ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 40210— 2014

Информационные технологии

W3C SOAP — Версия 1.2

Часть 1

Основы обмена сообщениями (Вторая редакция)

ISO/IEC 40210:2011
Information technology — W3C SOAP Version 1.2
Part 1: Messaging Framework (Second Edition)
(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО «ИАВЦ») и Закрытым акционерным обществом «ИТ Эксперт» (ЗАО «ИТ Эксперт») на основе собственного аутентичного перевода на русский язык международного 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 11 июня 2014 г. № 558-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 40210:2011 «Информационные технологии. W3C SOAP — Версия 1.2. Часть 1. Основы обмена сообщениями (Вторая редакция)» [ISO/IEC 40210:2011 «Information technology — W3C SOAP Version 1.2 Part 1. Messaging Framework (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 годовом (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячно издаваемом 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.ru)

© Стандартинформ, 2014

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 ... 2
3 Условные обозначения ... 2
4 Соответствие ... 2
4.1 Общие положения ... 2
4.2 Связь с другими спецификациями ... 3
4.3 Пример сообщения SOAP ... 4
4.4 Терминология SOAP ... 4
5 Модель обработки SOAP ... 5
5.1 Узлы SOAP ... 6
5.2 Роли SOAP и узлы SOAP ... 6
5.3 Предназначение блоков заголовка SOAP ... 7
5.4 Понимание блоков заголовка SOAP ... 7
5.5 Структура и интерпретация тела SOAP ... 7
5.6 Обработка сообщений SOAP ... 7
5.7 Пересылка сообщений SOAP ... 9
5.8 Модель управления версиями SOAP ... 11
6 Модель расширяемости SOAP ... 12
6.1 Функции расширения SOAP ... 12
6.2 Шаблоны обмена сообщениями SOAP (MEPs) ... 13
6.3 Модули SOAP ... 13
7 Структура привязки SOAP к протоколам ... 14
7.1 Назначение структуры привязки ... 14
7.2 Структура привязки ... 14
8 Логическая структура сообщения SOAP ... 15
8.1 Конверт SOAP ... 16
8.2 Заголовок SOAP ... 17
8.3 Тело SOAP ... 18
8.4 Отказ SOAP ... 19
9 Использование URL в SOAP ... 25
10 Соображения безопасности ... 26
10.1 Узлы SOAP ... 26
10.2 Посредники SOAP ... 26
10.3 Привязка нижележащего протокола ... 27
Приложение А (справочное) Переход от версии SOAP/1.1 к SOAP версии 1.2 ... 28
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации ... 29
Библиография ... 30

Введение

SOAP версия 1.2 — это упрощенный протокол, предназначенный для обмена структурированной информацией в децентрализованной, распределенной среде. «Часть 1. Основы обмена сообщениями», используя технологии XML, определяет расширяемые основы обмена сообщениями, определяющие логическую структуру сообщений, которыми можно обмениваться с использованием различных нижележащих протоколов.

Настоящий документ является Рекомендацией W3C. Он был разработан группой XML_Protocol Working Group, которая является частью Web Services Activity. Эта вторая обновленная редакция заменяет исходную версию рекомендации, и в нее включены исправления всех обнаруженных опечаток. Различия между этими двумя версиями описаны в протоколе различий.

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

В случае обнаружения ошибок в этом документе, пожалуйста, отправьте их по адресу публичного списка рассылки: gost.e-ecolog.ru (архив). Письма дискуссионного характера по этому адресу нежелательны.

Версия 1.2 SOAP заменяет все предыдущие версии SOAP, включая версию SOAP 1.1 [SOAP 1.1].

Отчеты о реализациях SOAP 1.2 представлены в gost.e-ecolog.ru.

Этим документ соответствует Текущей патентной практике CPP от 24 января 2002 с учетом исправлений процедуры перехода патентной политики W3C. W3C поддерживает общедоступный список любых патентных сведений, составленный в соответствии с публикациями группы. Кроме того, на этой странице имеются инструкции, как раскрыть патент. Если кто-либо обладает действительным знанием патенте, который удовлетворяет существенным требованиям, то он должен раскрыть эту информацию в соответствии с разделом 6 патентной политики W3C.

Список текущих рекомендаций W3C и другие технические отчеты можно найти по адресу: gost.e-ecolog.ru.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

W3C SOAP — Версия 1.2.

Часть 1

Основы обмена сообщениями (Вторая редакция)

Information technologies. W3C SOAP version 1.2.
Part 1. Messaging framework (second edition)

Дата введения — 2015—06—01

1 Область применения

SOAP версии 1.2 (SOAP) является упрощенным протоколом, предназначенным для обмена структурированной информацией в децентрализованной, распределенной среде. При этом, для определения расширяемой структуры обмена сообщениями, обеспечивающих логическую структуру сообщения, которыми можно обмениваться посредством различных нижележащих протоколов, используются технологии XML. Структура разрабатывалась с учетом требования независимости от любой конкретной модели программирования и любой реализационно-зависимой семантики.

Две главные цели проекта для SOAP — это простота и расширяемость (см. требования XMLP [XMLP Requirements]). При разработке SOAP была сделана попытка удовлетворить этим целям, исключив из структуры обмена сообщениями функции, которые часто обеспечиваются непосредственно распределенными системами. В список таких функций входят «надежность», «безопасность», «взаимозависимость», «маршрутизация» и «шаблоны обмена сообщениями» (MEPs), однако, этот список не ограничен только этими функциями. Несмотря на то, что казалось бы, что многие функции будут определены, данный документ ограничен спецификацией только двух шаблонов MEPs. Остальные функции оставлены для определения другими спецификациями в качестве расширений.

Спецификация SOAP версии 1.2 состоит из трех частей. Часть 1 спецификации SOAP версии 1.2 (настоящий документ) определяет основы обмена сообщениями SOAP и включает в себя:

1) модель обработки SOAP, определяющую правила обработки сообщений SOAP (см. раздел 5);
2) модель расширяемости SOAP, определяющую понятия функций SOAP и модулей SOAP (см. раздел 6);
3) базовую структуру привязки протокола SOAP, описывающую правила задания привязки к нижележащему протоколу, который может использоваться для обмена сообщениями SOAP между узлами SOAP (см. раздел 7);
4) логическую структуру сообщения SOAP, определяющую структуру сообщения SOAP (см. раздел 8).

Учебник для начинающих по SOAP 1.2 [Part 0: SOAP] является ненормативным документом, предназначенным для использования в качестве доступного учебного пособия по функциям спецификации SOAP версии 1.2.

Часть 2. SOAP 1.2 [Part 2: SOAP] описывает ряд дополнений, которые могут использоваться вместе со структурой обмена сообщениями SOAP.

Примечание — В предыдущих версиях данной спецификации название SOAP было аббревиатурой. Теперь это не так.

Издание официальное

2 Нормативные ссылки

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

ИСО/МЭК 40220:2011 Информационные технологии. Версия 1.2 W3C SOAP. Часть 2. Дополнения (второе издание) (ISO/IEC 40220:2011, Information technology-SOAP Version 1.2 Part 2: Adjuncts (Second Edition)

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

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

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

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

Универсальный идентификатор ресурса (URI): Основы синтаксиса (Uniform Resource Identifiers (URI): Generic Syntax)

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

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

Информационный набор XML (вторая редакция) (XML Information Set (Second Edition)

Основы XML (XML Base)

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 RFC2119 [RFC 2119].

Префиксы пространств имен, использующиеся в данной спецификации, перечислены в таблице 1. Необходимо отметить, что выбор любого префикса пространства имен произволен и не является семантически существенным (см. Инфо-набор XML [XML InfoSet]).

Таблица 1 – Префиксы и пространства имен, используемые в данной спецификации.

Префикс	Пространство имен	Примечания
ENV	http://www.w3.org/2003/05/soap-envelope	Описание нормативной схемы XML [Часть 1 XML-схемы], [Часть 2 XML-схемы]XML для пространства имен gost.e-ecolog.ru может быть найдено по адресу: gost.e-ecolog.ru
xs	http://www.w3.org/2001/XMLSchema	Спецификация пространства имен XML-схемы [Часть 1 XML-схемы], [Часть 2 XML-схемы]

Имена из пространства имен общей формы gost.e-ecolog.ru gost.e-ecolog.ru представляют собой универсальные идентификаторы ресурсов (URI) приложений или контекстно-зависимые URI (см. RFC 3986 [RFC 3986]).

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

4 Соответствие

4.1 Общие положения

Настоящая спецификация описывает форматы данных и правила для создания, обмена и обработки сообщений с использованием этих форматов. Эта спецификация не налагает требования на область применения какой-либо определенной реализации, хотя и требует, чтобы никакая реализация не нарушала обязательных требований.

Для соответствия спецификации SOAP версии 1.2 конкретная реализация ДОЛЖНА корректно реализовать все обязательные (MUST) требования, сформулированные в части 1 спецификации SOAP версии 1.2 (настоящий документ) и которые необходимы для выполнения операций. Необходимо отметить, что от реализации не требуется выполнение всех обязательных требований. Например, специфическая реализация, которая никогда не отправляет блок заголовка SOAP, может претендовать на соответствие при условии, что правильно выполняются обязательные требования для сообщений, которые она фактически отправляет.

В реализации МОЖЕТ быть использовано любое число дополнений, определенных в Части 2 SOAP версии 1.2 [Часть 2 SOAP]. Нужно заметить, что соответствие никак не связано с соглашением для описания функций и привязки (см. разделы 6 и 7). Для соответствия дополнениям реализации дополнения ДОЛЖНЫ выполнять все необходимые обязательные требования, определенные в спецификации дополнения.

SOAP версии 1.2 может использоваться в качестве основы для других технологий, которые предоставляют более широкие или более специализированные услуги. Для соответствия спецификации SOAP версии 1.2 спецификации и реализации таких технологий не должны противоречить необходимым обязательным требованиям, изложенным в части 1 спецификации SOAP версии 1.2 (настоящий документ). Правила соответствия для подобных новых спецификаций выходят за рамки спецификации SOAP версии 1.2. Рекомендуется, чтобы спецификации для таких технологий обеспечивали выполнение надлежащих правил соответствия.

Версия 1.2 SOAP разработана для того, чтобы обеспечить выполнение, по крайней мере, сценариев использования, описанных в «Сценарии использования SOAP 1.2» [Сценарии использования SOAP], и, возможно, других сценариев. Неформальные описания, демонстрирующие XML представление конкретных сообщений SOAP, используемых в некоторых общих сценариях, приводятся в Части 0 SOAP 1.2 [Часть 0 SOAP].

4.2 Связь с другими спецификациями

Сообщение SOAP определено в виде информационного набора (инфо-набора) XML [XML InfoSet]. Хотя в настоящем документе все примеры сообщений SOAP приводятся с использованием синтаксиса XML 1.0 [XML 1.0], для передачи сообщения SOAP между узлами МОГУТ использоваться и другие представления (см. раздел 7).

Некоторые информационные объекты, определенные в этом документе (см. раздел 8), идентифицированы с использованием соответствующих требований имен пространства имен [Namespaces in XML]. Список имен пространств имен, определенных в этом документе, приводится в Таблице 1.

Примечание — Настоящая спецификация использует термин «Расширенное имя XML», подразумевая под этим для значения типа xsd:Qname, заданных как пара из пространства значений [абсолютная ссылка на универсальный идентификатор объекта uri, локальное имя]. Такая терминология обсуждается для включения ее в будущие версии пространства имен в XML [Namespaces in XML]. Если в будущих версиях пространства имен в XML [Namespaces in XML] будет принята другая терминология, то предполагается, что соответствующие изменения будут внесены в эту рекомендацию либо в виде исправления опечаток, либо будут внесены в будущем пересмотре версии.

SOAP не требует обработки XML-схемы (оценка или проверка допустимости) для установления правильности или соответствия схеме значений информационных объектов элементов и атрибутов, определенных разделами 4 и 5 данной спецификации. Значения, связанные с информационными объектами элементов и атрибутов, определенными в этой спецификации, ДОЛЖНЫ явно задаваться в отправленном сообщении SOAP за исключением лишь тех случаев, когда заявлено обратное (см. раздел 8).

Типы информационных объектов-атрибутов SOAP описываются XML-схемой [Часть 2 XML-схемы]. Если не указано иначе, то для каждого такого атрибута поддерживаются все лексические формы. Кроме того, лексические формы, представляющие одно и то же значение в пространстве значений XML-схемы, считаются эквивалентными в смысле обработки SOAP, например, булевы лексические формы «1» и «true» взаимозаменяемы. Для краткости текст в настоящей спецификации дается только в одной лексической форме для каждого значения, например, «если значение информационного объекта-атрибута mustUnderstand равно 'true'».

Спецификации обработки данных приложений, которые передаются в сообщении SOAP, но не определены данной спецификацией, МОГУТ требовать дополнительной проверки допустимости сообщения SOAP в сочетании с обработкой уровня приложения. В таких случаях выбор языка схемы и/или технологии проверки допустимости определяется приложением.

SOAP использует спецификацию XML Base [XML Base] при определении базовых URI для относительных ссылок URI, используемых в качестве значений в информационных объектах, определенных данной спецификацией (см. раздел 9).

Для сериализации инфо-набора XML 1.0 сообщения SOAP ДОЛЖЕН использоваться тип медиа «application/soap+xml» (см. SOAP 1.2. Часть 2 [Часть 2 SOAP], Тип медиа «application/soap+xml»).

