-Anytime не использует обмен сообщениями для удаленного вызова процедуры, так как это означает, что каждый параметр должен быть включен в вызов процедуры, которая не подходит для операций TV-Anytime, имеющих несколько опциональных типов параметров. Однако в соответствии с соглашением об удаленном вызове процедуры выполняется присвоение имени корневому элементу в теле протокола SOAP в соответствии с названием операции.

Примечания

1 Кодирование SOAP не используется в запросе или ответе (элемент в корне тела SOAP будет принадлежать пространству имен типов транспорта TV-Anytime, «urn:tva:transport:2005-03»). Серверы отвергнут любой запрос, который приходит с атрибутом SOAP в сообщении «SOAP Fault».

2 Функция «SOAP Actor» не поддерживается системой TV-Anytime, серверы отклонят любой запрос, который прибывает с атрибутом «SOAP Actor» в заголовке SOAP с соответствующим сообщением «SOAP Fault».

В подразделе 6.2.1 определены конкретные условия сбоев приложений, которые будут использоваться в элементе «SOAP Fault». Использование SOAP установлено при применении интерфейса языка WSDL в приложении А.

```
POST /tva/md-service HTTP/1.0
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
Accept-Encoding: deflate
SOAPAction: "get_Data"

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns=" gost.e-ecolog.ru                                    ">

   <Body>
      <get_Data xmlns="urn:tva:transport:2012"
            xmlns:tvaf=" urn:tva:transport:fieldIDs:2012">
         <QueryConstraints type="OR">
            <Predicate fieldID="tvaf:CRID"
                              fieldValue="crid://example.com/foo"/>
            <Predicate fieldID="tvaf:CRID"
                              fieldValue="crid://example.com/bar"/>
         </QueryConstraints>
         <RequestedTables>
            <Table type="ContentReferencingTable"/>
         </RequestedTables>
      </get_Data>
   </Body>
</Envelope>
```

Рисунок 2 — Пример запроса HTTP

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
Content-Encoding: deflate

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://www.w3.org/2002/06/soap-envelope">

   <Body>
      <get_Data_Result xmlns=" gost.e-ecolog.ru                                    >
         <ContentReferencingTable version="1"
               xmlns="urn:tva:ContentReferencing:2012">
            <!-- ... etc. -->
         </ContentReferencingTable>
      </get_Data_Result>
   </Body>
</Envelope
```

Рисунок 3 — Пример ответа HTTP

6.2.1 Коды ошибок

Первая строка сообщения об отказах регулируется спецификацией SOAP [3]. Отчеты об отказах и коды отказов SOAP возвращаются абонентам, по которым их намерение не может быть определено.

В соответствии с правилами обработки SOAP коды состояния HTTP будут использоваться для передачи информации о статусе в HTTP. Как и в случае с SOAP, для отчета об успехе используется код состояния 200, указывающий, что запрос клиента, включая компонент SOAP, успешно обработан.

Серверы используют элемент «ErrorReport» для передачи сообщений об ошибках уровня приложений, которые являются специфическими для службы метаданных TV-Anytime. В соответствии со спецификацией SOAP, если содержание тела сообщения SOAP не может быть обработано успешно, ошибка SOAP содержит элемент детализации, приведенный в «ErrorReport». Сообщение об ошибке содержит ее описание и код, используемый для определения причины ошибки. Об ошибках, которые возникают на уровнях HTTP или SOAP, сообщается с помощью механизмов ошибок, обеспеченных этими уровнями.

Ошибки уровня приложений TV-Anytime передаются с использованием стандартных кодов состояний HTTP, где код «500-level» (уровень 500) указывает на ошибку, вызванную сервером. В этом случае

служба метаданных выпускает ответ HTTP «500 Internal Server Error» и возвращает «ErrorReport» в отчете об ошибке SOAP.

Любые ошибки, обнаруженные в запросе, делают недействительным весь запрос и приводят к передаче элемента «ErrorReport», который будет создан по поводу ошибки SOAP, как описано ниже. Сервер может сообщить о нескольких ошибках, хотя требования к службе метаданных о продолжении обработки запроса после обнаружения первой ошибки не предъявляются. В соответствии со спецификацией SOAP дополнительные элементы ответа в тело приложения запроса SOAP не включаются.

Вид элемента «ErrorReport» представлен на рисунке 4.

