ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 29361— 2012

Информационная технология

ИНТЕРОПЕРАБЕЛЬНОСТЬ СЕТЕВЫХ УСЛУГ

Базовый профиль WS-I. Версия 1.1

ISO/IEC 29361:2008
Information technology — Web Services Interoperability — WS-I Basic Profile. Version 1.1
(IDT)

Издание официальное

Москва
Стандартинформ
2014

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным унитарным предприятием Государственный научно-исследовательский и конструкторско-технологический институт «ТЕСТ» (ФГУП ГосНИИ «ТЕСТ») на основе собственного аутентичного перевода на русский язык указанного в пункте 4 международного стандарта

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 24 сентября 2012 г. № 421-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 29361:2008 «Информационная технология. Интероперабельность сетевых услуг. Базовый профиль WS-I. Версия 1.1» (ISO/IEC 29361:2008 «Information technology — Web Services Interoperability — WS-I Basic Profile. Version 1.1»)

5 ВВЕДЕН ВПЕРВЫЕ

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

Содержание

Введение

В разделе 1 определен Профиль и объяснены его взаимосвязи с другими профилями.

В разделе 2 «Соответствие Профилю» объяснено, что значит быть соответствующим профилю.

В каждом из последующих разделов рассмотрены компоненты Профиля. Подразделы состоят из двух частей: обзора, детализирующего спецификации компонентов, и точек их расширения, с последующими подразделами, в которых рассмотрены отдельные части компонентов спецификаций. Между номерами разделов настоящего стандарта и ссылочных спецификаций нет связи.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ИНТЕРОПЕРАБЕЛЬНОСТЬ СЕТЕВЫХ УСЛУГ

Базовый профиль WS-I. Версия 1.1

Information technology. Web services interoperability. WS-I basic profile. Version 1.1

Дата введения — 2014—01—01

1 Область применения

1.1 Область применения

В настоящем стандарте определен Базовый профиль WS-I 1.1 (далее "Профиль"), состоящий из набора непроприентарных спецификаций сетевых услуг вместе с пояснениями, уточнениями, интерпретациями и расширениями этих спецификаций, которые обеспечивают интероперабельность.

1.2 Взаимосвязи с другими профилями

Настоящий Профиль получен из Базового Профиля 1.0 путем включения всех поправок и отделения требований, относящихся к сериализации оболочки и ее представления в сообщениях. Такие требования теперь являются частью Простого Профиля привязки SOAP 1.0 и идентифицируются в отдельном заявлении о соответствии.

Данное разделение сделано для облегчения композиции Базового Профиля 1.1 с любым профилем, специфицирующим сериализацию оболочки, включая Простой Профиль привязки SOAP 1.0 и Профиль Присоединений 1.0. Комбинация заявлений о соответствии Базовому Профилю 1.1 и Простому Профилю привязки SOAP 1.0 приблизительно эквивалентна заявлению о соответствии Базовому Профилю 1.0 со всеми дополнениями.

Настоящий профиль в комбинации с Простым Профилем привязки SOAP 1.0 перекрывает Базовый Профиль 1.0. Профиль Присоединений 1.0 добавляет поддержку присоединений в SOAP и предназначен для использования совместно с настоящим Профилем.

1.3 Изменения по сравнению с Базовым Профилем версии 1.0

Настоящая спецификация получена из Базового Профиля версии 1.0[1)] и включает в себя опубликованные поправки к этой спецификации. Наиболее существенными изменениями являются:

- Цель соответствия СООБЩЕНИЕ — Некоторые требования, которые имели целью соответствия СООБЩЕНИЕ, теперь в БП1.0 имеют новую цель, ОБОЛОЧКА. Тем самым упрощаются альтернативные сериализации сообщений такие, как описанные в Профиле Присоединений.
- Привязка SOAP — Требования, относящиеся к привязкам SOAP сериализации сообщения, были перемещены в Простой Профиль привязки SOAP для упрощения других сериализаций.

1.4 Руководящие принципы

Профиль разработан в соответствии с набором принципов, которые вместе образуют идеологию Профиля в отношении его вклада в интероперабельность. В настоящем подразделе задокументированы эти руководящие принципы.

Нет гарантии интероперабельности

[1)] http://ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.htm.

Издание официальное

Невозможно полностью гарантировать интероперабельность конкретной услуги. Однако, Профиль направлен на наиболее общие проблемы, с которыми сталкивался опыт реализации до настоящего времени.

Прикладные семантики

Хотя передача прикладных семантик может быть упрощена технологиями, образующими Профиль, Профиль не нацелен на обеспечение общего понимания этих семантик.

Тестируемость

По мере возможности Профиль содержит утверждения, которые являются тестируемыми. Однако, такая тестируемость не является обязательной. Предпочтительно, чтобы тестируемость достигалась не навязываемым способом (например, тестированием артефактов "на проводе").

Сила требований

По мере возможности Профиль содержит сильные требования (например, ДОЛЖЕН, НЕ ДОЛЖЕН); в законных случаях, когда такие требования не могут быть установлены, использованы условные требования (например, СЛЕДУЕТ, НЕ СЛЕДУЕТ). Факультативные и условные требования вносят двусмысленность и несогласованность между реализациями.

Ограничения или ослабления

При применении требований ссылочных спецификаций Профиль может усиливать их, но не ослаблять (например, изменять ДОЛЖЕН на МОЖЕТ).

Несколько методов

Когда ссылочная спецификация допускает использование нескольких взаимозаменяемых методов, в Профиле выбран тот, который является хорошо понятным, широко реализованным и полезным. Сторонние или неспецифицированные методы и расширения привносят сложность и, следовательно, снижают интероперабельность.

Будущая совместимость

По мере возможности в требованиях Профиля учитывались разрабатываемые пересмотры ссылочных спецификаций. Это помогает реализаторам, допуская плавный переход и обеспечивая, чтобы WS-I не «разветвлялась» в результате этих усилий. Когда Профиль не мог решить проблему в ссылочной спецификации, эта информация передавалась соответствующему органу для обеспечения ее последующего рассмотрения.

Совместимость с задействованными услугами

Обратная совместимость с задействованными сетевыми услугами не является целью Профиля, но ее необходимое рассмотрение проведено; Профиль не вносит изменений в требования ссылочных спецификаций, если только они не направлены на конкретные вопросы интероперабельности.

Нацеленность на интероперабельность

Хотя потенциально в ссылочных спецификациях имеется ряд несогласованностей и недостатков проектов, Профиль учитывает только те из них, которые влияют на интероперабельность.

Цели соответствия

По мере возможности в Профиле установлены требования к артефактам (например, описаниям WSDL, сообщениям SOAP), а не к созданию или использованию поведения или ролей программного обеспечения. Артефакты являются конкретными, что делает их простыми для верификации и, следовательно, делает соответствие простым для понимания и менее склонным к ошибкам.

Интероперабельность нижележащего уровня

В Профиле идет речь об интероперабельности прикладного уровня; принято, что интероперабельность протоколов нижележащих уровней (например, TCP, IP, Ethernet) является адекватной и хорошо понятной. Аналогично, утверждения о составляющих протоколах прикладного уровня (например,

SSL/TLS, HTTP) сделаны только в тех случаях, когда имеется конкретное влияние на сетевые услуги; WS-I не пытается обеспечить интероперабельность этих протоколов в целом. Тем самым обеспечивается эффективное использование экспертизы WS-I, сфокусированной на стандартах сетевых услуг.

1.5 Соглашения об обозначениях

В настоящем стандарте ключевые слова «ДОЛЖЕН», «НЕ ДОЛЖЕН» («MUST», «MUST NOT»), «ТРЕБУЕМЫЙ» («REQUIRED»), «НУЖНО», «НЕ НУЖНО» («SHALL», «SHALL NOT»), «СЛЕДУЕТ», «НЕ СЛЕДУЕТ» («SHOULD», «SHOULD NOT»), «РЕКОМЕНДУЕМЫЙ» («RECOMMENDED»), «МОЖЕТ» («MAY») и «ФАКУЛЬТАТИВНЫЙ» («OPTIONAL») должны пониматься в соответствии с RFC 2119[1].

Нормативные утверждения требований в Профиле (т. е. те, которые влияют на соответствие, как определено в «Требованиях соответствия») представлены следующим образом:

Rnnnn*Текст утверждения*,

где «nnnn» заменяется уникальным в Профиле номером требования, образующим уникальный идентификатор требования.

Идентификаторы требований могут рассматриваться как квалификаторы пространства имен так, чтобы быть совместимыми с QNames в соответствии с Пространствами имен в XML[2]. Когда явный префикс пространства имен в идентификаторе требования отсутствует (например, «R9999» вместо «bp10:R9999»), его следует интерпретировать как находящийся в пространстве имен, идентифицированном соответствующим URI раздела документа, в котором он встретился. Когда идентификатор квалифицирован, префикс следует интерпретировать в соответствии с действующими отображениями пространств имен так, как описано ниже.

Некоторые требования поясняют ссылочные спецификации, но не устанавливают дополнительные ограничения на реализации. Для удобства пояснения отмечены следующим образом: C

Некоторые требования получены на основе незавершенных работ по стандартизации ссылочных спецификаций. Для удобства такие полученные с опережением утверждения отмечены следующим образом: xxxx, где «xxxx» — идентификатор спецификации (например, «WSDL20» для WSDL, Версия 2.0). Поскольку на момент публикации настоящего стандарта указанные работы не завершены, спецификация, из которой получено требование, может измениться; данная информация включена только для удобства реализаторов.

Точки расширения в поддерживающих спецификациях (см. раздел «Область соответствия») представлены аналогично:

Ennnn*Имя точки расширения — Описание*,

где "nnnn" заменяется уникальным для точек расширения в Профиле номером. Как и утверждения требований, утверждения расширения можно рассматривать как квалификаторы пространства имен.

В настоящем стандарте использовано несколько префиксов пространств имен; ниже перечислены соответствующие им URI. Выбор какого-либо префикса пространства имен является произвольным и не имеет семантического значения.

- **soap** — "http://schemas.xmlsoap.org/soap/envelope/"
- **xsi** — "http://www.w3.org/2001/XMLSchema-instance"
- **xsd** — "http://www.w3.org/2001/XMLSchema"
- **soapenc** — "http://schemas.xmlsoap.org/soap/encoding/"
- **wsdl** — "http://schemas.xmlsoap.org/wsdl/"
- **soapbind** — "http://schemas.xmlsoap.org/wsdl/soap/"
- **uddi** — "urn:uddi-org:api_v2"

1.6 Идентификация профиля и версии

Настоящий стандарт идентифицирован именем (Базовый Профиль) и номером версии (1.1). Вместе они идентифицируют конкретный *экземпляр профиля*.

Номер версии состоит из старшей и младшей части в виде «старшая.младшая». Он может быть использован для определения предшествования экземпляров профиля; больший номер версии (состоящий из старшей и младшей частей) указывает на то, что экземпляр более новый и, следовательно, заменяет более ранние.

---

[1] http://www.ietf.org/rfc/rfc2119.txt.

[2] http://www.w3.org/TR/REC-xml-names/.

Экземпляры профиля с одним и тем же именем (например, «Пример Профиля 1.1» и «Пример Профиля 5.0») направлены на проблемы интероперабельности в одной и той же области (хотя развитие может потребовать изменения от экземпляра к экземпляру точной области применения профиля).

Указанную информацию можно использовать для определения обратной совместимости экземпляров профиля, а именно, определения того, можно ли считать, что соответствие более раннему экземпляру профиля подразумевает соответствие более позднему экземпляру. Экземпляры профиля с одной и той же старшей частью номера версии (например, «Пример Профиля 1.0» и «Пример Профиля 1.1») МОГУТ рассматриваться как совместимые. Это не подразумевает совместимость в каком-либо другом отношении, а именно, нельзя считать, что соответствие более позднему экземпляру профиля подразумевает соответствие более раннему.

2 Соответствие Профилю

Соответствие Профилю определяется соблюдением набора *требований*, установленных для конкретной *цели* в пределах *области применения* Профиля. В настоящем разделе разъяснены эти термины и описано, как соответствие определяется и используется.

2.1 Требования соответствия

Требования устанавливают критерии соответствия Профилю. Обычно они относятся к существующей спецификации и включают в себя ее уточнения, усиления, интерпретацию и разъяснения с целью улучшения интероперабельности. Все требования в Профиле считаются нормативными и все требования, входящие в область применения Профиля (см. раздел «Область применения соответствия»), в спецификациях, на которые он ссылается, также следует считать нормативными. Когда требования в Профиле и спецификациях, на которые он ссылается, противоречат друг другу, требования Профиля имеют предпочтение для целей соответствия Профилю.

Уровни требований, выраженные с помощью терминов RFC 2119 (например, ДОЛЖЕН, МОЖЕТ, СЛЕДУЕТ), указывают характер требований и их влияние на соответствие. Для удобства каждое требование идентифицировано (например, R9999).

Например:

R9999 *ВИДЖЕТу СЛЕДУЕТ иметь округлую форму.*

Это требование идентифицировано как «R9999», применяется к цели (см. ниже) ВИДЖЕТ, и устанавливает условное требование на виджеты; т.е. хотя это требование должно выполняться для обеспечения соответствия в большинстве случаев, но есть ситуации, когда могут быть допустимые основания для его невыполнения (что объясняется в самом требовании или в сопутствующем тексте).

Каждое требование соответствия содержит ровно одно ключевое слово уровня требований (например, «ДОЛЖЕН») и одно ключевое слово цели соответствия (например, «СООБЩЕНИЕ»). Ключевое слово цели соответствия приводится жирным шрифтом (например, «**СООБЩЕНИЕ**»). Прочие цели соответствия, набранные светлым шрифтом, используются только для определения, но НЕ как цель соответствия. Для пояснения требования или группы требований может быть включен дополнительный текст (например, обоснование и примеры); однако текст, сопутствующий утверждениям требования, не должен учитываться при рассмотрении соответствия.

Определения терминов в Профиле считаются действительными для целей определения соответствия.

Никакие требования в Профиле независимо от их уровня соответствия не следует интерпретировать как ограничивающие возможности соответствующей реализации в применении контрмер безопасности в ответ на реальные или предполагаемые угрозы (например, атака «отказ в обслуживании»).

2.2 Цели соответствия

Цели соответствия идентифицируют те артефакты (например, сообщение SOAP, описание WSDL, регистрационные данные UDDI) или стороны (например, процессор SOAP, конечный пользователь), к которым применяют требования.

Сказанное позволяет определению соответствия в разных контекстах гарантировать недвусмысленную интерпретацию применимости требований и допускает тестирование соответствия артефактов (например, сообщений SOAP и описаний WSDL) и поведения различных сторон сетевых услуг (например, реализаций клиентов и услуг).

Для упрощения тестирования и во избежание недоразумений цели соответствия требований являются, по мере возможности, физическими артефактами.

В Профиле использованы следующие цели соответствия:

- **СООБЩЕНИЕ** — элемент протокола, который передает ОБОЛОЧКА (например, сообщения SOAP/HTTP);
- **ОБОЛОЧКА** — сериализация элемента soap:Envelope и его содержимого;
- **ОПИСАНИЕ** — описание типа, сообщения, интерфейса и их привязки к конкретному протоколу и формату данных, а также точек доступа к сети, ассоциированных с сетевой услугой (например, описания WSDL) (по Базовому Профилю 1.0);
- **ЭКЗЕМПЛЯР** — программное обеспечение, реализующее wsdl:port или uddi:bindingTemplate (по Базовому Профилю 1.0);
- **ПОТРЕБИТЕЛЬ** — программное обеспечение, вызывающее ЭКЗЕМПЛЯР (по Базовому Профилю 1.0);
- **ОТПРАВИТЕЛЬ** — программное обеспечение, генерирующее сообщение в соответствии с ассоциированным(и) с ним проколом(ами) (по Базовому Профилю 1.0);
- **ПОЛУЧАТЕЛЬ** — программное обеспечение, которое пользуется сообщением в соответствии с ассоциированным(и) с ним проколом(ами) (например, процессор SOAP) (по Базовому Профилю 1.0);
- **REGDATA** — элемент регистра, который используется при регистрации и обнаружении сетевых услуг (например, UDDI tModel) (по Базовому Профилю 1.0).