4.2.1 Требования к обработке

Возможность использовать SOAP в определенной среде зависит от фактических ограничений, выбора инструментов, модели обработки или природы сообщений, которыми обмениваются. SOAP разрабатывался таким образом, чтобы иметь сравнительно небольшое количество зависимостей от других спецификаций XML, ни одна из которых не предъявляет запрещающих требований к обработке. Более того, ограничения использования SOAP для маленьких сообщений вместо поддержки сообщений любого размера и поддержки только нескольких определенных типов сообщения вместо реализации обобщенной обработки могли бы значительно снизить требования к обработке.

4.3 Пример сообщения SOAP

Следующий пример демонстрирует сообщение-уведомление, выраженное посредством SOAP. Сообщение содержит две части данных, определенных приложением и не определенных данной спецификацией: блок заголовка SOAP с локальным именем alertcontrol и элемент тела с локальным именем alert. Как правило, блоки заголовка SOAP содержат информацию, которая могла бы быть полезной для посредников SOAP, а также для адресата сообщения. В этом примере посредник (intermediary) мог бы установить приоритет доставки сообщения на основе приоритета и информации о сроке действия в блоке заголовка SOAP. Тело содержит фактическую полезную информацию сообщения, в данном случае предупреждение.

Пример — Сообщение SOAP, содержащее блок заголовка SOAP и тело SOAP

```
<env:Envelope xmlns:env="gost.e-ecolog.ru                    ">
  <env:Header>
    <n:alertcontrol xmlns:n="gost.e-ecolog.ru                    ">
    <n:priority> 1 </n:priority>
    <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="gost.e-ecolog.ru                    ">
    <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

4.4 Терминология SOAP

В этом разделе описываются условия и понятия, представленные в Части 1 спецификации SOAP версии 1.2 (настоящий документ).

4.4.1 Понятия протокола

SOAP: Формальный набор соглашений, управляющих форматом и правилами обработки сообщения SOAP. Эти соглашения включают взаимодействия между узлами SOAP, генерирующими и принимающими сообщения SOAP в целях передачи информации на пути следования сообщения SOAP.

Узел SOAP: Реализация логики обработки, необходимой для посылки, получения, обработки и/или передачи сообщения SOAP в соответствии с набором соглашений, определенных этой рекомендацией. Узел SOAP ответственен за выполнение правил, которые управляют обменом сообщениями SOAP (см. раздел 5). Узел получает доступ к услугам, предоставляемым нижележащими протоколами посредством одной или более привязки SOAP.

Роль SOAP: Ожидаемая функция получателя SOAP в обработке сообщения. Получатель SOAP может действовать более чем в одной роли.

Привязка SOAP: Формальный ряд правил для того, чтобы передать сообщение SOAP внутри или поверх другого протокола (нижележащего протокола) в целях обмена сообщениями (см. раздел 7). Примеры привязки SOAP включают передачу сообщения SOAP в теле объекта HTTP, или по потоку TCP.

Дополнительная функция SOAP: Расширение структуры обмена сообщениями SOAP (см. раздел 6). Примеры дополнительных функций включают «надежность», «безопасность», «взаимодействие», «маршрутизацию» и «шаблоны обмена сообщениями» (MEP).

Модуль SOAP: Модуль SOAP — это спецификация, которая содержит объединенный синтаксис и семантику блоков заголовка SOAP, определенные согласно правилам из пункта 6.3. Модуль SOAP реализует ноль или более дополнительных функций SOAP.

Шаблон обмена сообщениями SOAP (MEP): Шаблон для обмена сообщениями SOAP между узлами SOAP, поддерживаемый одной или более привязкой SOAP к нижележащему протоколу (см. раздел 7). SOAP MEP — пример дополнительной функции SOAP (см. пункт 6.2).

Приложение SOAP: Объект, обычно программное обеспечение, которое производит, использует или иным образом реагирует на сообщения SOAP способом, соответствующим модели обработки SOAP (см. раздел 5).

4.4.2 Понятие инкапсуляции данных

Сообщение SOAP: Основная единица передачи между узлами SOAP.

Конверт SOAP: Информационный объект-элемент, обрамляющий сообщение SOAP.

Заголовок SOAP: Совокупность нуля или большего количества блоков заголовка SOAP, каждый из которых может быть предназначен любому получателю SOAP на пути следования сообщения SOAP.

Блок заголовка SOAP: Информационный объект-элемент, используемый, чтобы разграничить данные, которые логически составляют единственный вычислительный модуль внутри заголовка SOAP. Тип блока заголовка SOAP идентифицирован расширенным именем XML информационного объекта-элемента блока заголовка.

Тело SOAP: Набор нуля или больше информационных элементов, предназначенный для конечного получателя SOAP на пути следования сообщения SOAP (см. пункт 8.3).

Отказ SOAP: Информационный объект-элемент SOAP, который содержит информацию об отказе, сгенерированном узлом SOAP.

4.4.3 Понятия отправителя и получателя сообщения

Отправитель SOAP: Узел SOAP, который передает сообщение SOAP.

Получатель SOAP: Узел SOAP, который принимает сообщение SOAP.

Путь следования сообщения SOAP: Набор узлов SOAP, через которые передается отдельное сообщение SOAP. В путь также входят начальный отправитель SOAP, ноль или более посредников SOAP, и конечный получатель SOAP.

Начальный отправитель SOAP: Узел SOAP, который создает сообщение SOAP в начальной точке пути следования сообщения SOAP.

Посредник SOAP: Посредник SOAP — это и получатель, и отправитель SOAP. При этом он может быть упомянут явно в сообщении SOAP. Посредник обрабатывает блоки заголовка SOAP, предназначенные для него, и выполняет действия, необходимые для передачи сообщения SOAP к конечному получателю SOAP.

Конечный получатель SOAP: Получатель SOAP, который является конечным пунктом назначения сообщения SOAP. Он отвечает за обработку содержимого тела SOAP и любых блоков заголовка SOAP, предназначенных для него. В некоторых случаях сообщение SOAP может не достигнуть конечного получателя SOAP, например, из-за проблемы посредника SOAP. Для одного и того же сообщения SOAP конечный получатель SOAP не может быть также и посредником SOAP (см. раздел 5).

5 Модель обработки SOAP

SOAP реализует модель распределенной разработки, которая предполагает, что сообщение SOAP генерируется начальным отправителем SOAP и отправляется конечному получателю SOAP через ноль или более посредников SOAP. Необходимо отметить, что модель распределенной разработки SOAP может поддерживать многие шаблоны MEPs и включает, не ограничиваясь ими, односторонние сообщения, взаимодействия типа «запрос/ответ» и одноранговые переговоры (описание функциональной зависимости между шаблонами обмена сообщениями SOAP и моделью расширяемости SOAP приводится в пункте 6.2).

Данный пункт определяет модель распределенной разработки SOAP. Модель обработки SOAP определяет, как получатель SOAP обрабатывает сообщение SOAP. Модель определяет обработку только отдельных сообщений, изолированно от любых других сообщений SOAP. Модель обработки SOAP сама по себе не поддерживает взаимосвязи между сообщениями или согласованную обработку со-

общений, даже, например, в случае использования в сочетании с дополнительной функцией SOAP, которая задействует последовательную передачу нескольких сообщений SOAP, когда каждое последующее сообщение зависит от ответа на предыдущее сообщение. В подобном случае ответственность за любую сложную обработку лежит на такой дополнительной функции.

Раздел 6 описывает, как SOAP может быть расширен, и как расширения SOAP могут взаимодействовать с моделью обработки SOAP и структурой привязки SOAP. Раздел 4 описывает структуру привязки SOAP к протоколам, определяет схему описания правил обмена сообщениями SOAP с использованием различных нижележащих протоколов.

5.1 Узлы SOAP

Узел SOAP может быть начальным отправителем SOAP, конечным получателем SOAP или посредником SOAP. Узел SOAP, получающий сообщение SOAP, ДОЛЖЕН выполнить обработку согласно модели обработки SOAP как описано в этом разделе и далее в этой спецификации. Узел SOAP идентифицируется универсальным идентификатором ресурса URI (см. пункт 8.4.3).

5.2 Роли SOAP и узлы SOAP

При обработке сообщения SOAP узел SOAP действует в одной или более ролях SOAP, каждая из которых имеет ролевое имя SOAP и идентифицирована URI. Роли, принятые узлом, ДОЛЖНЫ быть инвариантными во время обработки отдельного сообщения SOAP. Настоящая спецификация ограничена только обработкой отдельных сообщений SOAP. Нет никаких ограничений на возможность данного узла SOAP быть или не быть способным действовать в разных ролях при обработке более чем одного сообщения SOAP.

Таблица 2 определяет три ролевых имени, у которых есть специальное значение для сообщений SOAP (см. пункт 5.6).

Таблица 2 — Роли SOAP, определяемые данной спецификацией

Краткое название	Имя	Описание
next	http://www.w3.org/2003/05/soap-envelope/role/next	Каждый посредник SOAP и конечный получатель SOAP ДОЛЖНЫ действовать в этой роли.
none	http://www.w3.org/2003/05/soap-envelope/role/none	Узлы SOAP не ДОЛЖНЫ действовать в этой роли.
ultimateReceiver	http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver	Конечный получатель ДОЛЖЕН действовать в этой роли.

Для удовлетворения потребностей приложений SOAP, по мере необходимости, МОГУТ быть использованы и другие ролевые имена в дополнение к ролевым именам SOAP, определенным в Таблице 2.

Назначение ролевого имени SOAP состоит в том, чтобы идентифицировать узел или узлы SOAP, однако, при этом с ролевым именем SOAP не связана никакая маршрутизация или семантика обмена сообщениями. Например, роль SOAP МОЖЕТ быть идентифицирована URI, применимым для маршрутизации сообщения SOAP к надлежащему узлу SOAP. С другой стороны, целесообразно использовать роли SOAP с именами, менее тесно связанными с маршрутизацией (например, «gost.e-ecolog.ru
org/banking/anyAccountMgr») или совсем с ней не связанными (например, URI, предназначенный для идентификации «всего программного обеспечения управления кэшем». Блок заголовка SOAP, предназначенный для такой роли, мог бы использоваться, чтобы передать любому надлежащему программному обеспечению указание о том, что содержание сообщения SOAP является идемпотентным и может безопасно кэшироваться, и воспроизводиться).

За исключением трех ролевых имен SOAP, определенных в Таблице 2, настоящая спецификация не устанавливает никаких ограничений на определение конкретным узлом набора ролей, в которых он действует при обработке отдельного сообщения. Это может, например, определяться реализациями на основе следующих факторов, однако, не ограничиваться только ими: роли зафиксированы в исходном коде реализации; информация, предоставленная привязкой к нижележащему протоколу (например, URI, к которому сообщение было физически передано); или настроечная информация, предоставленная пользователями во время установки системы.

5.3 Предназначение блоков заголовка SOAP

Блок заголовка SOAP МОЖЕТ нести в себе информационный объект-атрибут role (см. пункт 8.2.2), который используется для указания того, что блок заголовка предназначен для узлов SOAP, действующих в указанной роли. Данная спецификация рассматривает значение информационного объекта-атрибута role SOAP как роль SOAP для соответствующего блока заголовка SOAP.

Считается, что блок заголовка SOAP предназначен для узла SOAP, если роль SOAP в блоке заголовка совпадает с именем роли, в которой действует узел SOAP. Блоки заголовка SOAP, предназначенные для специальной роли «gost.e-ecolog.ru, формально никогда не обрабатываются. Такие блоки заголовка SOAP МОГУТ нести данные, которые необходимы для обработки других блоков заголовка SOAP. Если они не удалены в процессе обработки посредниками (см. пункт 5.7), то такие блоки передаются вместе с сообщением к конечному получателю (см. также пункт 6.3).

5.4 Понимание блоков заголовка SOAP

Вероятно, со временем будут разработаны спецификации разнообразных функций заголовка (т.е., модулей SOAP) (см. пункт 6.3), и на отдельных узлах SOAP может быть установлено программное обеспечение, необходимое для реализации одного или более таких возможных расширений. Считается, что блок заголовка SOAP понятен узлу SOAP, если программное обеспечение на этом узле SOAP разработано таким образом, чтобы полностью соответствовать и реализовать семантику, заданную развернутым именем XML самого внешнего информационного объекта-элемента данного блока заголовка.

Блок заголовка SOAP МОЖЕТ содержать информационный объект-атрибут mustUnderstand (см. пункт 8.2.3). Считается, что блок заголовка SOAP обязателен в том случае, если значение этого информационного объекта-атрибута равно «true».

Предполагается, что обязательные блоки заголовка SOAP так или иначе изменяют семантику других блоков заголовка SOAP или элементов тела SOAP. Поэтому каждый обязательный блок заголовка SOAP, предназначенный для определенного узла, этот узел ДОЛЖЕН либо обработать, либо отказаться от обработки сообщения SOAP вообще и вместо обработки сгенерировать отказ (см. пункт 5.6 и пункт 8.4). Пометка блока заголовка SOAP как «обязательный» таким образом гарантирует, что подобные изменения не будут (очевидно, ошибочно) проигнорированы без каких-либо сообщений узлом SOAP, для которого предназначен блок заголовка.

Информационный объект-атрибут mustUnderstand не предназначен служить механизмом обнаружения ошибок маршрутизации, ошибочной идентификации узлов, отказа узла действовать в назначенной роли или ролях и т.д. Любая из этих ситуаций может привести к отказу даже от попытки обработки данного блока заголовка SOAP конверта SOAP. Поэтому настоящая спецификация не требует, чтобы любой отказ был сгенерирован на основе присутствия или значения *информационного объекта атрибута* mustUnderstand в блоке заголовка SOAP, не предназначенном для обработки в текущем узле. В частности, для конечного получателя SOAP получение сообщения, содержащего обязательный блок заголовка SOAP, который предназначен для роли, не выполняемой конечным получателем SOAP, не является ошибкой. Подобное возможно, например, в случае, если блок заголовка SOAP остался в результате ошибки в маршрутизации или ошибки назначения блока на одном из предыдущих посредников.

5.5 Структура и интерпретация тела SOAP

Конечный получатель SOAP ДОЛЖЕН правильно обработать непосредственные дочерние элементы тела SOAP (см. пункт 8.3). Однако, за исключением отказов SOAP (см. пункт 8.4), Часть 1 данной спецификации (настоящий документ) не налагает требований на определенную структуру или интерпретацию этих элементов, и не предоставляет никаких средств для определения технологии необходимой обработки.

5.6 Обработка сообщений SOAP

В данном разделе изложены правила, по которым обрабатываются сообщения SOAP.

В настоящей спецификации ничто не препятствует использованию методов, которые могли бы обеспечить большую гибкость в порядке обработки, таких как оптимистичный параллелизм, откаты и т.д. Если не указано иначе, обработка всех сгенерированных сообщений SOAP, отказов SOAP и побочных эффектов уровня приложения ДОЛЖНА быть семантически эквивалентна выполнению по отдельности следующих шагов в указанном порядке.

1 Определить набор ролей, в которых должен действовать узел. В процессе такого определения МОЖЕТ быть проверено содержимое конверта SOAP, включая любые блоки заголовка SOAP и тело SOAP.

2 Идентифицировать все обязательные блоки заголовка, предназначенные для узла.

3 Если один или больше блоков заголовка SOAP, определенных на предыдущем шаге, не понятны данному узлу, то сгенерировать отдельный отказ SOAP со значением Value кода Code установленным в «env:MustUnderstand» (см. пункт 8.4.8). В случае такого отказа, дальнейшая обработка НЕ ДОЛЖНА производиться. Отказы, относящиеся к содержимому тела SOAP, НЕ ДОЛЖНЫ производиться на данном шаге.

Примечание — Настоящая спецификация использует термин *«Расширенное имя XML»*, подразумевая под этим для значения типа xsd:Qname, заданных как пара из пространства значений {абсолютная ссылка на универсальный идентификатор объекта uri, локальное имя}. Такая терминология обсуждается для включения ее в будущие версии пространства имен в XML [Namespaces in XML]. Если в будущих версиях пространства имен в XML [Namespaces in XML] будет принята другая терминология, то предполагается, что соответствующие изменения будут внесены в эту рекомендацию либо в виде исправления опечаток, либо будут внесены в будущем пересмотре версии.

4 Обработать все обязательные блоки заголовка SOAP, предназначенные для данного узла в случае конечного получателя SOAP, и тело SOAP. Узел SOAP МОЖЕТ также принять решение обработать необязательные блоки заголовка SOAP, предназначенные для него.

5 В случае, если узел является посредником SOAP и если шаблон обмена сообщениями SOAP и результаты обработки (например, не был сгенерирован какой-либо отказ) требуют, чтобы сообщение SOAP было отправлено далее по пути следования сообщения SOAP, передать сообщение далее, как описано в пункте 5.7.

Во всех случаях, при обработке блока заголовка SOAP, узел SOAP ДОЛЖЕН понять блок заголовка SOAP и ДОЛЖЕН произвести необходимую обработку способом, полностью совместимым со спецификацией для данного блока заголовка. Успешная обработка одного блока заголовка не гарантирует успешной обработки другого блока с тем же развернутым именем XML в том же сообщении. Обстоятельства, при которых такая обработка привела бы к отказу, определяет спецификация блока заголовка. Конечный получатель SOAP ДОЛЖЕН обработать тело SOAP способом, соответствующим пункту 5.5.

Признаком ошибки является генерация отказа (см. пункт 8.4). Обработка сообщения SOAP МОЖЕТ привести к генерации отказа SOAP, однако, при обработке сообщения SOAP НЕ ДОЛЖНО генерироваться более одного отказа SOAP.

Сообщение может содержать или вызывать несколько ошибок во время обработки. За исключением специально оговоренных случаев порядка обнаружения ошибок (как, например, в пункте 5.4), узел SOAP наделен свободой выбора любого отдельного отказа из набора возможных отказов, предписанных для различных возможных ошибок. Выбор отказа для генерации отказа не должен быть основан на выборе применения ключевого слова «ДОЛЖЕН», «СЛЕДУЕТ» или «МОЖЕТ», исключая тот случай, если один или больше установленных отказов квалифицируется с ключевым словом «ДОЛЖЕН». В последнем случае ДОЛЖЕН быть сгенерирован любой отказ из набора возможных отказов.

