ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 40220— 2015

Информационные технологии

W3C SOAP. ВЕРСИЯ 1.2

Часть 2
Дополнения
(вторая редакция)

ISO/IEC 40220:2011
Information technology W3C SOAP Version 1.2 Part 2: Adjuncts
(Second Edition)

(IDT)

Издание официальное

Москва
Стандартинформ
2015

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО ИАВЦ) на основе собственного аутентичного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 мая 2015 г. № 462-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 40220:2011 «Информационные технологии. W3C SOAP. Версия 1.2. Часть 2. Дополнения (вторая редакция)» [ISO/IEC 40220:2011 «Information technology — W3C SOAP Version 1.2 Part 2: Adjuncts (Second Edition)»]

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты Российской Федерации, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, 2015

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 Условные обозначения ... 2
4 Модель данных SOAP ... 3
4.1 Ребра графа ... 3
4.2 Узлы графа ... 3
4.3 Значения ... 3
5 Кодирование SOAP ... 3
5.1 Отображение между XML и моделью данных SOAP ... 4
5.2 Декодирование отказов ... 6
6 Представление SOAP RPC ... 7
6.1 Использование RPC в сети Интернет ... 7
6.2 RPC и элемент Body SOAP ... 8
6.3 RPC и заголовок SOAP ... 9
6.4 Отказы RPC ... 9
7 Соглашение для описания функций и привязок ... 10
7.1 Модель и свойства ... 10
8 Шаблоны обмена сообщениями и функции SOAP ... 11
8.1 Соглашения о свойствах для шаблонов обмена сообщениями SOAP ... 11
8.2 Шаблон обмена сообщениями SOAP «запрос-ответ» ... 12
8.3 Шаблон обмена сообщениями SOAP «ответ SOAP» ... 16
8.4 Функция SOAP «Веб-метод» ... 20
8.5 Функция SOAP «Действие» ... 20
9 Привязка SOAP к HTTP ... 21
9.1 Введение ... 21
9.2 Имя привязки ... 22
9.3 Поддерживаемые шаблоны обмена сообщениями ... 22
9.4 Поддерживаемые функции ... 22
9.5 Операции шаблонов обмена сообщениями ... 23
9.6 Соображения безопасности ... 27
Приложение А (справочное) Тип медиа «application/soap+xml» ... 28
Приложение В (справочное) Отображение имен, определенных приложениями, в имена XML ... 28
Приложение С (справочное) Использование W3C XML Schema с кодировкой SOAP ... 30
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации ... 31
Библиография ... 31

Введение

SOAP версия 1.2 — это упрощенный протокол, предназначенный для обмена структурированной информацией в децентрализованной, распределенной среде. «W3C SOAP версия 1.2. Часть 2. Дополнения» (данный документ) определяет ряд дополнений, которые МОГУТ использоваться в инфраструктуре обмена сообщениями SOAP.

Данный документ является Рекомендацией W3C, разработанной рабочей группой Протокола XML, которая является частью Действия Веб-служб. Эта вторая обновленная редакция заменяет исходную версию Рекомендации, и в нее включены исправления всех обнаруженных опечаток. Различия между этими версиями описаны в отдельном документе[1)]. Помимо этого, настоящий документ включает изменения к Шаблону обмена сообщениями «Запрос-ответ» SOAP, разрешающие не использовать конверт SOAP в ответе, для поддержки однонаправленного взаимодействия.

Данный документ рецензировался членами консорциума W3C, разработчиками программного обеспечения, другими группами W3C и заинтересованными сторонами и утвержден директором в качестве Рекомендации W3C. Это устоявшийся документ, на который можно ссылаться и цитировать в других документах в качестве нормативного. Участие W3C в создании Рекомендации должно привлечь внимание к спецификации и способствовать ее широкому применению. Это улучшает функциональность и функциональную совместимость Всемирной паутины.

Отчет о реализациях SOAP 1.2 может быть найден в отдельном документе[2)].

Дополнительные сведения о реализации шаблона «Запрос с необязательным ответом» можно найти в отчете о тестировании реализаций WSDL 2.0[3)].

Данный документ соответствует текущей патентной политике W3C CPP[4)] и переходной патентной политике W3C[5)]. W3C поддерживает общедоступный список патентов[6)], имеющих отношение к спецификациям, подготовленным рабочей группой. Этот документ, содержащий перечень патентов, также включает инструкции по раскрытию патентов. Если кто-либо обладает действительным знанием патента, который удовлетворяет существенным требованиям, то он должен раскрыть эту информацию в соответствии с разделом 6 патентной политики W3C[7)].

Список текущих Рекомендаций W3C и других технических отчетов можно найти по адресу gost.e-ecolog.ru

1) gost.e-ecolog.ru
2) gost.e-ecolog.ru
3) gost.e-ecolog.ru
4) gost.e-ecolog.ru
5) gost.e-ecolog.ru
6) gost.e-ecolog.ru
7) gost.e-ecolog.ru

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

W3C SOAP. Версия 1.2 Часть 2 Дополнения (вторая редакция)

Information technology. W3C SOAP. Version 1.2.
Part 2. Adjuncts (second edition)

Дата введения — 2016—06—01

1 Область применения

SOAP версии 1.2 (SOAP) является упрощенным протоколом, предназначенным для обмена структурированной информацией в децентрализованной, распределенной среде. Спецификация SOAP состоит из трех частей. Часть 2 (настоящий стандарт) определяет ряд дополнений, которые МОГУТ использоваться в инфраструктуре обмена сообщениями SOAP:

1 Модель данных SOAP представляет определенные приложением структуры данных и значения в виде ориентированного помеченного графа (см. раздел 4).

2 Кодирование SOAP определяет набор правил для кодирования экземпляров данных, соответствующих модели данных SOAP для включения в SOAP-сообщения (см. раздел 5).

3 Представление SOAP RPC определяет соглашение о том, как использовать модель данных SOAP для представления вызовов и ответов RPC (см. раздел 6).

4 Соглашение для описания функций и привязок описывает функции и привязки в терминах свойств и значений свойств (см. раздел 7).

5 Шаблоны обмена сообщениями и функции SOAP определяют шаблон обмена сообщениями «запрос-ответ» и шаблон обмена сообщениями, поддерживающий запросы, не соответствующие протоколу SOAP, для получения ответов по протоколу SOAP (см. раздел 8).

6 Функция SOAP «Веб-метод» определяет функцию для управления методами, используемыми в сети Интернет (см. 8.4).

7 Привязка SOAP к HTTP определяет привязку SOAP к HTTP (см. [RFC 2616]) в соответствии с правилами инфраструктуры привязки SOAP протокола (см. [ИСО/МЭК 40210, раздел 7]).

SOAP 1.2. Часть 0 [SOAP часть 0] не является нормативным документом, предназначенным для использования в качестве простого учебного пособия по функциям спецификаций SOAP версии 1.2.

SOAP 1.2. Часть 1 [ИСО/МЭК 40210] определяет инфраструктуру обмена сообщениями SOAP.

Примечание — В предыдущих версиях данной спецификации название SOAP считалось аббревиатурой. В этой версии это не так.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие международные стандарты и документы. Для датированных документов используют только указанное издание. Для недатированных документов используют самое последнее издание ссылочного документа (с учетом всех его изменений).

ИСО/МЭК 40210:2014 Информационные технологии. W3C SOAP — Версия 1.2. Часть 1. Основы обмена сообщениями (вторая редакция) (ISO/IEC 40210:2014 Information technology. W3C SOAP Version 1.2 Part 1: Messaging Framework (Second Edition))

Издание официальное

Гипертекстовый Протокол передачи — HTTP/1.1 (RFC 2616 Hypertext Transfer Protocol — HTTP/1.1) [RFC 2616]

Ключевые слова, используемые в RFCs, чтобы указать на уровни требования (RFC 2119 Key words for use in RFCs to Indicate Requirement Levels) [RFC 2119]

XML-схема. Часть 1. Структуры. Вторая редакция (XML Schema Part 1: Structures Second Edition) [XML Schema Part 1]

XML-схема. Часть 2. Типы данных. Вторая редакция (XML Schema Part 2: Datatypes Second Edition) [XML Schema Part 2]

Универсальный идентификатор ресурса (URI). Основы синтаксиса (Uniform Resource Identifiers (URI): Generic Syntax) [RFC 3986]

Пространства имен в XML (вторая редакция) (Namespaces in XML (Second Edition)) [Namespaces in XML]

Расширяемый язык разметки (XML) 1.0 (четвертая редакция) (Extensible Markup Language (XML) 1.0 (Fourth Edition)) [XML 1.0]

Информационный набор XML (вторая редакция) (XML Information Set (Second Edition)) [XML InfoSet]

Типы документов XML (XML Media Types) [RFC 3023]

Тип документа «application/soap+xml» (The «application/soap+xml» media type) [RFC 3902]

3 Условные обозначения

Ключевые слова «ДОЛЖЕН» (MUST), «НЕ ДОЛЖЕН» (MUST NOT), «ТРЕБУЕМЫЙ» (REQUIRED), «БУДЕТ» (SHALL), «НЕ БУДЕТ» (SHALL NOT), «СЛЕДУЕТ» (SHOULD), «НЕ СЛЕДУЕТ» (SHOULD NOT), «РЕКОМЕНДУЕМЫЙ» (RECOMMENDED), «МОЖЕТ» (MAY) и «ДОПОЛНИТЕЛЬНЫЙ» (OPTIONAL) в данном документе должны интерпретироваться в соответствии с RFC 2119 (см. [RFC 2119]).

Префиксы пространств имен, использующиеся в данной спецификации, перечислены в таблице 1. Выбор любого префикса пространства имен произволен и не является семантически существенным (см. [XML Info-set]).

Таблица 1 — Префиксы и пространства имен, используемые в данной спецификации

Префикс	Пространство имен	Примечания
enc	«gost.e-ecolog.ru	Нормативная XML-схема [XML Schema Part 1], документ [XML Schema Part 2] для пространства имен «gost.e-ecolog.ru может быть найден в gost.e-ecolog.ru
env	«gost.e-ecolog.ru	Определен в SOAP 1.2. Часть 1 [ИСО/МЭК 40210]
rpc	«gost.e-ecolog.ru	Нормативная XML-схема [XML Schema Part 1], документ [XML Schema Part 2] для пространства имен «gost.e-ecolog.ru может быть найден в http://www.w3.org/2003/05/soap-rpc
xs	«gost.e-ecolog.ru	Определен в спецификации XML-схемы W3C [XML Schema Part 1], [XML Schema Part 2]
xsi	«http://www.w3.org/2001/XMLSchema-instance»	Определен в спецификации XML-схемы W3C [XML Schema Part 1], [XML Schema Part 2]

Имена пространства имен вида «gost.e-ecolog.ru и «gost.e-ecolog.ru представляют определенные приложением или контекстно-зависимые универсальные идентификаторы ресурсов (URI) (см. [RFC 3986]).

Данная спецификация использует расширенную форму Бэкуса — Наура (РБНФ), определенную в [XML 1.0].

Все части настоящей спецификации являются нормативными, за исключением примеров и разделов, отмеченных как «справочное».

4 Модель данных SOAP

Модель данных SOAP представляет определенные приложением структуры данных и значения в виде ориентированного помеченного графа. Компоненты этого графа описаны в следующих подразделах.

Модель данных SOAP предназначена обеспечить отображение данных, не основанных на языке XML, в представлении для передачи по каналам данных. Использование модели данных SOAP, соответствующего кодирования SOAP (см. раздел 5) и/или представления SOAP RPC (см. раздел 6) НЕ ОБЯЗАТЕЛЬНО. Приложения, данные которых уже представлены в XML, могут не использовать модель данных SOAP. Так как модель данных SOAP не является обязательной, данная спецификация не требует, чтобы реализация узла SOAP поддерживала модель данных SOAP, кодирование SOAP и/или представление SOAP RPC.

4.1 Ребра графа

Ребро графа *исходит* из узла графа и *заканчивается* в узле графа. Ребро, исходящее из узла графа, называется *исходящим ребром* данного узла графа. Ребро, заканчивающееся в узле графа, называется *входящим ребром* данного узла графа. Ребро МОЖЕТ исходить и заканчиваться в одном и том же узле графа. У ребра МОЖЕТ быть только начальный узел графа, такое ребро является только исходящим. У ребра МОЖЕТ быть только завершающий узел графа, такое ребро является только входящим.

Исходящие ребра узла графа МОГУТ различаться меткой или позицией. Позиция задает полный порядок таких ребер. Таким образом, если какие-либо исходящие ребра данного узла различаются позицией, то все исходящие ребра этого узла различаются позицией.

4.1.1 Метки ребер

Метка ребра — квалифицированное имя XML. Две метки ребра равны тогда и только тогда, когда полные имена XML равны, то есть когда выполняются оба следующих утверждения:

1 Значения локальных имен совпадают.

2 Выполнено одно или оба из следующих утверждений:

1 У обоих имен отсутствуют значения пространства имен.

2 У обоих имен присутствуют значения пространства имен, и эти значения совпадают.

В 4.3 описано, как метки ребер и позиции ребер используются для различения элементов закодированных значений. Дополнительная информация о сравнении квалифицированных имен XML представлена в стандарте «Схема XML» [XML Schema Part 2].

4.2 Узлы графа

Количество исходящих ребер узла графа может быть больше или равно нулю. У узла графа, не имеющего исходящих ребер, может быть необязательное дополнительное лексическое значение. У всех узлов графа имеется необязательное дополнительное имя типа, представленное значением типа *xs:QName* из пространства имен «http://www.w3.org/2001/XML Schema» (см. [XML Schema Part 2]).