```
<simpleType name="errorCodeType">
  <restriction base="string">
    <enumeration value="FatalError"/>
    <enumeration value="InvalidRequest"/>
    <enumeration value="Unsupported"/>
    <enumeration value="UnrecognizedVersion"/>
    <enumeration value="UnspecifiedError"/>
    <enumeration value="UnsupportedQueryField"/>
    <enumeration value="UnsupportedSortField"/>
    <enumeration value="InvalidFieldID"/>
    <enumeration value="InvalidFieldValue"/>
  </restriction>
</simpleType>
<complexType name="ErrorType">
  <sequence>
    <element name="Reason" type="mpeg7:TextualType" minOccurs="0"
    maxOccurs="unbounded"/>
  </sequence>
  <attribute name="errorCode" use="required" type="tns:errorCodeType"/>
  <attribute name="fields" type="tns:fieldIDListType"/>
 </complexType>
 <complexType name="ErrorReportType">
   <sequence>
  <element name="Error" type="tns:ErrorType" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<element name="ErrorReport" type="tns:ErrorReportType"/>
```

Рисунок 4 — Вид элемента «ErrorReport»

Определения типов в составе элемента «ErrorReport» представлены в таблице 6.1.

Таблица 6.1 — Определение типов в составе элемента «ErrorReport»

Имя	Определение
errorCodeType	Последовательность цифр или коды ошибок, указывающие причину сбоя. Допустимые значения для этой строки перечислены в 6.2.2 и 6.2.3
ErrorType	Сложный тип используется для описания причины сбоя
Reason	Опциональное человеческое описание ошибки
errorCode	Обязательная строка, которая точно определяет характер ошибки
fields	Опциональный атрибут, который содержит список значений «fieldID», касающихся этой ошибки
ErrorReportType	Сложный тип для обозначения ошибок на уровне приложения, которые произошли в результате вызова службы метаданных
Error	Описывает отдельные ошибки на уровне приложений
ErrorReport	Списки ошибок уровня приложений, которые произошли в результате вызова службы метаданных

6.2.2 Общие условия обработки ошибок

Ниже перечисленные коды ошибок инициируются вызовом любой из операций, перечисленных в настоящем стандарте. Атрибут поля не имеет отношения к этим ошибкам и поэтому не присутствует при следующих условиях появления ошибок:

- «FatalError»: означает серьезную техническую ошибку во время обработки запроса;
- «InvalidRequest»: запрос правильно сформирован, но не действует в соответствии со схемой, определенной настоящим документом;
- «Unsupported»: браузер службы метаданных не поддерживает дополнительную функцию, которая необходима для правильной обработки запроса. Возможной причиной этой ошибки является взятие клиентом на себя функций, которые не согласуются с возможностями, указанными в описании функциональных возможностей;
- «UnrecognizedVersion»: пространство имен дочернего элемента внутри элемента тела запроса (например, «urn:tva:transport:2005-03 namespace») не поддерживается с помощью этой службы метаданных;
- «UnspecifiedError»: любая другая ошибка.

Состояние ошибок не является взаимоисключающим, некоторые являются частными случаями других; например, некоторые ошибки «get_Data» по условиям, приведенным ниже, являются особыми случаями ошибок «Unsupported». Служба метаданных должна обеспечить конкретный код для описания фактической ошибки.

В 6.2.3 определяются коды ошибок, которые являются характерными для операции «get_Data». Коды состояний ошибок для операций «submit_Data», «describe_get_Data» и «describe_submit_Data» настоящим стандартом не нормируются.

6.2.3 Условия обработки ошибок операции «get_Data»

При возникновении любой из ошибок, перечисленных ниже, операция «get_data» должна возвратить поле атрибута со списком идентификаторов полей, вызвавших ошибку:

- «UnsupportedQueryField»: запрос содержит значение «fieldID», которое не поддерживается службой метаданных, при возникновении этой ошибки необходимо возвратить атрибут поля со списком неподдерживаемых полей;
- «UnsupportedSortField»: служба метаданных не поддерживает дополнительную функцию, которая необходима, чтобы правильно обработать запрос. Возможная причина этой ошибки состоит в том, что клиент запросил функцию, которая отсутствует в описании возможности;
- «InvalidFieldID»: идентификатор «fieldID» в любом типе предиката или элемента «sortCriteria» содержит идентификатор поля, не определенный ни списком идентификаторов поля TV-Anytime, ни списком идентификаторов поля данной службы описания возможности этой операции;
- «InvalidFieldValue»: элемент «fieldValue» в элементе «BinaryPredicateBagType» содержит строку, которая не соответствует «fieldID» в этом предикате. Это может быть связано с тем, что запрашиваемое поле или относится к перечисляемому типу, или имеет синтаксические ограничения (например, поле «tvaf:CRID»).

При запросе «ContentReferencingTable», когда запрос содержит несколько полей CRID, служба метаданных должна вести себя корректно, если она неспособна обеспечить данные разрешения местоположения для всех CRID. Если CRID синтаксически корректен, но информация о ссылке контента недоступна, то элемент «Result» должен быть возвращен для CRID, имеющих соответствующий флаг статуса.

На рисунке 5 приведен пример ответа, указывающего на ошибку.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
Content-Encoding: deflate

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://www.w3.org/2002/06/soap-envelope">

   <Body>
      <Fault>
```