При обработке блока заголовка SOAP или тела SOAP узлы SOAP МОГУТ сослаться на любую информацию в конверте SOAP. Например, функция кэширования может кэшировать все сообщение SOAP, если это необходимо.

Обработка одного или более блоков заголовка SOAP МОЖЕТ определить или управлять порядком обработки других блоков заголовка SOAP и/или тела SOAP. Например, можно создать блок заголовка SOAP, требующий производить обработку других блоков заголовка SOAP в лексикографическом порядке. При отсутствии подобного управляющего блока заголовка SOAP порядок обработки заголовка и обработки тела определяются непосредственно узлом SOAP. Блоки заголовка МОГУТ быть обработаны в произвольном порядке. Обработка блока заголовка МОЖЕТ предшествовать, МОЖЕТ чередоваться или МОЖЕТ следовать за обработкой тела SOAP. Например, обработка блока заголовка «начать транзакцию» обычно предшествовала бы обработке тела, функция «протоколирования» могла бы работать одновременно с обработкой тела, а обработка блока заголовка «завершение транзакции» могла бы следовать после завершения всей другой работы.

Примечание — Вышеперечисленные правила регламентируют обработку в отдельном узле. Расширения SOAP могут быть разработаны таким образом, чтобы обеспечить, надлежащий порядок обработки блоков заголовка SOAP на всем пути следования сообщения к конечному получателю SOAP. В частности, такие расширения могли бы определить, что должен быть сгенерирован отказ со значением Value кода Code, установленным

«env:Sender», в том случае, если некоторые блоки заголовка SOAP случайно остались после прохождения некоторой намеченной точки в пути следования сообщения. Такие расширения при определении, произошла ли ошибка, могут зависеть от присутствия или значения *информационного объекта атрибута* mustUnderstand в оставшихся случайно блоках заголовка SOAP.

5.7 Пересылка сообщений SOAP

Как отмечалось ранее в этом разделе, предполагается, что сообщение SOAP создается в начальном отправителе SOAP и отправлено конечному получателю SOAP через ноль или более посредников SOAP. Хотя SOAP сам по себе не определяет маршрутизацию или семантику пересылки, можно ожидать, что такая функциональность может быть определена как одна или более функций расширения SOAP (см. раздел 6). Цель данного раздела заключается в описании взаимодействия передачи сообщения с моделью распределенной разработки SOAP.

SOAP определяет два различных типов посредников: пересылающих посредников и активных посредников. Оба типа посредников описаны в этом разделе.

5.7.1 Пересылка блоков заголовка SOAP

Пересылка блоков заголовка SOAP с узла-посредника SOAP зависит от того, обработаны ли эти блоки заголовка данным узлом SOAP или нет. Блок заголовка SOAP вставляется повторно, если при обработке этого блока заголовка определяется, что блок заголовка должен быть повторно вставлен в передаваемое сообщение. Спецификация блока заголовка SOAP может потребовать пересылки блока заголовка в передаваемом сообщении в том случае, если блок заголовка нацелен на роль, в которой действует посредник SOAP, но не в случае обработки данным посредником. Такие блоки заголовка называются «пересылаемыми» (relayable).

Блок заголовка SOAP МОЖЕТ нести в себе информационный объект-атрибут relay (см. пункт 8.2.4). В случае, если значение такого информационного объекта атрибута «истинно», блок заголовка является пересылаемым (relayable). Передача пересылаемых (relayable) блоков заголовка описана в пункте 5.7.2.

Информационный объект-атрибут relay не влияет на блоки заголовка SOAP, предназначенные для роли отличной от роли, в которой действует посредник SOAP.

Информационный объект-атрибут relay не влияет на модель обработки SOAP в случае, если блок заголовка также содержит информационный объект-атрибута mustUnderstand со значением «true».

Информационный объект-атрибут relay не влияет на обработку сообщений SOAP конечным получателем SOAP.

Таблица 3 суммирует возможные варианты пересылки некоторого блока заголовка узлом SOAP. Каждая строка показывает, будет ли блок заголовка передан или удален в зависимости от значения информационного атрибута role блока заголовка, действует ли узел SOAP в этой роли и был ли блок заголовка понят и обработан.

Таблица 3 — Поведение узлов SOAP по пересылке

Роль		Блок заголовка	
Краткое название	Принято	Понято и обработано	Передано
next	Да	Да	Нет, если не вставлено повторно
		Нет	Нет, если relay = «true»
Определяемый пользователем	Да	Да	Нет, если не повторно вставлено
		Нет	Нет, если relay = «true»
	Нет	Не применимо	Да
ultimateReceiver	Да	Да	Не применимо
		Нет	Не применимо
None	Нет	Не применимо	Да

5.7.2 Пересылающие посредники SOAP

Семантика одного или более блоков заголовка SOAP в сообщении SOAP или использование шаблона MEP, МОЖЕТ потребовать, чтобы сообщение SOAP было передано к другому узлу SOAP от имени инициатора входящего сообщения SOAP. В этом случае обрабатывающий узел SOAP действует в роли пересылающего посредника SOAP.

Пересылающий посредник SOAP ДОЛЖЕН обработать сообщение согласно модели обработки SOAP, определенной в пункте 5.6. Кроме того, при подготовке сообщения SOAP для пересылки посредник ДОЛЖЕН:

1 Удалить все обработанные блоки заголовка SOAP.

2 Удалить все не пересылаемые (non-relayable) блоки заголовка SOAP, которые были предназначены для пересылающего узла, но были проигнорированы во время обработки.

3 Сохранить все пересылаемые (relayable) блоки заголовка SOAP, которые были предназначены для пересылающего узла, но были проигнорированы во время обработки.

Пересылающие посредники SOAP ДОЛЖНЫ также соответствовать спецификациям использованных функций пересылки SOAP. Спецификация для каждой такой функции ДОЛЖНА определять необходимую семантику, включая правила, описывающие, как формируется передаваемое сообщение. В таких правилах МОЖЕТ быть описано размещение вставленных или повторно вставленных блоков заголовка SOAP. Вставленные блоки заголовка SOAP могут не отличаться от одного или более блоков заголовка, удаленных посредником. В данном случае обработка определена в терминах повторной вставки блоков заголовка (вместо того, чтобы оставить их на месте) с тем, чтобы подчеркнуть необходимость их обработки в каждом узле SOAP на всем пути следования сообщения SOAP.

5.7.2.1 Пересылаемый инфо-набор

В данном разделе описываются функциональные возможности пересылающих посредников SOAP относительно сохранения свойств инфо-набора XML передаваемого сообщения SOAP.

В общем случае, если только не переопределено функциями обработки в посреднике SOAP (см. пункт 5.7.2), применяются следующие правила:

1 Все свойства инфо-набора XML сообщения ДОЛЖНЫ быть сохранены, за исключением случаев, перечисленных в правилах 2—22.

2 Информационный объект-элемент для блока заголовка, предназначенного для посредника, МОЖЕТ быть удален этим посредником из свойства [children] информационного объекта-элемента Header SOAP, как это описано в пункте 5.7.2.

3 Информационные объекты-элементы для дополнительных блоков заголовка МОГУТ быть добавлены к свойству [children] информационного объекта-элемента Header SOAP, как это описано в пункте 5.7.2. В таком случае информационный объект-элемент Header SOAP МОЖЕТ быть добавлен как первый элемент свойства [children] информационного объекта-элемента Envelope SOAP, если другого НЕТ.

4 Информационные объекты пробельного символа МОГУТ быть удалены из свойства [children] информационного объекта-элемента Envelope SOAP.

5 Информационные объекты пробельного символа МОГУТ быть добавлены к свойству [children] информационного объекта-элемента Envelope SOAP.

6 Информационные объекты пробельного символа МОГУТ быть удалены из свойства [children] информационного объекта-элемента Header SOAP.

7 Информационные объекты пробельного символа МОГУТ быть добавлены к свойству [children] информационного объекта-элемента Header SOAP.

8 Информационные объекты комментариев МОГУТ быть добавлены к свойству [children] информационного объекта-элемента Envelope SOAP.

9 Информационные объекты комментариев МОГУТ быть удалены из свойства [children] информационного объекта-элемента Envelope SOAP.

10 Информационные объекты комментариев МОГУТ быть добавлены к свойству [children] информационного объекта-элемента Header SOAP.

11 Информационные объекты комментариев МОГУТ быть удалены из свойства [children] информационного объекта-элемента Header SOAP.

12 Информационные объекты атрибута МОГУТ быть добавлены к свойству [attributes] информационного объекта-элемента Envelope SOAP.

13 Информационные объекты атрибута МОГУТ быть добавлены к свойству [attributes] информационного объекта-элемента Header SOAP.

14 Информационные объекты атрибута МОГУТ быть добавлены к свойству [namespace attributes] информационного объекта-элемента Envelope SOAP.

15 Информационные объекты атрибутов МОГУТ быть добавлены к свойству [namespace attributes] информационного объекта-элемента Header SOAP.

16 Информационные объекты атрибута role SOAP, которые присутствуют в свойстве [attributes] информационных объектов элементов блока заголовка SOAP, могут быть преобразованы как описано в пункте 8.2.2.

17 Информационные объекты атрибута mustUnderstand SOAP, которые присутствуют в свойстве [attributes] информационных объектов элементов блока заголовка SOAP, могут быть преобразованы, как описано в пункте 8.2.3.

18 Информационные объекты атрибута relay SOAP, присутствующие в свойстве [attributes] информационных объектов элементов блока заголовка SOAP, могут быть преобразованы как описано в пункте 8.2.4.

19 Свойство [base URI] информационного объекта document не обязательно должно сохраняться.

20 Свойство [base URI] информационных объектов элементов МОЖЕТ быть изменено или удалено.

21 Свойство [character encoding scheme] информационного объекта document МОЖЕТ быть изменено или удалено.

Все информационные элементы namespace в [in-scope namespaces] информационных элементов ДОЛЖНЫ быть сохранены. МОГУТ быть добавлены дополнительные информационные объекты namespace.

Примечание — Приведенные правила допускают подписание блоков заголовка SOAP, тела SOAP и комбинаций блоков заголовка SOAP и тела SOAP.

При отсутствии алгоритма канонизации для нормализации преобразований инфо-набора и в случае использования алгоритма канонизации «gost.e-ecolog.ru, параграфы 1 — 6 и 11 — 14 несовместимы с подписанием конверта SOAP, а параграфы 1, 2, 5, 6, 12 и 14 несовместимы с подписанием заголовка SOAP.

Аналогично, если используется алгоритм канонизации «http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments» то, с подписанием конверта SOAP несовместимы параграфы 7 и 8, а с подписанием заголовка SOAP несовместимы параграфы 9 и 10.

Примечание — *Информационными элементами пробельного символа* считаются те, свойство [character code] которых имеет значение #x20, #x9, #xD или #xA.

5.7.3 Активные посредники SOAP

В дополнение к обработке, выполняемой пересылающими посредниками SOAP, активные посредники SOAP осуществляют дополнительную обработку, которая может внести изменения в исходящее сообщение SOA, *не* описанные в самом входящем сообщении SOAP. Т.е. они могут выполнить обработку, не описанную блоками заголовка SOAP во входящем сообщении SOAP. Потенциальный набор услуг, предоставленных активным посредником SOAP, включает в себя службы безопасности, службы аннотации и службы манипулирования контентом, однако, не ограничен этим списком.

Результаты такой активной обработки могут повлиять на интерпретацию сообщений SOAP следующими узлами SOAP. Например, в процессе генерации исходящего сообщения SOAP активный посредник SOAP может удалить или закодировать отдельные или все блоки заголовка SOAP, обнаруженные во входящем сообщении SOAP. Настоятельно рекомендуется, чтобы функции SOAP, предоставляемые активными посредниками SOAP, были определены таким образом, чтобы такие изменения могли быть обнаружены узлами SOAP, на которые они могут повлиять на пути следования сообщения.

5.8 Модель управления версиями SOAP

Версия сообщения SOAP идентифицируется расширенным именем XML информационного объекта-элемента, дочернего элемента информационного объекта document. У информационного объекта document сообщения SOAP версии 1.2 имеется дочерний информационный объект-элемент с [local name] Envelope и [namespace name] «gost.e-ecolog.ru (см. пункт 8.1).

Узел SOAP определяет, поддерживает ли он версию сообщения SOAP на основании анализа каждого сообщения. В данном контексте «поддерживать» означает понимать семантику данной версии конверта SOAP. Модель управления версиями ограничена только информационным объектом элемента Envelope SOAP. При этом не рассматриваются версии блоков заголовка SOAP, кодировок, привязки протокола или чего-либо еще.

Узел SOAP МОЖЕТ поддерживать несколько версий конверта. Однако, обрабатывая сообщение, узел SOAP ДОЛЖЕН использовать семантику, определенную версией этого сообщения.

Если узел SOAP получает сообщение, версию которого он не поддерживает, он ДОЛЖЕН генерировать отказ (см. пункт 8.4) с значением Value элемента Code, установленным как «env:VersionMismatch». Любое другое несоответствие логической структуры сообщения ДОЛЖНО привести к генерации отказа с значением Value элемента Code, установленным как «env:Sender».

Приложение А. Переход версии от SOAP/1.1 к SOAP версии 1.2 определяет механизм перехода от SOAP/1.1 к SOAP версии 1.2 с использованием информационного объекта-элемента Upgrade (см. пункт 8.4.7).

6 Модель расширяемости SOAP

SOAP обеспечивает простую структуру обмена сообщениями, базовая функциональность которой разрабатывалась с учетом обеспечения расширяемости. Механизмы расширяемости, описанные ниже, можно использовать, чтобы добавить дополнительные возможности, доступные в других системах обмена сообщениями.

6.1 Функции расширения SOAP

Функция расширения SOAP — расширение основ обмена сообщениями SOAP. SOAP не налагает никаких ограничений на потенциальные возможности таких функций. Например, такие функции включать в себя «надежность», «безопасность», «взаимозависимость», «маршрутизацию» и «шаблоны обмена сообщениями» (MEPs), а также запрос/ответ, односторонний и одноранговый обмен сообщениями.

Модель расширяемости SOAP обеспечивается двумя механизмами, посредством которых могут быть реализованы функции расширения — это: Модель обработки SOAP и Структура привязки протокола SOAP (см. разделы 5 и 7). Первый из них определяет функциональные возможности отдельного узла SOAP, относящиеся к обработке отдельного сообщения. Последний — обеспечивает выполнение отправки и получения сообщений SOAP узлом SOAP через нижележащий протокол.

Модель обработки SOAP позволяет узлам SOAP, в которых имеются механизмы, необходимые для выполнения одной или более функций, реализовать такие функции в виде блоков заголовка SOAP внутри конверта SOAP (см. пункт 5.4). Такие блоки заголовка могут быть предназначены для любого узла или узлов SOAP на пути следования сообщения SOAP (см. пункт 5.3). Сочетание синтаксиса и семантики блоков заголовка SOAP представляет собой модуль SOAP и определено согласно правилам, перечисленным в пункте 6.3

Напротив, привязка протокола SOAP работает между двумя смежными узлами SOAP на пути следования сообщения SOAP. Не требуется, чтобы один и тот же нижележащий протокол использовался для всех транзитных участков на пути следования сообщения SOAP. В некоторых случаях нижележащие протоколы имеют сами по себе или посредством расширения механизмы выполнения определенных функций. Посредством спецификации привязки структура привязки протокола SOAP обеспечивает схему для описания таких функций и их связи с узлами SOAP (см. раздел 7).

Некоторые функции расширения могут требовать семантики обработки от начального узла до конечного (end-to-end) в отличие от пошаговой от узла до узла (hop-by-hop). Несмотря на то, что структура привязки протокола SOAP позволяет выражать сквозные (end-to-end) функции вне конверта SOAP, какого-либо стандартного механизма обработки посредниками созданных таким образом сообщений не предусмотрено. Спецификация привязки, которая определяет такие функции вне конверта SOAP, должна определять свои собственные правила обработки для таких внешне выраженных функций. Ожидается, что узел SOAP будет соответствовать этим правилам обработки (например, посредством описания того, какая информация передается вместе с сообщением SOAP, исходящем от него, как посредника). Обработка конвертов SOAP в соответствии с моделью обработки SOAP (см. раздел 5) НЕ ДОЛЖНА переопределяться спецификациями привязки.

Там, где это целесообразно, рекомендуется, чтобы сквозные (end-to-end) функции расширения были выражены в виде блоков заголовка SOAP таким образом, чтобы могли использоваться правила, определенные моделью обработки SOAP.

6.1.1 Требования к функциям расширения

Спецификация функции расширения ДОЛЖНА включать следующее:

1 Используемый URI имени функции расширения. Это обеспечивает функции возможность ссылаться на нее в языках описания или во время переговоров.

2 Информация (состояние), требуемая в каждом узле для реализации функции.

3 Обработка, требуемая в каждом узле для того, чтобы выполнить необходимые действия функции, включая и обработку любых коммуникационных отказов, которые могли бы произойти на уровне нижележащего протокола (см. также пункт 7.2).

4 Информация, которую нужно передавать от узла к узлу.

Дополнительные требования к функции расширения MEP приведены в пункте 6.2.

6.2 Шаблоны обмена сообщениями SOAP (MEPs)

Шаблон обмена сообщениями (MEP) — это шаблон, который устанавливает образец для обмена сообщениями между узлами SOAP. MEP — это вид функции расширения, и, если не указано иное, ссылки в этой спецификации на термин «функция расширения» применимы также к MEPs. MEP «запрос-ответ» определенный в Части 2 SOAP 1.2 [Часть 2 SOAP] иллюстрирует спецификацию функции MEP.