4.2.1 Одно- и многоссылочные узлы

Узел графа может быть *односсылочным* и *многоссылочным*. У односсылочного узла имеется единственное входящее ребро. У многоссылочного узла число входящих ребер более одного.

4.3 Значения

Простое значение представляется как узел графа с лексическим значением.

Составное значение представляется как узел графа с нулевым или бóльшим количеством исходящих ребер следующим образом:

1 Узел графа, исходящие ребра которого отличаются исключительно метками, называется «структура». Исходящие ребра структуры ДОЛЖНЫ быть маркированы различающимися именами (см. 4.1.1).

2 Узел графа, исходящие ребра которого отличаются исключительно позицией, называется «массив». Исходящие ребра массива НЕ ДОЛЖНЫ быть помечены.

5 Кодирование SOAP

Кодирование SOAP обеспечивает средство кодирования экземпляров данных, которые соответствуют модели данных, описанной в разделе 4. Это кодирование МОЖЕТ использоваться для пере-

дачи данных в блоках заголовка SOAP и/или теле SOAP. Другие модели данных, альтернативные кодировки модели данных SOAP, незакодированные данные МОГУТ также использоваться в сообщениях SOAP (спецификация альтернативных стилей кодирования описана в «Атрибут SOAP encodingStyle» [ИСО/МЭК 40210, пункт 8.1.1]; ограничения модели данных и кодирования в удаленных вызовах процедур SOAP (SOAP RPC) описаны в разделе 6).

Правила преобразования в последовательную форму, определенные в этом разделе, описаны в документе, идентифицированном URI «gost.e-ecolog.ru . Сообщение SOAP, использующее данное конкретное преобразование в последовательную форму, СЛЕДУЕТ обозначать при помощи *информационного объекта-атрибута* SOAP encodingStyle [ИСО/МЭК 40210, пункт 8.1.1].

5.1 Отображение между XML и моделью данных SOAP

XML допускает очень гибкое кодирование данных. Кодирование SOAP определяет более узкий набор правил для кодирования графов, описанных в разделе 4. Данный раздел определяет кодирование на высоком уровне, а последующие разделы описывают правила кодирования более подробно. Кодировки, описанные в этом разделе, могут использоваться в сочетании с отображением запросов и ответов RPC, определенных в разделе 6.

Кодировки, описанные ниже, представлены с точки зрения десериализатора. В каждом случае предполагается наличие сериализованного XML и описывается его отображение на соответствующий граф.

Обычно для заданного графа возможны несколько вариантов кодирования. При сериализации графа для передачи в сообщении SOAP ДОЛЖНО использоваться представление, десериализация которого порождает тождественный граф; если возможны несколько разных представлений, МОЖЕТ использоваться любое из них. При получении закодированного сообщения SOAP ДОЛЖНЫ приниматься все представления.

5.1.1 Кодирование ребер и узлов графа

Каждое ребро графа кодируется как *информационный объект-элемент*, и каждый *информационный объект-элемент* представляет ребро графа. В 5.1.3 описывается отношение между метками ребер и свойствами [local name] и [namespace name] таких *информационных объектов-элементов*.

Узел графа, в котором завершается ребро, определяется при анализе сериализированного XML следующим образом:

1 Если *информационный объект-элемент*, представляющий ребро, не имеет среди своих атрибутов *информационного объекта-атрибута ref* (см. 5.1.5.2), тогда говорят, что *информационный объект-элемент представляет* узел графа, и ребро заканчивается в этом узле. В таких случаях *информационный объект-элемент* представляет и ребро графа, и узел графа.

2 Если *информационный объект-элемент*, представляющий ребро, имеет среди своих атрибутов *информационный объект-атрибут* ref (см. 5.1.5.2), то значение этого *информационного объекта-атрибута* ДОЛЖНО быть идентично значению ровно одного *информационного объекта-атрибута* id (см. 5.1.5.1) в том же конверте. В этом случае ребро заканчивается в узле графа, представленном *информационным объектом-элементом*, к которому относится *информационный объект-атрибут* id. Этот *информационный объект-элемент* ДОЛЖЕН находиться в области действия атрибута encodingStyle со значением «gost.e-ecolog.ru (см. [ИСО/МЭК 40210, пункт 8.1.1]).

Все узлы графа кодируются, как описано выше в случае 1. Дополнительные входящие ребра для многоссылочных узлов графа кодируются, как описано выше в случае 2.

5.1.2 Кодирование простых значений

Лексическое значение узла графа, представляющего простое значение, описывается последовательностью символов Unicode, идентифицированных дочерними элементами *информационного объекта-символа информационного объекта-элемента*, представляющего данный узел. *Информационный объект-элемент*, представляющий узел с простым значением, МОЖЕТ содержать *информационный объект-атрибут* nodeType (см. 5.1.7). Необходимо отметить, что определенные символы Unicode не могут быть представлены в XML [XML 1.0].

5.1.3 Кодирование составных значений

Исходящее ребро узла графа кодируется как дочерний *информационный объект-элемент информационного объекта-элемента*, представляющего узел (см. 5.1.1). Различные правила применяются в зависимости от того, какой вид составного значения представляет узел графа. Эти правила состоят в следующем:

1 Для ребра графа, которое характеризуется меткой, свойства [local name] и [namespace name] дочернего *информационного объекта-элемента* вместе определяют значение метки ребра.

2 Для ребра графа, которое характеризуется позицией:

- порядковая позиция ребра графа соответствует позиции дочернего *информационного объекта-элемента* относительно его одноуровневых элементов;
- свойства [local name] и [namespace name] дочернего *информационного элемента* не являются значимыми.

3 Информационный объект-элемент, представляющий узел с составным значением, среди своих атрибутов МОЖЕТ иметь *информационный объект-атрибут* nodeType (см. 5.1.7).

4 Следующие правила применяются к кодированию узла графа, который представляет «массив»:

- *информационный объект-элемент*, представляющий узел массива, среди своих атрибутов МОЖЕТ иметь *информационный объект-атрибут* itemType (см. 5.1.4.1);
- *информационный объект-элемент*, представляющий узел массива, среди своих атрибутов МОЖЕТ иметь *информационный объект-атрибут* arraySize (см. 5.1.6).

5 Если ребро графа не завершается в узле графа, тогда оно может быть опущено при сериализации или закодировано как *информационный объект-элемент* с *информационным объектом-атрибутом* xsi:nil со значением true.

5.1.4 Вычисление свойства «имя типа»

Свойство «имя типа» узла графа — это пара {namespace name, local name}, вычисляемая следующим образом:

1 Если *у информационного объекта-элемента*, представляющего узел графа, среди своих атрибутов есть *информационный объект-атрибут* xsi:type, тогда свойство «имя типа» узла графа — это значение *информационного объекта-атрибута* xsi:type.

Примечание — Данный атрибут имеет тип *xs:QName* [Схема XML ч.2]; его значение состоит из пары {namespace name, local name}. Ни префикс, используемый для создания QName, ни любая информация, касающаяся любых определений типа, не являются частью значения. Граф SOAP содержит только полностью квалифицированное имя типа.

2 В случае, если родительский *информационный объект-элемент информационного объекта-элемента*, представляющий узел графа, среди своих атрибутов имеет *информационный объект-атрибут* enc:itemType (см. 5.1.4.1), тогда свойство «имя типа» узла графа — это значение *информационного объекта-атрибута* enc:itemType.

3 В противном случае значение свойства «имя типа» узла графа не определено.

Примечания

1 Эти правила определяют, как свойство «имя типа» узла графа вычисляется из сериализированного кодирования. Данная спецификация не требует проверки корректности с использованием специфического языка для описания схем или системы типов. Также она не включает встроенные типы и не приводит стандартизованных ошибок для описания конфликтов значений и имен типов.

2 Тем не менее, разработка дополнительных спецификаций для описания использования кодирования SOAP, основанного на специфических языках для описания схем или системы типов, не запрещена. Такие дополнительные спецификации МОГУТ накладывать требования на проверку корректности с использованием определенного языка описания схемы и МОГУТ определять, какие должны быть сгенерированы ошибки в случае неуспешно завершившейся проверки корректности. Такие дополнительные спецификации МОГУТ определять дополнения к десериализованному графу на основе информации, полученной после проверки корректности. Использование кодировщиком SOAP атрибута xsi:type предназначено для упрощения интеграции с языком W3C XMLSchema (см. приложение С). Другие языки для описания схем, схемы данных и программных систем типов, основанные на XML, МОГУТ использоваться, но только при условии, что они совместимы с сериализацией, описанной в данной спецификации.

5.1.4.1 Информационный объект-атрибут itemType

Информационный объект-атрибут itemType имеет следующие свойства инфо-набора:

[local name] itemType;

[namespace name] «http://www.w3.org/2003/05/soap-encoding»;

[specified] со значением true.

Тип *информационного объекта-атрибута* itemType — *xs:QName*. Значение *информационного объекта-атрибута* itemType используется для вычисления свойства «имя типа» (см. 5.1.4) элемента массива.

5.1.5 Уникальные идентификаторы

5.1.5.1 Информационный объект-атрибут id

Информационный объект-атрибут id имеет следующие свойства инфо-набора:

- [local name] id;
- [namespace name] «gost.e-ecolog.ru ;
- [specified] со значением true.

Тип *информационного объекта-атрибута* id — *xs:ID*. Значение *информационного объекта-атрибута* id — уникальный идентификатор, на который можно ссылаться из *информационного объекта-атрибута* ref (см. 5.1.5.2).

5.1.5.2 Информационный объект-атрибут ref

Информационный объект-атрибут ref имеет следующие свойства инфо-набора:

- [local name] ref;
- [namespace name] «gost.e-ecolog.ru ;
- [specified] со значением true.

Тип *информационного объекта-атрибута* ref — *xs:IDREF*. Значение *информационного объекта-атрибута* ref является ссылкой на уникальный идентификатор, определенный *информационным объектом-атрибутом* id (см. 5.1.5.1).

5.1.5.3 Ограничения на информационные объекты-атрибуты id и ref

Значение *информационного объекта-атрибута* ref ДОЛЖНО быть идентично значению ровно одного *информационного объекта-атрибута* id.

Информационный объект-атрибут ref и *информационный объект-атрибут* id не ДОЛЖНЫ появляться в одном и том же *информационном элементе*.

5.1.6 *Информационный объект-атрибут arraySize*

Информационный объект-атрибут arraySize имеет следующие свойства инфо-набора:

- [local name] arraySize;
- [namespace name] «gost.e-ecolog.ru .

Тип *информационного объекта-атрибута* arraySize — *enc:arraySize*. Значение *информационного объекта-атрибута* arraySize ДОЛЖНО соответствовать следующей грамматике РФБН:

[1]	arraySizeValue	::=	(«*» | concreteSize) nextConcreteSize*
[2]	nextConcreteSize	::=	whitespace concreteSize
[3]	concreteSize	::=	[0-9] +
[4]	whitespace	::=	(#x20 | #x9 | #xD | #xA) +

Атрибут arraySize передает предложенное отображение массива SOAP в многомерную структуру данных программы. Количество элементов списка arraySize представляет число размерностей, а значения — информацию о значениях соответствующих размерностей. Для кодирования SOAP многомерных массивов, узлы выбираются так, чтобы последний подстрочный индекс (т.е. подстрочный индекс, соответствующий последней указанной размерности) менялся чаще всего, а первый — реже всего. Звездочка МОЖЕТ использоваться только на месте первого измерения, чтобы показать, что число элементов в этой размерности не установлено; звездочки не ДОЛЖНЫ появляться в других позициях в списке. Значение по умолчанию *информационного объекта-атрибута* arraySize — это звездочка «*», т.е. единственное измерение с неуказанным числом элементов.

5.1.7 *Информационный объект-атрибут nodeType*

Информационный объект-атрибут nodeType имеет следующие свойства инфо-набора:

- [local name] nodeType;
- [namespace name] «gost.e-ecolog.ru ;
- [specified] со значением true.

Тип *информационного объекта-атрибута* nodeType — *enc:nodeType*.

Если имеется *информационный объект-атрибут* nodeType, то его значение ДОЛЖНО быть одной из строк: «simple», «struct» или «array». Значение атрибута указывает на тип значения, которое представляет узел: простое значение, значение составной структуры или значение составного массива соответственно.

5.2 Декодирование отказов

Во время десериализации получатель SOAP:

- ДОЛЖЕН генерировать отказ SOAP «env:Sender» с дополнительным кодом (subcode) enc:MissingID, если сообщение содержит *информационный объект-атрибут* ref, но не содержит никакого соответствующего *информационного объекта-атрибута* id (см. 5.1.5.3);

- ДОЛЖЕН генерировать отказ SOAP «env:Sender» с дополнительным кодом enc:DuplicateID, если сообщение содержит два или более *информационных объекта-атрибута* id с одним и тем же значением (см. 5.1.5.3);

- МОЖЕТ генерировать отказ SOAP «env:Sender» с дополнительным кодом enc:UntypedValue, если свойство «имя типа» закодированного узла графа не указано.

6 Представление SOAP RPC

Одна из целей протокола SOAP состоит в том, чтобы упростить обмен сообщениями, которые легко отображаются на определения и вызовы методов и процедур в обычных языках программирования. Для этого данный раздел определяет универсальное представление запросов и ответов вызова удаленной процедуры (RPC). Данный раздел не определяет фактические привязки ни к каким определенным языкам программирования. Данное универсальное представление полностью независимо от платформы, и, благодаря значительным усилиям, SOAP стал совместимым с сетью Интернет.

Как упомянуто в разделе 4, использование и реализация представления SOAP RPC НЕ ОБЯЗАТЕЛЬНЫ.

Информационный объект-атрибут SOAP encodingStyle [ИСО/МЭК 40210, пункт 8.1.1] используется для указания стиля кодирования представления RPC. Кодирование, определенное таким образом, ДОЛЖНО поддерживать требования, изложенные в разделе 4. Стиль кодирования, определенный в разделе 5, поддерживает такие конструкции и поэтому подходит для использования в представлении SOAP RPC.