Рисунок 5, лист 1 — Пример ответа, указывающего на ошибку

```
        <faultcode>Client</faultcode>
        <detail>
            <ErrorReport xmlns="urn:tva:transport:2012"
                xmlns:tvaf="urn:tva:transport:fieldIDs:2012">
                <Error errorCode="UnsupportedQueryField" fields="tvaf:Title">
                  <Reason xml:lang="en">
                            Searching on Title field is unsupported</Reason>
</Error>
</ErrorReport>
</detail>
</Fault>
</Body>
</Envelope>
```

Рисунок 5, лист 2

6.3 Протокол HTTP

Протокол HTTP обеспечивает связку служб метаданных TV-Anytime и может поддерживать другие транспортные связи при следующих условиях:

- сервер HTTP и клиент должны полностью поддерживать HTTP/1.0;
- заголовок запроса к узлу, который направляет клиент, должен соответствовать HTTP/1.1;
- клиент HTTP и сервер договариваются об использовании соответствующего формата сжатия поддержкой заголовка «Accept-Encoding»;
- клиенты должны быть готовы выполнять переадресации HTTP, чтобы позволить серверу работать в условиях отказа системы и выполнять балансирование загрузки. Следует учитывать, что в отличие от HTTP/1.0 в данном случае перенаправления должны выполняться автоматически, без вмешательства клиента, даже при использовании метода POST.

Полю заголовка HTTP «SOAPAction» должно быть присвоено имя вызываемой операции.

6.4 Инкапсуляция метаданных

6.4.1 Данные запроса и ответа

Во всех случаях данные запроса и ответа образуют единый действующий экземпляр документа XML. Документы экземпляра XML свои описания носят в себе и не нуждаются в дальнейшей инкапсуляции. Тем не менее параметры инкапсуляции метаданных для операции «get_data» определяются в 6.4.2 более подробно.

6.4.2 Инкапсуляция ответа операции «get_Data»

Контент в контейнере SOAP является элементом «TVAMain» и/или «ContentReferencingTable». Эти экземпляры документа XML обеспечивают формат инкапсуляции для фрагментов контента. Идентификаторы «fragmentVersion» и «fragmentId» могут появиться на уровне фрагмента.