Спецификация шаблона обмена сообщениями ДОЛЖНА:

- как требуется в пункте 6.1.1, предоставить URI как имя MEP;
- определить жизненный цикл обмена сообщениями, соответствующего шаблону;
- определить временные / причинно-следственные связи, если таковые имеются для множественных сообщений, обмен которыми производится в соответствии с шаблоном (например, ответы следуют за запросами и отправлены инициатору запросов.);
- определить нормальное и аварийное завершение процесса обмена сообщения, соответствующего шаблону.

Базовые спецификации привязки протокола могут определить свою собственную поддержку одного или нескольких именованных MEPs.

MEP является функцией расширения SOAP, таким образом, спецификация MEP ДОЛЖНА соответствовать требованиям к спецификации функции расширения SOAP (см. пункт 6.1.1). Спецификация MEP ДОЛЖНА также включать:

1 Любые требования к генерации дополнительных сообщений (например, ответов на запросы в MEP «запрос/ответ»).

2 Правила для доставки или других действий с отказами SOAP, произошедших в процессе работы MEPs.

6.3 Модули SOAP

Термин «SOAP модуль» относится к спецификации синтаксиса и семантики одного или более блоков заголовка SOAP. Модуль SOAP реализует ноль или более функций SOAP. Спецификация модуля придерживается следующих правил. Модуль:

1 ДОЛЖЕН идентифицировать себя посредством URI. Это позволяет однозначно ссылаться на модуль на языках описания или во время переговоров.

2 ДОЛЖЕН определить функции, обеспечиваемые модулем (см. пункт 6.1).

3 ДОЛЖЕН четко и полностью определить содержание и семантику блоков заголовка SOAP, используемых для реализации рассматриваемых функциональных возможностей, включая в необходимых случаях внесение каких-либо изменений в модель обработки SOAP. Модель расширяемости SOAP не ограничивает степень возможного расширения SOAP, и при этом не препятствует тому, чтобы расширения изменили модель обработки SOAP, описанную в Части 2 «Модель обработки SOAP».

4 В описании функциональности, которую обеспечивает модуль, МОЖНО использовать соглашения о свойствах, определенные в SOAP 1.2 Части 2 [Часть 2 SOAP], раздел Соглашение по описанию функций расширения и привязки. Если эти соглашения соблюдаются, то спецификация модуля ДОЛЖНА четко определить отношения между абстрактными свойствами и их представлениями в конверте SOAP. Следует отметить, что можно специфицировать функцию расширения исключительно в терминах абстрактных свойств, а затем написать отдельную спецификацию модуля, который реализует эту функцию, отображая в модуле SOAP свойства, определенные в спецификации функции, в блоки заголовка SOAP.

5 ДОЛЖНЫ быть четко определены любые известные взаимодействия с телом SOAP или какие-либо изменения в интерпретации его. Кроме того, ДОЛЖНЫ быть четко определены любые известные взаимодействия с другими функциями расширения SOAP и модулями SOAP или какие-либо изменения в их интерпретации. Например, можно представить себе модуль, который шифрует и удаляет тело SOAP, вставляя вместо этого блок заголовка SOAP, содержащий контрольную сумму и указание относительно используемого механизма шифрования. Спецификация такого модуля указала бы, что алгоритм расшифрования на принимающей стороне должен быть применен ранее выполнения каких-либо других модулей, которые имеют дело с содержимым тела SOAP.

7 Структура привязки SOAP к протоколам

SOAP позволяет обмениваться сообщениями SOAP, используя различные надлежащие протоколы. Формальный набор правил передачи сообщения SOAP внутри или поверх другого протокола (нижележащего протокола) с целью обмена сообщениями называют привязкой. Структура привязки протокола SOAP обеспечивает общие правила для спецификации привязки протокола. Структура также определяет отношения между привязкой и узлами SOAP, которые реализуют такую привязку. Привязка к HTTP, описанная в Части 2 SOAP 1.2 [Часть 2 SOAP], иллюстрирует спецификацию привязки. Дополнительные привязки могут быть созданы посредством спецификаций, которые соответствуют структуре привязки, представленной в этой главе.

Спецификация привязки SOAP:

- определяет функции расширения, обеспечиваемые привязкой;
- определяет, каким образом службы нижележащего протокола используются, чтобы передать инфо-наборы сообщения SOAP;
- определяет, как службы нижележащего протокола используются для выполнения условий, накладываемых функциями расширения, которые поддерживаются данной привязкой;
- определяет обработку всех потенциальных отказов, которые могут ожидаться в этой привязке;
- определяет требования к созданию совместимой реализации определенной привязки.

Привязка не представляет отдельную модель обработки и не является собственно узлом SOAP. Скорее, привязка SOAP — это неотъемлемая часть узла SOAP (см. 5 Модель обработки SOAP).

7.1 Назначение структуры привязки

Назначение структуры привязки заключается в следующем:

1 Определить требования и понятия, которые характерны для всех спецификаций привязки.

2 Упростить однородное описание в ситуациях, когда несколько привязок поддерживает одни и те же функции, способствуя повторному использованию привязок.

3 Обеспечить согласованность спецификаций дополнительных функций.

Две или более привязок могут предоставить некоторую дополнительную функцию, например надежную доставку, с использованием различных средств. Одна привязка могла бы использовать нижележащий протокол, который непосредственно обеспечивает эту функцию (например, протокол надежен сам по себе), а другая привязка могла бы обеспечить необходимую логику (например, надежность достигается ведением журнала и повторной передачей). В таких случаях функция может быть доступной приложению единообразно, независимо от того, какая используется привязка.

7.2 Структура привязки

Создание, передача и обработка сообщения SOAP, возможно, через одного или более посредников специфицируется как распределенный конечный автомата. Состояние складывается из информации, известной узлу SOAP в данный момент времени, в том числе и содержимым сообщений, создаваемых для передачи или полученных для обработки. Состояние каждого узла может быть обновлено либо в ходе локальной обработки, либо информацией, полученной от смежного узла.

В разделе 5 настоящей спецификации описывается обработка, характерная для всех узлов SOAP при получении сообщения. Назначение спецификации привязки состоит в том, чтобы добавить к этим базовым правилам SOAP дополнительную обработку, которая является особенностью привязки, а также определить, каким образом нижележащий протокол используется для передачи информации между смежными узлами на пути следования сообщения.

Распределенный конечный автомат, который управляет передачей данного сообщения SOAP на пути следования сообщения, является комбинацией базовой обработки SOAP (см. раздел 5), работающий в каждом узле, в сочетании со спецификацией привязки, соединяющей каждую пару узлов. Спецификация привязки ДОЛЖНА содержать один или более MEP.

В случаях, когда спецификацией привязки поддерживается несколько функций расширения, спецификации этих функций ДОЛЖНЫ предоставлять всю информацию, необходимую для успешного применения функций в комбинации. Точно так же любые зависимости одной функции от другой (т.е., если успешное использование одной функции зависит от использования или неиспользования другой функции) ДОЛЖНЫ быть определены. Данная структура привязки не обеспечивает явного механизма контроля за использованием таких взаимозависимых функций.

Структура привязки не обеспечивает каких-либо средств для наименования или утилизирования информации, включающей в себя состояние в данном узле. Отдельная функция и спецификация привязки могут вводить собственные соглашения для спецификации состояние. Следует отметить, однако, что согласованность между привязками и функциями расширения, вероятно, будет лучше в тех ситуациях, когда спецификации нескольких функций принимают непротиворечивые соглашения для представления состояния. Например, несколько функций могли бы воспользоваться согласованной спецификацией для учетных данных аутентификации, идентификатора транзакции и т.д. Привязка HTTP в Части 2 SOAP 1.2 [Часть 2 SOAP] иллюстрирует одно из таких соглашений.

Как описано в разделе 8, каждое сообщение SOAP определено как инфо-набор XML, который состоит из информационного объекта document с одним и только одним дочерним элементом — конвертом: информационным объектом элемента Envelope SOAP. Поэтому минимальная задача привязки при передаче сообщения состоит в том, чтобы определить средства, которыми инфо-набор сообщения SOAP передается и воссоздается привязкой на получающем узле SOAP, и определить, каким образом осуществляется передача конверта с использованием функций нижележащего протокола.

В разделе 8 определено, что все конверты SOAP могут быть сериализованы с использованием сериализации XML 1.0. Таким образом привязки МОГУТ в качестве представления для передачи (on the wire) инфо-набора XML использовать XML 1.0 или более поздние версии. Однако структура привязки не требует, чтобы каждая привязка использовала бы сериализацию XML для передачи. Когда это необходимо, могут использоваться сжатие, шифрование, фрагментированные представления и т.д. В случае использования сериализации XML инфо-набора XML привязка МОЖЕТ потребовать использования определенной кодировки символов или набора кодировок.

В случае использования сериализации XML для сериализации инфо-набора или делегирования сериализации другим средствам (таким, как описание типа медиа) привязка должна перечислить используемые версии XML. Для обеспечения функциональной совместимости список поддерживаемых версий XML должен быть исчерпывающим.

Обработка сообщений в привязке МОЖЕТ обеспечить потоковую передачу. То есть узлы SOAP МОГУТ начать обрабатывать полученное сообщение SOAP сразу же, как только доступна необходимая информация. Обработка SOAP определена в терминах инфо-наборов сообщения SOAP (см. раздел 8). Несмотря на то, что при потоковой передаче получатель SOAP принимает такие инфо-наборы XML постепенно, результат обработки SOAP ДОЛЖЕН быть идентичным результату обработки в случае, когда конверт SOAP полностью был доступен до начала обработки. Например, как предусмотрено в пункте 5.6, идентификация предназначенных блоков заголовка SOAP и проверка всех атрибутов mustUnderstand должны быть сделаны прежде, чем будет продолжена дальнейшая обработка. В зависимости от представления, используемого для инфо-набора XML и порядка, в котором он передается, данное правило может ограничить характеристику потока передачи, которая может быть достигнута.

Привязка МОЖЕТ зависеть от состояния, которое моделируется за пределами инфо-набора сообщения SOAP (например, счетчики повторной передачи), и МОЖЕТ передавать такую информацию смежным узлам. Например, некоторая привязка может принимать адрес доставки сообщения (обычно URI), которого нет в конверте.

8 Логическая структура сообщения SOAP

Сообщение SOAP определено как инфо-набор XML, в котором информационные объекты комментариев, элементов, атрибутов, пространств имен и символов (character) могут быть сериализованы как XML 1.0. Необходимо отметить, что требование возможности сериализации указанных информационных элементов инфо-наборов сообщения SOAP как XML 1.0 НЕ ЯВЛЯЕТСЯ требованием сериализации с использованием XML 1.0. Инфо-набор сообщения SOAP содержит информационный объект — document с одним и только одним элементом в его свойстве [children], которое ДОЛЖНО быть информационным объектом элемента Envelope SOAP (см. пункт 8.1). Этот информационный объект является также и значением свойства [document element]. Свойства [notations] и [unparsed entities] оба пусты. Рекомендация «Информационный набор XML» [XML InfoSet] допускает контент, который не может быть непосредственно сериализован с использованием XML; например, символ #x0 допускается в инфо-наборе, но запрещен в XML. XML инфо-набор сообщения SOAP ДОЛЖЕН соответствовать требованиям сериализации XML 1.0 [XML 1.0].

Инфо-набор XML сообщения SOAP НЕ ДОЛЖЕН содержать информационный объект декларации типа документа (document type declaration).

Сообщения SOAP, отправленные начальными отправителями SOAP, НЕ ДОЛЖНЫ содержать информационные объекты инструкций обработки (processing instruction). Посредники SOAP НЕ ДОЛЖНЫ вставлять информационные объекты инструкций обработки в сообщения SOAP, которые они передают. Получатели SOAP, получающие сообщение SOAP, содержащее информационные объекты инструкций обработки, ДОЛЖНЫ генерировать отказ SOAP со значением Value элемента Code, установленным как «env:Sender». Однако в тех случаях, когда обнаружение посредником информационных элементов объектов инструкций обработки в пересылаемом сообщении нецелесообразно из соображений производительности, посредник МОЖЕТ оставить такие информационные объекты инструкций обработки неизменными в пересылаемом сообщении.

У информационных объектов элементов, определенных данной спецификацией, для которых определено, что свойство [children] может содержать только информационные объекты элементов, может также быть ноль или более дочерних символьных (character) информационных объектов. Символьный код каждого такого символьного информационного объекта ДОЛЖЕН принадлежать множеству пробельных символов, как определено XML 1.0 [XML 1.0]. Если не указано иначе, такие символьные информационные элементы считаются незначащими.

Информационные объекты комментариев МОГУТ присутствовать как дочерние элементы и/или потомки информационного объекта [document element], но не перед или после этого информационного объекта. В модели обработки имеются некоторые ограничения относительно того, когда информационные объекты комментариев могут быть добавлены и/или удалены (см. пункт 5.7.2.1).

8.1 Конверт SOAP

Информационный объект-элемент Envelope SOAP имеет:

- [local name] Envelope;
- [namespace name] gost.e-ecolog.ru ;
- ноль или более соответствующих требованиям пространства имен информационных объектов атрибутов в свойстве [attributes];
- один или два информационных объекта элементов в его свойстве [children] в следующем порядке:

1 Необязательный информационный объект-элемент Header (см. пункт 8.2).

2 Обязательный информационный объект-элемент body (см. пункт 8.3).

8.1.1 Атрибут encodingStyle SOAP

Информационный объект-атрибут encodingStyle указывает на правила кодирования, используемые для сериализации частей сообщения SOAP.

Информационный объект-атрибут encodingStyle имеет:

- [local name] encodingStyle;
- [namespace name]"gost.e-ecolog.ru .

Информационный объект-атрибут encodingStyle имеет тип xs:anyURI. Его значение идентифицирует набор правил преобразования в последовательную форму, которые могут использоваться, чтобы десериализовать сообщение SOAP.

Информационный объект-атрибут encodingStyle МОЖЕТ присутствовать в следующих компонентах:

1 Блок заголовка SOAP (см. пункт 8.2.1).

2 Дочерний информационный объект-элемент информационного объекта-элемента Body SOAP (см. пункт 8.3.1), если только этот дочерний элемент не является информационным объектом-элементом отказа SOAP (см. пункт 8.4).

3 Дочерний информационный объект-элемент информационного объекта-элемента Detail SOAP (см. пункт 8.4.5.1 Элементы детализации SOAP).

4 Любой потомок для вышеперечисленных в 1, 2 и 3.

Информационный объект-атрибут encodingStyle не ДОЛЖЕН появляться ни в каких элементах инфо-набора сообщения SOAP, кроме вышеупомянутых.

Область действия информационного объекта атрибута encodingStyle — это область действия его [owner element] и потомков этого информационного объекта-элемента, за исключением области действия какого-либо внутреннего информационного объекта-атрибута encodingStyle. Если в области действия нет никакого информационного объекта-атрибута encodingStyle или значением такого информационного объекта-атрибута является «http://gost.e-ecolog.ru , то никаких требований относительно кодирования этого информационного объекта-элемента и его потомков не накладывается.

Пример – Возможные значения для информационного объекта атрибута encodingStyle.

«gost.e-ecolog.ru

«gost.e-ecolog.ru

«gost.e-ecolog.ru

8.2 Заголовок SOAP

Информационный объект-элемент Header SOAP обеспечивает механизм расширения сообщения SOAP децентрализованным и модульным способом (см. раздел 6 и пункт 5.4).

Информационный объект-элемент Header имеет:

- [local name] Header;
- [namespace name]http://www.w3.org/2003/05/soap-envelope;
- ноль или более соответствующих требованиям пространства имен информационных объектов атрибута в его свойстве [attributes];
- ноль или более соответствующих требованиям пространства имен информационных объектов элементов в его свойстве [children].

Каждый дочерний информационный объект-элемент заголовка SOAP называют блоком заголовка SOAP.

8.2.1 Блок заголовка SOAP

Каждый информационный объект-элемент блока заголовка SOAP:

- ДОЛЖЕН иметь свойство [namespace name], у которого есть значение; т.е. имя элемента ДОЛЖНО быть соответствующим требованиям пространства имен;
- МОЖЕТ иметь любое число дочерних символьных (character) информационных объектов. Дочерние символьные информационные объекты, символьный код которых принадлежит множеству пробельных символов, как определено в XML 1.0 [XML 1.0], считаются значимыми;
- МОЖЕТ иметь любое число дочерних информационных объектов элементов. Такие информационные объекты элементов МОГУТ соответствовать требованиям пространства имен;
- МОЖЕТ иметь ноль или более информационных объектов атрибутов в свойстве [attributes]. Среди них МОГУТ быть любые из следующих, у которых есть специальное назначение для обработки SOAP:
- информационный объект-атрибут encodingStyle (см. пункт 8.1.1);
- информационный объект-атрибут role (см. 8.2.2 Атрибут role SOAP);
- информационный объект-атрибут mustUnderstand (см. пункт 8.2.3);
- информационный объект-атрибут relay (см. пункт 8.2.4).

Пример – Заголовок SOAP с единственным блоком заголовка SOAP

```
<env:Header xmlns:env = "gost.e-ecolog.ru                    >
  <t:Transaction xmlns:t = "gost.e-ecolog.ru                 "
       env:mustUnderstand = "true">
  5
  </t:Transaction>
</env:Header>
```

8.2.2 Атрибут role SOAP

Роль SOAP используется, чтобы указать на узел SOAP, для которого предназначен определенный блок заголовка SOAP (см. пункт 5.2).

Информационный объект-атрибут role имеет следующие свойства инфо-набора XML:

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

Тип информационного объекта атрибута role - xs:anyURI. Значение информационного объекта атрибута role – это URI, который определяет роль, которую может выполнять узел SOAP.

Если информационный объект-атрибут role SOAP опущен, то это эквивалентно предоставлению этого атрибута со значением «gost.e-ecolog.ru .

Отправители SOAP НЕ ДОЛЖНЫ генерировать, но получатели SOAP ДОЛЖНЫ воспринимать информационный объект-атрибут role SOAP со значением «gost.e-ecolog.ru ultimateReceiver».

Передавая сообщение, посредник SOAP МОЖЕТ пропустить информационный объект-атрибут role SOAP, если его значение «gost.e-ecolog.ru (см. пункт 5.7).

Отправитель SOAP, генерирующий сообщение SOAP, ДОЛЖЕН использовать информационный объект-атрибут role только в блоках заголовка SOAP. Получатель SOAP ДОЛЖЕН проигнорировать этот информационный объект-атрибут, если он располагается внутри потомка блока заголовка SOAP или внутри дочернего информационного объекта-элемента тела SOAP (или его потомка).

8.2.3 Атрибут mustUnderstand SOAP

Информационный объект-атрибут mustUnderstand SOAP используется, чтобы указать, обязательна ли обработка блока заголовка SOAP, или она является дополнительной (см. пункт 5.4).

Информационный объект-атрибут mustUnderstand имеет следующие свойства инфо-набора XML:

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

Тип информационного объекта атрибута mustUnderstand - xs:boolean.

Если этот информационный объект-атрибут опущен, то по определению это семантически эквивалентно его присутствию со значением «false».

Отправители SOAP не ДОЛЖНЫ генерировать, но получатели SOAP ДОЛЖНЫ воспринимать информационный объект-атрибут mustUnderstand SOAP со значением «false» или «0».

Генерируя информационный объект-атрибут mustUnderstand SOAP, отправитель SOAP ДОЛЖЕН использовать каноническое представление «true» для значения атрибута (см. XML-схему [Часть 2 XML-схемы]). Получатель SOAP ДОЛЖЕН воспринимать любое действительное лексическое представление значения атрибута.

Передавая сообщение, посредник SOAP МОЖЕТ заменить значение «true» на значение «1», или «false» на «0». Кроме того, посредник SOAP МОЖЕТ опустить информационный объект-атрибут mustUnderstand SOAP, если его значение — «false» (см. пункт 5.7).

Отправитель SOAP, генерирующий сообщение SOAP, ДОЛЖЕН использовать информационный объект-атрибут mustUnderstand только в блоках заголовка SOAP. Получатель SOAP ДОЛЖЕН проигнорировать этот информационный объект-атрибут, если он размещен в потомках блока заголовка SOAP или в дочерних информационных объектах элементов тела SOAP (или его потомках).

8.2.4 Атрибут relay SOAP

Информационный объект-атрибут relay SOAP используется, чтобы указать, должен ли передаваться блок заголовка SOAP, предназначенный для получателя SOAP, если он не был обработан (см. пункт 5.7.1).

Информационной объект-атрибут relay имеет следующие свойства инфо-набора XML:

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

Тип информационного объекта атрибута relay - xs:boolean.

Если этот информационный объект-атрибут опущен, то по определению это семантически эквивалентно его включению со значением «false».

Отправители SOAP не ДОЛЖНЫ генерировать, но получатели SOAP ДОЛЖНЫ воспринимать, информационный объект-атрибут relay SOAP со значением «false» или «0».

Генерируя информационный объект-атрибут relay SOAP, отправитель SOAP ДОЛЖЕН использовать каноническое представление, «true» для значения атрибута (см. XML-схему [Часть 2 XML-схемы]). Получатель SOAP ДОЛЖЕН воспринимать любое действительное лексическое представление значения атрибута.

Передавая сообщение, посредник SOAP МОЖЕТ заменить значение «true» на значение «1» или «false» на «0». Кроме того, посредник SOAP МОЖЕТ опустить информационный объект-атрибут relay SOAP, если его значение – «false» (см. пункт 5.7).

Отправитель SOAP, генерирующий сообщение SOAP, ДОЛЖЕН использовать информационный объект-атрибут relay только в блоках заголовка SOAP. Получатель SOAP ДОЛЖЕН проигнорировать этот информационный объект-атрибут, если он размещен в потомках блока заголовка SOAP или в дочерних информационных элементах тела SOAP (или его потомках).

8.3 Тело SOAP

Тело SOAP обеспечивает механизм передачи информации к конечному получателю SOAP (см. пункт 5.5).

Информационный объект-элемент Body имеет:

- [local name] Body;
- [namespace name]http://www.w3.org/2003/05/soap-envelope;
- ноль или более соответствующих требованиям пространства имен информационных объектов атрибутов в своем свойстве [attributes];
- ноль или более соответствующих требованиям пространства имен информационных объектов элементов в своем свойстве [children].

У информационного объекта-элемента Body МОЖЕТ быть любое число дочерних символьных информационных объектов. Символьный код таких символьных информационных объектов ДОЛЖЕН принадлежать множеству пробельных символов, как определено в XML 1.0 [XML 1.0]. Они рассматриваются как значимые.

8.3.1 Дочерний элемент тела SOAP

Каждый дочерний информационный элемент информационного объекта-элемента тела SOAP должен удовлетворять следующим требованиям:

- ДОЛЖЕН иметь свойство [namespace name], у которого есть значение; т.е. имя элемента ДОЛЖНО соответствовать требованиям пространства имен;

Примечание — Элементы, чьи имена квалифицированы некоторым пространством имен, как правило, генерируют сообщения, интерпретация которых менее неоднозначна по сравнению с элементами без пространства имен. Поэтому использование элементов без пространства имен не желательно.

- МОЖЕТ иметь любое число дочерних символьных информационных объектов. Дочерние символьные информационные объекты, символьный код которых принадлежит множеству пробельных символов, как определено в XML 1.0 [XML 1.0], считаются значимыми;
- МОЖЕТ иметь любое число дочерних информационных объектов элементов. Имена таких информационных объектов элементов МОГУТ быть квалифицированы пространством имен;
- МОЖЕТ иметь ноль или более информационных объектов атрибутов в своем свойстве [attributes]. Среди них МОЖЕТ быть следующий атрибут со специальным значением для обработки SOAP:
- информационный объект-атрибут encodingStyle (см. пункт 8.1.1).

SOAP определяет один конкретный прямой дочерний элемент тела SOAP — отказ SOAP, который используется для сообщений об ошибках (см. пункт 8.4).

8.4 Отказ SOAP

Отказ SOAP используется, чтобы перенести информацию об ошибке в сообщении SOAP.

Информационный элемент Fault имеет:

- [local name] Fault;
- [namespace name]http://www.w3.org/2003/05/soap-envelope;
- два или более дочерних информационных объекта элементов в его свойстве [children] в следующем порядке:

1. Обязательный информационный объект-элемент Code (см. пункт 8.4.1).
2. Обязательный информационный объект-элемент Reason (см. пункт 8.4.2).
3. Необязательный информационный объект-элемент Node (см. пункт 8.4.3).
4. Необязательный информационный объект-элемент Role (см. пункт 8.4.4).
5. Необязательный информационный элемент Detail (см. пункт 8.4.5).

Чтобы быть распознанным в качестве несущего информацию об ошибке SOAP, сообщение SOAP ДОЛЖНО содержать единственный информационный объект-элемент Fault SOAP как единственного дочернего информационного объекта-элемента Body SOAP.

Генерируя отказ, отправители SOAP не ДОЛЖНЫ включать дополнительные информационные объекты элементов в Body SOAP. Семантика сообщения Body, которое содержит Fault плюс дополнительные информационные объекты элементов, в SOAP не определена.

Информационный объект-элемент Fault SOAP МОЖЕТ появиться либо внутри блока заголовка SOAP, либо как потомок дочернего информационного объекта-элемента Body SOAP. В обоих случаях семантика элемента не определена в SOAP.

8.4.1 Элемент Code SOAP

Информационный объект-элемент Code имеет:

- [local name] Code;
- [namespace name]http://www.w3.org/2003/05/soap-envelope;

- один или два дочерних информационных объекта элементов в его свойстве [children], в следующем порядке:

1 Обязательный информационный объект-элемент Value как описано ниже (см. пункт 8.4.1.1).

2 Необязательный информационный объект-элемент Subcode как описано ниже (см. пункт 8.4.1.2).

8.4.1.1 Элемент Value SOAP (с родителем Code)

Информационный объект-элемент Value имеет:

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

Тип информационного объекта-элемента Value - env:faultCodeEnum. SOAP определяет небольшой набор кодов отказа SOAP, покрывающих отказы SOAP высокого уровня (см. пункт 8.4.6).

8.4.1.2 Элемент Subcode SOAP

Информационный объект-элемент Subcode имеет:

- [local name] Subcode;
- [namespace name] gost.e-ecolog.ru ;
- один или два дочерних информационных объектов элементов в своем свойстве [children], в следующим порядке:

1 Обязательный информационный объект-элемент Value как описано ниже (см. пункт 8.4.1.3).

2 Необязательный информационный объект-элемент Subcode (см. пункт 8.4.1.2).

8.4.1.3 Элемент Value SOAP (с родителем Subcode)

Информационный объект-элемент Value имеет:

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

Тип информационного объекта-элемента Value - xs:QName. Определенное приложением значение этого элемента — это значение подкатегории дочернего информационного объекта-элемента Value родительского информационного объекта-элемента Subcode (см. пункт 8.4.6).

8.4.2 Элемент Reason SOAP

Информационный объект-элемент Reason предназначен обеспечить понятное человеку объяснение причины отказа.

Информационный элемент Reason имеет:

- [local name] Reason;
- [namespace name] gost.e-ecolog.ru ;
- один или более дочерних информационных объектов элементов Text (см. пункт 8.4.2.1). Каждый дочерний информационный объект-элемент Text ДОЛЖЕН иметь уникальное значение своего информационного объекта атрибута xml:lang.

Тип информационного объекта-элемента Reason - env:faultReason.

8.4.2.1 Элемент Text SOAP

Информационный объект-элемент Text предназначен для передачи текста понятного человеку объяснения отказа.

Информационный объект-элемент Text имеет:

- [local name] Text;
- [namespace name] gost.e-ecolog.ru ;
- обязательный информационный объект-атрибут с [local name] lang и [namespace name] «gost.e-ecolog.ru . Следует обратить внимание на то, что в определении информационного объекта атрибута lang требуется, чтобы [prefix] был «xml» любым написанием строчных и прописных букв (см. XML 1.0 [XML 1.0], Идентификация Языка);
- любое число дочерних символьных информационных объектов. Дочерние символьные информационные объекты, символьный код которых принадлежит множеству пробельных символов, как определено в XML 1.0 [XML 1.0], рассматриваются как значимые.

Тип информационного объекта-элемента Text - env:reasontext

Этот информационный объект-элемент аналогичен [заголовку] 'Reason-Phrase', определенному HTTP [RFC 2616], и ДОЛЖЕН предоставить информацию, поясняющую природу отказа. Этот элемент не предназначен для алгоритмической обработки.

8.4.3 Элемент Node SOAP

Информационный объект-элемент Node предназначен предоставить информацию, в каком узле SOAP на пути следования сообщения SOAP произошел отказ (см. раздел 5).

Информационный объект-элемент Node имеет.

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

Тип информационного объекта-элемента Node - xs:anyURI.

Как описано в пункте 5.1, каждый узел SOAP идентифицирован URI. Значение информационного объекта-элемента Node это URI, идентифицирующий узел SOAP, в котором произошел отказ. Узлы SOAP, действующие не как конечный получатель SOAP, ДОЛЖНЫ включать этот информационный объект-элемент. Конечный получатель SOAP МОЖЕТ включить этот информационный объект-элемент, чтобы явно указать, что он сгенерировал отказ.

8.4.4 Элемент Role SOAP

Информационный объект-элемент Role идентифицирует роль, в которой работал узел в точке, где произошел отказ.

Информационный объект-элемент Role имеет:

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

Тип информационного объекта-элемента Role - xs:anyURI.

Значение информационного объекта-элемента Role ДОЛЖНО быть одной из ролей, принятой узлом во время обработки сообщения (см. пункт 5.2).

8.4.5 Элемент Detail SOAP

Информационный объект-элемент Detail предназначен для передачи специализированной информации об ошибке.

Информационный объект-элемент Detail имеет:

- [local name] Detail;
- [namespace name] gost.e-ecolog.ru
- ноль или более информационных объектов атрибутов в своем свойстве [attributes];
- ноль или более дочерних информационных объектов элементов в своем свойстве [children].

У информационного объекта-элемента Detail МОЖЕТ быть любое число дочерних символьных информационных объектов. Символьный код каждого такого символьного информационного объекта ДОЛЖЕН принадлежать множеству пробельных символов как определено в XML 1.0 [XML 1.0]. Они считаются значимыми.

Информационный объект-элемент Detail МОЖЕТ присутствовать в отказе SOAP, и в этом случае с его помощью передается дополнительная информация о коде отказа SOAP, описывающая отказ (см. пункт 8.4.6). Например, информационный элемент Detail мог бы содержать информацию о том, что сообщение содержит ненадлежащие учетные данные, что произошел тайм-аут, и т.д. Наличие информационного объекта-элемента Detail не имеет значения относительно того, какие части некорректного сообщения SOAP были обработаны.

Все дочерние информационные объекты-элементы информационного объекта-элемента Detail называют элементами детализации (см. пункт 8.4.5.1).

8.4.5.1 Элементы детализации SOAP

Каждый элемент детализации:

- МОЖЕТ иметь свойство [namespace name] с непустым значением; т. е. имя элемента МОЖЕТ быть квалифицировано неким пространством имен;
- МОЖЕТ иметь любое число дочерних информационных объектов элементов;
- МОЖЕТ иметь любое число дочерних символьных информационных объектов. Дочерние символьные информационные объекты, символьный код которых принадлежит множеству пробельных символов, как определено в XML 1.0 [XML 1.0], рассматриваются как значимые;
- МОЖЕТ иметь ноль или более информационных объектов атрибутов в своем свойстве [attributes]. Среди них МОЖЕТ быть следующий атрибут со специальным значением для обработки SOAP:

информационный объект-атрибут encodingStyle (см. пункт 8.1.1).

Информационный объект-атрибут encodingStyle SOAP при условии, если он есть, указывает в данном случае на метод кодирования, используемый для элемента детализации (см. пункт 8.1.1).

8.4.6 Коды отказа SOAP

Коды отказа SOAP представляют собой развернутые имена XML и предназначены обеспечить инструмент для классификации отказов. В каждое сообщение об отказе SOAP включен иерархический список кодов SOAP с соответствующей вспомогательной информаций для каждого такого кода, идентифицирующего категорию отказа при увеличении уровня детализации.

Значения дочернего информационного объекта-элемента Value информационного объекта-элемента Code ограничены значениями, определенными типом env:faultCodeEnum (см. таблицу 4). МОГУТ быть созданы дополнительные подкоды отказа для использования приложениями или функциями рас-

ширения. Такие подкоды передаются в дочернем информационном объекте элемента Value информационного объекта-элемента Subcode.

Коды отказа SOAP нужно интерпретировать как модификаторы содержимого информационного объекта-элемента Detail в том смысле, что они предоставляют контекст для информационного объекта-элемента Detail. Для того, чтобы быть в состоянии интерпретировать информационный объект-элемент Detail в отказе SOAP, узел SOAP ДОЛЖЕН понимать все коды отказа SOAP в сообщении отказа SOAP.

Пример – Пример отказа SOAP, в котором информационный объект-элемент Detail должен быть интерпретирован в контексте кодов отказа «env:Sender» и «m:MessageTimeout».