2.3 Область соответствия

Область применения Профиля описывает технологии, на которые он направлен; другими словами, Профиль только пытается улучшить интероперабельность в пределах своей области применения. В общем случае область применения Профиля ограничена спецификациями, на которые он ссылается.

Область применения Профиля уточняется в точках расширения. Часто в ссылочных спецификациях предоставлены методы расширения и неспецифицированные или открытые параметры конфигурации; когда такой метод или параметр идентифицирован в Профиле как точка расширения, тогда он находится вне области применения Профиля и его использование или неиспользование не влияет на соответствие.

В Профиле также могут быть установлены требования к использованию точки расширения. Для улучшения интероперабельности конкретное использование точек расширения может быть также ограничено другими профилями при их применении совместно с настоящим Профилем.

Поскольку использование точек расширения может испортить интероперабельность, их применение для сетевой услуги должно быть некоторым образом согласовано или задокументировано сторонами; например, это может иметь вид стороннего соглашения.

Область применения Профиля определена ссылочными спецификациями в приложении А и уточнена точками расширения в приложении В.

2.4 Заявление о соответствии

Заявление о соответствии Профилю может быть сделано с использованием следующих методов, описанных в Методах присоединения заявления о соответствии[1)], когда удовлетворены требования применяемого Профиля, ассоциированные с перечисленными целями:

- **Метод присоединения заявления WSDL 1.1 для экземпляров сетевых услуг** — СООБЩЕНИЕ, ОПИСАНИЕ, ЭКЗЕМПЛЯР, ПОЛУЧАТЕЛЬ;
- **Метод присоединения заявления WSDL 1.1 для описаний конструкций** — ОПИСАНИЕ;
- **Метод присоединения заявления UDID для экземпляров сетевых услуг** — СООБЩЕНИЕ, ОПИСАНИЕ, ЭКЗЕМПЛЯР, ПОЛУЧАТЕЛЬ;
- **Метод присоединения заявления UDDI для регистрации сетевых услуг** — REGDATA.

URI заявления о соответствии настоящему Профилю является «http://ws-i.org/profiles/basic/1.1».

3 Сообщения

Настоящий раздел Профиля включает в себя по ссылке следующие спецификации и определяет точки расширения в них:

[1)] http://www.ws-i.org/Profiles/ConformanceClaims-1.0html.

• Простой протокол доступа к объекту [Simple Object Access Protocol (SOAP)] 1.1[1].

Точки расширения:

• E0001 — Блоки заголовков — Блоки заголовков являются главными методами расширения в SOAP.

• E0002 — Порядок обработки — Порядок обработки компонентов оболочки SOAP (например, заголовков) не специфицирован и, следовательно, должен быть согласован дополнительно.

• E0003 — Использование посредников — Посредники SOAP являются не специфицированным в SOAP 1.1 методом, и их использование может потребовать дополнительного согласования. Их использование также может вызвать необходимость тщательного рассмотрения при оценке соответствия Профилю.

• E0004 — Значения soap:actor — Значения атрибута soap:actor, отличные от специального uri «http://schemas.xmlsoap.org/soap/actor/next», представляют собой частное соглашение между сторонами сетевой услуги.

• E0005 — Подробности отказа — Содержимое элемента Подробности отказа не установлены в SOAP 1.1.

• E0006 — Сериализация оболочки — Профиль не ограничивает некоторые аспекты представления оболочки в виде сообщения.

• RFC2616: Протокол передачи гипертекста (RFC2616: Hypertext Transfer Protocol — HTTP/1.1)[2]

Точки расширения:

• E0007 — Аутентификация HTTP — Аутентификация HTTP допускает схемы расширения, произвольные цифровые хэш-алгоритмы и параметры.

• E0008 — Неспецифицированные поля заголовка — HTTP допускает появление в сообщениях произвольных заголовков.

• E0009 — Расширения ожидания — Механизм Ждать/Продолжить в HTTP допускает расширения ожидания.

• E0010 — Кодирование содержимого — Множество кодирований содержимого, допускаемых HTTP, является открытым и любое кодирование, кроме «gzip», «compress» или «deflate», является точкой расширения.

• E0011 — Кодирование передачи — Множество кодирований передачи, допускаемых HTTP, является открытым.

• E0012 — Обновление — HTTP позволяет изменять соединение на произвольный протокол, используя заголовок Upgrade (обновить).

• E0024 — Атрибуты пространства имен — Атрибуты пространства имен элементов soap:Envelope и soap:Header.

• E0025 — Атрибуты элементов soap:Body — SOAP 1.1 не устанавливает ограничений ни на пространство имен, ни на локальные атрибуты.

• RFC2965: Метод управления состоянием HTTP (RFC2965: HTTP State Management Mechanism)[3].

3.1 Оболочка SOAP

В настоящем разделе Профиля даны ссылки на следующие спецификации (или их разделы):

• SOAP 1.1, раздел 4.

SOAP 1.1 определяет структуру для составных сообщений, оболочки. В Профиле установлена обязательность использования этой структуры и следующие ограничения на ее использование:

3.1.1 Структура оболочки SOAP

R9980 **ОБОЛОЧКА** *ДОЛЖНА соответствовать структуре, определенной в SOAP 1.1, раздел 4, «Оболочка SOAP» (вопрос для исправления Профилем).*

R9981 **ОБОЛОЧКА** *ДОЛЖНА иметь ровно один дочерний элемент элемента soap:Body или не иметь их совсем.*

Хотя комбинация R2201 и R2210 (см. ниже) подразумевает, что у soap:Body может быть не более одного дочернего элемента, в Профиле нет явного требования, озвучивающего это ограничение, что приводит к некоторой путанице.

3.1.2 Пространство имен оболочки SOAP

В SOAP 1.1 установлено, что оболочку с элементом документа, пространство имен которого отлично от «http://schemas.xmlsoap.org/soap/envelope/», следует отбросить. Для обеспечения недвусмысленности операции в Профиле требуется, чтобы вместо этого был сгенерирован отказ.

---

[1] http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

[2] http://www.ietf.org/rfc/rfc2616.txt.

[3] http://www.ietf.org/rfc/rfc2965.txt.

R1015 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН сгенерировать отказ, если ему встретилась оболочка, элемент документа которой не является soap:Envelope.*

3.1.3 Квалификация пространства имен тела SOAP

Использование неквалифицированных имен элементов может вызвать конфликты имен, следовательно, для потомков soap:Body должны использоваться квалифицированные имена.

R1014 *Потомки элемента soap:Body в* **ОБОЛОЧКЕ** *ДОЛЖНЫ быть квалифицированным пространством имен.*

3.1.4 Запрещенные конструкции

ДТД (декларация типа документа) и ИО (инструкция обработки) XML при их использовании в оболочках могут приводить к уязвимостям безопасности, избыточной обработке и семантической двусмысленности. Поэтому некоторые конструкции XML запрещены разделом 3 SOAP 1.1.

Хотя опубликованная поправка NE05 (см. http://www.w3.org/XML/xml-names-19990114-errata) допускает появление декларации пространства имен xmlns:xml="http://www.w3.org/XML/1998/namespace", некоторые старые процессоры рассматривают такую декларацию как ошибку. Следующие требования обеспечивают, чтобы соответствующие артефакты имели наиболее широко возможную интероперабельность.

R1008 **ОБОЛОЧКА** *НЕ ДОЛЖНА содержать декларацию типа документа.* С

R1009 **ОБОЛОЧКА** *НЕ ДОЛЖНА содержать инструкции обработки.* С

R1033 **ОБОЛОЧКЕ** *НЕ СЛЕДУЕТ содержать декларацию пространства имен xmlns:xml="http://www.w3.org/XML/1998/namespace".* С

R1034 **ОПИСАНИЮ** *НЕ СЛЕДУЕТ содержать декларацию пространства имен xmlns:xml="http://www.w3.org/XML/1998/namespace".* С

3.1.5 Приложения SOAP

Интерпретация элементов одного уровня, следующих за элементом soap:Body не ясна. Следовательно, такие элементы запрещены.

R1011 **ОБОЛОЧКА** *НЕ ДОЛЖНА иметь дочерних элементов soap:Envelope, следующих за элементом soap:Body.*

Настоящее требование уточняет расхождение между спецификацией SOAP 1.1 и Схемой XML SOAP 1.1.

Например,

НЕПРАВИЛЬНО:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Body>
       <p:Process xmlns:p='http://example.org/Operations' />
  </soap:Body>
  <m:Data xmlns:m='http://example.org/information' >
      Здесь некоторые данные с сообщением.
  </m:Data>
</soap:Envelope>
```

ПРАВИЛЬНО:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
   <soap:Body>
        <p:Process xmlns:p='http://example.org/Operations' >
                  <m:Data xmlns:m='http://example.org/information' >
        Здесь некоторые данные с сообщением.
                 </m:Data>
        </p:Process>
   </soap:Body>
</soap:Envelope>
```

3.1.6 Атрибут SOAP encodingStyle

Атрибут soap:encodingStyle используют для указания применения конкретной схемы кодирования данных в XML. Однако это привносит сложность, так как эта функция может также служить для использования пространствами имен XML. Поэтому в Профиле предпочтительнее использовать литеральный, некодированный XML.

R1005 **ОБОЛОЧКА** *НЕ ДОЛЖНА содержать атрибуты soap:encodingStyle ни для каких элементов, пространством имен которых является «http://schemas.xmlsoap.org/soap/envelope/».*

R1006 **ОБОЛОЧКА** *НЕ ДОЛЖНА содержать атрибуты soap:encodingStyle ни для каких дочерних элементов soap:Body.*

R1007 **ОБОЛОЧКА**, *описанная в привязке rpc-литерал, НЕ ДОЛЖНА содержать атрибут soap:encodingStyle ни для какого элемента, являющегося потомком soap:Body.*

3.1.7 Атрибут SOAP mustUnderstand

Атрибут soap:mustUnderstand имеет ограниченный тип «xsd:boolean», который принимает значения только «0» или «1». Следовательно, допустимы только эти два значения.

R1013 **ОБОЛОЧКА**, *содержащая атрибут soap:mustUnderstand ДОЛЖНА использовать только лексические формы «0» и «1».* С

3.1.8 Атрибуты xsi:type

Во многих случаях отправители и получатели будут совместно использовать некоторые формы типа информации, относящиеся к оболочке, которой они обмениваются.

R1017 **ПОЛУЧАТЕЛЬ** *НЕ ДОЛЖЕН обязывать использовать в оболочках атрибут xsi:type, за исключением случаев, когда это требуется для указания производного типа (см. Схема XML. Часть 1: Структуры, Раздел 2.6.1).*

3.1.9 Атрибуты SOAP1.1 на элементах SOAP1.1

R1032 *Элементы soap:Envelope, soap:Header и soap:Body в* **ОБОЛОЧКЕ** *НЕ ДОЛЖНЫ иметь атрибутов в пространстве имен "http://schemas.xmlsoap.org/soap/envelope/".*

3.2 Модель обработки SOAP

В настоящем разделе Профиля содержатся ссылки на следующие спецификации (или их разделы):

- SOAP 1.1, раздел 2.

В SOAP 1.1 определена модель обработки оболочек. В частности, определены правила обработки блоков заголовка и тела оболочки. Также определены правила, относящиеся к генерации отказа. В Профиле установлены следующие ограничения на модель обработки:

3.2.1 Обязательные заголовки

Модель обработки SOAP 1.1 недоопределена в отношении обработки обязательных блоков заголовков. Обязательным является тот блок заголовка, дочерний элемент soap:Header которого порождает атрибут soap:mustUnderstand со значением «1».

R1025 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН обрабатывать оболочки таким образом, чтобы это выглядело так, как если бы обработка обязательных блоков осуществлялась до какой-либо фактической обработки.* SOAP12

Это требование гарантирует, что нежелательные побочные эффекты не будут появляться в результате объявления обязательного блока заголовка после обработки других частей сообщения.

3.2.2 Генерация отказа mustUnderstand

В Профиле требуется, чтобы получатели генерировали отказ, когда встречают направленные им блоки заголовков, которые они не понимают.

R1027 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН генерировать отказ «soap:MustUnderstand», когда оболочка содержит направленный ему (через soap:actor) обязательный блок заголовка (т. е., имеющий атрибут soap:mustUnderstand со значением «1»), который получатель не понимает.* SOAP12

3.2.3 Обработка отказа SOAP

При генерации отказа дальнейшая обработка не осуществляется. В обмене запрос-ответ сообщение об отказе должно быть передано отправителю получателем, а пользователю должно быть просигнализировано о какой-либо ошибке прикладного уровня.

Как в SOAP, так и в настоящем Профиле термин «генерировать» используют для обозначения создания отказа SOAP. Генерация отказа отличается от его передачи, которая в некоторых случаях и не требуется.

R1028 Когда **ПОЛУЧАТЕЛЬ** *генерирует отказ, НЕ СЛЕДУЕТ проводить дальнейшую обработку оболочки SOAP за исключением той, которая необходима для возврата к предыдущему шагу или компенсации результатов обработки оболочки до генерации отказа.* SOAP12

R1029 *Когда нормальный результат обработки оболочки SOAP привел бы к передаче ответа SOAP, но вместо этого был сгенерирован отказ,* **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН передать отказ вместо ответа.* SOAP12

R1030 **ПОЛУЧАТЕЛЮ**, *генерирующему отказ, СЛЕДУЕТ известить конечного пользователя о том, что был сгенерирован отказ, любыми, пригодными для данных обстоятельств средствами.* SOAP12

3.3 Отказы SOAP

3.3.1 Идентификация отказов SOAP

Некоторые потребительские реализации для определения наличия отказа неправильно используют только код статуса HTTP. Так как существуют ситуации, при которых инфраструктура сети изменяет код статуса HTTP, и для большей надежности в Профиле требуется проверка оболочки. Отказ является оболочкой, которая имеет единственный дочерний элемент элемента soap:Body и этот дочерний элемент есть soap:Fault.

R1107 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН интерпретировать сообщение SOAP как отказ, когда элемент soap:Body сообщения имеет единственного потомка soap:Fault.*

3.3.2 Структура отказа SOAP

Профиль ограничивает содержимое элемента soap:Fault элементами, явно описанными в SOAP 1.1.

R1000 *Когда* **ОБОЛОЧКА** *есть отказ, элемент soap:Fault НЕ ДОЛЖЕН иметь дочерних элементов, отличных от faultcode, faultstring, faultactor и detail.*

Например,

НЕПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' >
   <faultcode>soap:Client</faultcode>
   <faultstring>Invalid message format</faultstring>
   <faultactor>http://example.org/someactor</faultactor>
   <detail>В сообщении есть <b>много</b> элементов,
         которые я не понимаю
   </detail>
   <m:Exception xmlns:m='http://example.org/faults/exceptions' >
      <m:ExceptionType>Severe</m:ExceptionType>
   </m:Exception>
</soap:Fault>
```

ПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' >
   <faultcode>soap:Client</faultcode>
   <faultstring>Invalid message format</faultstring>
   <faultactor>http://example.org/someactor</faultactor>
   <detail>
      <m:msg xmlns:m='http://example.org/faults/exceptions'>
         В сообщении есть <b>много</b> элементов,
         которые я не понимаю
      </m:msg>
      <m:Exception xmlns:m='http://example.org/faults/exceptions'>
         <m:ExceptionType>Severe</m:ExceptionType>
      </m:Exception>
   </detail>
</soap:Fault>
```

3.3.3 Квалификация пространства имен отказа SOAP

Потомки элемента soap:Fault являются локальными для этого элемента, следовательно, квалификация пространства имен не требуется.