Каждая операция «get_data» может рассматриваться как единственное описание метаданных. Из этого описания каждый запрос выбирает подмножество метаданных. Таким образом, контент фрагмента «TVAMain» фиксируется в данный момент времени. Атрибут версии фрагмента «TVAMain» применяется только к фрагменту непосредственно, а не к его дочерним фрагментам. Следовательно, несмотря на то что этот фрагмент отправляется с каждым ответом, клиент при увеличении номера версии атрибута элемента «TVAMain» должен только обновить свой кэш фрагмента (если он существует).

6.5 Кодирование метаданных

Сервер должен поддерживать формат текстового кодирования UTF-8 всех запросов и ответов. Он может поддерживать другие кодировки, в этом случае кодировка должна быть указана в заголовке XML. Если кодировка соответствует указанной, то она должна соответствовать и кодировке, указанной в заголовке «Content-Type» HTTP/1.0.

В целях повышения эффективности протокола HTTP клиенты и серверы должны поддерживать по крайней мере один известный каждому из них формат сжатия. Клиенты должны указать на форматы сжатия, которые они могут применять в заголовке HTTP «Accept-Encoding». Сервер HTTP должен использовать самое эффективное кодирование, которое доступно и клиенту, и серверу и указано в ответе HTTP в заголовке «Content-Encoding». Рекомендуется, чтобы клиенты и серверы TV-Anytime HTTP поддерживали скопированный тип «Content-Encoding».

Допускается поддержка других форматов сжатия. Во всех случаях сжатие транспортного слоя должно быть прозрачным для процессора SOAP и на более высоких уровнях.

6.6 Службы безопасности метаданных

Набор спецификаций TV-Anytime на этапе 1 предоставляет клиентам гарантированную целостность доставки метаданных от аутентифицированного сервера использованием протокола TLS при условии поддержки этого протокола и клиентом, и сервером.

Для проверки целостности доставки метаданных сервер метаданных должен быть способен к общению с клиентами, которые реализуют технологию проверки целостности доставки применением протокола TLS. В этом случае сервер определяет источник запроса, используя стандартную аутентификацию HTTP, и выполняет процедуры создания защищенного сеанса связи с клиентом.

Применение любых дополнительных служб безопасности не является обязательным.

Технология блокирования дополнительных служб безопасности настоящим стандартом не нормируется.

Приложение А
(обязательное)

Формальное определение служб метаданных

Это приложение содержит определение интерфейса WSDL [3] для всех служб метаданных TV-Anytime. WSDL определяет ряд терминов (**выделены полужирным шрифтом**), используемых для описания веб-служб. Ниже показана их связь со службами метаданных TV-Anytime:

Operation: Все операции TV-Anytime на основе «запрос — ответ» можно рассматривать как один из видов удаленного вызова процедуры. Примерами операций TV-Anytime являются «submit_Data» или «describe_get_Data»;

PortType: Совокупность операций. Когда определены привязка и конкретная конечная точка, то «PortType» получает имя «port». Должна быть предусмотрена возможность использования всех операций данного порта. При той же привязке не допускается возможность выполнения только части операций порта. Форум TV-Anytime определяет два типа «portType» («get_data» и «submit_Data»), каждый из которых содержит две операции — «get_data» или «submit_Data» и соответствующее описание операции «describe_get_Data» или «describe_submit_Data»;

Binding: привязка (например, протоколов SOAP или GET HTTP) к «portType». Для каждого «portType» может обеспечиваться более одной привязки. Каждый из «portType» представляет собой отдельный порт, предлагающий альтернативные средства для доступа к такому же «portType». Сервер TV-Anytime может предоставлять другие привязки, такие как реализации POST HTTP. В случае привязки к SOAP гарантируется минимальный уровень функциональной совместимости;

Service (служба): семейство связанных портов WSDL. Служба метаданных может содержать порт «get_Data» и/или порт «submit_Data». Практически у большинства серверов имеется только одна служба WSDL. Однако провайдер сервера может объединить в группу определенные порты (например, служба фильма, содержащая порт «get_Data» и порт «submit_Data» и детскую службу, содержащую только порт «get_Data»).