```
<env:Envelope xmlns:env = "gost.e-ecolog.ru                    "
      xmlns:m = "gost.e-ecolog.ru                    "
      xmlns:xml = "gost.e-ecolog.ru                    >
 <env:Body>
  <env:Fault>
    <env:Code>
      <env:Value> env:Sender </env:Value>
        <env:Subcode>
          <env:Value> m:MessageTimeout </env:Value>
        </env:Subcode>
    </env:Code>
    <env:Reason>
      <env:Text xml:lang = "en"> Sender Timeout </env:Text>
    </env:Reason>
    <env:Detail>
      <m:MaxTime> P5M </m:maxtime>
    </env:Detail>
  </env:Fault>
 </env:Body>
</env:Envelope>
```

Настоящая спецификация не ограничивает количество информационных объектов элементов Subcode в отказе SOAP. Но, хотя это и не является требованием данной спецификации, ожидается, что наиболее распространенные практические примеры могут быть ограничены относительно небольшим количеством информационных объектов элементов Subcode.

Таблица 4 — Коды Отказа SOAP

Локальное имя	Значение
VersionMismatch	Отказывающий узел обнаружил недопустимый информационный объект-элемент вместо ожидаемого информационного объекта-элемента Envelope. Пространство имен, локальное имя или оба не соответствуют требованиям данной спецификации для информационного объекта-элемента Envelope (см. пункты 5.8 и 8.4.7)
MustUnderstand	Непосредственный дочерний информационный объект-элемент информационного объекта-элемента Header SOAP, предназначенный для обработки в отказывающем узле и содержащий информационный объект-атрибут mustUnderstand SOAP со значением «true», не был понят отказывающим узлом (см. пункты 8.2.3 и 8.4.8)
DataEncodingUnknown	Кодировка данных содержимого блока заголовка SOAP или дочернего информационного объекта-элемента тела SOAP, предназначенных для обработки в отказывающем узле SOAP (см. пункт 8.1.1), не поддерживается отказывающим узлом
Sender	Сообщение было неправильно сформировано или не содержало надлежащую необходимую информацию. Например, в сообщении может отсутствовать информация, необходимая для аутентификации или подтверждения платежного баланса. В общем случае это означает, что сообщение не должно быть снова послано без изменения (см. также описание подэлемента detail отказа SOAP в пункте 8.4)

Окончание таблицы 4

Локальное имя	Значение
Receiver	Сообщение не могло быть обработано по причинам, относящимся к обработке сообщения, а не к содержанию самого сообщения. Например, обработка могла включать пересылку на последующий узел SOAP, который не отвечал. Такое сообщение может быть успешно доставлено, если оно будет снова послано позже (см. также описание подэлемента detail отказа SOAP в пункте 8.4)

8.4.7 Отказы VersionMismatch

Если узел SOAP генерирует отказ со значением Value информационного объекта-элемента Code, равным «env:VersionMismatch», то в сгенерированном сообщении отказа он ДОЛЖЕН создать блок заголовка SOAP Upgrade. Блок заголовка SOAP Upgrade, как описано ниже, детализирует квалифицированные XML имена (согласно XML-схеме [Часть 2 XML-схемы]) конвертов SOAP, которые поддерживаются данным узлом SOAP (см. пункт 5.8).

8.4.7.1 Блок заголовка Upgrade SOAP

Блок заголовка SOAP Upgrade состоит из информационного объекта-элемента Upgrade, содержащего упорядоченный список квалифицированных XML имен конвертов SOAP, которые узел SOAP поддерживает в порядке от более предпочтительного к менее предпочтительному.

Информационный объект-элемент Upgrade имеет:

- [local name] Upgrade;
- [namespace name] gost.e-ecolog.ru ;
- один или более информационных объектов-элементов SupportedEnvelope в своем свойстве [children] в пункте 8.4.7.2.

У информационного объекта-элемента Upgrade не ДОЛЖНО быть информационного объекта-атрибута encodingStyle.

8.4.7.2 Элемент SupportedEnvelope SOAP

Информационный объект-элемент SupportedEnvelope имеет:

- [local name] SupportedEnvelope;
- [namespace name] gost.e-ecolog.ru ;
- информационный объект-атрибут qname в своем свойстве [attributes] как описано в пункте 8.4.7.3.

8.4.7.3 Атрибут QName SOAP

Информационный объект-атрибут qname имеет следующие свойства инфо-набора XML:

- [local name] qname;
- [namespace name], у которого нет значения;
- свойство [specified] со значением «true».

Тип информационного объекта-атрибута qname - xs:QName. Его значением является квалифицированное XML имя информационного объекта-элемента Envelope SOAP, который может понять отказывающий узел.

Примечание — При сериализации информационного объекта-атрибута qname необходимо, чтобы в области видимости была декларация пространства имен, используемого в имени информационного объекта-элемента Envelope SOAP, который понятен отказывающему узлу. Значение информационного объекта-атрибута использует префикс этого объявления пространства имен.

8.4.7.4 Пример VersionMismatch

Следующий пример демонстрирует случай узла SOAP, который поддерживает и SOAP версии 1.2, и SOAP/1.1, но который предпочитает SOAP версии 1.2 (механизм перехода от SOAP/1.1 к SOAP версии 1.2 описан в приложении А. Переход от версии SOAP/1.1 к SOAP версии 1.2). На это укажет включение блока заголовка Upgrade SOAP с двумя информационными объектами-элементами SupportedEnvelope, первое из которых содержит локальное имя и имя пространства имен информационного объекта-элемента Envelope SOAP версии 1.2, а второй — локальное имя и имя пространства имен информационного объекта-элемента Envelope SOAP/1.1.

Пример — Отказ несоответствия версии сгенерирован узлом SOAP. В сообщение вставлен блок заголовка Upgrade SOAP, показывающий, что поддерживаются как SOAP версии 1.2, так и SOAP/1.1, но с предпочтением SOAP версии 1.2.

```
<? версия xml = "1.0"?>
<env:Envelope xmlns:env = "gost.e-ecolog.ru                                    "
      xmlns:xml = "gost.e-ecolog.ru                                    ">
  <env:Header>
    <env:Upgrade>
      <env:SupportedEnvelope qname = "ns1:Envelope"
            xmlns:ns1 = "gost.e-ecolog.ru                                    >
      <env:SupportedEnvelope qname = "ns2:Envelope"
            xmlns:ns2 = "gost.e-ecolog.ru                                    >
    </env:Upgrade>
  </env:Header>
  <env:Body>
    <env:Fault>
    <env:Code><env:Value>env:VersionMismatch</env:Value></env:Code>
    <env:Reason>
      <env:Text xml:lang = "en"> Version Mismatch </env:Text>
    </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

8.4.8 Отказы mustUnderstand SOAP

Если узел SOAP генерирует отказ со значением Value информационного объекта-элемента Code "env:MustUnderstand", то в сгенерированном сообщении отказа ДОЛЖНЫ присутствовать блоки заголовка NotUnderstood SOAP. Блоки заголовка NotUnderstood SOAP, как описано ниже, детализируют квалифицированные XML имена (соответственно XML-схеме [Часть 2 XML-схемы]) конкретных блока или блоков заголовка SOAP, которые не были поняты.

Узел SOAP МОЖЕТ генерировать отказ SOAP для любого одного или более блоков заголовка SOAP, которые не были поняты в сообщении SOAP. Однако не требуются, чтобы отказ содержал квалифицированные XML имена всех таких блоков заголовка SOAP.

8.4.8.1 SOAP Элемент NotUnderstood

Каждый информационный объект-элемент блока заголовка NotUnderstood имеет:

- [local name] NotUnderstood;
- [namespace name]gost.e-ecolog.ru ;
- qname информационный объект-атрибут в своем свойстве [attributes] как описано в пункте 8.4.8.2.

Информационный объект-элемент NotUnderstood НЕ ДОЛЖЕН иметь информационный объект-атрибут encodingStyle.

8.4.8.2 Атрибут QName SOAP

Информационный объект-атрибут qname имеет следующие свойства инфо-набора XML:

- [local name] qname;
- [namespace name], у которого нет значения;
- свойство [specified] со значением «true».

Тип qname информационного объекта-атрибута - xs:QName. Его значение — квалифицированное XML имя блока заголовка SOAP, который не удалось понять отказывающему узлу.

Примечание — При сериализации *информационного объекта атрибута* qname необходимо, чтобы в области видимости была декларация пространства имен блока заголовка SOAP, который не был понят. При этом значение *информационного объекта атрибута использует* префикс этого объявления пространства имен. Не требуется, чтобы используемый префикс совпадал с префиксом, используемым в сообщении SOAP, которое не было понято.

8.4.8.3 Пример NotUnderstood

Рассмотрим в качестве примера следующее сообщение:

Пример – Конверт SOAP вызовет отказ, если Extension1 или Extension2 не будут поняты:

```
<? версия xml = "1.0"?>
<env:Envelope xmlns:env ='gost.e-ecolog.ru                                    >
```

```
<env:Header>
  <abc:Extension1 xmlns:abc ='gost.e-ecolog.ru
    env:mustUnderstand ='true'/>
  <def:Extension2 xmlns:def ='gost.e-ecolog.ru
    env:mustUnderstand ='true'/>
</env:Header>
<env:Body>
. . .
 </env:Body>
</env:Envelope>
```

В случае, если конечный получатель сообщения SOAP не поймет два блока заголовка SOAP abc:Extension1 и def:Extension2, результатом сообщения в приведенном примере будет сообщение об отказе, показанное далее.

Пример – Отказ SOAP, сгенерированный в результате непонимания Extension1 и Extension2

```
<? версия xml = "1.0"?>
<env:Envelope xmlns:env ='gost.e-ecolog.ru
      xmlns:xml ='gost.e-ecolog.ru                ML/1998/namespace'>
      <env:Header>
  <env:NotUnderstood qname ='abc:Extension1'
        xmlns:abc ='gost.e-ecolog.ru                001/06/ext'/>
  <env:NotUnderstood qname ='def:Extension2'
        xmlns:def ='gost.e-ecolog.ru                stuff'/>
 </env:Header>
 <env:Body>
  <env:Fault>
  <env:Code><env:Value>env:MustUnderstand</env:Value></env:Code>
   <env:Reason>
    <env:Text xml:lang ='en'> One or more mandatory
     SOAP header blocks not understood
    </env:Text>
   </env:Reason>
  </env:Fault>
 </env:Body>
</env:Envelope>
```

9 Использование URI в SOAP

В SOAP URI используются для некоторых идентификаторов включая, но не ограничиваясь, значения информационных объектов-атрибутов encodingStyle (см. пункт 8.1.1), и role (см. пункт 8.2.2). Для SOAP URI — это просто отформатированная строка, которая идентифицирует веб-ресурс.

Там, где данная спецификация ссылается на URI, представленная строка ДОЛЖНА соответствовать синтаксису URI, как описано в RFC 3986 [RFC 3986]. Примечание: RFC 3987 [RFC 3987] предоставляет средства кодирования интернационализированных идентификаторов ресурсов IRI в соответствующие URI.

Несмотря на то, что этот раздел касается только URI, которые непосредственно используются информационными объектами, определенными настоящей спецификацией, РЕКОМЕНДУЕТСЯ, чтобы данные, определенные приложениями, которые переносятся внутри конверта SOAP, использовали те же механизмы и руководящие принципы, определенные данным документом для обработки URI.

Идентификаторы URI, используемые в качестве значений в информационных объектах, идентифицированных посредством пространств имен XML «http://www.w3.org/2003/05/soap-envelope» и «gost.e-ecolog.ru , могут быть как относительными, так и абсолютными.

SOAP не определяет базовый URI, однако опирается на механизмы, определенные в XML Base [Основы XML] и RFC 3986 [RFC 3986] для определения базового URI, основываясь на котором относительный URI может быть сделан абсолютным.

Нижележащий протокол МОЖЕТ определить базовый URI, который может использоваться в качестве базового URI для конверта SOAP (см. раздел 7 и SOAP 1.2 часть 2 [Часть 2 SOAP], раздел Привязка HTTP).