R1000 *Когда* **ОБОЛОЧКА** *является отказом, дочерние элементы элемента soap:Fault ДОЛЖНЫ быть неквалифицированными.*

Например,

НЕПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' >
    <soap:faultcode>soap:Client</soap:faultcode>
    <soap:faultstring>Invalid message format</soap:faultstring>
    <soap:faultactor>http://example.org/someactor</soap:faultactor>
    <soap:detail>
        <m:msg xmlns:m='http://example.org/faults/exceptions'>
```

```
            В сообщении есть <b>много</b> элементов,
            которые я не понимаю
        </m:msg>
    </soap:detail>
</soap:Fault>
```

ПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' xmlns=' ' >
    <faultcode>soap:Client</faultcode>
    <faultstring>Invalid message format</faultstring>
    <faultactor>http://example.org/someactor</faultactor>
    <detail>
        <m:msg xmlns:m='http://example.org/faults/exceptions'>
            В сообщении есть <b>много</b> элементов,
            которые я не понимаю
        </m:msg>
    </detail>
</soap:Fault>
```

3.3.4 Расширение отказа SOAP

В целях расширения допускается появление дополнительных атрибутов в элементе detail и дополнительных дочерних элементов элемента detail.

R1002 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН принимать отказы, имеющие любое число элементов, включая нуль, дочерних для элемента detail. Такие потомки могут быть и могут не быть квалифицированными.*

R1003 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН принимать отказы, имеющие любое число квалифицированных и неквалифицированных атрибутов, включая нуль, в элементе detail. Пространство имен квалифицированных атрибутов может быть любым, отличным от "http://schemas.xmlsoap.org/soap/envelope/".*

3.3.5 Язык отказа SOAP

Строки отказа являются человекочитаемыми указаниями на характер отказа. Следовательно, они могут быть на любом языке, а атрибут xml:lang может быть использован для указания языка строки отказа.

Это требование конфликтует со схемой для SOAP в его URL пространства имен. Схема без конфликтов находится в "http://ws-i.org/profiles/basic/1.1/soap-envelope-2004-01-21.xsd".

R1016 **ПОЛУЧАТЕЛЬ** *ДОЛЖЕН принимать отказы, имеющие атрибут xml:lang в элементе faultstring.*

3.3.6 Потребительские коды отказов SOAP

SOAP 1.1 допускает появление пользовательских кодов отказов в элементе faultcode путем использования нотации «точка».

Использование этого метода для расширения смысла определенных в SOAP 1.1 кодов отказов может привести к коллизии пространства имен. Следовательно, его использования следует избегать, так как оно может вызвать проблемы с интероперабельностью, когда одно и тоже имя используется справа от «.» (точки) для передачи разных смыслов.

Вместо этого в Профиле одобряется использование определенных в SOAP 1.1 кодов отказов вместе с дополнительной информацией в элементе detail для передачи характера отказа.

Альтернативно допустимо определять потребительские коды отказов в пространстве имен, контролируемом конкретным уполномоченным.

Уже существует ряд спецификаций, определяющих потребительские коды отказов с использованием нотации «.» (точка). Их использование в будущих спецификациях запрещено.

R1004 *Когда* **ОБОЛОЧКА** *содержит элемент faultcode, содержимому этого элемента СЛЕДУЕТ быть либо одним из кодов отказов, определенных в SOAP 1.1 (с предоставлением, при необходимости, дополнительной информации в элементе detail), либо Qname, пространство имен которого контролируется уполномоченным, определяющим отказ.*

R1031 *Когда* **ОБОЛОЧКА** *содержит элемент faultcode, в содержимом этого элемента для уточнения смысла отказа НЕ СЛЕДУЕТ использовать нотацию SOAP 1.1 «точка».*

Рекомендуется, чтобы приложения, которым требуются потребительские коды отказов, либо использовали определенные в SOAP 1.1 коды отказов и прилагали дополнительную информацию в элементе подробностей, либо определяли эти коды в пространстве имен, контролируемым указанным уполномоченным.

Например,

НЕПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
        xmlns:c="http://example.org/faultcodes' >
    <faultcode>soap:Server.ProcessingError</faultcode>
    <faultstring>При обработке сообщения произошла ошибка
    </faultstring>
</soap:Fault>
```

ПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'
        xmlns:c='http://example.org/faultcodes' >
    <faultcode>c:ProcessingError</faultcode>
    <faultstring>При обработке сообщения произошла ошибка
    </faultstring>
</soap:Fault>
```

ПРАВИЛЬНО:

```
<soap:Fault xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/' >
    <faultcode>soap:Server</faultcode>
    <faultstring>При обработке сообщения произошла ошибка
    </faultstring>
</soap:Fault>
```

3.4 Использование SOAP в HTTP

В настоящем разделе Профиля используют ссылки на следующие спецификации (или их разделы):

- SOAP 1.1, Раздел 6
- HTTP/1.1
- Метод управления состоянием HTTP

В SOAP 1.1 определен единственный протокол привязки к HTTP. В Профиле обязательно использование этой привязки и установлены следующие ограничения на ее использование:

3.4.1 Протокол привязки к HTTP

Определено несколько версий HTTP. HTTP/1.1 имеет преимущества в производительности и более ясно специфицирован, чем HTTP/1.0.

R1141 **СООБЩЕНИЕ** *ДОЛЖНО быть отправлено с использованием HTTP/1.1 или HTTP/1.0.*

R1140 **СООБЩЕНИЕ** *СЛЕДУЕТ отправлять, используя HTTP/1.1.*

Отметим, что в HTTP/1.1 подразумевается поддержка HTTP/1.0 и что промежуточные системы могут изменить версию сообщения; подробнее о версиях HTTP см. в RFC2145, «Использование и интерпретация номера версии HTTP».

3.4.2 Методы и расширения HTTP

В спецификации SOAP 1.1 определена такая его привязка к HTTP, что возможно использование двух методов: метод HTTP POST и метод схемы расширения HTTP M-POST. В Профиле требуется использование только метода HTTP POST и запрещается использование схемы расширения HTTP.

R1132 *Запрос* **СООБЩЕНИЯ** *HTTP ДОЛЖЕН использовать метод HTTP POST.*

R1108 **СООБЩЕНИЕ** *НЕ ДОЛЖНО использовать схему расширения HTTP (RFC2774).*

Схема расширения HTTP является экспериментальным методом модульного расширения HTTP. Поскольку этот метод широко не используется, а преимущества от его использования в SOAP не ясны, то Профиль не допускает его использование.

3.4.3 Заголовок HTTP SOAPAction

Тестирование показало, что взятие в кавычки значения поля в заголовке HTTP SOAPAction повышает интероперабельность реализаций. Хотя HTTP позволяет не брать в кавычки значения полей в заголовке, некоторые реализации SOAP требуют наличия кавычек.

SOAPAction является указанием процессорам. Вся существенная информация, относящаяся к предназначению сообщения, передается в soap:Envelope.

R1109 *Значение поля заголовка HTTP SOAPAction в HTTP-запросе* **СООБЩЕНИЯ** *ДОЛЖНО быть строкой в кавычках.* С

R1119 **ПОЛУЧАТЕЛЬ** *МОЖЕТ ответить отказом, если значение поля заголовка HTTP SOAPAction в сообщении не взято в кавычки.* C

R1127 **ПОЛУЧАТЕЛЬ** *НЕ ДОЛЖЕН полагаться на значение заголовка HTTP SOAPAction для правильной обработки сообщения.* SOAP12

Например,

ПРАВИЛЬНО:

Описание WSDL, в котором имеется:

<soapbind:operation soapAction="foo" />

приведет к сообщению со следующим полем заголовка HTTP SOAPAction:

"foo"

ПРАВИЛЬНО:

Описание WSDL, в котором имеется:

<soapbind:operation />

или

<soapbind:operation soapAction="" />

приведет к сообщению со следующим полем заголовка HTTP SOAPAction: ""

3.4.4 Успешные коды статуса HTTP

В HTTP используют коды статуса вида 2xx для сообщения об успешном завершении. В частности, код 200 является принимаемым по умолчанию для успешных сообщений, а код 202 может быть использован для указания того, что сообщение было передано на обработку. Дополнительно могут быть использованы другие коды статуса вида 2xx в зависимости от характера взаимодействия HTTP.

R1124 **ЭКЗЕМПЛЯР** *ДОЛЖЕН использовать код статуса HTTP вида 2xx в ответном сообщении, которое указывает на успешный результат запроса HTTP.*

R1111 **ЭКЗЕМПЛЯРУ** *СЛЕДУЕТ использовать код статуса HTTP «200 OK» HTTP в ответном сообщении, которое содержит оболочку, не являющуюся отказом.*

R1112 **ЭКЗЕМПЛЯРУ** *СЛЕДУЕТ использовать код статуса HTTP «200 OK» или «202 Accepted» для ответного сообщения, которое не содержит оболочку SOAP, но указывает на успешный результат запроса HTTP.*

Несмотря на то, что в HTTP 1.1 присвоен разный смысл кодам статуса "200" и "202", в контексте Профиля инициатору запроса следует рассматривать их эквивалентными. В Профиле приняты оба кода статуса, так как некоторые реализации SOAP слабо управляют реализацией протокола HTTP и не могут контролировать, какой из этих кодов статуса отправлен.

3.4.5 Коды статуса перенаправления HTTP

Имеются проблемы интероперабельности с использованием многих кодов статуса перенаправления HTTP в общем случае относящиеся к вопросу о том, использовать ли исходный метод или GET. В Профиле обязательным является использование кода «307 Temporary Redirect» в качестве правильного кода статуса для перенаправления, который имеет семантику перенаправления с тем же самым методом HTTP. Подробнее см. описание кодов статуса 3xx в RFC2616.

R1130 **ЭКЗЕМПЛЯР** *ДОЛЖЕН использовать код статуса HTTP «307 Temporary Redirect» при перенаправлении запроса в другую конечную точку.*

R1131 **ПОТРЕБИТЕЛЬ** *МОЖЕТ автоматически перенаправить запрос, когда в ответе он встретил код статуса HTTP «307 Temporary Redirect».*

В RFC2616 отмечено, что агентам пользователей не следует автоматически перенаправлять запросы; однако, это требование ориентировано на броузеры, а не на автоматические процессы (каковыми будут являться сетевые услуги). Следовательно, в Профиле потребителям разрешается, но не требуется, автоматически следовать за перенаправлениями.

3.4.6 Коды статуса ошибки клиента HTTP

В HTTP используют коды статуса вида 4xx для указания отказов, связанных с ошибками клиента. Хотя имеется ряд ситуаций, которые могут привести к одному из этих кодов, в Профиле выделяют те из них, когда запрос HTTP не имеет нужный тип оболочки и когда не используется ожидаемый метод («POST»).

R1125 **ЭКЗЕМПЛЯР** *ДОЛЖЕН использовать код статуса HTTP 4xx для ответа, указывающего на проблему с форматом запроса.*

R1113 **ЭКЗЕМПЛЯРУ** *СЛЕДУЕТ использовать код статуса HTTP «400 Bad Request», когда сообщение запроса HTTP неправильно сформировано.*

R1114 **ЭКЗЕМПЛЯРУ** *СЛЕДУЕТ использовать код статуса HTTP «405 Method not Allowed», когда методом сообщения запроса HTTP не является «POST».*

R1115 **ЭКЗЕМПЛЯРУ** *СЛЕДУЕТ использовать код статуса HTTP «415 Unsupported Media Type», когда значение поля заголовка Content-Type в сообщении HTTP не допускается его описанием WSDL.*

Эти требования не обязывают экземпляр отвечать на запросы. В некоторых случаях, таких как атаки «Отказ в обслуживании» (Denial of Service), экземпляр может игнорировать запросы.

В SOAP 1.1, раздел 6.2 требуется, чтобы отказ SOAP мог быть возвращен только с кодом HTTP 500 «Internal Server Error» (внутренняя ошибка сервера). В настоящем профиле это требование не изменено. Когда используют код статуса ошибки HTTP 4xx, сообщение ответа не должно содержать отказ SOAP.

3.4.7 Коды статуса ошибки сервера HTTP

В HTTP используют коды статуса вида 5xx для указания отказов, вызванных ошибками сервера.

R1126 **ЭКЗЕМПЛЯР** *ДОЛЖЕН возвращать код статуса HTTP «500 Internal Server Error», когда оболочкой ответа является Отказ.*

3.4.8 Куки HTTP

Метод управления состоянием HTTP (State Management Mechanism) [«Куки» («Cookies»)] допускает создание устойчивых сессий между сетевыми броузерами и серверами. Будучи спроектированными для гипертекстовых броузеров, куки не имеют хорошо определенной семантики для сетевых услуг и, поскольку они являются внешними по отношению к оболочке, не приняты ни в SOAP 1.1, ни в WSDL 1.1. Однако, существуют ситуации, когда использование куки может оказаться необходимым; например, для балансировки загрузки между серверами или для интеграции с наследуемыми системами, использующими куки. По этим причинам в Профиле куки не запрещены, но ограничены способы их использования.

R1120 **ЭКЗЕМПЛЯР** *МОЖЕТ использовать метод управления состоянием HTTP («Cookies»).*

R1122 **ЭКЗЕМПЛЯРУ**, *использующему куки, СЛЕДУЕТ соответствовать RFC2965.*

R1121 **ЭКЗЕМПЛЯРУ** *НЕ СЛЕДУЕТ требовать от потребителя поддержки куки для корректного выполнения функции.*

R1123 *Значение куки ДОЛЖНО рассматриваться* **ПОТРЕБИТЕЛЕМ** *как непрозрачное.*

В Профиле рекомендуется, чтобы куки не требовались экземплярами для правильной работы; куки могут быть указаниями, используемыми для оптимизации, без заметного влияния на выполнение сетевой услуги. Однако, они могут потребоваться для наследуемой интеграции и в других исключительных случаях, поэтому потребность в куки не делает экземпляр несоответствующим. Хотя куки могут иметь смысл для экземпляра, их не следует использовать как побочный канал данных между экземпляром и потребителем. Следовательно, интерпретация куки потребителем не допускается — требуется трактовать их как непрозрачные (т. е., не имеющие смысла для потребителя).

4 Описание услуги

Для описания услуг как набора операций в конечных точках над сообщениями в Профиле используют язык описания сетевых услуг [Web Services Description Language (WSDL)].

В настоящем разделе Профиля включены по ссылке следующие спецификации и определены точки их расширения:

• Расширяемый язык разметки [Extensible Markup Language (XML) 1.0 (Second Edition)][1];

• Пространства имен в XML (Namespaces in XML 1.0)[2];

• Схема XML. Часть 1: Структуры (XML Schema Part 1: Structures)[3]

Точки расширения.

• E0017 — Аннотации схемы — Схема XML допускает аннотации, которые могут переносить дополнительную информацию о структурах данных.

• Схема XML. Часть 2: Типы данных (XML Schema Part 2: Datatypes)[4];

• Язык описания сетевых услуг [Web Services Description Language (WSDL) 1.1][5]

---

[1] http://www.w3.org/TR/2000/REC-xml-20001006.

[2] http://www.w3.org/TR/1999/REC-xml-names-19990114.

[3] http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/.

[4] http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[5] http://www.w3.org/TR/2001/NOTE-wsdl-2-20010315.

Точки расширения:

• E0013 — Расширения WSDL — WSDL допускает в определенных местах расширения элементов и атрибутов; использование таких расширений требует стороннего соглашения.

• E0014 — Режим валидации — осуществляет или нет синтаксический анализатор, используемый для чтения документов WSDL и Схемы XML, валидацию DTD.

• E0015 — Привлечение внешних ресурсов — привлекает или нет синтаксический анализатор, используемый для чтения документов WSDL и Схемы XML, внешние категории и DTD.

• E0016 — Относительные URI — в WSDL нет адекватной спецификации использования относительных URI для следующих конструкций: soapbind:body/@namespace, soapbind:address/@location, wsdl:import/@location, xsd:schema/@targetNamespace и xsd:import/@schemaLocation. Их использование может потребовать дополнительной координации; подробнее см. XML Base.

4.1 Требуемое описание

Требуется, чтобы экземпляр сетевой услуги создавал контракт, по которому он действует как достижимый соответствующим образом.

R0001 *Либо описание WSDL 1.1* **ЭКЗЕМПЛЯРА**, *либо его шаблон привязки UDDI, либо то и другое ДОЛЖНО быть доступно по запросу авторизованному потребителю.*

Это означает, что если авторизованный потребитель запрашивает описание услуги соответствующего экземпляра услуги, то поставщик экземпляра услуги должен создать документ WSDL, или шаблон привязки UDDI, или то и другое, доступное этому потребителю. Экземпляр услуги может предоставлять во время выполнения доступ к документам WSDL на сервере, но это не требуется для того, чтобы считать его соответствующим. Аналогично, поставщик экземпляра услуги может зарегистрировать поставщика экземпляра в регистре UDDI, но это не требуется для того, чтобы считать его соответствующим. Во всех этих сценариях должен существовать контракт WSDL, но он может быть сделан доступным различными способами, зависящими от обстоятельств.

4.2 Структура документа

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

• WSDL 1.1, раздел 2.1

В WSDL 1.1 определена основанная на XML структура для описания сетевых услуг. В Профиле установлена обязательность использования этой структуры со следующими ограничениями:

4.2.1 Определения схемы WSDL

Нормативные схемы WSDL в приложении 4 спецификации WSDL 1.1 не согласованы с нормативным текстом спецификации. Профиль ссылается на новые документы схем, в которых исключены известные ошибки.

R2028 **ОПИСАНИЕ**, *использующее пространство имен WSDL (с префиксом «wsdl» в настоящем Профиле) ДОЛЖНО быть валидным в соответствии со следующей схемой XML: «http://ws-i.org/profiles/basic/1.1/wsdl-2004-08-24.xsd».*

R2029 **ОПИСАНИЕ**, *использующее пространство имен привязки SOAP WSDL (с префиксом «soapbind» в настоящем Профиле), ДОЛЖНО быть валидным в соответствии со следующей схемой XML: «http://ws-i.org/profiles/basic/1.1/wsdlsoap-2004-08-24.xsd».*

Хотя в Профиле требуется, чтобы описание WSDL было валидной схемой, от потребителя не требуется проводить валидацию документов WSDL. Автор документа WSDL отвечает за то, чтобы обеспечить валидность его схемы.

4.2.2 WSDL и импорт схемы

Некоторые примеры в WSDL 1.1 некорректно показывают утверждение импорта WSDL, используемое для импорта определений схем XML. В Профиле уточняется использование метода импорта для сохранения согласованности и ограниченности соответствующих областей применения. Импортируемые документы схем так же ограничены версией XML и требованиями к кодированию, согласованными с импортирующими документами WSDL.

R2001 **ОПИСАНИЕ** *ДОЛЖНО использовать утверждение WSDL «import» только для импорта другого описания WSDL.*

R2803 *В* **ОПИСАНИИ** *атрибут пространства имен в wsdl:import НЕ ДОЛЖЕН быть относительным URI.*

R2002 *Для импорта определений схем XML,* **ОПИСАНИЕ** *ДОЛЖНО использовать утверждение "import" схемы XML.*

R2003 **ОПИСАНИЕ** *ДОЛЖНО использовать утверждение "import" схемы XML только в элементе xsd:schema раздела типов.*

R2004 *В* **ОПИСАНИИ** *атрибут schemaLocation элемента xsd:import НЕ ДОЛЖЕН разрешаться до какого-либо документа, корневой элемент которого не является «schema» из пространства имен «http://www.w3.org/2001/XMLSchema».*

R2009 *Схема XML Schema, прямо или косвенно импортированная* **ОПИСАНИЕМ**, *МОЖЕТ включать в себя маркер порядка байтов [Unicode Byte Order Mark (BOM)].*

R2010 *Схема XML Schema, прямо или косвенно импортированная* **ОПИСАНИЕМ**, *ДОЛЖНА использовать кодирование UTF-8 или UTF-16.*

R2011 *Схема XML Schema, прямо или косвенно импортированная* **ОПИСАНИЕМ**, *ДОЛЖНА использовать версию 1.0 XML (eXtensible Markup Language W3C Recommendation).*

Например,

НЕПРАВИЛЬНО:

```
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote/definitions"
    xmlns:xsd1="http://example.com/stockquote/schemas"
                ...
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <import namespace="http://example.com/stockquote/schemas"
          location="http://example.com/stockquote/stockquote.xsd"/>
    <message name="GetLastTradePriceInput">
      <part name="body" element="xsd1:TradePriceRequest"/>
    </message>
          ...
</definitions>
```

ПРАВИЛЬНО:

```
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote/definitions"
                ...
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <import namespace="http://example.com/stockquote/definitions"
          location="http://example.com/stockquote/stockquote.wsdl"/>
    <message name="GetLastTradePriceInput">
       <part name="body" element="..."/>
    </message>
              ...
</definitions>
```

ПРАВИЛЬНО:

```
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote/"
    xmlns:xsd1="http://example.com/stockquote/schemas"
                ...
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <import namespace="http://example.com/stockquote/definitions"
         location="http://example.com/stockquote/stockquote.wsdl"/>
    <message name="GetLastTradePriceInput">
       <part name="body" element="xsd1:TradePriceRequest"/>
    </message>
          ...
</definitions>
```

4.2.3 Структура импорта атрибута location WSDL

В WSDL 1.1 непонятно требуется ли атрибут location утверждения wsdl:import, или требуется наличие его содержимого.

R2007 **ОПИСАНИЕ** *ДОЛЖНО специфицировать непустой атрибут location элемента wsdl:import.*

Хотя утверждение wsdl.import моделируется после утверждения xsd:import, атрибут location требуется в wsdl:import, но соответствующий атрибут в xsd:import, schemaLocation, является факультативным. Согласованность с обязательным атрибутом location, подразумевает, что его содержимое не пусто.

4.2.4 Схема импорта атрибута location WSDL

В WSDL 1.1 неясно, должен ли процессор WSDL фактически получать и обрабатывать документ WSDL из URI, специфицированного в атрибуте location встреченного им утверждения wsdl:import.

R2008 *ПОТРЕБИТЕЛЬ МОЖЕТ, но не обязан, получать описание WSDL из URI, специфицированного в атрибуте location элемента wsdl:import.* С

Значение атрибута location элемента wsdl:import является необязательным указанием. У процессора WSDL могут быть другие пути найти описания WSDL для данного пространства имен.

4.2.5 Расположение элементов import WSDL

Пример 3 в WSDL 1.1, раздел 3.1 вызывает путаницу относительно расположения элемента wsdl:import.

R2022 *При их наличии в* **ОПИСАНИИ** *элементы wsdl:import ДОЛЖНЫ предшествовать всем другим элементам пространства имен WSDL за исключением wsdl:documentation.*

R2023 *При их наличии в* **ОПИСАНИИ** *элементы wsdl:types ДОЛЖНЫ предшествовать всем другим элементам пространства имен WSDL за исключением wsdl:documentation и wsdl:import.*

Например,

НЕПРАВИЛЬНО:

```
<definitions name="StockQuote"
                ...
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <import namespace="http://example.com/stockquote/definitions"
        location="http://example.com/stockquote/stockquote.wsdl"/>
    <message name="GetLastTradePriceInput">
        <part name="body" type="tns:TradePriceRequest"/>
    </message>
        ...
    <service name="StockQuoteService">
        <port name="StockQuotePort" binding="tns:StockQuoteSoap">
            ....
        </port>
    </service>
    <types>
        <schema targetNamespace="http://example.com/stockquote/schemas"
            xmlns="http://www.w3.org/2001/XMLSchema">
        .......
        </schema>
    </types>
</definitions>
```

ПРАВИЛЬНО:

```
<definitions name="StockQuote"
      targetNamespace="http://example.com/stockquote/definitions">
      <import namespace="http://example.com/stockquote/base"
            location="http://example.com/stockquote/stockquote.wsdl"/>
      <message name="GetLastTradePriceInput">
            <part name="body" element="..."/>
      </message>
            ...
</definitions>
```

ПРАВИЛЬНО:

```
<definitions name="StockQuote"
                ...
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
```

```
        <schema targetNamespace="http://example.com/stockquote/schemas"
            xmlns="http://www.w3.org/2001/XMLSchema">
            .....
        </schema>
    </types>
    <message name="GetLastTradePriceInput">
            <part name="body" element="tns:TradePriceRequest"/>
    </message>
            ...
    <service name="StockQuoteService">
        <port name="StockQuotePort" binding="tns:StockQuoteSoap">
            ....
        </port>
    </service>
</definitions>
```

4.2.6 Требование версии XML

Ни WSDL 1.1, ни Схема XML 1.0 не устанавливают обязательность конкретной версии XML. В целях обеспечения интероперабельности документы и схемы WSDL, выраженные в XML, должны использовать версию 1.0.

R4004 **ОПИСАНИЕ** *ДОЛЖНО использовать версию 1.0 XML (eXtensible Markup Language W3C Recommendation).*

4.2.7 Декларация пространства имен XML

Хотя опубликованная поправка NE05 (см. http://www.w3.org/XML/xml-names-19990114-errata) допускает появление декларации этого пространства имен, некоторые старые процессоры рассматривают такую декларацию как ошибку. Настоящее требование обеспечивает, чтобы соответствующие артефакты имели наиболее широкую возможную интероперабельность.

R4005 *ОПИСАНИЮ НЕ СЛЕДУЕТ содержать декларацию пространства имен xmlns:xml="http://www.w3.org/XML/1998/namespace".* С

4.2.8 WSDL и Unicode BOM

В XML 1.0 допускается, чтобы документы, которые используют кодирование символов UTF-8, включали в себя BOM; следовательно, процессоры описаний должны быть готовы принять такие документы.

R4002 **ОПИСАНИЕ** *МОЖЕТ включать в себя Unicode Byte Order Mark (BOM).* С

4.2.9 Принимаемые кодирования символов WSDL

В Профиле требуется согласованное кодирование UTF-8 или UTF-16 как для SOAP, так и для WSDL.

R4003 **ОПИСАНИЕ** *ДОЛЖНО использовать кодирование либо UTF-8, либо UTF-16.*

4.2.10 Принудительное пространство имен

В Профиле запрещено принудительное пространство имен для wsdl:import.

R2005 *Атрибут targetNamespace элемента wsdl:definitions в импортируемом описании ДОЛЖЕН иметь то же самое значение, что и атрибут namespace элемента wsdl:import в импортирующем* **ОПИСАНИИ.**

4.2.11 Элемент documentation WSDL

Схема WSDL 1.1 и спецификация WSDL 1.1 несогласованы относительно того, где могут размещаться элементы wsdl:documentation.

R2030 *Элемент wsdl:documentation в* **ОПИСАНИИ** *МОЖЕТ присутствовать в качестве первого дочернего элемента wsdl:import, wsdl:part и wsdl:definitions в дополнение к элементам, указанным в спецификации WSDL1.1.WSDL20*

4.2.12 Расширения WSDL

Требования поддержки расширений WSDL, которые не специфицированы явным образом в настоящем или каком-либо ином профиле WS-I, могут привести к проблемам интероперабельности со средствами разработки, которые не были предназначены для понимания таких расширений.

R2025 **ОПИСАНИЕ**, *содержащее расширения WSDL, НЕ ДОЛЖНО использовать их в противоречии с другими требованиями настоящего Профиля.*

R2026 *В* **ОПИСАНИЕ** *НЕ СЛЕДУЕТ включать элементы расширения со значением атрибута wsdl:required равным «true» в любых конструкциях WSDL (wsdl:binding, wsdl:portType, wsdl:message, wsdl:types или wsdl:import), для которых заявляется о соответствии Профилю.*

R2027 *Если в ходе обработки описания потребитель встретил элемент расширения WSDL, который имеет атрибут wsdl:required с булевским значением «true» и который потребитель не понимает или не может обработать, то* **ПОТРЕБИТЕЛЬ** *ДОЛЖЕН отказать в обработке.*

Средства разработки, которые получают описание WSDL и генерируют программное обеспечение для экземпляра сетевой услуги, могут не иметь встроенного понимания неизвестных расширений WSDL. Следовательно, следует избегать использования обязательных расширений WSDL. Использование обязательного расширения WSDL, не имеющего доступной спецификации его применения и семантики, навязывает потенциально непреодолимые проблемы интероперабельности. Использование обязательного расширения WSDL, которое имеет доступную спецификацию его применения и семантики, уменьшает, но не исключает проблемы интероперабельности.

В настоящем Профиле все элементы в пространстве имен "http://schemas.xmlsoap.org/wsdl/" являются расширяемыми через элементы и атрибуты. Для удобства WS-I опубликовала версию схемы WSDL1.1, которая отражает эти возможности: http://ws-i.org/profiles/basic/1.1/wsdl11.xsd.

4.3 Типы

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

- WSDL 1.1, раздел 2.2

Элемент wsdl:types WSDL 1.1 включает в себя определения типов данных, которые относятся к описываемой сетевой услуге. В Профиле установлены следующие ограничения, относящиеся к тем частям содержимого элемента wsdl:types, на которые ссылаются элементы WSDL, заявляющие о соответствии Профилю.

4.3.1 Ссылки QName

В Схеме XML требуется, чтобы каждая ссылка QName использовала либо целевое, либо импортированное пространство имен (явно указанное элементом xsd:import). Ссылки QName на пространства имен, представленные только в виде вложенных импортов, не допускаются.

В WSDL 1.1 не ясно, какие схемы целевых пространств имен подходят для ссылок QName из элемента WSDL. В Профиле допустимы ссылки QName из элементов WSDL как на целевое пространство имен, определенное элементом xsd:schema, так и на импортированные пространства имен. Ссылки QName на пространства имен, определенные только через вложенный импорт, недопустимы.

R2101 **ОПИСАНИЕ** *НЕ ДОЛЖНО использовать ссылки QName на компоненты WSDL в пространствах имен, которые не были ни импортированы, ни определены в ссылающемся документе WSDL.*

R2102 *Ссылка QName на компонент Схемы в* **ОПИСАНИИ** *ДОЛЖНА использовать пространство имен, определенное в атрибуте targetNamespace элемента xsd:schema, или в атрибуте namespace элемента xsd:import в элементе xsd:schema.*

4.3.2 Структура Схемы targetNamespace

Хорошая практика состоит в требовании атрибута targetNamespace во всех элементах xsd:schema, являющихся потомками wsdl:types, что минимально обременяет авторов документов WSDL и позволяет избежать случаев неясных определений.

R2105 *Все элементы xsd:schema, содержащиеся в элементе wsdl:types* **ОПИСАНИЯ**, *ДОЛЖНЫ иметь атрибут targetNamespace с допустимым и ненулевым значением, если только элемент xsd:schema не имеет xsd:import и/или xsd:annotation в качестве единственного(ых) дочернего(их) элемента(ов).*

4.3.3 soapenc:Array

Рекомендация WSDL 1.1, раздел 2.2, декларация типов массивов интерпретируется различными способами, приводя к проблемам с интероперабельностью. Более того, имеются другие способы декларации массивов.

R2110 *Декларации в* **ОПИСАНИИ** *НЕ ДОЛЖНЫ расширять или ограничивать тип soapenc:Array.*

R2111 *Декларации в* **ОПИСАНИИ** *НЕ ДОЛЖНЫ использовать атрибут wsdl:arrayType в декларации типа.*

R2112 *Элементы в* **ОПИСАНИИ** *НЕ СЛЕДУЕТ именовать, используя соглашение ArrayOfXXX.*

R2113 **ОБОЛОЧКА** *НЕ ДОЛЖНА включать в себя атрибут soapenc:arrayType.*

Например,

НЕПРАВИЛЬНО:

Для данного описания WSDL:

<xsd:element name="MyArray2" type="tns:MyArray2Type"/>

```
<xsd:complexType name="MyArray2Type"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" >
   <xsd:complexContent>
      <xsd:restriction base="soapenc:Array">
       <xsd:sequence>
        <xsd:element name="x" type="xsd:string"
           minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute ref="soapenc:arrayType"
            wsdl:arrayType="tns.MyArray2Type[]"/>
        </xsd:restriction>
   </xsd:complexContent>
</xsd:complexType>
```

оболочка была бы сериализована следующим образом (для ясности опущены декларации пространств имен):

```
<MyArray2 soapenc:arrayType="tns:MyArray2Type[]" >
    <x>abcd</x>
    <x>efgh</x>
</MyArray2>
```

ПРАВИЛЬНО:

Для данного описания WSDL:

```
<xsd:element name="MyArray1" type="tns:MyArray1Type"/>
<xsd:complexType name="MyArray1Type">
   <xsd:sequence>
      <xsd:element name="x" type="xsd:string"
          minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

оболочка была бы сериализована следующим образом (для ясности опущены декларации пространств имен):

```
<MyArray1>
   <x>abcd</x>
   <x>efgh</x>
</MyArray1>
```

4.3.4 Целевые пространства имен WSDL и определения Схемы

Имена, определенные схемой, и имена, присвоенные определениям WSDL, являются раздельными пространствами символов.

R2114 *Целевое пространство имен для определений WSDL и целевое пространство имен для определений схемы в* **ОПИСАНИИ** *МОГУТ быть одинаковы.* WSDL20

4.4 Сообщения

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

- WSDL 1.1, раздел 2.3

В WSDL 1.1 элементы wsdl:message используют для представления абстрактных определений данных, предназначенных для передачи. Элементы wsdl:binding используют для определения того, как абстрактные определения привязываются к конкретной сериализации сообщения. В Профиле установлены ограничения на элементы wsdl:message и на то, как соответствующие элементы wsdl:binding могут использовать элемент(ы) wsdl:message.

Для того, чтобы сделать требования более компактными и простыми для понимания, в настоящем разделе используют следующие определения:

Определение: привязка rpc-литерал

«Привязка rpc-литерал» есть элемент wsdl:binding, все дочерние элементы wsdl:operation которого являются операциями rpc-литерал.

«Операция rpc-литерал» есть дочерний элемент wsdl:operation элемента wsdl:binding, потомки которого soapbind:body специфицируют атрибут use со значением «literal» и:

1. либо атрибут style со значением «rpc» специфицирован в дочернем элементе soapbind:operation,

2. либо атрибут style отсутствует в дочернем элементе soapbind:operation и элемент soapbind:binding в охватывающем элементе wsdl:binding специфицирует атрибут style со значением «rpc».

Определение: привязка документ-литерал

«Привязка документ-литерал» есть элемент wsdl:binding, все дочерние элементы wsdl:operation которого есть операции документ-литерал.

«Операция документ-литерал» есть дочерний элемент wsdl:operation элемента wsdl:binding, потомки которого soapbind:body специфицируют атрибут use со значением «literal» и:

1. либо атрибут style со значением «document» специфицирован в дочернем элементе soapbind:operation,

2. либо атрибут style отсутствует в дочернем элементе soapbind:operation и элемент soapbind:binding в охватывающем элементе wsdl:binding специфицирует атрибут style со значением «document»,

3. либо атрибут style отсутствует как в дочернем элементе soapbind:operation, так и в элементе soapbind:binding в охватывающем элементе wsdl:binding.

4.4.1 Привязки и части

Имеются различные интерпретации того, как много элементов wsdl:part допустимо или требуется для привязок документ-литерал и rpc-литерал и как они должны быть определены.

R2201 *Привязка документ-литерал в* **ОПИСАНИИ** *ДОЛЖНА в каждом своем элементе soapbind:body иметь самое большее одну часть, перечисленную в атрибуте parts, если этот атрибут специфицирован.*

R2209 *Элементу wsdl:binding в* **ОПИСАНИИ** *СЛЕДУЕТ связывать каждый элемент wsdl:part из wsdl:message с wsdl:portType, на который он ссылается, с элементом расширения привязки.*

R2210 *Если привязка документ-литерал в* **ОПИСАНИИ** *не специфицирует атрибут parts в элементе soapbind:body, то соответствующее абстрактное сообщение wsdl:message ДОЛЖНО определять нуль или один элемент wsdl:part.*

R2202 *Элемент wsdl:binding в* **ОПИСАНИИ** *МОЖЕТ содержать элемент(ы) soapbind:body, специфицирующий(е), что нуль частей образуют soap:Body.*

R2203 *Привязка rpc-литерал в* **ОПИСАНИИ** *ДОЛЖНА относиться, в ее элементе(ах) soapbind:body, только к элементу(ам) wsdl:part, который(ые) определен(ы) с использованием атрибута type.*

R2211 **ОБОЛОЧКА**, *описанная с привязкой rpc-литерал, НЕ ДОЛЖНА иметь атрибут xsi:nil со значением «1» или «true» в присоединенной части.*

R2207 *Элемент wsdl:message в* **ОПИСАНИИ** *МОЖЕТ содержать элементы wsdl:part, которые используют атрибут element, предоставляющий те wsdl:part, на которые не ссылается soapbind:body в привязке rpc-литерал.*

R2204 *Привязка документ-литерал в* **ОПИСАНИИ** *ДОЛЖНА относиться, в каждом своем элементе soapbind:body, только к элементу(ам) wsdl:part, который(ые) определен(ы) с использованием атрибута element.*

R2208 *Привязка в* **ОПИСАНИИ** *МОЖЕТ содержать элемент(ы) soapbind:header, относящийся(еся) к wsdl:part в том же самом wsdl:message, который указывается ее элементом(ами) soapbind:body.*

R2212 **ОБОЛОЧКА** *ДОЛЖНА содержать ровно один элемент присоединенной части для каждого элемента wsdl:part, ограничивающего соответствующий оболочке элемент soapbind:body.*

R2213 *В описании документ-литерал, когда значением атрибута part элемента soapbind:body является пустая строка, соответствующая* **ОБОЛОЧКА** *не ДОЛЖНА иметь содержимое в элементе soap:Body.*

R2214 *В описании rpc-литерал, когда значением атрибута part элемента soapbind:body является пустая строка, соответствующая* **ОБОЛОЧКА** *не ДОЛЖНА иметь элементов присоединенной части.*

Использование элементов wsdl:message с нулем частей разрешается в стилях документов для того, чтобы допустить операции, которые могут отправить или принять оболочки с пустыми элементами soap:Body. Использование элементов wsdl:message с нулем частей разрешается в стилях RPC для того, чтобы допустить отсутствие параметров и/или возвращаемого значения.

Для привязки документ-литерал в Профиле требуется, чтобы самое большее одна часть, абстрактно определенная с атрибутом element, была сериализована в элемент soap:Body.

Когда элемент wsdl:part определен с использованием атрибута type, сериализация этой части в сообщении эквивалента неявной (Схемой XML) квалификации атрибута minOccurs со значением «1», атрибута maxOccurs со значением «1» и атрибута nillable со значением «false».

Необходимо специфицировать эквивалентную неявную квалификацию, так как элемент wsdl:part не позволяет специфицировать правила для кардинального числа и возможности обнуления. Спецификация этих правил облегчает интероперабельность между реализациями. Эквивалентная неявная квалификация для атрибута обнуления имеет значение «false», так как если бы она имела значение «true», то невозможно было бы спроектировать часть, поскольку от клиента всегда требовалась бы отправка значения. Ожидается, что в приложениях, для которых желательно разрешить обнуление wsdl:part, приложение будет генерировать оболочку complexType и специфицировать правила обнуления для элементов, содержащихся в такой оболочке.

4.4.2 Привязки и отказы

Существует несколько интерпретаций того, как могут быть определены элементы wsdl:part, которые описывают soapbind:fault, soapbind:header и soapbind:headerfault.

R2205 *В* **ОПИСАНИИ** *элементы wsdl:binding ДОЛЖНЫ ссылаться, в каждом из своих элементов soapbind:header, soapbind:headerfault и soapbind:fault, только на элемент(ы) wsdl:part, который(ые) был(и) определен(ы) с использованием атрибута element.*

Поскольку отказы и заголовки не содержат параметров, в soapbind:fault, soapbind:header и soapbind:headerfault принято, по WSDL 1.1, что значением атрибута style является «document». В R2204 требуется, чтобы все элементы wsdl:part с атрибутом style, значением которого является «document», связанные с soapbind:body, были определены с использованием атрибута element. Это же требование справедливо для элементов soapbind:fault, soapbind:header и soapbind:headerfault.

4.4.3 Декларация элементов part

В примерах 4 и 5 WSDL 1.1, раздел 3.1, некорректно показано использование типов Схемы XML (например, «xsd:string») в качестве допустимых значений атрибута element элемента wsdl:part.

R2206 *В* **ОПИСАНИИ** *элемент wsdl:message, содержащий wsdl:part, который использует атрибут element, ДОЛЖЕН ссылаться, в этом атрибуте, на глобальную декларацию элемента.*

Например,

НЕПРАВИЛЬНО:

```
<message name="GetTradePriceInput">
    <part name="tickerSymbol» element="xsd:string"/>
    <part name="time" element="xsd:timeInstant"/>
</message>
```

НЕПРАВИЛЬНО:

```
<message name="GetTradePriceInput">
    <part name="tickerSymbol" element="xsd:string"/>
</message>
```

ПРАВИЛЬНО:

```
<message name="GetTradePriceInput">
    <part name="body" element="tns:SubscribeToQuotes"/>
</message>
```

4.5 Типы портов

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

- WSDL 1.1, раздел 2.4

В WSDL 1.1 элементы wsdl:portType используют для группировки наборов абстрактных операций. В Профиле установлены следующие ограничения на соответствующие элементы wsdl:portType:

4.5.1 Порядок элементов part

Разрешение использовать parameterOrder помогает генераторам кода отображать сигнатуру метода и передается в виде сообщения.

R2301 *Порядок элементов в soap:Body* **ОБОЛОЧКИ** *ДОЛЖЕН быть тем же самым, что и порядок wsdl:part в wsdl:message, которое описывает его для каждого элемента wsdl:part, связанного с соответствующим оболочке элементом soapbind:body.*

R2302 *В* **ОПИСАНИИ** *МОЖНО использовать атрибут parameterOrder элемента wsdl:operation для указания возвращаемого значения и сигнатуры метода в качестве подсказки генераторам кода.*

4.5.2 Допустимые операции

Операции запрос-ответ и извещение не являются хорошо определенными в WSDL 1.1; более того, в WSDL 1.1 не определена привязка к ним.

R2303 **ОПИСАНИЕ** *НЕ ДОЛЖНО использовать операции типов запрос-ответ и извещение в определении wsdl:portType.*

4.5.3 Различные операции

Перезагрузка имени операции в wsdl:portType запрещается в Профиле.

R2304 *В* **ОПИСАНИИ** *элемент wsdl:portType ДОЛЖЕН иметь операции с различающимися значениями их атрибута name.*

Это требование применяют только к wsdl:operation в пределах данного wsdl:portType. Элемент wsdl:portType может иметь операцию wsdl:operation с тем же самым именем, которое может быть найдено в другом wsdl:portType.

4.5.4 Конструкция атрибута parameterOrder

В WSDL 1.1 точно не установлено, как должен быть устроен атрибут parameterOrder элемента wsdl:operation (который является дочерним для элемента wsdl:portType).

R2305 *В* **ОПИСАНИИ** *элемент wsdl:operation, дочерний элемента wsdl:portType, ДОЛЖЕН быть построен так, что атрибут parameterOrder, при его наличии, опускает самое большее один элемент wsdl:part из выходного сообщения.*

Если wsdl:part из выходного сообщения опущен в списке wsdl:part, который является значением атрибута parameterOrder, то единственный опущенный элемент wsdl:part является возвращаемым значением. Нет ограничений на тип возвращаемого значения. Если все part присутствуют, то нет возвращаемого значения.

4.5.5 Исключение атрибутов type и element

В WSDL 1.1 точно не установлено, что атрибуты type и element не могут быть специфицированы для определения элемента wsdl:part в элементе wsdl:message.

R2306 *Элемент wsdl:message в* **ОПИСАНИИ** *НЕ ДОЛЖЕН специфицировать атрибуты type и element в одном и том же элементе wsdl:part.*

4.6 Привязки

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

- WSDL 1.1, раздел 2.5

В WSDL 1.1 элемент wsdl:binding предоставляет конкретные спецификации протокола и формата данных для операций и сообщений, определенных конкретным wsdl:portType. В Профиле установлены следующие ограничения на соответствующие спецификации привязок:

4.6.1 Использование привязки SOAP

В Профиле выбор привязок ограничен хорошо определенными и наиболее широко используемыми привязками SOAP.

R2401 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН использовать привязку WSDL SOAP так, как определено в WSDL 1.1, раздел 3.*

Тем самым устанавливаются требования к конструкции соответствующих элементов wsdl:binding. Требования к описанию в целом не установлены; в частности, не запрещаются документы WSDL, содержащие несоответствующие элементы wsdl:binding. Кроме того, в привязке могут присутствовать элементы расширения WSDL, изменяющие сериализацию сообщений.

4.7 Привязка SOAP

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

- WSDL 1.1, раздел 3.0.

В WSDL 1.1 определена привязка для конечных точек SOAP 1.1. В Профиле установлена обязательность использования привязки SOAP по WSDL 1.1 и следующие ограничения:

4.7.1 Спецификация атрибута transport

Имеется несогласованность между спецификацией WSDL 1.1 и схемой WSDL 1.1, касающаяся атрибута transport. Согласно спецификации WSDL 1.1 он является обязательным; однако, в схеме он показан как факультативный.

R2701 *Элемент wsdl:binding в* **ОПИСАНИИ** *должен быть построен так, что его дочерний элемент soapbind:binding специфицирует атрибут transport.*

4.7.2 Транспортный протокол HTTP

В Профиле установлено ограничение для нижележащего транспортного протокола HTTP.

R2702 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН специфицировать транспортный протокол HTTP с привязкой SOAP. В частности, атрибут transport его дочернего элемента soapbind:binding ДОЛЖЕН иметь значение «http://schemas.xmlsoap.org/soap/http».*

Это требование не запрещает использовать HTTPS; см. R5000.

4.7.3 Согласованность атрибута style

Атрибут взаимодействия style, равный «document» или «rpc», специфицирован на уровне элемента wsdl:operation, допуская элементы wsdl:binding, у которых wsdl:operation имеют разные значения style. Это приводит к проблемам интероперабельности.

R2705 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН быть либо привязкой rpc-литерал, либо привязкой документ-литерал.*

4.7.4 Кодирование и атрибут use

Профилем запрещено использование кодирований, включая кодирование SOAP.

R2706 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН использовать значение «literal» для атрибута use во всех элементах soapbind:body, soapbind:fault, soapbind:header и soapbind:headerfault.*

4.7.5 Кратные привязки для элементов portType

Профиль явно разрешает кратные привязки для одного элемента portType.

R2709 *Элемент wsdl:portType в* **ОПИСАНИИ** *МОЖЕТ иметь ноль и более ссылающихся на него элементов wsdl:binding, определенных в том же самом или в других документах WSDL.*

4.7.6 Сигнатуры операций

Определение: сигнатура операции

В Профиле «сигнатура операции» определена как полностью квалифицированное имя дочернего элемента тела SOAP входного сообщения SOAP, описанное операцией в привязке WSDL.

В случае привязки rpc-литерал имя операции используют как оболочку для присоединенных частей. В случае привязки документ-литерал, так как оболочка с именем операции отсутствует, сигнатуры сообщений должны быть корректно спроектированы так, чтобы удовлетворять настоящим требованиям.

Конечная точка, поддерживающая кратные операции, должна недвусмысленно идентифицировать операцию, которая будет вызвана на основании полученного входящего сообщения. Это возможно только в том случае, когда все операции, специфицированные в связанном с конечной точкой элементе wsdl:binding, имеют уникальные сигнатуры операций.

R2710 *Операции в wsdl:binding в* **ОПИСАНИИ** *ДОЛЖНЫ приводить к отличающимся друг от друга сигнатурам операций.*

4.7.7 Кратные порты в конечной точке

Когда входящие сообщения, предназначенные для двух различных wsdl:port в одной и той же конечной точке сети, неразличимы на физическом уровне, может оказаться невозможным определить вызываемый ими wsdl:port. Это может привести к проблемам интероперабельности. Однако, возможны ситуации (например, наличие нескольких версий SOAP или нескольких прикладных версий, соответствие разным профилям), когда желательно разместить несколько портов в одной конечной точке; поэтому Профиль допускает такую возможность.

R2711 *В* **ОПИСАНИИ** *НЕ СЛЕДУЕТ иметь более одного элемента wsdl:port с одним и тем же значением для атрибута location элемента soapbind:address.*

4.7.8 Дочерний элемент привязки документ-литерал

В WSDL 1.1 не вполне понятно, что является дочерним элементом soap:Body в привязке документ-литерал.

R2712 *Привязка документ-литерал ДОЛЖНА быть сериализована, как* **ОБОЛОЧКА** *с элементом soap:Body, дочерний элемент которого является экземпляром глобальной декларации элемента, указанной соответствующей частью wsdl:message.*

4.7.9 Односторонние операции

Имеются различные интерпретации того, как должен использоваться протокол HTTP при односторонних операциях.

R2714 *Для односторонних операций* **ЭКЗЕМПЛЯР** *НЕ ДОЛЖЕН возвращать ответ HTTP, содержащий оболочку. Конкретно, тело ответа HTTP должно быть пустым.*

R27501 **ПОТРЕБИТЕЛЬ** *при односторонней операции ДОЛЖЕН игнорировать оболочку, переданную в ответном сообщении HTTP.*

R2727 *При односторонних операциях* **ПОТРЕБИТЕЛЬ** *НЕ ДОЛЖЕН интерпретировать успешный ответный код статуса HTTP (т. е., 2xx) в том смысле, что сообщение валидно или что получатель его обработал.*

Односторонние операции не создают ответов SOAP. Следовательно, Профиль запрещает отправлять оболочку SOAP в ответ на одностороннюю операцию. Это означает, что передача односторонней операции не может привести к ответам или ошибкам уровня обработки. Например, в этой ситуации не может быть возвращен ответ HTTP «500 Internal Server Error» (внутренняя ошибка сервера), содержащий отказ.

Ответ HTTP на одностороннюю операцию указывает на успех или отказ передачи сообщения. Основываясь на семантике различных ответных кодов статуса, поддерживаемых протоколом HTTP, в Профиле установлено, что предпочтительными кодами статуса, которые следует ожидать отправителю, являются «200» и «202», означающими, что одностороннее сообщение было получено. Успешная передача не означает, что уровни обработки SOAP и прикладной логики имели возможность проверить валидность оболочки или провести ее обработку.

4.7.10 Пространства имен для элементов soapbind

Имеется противоречие в вопросе о том, какое пространство имен ассоциировано с различными дочерними элементами soap:Envelope, что приводит к проблемам интероперабельности. В Профиле уточнен этот вопрос.

R2716 *Привязка документ-литерал в* **ОПИСАНИИ** *НЕ ДОЛЖНА иметь атрибут namespace, специфицированный во внутренних элементах soapbind:body, soapbind:header, soapbind:headerfault и soapbind:fault.*

R2717 *Привязка rpc-литерал в* **ОПИСАНИИ** *ДОЛЖНА иметь в элементах soapbind:body специфицированный атрибут namespace, значением которого ДОЛЖНО быть абсолютное URI.*

R2726 *Привязка rpc-литерал в* **ОПИСАНИИ** *НЕ ДОЛЖНА иметь атрибут namespace, специфицированный во внутренних элементах soapbind:header, soapbind:headerfault и soapbind:fault.*

В привязке SOAP документ-литерал сериализованный дочерний элемент элемента soap:Body получает свое пространство имен из атрибута targetNamespace схемы, определяющей элемент. Использование атрибута namespace элемента soapbind:body изменило бы пространство имен элемента. Профилем это запрещается.

Напротив, в привязке SOAP rpc-литерал сериализованный дочерний элемент элемента soap:Body состоит из окружающего элемента, пространство имен которого является значением атрибута namespace элемента soapbind:body, а локальное имя которого является либо именем операции, либо именем операции с суффиксом «Response». Атрибут namespace является обязательным для обеспечения того, что потомки элемента soap:Body квалифицированы пространством имен.

4.7.11 Согласованность элементов portType и binding

Описание WSDL должно быть согласованным на уровнях элементов wsdl:portType и wsdl:binding.

R2718 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН иметь тот же самый набор wsdl:operation, что и элемент wsdl:portType, к которому он относится.*

4.7.12 Описание элементов headerfault

Имеется несогласованность между текстом спецификации WSDL и схемой WSDL в части элемента soapbind:headerfault.

R2719 *Элемент wsdl:binding в* **ОПИСАНИИ** *МОЖЕТ не содержать элементы soapbind:headerfault, если нет известных отказов заголовка.*

В схеме WSDL 1.1 элемент soapbind:headerfault является обязательным в элементах операции wsdl:input и wsdl:output, тогда как в спецификации WSDL 1.1 он является факультативным. Правильным является утверждение в спецификации.

4.7.13 Нумерация отказов

Описание сетевой услуги должно включать в себя все отказы, известные на момент определения этой услуги. Также имеется потребность в генерации новых отказов, которые не были идентифицированы на момент определения сетевой услуги.

R2740 *Элементу wsdl:binding в* **ОПИСАНИИ** *СЛЕДУЕТ содержать элементы soapbind:fault, описывающие каждый известный отказ.*

R2741 *Элементу wsdl:binding в* **ОПИСАНИИ** *СЛЕДУЕТ содержать элементы soapbind:headerfault, описывающие каждый известный отказ заголовка.*

R2742 **ОБОЛОЧКА** *МОЖЕТ содержать отказ с элементом detail, который не описан элементом soapbind:fault в соответствующем описании WSDL.*

R2743 **ОБОЛОЧКА** *МОЖЕТ содержать подробности обработки заголовка, относящиеся к отказу в блоке заголовка SOAP, который не описан элементом soapbind:headerfault в соответствующем описании WSDL.*

4.7.14 Типы и имена элементов привязки SOAP

Схема WSDL 1.1 не согласована со спецификацией WSDL 1.1 в части имен и типов атрибутов элементов soapbind:header и soapbind:headerfault.

R2720 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН использовать атрибут part со схемой типа «NMTOKEN» во всех содержащихся в нем элементах soapbind:header и soapbind:headerfault.*

R2749 *Элемент wsdl:binding в* **ОПИСАНИИ** *НЕ ДОЛЖЕН использовать атрибут parts в содержащихся в нем элементах soapbind:header и soapbind:headerfault.*

Схема WSDL получает имя атрибута как «parts» и тип как «NMTOKENS». Схема некорректна, так как каждый элемент soapbind:header и soapbind:headerfault относится к единственному элементу wsdl:part.

Например,

ПРАВИЛЬНО:

```
<binding name="StockQuoteSoap" type="tns:StockQuotePortType">
    <soapbind:binding style="document"
            transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="SubscribeToQuotes">
            <input message="tns:SubscribeToQuotes">
                <soapbind:body parts="body" use="literal"/>
                <soapbind:header message="tns:SubscribeToQuotes"
                                part="subscribeheader" use="literal"/>
            </input>
        </operation>
</binding>
```

4.7.15 Атрибут name в отказах

Имеется несогласованность между спецификацией WSDL 1.1 и схемой WSDL 1.1, в которой не перечислен атрибут name.

R2721 *Элемент wsdl:binding в* **ОПИСАНИИ** *ДОЛЖЕН иметь атрибут name, заданный во всех содержащихся в нем элементах soapbind:fault.*

R2754 *В* **ОПИСАНИИ** *значение атрибута name в элементе soapbind:fault ДОЛЖНО быть согласованным со значением атрибута name в его родительском элементе wsdl:fault.*

4.7.16 Отсутствие атрибута use

Имеется несогласованность между спецификацией WSDL 1.1 и схемой WSDL 1.1, касающаяся атрибута use.

R2722 *Элемент wsdl:binding в* **ОПИСАНИИ** *МОЖЕТ специфицировать атрибут use в содержащихся в нем элементах soapbind:fault.*

R2723 *Если в элементе wsdl:binding в* **ОПИСАНИИ** *присутствует атрибут use в элементе soapbind:fault, то его значение ДОЛЖНО быть равно «literal».*

В WSDL 1.1, раздел Section 3.6 указано, что атрибут use элемента soapbind:fault является обязательным, хотя в схеме атрибут use определен как факультативный. Профиль для согласованности с soapbind:body определяет его как факультативный.

Так как атрибут use факультативный, Профиль идентифицирует принимаемое по умолчанию значение в случаях, когда атрибут опущен.

Наконец, для обеспечения самосогласованности Профиля, единственным допустимым значением для атрибута use является «literal».

4.7.17 Принимаемые по умолчанию значения для атрибута use

Имеется несогласованность между спецификацией WSDL 1.1 и схемой WSDL 1.1 относительно того, является ли атрибут use факультативным для soapbind:body, soapbind:header и soapbind:headerfault, и если это так, то что означает его отсутствие.

R2707 *Элемент wsdl:binding в* **ОПИСАНИИ**, *содержащий один или несколько элементов soapbind:body, soapbind:fault, soapbind:header или soapbind:headerfault, в которых не специфицирован атрибут use, ДОЛЖЕН быть интерпретирован так, как если бы каждый раз для указанного атрибута было задано значение «literal».*

4.7.18 Согласованность оболочки с описанием

Настоящие требования устанавливают: когда экземпляр получает оболочку, которая не соответствует описанию WSDL, следует генерировать отказ.

Как устанавливает модель обработки SOAP: (a) если пространство имен элемента «Envelope» является некорректным, то должен быть сгенерирован код отказа «VersionMismatch», (b) если экземпляр не понимает блок заголовка SOAP со значением «1» для атрибута soap:mustUnderstand, то должен быть сгенерирован отказ «MustUnderstand». Во всех других случаях, когда оболочка не согласуется с описанием WSDL, следует генерировать отказ с кодом «Client».

R2724 *Если* **ЭКЗЕМПЛЯР** *получает оболочку, которая не согласована с описанием WSDL, то СЛЕДУЕТ генерировать элемент soap:Fault с кодом отказа «Client» во всех случаях, когда не генерируется отказ «MustUnderstand» или «VersionMismatch».*

R2725 *Если* **ЭКЗЕМПЛЯР** *получает оболочку, которая не согласована с описанием WSDL, то он ДОЛЖЕН проверить условия отказов «VersionMismatch», «MustUnderstand» и «Client» в указанном порядке.*

4.7.19 Окружение ответа

WSDL 1.1 раздел 3.5 может быть интерпретирован в том смысле, что охватывающий элемент ответа RPC должен быть назван идентично имени wsdl:operation.

R2729 **ОБОЛОЧКА**, *описанная с привязкой rpc-литерал, которая является ответом, ДОЛЖНА иметь охватывающий элемент, имя которого есть соответствующее имя wsdl:operation с суффиксом «Response».*

4.7.20 Дополнения частей

Для оболочек rpc-литерал в WSDL 1.1 не ясно, что является пространством имен, если оно есть, дополнительных элементов для параметров и возвращаемого значения. В разных реализациях сделан разный выбор, что приводит к проблемам с интероперабельностью.

R2735 **ОБОЛОЧКА**, *описанная с привязкой rpc-литерал, ДОЛЖНА размещать элементы дополнений частей для параметров и возвращаемого значения без пространства имен.*

R2755 *Элемент дополнения части в* **СООБЩЕНИИ**, *описанном с привязкой rpc-литерал, ДОЛЖЕН иметь локальное имя с тем же самым значением, что и атрибут name соответствующего элемента wsdl:part.*

Выбор одной из альтернатив является критическим для достижения интероперабельности. Профиль помещает элементы дополнения частей вне пространства имен, так как это просто охватывает все случаи и не приводит к логической несогласованности.

4.7.21 Пространства имен для потомков дополнений частей

Для оболочек rpc-литерал в WSDL 1.1 не ясно, что является правильной квалификацией пространства имен для дочерних элементов дополнений частей, когда соответствующие абстрактные части определены как имеющие типы из пространств имен, отличных от targetNamespace описания WSDL для абстрактных частей.

R2737 **ОБОЛОЧКА**, *описанная с привязкой rpc-литерал, ДОЛЖНА квалифицировать пространство имен потомков элементов дополнительных частей для параметров и возвращаемого значения так, как определено схемой, в которой определены типы дополнений частей.*

WSDL 1.1 раздел 3.5 устанавливает: «Имена, типы и значения частей атрибута пространства имен являются входом для кодирования, хотя атрибут пространства имен применяется только к содержимому, неявно определенному абстрактными типами».

Однако, это не устанавливает явно, что содержимое элемента и атрибута абстрактных (complexType) типов является пространством имен, квалифицированным targetNamespace, в котором определены эти элементы и атрибуты. WSDL 1.1 был предназначен для функционирования таким же образом, что и Схема XML. Следовательно, реализации должны следовать тем же правилам, что и для Схемы XML. Если complexType, определенный в targetNamespace «A», был импортирован и указан в декларации элемента в схеме с targetNamespace «B», то содержимое атрибута и элемента дочерних элементов этого complexType будут квалифицированы пространством имен «A», а элемент будет квалифицирован пространством имен «B».

Например.

ПРАВИЛЬНО:

Для данного WSDL, который определяет некоторую схему в пространстве имен «http://example.org/foo/» в разделе wsdl:types, содержащемся в wsdl:definitions, который имеет атрибут targetNamespace со значением «http://example.org/bar/» (следовательно, имеет тип, объявленный в одном пространстве имен, и содержит элемент, определенный в другом пространстве имен):

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:bar="http://example.org/bar/"
targetNamespace="http://example.org/bar/"
xmlns:foo="http://example.org/foo/">
<types>
    <xsd:schema targetNamespace="http://example.org/foo/"
        xmlns:tns="http://example.org/foo/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">
        <xsd:complexType name="fooType">
            <xsd:sequence>
                <xsd:element ref="tns:bar"/>
                <xsd:element ref="tns:baf"/>
            </xsd:sequence>
        </xsd:complexType>
        <xsd:element name="bar" type="xsd:string"/>
        <xsd:element name="baf" type="xsd:integer"/>
    </xsd:schema>
</types>
<message name="BarMsg">
    <part name="BarAccessor" type="foo:fooType"/>
</message>
    <portType name="BarPortType">
        <operation name="BarOperation">
            <input message="bar:BarMsg"/>
        </operation>
</portType>
<binding name="BarSOAPBinding" type="bar:BarPortType">
        <soapbind:binding
        transport="http://schemas.xmlsoap.org/soap/http"
        style="rpc"/>
        <operation name="BarOperation">
            <input>
                <soapbind:body use="literal" namespace="http://example.org/bar/"/>
            </input>
        </operation>
        </binding>
        <service name="serviceName">
            <port name="BarSOAPPort" binding="bar:BarSOAPBinding">
                <soapbind:address location="http://example.org/myBarSOAPPort"/>
            </port>
        </service>
</definitions>
```

Результирующей оболочкой для BarOperation является:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:foo="http://example.org/foo/">
    <s:Header/>
        <s:Body>
```