Представление SOAP RPC не основывается ни на каких привязках протокола SOAP. Когда используется привязка SOAP к HTTP, вызов RPC легко отображается на запрос HTTP, и ответ RPC отображается на ответ HTTP (см. раздел 9). Однако, представление SOAP RPC не ограничено привязкой SOAP к HTTP.

Чтобы вызвать RPC, необходима следующая информация:

- адрес целевого узла SOAP;
- имя процедуры или имя метода;
- идентификационные данные и значения любых параметров, которые должны быть переданы процедуре или методу. Параметрам, используемым для идентификации Веб-ресурсов, СЛЕДУЕТ отличаться от параметров, представляющих данные или управляющую информацию (см. 6.1.1);
- значения для свойств, требуемых используемой привязкой. Например, «GET» или «POST» для свойства «http://gost.e-ecolog.ru (см. 8.4);
- дополнительные данные заголовка.

SOAP RPC использует привязку протокола для обеспечения механизма переноса URI целевого узла SOAP. Для HTTP протокола URI запроса указывает на ресурс, к которому обращен вызов. Кроме требования на правильность URI, SOAP не устанавливает ограничений на форму идентификатора (см. RFC 3986 [RFC 3986] для получения дополнительной информации по URI). В 6.1.1 далее обсуждается использование URI для идентификации ресурсов RPC.

Представление SOAP RPC использует шаблоны обмена сообщениями «запрос-ответ» и «ответ SOAP» (см. 8.2 и 8.3). Представление SOAP RPC МОЖЕТ использоваться с другими шаблонами обмена сообщениями, но это выходит за рамки данной спецификации.

6.1 Использование RPC в сети Интернет

СЛЕДУЕТ руководствоваться нижеизложенными принципами при развертывании приложений SOAP RPC в сети Интернет.

6.1.1 Идентификация ресурсов RPC

В сети Интернет ресурсы идентифицируются с помощью URI, но, согласно общим соглашениям в программировании, идентификационная информация передается процедурам в параметрах или в именах самих процедур. Например, вызов:

updateQuantityInStock (PartNumber = «123», NewQuantity = «200»)

предполагает, что ресурсом, который будет обновлен, является QuantityInStock для значения параметра PartNumber, равного «123». Соответственно, при сопоставлении с методом или процедурой некоторого языка программирования любые фактические параметры, которые служат для идентификации ресурсов (как номер детали PartNumber в примере выше) должны, когда это целесообразно, быть представлены в URI, к которому адресованы сообщения SOAP. Если имя метода или процедуры идентифицирует или уточняет идентификацию ресурса (как QuantityInStock в примере выше), то при сопоставлении с методом или процедурой некоторого языка программирования такое именование или уточнение должно быть представлено, когда это целесообразно, в URI, к которому адресовано сообщение SOAP. Данная спецификация не определяет никаких стандартных средств представления параметров или имен методов.

Примечание — Соглашения для специфического URI-кодирования имен процедур и параметров, а также для управления включением таких параметров в тело SOAP RPC могут быть установлены совместно с разработкой языков описания интерфейсов Веб-службы. Они могут быть разработаны, если SOAP привязан к определенным языкам программирования, или могут быть установлены на базе конкретных приложений или процедур.

6.1.2 Отличие получения ресурсов от других RPC

Всемирная паутина зависит от механизмов, которые оптимизируют часто выполняемые задачи информационного поиска. В частности протоколы, такие как HTTP [RFC 2616], предоставляют метод «GET», который используется для безопасного извлечения информации. Это такое извлечения информации, которое не изменяет данные, не имеет побочных эффектов и которому из соображений безопасности, не запрещено использовать кэшированные результаты или идентификацию, основанную на URI.

Определенные процедуры или вызовы методов представляют собой запросы на получение информации. Например, вызов:

getQuantityInStock (PartNumber = «123»)

мог бы использоваться для получения количества, установленного в примере выше.

Следующие соглашения могут использоваться для реализации вызовов SOAP и других вызовов RPC в сети Интернет:

- соглашения, описанные в 6.1.1, используются для идентификации ресурсов по URI;
- в случаях, когда все параметры представлены в URI, никакие блоки заголовка SOAP не передаются, и операция извлечения информации является безопасной, используются функция «Веб-метод» (см. 8.4) и шаблон обмена сообщениями «ответ SOAP» (см. 8.3). Соответственно, никакой конверт SOAP не передается для запроса, и значение свойства gost.e-ecolog.ru
gost.e-ecolog.ru авливается в значение «GET». Результатом извлечения информации является ответ SOAP RPC, описанный в 6.2.2;
- в случаях, когда выполняемая операция не является извлечением информации, когда должны быть переданы блоки заголовка SOAP (например, в случае цифровой подписи) или когда извлечение информации не является безопасным, используются функция «Веб-метод» (см. 8.4) и шаблон обмена сообщениями «запрос-ответ» (см. 8.2). Конверт запроса кодируется, как описано в 6.2.1, а результаты извлечения информации описаны в 6.2.2.

Свойство http://gost.e-ecolog.ru устанавливается в значение «POST».

Представление SOAP RPC не определяет никаких других значений для gost.e-ecolog.ru soap/features/web-method/Method.

6.2 RPC и элемент Body SOAP

Как RPC вызовы (за исключением безопасных методов извлечения информации, см. 6.1.2), так и ответы содержат элемент SOAP Body [ИСО/МЭК 40210, подраздел 8.3], используемый для представлений, описанных ниже.

6.2.1 Вызов RPC

Вызов RPC моделируется следующим образом.

- вызов представляется единственной структурой, содержащей исходящее ребро для каждого входящего [in] или входящего/исходящего [in/out] параметра. Структура именуется идентично процедуре или методу. Для представления имен методов, которые не являются допустимыми XML именами, СЛЕДУЕТ использовать соглашения из приложения В;
- у каждого исходящего ребра есть метка, соответствующая имени параметра. Для представления имен параметров, которые не являются допустимыми XML именами, СЛЕДУЕТ использовать соглашения из приложения В.

Приложения МОГУТ обрабатывать вызовы с недостающими параметрами, но также МОГУТ не обрабатывать такие вызовы и возвращать отказы.

6.2.2 Ответ RPC

Ответ RPC моделируется следующим образом:

- ответ представляется единственной структурой, содержащей исходящее ребро для возвращаемого значения и по исходящему ребру для каждого исходящего [out] или входящего/исходящего [in/out] параметра. Имя структуры не имеет значения;

- каждый параметр представляется исходящим ребром с меткой, соответствующей имени параметра. Для представления названий параметров, которые не являются допустимыми XML именами, СЛЕДУЕТ использовать соглашения из приложения В;

- непустое возвращаемое значение представляется следующим образом:

1 ДОЛЖНО присутствовать исходящее ребро с локальным именем result и именем пространства имен «gost.e-ecolog.ru , заканчивающееся в конечном узле;

2 Тип конечного узла — xs:QName и его значение — имя исходящего ребра, которое заканчивается в фактическом возвращаемом значении;

- если возвращаемое значение процедуры пустое, то исходящее ребро с локальным именем result и именем пространства имен «gost.e-ecolog.ru НЕ ДОЛЖНО присутствовать;

- отказы вызова обрабатываются согласно правилам, изложенным в 6.4. Если привязка протокола накладывает дополнительные правила для обработки отказа, то они также ДОЛЖНЫ быть выполнены.

6.2.3 Ограничение на кодирование SOAP

При использовании кодирования SOAP (см. раздел 5) в сочетании с соглашением RPC, описанным здесь, элемент Body SOAP ДОЛЖЕН содержать единственный дочерний *информационный объект-элемент*, который представляет собой сериализованную структуру вызова или ответа RPC.

6.3 RPC и заголовок SOAP

Дополнительная информация, относящаяся к кодированию вызова RPC, но не являющаяся частью формальной сигнатуры процедуры или метода, МОЖЕТ быть представлена в конверте SOAP, переносящем RPC вызов или ответ. Такая дополнительная информация ДОЛЖНА быть представлена в виде блоков заголовка SOAP.

6.4 Отказы RPC

Представление SOAP RPC вводит дополнительные, представленные на внутреннем коде, обозначения отказов SOAP, которые используются в сочетании с кодами отказов (см. [ИСО/МЭК 40210, пункт 8.4.6 «Коды отказа SOAP»]).

Об ошибках, возникающих во время выполнения вызовов RPC, сообщается согласно следующим правилам:

1 Отказ со значением Value элемента Code, равным «env:Receiver», СЛЕДУЕТ генерировать, если получатель временно не может обработать сообщение, например, в случае отсутствия достаточной памяти.

Примечание — Всюду в данном документе термин «значение Value элемента Code» используется как сокращение для «значение дочернего *информационного объекта-элемента* Value *информационного объекта-элемента* Code» (см. [ИСО/МЭК 40210, пункт 8.4.1]).

2 Отказ со значением Value элемента Code, равным «env:DataEncoding Unknown», СЛЕДУЕТ генерировать, если параметры закодированы в кодировку, неизвестную получателю.

3 Отказ со значением Value элемента Code, равным «env:Sender», и значением Value элемента Subcode, равным «rpc:ProcedureNotPresent», МОЖЕТ быть сгенерирован, если получатель не поддерживает определенную процедуру или метод.

Примечание — Всюду в данном документе термин «значение Value элемента Subcode» используется как сокращение для «значение дочернего *информационного объекта-элемента* Value *информационного объекта-элемента* Subcode» (см. [ИСО/МЭК 40210, подпункт 8.4.1.2]).

4 Отказ со значением Value элемента Code, равным «env:Sender», и значением Value элемента Subcode, равным «rpc:BadArguments», ДОЛЖЕН быть сгенерирован, когда получатель не может произ-

вести анализ параметров или когда есть несоответствие в количестве и/или типах параметров между тем, что ожидает получатель и тем, что было отправлено.

5 Другие отказы, возникающие в расширении или в приложениях, СЛЕДУЕТ генерировать как описано в пункте «Коды Отказа SOAP» [ИСО/МЭК 40210, пункт 8.4.1].

Во всех случаях значения *информационных объектов-элементов* Detail и Reason определяются реализацией. Детали их использования МОГУТ задаваться внешним документом.

Примечание — В ответ на вызов RPC отправители могут получать различные отказы из перечисленных выше, если получатель не поддерживает описанное здесь (необязательное) соглашение RPC.

7 Соглашение для описания функций и привязок

В данном разделе определяется соглашение, описывающее функции (включая шаблоны обмена сообщениями (ШОС)) и привязки в терминах свойств и значений свойств. Соглашения достаточно для описания распределенных состояний спецификаций функций и привязок, как требует спецификация инфраструктуры привязок [ИСО/МЭК 40210, раздел 7]. Оно также используется для описания ШОС «запрос-ответ» (см. 8.2), ШОС «ответ SOAP» (см. 8.3), функции SOAP «Веб-метод» (см. 8.4) и привязки SOAP к HTTP (см. раздел 9) всюду в настоящем стандарте. Помимо самого соглашения, в данном разделе определена неформальная модель, описывающая процесс передачи свойства через систему SOAP. Отметим, что данная модель только иллюстративна и не включает в себя каких-либо ограничений на структуру или иерархическое представление любой конкретной реализации SOAP.

7.1 Модель и свойства

В целом, сообщение SOAP — это информация, которой один узел SOAP хочет обменяться с другим узлом SOAP согласно определенным соглашениям, включая шаблоны обмена сообщениями. Кроме того, может присутствовать информация, важная для обмена сообщениями, но не являющаяся частью самих сообщений. Такую информацию иногда называют метаданными сообщения. В модели сообщения любые метаданные сообщения и различные информационные объекты, определяющие функции, представляются как абстракции, называемые свойствами.

7.1.1 Свойства

В соответствии с соглашением свойства представляются следующим образом:

- свойства именуются посредством URI;
- где это уместно, спецификации, вводящей свойство, СЛЕДУЕТ определить тип значения свойства в соответствии со спецификацией XML Schema (см. [XML Schema Part 1], [XML Schema Part 2]).

7.1.2 Область применения свойства

Свойства в узле SOAP различаются по области применения и по источникам их значений. Как показано ниже на рисунке 1, свойства делятся на две группы: свойства для обмена сообщениями и свойства с более широкой областью применения, относящиеся к различным контейнерам. Эти группы называются «контекст обмена сообщениями» и «контекст окружения», соответственно. Все свойства, независимо от областей их применения, совместно используются узлом SOAP и определенной привязкой.

7.1.2.1 Контекст обмена сообщениями

Контекст обмена сообщениями — это набор свойств, область применения которых ограничена экземпляром данного шаблона обмена сообщениями. Пример свойства контекста обмена сообщениями — идентификатор используемого шаблона обмена сообщениями.

7.1.2.2 Контекст окружения

Контекст окружения — это набор свойств, область применения которых выходит за рамки экземпляра данного шаблона обмена сообщениями. Примеры свойств контекста окружения — IP-адрес узла SOAP или текущие дата и время.

Значения свойств в контексте окружения могут зависеть от локальных условий (на рисунке 1 это показано стрелкой, исходящей из контейнера «контекст окружения»). В частности, на свойства в примере может повлиять идентификатор пользователя операционной системы, от имени которого выполняется обмен сообщениями. Отображение информации конкретной реализацией в такие свойства выходит за рамки инфраструктуры привязки, хотя она включает в себя абстрактное представление такой информации как свойства.