SOAP не определяет никаких правил эквивалентности для URI в целом, так как это определено отдельными схемами URI и RFC 3986 [RFC 3986]. Однако из-за несогласованности относительно правил эквивалентности URI во многих существующих синтаксических анализаторах URI, РЕКОМЕНДУЕТСЯ, чтобы для определения эквивалентности значений URI, используемых в сообщении SOAP, отправители SOAP не полагались на какие-либо специальные правила эквивалентности у получателей SOAP.

Использования IP-адресов в URIs СЛЕДУЕТ избегать всегда, когда это возможно (см. RFC 1900 [RFC 1900]). Однако текстовый формат адресов IPv6 в URI СЛЕДУЕТ поддерживать, как это описано RFC 3986 [RFC 3986].

SOAP не устанавливает априорных ограничений на длину URI. Любой узел SOAP ДОЛЖЕН быть в состоянии обработать передаваемый URI любой длины. Как отправители SOAP, так и получатели SOAP ДОЛЖНЫ оперировать с URI длиной по крайней мере 2048 символов.

10 Соображения безопасности

Структура обмена сообщениями SOAP непосредственно не обеспечивает механизмы, позволяющие контролировать управление доступом, конфиденциальность, целостность и неотказуемость. Подобные механизмы могут быть добавлены как расширения SOAP посредством применения модели расширяемости SOAP (см. раздел 6). В данном разделе описываются соображения безопасности, которые должны быть учтены разработчиками и конструкторами при разработке и эксплуатации подобных механизмов.

Разработчики SOAP должны учитывать возможность появления вредоносных приложений SOAP, намеренно отправляющих вредоносные данные на узел SOAP (см. раздел 5). Настоятельно рекомендуется, чтобы узел SOAP, получающий сообщение SOAP, был способен оценить, насколько он может доверять отправителю сообщения SOAP и содержимому этого сообщения.

10.1 Узлы SOAP

SOAP позволяет пересылать данные приложений как в блоках заголовка SOAP, так и в содержимом тела SOAP. Обработка блока заголовка SOAP может включать в себя побочные эффекты, такие как изменения состояния, ведение журнала или формирование дополнительных сообщений. Для любого сценария развертывания настоятельно рекомендуется, чтобы узлом SOAP обрабатывались только строго специфицированные блоки заголовка SOAP с хорошо понятыми последствиями любых побочных эффектов для безопасности.

Точно так же обработка тела SOAP может вызвать возникновение побочных эффектов, которые могут приводить к серьезным последствиям для принимающего узла SOAP в случае, если тело понято не должным образом. Настоятельно рекомендуется, чтобы обрабатывалось только строго определенное содержимое тела с известными последствиями для безопасности.

Соображения безопасности, однако, не должны ограничиваться распознаванием только непосредственных дочерних элементов блока заголовка SOAP и тела SOAP. Разработчики должны обратить особое внимание на последствия для безопасности всех данных, передаваемых в сообщении SOAP, которое может вызвать удаленное выполнение каких-либо действий в среде получателя. К этому относятся не только данные, выраженные как свойства инфо-набора XML, но и данные, которые могут быть закодированы как значения свойств, включая двоичные данные или параметры, например строки запроса URI. Прежде чем принять данные любого типа, приложение должно знать о конкретных последствиях для безопасности, связанных с этими данными в контексте их использования.

Если обработка различных частей сообщения SOAP обеспечивается посредством программного обеспечения с модульной архитектурой, то разработчики SOAP должны принять все возможные меры для того, чтобы обеспечить каждый модуль полной информацией о контексте безопасности. Например, без знания контекста, в котором оно получено, тело SOAP не должно обрабатываться.

10.2 Посредники SOAP

По своей сущности, SOAP обеспечивает модель распределенной разработки, которая может включать прохождение сообщения SOAP через несколько узлов SOAP (см. раздел 5). Посредники SOAP — это по определению узлы в середине, представляющие возможность для атак типа «человек посередине» (man-in-the-middle). Нарушения правил безопасности в системах, которые действуют как посредники SOAP,

могут привести к серьезным проблемам безопасности и конфиденциальности. Посредник SOAP с дефектами безопасности или посредник, реализованный или сконфигурированный без учета соображений безопасности и конфиденциальности, может быть использован в потенциальных атаках широкого диапазона.

При анализе последствий безопасности, связанных с SOAP потенциальных проблем безопасности, важно понять, что сфера действия механизмов безопасности, обеспеченных нижележащим протоколом, может отличаться от сферы действия для всего пути следования сообщения SOAP. SOAP не требует, чтобы все транзитные участки между принимающими участие в пересылке узлами SOAP использовали один и тот же нижележащий протокол. Даже если бы это имело место, то использование посредников SOAP само по себе, вероятно, выходит за рамки безопасности транспортного уровня.

10.3 Привязка нижележащего протокола

Влияние на безопасность того, что не были реализованы пункты спецификации, отмеченные «ДОЛЖЕН» или «СЛЕДУЕТ», или того, что были реализованы пункты «НЕ ДОЛЖЕН» или «НЕ СЛЕДУЕТ», может быть очень тонким. Авторы спецификации привязки должны подробно описать последствия для безопасности невыполнения рекомендаций или требований, поскольку большинство разработчиков не будет иметь того опыта разработки и обсуждений, который накапливается в процессе создания спецификации (см. раздел 7).

Кроме того, в спецификации привязки невозможно рассмотреть или обеспечить контрмеры для всех аспектов, свойственных угрозам безопасности. Авторы спецификации привязки должны идентифицировать любые такие риски, которые могут остаться, и указать, какие дополнительные контрмеры будут необходимы помимо тех, которые предусмотрены в спецификации привязки.

Авторы спецификаций привязки должны понимать, что модули расширения SOAP, оформленные в виде блоков заголовка SOAP, могут повлиять на нижележащий протокол непредвиденными способами. Передача сообщения SOAP через привязку к определенному протоколу может привести к возможному конфликту функций. Пример этого — сообщение SOAP, передаваемое через HTTP с использованием механизма стандартной аутентификации HTTP в сочетании с механизмом аутентификации, основанным на SOAP. Настоятельно рекомендуется, чтобы в спецификации привязки были бы описаны все подобные взаимодействия между расширениями и нижележащими протоколами.

10.3.1 Привязка к протоколам, предназначенным для определенных приложений

Некоторые нижележащие протоколы могут быть спроектированы специально для отдельных приложений или класса приложений. Привязка SOAP к таким протоколам МОЖЕТ использовать ту же идентификацию конечной точки (например, номер порта TCP), что и нижележащий протокол, чтобы использовать в этом случае существующую инфраструктуру, связанную с нижележащим протоколом.

Однако использование закрепленных номеров портов в SOAP может вызвать дополнительную непреднамеренную обработку посредниками и нижележащими реализациями. Например, HTTP обычно считается протоколом «Просмотра веб-страниц», и администраторы сети могут определенным образом ограничить его использование или могут добавить такие сервисы, как фильтрация, изменение контента, маршрутизация, и т.д. Зачастую эти службы добавлены с использованием эвристического номера порта.

В результате в определении привязки к нижележащим протоколам с фиксированными номерами портов или профилями приложений СЛЕДУЕТ документировать потенциальные взаимодействия с типичной инфраструктурой, которая по умолчанию использует эти значения портов по умолчанию или эти профили приложений. В определении привязки СЛЕДУЕТ также иллюстрировать использование привязки на портах не по умолчанию как средство избежать непреднамеренного взаимодействия с такими службами.

Приложение А
(справочное)

Переход от версии SOAP/1.1 к SOAP версии 1.2

Это приложение описывает правила управления версией для узла SOAP. Если узел SOAP поддерживает управление версиями от SOAP 1.1 до SOAP 1.2, то узел SOAP ДОЛЖЕН реализовать правила, описанные в настоящем приложении.

Правила для возможной совместной обработки сообщений SOAP/1.1 и SOAP версии 1.2:

1. Узел SOAP/1.1, получающий сообщение SOAP версии 1.2, будет, согласно SOAP/1.1, генерировать отказ SOAP «несоответствие версии» на основе логической структуры сообщения SOAP/1.1. Т.е. конверт будет иметь [local name] Envelope и [namespace name] "http://schemas.xmlsoap.org/soap/envelope/".
2. Для узла SOAP версии 1.2, получающего сообщение SOAP/1.1, возможны варианты:

- МОЖЕТ обработать сообщение как сообщение SOAP/1.1 (если эта версия узлом поддерживается), или
- ДОЛЖЕН сгенерировать отказ SOAP «несоответствие версии» на основе логической структуры сообщения SOAP/1.1, следуя семантике SOAP/1.1, используя привязку SOAP/1.1 к нижележащему протоколу (см. SOAP 1.1 [SOAP 1.1]). Отказ SOAP ДОЛЖЕН включать блок заголовка SOAP Upgrade, в котором должно быть указание на поддержку SOAP версии 1.2, как это определено в данной спецификации (см. пункт 8.4.7). Это позволяет узлу-получателю SOAP/1.1 правильно интерпретировать отказ SOAP, сгенерированный узлом SOAP версии 1.2.

Следующий пример демонстрирует отказ «несоответствие версии» SOAP, сгенерированный узлом SOAP версии 1.2 в результате получения сообщения SOAP/1.1. Сообщение об отказе представляет собой сообщение SOAP/1.1 с блоком заголовка Upgrade SOAP, указывающим на поддержку SOAP версии 1.2.

Пример – Узел SOAP версии 1.2, генерирующий сообщение отказа «несоответствие версии» SOAP/1.1, включая блок заголовка Upgrade SOAP, указывающий на поддержку SOAP версии 1.2.

```
<? версия xml = «1.0»?>
<env:Envelope xmlns:env = "gost.e-ecolog.ru                    >
  <env:Header>
    <env:Upgrade>
      <env:SupportedEnvelope qname = "ns1:Envelope"
            xmlns:ns1 = "gost.e-ecolog.ru                    >
    </env:Upgrade>
  </env:Header>
  <env:Body>
    <env:Fault>
      <faultcode>env:VersionMismatch</faultcode>
      <faultstring> Version Mismatch </faultstring>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Примечания

1 Для того чтобы узлы SOAP поддерживали и SOAP/1.1, и SOAP версии 1.2, необходимо использовать привязку протокола, связанную с надлежащей версией SOAP.

2 Нежелательно, чтобы отдельный узел SOAP/1.1, генерирующий отказ несоответствия версии SOAP, указывал, какие версии он поддерживает с использованием *информационного объекта-элемента* Upgrade (см. пункт 8.4.7). Если ничто не указано, то это означает, что SOAP/1.1 — единственная поддерживаемая версия. Следует отметить, однако, что несоместимость между привязками нижележащего протокола могла бы препятствовать тому, чтобы узел SOAP/1.1, получая сообщение SOAP версии 1.2, генерировал бы отказ несоответствия версии SOAP. Например, узел SOAP/1.1, поддерживающий HTTP-привязку SOAP/1.1 (см. SOAP 1.1 [SOAP 1.1]), получив сообщение SOAP версии 1.2 с использованием привязки протокола HTTP SOAP 1.2 (см. SOAP 1.2 Части 2 [SOAP Часть 2], Привязка HTTP SOAP), не смог бы понять различие между этими двумя привязками и в результате сгенерировал бы определенный ответ HTTP.

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 40220:2011	—	*
* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.		

Библиография

[1] SOAP Version 1.2 Part 0: Primer (Second Edition) [SOAP Версия 1.2 Часть 0: Учебник для начинающих (вторая редакция). Редакторы: Nilo Mitra, Ив Лафон, Консорциум World Wide Web, 27 апреля 2007 г.

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

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

[4] XML Protocol Charter [Чартер XMLP] Чартер протокола XML (см. gost.e-ecolog.ru gost.e-ecolog.ru

[5] XML Protocol (XMLP) Requirements [Требования XMLP] Протокол XML (XMLP). требования. Редакторы: Видур Аппарао, Алекс Сепонкус, Пол Коттон, Дэвид Эзелл, Дэвид Фаллсайд, Мартин Гадджин, Оизин Херли, Джон Ибботсон, Р. Александр Миловский, Кевин Митчелл, Жан-Жак Моро, Эрик Ньюкомер, Хенрик Фристик Нильсен, Марк Ноттингем, Вакар Садик, Стюарт Уильямс, Амр Яссин, Консорциум «World Wide Web», 28 июля 2003 г.

[6] SOAP Version 1.2 Usage Scenarios [Сценарии использования SOAP] Сценарии Использования SOAP версии 1.2. Редактор: Джон Ибботсон. Консорциум «World Wide Web», 30 июля 2003 г.

[7] [SOAP 1.1] Простой протокол доступа к объектам [Simple Object Access Protocol (SOAP) 1.1]. Редакторы: Поле Дона, Дэвид Энебаск, Гопал Какивая, Эндрю Леймен, Ноа Мендельсон, Хенрик Нильсен, Сатиш Тэйтт, Дэйв Винер, DevelopMentor, IBM, Microsoft, Lotus Development Corp., UserLand Software, Inc., 30 июля 2003 г.

[8] Internationalized Resource Identifiers (IRIs) [RFC 3987]. Интернационализированные идентификаторы ресурса (IRIs). Редактор: М. Дуерст, IETF, январь 2005 г.

[9] [RFC 1900] Renumbering Needs Work. Редакторы: Б. Карпентер, И. Рехтер, IETF, февраль 1996 г.

УДК 004.057.4:006.354 ОКС 35.080

Ключевые слова: информационные технологии, системы обмена сообщениями, протоколы SOAP

Редактор *В.Р. Волменских*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Д. Дульнева*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 07.07.2014 Подписано в печать 28.08.2014 Формат 60×84½. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 3,60. Тираж 43 экз. Зак. 2785.

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

gost.e-ecolog.ru