```
        <m:BarOperation xmlns:m="http://example.org/bar/">
            <BarAccessor>
                <foo:bar>String</foo:bar>
                <foo:baf>0</foo:baf>
            </BarAccessor>
        </m:BarOperation>
    </s:Body>
</s:Envelope>
```

4.7.22 Обязательные заголовки

В WSDL 1.1 нет ясной спецификации того, все ли элемнты soapbind:header, заданные в элементах wsdl:input или wsdl:output элемента wsdl:operation в разделе привязки SOAP описания WSDL, должны быть включены в результирующие оболочки при передаче. В Профиле установлена обязательность таких заголовков, так как в WSDL 1.1 нет способа пометить факультативный заголовок.

R2738 **ОБОЛОЧКА** *ДОЛЖНА включать в себя все элементы soapbind:header, специфицированные в элементах wsdl:input или wsdl:output элемента wsdl:operation в элементе wsdl:binding, который их описывает.*

4.7.23 Допустимость неописанных заголовков

Заголовки являются методом расширения SOAP. По различным причинам может потребоваться включить в оболочку заголовки, не определенные в описании WSDL.

R2739 **ОБОЛОЧКА** *МОЖЕТ содержать блоки заголовков SOAP, которые не описаны в описывающем ее wsdl:binding.*

R2753 **ОБОЛОЧКА**, *содержащая блоки заголовков SOAP, которые не описаны в соответствующем wsdl:binding, МОЖЕТ иметь в таких блоках заголовков SOAP атрибут mustUnderstand равный «1».*

4.7.24 Порядок заголовков

Нет связи между порядком элементов soapbind:header в описании и порядком блоков заголовков SOAP в оболочке. Аналогично, в оболочке может встретиться несколько экземпляров каждого специфицированного блока заголовка SOAP.

R2751 *Порядок элементов soapbind:header в разделе soapbind:binding* **ОПИСАНИЯ** *ДОЛЖЕН рассматриваться независимо от порядка блоков заголовков SOAP в оболочке.*

R2752 **ОБОЛОЧКА** *МОЖЕТ содержать несколько экземпляров каждого блока заголовка SOAP для каждого элемента soapbind:header в соответствующем потомке элемента soapbind:binding соответствующего описания.*

4.7.25 Описание SOAPAction

Тестирование интероперабельности показало, что требование заключать в кавычки значение поля SOAPAction заголовка HTTP повышает интероперабельность реализаций. Хотя в HTTP допускается не заключать значения полей в кавычки, некоторые реализации требуют наличия кавычек.

Заголовок SOAPAction является лишь подсказкой процессорам. Вся существенная информация, относящаяся к сообщению, передается в оболочке.

R2744 **СООБЩЕНИЕ** *запроса HTTP ДОЛЖНО содержать поле заголовка HTTP SOAPAction с заключенным в кавычки значением, равным значению атрибута soapAction элемента soapbind:operation, при его наличи в соответствующем описании WSDL.*

R2745 **СООБЩЕНИЕ** *запроса HTTP ДОЛЖНО содержать поле заголовка HTTP SOAPAction с заключенной в кавычки пустой строкой, если в соответствующем описании WSDL soapAction в soapbind:operation либо отсутствует, либо имеется с пустой строкой в качестве значения.*

Подробнее о SOAPAction см. в R1119 и относящихся к нему требованиях.

Например.

ПРАВИЛЬНО:

Описание WSDL, которое содержит:

```
<soapbind:operation soapAction="foo" />
```

приведет к сообщению с полем SOAPAction заголовка HTTP вида:

```
SOAPAction: "foo"
```

ПРАВИЛЬНО:

Описание WSDL, которое содержит:

```
<soapbind:operation />
```

или
<soapbind:operation soapAction="" />
приведет к сообщению с полем SOAPAction заголовка HTTP вида:
SOAPAction: ""

4.7.26 Расширения привязки SOAP

Атрибут wsdl:required часто понимают ошибочно и авторы WSDL иногда неправильно используют его для указания факультативности soapbind:header. Атрибут wsdl:required, по WSDL1.1, является методом расширения, предназначенным для процессоров WSDL. Он позволяет удобно вводить новые элементы расширения WSDL. Атрибут wsdl:required должен сигнализировать процессору WSDL, должен ли процессор распознавать и понимать элемент расширения для правильной обработки описания WSDL. Он не указывает на условность или факультативность какой-либо конструкции, входящей в оболочку. Например, атрибут wsdl:required со значением «false» в элементе soapbind:header не следует интерпретировать как сигнал WSDL процессору, что описанный блок заголовка SOAP является условным или факультативным в оболочках, генерируемых по этому описанию WSDL. Его нужно интерпретировать следующим образом: «для отправки в конечную точку оболочки, в описании которой содержится элемент soapbind:header, процессор WSDL не ДОЛЖЕН понимать семантику, подразумеваемую элементом soapbind:header».

Значением по умолчанию для атрибута wsdl:required элементов расширения привязки SOAP в WSDL 1.1 является «false». На практике в большинстве описаний WSDL атрибут wsdl:required не специфицирован для элементов расширения привязки SOAP, что может быть интерпретировано процессорами WSDL в том смысле, что элементы расширения могут быть проигнорированы. В Профиле требуется, чтобы все расширения SOAP WSDL 1.1 были поняты и обработаны потребителем независимо от наличия и значения атрибута wsdl:required в элементе расширения.

R2747 **ПОТРЕБИТЕЛЬ** *ДОЛЖЕН понимать и обрабатывать все элементы расширения привязки SOAP WSDL 1.1 независимо от наличия или отсутствия атрибута wsdl:required в элементе расширения и независимо от значения атрибута wsdl:required при его наличии.*

R2748 **ПОТРЕБИТЕЛЬ** *НЕ ДОЛЖЕН интерпретировать наличие атрибута wsdl:required в элементе расширения soapbind со значением «false» в том смысле, что элемент расширения является факультативным в оболочках, генерируемых по описанию WSDL.*

4.8 Использование Схемы XML

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

- XML Schema Part 1: Structures
- XML Schema Part 2: Datatypes

В WSDL 1.1 Схема XML используется как одна из систем типов. В Профиле установлена обязательность использования Схемы XML в качестве системы типов для описаний WSDL сетевых услуг.

R2800 **ОПИСАНИЕ** *МОЖЕТ использовать любую конструкцию из Схемы XML 1.0.*

R2801 **ОПИСАНИЕ** *ДОЛЖНО использовать XML Schema 1.0 Recommendation как основу для определенных пользователем типов данных и структур.*

5 Публикация и обнаружение услуги

Когда требуется публикация или обнаружение услуги, в Профиле принят механизм UDDI для описания поставщиков сетевых услуг и услуг, ими предоставляемых. Описания области применения, использования и типа сетевой услуги даются в терминах UDDI; подробные технические описания даются в терминах WSDL. В случаях, когда в двух спецификациях определены перекрывающиеся описательные данные и используются обе формы описания, в Профиле требуется, чтобы описания не были противоречивыми.

Регистрация экземпляра сетевой услуги в регистрах UDDI является факультативной. Нет смысла обеспечивать метаданные и обнаружение через UDDI всех используемых сценариев, но когда такая возможность необходима, UDDI является утвержденным методом.

Сетевые услуги, образующие UDDI V2, не полностью соответствуют Профилю 1.0, поскольку они не принимают сообщения, оболочки которых закодированы с помощью UTF-8 и UTF-16, как требуется Профилем. (Они принимают только UTF-8.) Это расхождение вызвано тем, что UDDI V2 был спроектирован и, во многих случаях, реализован до разработки Профиля. Разработчики UDDI V2 знают об этом несоответствии и примут его во внимание в дальнейшей работе.

В настоящем разделе Профиля использованы ссылки на следующие спецификации:

• UDDI, Версия 2.04, Спецификация API (UDDI Version 2.04 API Specification, Dated 19 July 2002)[1];

• UDDI, Версия 2.03, Базовая структура данных (UDDI Version 2.03 Data Structure Reference, Dated 19 July 2002)[2];

• UDDI Версия 2, Схема XML (UDDI Version 2 XML Schema)[3].

5.1 bindingTemplates

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

UDDI Version 2.03 Data Structure Reference, раздел 7.

В UDDI экземпляры сетевых услуг представлены в виде элементов uddi:bindingTemplate. Элемент uddi:bindingTemplate играет роль, аналогичную wsdl:port, но предоставляет опции, которые не могут быть выражены в WSDL. Для сохранения описания WSDL экземпляра услуги и его согласованности с описанием UDDI в профиле установлены следующие ограничения на конструкцию элементов uddi:bindingTemplate.

В элементе WSDL soapbind:address требуется, чтобы был явно специфицирован сетевой адрес экземпляра услуги. Напротив, в UDDI V2 предоставлено две альтернативы для спецификации сетевого адреса представляемой услуги. Одна из них, uddi:accessPoint, зеркально отражает метод WSDL, непосредственно специфицируя адрес. Другая, uddi:hostingRedirector, предоставляет для разрешения адреса основанный на сетевой услуге метод перенаправления и не согласуется с методом WSDL.

R3100 **REGDATA** *типа uddi:bindingTemplate, представляющего соответствующий ЭКЗЕМПЛЯР, ДОЛЖНЫ содержать элемент uddi:accessPoint.*

Например.

НЕПРАВИЛЬНО:

```
<bindingTemplate bindingKey="...">
   <description xml:lang="EN">BarSOAPPort</description>
   <hostingRedirector bindingKey="..."/>
<tModelInstanceDetails>
   ...
</tModelInstanceDetails>
</bindingTemplate>
```

ПРАВИЛЬНО:

```
<bindingTemplate bindingKey="...">
     <description xml:lang="EN">BarSOAPPort</description>
     <accessPoint>http://example.org/myBarSOAPPort</accessPoint>
     <tModelInstanceDetails>
        ...
     </tModelInstanceDetails>
</bindingTemplate>
```

5.2 tModels

В настоящем разделе Профиля использованы ссылки на следующие спецификации (или их разделы):

• UDDI Version 2.03 Data Structure Reference, раздел 8

В UDDI типы сетевых услуг представлены в виде элементов uddi:tModel. (См. UDDI Data Structures section 8.1.1.) Может существовать, но не обязательно, точка (использующая URI) для документа, который содержит фактическое описание. В UDDI неизвестен метод, использованный для описания типа сетевой услуги. В Профиле это должно быть известно потому, что интерпретация будет очень сложной, если типы сетевых услуг не имеют описаний или если описания могут иметь произвольный вид.

В UDDI API Specification, приложение I.1.2.1.1 допускаются, но не требуются элементы uddi:tModel, которые в качестве языка описания используют WSDL для описания представленного типа сетевой услуги. Это приводит к проблемам интероперабельности, так как непонятно, какой язык описания следует использовать.

---

[1] http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm.

[2] http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm.

[3] http://uddi.org/schema/uddi_v2.xsd.

Поэтому в Профиле установлены следующие ограничения на то, как могут быть устроены элементы uddi:tModel, которые описывают типы сетевых услуг:

В Профиле в качестве языка описания выбран WSDL потому, что он является наиболее широко распространенным языком такого рода.

R3002 **REGDATA** *типа uddi:tModel, представляющего соответствующий тип сетевой услуги, ДОЛЖНЫ использовать WSDL в качестве языка описания.*

Для спецификации того, что соответствующий тип сетевой услуги использует WSDL, в Профиле принята категоризация UDDI для подобных утверждений.

R3003 **REGDATA** *типа uddi:tModel, представляющего соответствующий тип сетевой услуги, ДОЛЖНЫ быть категоризированы с помощью таксономии uddi:types и категоризации «wsdlSpec».*

Относительно uddi:overviewURL в uddi:tModel для разрешения до wsdl:binding, в Профиле должно быть принято соглашение для различия между кратными wsdl:binding в документе WSDL. Хорошая практика UDDI использования WSDL в Регистре UDDI специфицирует наиболее широко признаваемое из таких соглашений.

R3010 **REGDATA** *типа uddi:tModel, представляющего соответствующий тип сетевой услуги, ДОЛЖНЫ следовать Версии 1.08 хорошей практики UDDI использования WSDL в Регистре UDDI (V1.08 of the UDDI Best Practice for Using WSDL in a UDDI Registry)*[1)]

Если wsdl:binding, указанный элементом uddi:tModel, не соответствует Профилю, то возникнет несогласованность.

R3011 *Элемент wsdl:binding, указанный* **REGDATA** *типа uddi:tModel ДОЛЖЕН соответствовать Профилю.*

6 Безопасность

Как и для всех сетевых информационных технологий, вопрос безопасности является критическим для сетевых услуг. Для сетевых услуг, как и для других информационных технологий, безопасность состоит в понимании потенциальных угроз, которые могут возникнуть, и принятии операционных, физических и технологических контрмер для уменьшения риска успешной атаки до приемлемого уровня. Так как «приемлемый уровень риска» и стоимость контрмер широко изменяются в зависимости от приложения, то не может быть универсального «правильного ответа» для безопасности сетевых услуг. Выбор правильного баланса контрмер и приемлемого риска возможен только для каждого конкретного случая.

Существуют общие образцы контрмер, для которых опыт показал снижение риска до приемлемого уровня для многих сетевых услуг. В Профиле рекомендуются, но не являются обязательными, наиболее широко применяемые из них:

обеспечение безопасности HTTP с помощью TLS 1.0 или SSL 3.0 (HTTPS). Это означает, что соответствующие сетевые услуги могут использовать HTTPS; они могут использовать и другие контрмеры или не использовать никакие.

HTTPS рассматривается как зрелый стандарт зашифрованной связи для обеспечения базового уровня конфиденциальности. Таким образом, HTTPS образует первый и простейший способ достижения некоторых базовых характеристик безопасности, которые требуются многими реальными приложениями сетевых услуг. HTTPS также может быть использован для обеспечения аутентификации клиента с применением сертификатов на стороне клиента.

В настоящем разделе Профиля использованы ссылки на следующие спецификации и определены точки их расширения:

- RFC2818: HTTP через TLS (RFC2818: HTTP Over TLS)[2)];
- RFC2246: Протокол TLS, версия 1.0 (RFC2246: The TLS Protocol Version 1.0)[3)]

Точки расширения.

E0019 — Цифровой набор TLS — TLS допускает использование произвольных алгоритмов шифрования.

E0020 — Расширения TLS — TLS допускает расширения на фазе рукопожатия.

[1)] http://www.oasis-open.org/committees/uddi-spec/doc/bp/uddi-spec-tc-bp-using-wsdl-v108-20021110.htm.

[2)] http://www.ietf.org/rfc/rfc2818.txt.

[3)] http://www.ietf.org/rfc/rfc2246.txt.

• Протокол SSL, версия 3.0 (The SSL Protocol Version 3.0)[1]

Точки расширения:

• E0021 — Цифровой набор SSL — SSL допускает использование произвольных алгоритмов шифрования.

• RFC2459: Сертификат инфраструктуры публичного ключа Интернет X.509 и Профиль CRL (RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile)[2]

Точки расширения:

• E0022 — Уполномоченный по сертификатам — Выбор уполномоченного по сертификации является частным соглашением между сторонами.

• E0023 — Расширения сертификатов — X509 допускает произвольные расширения сертификатов.

6.1 Использование HTTPS

HTTPS является настолько полезным, широко принятым базовым методом безопасности, что Профиль должен его допускать.

R5000 **ЭКЗЕМПЛЯР** *МОЖЕТ требовать использования HTTPS.*

R5001 *Если* **ЭКЗЕМПЛЯР** *требует использования HTTPS, то атрибут location элемента soapbind:address в описании wsdl:port ДОЛЖЕН быть равен URI схемы «https»; в противном случае ДОЛЖЕН быть равен URI схемы «http».*

Простой HTTPS обеспечивает аутентификацию экземпляра сетевой услуги потребителем, но не аутентификацию потребителя экземпляром. Для многих экземпляров такой уровень риска слишком высок, чтобы допустить взаимодействие. Включение возможности взаимной аутентификации HTTPS в Профиль позволяет экземплярам использовать контрмеры аутентификации потребителя. В случаях, когда аутентификации экземпляра потребителем недостаточно, это часто достаточно снижает риск для того, чтобы разрешить взаимодействие.

R5010 **ЭКЗЕМПЛЯР** *МОЖЕТ требовать использования HTTPS со взаимной аутентификацией.*

---

[1] http://wp.netscape.com/eng/ssl3/draft302.txt.

[2] http://www.ietf.org/rfc/rfc2459.txt.

Приложение А

Ссылочные спецификации

В Профиль через ссылку включены требования следующих спецификаций, за исключением тех, которые были изменены в самом Профиле:

- Простой протокол доступа к объекту [Simple Object Access Protocol (SOAP)] 1.1[1]
- RFC2616: Протокол передачи гипертекста (RFC2616: Hypertext Transfer Protocol — HTTP/1.1)[2]
- RFC2965: Метод управления состоянием HTTP (RFC2965: HTTP State Management Mechanism)[3]
- Расширяемый язык разметки [Extensible Markup Language (XML) 1.0 (Second Edition)][4]
- Пространства имен в XML (Namespaces in XML 1.0)[5]
- Схема XML. Часть 1: Структуры (XML Schema Part 1: Structures)[6]
- Схема XML. Часть 2: Типы данных (XML Schema Part 2: Datatypes)[7]
- Язык описания сетевых услуг [Web Services Description Language (WSDL) 1.1][8]
- UDDI, версия 2.04, Спецификация API (UDDI Version 2.04 API Specification, Dated 19 July 2002)[9]
- UDDI, версия 2.03, Базовая структура данных (UDDI Version 2.03 Data Structure Reference, Dated 19 July 2002)[10].
- UDDI, версия 2, Схема XML (UDDI Version 2 XML Schema)[11]
- RFC2818: HTTP через TLS (RFC2818: HTTP Over TLS)[12]
- RFC2246: Протокол TLS, версия 1.0 (RFC2246: The TLS Protocol Version 1.0)[13]
- Протокол SSL, версия 3.0 (The SSL Protocol Version 3.0)[14]
- RFC2459: Сертификат инфраструктуры публичного ключа Интернет X.509 и Профиль CRL (RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile)[15]

---

[1] http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.
[2] http://www.ietf.org/rfc/rfc2616.txt.
[3] http://www.ietf.org/rfc/rfc2965.txt.
[4] http://www.w3.org/TR/2000/REC-xml-20001006.
[5] http://www.w3.org/TR/1999/REC-xml-names-19990114.
[6] http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/.
[7] http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.
[8] http://www.w3.org/TR/2001/NOTE-wsdl-20010315.
[9] http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm.
[10] http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm.
[11] http://uddi.org/schema/uddi_v2.xsd.
[12] http://www.ietf.org/rfc/rfc2818.txt.
[13] http://www.ietf.org/rfc/rfc2246.txt.
[14] http://wp.netscape.com/eng/ssl3/draft302.txt.
[15] http://www.ietf.org/rfc/rfc2459.txt.

Приложение В

Точки расширения

В настоящем приложении идентифицированы точки расширения для спецификаций, входящих в Профиль.

Указанные в этих точках методы не входят в область применения Профиля; их использование может повлиять на интероперабельность и потребовать дополнительного частного соглашения между сторонами сетевой услуги.

В Простом протоколе доступа к объекту [Simple Object Access Protocol (SOAP)] 1.1[1)]

• E0001 — Блоки заголовков — Блоки заголовков являются главными методами расширения в SOAP.

• E0002 — Порядок обработки — Порядок обработки компонентов оболочки SOAP (например, заголовков) не специфицирован и, следовательно, должен быть согласован дополнительно.

• E0003 — Использование посредников — Посредники SOAP являются не специфицированным в SOAP 1.1 методом, и их использование может потребовать дополнительного согласования. Их использование также может вызвать необходимость тщательного рассмотрения при оценке соответствия Профилю.

• E0004 — Значения soap:actor — Значения атрибута soap:actor, отличные от специального uri «http://schemas.xmlsoap.org/soap/actor/next», представляют собой частное соглашение между сторонами сетевой услуги.

• E0005 — Подробности отказа — Содержимое элемента Подробности отказа не установлены в SOAP 1.1.

• E0006 — Сериализация оболочки — Профиль не ограничивает некоторые аспекты представления оболочки в виде сообщения.

В Протоколе передачи гипертекста (RFC2616: Hypertext Transfer Protocol — HTTP/1.1)[2)]

• E0007 — Аутентификация HTTP — Аутентификация HTTP допускает схемы расширения, произвольные цифровые хэш-алгоритмы и параметры.

• E0008 — Неспецифицированные поля заголовка — HTTP допускает появление в сообщениях произвольных заголовков.

• E0009 — Расширения ожидания — Механизм Ждать/Продолжить в HTTP допускает расширения ожидания.

• E0010 — Кодирование содержимого — Множество кодирований содержимого, допускаемых HTTP, является открытым и любое кодирование, кроме «gzip», «compress» или «deflate», является точкой расширения.

• E0011 — Кодирование передачи — Множество кодирований передачи, допускаемых HTTP, является открытым.

• E0012 — Обновление — HTTP позволяет изменять соединение на произвольный протокол, используя заголовок Upgrade (обновить).

• E0024 — Атрибуты пространства имен — Атрибуты пространства имен элементов soap:Envelope и soap:Header.

• E0025 — Атрибуты элементов soap:Body — SOAP 1.1 не устанавливает ограничений ни на пространство имен, ни на локальные атрибуты.

В Схеме XML. Часть 1: Структуры (XML Schema Part 1: Structures)[3)]:

• E0017 — Аннотации схемы — Схема XML допускает аннотации, которые могут переносить дополнительную информацию о структурах данных.

В Языке описания сетевых услуг [Web Services Description Language (WSDL) 1.1][4)]:

• E0013 — Расширения WSDL — WSDL допускает в определенных местах расширения элементов и атрибутов; использование таких расширений требует стороннего соглашения.

• E0014 — Режим валидации — осуществляет или нет синтаксический анализатор, используемый для чтения документов WSDL и Схемы XML, валидацию DTD.

• E0015 — Привлечение внешних ресурсов — привлекает или нет синтаксический анализатор, используемый для чтения документов WSDL и Схемы XML, внешние категории и DTD.

• E0016 — Относительные URI — в WSDL нет адекватной спецификации использования относительных URI для следующих конструкций: soapbind:body/@namespace, soapbind:address/@location, wsdl:import/@location, xsd:schema/@targetNamespace и xsd:import/@schemaLocation. Их использование может потребовать дополнительной координации; подробнее см. XML Base.

---

1) http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

2) http://www.ietf.org/rfc/rfc2616.txt.

3) http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/.

4) http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

В Протоколе TLS, версия 1.0 (RFC2246: The TLS Protocol Version 1.0)[1]:

• E0019 — Цифровой набор TLS — TLS допускает использование произвольных алгоритмов шифрования.

• E0020 — Расширения TLS — TLS допускает расширения на фазе рукопожатия.

В Протоколе SSL, версия 3.0 (The SSL Protocol Version 3.0)[2]:

• E0021 — Цифровой набор SSL — SSL допускает использование произвольных алгоритмов шифрования.

В Сертификате инфраструктуры публичного ключа Интернет X.509 и Профиле CRL (RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile)[3]:

• E0022 — Уполномоченный по сертификатам — Выбор уполномоченного по сертификации является частным соглашением между сторонами.

• E0023 — Расширения сертификатов — X509 допускает произвольные расширения сертификатов.

---

[1] http://www.ietf.org/rfc/rfc2246.txt.

[2] http://wp.netscape.com/eng/ssl3/draft302.txt.

[1] http://www.ietf.org/rfc/rfc2459.txt

Приложение С

Нормативные ссылки

В дополнение к вошедшим в Профиль спецификациям, перечисленным в Приложении А, в настоящем стандарте приведены ссылки на следующие спецификации:

• RFC2119 Ключевые слова, используемые в RFC для указания уровня требований (RFC2119, http://ietf.org/rfc/rfc2119 Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997)

• Базовый Профиль WS-I версии 1.0 (WS-I Basic Profile 1.0, http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html, K. Ballinger et al., April 2004)

• Пространства имен в XML 1.0 [Namespaces in XML 1.0 (Second Edition), http://www.w3.org/TR/2006/REC-xml-names-20060816, T. Bray et al., August 2006]

• Методы присоединения заявления о соответствии WS-I, версия 1.0 (WS-I Conformance Claim Attachment Mechanisms Version 1.0, http://www.ws-i.org/Profiles/ConformanceClaims-1.0-2004-11-15.html, M. Nottingham et al., November 2004)

Приложение D

Определенные термины

Перечисленные ниже термины имеют специфические определения, принятые в настоящем Профиле:

привязка rpc-литерал

«Привязка rpc-литерал» есть элемент wsdl:binding, все дочерние элементы wsdl:operation которого являются операциями rpc-литерал.

«Операция rpc-литерал» есть дочерний элемент wsdl:operation элемента wsdl:binding, потомки которого soapbind:body специфицируют атрибут use со значением «literal» и:

1.либо атрибут style со значением «rpc» специфицирован в дочернем элементе soapbind:operation,

2.либо атрибут style отсутствует в дочернем элементе soapbind:operation и элемент soapbind:binding в охватывающем элементе wsdl:binding специфицирует атрибут style со значением «rpc».

привязка документ-литерал

«Привязка документ-литерал» есть элемент wsdl:binding, все дочерние элементы wsdl:operation которого есть операции документ-литерал.

«Операция документ-литерал» есть дочерний элемент wsdl:operation элемента wsdl:binding, потомки которого soapbind:body специфицируют атрибут use со значением «literal» и:

1.либо атрибут style со значением «document» специфицирован в дочернем элементе soapbind:operation,

2.либо атрибут style отсутствует в дочернем элементе soapbind:operation и элемент soapbind:binding в охватывающем элементе wsdl:binding специфицирует атрибут style со значением «document»,

3.либо атрибут style отсутствует как в дочернем элементе soapbind:operation, так и в элементе soapbind:binding в охватывающем элементе wsdl:binding.

сигнатура операции

В Профиле «сигнатура операции» определена как полностью квалифицированное имя дочернего элемента тела SOAP входного сообщения SOAP, описанное операцией в привязке WSDL.

В случае привязки rpc-литерал имя операции используют как оболочку для присоединенных частей. В случае привязки документ-литерал, так как оболочка с именем операции отсутствует, сигнатуры сообщений должны быть корректно спроектированы так, чтобы удовлетворять настоящим требованиям.

Приложение Е

Члены Рабочей группы по базовому профилю WS-I

Настоящий документ является результатом работы Рабочей группы по базовому профилю WS-I (WS-I Basic Profile Working Group), членами которой являются:

Mark Allerton (Crystal Decisions Corp), Steve Anderson (OpenNetwork), George Arriola (Talking Blocks, Inc.), Siddharth Bajaj (Verisign), Keith Ballinger (Microsoft Corp.), David Baum (Kantega AS), Ilya Beyer (KANA), Rich Bonneau (IONA Technologies), Don Box (Microsoft Corp.), Andrew Brown (Verisign), Heidi Buelow (Quovadx), David Burdett (Commerce One, Inc.), Luis Felipe Cabrera (Microsoft Corp.), Maud Cahuzac (France Telecom), Mike Chadwick (Kaiser Permanente), Martin Chapman (Oracle Corporation), Richard Chennault (Kaiser Permanente), Roberto Chinnici (Sun Microsystems), Dipak Chopra (SAP AG), Jamie Clark (OASIS), David Cohen (Merrill Lynch), Ugo Corda (SeeBeyond Tech), Paul Cotton (Microsoft Corp.), Joseph Curran (Accenture), Alex Deacon (Verisign), Mike DeNicola (Fujitsu Limited), Paul Downey (BT Group), Jacques Durand (Fujitsu Limited), Aladin Eajani (Hummingbird, Ltd.), Michael Eder (Nokia), Dave Ehnebuske (IBM), Mark Ericson (Mindreef Inc), Colleen Evans (Microsoft Corp.), Tim Ewald (Microsoft Corp.), Chuck Fay (FileNET Corp.), Chris Ferris (IBM), Daniel Foody (Actional Corporation), Satoru Fujita (NEC Corporation), Shishir Garg (France Telecom), Yaron Goland (BEA Systems Inc), Marc Goodner (SAP AG), Pierre Goyette (Hummingbird, Ltd.), Hans Granqvist (Verisign), Martin Gudgin (Microsoft Corp.), Marc Hadley (Sun Microsystems), Norma Hale (Webify Solutions Inc), Bob Hall (Unisys Corporation), Scott Hanselman (Corillian), Muir Harding (Autodesk Inc.), Loren Hart (Verisign), Andrew Hately (IBM), Harry Holstrom (Accenture), Lawrence Hsiung (Quovadx), Hemant Jain (Tata Consultancy), Steve Jenisch (SAS Institute), Erik Johnson (Epicor Software), Bill Jones (Oracle Corporation), Anish Karmarkar (Oracle Corporation), Dana Kaufman (Forum Systems), Takahiro Kawamura (Toshiba), Oldre Kepka (Systinet), Bhushan Khanal (WRQ Inc.), Sandy Khaund (Microsoft Corp.), Jacek Kopecky (Systinet), Sanjay Krishnamurthi (Informatica), Sundar Krishnamurthy (Verisign), Eva Kuiper (Hewlett-Packard), Sunil Kunisetty (Oracle Corporation), Christopher Kurt (Microsoft Corp.), Lars Laakes (Microsoft Corp.), Canyang Kevin Liu (SAP AG), Ted Liu (webMethods Inc.), Donna Locke (Oracle Corporation), Brad Lund (Intel), Michael Mahan (Nokia), Ron Marchi (EDS), Jonathan Marsh (Microsoft Corp.), Eric Matland (Hummingbird, Ltd.), Barbara McKee (IBM), Derek Medland (Hummingbird, Ltd.), David Meyer (Plumtree Software Inc.), Jeff Mischkinsky (Oracle Corporation), Ray Modeen (MITRE Corp.), Tom Moog (Sarvega Inc.), Gilles Mousseau (Hummingbird, Ltd.), Greg Mumford (MCI), Jim Murphy (Mindreef Inc), Bryan Murray (Hewlett-Packard), Richard Nikula (BMC Software, Inc.), Eisaku Nishiyama (Hitachi, Ltd.), Mark Nottingham (BEA Systems Inc), David Orchard (BEA Systems Inc), Vivek Pandey (Sun Microsystems), Jesse Pangburn (Quovadx), Eduardo Pelegri-Llopart (Sun Microsystems), Mike Perham (Webify Solutions Inc), Eric Rajkovic (Oracle Corporation), Shaan Razvi (MITRE Corp.), Rimas Rekasius (IBM), Mark Richards (Fidelity), Graeme Riddell (Bowstreet), Sam Ruby (IBM), Tom Rutt (Fujitsu Limited), Saikat Saha (Commerce One, Inc.), Roger Sanborn (Crystal Decisions Corp), Matt Sanchez (Webify Solutions Inc), Krishna Sankar (Cisco Systems Inc.), Jeffrey Schlimmer (Microsoft Corp.), Don Schricker (Micro Focus), Dave Seidel (Mindreef Inc), AKIRA SHIMAYA (NTT), David Shoaf (Hewlett-Packard), Yasser Shohoud (Microsoft Corp.), David Smiley (Ascential Software), Seumas Soltysik (IONA Technologies), Joseph Stanko (Plumtree Software Inc.), Andrew Stone (Accenture), Julie Surer (MITRE Corp.), YASUO TAKEMOTO (NTT), Nobuyoshi Tanaka (NEC Corporation), Jorgen Thelin (Microsoft Corp.), Sameer Vaidya (Talking Blocks, Inc.), William Vambenepe (Hewlett-Packard), Claus von Riegen (SAP AG), Rick Weil (Eastman Kodak Company), Scott Werden (WRQ Inc.), Ajamu Wesley (IBM), Ian White (Micro Focus), Dave Wilkinson (Vignette), Mark Wood (Eastman Kodak Company), Prasad Yendluri (webMethods Inc.) и Brandon Zhu (NetManage Inc).

УДК 681.3.01:006.354 ОКС 35.100.05 П85

Ключевые слова: интероперабельность, спецификация сетевых услуг, атрибут, базовый профиль

Редактор *Е. В. Вахрушева*
Технический редактор *В. Н. Прусакова*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *В. Н. Романовой*

Сдано в набор 22.12.2013. Подписано в печать 19.03.2014. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 4,65. Уч.-изд. л. 4,00. Тираж 76 экз. Зак. 1.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.