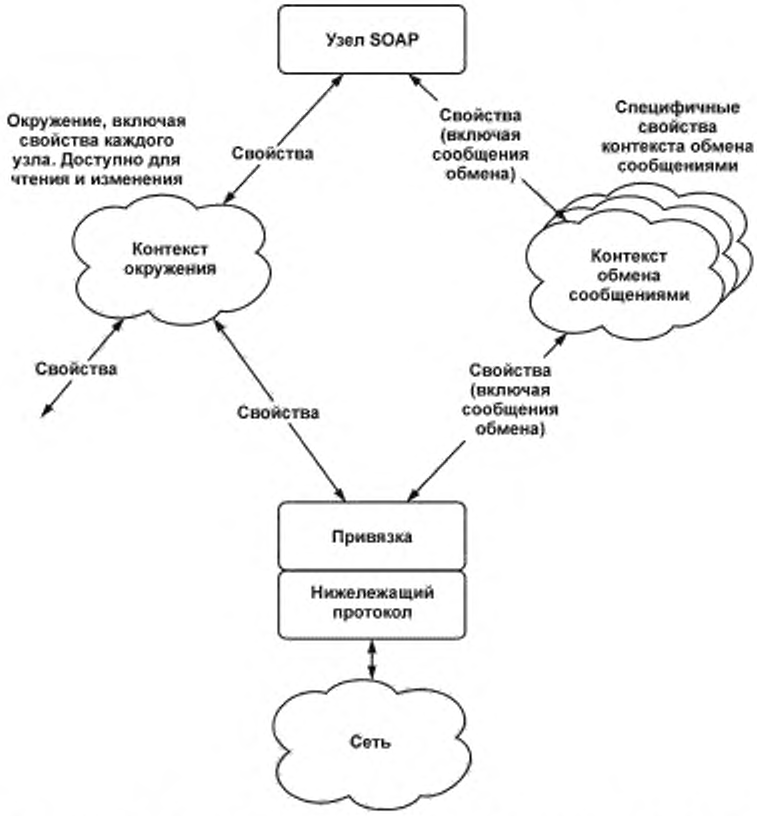


Рисунок 1 — Модель, описывающая свойства, совместно используемые SOAP и привязкой

7.1.3 Свойства и функции

Функция может использовать сразу несколько свойств, и, наоборот, одно свойство может участвовать в нескольких функциях. Например, свойства «идентификатор пользователя» и «пароль» могут определять функцию «аутентификация». Еще пример: одно свойство «идентификатор сообщения» может использоваться и в функции «транзакция», и в функции «корреляция сообщений».

8 Шаблоны обмена сообщениями и функции SOAP

8.1 Соглашения о свойствах для шаблонов обмена сообщениями SOAP

В таблице 2 описаны свойства (в соответствии с представленными в настоящем стандарте соглашениями об именовании свойств), на которые опирается описание шаблонов обмена сообщениями (ШОС). Другие свойства могут быть включены в спецификацию конкретных ШОС, но свойства, приведенные в данной таблице, в целом применимы ко всем ШОС.

Таблица 2 — Определения свойств, на которые опирается описание ШОС

Имя свойства: http://www.w3.org/2003/05/soap/bindingFramework/ExchangeContext/ExchangePatternName. **Значение:** имя используемого ШОС. **Тип:** xs:anyURI
Имя свойства: gost.e-ecolog.ru **Значение:** значение, которое обозначает определенную шаблоном, независимую от привязки причину отказа обмена сообщениями. Спецификации привязки нижележащего протокола могут определять свойства для передачи деталей отказа, специфических для привязки. **Тип**: xs:anyURI

Окончание таблицы 2

Имя свойства: http://gost.e-ecolog.ru **Значение:** идентификатор определяемой шаблоном роли локального узла SOAP, участвующего в обмене сообщениями. **Тип:** xs:anyURI
Имя свойства: http://www.w3.org/2003/05/soap/bindingFramework/ExchangeContext/State. **Значение:** идентификатор текущего состояния обмена сообщениями. Этим значением управляет экземпляр привязки, и его значение может использоваться другими объектами, контролирующими процесс обмена сообщениями. **Тип:** xs:anyURI

8.2 Шаблон обмена сообщениями SOAP «запрос-ответ»

В данном разделе определяется шаблон обмена сообщениями (ШОС) «запрос-ответ». В нем описывается абстрактное представление работы этого ШОС. Данный раздел не предназначен для описания настоящей реализации или для предложений о том, как должна быть структурирована реальная реализация.

8.2.1 Имя функции SOAP

Идентификатор данного шаблона обмена сообщениями: URI [ИСО/МЭК 40210, подраздел 5.6] «gost.e-ecolog.ru

8.2.2 Описание

ШОС SOAP «запрос-ответ» определяет шаблон для обмена сообщениями SOAP, в котором за сообщением, исполняющим роль запроса, следует сообщение, исполняющее роль ответа. Сообщение-ответ МОЖЕТ содержать конверт SOAP, в противном случае ответ ДОЛЖЕН быть определяемым привязкой сообщением, указывающим, что запрос был получен. При отсутствии отказа нижележащего протокола данный ШОС состоит ровно из двух сообщений.

При нормальном функционировании обмена сообщениями, соответствующими ШОС «запрос-ответ», сообщение-запрос сначала передается от запрашивающего узла SOAP к отвечающему узлу SOAP. После успешной обработки сообщения-запроса отвечающим узлом SOAP сообщение-ответ передается от отвечающего узла SOAP запрашивающему узлу SOAP.

Аварийная работа во время обмена сообщениями «запрос-ответ» может быть вызвана отказом при передаче сообщения запроса, отказом отвечающего узла SOAP на обработку сообщения запроса или отказом при передаче сообщения ответа. Информирование о таких отказах может быть опущено на одном или обоих запрашивающем и отвечающем узлах SOAP, а также может быть осуществлено посредством генерации отказа SOAP или отказа, определяемого привязкой (см. 8.2.4). Кроме того, в случае аварийной работы каждый узел SOAP, участвующий в обмене сообщениями, может по-разному определять успешность выполнения операции обмена сообщениями.

Область применения ШОС «запрос-ответ» ограничена обменом сообщениями запроса и ответа между одним запрашивающим и одним отвечающим узлами SOAP. Данный шаблон не налагает требований ни на корреляции между множественными запросами, ни на определенную синхронизацию множественных запросов. Реализации МОГУТ поддерживать несколько запросов (и связанную с ними обработку ответов) одновременно.

8.2.3 Описание конечного автомата

ШОС «запрос-ответ» определяет ряд свойств, описанных в таблице 3.

Таблица 3 — Определения свойств для ШОС «запрос-ответ»

Имя свойства: gost.e-ecolog.ru **Значение:** абстрактная структура, представляющая текущее исходящее сообщение в обмене сообщениями. Данная структура абстрагирует и конверт SOAP и все остальные информационные структуры, которые передаются вместе с конвертом. **Тип:** не определен
Имя свойства: http://gost.e-ecolog.ru **Значение:** абстрактная структура, представляющая текущее входящее сообщение в обмене сообщениями. Данная структура абстрагирует и конверт SOAP и все остальные информационные структуры, которые передаются вместе с конвертом. **Тип:** не определен

Окончание таблицы 3

Имя свойства: http://www.w3.org/2003/05/soap/mep/ImmediateDestination. **Значение:** идентификатор следующего пункта назначения исходящего сообщения. **Тип:** xs:anyURI
Имя свойства: gost.e-ecolog.ru **Значение**: идентификатор непосредственного отправителя входящего сообщения. **Тип**: xs:anyURI

Чтобы инициировать обмен сообщениями, удовлетворяющий шаблону обмена сообщений «запрос-ответ», отвечающий узел SOAP создает экземпляр локального контекста обмена сообщениями. В таблице 4 описана инициализация контекста.

Таблица 4 — Создание экземпляра контекста обмена сообщениями для запрашивающего узла SOAP

Имя свойства: gost.e-ecolog.ru **Значение:** «gost.e-ecolog.ru
Имя свойства: gost.e-ecolog.ru **Значение:** «None» **Примечания:** относительный URI, базовый URI которого — значение свойства с именем gost.e-ecolog.ru 12/soap/bindingFramework/ExchangeContext/ExchangePatternName
Имя свойства: gost.e-ecolog.ru **Значение:** «RequestingSOAPNode». **Примечания:** относительный URI, базовый URI которого — значение свойства с именем gost.e-ecolog.ru soap/bindingFramework/ExchangeContext/ExchangePatternName
Имя свойства: http://www.w3.org/2003/05/soap/bindingFramework/ExchangeContext/State. **Значение:** «Init». **Примечания:** относительный URI, базовый URI которого — значение свойства с именем http://gost.e-ecolog.ru soap/bindingFramework/ExchangeContext/Role
Имя свойства: gost.e-ecolog.ru **Значение:** абстрактное сообщения запроса
Имя свойства: gost.e-ecolog.ru **Значение:** идентификатор (URI), указывающий на отвечающий узел SOAP

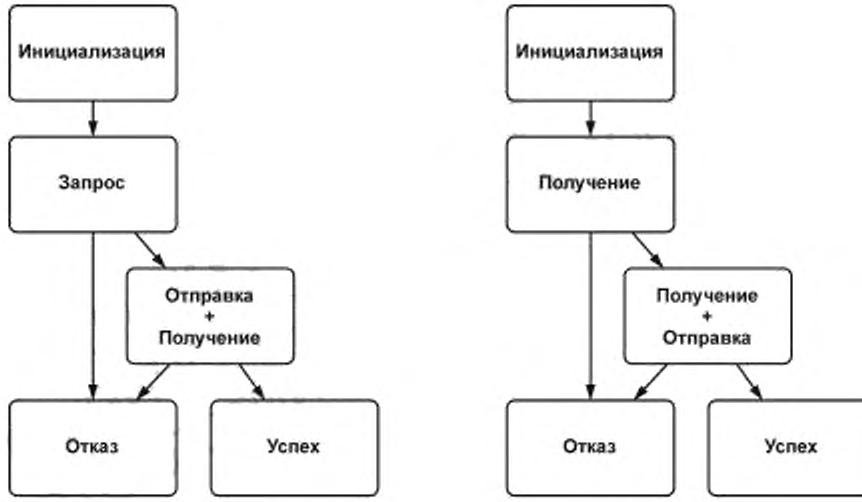
Могут присутствовать и другие свойства, имеющие отношения к операциям экземпляра контекста обмена сообщениями. Такие свойства инициализируются согласно их собственным спецификациям.

После инициализации контекста обмена сообщениями, управление окружением передается (соответствующему спецификации) локальному экземпляру привязки.

На рисунке 2 представлены логические переходы между состояниями запрашивающего и отвечающего узлов SOAP во время обмена сообщениями. В каждом узле SOAP локальные экземпляры привязки обновляют (логически) значение свойства gost.e-ecolog.ru

ExchangeContext/State для отражения текущего состояния процесса обмена сообщениями. Имена состояний — относительные URI, заданные относительно значения базового URI, которое содержится в свойстве gost.e-ecolog.ru локального контекста обмена сообщениями.

Когда локальный экземпляр привязки отвечающего узла SOAP начинает получать входящее сообщение запроса, он логически создает экземпляр окружения обмена сообщениями. В таблице 5 представлены свойства, которые привязка инициализирует при создании экземпляра контекста.



Запрашиваемый узел SOAP **Отвечающий узел SOAP**

Рисунок 2 — Диаграмма переходов ШОС «запрос-ответ»

Таблица 5 — Создание контекста обмена сообщениями для входящего запроса в отвечающем узле SOAP

Имя свойства: http://gost.e-ecolog.ru **Значение:** «http://www.w3.org/2003/05/soap/mep/request-response/». **Примечание:** инициализируется как можно раньше в жизненном цикле обмена сообщениями
Имя свойства: http://gost.e-ecolog.ru **Значение:** «None». **Примечание:** относительный URI, базовый URI которого — значение свойства с именем gost.e-ecolog.ru soap/bindingFramework/ExchangeContext/ExchangePatternName
Имя свойства: http://www.w3.org/2003/05/soap/bindingFramework/ExchangeContext/Role. **Значение:** «RespondingSOAPNode». **Примечания:** относительный URI, базовый URI которого — значение свойства с именем gost.e-ecolog.ru soap/bindingFramework/ExchangeContext/ExchangePatternName. Инициализируется как можно раньше в жизненном цикле обмена сообщениями
Имя свойства: http://gost.e-ecolog.ru **Значение:** «Init». **Примечания:** относительный URI, базовый URI которого — значение свойства с именем gost.e-ecolog.ru soap/bindingFramework/ExchangeContext/Role

Когда происходит переход между состояниями у запрашивающего и отвечающего узлов SOAP, локальный экземпляр привязки логически обновляет отдельные свойства. В таблицах 6 и 7 описаны эти обновления для запрашивающих и отвечающих узлов SOAP соответственно.