На рисунке А.1 дано определение интерфейса WSDL, которое устанавливает поведение всех веб-служб TV-Anytime. Это определение имеет два назначения:

- интерфейс WSDL формально определяет параметры входов, выходов, правил кодирования и транспортных привязок, используемых всеми веб-службами TV-Anytime. Приведенные определения соответствуют спецификации служб метаданных TV-Anytime, представленных ранее в настоящем стандарте;
- провайдеры служб метаданных, желающие представить реализации определений WSDL своим службам метаданных, могут импортировать это определение интерфейса WSDL.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="tva_webservice" targetNamespace="http://tva.webservice.namespace"
xmlns="gost.e-ecolog.ru
xmlns:soap="gost.e-ecolog.ru                                    "
xmlns:http="gost.e-ecolog.ru
xmlns:xs="gost.e-ecolog.ru
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="gost.e-ecolog.ru                              " xmlns:transport="urn:tva:transport:2008">
    <xs:documentation> WSDL Service Interface for a TV Anytime web service API.
        This WSDL document defines the API calls for the four types of TV Anytime ports
    </xs:documentation>
    <types>
        <xs:schema>
            <xs:import namespace="urn:tva:transport:2008" schemaLoction="tva_transport_6-
1_v151.xsd"/>
        </xs:schema>
    </types>
    <!-- Basic input and output messages. -->
    <message name="get_Data">
        <part name="body" element="transport:get_Data"/>
    </message>
    <message name="get_Data_Result">
        <part name="body" element="transport:get_Data_Result"/>
    </message>
    <message name="describe_get_Data">
```

Рисунок А.1, лист 1 — Листинг определения интерфейсов WSDL

```
<part name="body" element="transport:describe_get_Data"/>
    </message>
    <message name="describe_get_Data_Result">
<part name="body" element="transport:describe_get_Data_Result"/>
    </message>
    <message name="submit_Data">
<part name="body" element="transport:submit_Data"/>
</message>
<message name="submit_Data_Result">
<part name="body" element="transport:submit_Data_Result"/>
</message>
<message name="describe_submit_Data">
<part name="body" element="transport:describe_submit_Data"/>
</message>
<message name="describe_submit_Data_Result">
<part name="body" element="transport:describe_submit_Data_Result"/>
</message>
<message name="upload_Personal_Data">
<part name="body" element="transport:upload_Personal_Data"/>
</message>
<message name="upload_Personal_Data_Result">
<part name="body" element="transport:upload_Personal_Data_Result"/>
</message>
<message name="describe_upload_Personal_Data">
<part name="body" element="transport:describe_upload_Personal_Data"/>
</message>
<message name="describe_upload_Personal_Data_Result">
<part name="body" element="transport:describe_upload_Personal_Data_Result"/>
</message>
<message name="clear_Personal_Data">
<part name="body" element="transport:clear_Personal_Data"/>
</message>
<message name="clear_Personal_Data_Result">
<part name="body" element="transport:clear_Personal_Data_Result"/>
</message>
<message name="ErrorReportMessage">
<part name="body" element="transport:ErrorReport"/>
</message>
<!-- The different types of services (ports) with their inputs and outputs. -->
<portType name="get_Data_Port">
<operation name="get_Data">
<input message="tns:get_Data"/>
<output message="tns:get_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
<operation name="describe_get_Data">
<input message="tns:describe_get_Data"/>
<output message="tns:describe_get_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
</portType>
<portType name="submit_Data_Port">
<operation name="submit_Data">
<input message="tns:submit_Data"/>
<output message="tns:submit_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
<operation name="describe_submit_Data">
<input message="tns:describe_submit_Data"/>
<output message="tns:describe_submit_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
</portType>
<portType name="upload_Personal_Data_Port">
```

Рисунок А.1, лист 2