Таблица 6 — Переходы для запрашивающего узла SOAP

Текущее состояние	
Init *Инициализация*	**Условие перехода**: безусловный переход. **Следующее состояние**: «Запрос». **Действие**: начать передачу абстрактного сообщения запроса, содержащегося в http://www.w3.org/2003/05/soap/mep/OutboundMessage

Окончание таблицы 6

Текущее состояние	
Requesting *Запрос*	**Условие перехода:** отказ при передаче сообщения. **Следующее состояние:** «Отказ». **Действие:** установить ExchangeContextFailureReason в «transmissionFailure»
Requesting *Запрос*	**Условие перехода:** начато получение сообщения ответа. **Следующее состояние:** «Отправка и получение». **Действие:** изменить значение http://www.w3.org/2003/05/soap/mep/ImmediateSender для обозначения отправителя сообщения ответа (может отличаться от значений в http://). Начать конструировать абстрактное сообщение ответа в http://www.w3.org/2003/05/soap/mep/InboundMessage
Sending+ Receiving *Отправка и получение*	**Условие перехода:** отказ при передаче сообщения. **Следующее состояние:** «Отказ». **Действие:** установить значение ExchangeContext/FailureReason в «exchangeFailure»
	Условие перехода: завершена отправка сообщения запроса. Завершено получение сообщения ответа. **Следующее состояние:** «Успех». **Действие:** если в ответе получен конверт SOAP (т.е. в http://www.w3.org/2003/05/soap/mep/InboundMessage), то обработать его согласно модели обработки SOAP

Таблица 7 — Переходы для отвечающего узла SOAP

Текущее состояние	
Init *Инициализация*	**Условие перехода:** начато получение сообщения запроса. **Следующее состояние:** «Получение». **Действие:** изменить значение для обозначения отправителя сообщения запроса (если возможно его определить). Начать конструировать абстрактное сообщение запроса InboundMessage. Передать управление контекста обмена сообщениями процессору SOAP
Receiving *Получение*	**Условие перехода:** отказ приема сообщения. **Следующее состояние:** «Отказ». **Действие:** установить значение переменной http://bindingFramework/ExchangeContext/FailureReason в «receptionFailure»
	Условие перехода: начало сообщения ответа доступно в mep/OutboundMessage. **Следующее состояние:** «Получение и отправка». **Действие:** начать передачу сообщения ответа. Если в абстрактном сообщении в w3.org/2003/05/soap/mep/OutboundMessage представлен конверт, включить его в сообщение ответа
Receiving + Sending *Получение и отправка*	**Условие перехода:** отказ при обмене сообщениями. **Следующее состояние:** «Отказ». **Действие:** установить значение переменной bindingFramework/ExchangeContext/FailureReason в «exchangeFailure»
	Условие перехода: завершено получение сообщения запроса. Завершена отправка сообщения ответа. **Следующее состояние:** «Успех»

Привязка, которая реализует данный ШОС, МОЖЕТ предоставлять потоковую передачу ответов SOAP. Это означает, что отвечающие узлы SOAP МОГУТ начать передачу ответа SOAP в то время, пока запрос SOAP все еще находится в процессе получения и обработки. Если узлы SOAP реализуют привязку, поддерживающую потоковую передачу, применяются следующие правила:

- ДОЛЖНЫ быть выполнены все правила по 6.2, относящиеся к потоковой передачи индивидуальных сообщений SOAP, и для запросов, и для ответов SOAP;

- при использовании потоковой передачи привязки SOAP запрашивающие узлы SOAP ДОЛЖНЫ избегать взаимной блокировки при получении и при необходимости обработки информации из ответа SOAP, в то время как передается запрос SOAP.

Примечание — В зависимости от используемой реализации и размера обрабатываемых сообщений, это правило МОЖЕТ потребовать от приложений SOAP, чтобы потоковая обработка ответа уровня приложения происходила параллельно с генерацией запроса;

- запрашивающий узел SOAP МОЖЕТ перейти в состояние «Fail», и таким образом прервать передачу исходящего запроса SOAP, основываясь на информации, содержащейся в ответе SOAP входящего потока.

8.2.4 Обработка отказов

Участвующие в ШОС «запрос-ответ» узлы SOAP во время работы могут генерировать отказы SOAP.

Если отказ SOAP сгенерирован отвечающим узлом SOAP, находящимся в состоянии «Получение», то отказ SOAP помещается в gost.e-ecolog.ru, и конечный автомат переходит в состояние «Получение и отправка».

Данный ШОС не специфицирует порождение или обработку отказов SOAP, сгенерированных запрашивающим узлом SOAP во время обработки сообщения ответа, в состояниях, следующих за состоянием «Успех» в таблице переходов запрашивающего узла SOAP (см. таблицу 6).

8.3 Шаблон обмена сообщениями SOAP «ответ SOAP»

Данный раздел определяет шаблон обмена сообщениями (ШОС) «ответ SOAP». В нем описывается абстрактное представление работы данного ШОС. Данный раздел не предназначен для описания реальной реализации или для предложений, как должна быть структурирована реальная реализация.

8.3.1 Имя функции SOAP

Идентификатор данного шаблона обмена сообщениями: URI [ИСО/МЭК 40210, подраздел 5.6] «gost.e-ecolog.ru.

8.3.2 Описание

ШОС «ответ SOAP» определяет шаблон обмена сообщениями, в котором за сообщением, которое не является сообщением SOAP, но выступает в роли запроса, следует сообщение SOAP, выступающее в роли ответа. При условии, что нет отказа нижележащего протокола, данный ШОС состоит ровно из двух сообщений, причем только одно из них является сообщением SOAP:

- запрос передается методом, определенным привязкой, который не включает конверт SOAP, и, следовательно, не требует SOAP-обработки получающим узлом SOAP;

- сообщение ответа, которое содержит конверт SOAP. ШОС завершается обработкой конверта SOAP в соответствии с правилами модели обработки SOAP (см. [ИСО/МЭК 40210, раздел 5]).

Аварийная работа во время обмена сообщениями «ответ SOAP» может быть вызвана отказом при передаче сообщения запроса или ответа. Информирование о таких отказах может быть опущено на одном или обоих запрашивающем и отвечающем узлах SOAP или может быть осуществлено с помощью генерации отказа SOAP или отказа, определенного привязкой (см. 8.3.4). Кроме того, во время аварийной работы каждый узел SOAP, участвующий в обмене сообщениями, может по-разному определять успешность выполнения операции обмена сообщениями.

Область применения ШОС «ответ SOAP» ограничивается запросом на сообщение-ответ при обмене сообщениями между одним запрашивающим и одним отвечающим узлами SOAP. Данный шаблон не налагает требований ни на корреляции между множественными запросами, ни на определенную синхронизацию множественных запросов. Реализации МОГУТ поддерживать одновременно несколько запросов и связанную с ними обработку ответов.

Примечание — Данный ШОС не может использоваться в сочетании с функциями, которые задаются в блоках заголовка SOAP в запросе, потому что в данном случае нет конверта SOAP, в который можно было бы их включить.

8.3.3 Описание конечного автомата

ШОС «ответ SOAP» определяет ряд свойств, описанных в таблице 8.

Таблица 8 — Определения свойств для ШОС «ответ SOAP»

Имя свойства	Описание свойства	Тип свойства
gost.e-ecolog.ru mep/OutboundMessage	Абстрактная структура, представляющая текущее исходящее сообщение в обмене сообщениями. Данная структура абстрагирует и инфо-набор конверта SOAP (который МОЖЕТ отсутствовать), и любые другие информационные структуры, передающиеся вместе с конвертом	Не задан
gost.e-ecolog.ru mep/InboundMessage	Абстрактная структура, представляющая текущее входящее сообщение в обмене сообщениями. Данная структура абстрагирует и инфо-набор конверта SOAP (который МОЖЕТ отсутствовать), и любые другие информационные структуры, передающиеся вместе с конвертом	Не задан
http://www.w3.org/2003/05/soap/ mep/ImmediateDestination	Идентификатор непосредственного узла назначения исходящего сообщения	xs:anyURI
gost.e-ecolog.ru mep/ImmediateSender	Идентификатор непосредственного отправителя входящего сообщения	xs:anyURI

Чтобы инициировать обмен сообщениями, который соответствует ШОС «ответ SOAP», запрашивающий узел SOAP создает экземпляр локального контекста обмена сообщениями. В таблице 9 описано, как инициализируется контекст.

Таблица 9 — Создание экземпляра контекста обмена сообщениями для запрашивающего узла SOAP

Имя свойства: gost.e-ecolog.ru **Значение**: gost.e-ecolog.ru
Окончание таблицы 9
Имя свойства: gost.e-ecolog.ru **Значение**: «None». **Примечания**: относительный URI, который будет разрешен относительно значения свойства с именем http:// gost.e-ecolog.ru
Имя свойства: gost.e-ecolog.ru **Значение**: «RequestingSOAPNode». **Примечания**: относительный URI, который будет разрешен относительно значения свойства с именем http:// www.w3.org/2002/12/soap/bindingFramework/ExchangeContext/ExchangePatternName
Имя свойства: gost.e-ecolog.ru **Значение**: «Init». **Примечания**: относительный URI, базовый URI которого — значение свойства с именем gost.e-ecolog.ru soap/bindingFramework/ExchangeContext/Role
Имя свойства: gost.e-ecolog.ru **Значение**: абстракция сообщения запроса, которое не включает инфо-набор конверта SOAP
Имя свойства: gost.e-ecolog.ru **Значение**: идентификатор (URI), который обозначает отвечающий узел SOAP

Могут присутствовать и другие свойства, имеющие отношения к операциям экземпляра контекста обмена сообщениями. Такие свойства инициализируются согласно их собственным спецификациям.

Как только контекст обмена сообщениями инициализирован, управление контекстом передается к соответствующему локальному экземпляру привязки.

На рисунке 3 представлены логические переходы между состояниями запрашивающего и отвечающего узлов SOAP во время обмена сообщениями. В каждом узле SOAP локальные экземпляры привязки логически обновляют значение свойства gost.e-ecolog.ru
State для отражения текущего состояния процесса обмена сообщениями. Имена состояний — относительные URI, заданные относительно значения базового URI, которое содержится в свойстве gost.e-ecolog.ru w3.org/2003/05/soap/bindingFramework/ExchangeContext/Role локального контекста обмена сообщениями.

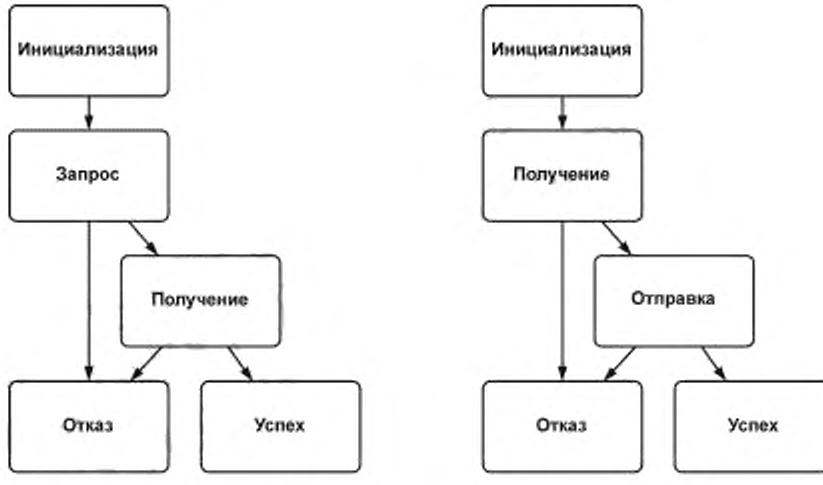


Рисунок 3 — Диаграмма переходов ШОС «ответ SOAP»

Когда локальный экземпляр привязки в отвечающем узле SOAP начинает получать входящее сообщение запроса, он логически создает экземпляр контекст обмена сообщениями. В таблице 10 описаны свойства, которые привязка инициализирует как часть создания экземпляра контекста.

Таблица 10 — Создание экземпляра контекста обмена сообщениями для входящего сообщения-запроса

Имя свойства: gost.e-ecolog.ru **Значение:** gost.e-ecolog.ru **Примечания:** инициализируется как можно раньше в жизненном цикле обмена сообщениями
Имя свойства: gost.e-ecolog.ru **Значение:** «None». **Примечания:** относительный URI, который будет разрешен относительно значения свойства с именем gost.e-ecolog.ru
Имя свойства: http://gost.e-ecolog.ru **Значение:** «RespondingSOAPNode». **Примечания:** относительный URI, который будет разрешен относительно значения свойства с именем http://www.w3.org/2002/12/soap/bindingFramework/ExchangeContext/ExchangePatternName, инициализируется во время жизненного цикла обмена сообщениями как можно раньше
Имя свойства: http://www.w3.org/2003/05/soap/bindingFramework/ExchangeContext/State. **Значение:** «Init». **Примечания:** относительный URI, который будет разрешен относительно значения свойства с именем gost.e-ecolog.ru

Когда происходит переход между состояниями запрашивающего и отвечающего узлов SOAP, локальный экземпляр привязки логически обновляет отдельные свойства. В таблицах 11 и 12 описаны эти обновления для запрашивающих и отвечающих узлов SOAP.

Таблица 11 — Переходы запрашивающего узла SOAP

Текущее состояние	
Init *Инициализация*	**Условие перехода**: безусловный переход. **Следующее состояние**: «Requesting». **Действие**: инициировать запрос

Окончание таблицы 11

Текущее состояние	
Requesting *Запрос*	**Условие перехода**: отказ при передаче сообщения. **Следующее состояние**: «Fail». **Действие**: установить http://gost.e-ecolog.ru FailureReason в «transmissionFailure»
	Условие перехода: начато получение сообщения ответа. **Следующее состояние**: «Receiving». **Действие**: установить значение http://www.w3.org/2003/05/soap/mep/ImmediateSender, для обозначения отправителя сообщения ответа (может отличаться от значений в http://www.w3.org/2003/05/soap/mep/ImmediateDestination). Начать конструировать абстрактное сообщение ответа в gost.e-ecolog.ru
Receiving Получение	**Условие перехода**: отказ при обмене сообщениями. **Следующее состояние**: «Fail». **Действие**: установить значение http://www.w3.org/2003/05/soap/bindingFramework/ExchangeContext/FailureReason в «exchangeFailure».
	Условие перехода: завершено получение сообщения ответа. **Следующее состояние**: «Success»

Таблица 12 — Переходы отвечающего узла SOAP

Текущее состояние	
Init Инициализация	**Условие перехода**: начато получение запроса. **Следующее состояние**: «Receiving». **Действие**: установить значение переменной gost.e-ecolog.ru gost.e-ecolog.ru ния отправителя сообщения запроса (если его можно определить). Передать управление контекстом обмена сообщениями процессору SOAP
Receiving Получение	**Условие перехода**: отказ при приеме сообщения. **Следующее состояние**: «Fail». **Действие**: установить значение переменной gost.e-ecolog.ru 05/soap/bindingFramework/ExchangeContext/FailureReason в «receptionFailure»
	Условие перехода: начало сообщения ответа доступно в gost.e-ecolog.ru mep/OutboundMessage. **Следующее состояние**: «Sending». **Действие**: инициировать передачу абстрактного сообщения ответа, содержащегося в gost.e-ecolog.ru
Sending *Отправка*	**Условие перехода**: отказ при передаче сообщения. **Следующее состояние**: «Fail». **Действие**: установить gost.e-ecolog.ru FailureReason в «exchangeFailure»
	Условие перехода: отправка сообщения ответа завершена. **Следующее состояние**: «Success»

8.3.4 Обработка отказов

Во время работы ШОС «ответ SOAP» участвующие узлы SOAP могут генерировать отказы SOAP.

Если отказ SOAP сгенерирован отвечающим узлом SOAP, находящимся в состоянии «Получение», то отказ SOAP сохраняется в http://www.w3.org/2003/05/soap/mep/OutboundMessage, и конечный автомат переходит в состояние «Отправка».

Данный ШОС не специфицирует порождение или обработку отказов SOAP, сгенерированных запрашивающим узлом SOAP во время обработки сообщения ответа в состояниях, следующих за состоянием «Успех» в таблице переходов запрашивающего узла SOAP (см. таблицу 11).

8.4 Функция SOAP «Веб-метод»

Данный раздел определяет функцию SOAP «Веб-метод».

8.4.1 Имя функции SOAP

Идентификатор функции SOAP «Веб-метод»: URI [ИСО/МЭК 40210, подраздел 5.6] «gost.e-ecolog.ru .

8.4.2 Описание

Нижележащие протоколы, разработанные для использования в сети Интернет, предусматривают для работы с ресурсами небольшой набор Веб-методов, таких как GET, PUT, POST и DELETE. Эти методы формально определены в спецификации HTTP [RFC 2616], но и нижележащие базовые протоколы могут также их поддерживать. Привязкам к HTTP или другим подобным протоколам СЛЕДУЕТ использовать функцию SOAP «Веб-метод» для предоставления приложениям контроля над Веб-методами, которые будут применяться при передаче сообщений SOAP.

Привязкам, поддерживающим эту функцию, СЛЕДУЕТ использовать подходящую реализацию этой функции, если она предоставляется базовым протоколом; например, привязка HTTP, описанная в данной спецификации, представляет метод «GET» как запрос HTTP GET и метод «POST» как запрос HTTP POST (см. раздел 9). Привязкам, поддерживающим эту функцию, СЛЕДУЕТ предоставлять получающему узлу информацию о Веб-методе, использованном для передачи.

Функция SOAP «Веб-метод» МОЖЕТ быть реализована привязкой к нижележащим транспортным протоколам, у которых нет предпочтительных реализаций определенных Веб-методов (например, не различающих GET и POST). Таким привязкам СЛЕДУЕТ предоставлять получающему узлу информацию о Веб-методе, использованном при передаче, но не требуется предпринимать дополнительных действий для поддержки данной функции.

8.4.3 Конечный автомат функции SOAP «Веб-метод»

Функция SOAP «Веб-метод» определяет единственное свойство, которое описано в таблице 13.

Таблица 13 — Определение свойства для функции SOAP «Веб-метод»

Имя свойства	Описание свойства	Тип свойства
http://www.w3.org/2003/05/soap/features/web-method/Method	Один из «GET», «POST», «PUT», «DELETE» (или другие, которые позднее могут быть добавлены к набору Веб-методов)	Не определен

Данная спецификация предусматривает использование функции SOAP «Веб-метод» в сочетании с шаблонами обмена сообщениями (см. 8.2 и 8.3). Данная функция МОЖЕТ использоваться с другими ШОС тогда и только тогда, когда это предусмотрено в спецификациях других ШОС.

Узел, отправляющий сообщение запроса, ДОЛЖЕН предоставлять значение для свойства http:// gost.e-ecolog.ru . Привязке протокола, поддерживающей эту функцию, СЛЕДУЕТ установить значение свойства gost.e-ecolog.ru Method в получающем узле, чтобы оно соответствовало значению, указанному отправителем; средство передачи этого свойства определяется привязкой.

Отвечающему узлу СЛЕДУЕТ отвечать способом, согласующимся с запрашиваемым Веб-методом (например, вызов метода «GET» должен привести к извлечению представления идентифицированного ресурса), или, если Веб-метод не поддерживается, СЛЕДУЕТ произвести отказ специфичным для приложения способом.

Привязки, реализующие эту функцию, ДОЛЖНЫ использовать шаблон обмена сообщениями с семантикой, которая совместима с выбранным Веб-методом. Например, шаблон обмена сообщениями SOAP «ответ SOAP» (см. 8.3) совместим с «GET».

8.5 Функция SOAP «Действие»

Данный раздел определяет функцию SOAP «Действие».

8.5.1 Имя функции SOAP

Идентификатор функции SOAP «Действие»: URI [ИСО/МЭК 40210, подраздел 5.6] «gost.e-ecolog.ru w3.org/2003/05/soap/features/action/».

8.5.2 Описание

Многие привязки SOAP 1.2 к нижележащим протоколам, вероятно, используют тип медиа «application/soap+xml» (см. приложение А) для передачи сериализованных XML сообщений SOAP. Тип медиа определяет дополнительный параметр action, который, помимо прочего, может использоваться для оптимизации обработки или маршрутизации. Функция «Действие» определяет известные URI, чтобы показать, что параметр action поддерживается в привязке, которая использует MIME, и также для того, чтобы ссылаться на значение самого параметра.

8.5.3 Конечный автомат функции SOAP «Действие»

Функция SOAP «Действие» определяет единственное свойство, которое описано в таблице 14. Значение данного свойства ДОЛЖНО быть абсолютным URI [RFC 3986] и НЕ ДОЛЖНО быть пустым.

Таблица 14 — Определение свойства для функции SOAP «Действие»

Имя свойства	Тип свойства
http://www.w3.org/2003/05/soap/features/action/Action	xsd:anyURI

gost.e-ecolog.ru

Если у свойства есть значение в отправителе SOAP, использующем привязку, поддерживающую данную функцию, то отправитель ДОЛЖЕН использовать значение свойства в качестве значения параметра action в указателе типа медиа.

Наоборот, если получатель SOAP обнаруживает значение в параметре action указателя типа медиа, то получатель ДОЛЖЕН сохранить это значение в свойстве http://www.w3.org/2003/05/soap/features/action/Action.

9 Привязка SOAP к HTTP

9.1 Введение

Привязка SOAP к HTTP обеспечивает использование HTTP для операций SOAP. Привязка соответствует спецификации «Структура привязки протокола SOAP» (см. [ИСО/МЭК 40210, раздел 7]) и поддерживает шаблоны обмена сообщениями и функции, описанные в разделе 8.

9.1.1 Опциональность

Привязка SOAP к HTTP опциональна, узлам SOAP НЕ ТРЕБУЕТСЯ ее реализовывать. Про узел SOAP, который правильно и полностью реализует привязку SOAP к HTTP, говорят, что он «соответствует привязке SOAP 1.2 к HTTP».

Спецификация SOAP версии 1.2 не исключает возможность разработки других привязок к HTTP или привязок к другим протоколам, но связь с узлами, использующими подобные другие привязки, не является целью. Заметим, что другие привязки SOAP к HTTP МОГУТ быть реализованы с целью поддержки других шаблонов обмена сообщениями, отличающихся от описанных в 8.2 и 8.3. Такие альтернативные привязки МОГУТ поэтому использовать функции и коды состояния HTTP, не требуемые для данной привязки.

9.1.2 Использование HTTP

Привязка SOAP к HTTP определяет базовый URI согласно правилам HTTP/1.1 [RFC 2616], то есть, базовый URI — это URI запроса HTTP (значение поля Request-URI в строке запроса) или значение поля Content-Location заголовка HTTP.

Данная привязка SOAP к HTTP предназначена для надлежащего использования протокола HTTP как протокола приложения. Например, успешные ответы содержат коды состояния 200 или 202, а отказы обозначаются как 4XX или 5XX. Данная привязка не предназначена для использования всех функций HTTP; она определяет использование протокола HTTP исключительно с целью связывания с другими узлами SOAP, реализующими эту же привязку. Поэтому данная привязка SOAP к HTTP не определяет использование и/или значение всех возможных методов HTTP, полей заголовка и кодов ответов. Она определяет только то, что имеет отношение к шаблонам, описанным в 8.2 и 8.3, или то, что, вероятно, будет представлено механизмами HTTP (такими, как прокси), действующими между узлами SOAP.

Определенные необязательные функции, предоставляемые данной привязкой, зависят от возможностей, предоставленных HTTP/1.1, например согласование содержимого. Поэтому реализациям СЛЕДУЕТ использовать HTTP/1.1 [RFC 2616] (или более поздние совместимые версии, которые

используют тот же старший номер версии[1]. Реализации МОГУТ также быть развернуты, используя HTTP/1.0, несмотря на то, что в таком случае определенные дополнительные функции привязки могут не быть предоставлены.

Примечание — Реализации привязки SOAP к HTTP должны учитывать тот факт, что промежуточные узлы HTTP/1.0 (которые также могут быть промежуточными узлами SOAP) могут изменять представление сообщений SOAP даже в ситуациях, где и начальный отправитель SOAP, и конечный получатель SOAP используют HTTP/1.1.

9.1.3 Взаимодействие с реализациями HTTP, не являющимися реализациями SOAP

В случае использования с шаблонами обмена сообщениями SOAP, прежде всего «ответ SOAP» (см. 8.3), сообщения HTTP, сформированные данной привязкой, скорее всего, не будут отличаться от сформированных реализацией HTTP, не поддерживающей SOAP. Соответственно, при использовании данной привязки между узлами SOAP и другими реализациями HTTP может быть достигнута определенная степень взаимодействия. Например, стандартный веб-сервер (т.е. не написанный специально для соответствия данной спецификации) может использоваться для ответов на HTTP GET запросы с Content-Type, равным «application/soap+xml», инициированные SOAP. Такое взаимодействие не является нормативной функцией данной спецификации.

Даже при том, что HTTP часто используется на известном порту TCP 80, использование HTTP не ограничивается данным портом. Тем самым возможно наличие выделенного HTTP сервера для обработки данных SOAP на индивидуальном порту TCP. Альтернативно этому, возможно использование отдельного виртуального узла для обработки данных SOAP. Однако, такая конфигурация связана с вопросами удобства и не является требованием данной спецификации (см. [ИСО/МЭК 40210, 10.3.1]).

9.1.4 Тип медиа HTTP

Соответствующие данной привязке реализации:

1 ДОЛЖНЫ быть способны отправлять и получать сообщения, сериализированные с использованием типа медиа «application/soap+xml»; надлежащее использование и параметры данного типа описаны в приложении А;

2 МОГУТ отправлять запросы и ответы, используя другие типы медиа, если эти типы медиа предусматривают, как минимум, передачу XML SOAP инфо-набора;

3 МОГУТ при отправке запросов генерировать поле Accept заголовка HTTP. Это поле заголовка должно отвечать условиям:

ДОЛЖНО указывать на возможность принять как минимум «application/soap+xml»;

МОЖЕТ дополнительно указывать на готовность принять другие типы медиа, которые удовлетворяют требованиям, указанным выше в перечислении 2.

9.2 Имя привязки

Идентификатор данной привязки: URI (см. [ИСО/МЭК 40210, раздел 7]) «gost.e-ecolog.ru soap/bindings/HTTP/».

9.3 Поддерживаемые шаблоны обмена сообщениями

Реализация привязки SOAP к HTTP ДОЛЖНА поддерживать следующие шаблоны обмена сообщениями (ШОС):

«gost.e-ecolog.ru (см. 8.2);

«gost.e-ecolog.ru (см. 8.3).

9.4 Поддерживаемые функции

Реализация привязки SOAP к HTTP ДОЛЖНА поддерживать следующие дополнительные функции:

«gost.e-ecolog.ru (см. 8.4);

«gost.e-ecolog.ru (см. 8.5).

Множество допустимых значений свойства gost.e-ecolog.ru Method в данной привязке к HTTP зависит от используемого ШОС (он указывается в свойстве http://gost.e-ecolog.ru).

[1] В данном случае речь идет о HTTP 1.

Таблица 15 — Возможные значения свойства функции «Веб-метод»

gost.e-ecolog.ru ExchangePatternName	gost.e-ecolog.ru features/web-method/Method
«gost.e-ecolog.ru	«POST»
«gost.e-ecolog.ru	«GET»

Примечание — Другая привязка SOAP Версии 1.2 к HTTP может допускать другие комбинации http://gost.e-ecolog.ru и gost.e-ecolog.ru soap/features/web-method/Method.

9.5 Операции шаблонов обмена сообщениями

Для экземпляров привязок, соответствующих данной спецификации:

- узел SOAP, инстанцированный на клиенте HTTP, может иметь роль (т.е. свойство gost.e-ecolog.ru) «RequestingSOAPNode»;
- узел SOAP, инстанцированный в сервере HTTP, может иметь роль (т.е. свойство gost.e-ecolog.ru) «RespondingSOAPNode».

В следующих пунктах данного подраздела описывается конечный автомат ШОС и его отношение к протоколу HTTP. В таблицах переходов, представленных ниже, состояния определены как значения свойства gost.e-ecolog.ru (см. 8.2 и 8.3) и имеют тип xs:anyURI. Для краткости используются относительные URI, а базовый URI представляет собой gost.e-ecolog.ru .

Используемый шаблон обмена сообщениями обозначается в методе HTTP запроса. HTTP GET соответствует ШОС «ответ SOAP», HTTP POST соответствует ШОС «запрос-ответ».

9.5.1 Поведение запрашивающего узла SOAP

В целом, поведение запрашивающего узла SOAP подчиняется описанию конечного автомата, соответствующего шаблону «запрос-ответ» (см. 8.2) или шаблону «ответ SOAP» (см. 8.3) (различия обозначены по мере необходимости). Данная привязка поддерживает потоковую передачу, поэтому запрашивающие узлы SOAP ДОЛЖНЫ избегать взаимной блокировки, принимая, и при необходимости, обрабатывая информацию ответа SOAP во время передачи запроса SOAP (см. 8.2.3). Следующие подпункты подробно описывают каждое состояние.

9.5.1.1 Init (Инициализация)

В состоянии «Init» формируется запрос HTTP (согласно таблице 16) и инициируется передача запроса.

Таблица 16 — Поля запроса HTTP

Поле	Значение
Метод HTTP	Согласно свойству http://www.w3.org/2003/05/soap/features/web-method/Method. POST и GET — единственные значения, поддерживаемые данной привязкой
URI запроса	Значение URI, содержащееся в свойстве gost.e-ecolog.ru gost.e-ecolog.ru а обмена сообщениями
Поле заголовка Content-Type	Если в заголовке присутствует поле Content-Type, то оно содержит тип медиа объекта запроса; в противном случае, поле заголовка опускается (см. 9.1 для описания допустимых типов медиа). Если инфо-набор конверта SOAP в свойстве gost.e-ecolog.ru равен null, то поле заголовка Content-Type МОЖЕТ быть опущено
Параметр Action	В соответствии со значением свойства http://www.w3.org/2003/05/soap/features/action/Action
Поле заголовка Accept (опционально)	Список типов медиа, которые принимаются в ответе на сообщение запроса

Окончание таблицы 16

Поле	Значение
Дополнительные поля заголовка	Генерируется в соответствии с правилами для выражения, определенного привязкой, или любых дополнительных функций при использовании в обмене сообщениями данного сообщения. Например, поле заголовка Content-Encoding (см. HTTP [RFC 2616], подраздел 14.11) может использоваться для того, чтобы указать, что используется опциональная функция сжатия
Тело объекта HTTP	Сообщение SOAP, сериализированное в тип типа медиа, равного значению поля заголовка Content-Type согласно правилам передачи сообщений SOAP. Правила для формирования сообщения SOAP для типа медиа «application/soap+xml» представлены в приложении А. Если инфо-набор конверта SOAP в свойстве gost.e-ecolog.ru w3.org/2003/05/soap/mep/OutboundMessage равен null, тело объекта опускается

9.5.1.2 Requesting (Запрос)

В состоянии «Requesting» продолжается отправка запроса одновременно с ожиданием начала сообщения ответа. В таблице 17 описаны переходы, происходящие, когда запрашивающий узел SOAP получает строку состояния и поля заголовка ответа HTTP. Для некоторых кодов состояния следующее состояние неоднозначно. В случаях, где состояние «Отказ» является одним из вариантов следующего состояния, переход зависит от того, присутствует ли сообщение SOAP в ответе HTTP. Если сообщение SOAP присутствует, следующее состояние — «Sending + Receiving» или «Receiving», в противном случае следующее состояние — «Fail». Выбор состояния между «Sending + Receiving» и «Receiving» зависит от используемого шаблона обмена сообщениями: «Sending + Receiving» — следующее состояние для шаблона «запрос-ответ», в то время как «Receiving» — следующее состояние для шаблона «ответ SOAP».

Таблица 17 — Переходы, зависимые от статус-кода HTTP

Код состояния	Поясняющая фраза	Значение/Действие	Следующее состояние
2xx	Successful		
200	OK	Сообщение ответа передается в теле объекта ответа HTTP. Начать конструирование абстрактного сообщения ответа в http://gost.e-ecolog.ru InboundMessage	«Sending + Receiving» или «Receiving»
202	OK	Запрос был принят, но или (a) никакой конверт ответа не получен, или (b) получен конверт, представляющий информацию, связанную с запросом — такие конверты СЛЕДУЕТ обрабатывать, используя модель обработки SOAP ([ИСО/МЭК 40210, раздел 5])	«Receiving» (после которого сразу произойдет переход в состояние «Success»)
301, 302, 307	Redirect	Запрашиваемый ресурс был перемещен. В случае небезопасного метода HTTP, такого как POST или PUT, для продолжения операции требуется явное подтверждение. В случае безопасного метода, такого как GET, или если перенаправление было подтверждено, СЛЕДУЕТ повторить HTTP запрос, поместив в свойство gost.e-ecolog.ru значение URI из соответствующего поля заголовка Location	«Init» или «Fail»
303	See Other	Запрашиваемый ресурс был перемещен. СЛЕДУЕТ повторить HTTP запрос, поместив в свойство gost.e-ecolog.ru 2003/05/soap/mep/ImmediateDestination значение URI из связанного поля заголовка Location. Значение http://www.w3.org/2003/05/soap/features/web-method/Method изменено на «GET», значение http://gost.e-ecolog.ru mep/OutboundMessage установлено в «null»*	«Init»

Окончание таблицы 17

Код состояния	Поясняющая фраза	Значение/Действие	Следующее состояние
4xx	Client Error		
400	Bad Request	Указывает на проблему, связанную с полученным сообщением запроса HTTP	«Sending + Receiving», «Receiving» или «Fail»
401	Unauthorized	Указывает, что запрос HTTP требует авторизации. Обмен сообщениями расценивается как завершавшийся неудачно	«Requesting» или «Fail»
405	Method not allowed	Указывает, что сервер HTTP не поддерживает требуемый HTTP метод в данном URI запроса. Обмен сообщениями расценивается как завершавшийся неудачно	«Fail»
415	Unsupported Media Type	Указывает, что сервер HTTP не поддерживает Content-type, использованный при кодировании сообщения запроса. Обмен сообщениями расценивается как завершавшийся неудачно	«Fail»
5xx	Server Error		
500	Internal Server Error	Указывает на проблему сервера или проблему с полученным запросом	«Sending + Receiving», «Receiving» или «Fail»
* Код состояния 303 ДОЛЖЕН быть отправлен только если конверт SOAP запроса обработан согласно модели обработки SOAP и ответ SOAP доступен по URI, присланного вместе с кодом статуса 303.			

Таблица 17 ссылается на некоторые, но не на все существующие коды состояния HTTP/1.1 [RFC 2616]. В дополнение к этим кодам состояния HTTP обеспечивает открытый механизм для поддержки кодов состояния, определенных расширениями HTTP (см. [RFC 2817], где описан механизм регистрации новых кодов состояния). Коды состояния HTTP разделены на классы кодов состояния в соответствии со спецификацией HTTP [RFC 2616, пункт 6.1.1]. Привязка SOAP к HTTP следует правилам для приложений HTTP: реализация привязки SOAP к HTTP должна понимать класс любого кода состояния, обозначенного первой цифрой, и обрабатывать любой нераспознанный ответ, как являющийся эквивалентным ответу с кодом состояния x00-го класса, с одним исключением — нераспознанный ответ не должен кэшироваться.

Примечание — В инфраструктуре HTTP могут присутствовать элементы, сконфигурированные для изменения сообщения ответа HTTP для кодов состояния 4xx и 5xx. Например, некоторые исходные HTTP сервера имеют такую опцию в качестве конфигурационного параметра. Это поведение может конфликтовать с поведением, использующим ответные коды состояния 4xx и 5xx, передающие сообщения отказа SOAP в HTTP, и рекомендуется, чтобы такое поведение было отключено для ресурсов, принимающих запросы SOAP/HTTP. Если поведение, изменяющее сообщения ответов для кодов состояния 4xx и 5xx не может быть отключено, то SOAP/HTTP не может использоваться в таких конфигурациях.

9.5.1.3 Sending + Receiving (Отправка и получение)

В состоянии «Sending + Receiving» (только ШОС «запрос-ответ», см. 8.2) передача сообщения запроса и получение сообщения ответа завершены. Только в случае, если получен код состояния 200, предполагается, что полученный ответ содержит конверт SOAP, сериализированный в тип медиа, указанный в поле заголовка Content-type, согласно правилам передачи сообщения SOAP.

Ответ МОЖЕТ иметь тип Content-Type не только «application/soap+xml». Такое использование считается ненормативным, и, соответственно, оно не моделируется в конечном автомате. Интерпретация таких ответов остается на усмотрение получателя. Точно так же получение любого тела объекта ответа с кодом состояния 202 ненормативно. Если такой неожиданный ответ имеет тип «application/soap+xml», то обработка SOAP такого ответа выходит за рамки данной спецификации привязки.

9.5.1.4 Receiving (Получение)

В состоянии «Receiving» (только ШОС «ответ SOAP», см. 8.3) получение сообщения ответа завершается. Только в случае кода состояния 200 предполагается, что сообщение ответа содержит конверт SOAP, сериализированный в тип медиа, указанный в поле заголовка Content-Type, согласно правилам передачи сообщения SOAP.

Ответ МОЖЕТ иметь тип контента (Content-Type), отличный от «application/soap+xml». Такой результат особенно вероятен, если запрос SOAP, отправленный с помощью метода «GET» со свойством gost.e-ecolog.ru, направлен (преднамеренно или иным образом) к серверу HTTP, не являющемуся в том числе и сервером SOAP. Такое использование считается ненормативным и, соответственно, не моделируется в конечном автомате. Интерпретация таких ответов остается на усмотрение получателя. Точно так же получение любого объекта-тела ответа с кодом состояния 202 ненормативно. Если такой неожиданный ответ имеет тип «application/soap+xml», то обработка SOAP такого ответа выходит за рамки спецификации данной привязки.

9.5.1.5 Success (Успех) и Fail (Отказ)

«Success» и «Fail» — терминальные состояния шаблонов «запрос-ответ» и «ответ SOAP». Управление контекстом обмена сообщениями возвращается к локальному узлу SOAP.

Если состояние «Success» было достигнуто, и конверт SOAP был получен, то локальный узел является *получателем SOAP* (см. [ИСО/МЭК 40210, пункт 4.4.3]) и ДОЛЖЕН выполнять требования раздела «Узлы SOAP» [ИСО/МЭК 40210, подраздел 5.1] для обработки сообщения согласно модели обработки SOAP [ИСО/МЭК 40210, раздел 5].

9.5.2 Поведение отвечающего узла SOAP

В целом, поведение отвечающего узла SOAP следует описанию конечного автомата, соответствующего шаблону «запрос-ответ» (см. 8.2) или шаблону «ответ SOAP» (см. 8.3) (различия обозначены по мере необходимости). Следующие подпункты подробно описывают каждое состояние.

9.5.2.1 Init (Инициализация)

В состоянии «Init» привязка ожидает поступление входящего сообщения запроса. В таблице 18 представлены ошибки, которые может генерировать отвечающий узел SOAP, находящийся в состоянии «Init». В этом состоянии никакие сообщения SOAP не принимаются, поэтому узел SOAP не может генерировать отказ SOAP.

Таблица 18 — Ошибки, возможные в состоянии Инициализация

Проблема с сообщением	Код состояния HTTP	Поясняющая фраза HTTP (информативная)
Плохо сформированное сообщение запроса	400	Bad request
Некорректный HTTP метод: не POST и не GET	405	Method not allowed
Неподдерживаемый метод инкапсуляции сообщения	415	Unsupported media

9.5.2.2 Receiving (Получение)

В состоянии «Receiving» привязка получает запрос и любое связанное с ним сообщение и ожидает готовности сообщения ответа. В таблице 19 представлены поля заголовка ответа HTTP, сгенерированные отвечающим узлом SOAP. В таблице 20 представлены коды состояния HTTP, связанные с отказами SOAP, которые могут быть сгенерированы отвечающим узлом SOAP.

Таблица 19 — Поля заголовков ответа HTTP

Поле	Значение
Строка состояния	Если конверт ответа SOAP доступен в gost.e-ecolog.ru 05/soap/mep/OutboundMessage, то устанавливается значение 200 или, если был сгенерирован отказ SOAP, значение устанавливается согласно таблице 20. В противном случае, если никакой конверт SOAP не предоставлен, то устанавливается значение 202
Поле заголовка Content-Type	Если код состояния 200, то устанавливается значение типа медиа тела ответа, см. описание допустимых типов медиа в 9.1. Если код состояния не 200, то заголовок Content-Type не отправляется
Дополнительные поля заголовка	Генерируются в соответствии с установленным привязкой выражением для любых дополнительных функций, использованных в данном обмене сообщениями. Например, поле заголовка Content-Encoding (см. HTTP [RFC 2616], подраздел 14.11) может использоваться для того, чтобы указать, что используется опциональная функции сжатия

Окончание таблицы 19

Поле	Значение
Тело объекта HTTP	Только в случае кода состояния 200 сообщение SOAP сериализируется в тип медиа, заданном в поле заголовка Content-Type, согласно правилам для отправки сообщений SOAP. Правила для передачи сообщения SOAP в типе медиа «application/soap+xml» приведены в приложении А

Таблица 20 — Отображение отказов SOAP в коды состояния HTTP

Отказ SOAP	Код состояния HTTP	Поясняющая фраза HTTP (информативная)
env:VersionMismatch	500	Internal server error
env:MustUnderstand	500	Internal server error
env:Sender	400	Bad request
env:Receiver	500	Internal server error
env:DataEncodingUnknown	500	Internal server error

9.5.2.3 Receiving + Sending (Получение и отправка)

В состоянии «Receiving + Sending» (только ШОС «запрос-ответ», см. 8.2) привязка завершает получение сообщения запроса и передачу сообщения ответа.

9.5.2.4 Отправка

В состоянии «Отправка» (см. только 8.3) привязка завершает передачу сообщения ответа.

9.5.2.5 Success (Успех) и Fail (Отказ)

«Success» и «Fail» — терминальные состояния для шаблонов «запрос-ответ» и «ответ SOAP». С точки зрения локального узла данный обмен сообщениями завершен.

9.6 Соображения безопасности

Привязка SOAP к HTTP (см. раздел 9) может рассматриваться как расширение прикладного протокола HTTP. Поэтому все соображения безопасности, идентифицированные и описанные в спецификации HTTP [RFC 2616, раздел 15], применимы к привязке SOAP к HTTP в дополнение к описанным в разделе 10 «Соображения Безопасности» спецификации SOAP 1.2. Часть 1 [ИСО/МЭК 40210, раздел 10]. Разработчики привязок SOAP к HTTP должны тщательно анализировать данный материал.

Приложение А
(справочное)

Тип медиа «application/soap+xml»

Исходное содержимое данного раздела неактуально (см. RFC3902 [RFC 3902]).

Приложение В
(справочное)

Отображение имен, определенных приложениями, в имена XML

Это приложение детализирует алгоритм отображения определенных приложениями имен, таких как имена переменных или полей в языках программирования, в символы Unicode для того, чтобы имена были корректными именами элементов и атрибутов XML в соответствии со спецификацией пространств имен в XML [Namespaces in XML].

Шестнадцатеричные цифры

[5]hexDigit ::= [0-9A-F]

В.1 Правила для отображения имен, определенных приложениями, в имена XML

1 XML имя состоит из двух частей: *Prefix* и *LocalPart*. Пусть *Prefix* определяется в соответствии с правилами и ограничениями, определенными в [Namespaces in XML].

2 Пусть *T* — это имя в приложении, представленное в виде последовательности символов, закодированных в определенной кодировке.

3 Пусть M — определенная реализацией функция для перекодирования символов, используемых в определенных приложениями именах, в эквивалентные строки символов Unicode.

Примечания

1 В идеальном варианте, если это перекодирование из не-Unicode кодировки, то оно должно быть и обратимым, и нормализующим по форме C Unicode (т.е. комбинирование последовательности представлены в предписанном каноническом порядке). Следует заметить, что некоторые перекодирования не могут быть совершенно обратимыми, и в некоторых случаях нормализация по форме C (NFC) может изменить исходную последовательность (см. [CharMod]). Для того, чтобы обеспечить соответствие имен после перекодировки, последовательности Unicode должны быть нормализованы с формы нормализации C Unicode.

2 Перекодирование преобразует в скалярные значения Unicode («кодовые позиции»), а не в какую-либо определенную схему кодирования символов Unicode, такую как UTF-8 или UTF-16.

3 Корректные последовательности суррогатных пар должны быть преобразованы в соответствующие им скалярные значения («кодовые позиции») (т.е. последовательность U+D800 U+DC00 должна быть перекодирована в символ U+10000). Если перекодирование определено для кодировки Unicode, то несоответствующие (не являющиеся самыми короткими формами) последовательности UTF -8 и UTF-16 должны быть преобразованы в соответствующие им скалярные значения.

4 Количество символов в T не обязательно совпадает с количеством символов в M, потому что перекодирование может быть и таким, что один символ кодируется несколькими, и наоборот, несколько символов кодируются одним. Детали перекодирования могут определяться реализацией. В редких случаях нет никакого эквивалентного представления Unicode для T. Такие случаи здесь не описаны.

1 Пусть *C* — последовательность скалярных значений Unicode (символов), составляющих M(T).

2 Пусть *N* — число символов в *C*. Пусть C_1, C_2 ... C_N — символы *C*, представленные в логическом порядке от наиболее значащего к наименее значащему.

3 Для каждого *i* между 1 (единицей) и *N*, пусть X_i — строка из символов Unicode, определенная по следующим правилам.

Варианты:

1) Если C_i не определен (т.е. некоторый символ или последовательность символов из символьной последовательности приложения *T* не содержат отображения в Unicode), то X_i определяется реализацией.

2) Если *i* <= *N*-1 и C_i = «_» (U+005F, НИЖНЕЕ ПОДЧЕРКИВАНИЕ), и C_{i+1} = «x» (U+0078, ЛАТИНСКАЯ СТРОЧНАЯ БУКВА X), то можно полагать, что X_i, станет равным «_x005F_».

3) Если *i* =1, *N* >= 3 и C_1 = «x» (U+0078, ЛАТИНСКАЯ СТРОЧНАЯ БУКВА X) или «X» (U+0058, ЛАТИНСКАЯ ПРОПИСНАЯ БУКВА X), и C_2 = «m» (U+006D, ЛАТИНСКАЯ СТРОЧНАЯ БУКВА M) или «M» (U+004D, ЛАТИНСКАЯ ПРОПИСНАЯ БУКВА M), и C_3 = «l» (U+006C, ЛАТИНСКАЯ СТРОЧНАЯ БУКВА L) или «L» (U+004C, ЛАТИНСКАЯ ПРОПИСНАЯ БУКВА L) (другими словами, в строке из трех или более символов, начинающейся с «xml», каждая буква может быть строчной или прописной). Если C_1 — это «x» (U+0078, ЛАТИНСКАЯ СТРОЧНАЯ БУКВА X), то

можно полагать, что X_1 станет равным «_x0078_». В противном случае, если C_1 — это «X» (U+0058, ЛАТИНСКАЯ ПРОПИСНАЯ БУКВА X), то можно полагать, что X_1 станет равным «_x0058_».

4) Если C_i не является допустимым символом строк типа NCName (см. [Namespaces in XML]) или, если $i=1$ (один) и C_1 не является допустимым первым символом строки типа NCName тогда:

Пусть U_1, U_2, ..., U_6 — шесть шестнадцатеричных цифр (см. грамматическое правило [5] выше) таких, что скалярное значение Unicode C_i есть «U+» $U_1 U_2 ... U_6$.

Варианты:

Если U_1=0, U_2=0, U_3=0, и U_4=0, тогда можно положим X_i = «_x» $U_5 U_6$ «_».

Этот случай подразумевает, что C_i — символ в базовой многоязыковой плоскости (Basic Multilingual Plane, BMP, Плоскость 0) Unicode и он может быть полностью представлен одной «кодовой позицией» UTF-16 — последовательностью $U+U_5U_6$.

В противном случае положим X_i, равным «_x» $U_1 U_2 U_3 U_4 U_5 U_6$ «_».

В противном случае положим X_i, равным M_i,т.е. любой символ из X, который является разрешенным символом для строки типа NCName, просто копируется.

1 Положим *LocalPart*, равной конкатенации строк X_1, X_2, ... , X_N в порядке от наиболее значимой к наименее значимой.

2 Положим XML имя, равным QName, согласно спецификации [Namespaces in XML].

В.2 Примеры

Hello world -> Hello_x0020_world
Hello_xorld -> Hello_x005F_xorld
Helloworld_ -> Helloworld_

x -> x
xml -> _x0078_ml
-xml -> _x002D_xml
x-ml -> x-ml

Ælfred -> Ælfred
άγνωστος -> άγνωστος

Hello world -> Hello_x0020_world
Hello_xorld -> Hello_x005F_xorld
Helloworld_ -> Helloworld_

x -> x
xml -> _x0078_ml
-xml -> _x002D_xml
x-ml -> x-ml

Ælfred -> Ælfred
άγνωστος -> άγνωστος

17 09	17 05	17 0E	17 08

-> _x1709__x1705__x170E__x1708_

13 D9	13 DA	13 A5

-> _x13D9__x13DA__x13A5_

Приложение С
(справочное)

Использование W3C XML Schema с кодировкой SOAP

Как отмечалось в 5.1.4, узлы графа SOAP маркированы именами типов, но соответствующие процессоры для проверки корректности закодированных сообщений SOAP не требуются.

Данные разделы описывают методы, которые могут применяться, когда для использования приложениями SOAP желательна проверка корректности по отношению к W3C XML-схемам. Любые ошибки или отказы, выявленные во время такой проверки корректности, выходят за рамки данной Рекомендации W3C; с точки зрения SOAP такие отказы рассматриваются как отказы уровня приложения.

С.1 Проверка корректности с использованием минимальной схемы

Несмотря на то что XML-схемы W3C обычно передаются как документы на языке XML Schema [XML Schema Part 1], схема Рекомендации основана на абстрактном определении схем, которым должны соответствовать все процессоры. Схема Рекомендации обеспечивает, что все такие схемы включают определения для базового набора встроенных типов, таких как целые числа, даты, и т.д. (см. [XML Schema Part 1, подраздел 3.14]). Таким образом, возможна проверка корректности сообщения SOAP на соответствие такой минимальной схеме, которая получается при отсутствии дополнительных определений и объявлений (т.е. не является схемой документа) для процессора схемы.

При использовании минимальной схемы хорошо сформированный документ XML считается корректным, если для всех элементов, кроме элементов с xsi:type, именованный тип является встроенным, и соответствующий элемент является корректным по отношению к этому типу. Таким образом, проверка корректности сообщения SOAP 1.2 с использованием минимальной схемы приближает поведение встроенных типов SOAP 1.1.

С.2 Проверка корректности с использованием схемы кодирования SOAP

Проверка корректности в отношении минимальной схемы (см. раздел С.1) не будет успешно выполняться в случаях, где у закодированных узлов графа есть многократные входящие ребра. Это вызвано тем, что элементы, представляющие такие узлы графа, будут содержать *информационные элементы-атрибуты* id, которые недопустимы для элементов типа «xs:string», «xs:integer» и т.д. Кодирование SOAP таких графов МОЖЕТ быть проверено на корректность по отношению к схеме кодирования SOAP[1)]. Для проверки кодирования метки ребер и, следовательно, свойства [local name] и [namespace name] *информационных объектов-элементов* должны соответствовать меткам и свойствам, определенным в схеме кодирования SOAP. Корректность закодированного графа по отношению к схеме кодирования SOAP приведет к тому, что свойству «имя типа» узла в графе будет присвоено соответствующее имя типа.

С.3 Проверка корректности с использованием более конкретных схем

Иногда схемы могут быть созданы для описания кодирования определенных графов. Корректность закодированного графа по отношению к такой схеме приведет к тому, что свойству «имя типа» узла в графе будет присвоено соответствующее имя типа. Такая схема может также предоставлять значения по умолчанию или фиксированные значения для одного или более *информационных элементов-атрибутов* itemType, arraySize или nodeType; значения по умолчанию атрибутов при десериализации графа обрабатываются таким же образом, как если были явно переданы в сообщении. Об ошибках или несогласованностях, представленных таким образом (например, если значение атрибута ошибочное или несоответствующее), нужно сообщать как об ошибках уровня приложения; об отказах из пространства имен «http://gost.e-ecolog.ru следует сообщать только в случае, если нарушены нормативные части данной спецификации.

1) gost.e-ecolog.ru

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 40210:2011	IDT	ГОСТ Р ИСО/МЭК 40210 – 2014 «Информационные технологии. W3C SOAP — Версия 1.2. Часть 1. Основы обмена сообщениями (Вторая редакция)»
ИСО/МЭК 40230:2011	IDT	ГОСТ Р ИСО/МЭК 40230 – 2014 «Информационные технологии. Механизм оптимизации передачи сообщения W3C SOAP»
Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов: - IDT — идентичные стандарты.		

Библиография

[1] Простой протокол доступа к объектам [Simple Object Access Protocol (SOAP) 1.1]

[2] SOAP Version 1.2 Part 0: Primer (Second Edition) [SOAP Версия 1.2 Часть 0] Учебник для начинающих (вторая редакция)

[3] XML Protocol Comments Archive [Замечания по XMLP] Архив замечаний по протоколу XML (см. gost.e-ecolog.ru w3.org/Archives/Public/xmlp-comments/)

[4] XML Protocol Discussion Archive [Dist-приложения XMLP] Архив обсуждения протокола XML (см. gost.e-ecolog.ru w3.org/Archives/Public/xml-dist-app/)

[5] XML Protocol Charter [Чартер XMLP] Чартер протокола XML (см. gost.e-ecolog.ru 07/XML-Protocol-Charter)

[6] Multipurpose Internet Mail Extensions (MIME) Part One. Многоцелевое расширение электронной почты (MIME), часть 1 — Формат тела сообщения (см. gost.e-ecolog.ru)

[7] The Internet Standards Process — Revision 3. Процесс разработки стандартов Интернета — Пересмотр 3 (см. gost.e-ecolog.ru)

[8] Upgrading to TLS Within HTTP/1.1. Обновление до TLS в HTTP/1.1 (см. gost.e-ecolog.ru)

[9] Internationalized Resource Identifiers (IRIs), Интернационализировавшие идентификаторы ресурса (IRI) (см. gost.e-ecolog.ru)

[10] Character Model for the World Wide Web 1.0. Модель символов для Интернета 1.0. Основные принципы (см gost.e-ecolog.ru)

УДК 004.057.4:006.354 ОКС 35.100.05

Ключевые слова: информационные технологии, протоколы SOAP, модель данных SOAP, кодирование данных SOAP, шаблоны обмена сообщениями (ШОС), функции SOAP

Редактор *Н.В. Таланова*
Технический редактор *В.Н. Прусакова*
Корректор *В.И. Варенцова*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 11.08.2015. Подписано в печать 15.09.2015. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 4,00. Тираж 34 экз. Зак. 2978.

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ» 123995 Москва, Гранатный пер., 4
www.gostinfo.ru gost.e-ecolog.ru

gost.e-ecolog.ru