```
<operation name="upload_Personal_Data">
<input message="tns:upload_Personal_Data"/>
<output message="tns:upload_Personal_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
<operation name="describe_upload_Personal_Data">
<input message="tns:describe_upload_Personal_Data"/>
<output message="tns:describe_upload_Personal_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
</portType>
<portType name="clear_Personal_Data_Port">
<operation name="clear_Personal_Data">
<input message="tns:clear_Personal_Data"/>
<output message="tns:clear_Personal_Data_Result"/>
<fault name="error" message="tns:ErrorReportMessage"/>
</operation>
</portType>
<!-- The bindings: defines how SOAP/HTTP is used to carry the service. -->
<binding name="get_Data_SOAP" type="tns:get_Data_Port">
<documentation>TV Anytime get_Data binding</documentation>
<soap:binding style="document" transport="gost.e-ecolog.ru                    >
<operation name="get_Data">
<soap:operation soapAction="get_Data"/>
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
<operation name="describe_get_Data">
<soap:operation soapAction="describe_get_Data"/>
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
</binding>
<binding name="submit_Data_SOAP" type="tns:submit_Data_Port">
<documentation>TV Anytime submit_Data binding</documentation>
<soap:binding style="document" transport="gost.e-ecolog.ru                    >
<operation name="submit_Data">
<soap:operation soapAction="submit_Data"/>
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
<operation name="describe_submit_Data">
<soap:operation soapAction="describe_submit_Data"/>
```

Рисунок А.1, лист 3

```
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
</binding>
<binding name="upload_Personal_Data_SOAP" type="tns:upload_Personal_Data_Port">
<documentation>TV Anytime upload_Personal_Data binding</documentation>
<soap:binding style="document" transport="gost.e-ecolog.ru >
<operation name="upload_Personal_Data">
<soap:operation soapAction="upload_Personal_Data"/>
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
<operation name="describe_upload_Personal_Data">
<soap:operation soapAction="describe_upload_Personal_Data"/>
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
</binding>
<binding name="clear_Personal_Data_SOAP" type="tns:clear_Personal_Data_Port">
<documentation>TV Anytime clear_Personal_Data binding</documentation>
<soap:binding style="document" transport="https://schemas.xmlsoap.org/soap/https"/>
<operation name="clear_Personal_Data">
<soap:operation soapAction="clear_Personal_Data"/>
<input>
<soap:body use="literal" parts="body"/>
</input>
<output>
<soap:body use="literal" parts="body"/>
</output>
<fault name="error">
<soap:fault name="error" use="literal"/>
</fault>
</operation>
</binding>
<service name="tva_get_web_service">
<port name="getDataPort" binding="tns:get_Data_SOAP">
<soap:address location="http://example.com/companyinfo"/>
</port>
</service>
<service name="tva_submit_web_service">
<port name="submitDataPort" binding="tns:submit_Data_SOAP">
<soap:address location="gost.e-ecolog.ru >
</port>
```

Рисунок А.1, лист 4

```
</service>
<service name="tva_uploadPData_web_service">
<port name="uploadPDataPort" binding="tns:upload  Personal_Data_SOAP">
<soap:address location="gost.e-ecolog.ru                    >
</port>
</service>
<service name="tva_clearPData_web_service">
<port name="clearPDataPort" binding="tns:clear_Personal_Data_SOAP">
<soap:address location="gost.e-ecolog.ru                    >
</port>
</service>
—          </definitions>
```

Рисунок А.1, лист 5

Библиография

[1] Хабибуллин И.Ш. Разработка веб-служб средствами Java. — СПб.: БХВ-Петербург, 2003. — 400 с.: ил.

УДК 621.397:681.327.8:006.354 ОКС 33.170

Ключевые слова: вещание цифровое, TV-Anytime, служба метаданных, провайдер, профиль клиента, схема метаданных, схема XML, двунаправленная сеть, HTTP, SOAP

Редактор переиздания *Н.Е. Рагузина*
Технический редактор *И.Е. Черепкова*
Корректор *Л.С. Лысенко*
Компьютерная верстка *И.А. Налейкиной*

Сдано в набор 02.03.2020. Подписано в печать 21.05.2020. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 2,79. Уч.-изд. л. 2,51.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении во ФГУП «СТАНДАРТИНФОРМ» для комплектования Федерального информационного фонда стандартов, 117418 Москва, Нахимовский пр-т, д. 31, к. 2.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru