

Международный союз электросвязи

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

X.782

(05/2012)

СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ,
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ
И БЕЗОПАСНОСТЬ

Управление в ВОС –
Функции общего управления и функции ODMA

**Руководящие указания по определению
веб-услуг для управляемых объектов
и интерфейсов управления**

Рекомендация МСЭ-Т X.782

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ X

СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ

СЕТИ ПЕРЕДАЧИ ДАННЫХ ОБЩЕГО ПОЛЬЗОВАНИЯ	
Службы и услуги	X.1–X.19
Интерфейсы	X.20–X.49
Передача, сигнализация и коммутация	X.50–X.89
Сетевые аспекты	X.90–X.149
Техническое обслуживание	X.150–X.179
Административные предписания	X.180–X.199
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ	
Модель и обозначение	X.200–X.209
Определения служб	X.210–X.219
Спецификации протоколов с установлением соединений	X.220–X.229
Спецификации протоколов без установления соединений	X.230–X.239
Проформы PICS	X.240–X.259
Идентификация протоколов	X.260–X.269
Протоколы обеспечения безопасности	X.270–X.279
Управляемые объекты уровня	X.280–X.289
Испытание на соответствие	X.290–X.299
ВЗАИМОДЕЙСТВИЕ МЕЖДУ СЕТЯМИ	
Общие положения	X.300–X.349
Спутниковые системы передачи данных	X.350–X.369
Сети, основанные на протоколе Интернет	X.370–X.379
СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ	
СПРАВОЧНИК	
ОРГАНИЗАЦИЯ СЕТИ ВОС И СИСТЕМНЫЕ АСПЕКТЫ	
Организация сети	X.600–X.629
Эффективность	X.630–X.639
Качество обслуживания	X.640–X.649
Наименование, адресация и регистрация	X.650–X.679
Абстрактно-синтаксическая нотация 1 (ASN.1)	X.680–X.699
УПРАВЛЕНИЕ В ВОС	
Структура и архитектура управления системами	X.700–X.709
Служба и протокол связи для общего управления	X.710–X.719
Структура управляющей информации	X.720–X.729
Функции общего управления и функции ODMA	X.730–X.799
БЕЗОПАСНОСТЬ	
ПРИЛОЖЕНИЯ ВОС	
Фиксация, параллельность и восстановление	X.850–X.859
Обработка транзакций	X.860–X.879
Удаленные операции	X.880–X.889
Общие приложения ASN.1	X.890–X.899
ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА	
БЕЗОПАСНОСТЬ ИНФОРМАЦИИ И СЕТЕЙ	
БЕЗОПАСНЫЕ ПРИЛОЖЕНИЯ И УСЛУГИ	
БЕЗОПАСНОСТЬ КИБЕРПРОСТРАНСТВА	
БЕЗОПАСНЫЕ ПРИЛОЖЕНИЯ И УСЛУГИ	
ОБМЕН ИНФОРМАЦИЕЙ, КАСАЮЩЕЙСЯ КИБЕРБЕЗОПАСНОСТИ	

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

Рекомендация МСЭ-Т X.782

Руководящие указания по определению веб-услуг для управляемых объектов и интерфейсов управления

Резюме

Рекомендация МСЭ-Т X.782 определяет набор руководящих принципов моделирования управляемых объектов и интерфейсов управления для управления сетями на основе веб-услуг. Вместе с Рекомендацией МСЭ-Т Q.818 она составляет платформу интерфейсов управления сетями на основе веб-услуг. В ней поясняется, как следует определять сервисно-ориентированные интерфейсы веб-услуг. Описываются прикладные сценарии, рассчитанные и не рассчитанные на применение веб-услуг в интерфейсах управления сетями, универсальные методы доступа к управляемым объектам на основе XML, а также моделирование информации на языке описания веб-услуг (WSDL) и в XML-схеме. Приводятся WSDL-определения и XML-схемы для описания некоторых основных типов данных: универсального управляемого объекта (МО) и универсальных методов доступа к МО. Настоящая Рекомендация вместе с Рекомендацией МСЭ-Т Q.818 составляет платформу интерфейсов управления сетями на основе веб-услуг с широким спектром приложений.

Хронологическая справка

Издание	Рекомендация	Утверждение	Исследовательская комиссия
1.0	МСЭ-Т X.782	14.05.2012 г.	2-я

Ключевые слова

Распределенная обработка, расширяемый язык разметки (XML), управляемые объекты, интерфейсы управления сетями, веб-услуги (WS), язык описания веб-услуг (WSDL), XML-схема.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи и информационно-коммуникационных технологий (ИКТ). Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним в целях стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения исследовательскими комиссиями МСЭ-Т, которые в свою очередь вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации осуществляется на добровольной основе. Однако данная Рекомендация может содержать некоторые обязательные положения (например, для обеспечения функциональной совместимости или возможности применения), и в таком случае соблюдение Рекомендации достигается при выполнении всех указанных положений. Для выражения требований используются слова "следует", "должен" (shall) или некоторые другие обязывающие выражения, такие как "обязан" (must), а также их отрицательные формы. Употребление таких слов не означает, что от какой-либо стороны требуется соблюдение положений данной Рекомендации.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или выполнение настоящей Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, действительности или применимости заявленных прав интеллектуальной собственности, независимо от того, доказываются ли такие права членами МСЭ или другими сторонами, не относящимися к процессу разработки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ не получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения настоящей Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что вышесказанное может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: <http://www.itu.int/ITU-T/ipr/>.

© ITU 2018

Все права сохранены. Ни одна из частей данной публикации не может быть воспроизведена с помощью каких бы то ни было средств без предварительного письменного разрешения МСЭ.

Содержание

Стр.

1	Сфера применения	1
2	Справочные документы	1
3	Определения	2
3.1	Термины, определенные в других документах.....	2
3.2	Термины, определенные в настоящей Рекомендации	2
4	Сокращения и акронимы	2
5	Условные обозначения	3
6	Обзор платформы управления на основе веб-услуг	4
7	Принципы проектирования сервисно-ориентированного WSDL-интерфейса	4
8	Определение универсального управляемого объекта с использованием XML-схемы	5
8.1	Роль веб-услуг в интерфейсах управления	5
8.2	Определение управляемых объектов с использованием XML-схемы	6
9	Методы доступа к управляемым объектам.....	9
10	Наследование управляемых объектов и операций интерфейса.....	11
10.1	Наследование атрибутов управляемых объектов	11
10.2	Соображения, касающиеся наследования операций интерфейса	12
11	Указания по моделированию информации для интерфейсов на основе веб-услуг	12
11.1	Пространство имен (namespace).....	12
11.2	Сложный тип (complexType)	12
11.3	Атрибут (attribute).....	12
11.4	Запрос (request)	12
11.5	Ответ (response).....	13
11.6	Уведомление (notification)	13
11.7	Соглашения по именованию для МОС, пакетов, атрибутов и типов данных.....	13
12	Идиомы проектирования для спецификаций XML-схемы.....	14
12.1	Модель данных с использованием XML-схемы	14
12.2	Вопросы проектирования XML-схемы.....	14
12.3	Рекомендации для разработчиков схем	16
12.4	Рекомендации по расширению схемы	18
13	Согласованность и соответствие	19
13.1	Согласованность документов по стандартам.....	19
13.2	Соответствие системы.....	19
13.3	Рекомендации в отношении заявления о соответствии	19
Приложение А.	Общие определения WSDL и XML-схемы	20
А.1	Определения XML-схемы для общих типов данных и универсального управляемого объекта	20

	Стр.
А.2 Определение WSDL и XML-схемы для общих методов доступа к объектам ..	28
Дополнение I. Обзор технологий и прикладных сценариев на основе веб-услуг в интерфейсах управления сетями	34
I.1 Характеристики технологий на основе веб-услуг	34
I.2 Прикладные сценарии, рассчитанные и не рассчитанные на применение веб-услуг в интерфейсах управления сетями.....	35
Библиография	37

Рекомендация МСЭ-Т X.782

Руководящие указания по определению веб-услуг для управляемых объектов и интерфейсов управления

1 Сфера применения

Архитектура управления сетями, определенная в [ITU-T M.3010], предполагает использование нескольких протоколов управления. До сих пор возможными вариантами на уровне приложения остаются GDMO/CMIP и GIOP/IIOP CORBA. Согласно методике спецификации интерфейса управления, определенной в [ITU-T M.3020], в интерфейсы управления сетями можно вводить больше технологических парадигм, и в настоящее время веб-услуги/XML становятся дополнительной парадигмой управления сетями.

Настоящая Рекомендация вместе с [ITU-T Q.818] устанавливает платформу для определения того, как следует моделировать интерфейсы, поддерживаемые системами управления и элементами сети с использованием веб-услуг/XML-схемы. В настоящей Рекомендации представлены руководящие указания и инструкции по следующим вопросам:

- сервисно-ориентированный подход к проектированию WSDL-интерфейсов;
- прикладные сценарии, рассчитанные и не рассчитанные на применение веб-услуг в интерфейсах управления сетями;
- универсальные методы доступа к управляемым объектам;
- наследование управляемых объектов и интерфейсов;
- указания по моделированию информации для интерфейсов на основе веб-услуг;
- условные обозначения стиля для спецификаций WSDL и XML-схем.

2 Справочные документы

Указанные ниже Рекомендации МСЭ-Т и другие источники содержат положения, которые путем ссылки на них в данном тексте составляют положения настоящей Рекомендации. На момент публикации указанные издания были действующими. Все Рекомендации и другие источники могут подвергаться пересмотру; поэтому всем пользователям настоящей Рекомендации предлагается изучить возможность применения последнего издания Рекомендаций и других источников, перечисленных ниже. Список действующих в настоящее время Рекомендаций МСЭ-Т регулярно публикуется. Ссылка на документ в настоящей Рекомендации не придает ему, как отдельному документу, статус Рекомендации

[ITU-T E.164]	Рекомендация МСЭ-Т E.164 (2010 г.), <i>Международный план нумерации электросвязи общего пользования.</i>
[ITU-T M.3010]	Recommendation ITU-T M.3010 (2000), <i>Principles for a telecommunications management network</i>
[ITU-T M.3020]	Рекомендация МСЭ-Т M.3020 (2011 г.), <i>Методика определения интерфейсов управления.</i>
[ITU-T M.3701]	Recommendation ITU-T M.3701 (2010), <i>Common management services – State management – Protocol neutral requirements and analysis.</i>
[ITU-T Q.818]	Recommendation ITU-T Q.818 (2012), <i>Web service-based management services.</i>
[ITU-T X.701]	Recommendation ITU-T X.701 (1997), <i>Information technology – Open Systems Interconnection – Systems management overview.</i>
[ITU-T X.703]	Recommendation ITU-T X.703 (1997), <i>Information technology – Open Distributed Management Architecture.</i>
[ATIS-I-000002]	ATIS Specification ATIS-I-000002 (2011), <i>ATIS XML Schema Development Guidelines.</i>

[OASIS WSN]	OASIS Specification (2006), <i>Web Services Base Notification v1.3</i> .
[OASIS UDDI]	OASIS Specification (2004), <i>Universal Description, Discovery and Integration (UDDI) v3.0.2</i> .
[W3C Primer]	W3C Recommendation (2004), <i>XML Schema Part 0: Primer Second Edition</i> .
[W3C SOAP]	W3C Recommendation (2007), <i>SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)</i> .
[W3C WSDL]	W3C Recommendation (2001), <i>Web Services Description Language (WSDL) 1.1</i> .
[W3C XML]	W3C Recommendation (2000), <i>Extensible Markup Language (XML) 1.0 Second Edition</i> .
[W3C XS-P1]	W3C Recommendation (2004), <i>XML Schema Part 1: Structures Second Edition</i> .
[W3C XS-P2]	W3C Recommendation (2004), <i>XML Schema Part 2: Datatypes Second Edition</i> .

3 Определения

3.1 Термины, определенные в других документах

В настоящей Рекомендации используются следующие термины, определенные в других документах:

3.1.1 агент (agent) [ITU-T M.3020].

3.1.2 класс управляемых объектов (managed object class) [ITU-T X.701].

3.1.3 менеджер (manager) [ITU-T M.3020].

3.1.4 уведомление (notification) [ITU-T X.703].

3.2 Термины, определенные в настоящей Рекомендации

В настоящей Рекомендации отсутствуют определения новых терминов.

4 Сокращения и акронимы

В настоящей Рекомендации используются следующие сокращения и акронимы.

B2B	Business to Business	Компания – компания
BPEL	Business Process Execution Language	Язык выполнения бизнес-процессов
C2B	Customer to Business	Клиент – компания
CMIP	Common Management Information Protocol	Протокол передачи общей управляющей информации
CORBA	Common Object Request and Broker Architecture	Общая архитектура брокера запроса объекта
DCOM	Distribute Component Object Model	Распределенная модель компонентных объектов
DN	Distinguished Name	Выделенное имя
DTD	Document Type Definition	Определение типа документа
EMS	Element Management System	Система управления элементами
GDMO	Guidelines for the Definition of Managed Objects	Руководство по определению управляемых объектов
GIOP	General Inter-ORB Protocol	Общий протокол между ORB
IDL	Interface Definition Language	Язык определения интерфейса
IIOP	Internet Inter-ORB Protocol	Интернет-протокол между ORB

IT	Information Technology	ИТ	Информационные технологии
LAN	Local Area Network		Локальная сеть
MO	Managed Object		Управляемый объект
MOC	Managed Object Class		Класс управляемых объектов
MOO	Multiple Object Operation		Операция над множеством объектов
NMS	Network Management System		Система управления сетью
OOAD	Object-Oriented Analysis and Design		Объектно-ориентированный анализ и проектирование
OS	Operating System	ОС	Операционная система
RDN	Relative Distinguished Name		Относительное выделенное имя
SOA	Service-Oriented Architecture		Сервисно-ориентированная архитектура
SOAP	Simple Object Access Protocol		Простой протокол доступа к объектам
TMN	Telecommunications Management Network		Сеть управления электросвязью
UDDI	Universal Description Discovery and Integration		Универсальное описание, поиск и взаимодействие
WS	Web Services		Веб-услуги
WSDL	Web Services Description Language		Язык описания веб-услуг
WSN	Web Services Notification		Уведомление на основе веб-услуг
XML	Extensible Markup Language		Расширяемый язык разметки
XSD	XML Schema Definition		Определение XML-схемы

5 Условные обозначения

В настоящей Рекомендации применяются некоторые условные обозначения, призванные информировать читателя о назначении текста. Хотя большая часть Рекомендации является нормативной, тем пунктам, в которых кратко излагаются обязательные требования, предъявляемые к системе управления (управляющей и/или управляемой), предшествуют выделенные жирным шрифтом значки "R", заключенные в круглые скобки, за которыми следуют краткое наименование предмета требования и номер. Например:

(R) ПРИМЕР-1. Пример обязательного требования.

Требованиям, которые могут быть факультативно реализованы системой управления, предшествует значок "O" вместо "R". Например:

(O) ПРИМЕР-2. Пример необязательного требования.

Для создания профилей соответствия и соблюдения используются операторы требований.

В настоящую Рекомендацию включены примеры на языках WSDL и XML, а нормативные WSDL- и XML-определения типов данных, базовых классов и других элементов сервисно-ориентированного моделирования включены в Приложение А. Примеры на языках WSDL и XML выделены шрифтом "Courier" размером 10 пунктов:

```
<!-- Пример XML-схемы -->
<xsd:complexType name="AType">
  <xsd:sequence>
    <xsd:element name="a" type="xsd:string"/>
    <xsd:element name="b" type="xsd:long"/>
  </xsd:sequence>
</xsd:complexType>
```

6 Обзор платформы управления на основе веб-услуг

В ИТ-индустрии широко используются веб-услуги. В Дополнении I содержится дополнительная информация о характерных чертах технологий на основе веб-услуг. Веб-услуги, подобные технологии CORBA, могут использоваться в интерфейсах управления сетями.

Настоящая Рекомендация вместе с [ITU-T Q.818] устанавливает платформу для определения того, как следует моделировать интерфейсы, поддерживаемые системами управления и элементами сети с использованием WSDL и XML-схемы.

Платформа управления на основе веб-услуг включает в себя следующие аспекты.

- 1) Указания по определению управляемого объекта и интерфейса:
 - принципы проектирования сервисно-ориентированного WSDL-интерфейса;
 - определение управляемых объектов с использованием XML-схемы;
 - методы доступа к МО;
 - наследование МО и операций интерфейса;
 - указания по моделированию информации для операций интерфейса на основе веб-услуг;
 - идиомы проектирования для спецификаций WSDL и XML-схемы.
- 2) Веб-услуги, поддерживающие услуги управления сетями:
 - определение услуг уведомления с использованием уведомлений на основе веб-услуг OASIS [OASIS WSN];
 - использование регистрации услуг OASIS UDDI [OASIS UDDI];
 - определение heartbeat-услуги;
 - определение услуги операций над множеством объектов (МОО);
 - определение услуги сдерживания.

В настоящей Рекомендации рассматриваются главным образом указания по определению управляемых объектов и интерфейсов, а в [ITU-T Q.818] – веб-услуги, поддерживающие услуги управления сетями. Вместе эти Рекомендации устанавливают платформу управления на основе веб-услуг.

7 Принципы проектирования сервисно-ориентированного WSDL-интерфейса

В этом разделе определяются некоторые вопросы по проектированию интерфейса, которые решаются с помощью данной платформы посредством сервисно-ориентированных интерфейсов. В нем рассматриваются принципы моделирования сервисно-ориентированных управляемых объектов и методы доступа к ним.

Вопросы сервисно-ориентированного проектирования, связанные с составом и моделированием WSDL и XSD, касаются суперклассов, именованых управляемых объектов и сервисно-ориентированных интерфейсов, операций и уведомлений.

Веб-услуги – это сервисно-ориентированные технологии, сравнимые с традиционными парадигмами управления CORBA/IDL и GDMO/CMIP. Анализ и проектирование объектно-ориентированных интерфейсов (ООАД) фокусируются на уровне классов, то есть инкапсулируют поведение и связанные с ним данные в один и тот же объект, который предоставляет один или несколько интерфейсов для доступа к своим инкапсулированным состояниям и атрибутам. При проектировании сервисно-ориентированных интерфейсов инкапсулированные данные и состояния отделяются от поведения, так что на более высоком уровне может обеспечиваться слабая взаимосвязь. При этом

подходе поведением некоторых (общих) объектов управляют уже не они сами, а несколько предопределенных операций интерфейса.

Эта Рекомендация наряду с [ITU-T Q.818] устанавливает принципы упрощенного универсального использования шаблонов проектирования сервисно-ориентированных интерфейсов. Функции управления и контроля определяются с помощью сервисно-ориентированного подхода, то есть операции интерфейса организованы в рамках определенной услуги (наборы функций управления, такие как управление конфигурацией, управление рабочими характеристиками и т. п.), а не отдельного класса объектов управления. Для реализации такого сервисно-ориентированного подхода требуется гибкость детализации доступа к конкретным приложениям, когда доступ к определенным наборам типов объектов TMN осуществляется посредством предопределенных WSDL-интерфейсов.

Данная платформа поддерживает следующие принципы определения сервисно-ориентированных моделей управления информацией и интерфейсов на основе WSDL и XSD.

- Все взаимодействия с интерфейсом определяются как WSDL-операции, каждая операция включает запрос и при необходимости может включать соответствующий ответ.
- При обмене через интерфейс управления каждый МОС определяется как тип complexType XML, а каждый атрибут или состояние МОС определяется как элемент типа complexType.
- Присвоение имен экземплярам МОС соответствует концепции DN, но это строка содержит весь перечень RDN.
- В данной платформе определена универсальная услуга, которая охватывает пять универсальных методов доступа к объектам: createMO, deleteMO, getMOAttribute, setMOAttribute и getPackages. Кроме того, универсальные методы доступа к объектам используют пары "имя-значение" для выражения свойств и их значений в экземплярах МОС разных типов.
- Другие функции управления интерфейсом определяются как WSDL-операции интерфейса, организованные как услуги с детализацией наборов функций управления.
- Общие типы данных определяются как XML-схемы, которые могут совместно использоваться определениями интерфейса, связанными с конкретным приложением.
- Уведомления, отправленные агентом менеджеру, должны соответствовать формату и поведению, указанным в [OASIS WSN]. Принципы руководства услугами управления уведомлениями дополнительно определены в [ITU-T Q.818].

8 Определение универсального управляемого объекта с использованием XML-схемы

8.1 Роль веб-услуг в интерфейсах управления

Для поддержки программных объектов, представляющих управляемые ресурсы, определен базовый класс, используемый при моделировании сетевых ресурсов. Другие МОС (классы управляемых объектов) в информационных моделях, предназначенных для работы в рамках данной платформы, должны разрабатываться на основе этих базовых классов. Для обеспечения интерфейсов управления МО определены некоторые универсальные методы доступа и некоторые другие расширенные услуги.

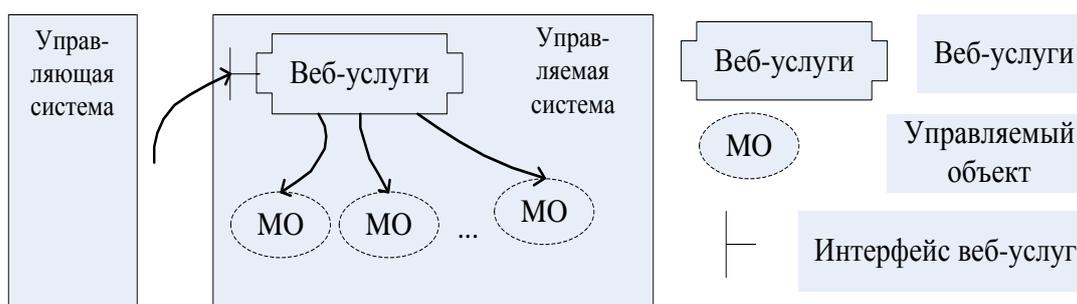


Рисунок 1 – Роль веб-услуг

На рисунке 1 показано, как управляющая система обращается к управляемой системе, поддерживающей интерфейс веб-услуг. Интерфейс веб-услуг выступает в качестве промежуточного объекта, который позволяет управляющей системе управлять соответствующими МО в составе управляемой системы, предоставляющей управляемые ресурсы.

8.2 Определение управляемых объектов с использованием XML-схемы

МО – это представление в системе управления OSI управляемого ресурса, такого как соединение или компонент физического оборудования. Таким образом, МО является абстрактным представлением такого ресурса, который предоставляет свои свойства для целей управления. МО может включать в себя атрибуты, содержащие информацию, используемую для характеристики самого объекта и операций, которые он выполняет. Цель данной платформы – предоставить набор возможностей для управления этими МО. Для описания свойств и поведения МО необходимы определенные подходы. Сервисно-ориентированный МО – это управляемый объект, который представляет собой управляемый ресурс обслуживания с точки зрения общего состояния и поведения и в котором состояние и поведение разделяются посредством передачи управления поведением некоторому назначенному "управляющему объекту" (например, услуге с ее интерфейсом), взявшему на себя роль управляющего в отношении поведения выделенных ему управляемых объектов. Поскольку состояние и поведение МО можно разделить, состояние можно описать XML-схемой, а поведение – посредством интерфейсов веб-услуг на языке WSDL. Одним из важных преимуществ использования XML-документа для хранения состояния МО является то, что веб-услуги используют XML-схему для описания типа данных сообщений, которыми они обмениваются, и эти данные в отношении МО на основе XML можно передавать без каких-либо изменений.

8.2.1 Определение класса универсальных управляемых объектов

Следующим абстрактным представлением управляемых объектов является класс управляемых объектов. Все сетевые ресурсы имеют некоторые общие атрибуты, и все МОС наследуют их, прямо или косвенно, от суперкласса, а именно ManagedObject. Использование ManagedObject для определения новых МОС осуществляется проще и быстрее и обеспечивает более качественное обслуживание. Как упоминалось выше, все МОС описаны в XML-схеме, тип данных ManagedObject приведен в таблице 1, а атрибуты – в таблице 2.

Таблица 1 – Тип данных суперкласса ManagedObject

```
<xsd:complexType name="ManagedObject_C">
  <xsd:sequence>
    <xsd:element name="objectClass" type="xsd:string"/>
    <xsd:element name="objectInstance" type="x782:NameType"/>
    <xsd:element name="packages" type="x782:PackageListType"/>
    <xsd:element name="creationSource" type="x782:SourceIndicatorType"/>
  </xsd:sequence>
</xsd:complexType>
```

Таблица 2 – Атрибуты суперкласса ManagedObject_C

Имя атрибута	Квалификатор поддержки	Квалификатор чтения	Квалификатор записи
objectClass	Обязательный	Обязательный	–
objectInstance	Обязательный	Обязательный	–
packages	Необязательный	Обязательный	–
creationSource	Необязательный	Обязательный	–

Как показано в таблице 2, ManagedObject состоит из четырех атрибутов: objectClass, objectInstance, packages и creationSource. С атрибутом связано значение, которое может иметь простую или

сложную структуру. Здесь атрибут МО отличается от атрибута в спецификации XML-схемы и может быть просто отображен на элемент XML-схемы. Атрибут `objectClass` используется для идентификации типа класса данного экземпляра МО. Атрибут `objectInstance` используется для однозначной идентификации экземпляра МО, а тип данных, используемый для `NameType` (он имеет ту же семантику, что и выделенное имя DN), указан в (1). Пакеты атрибутов представляют собой набор строк, указывающих возможности, поддерживаемые МО. Атрибут `creationSource` указывает, создается ли МО автоматически в управляемой системе, или управляющей системой посредством операции управления, или же это неизвестно.

```
DN(list of xsd:string) ::= "<attribute_name_1>=<attribute_value_1>",
    "<attribute_name_2>=<attribute_value_2>"
    ...
    "<attribute_name_n>=<attribute_value_n>"
```

(1)

В приведенной выше формуле атрибуты должны быть атрибутами именованного МОС.

Полное определение XML-схемы универсального `ManagedObject_C` дано в разделе А.1.

(R) ОБЪЕКТ-1. Все классы, используемые для моделирования ресурсов в управляемой системе, наследуют (прямо или косвенно) атрибуты описанного выше *ManagedObject_C* и определены в XML-схеме в разделе А.1. При этом должны поддерживаться описанные выше возможности.

8.2.2 Наследование управляемых объектов

Один МОС определяется как специализация другого МОС с использованием наследования, например, для всех остальных МО, которые прямо или косвенно наследуют атрибуты `ManagedObject`. Специализация МОС подразумевает, что все атрибуты, определенные в суперклассе, будут поддерживаться также и подклассом. Поскольку атрибуты МО описаны в XML-схеме, наследование может осуществляться путем расширения типов данных. Например, предположим, что МОС оборудования наследует непосредственно от базового класса `ManagedObject` и что оборудование может наследовать все атрибуты `ManagedObject` путем расширения и самостоятельно объявлять другие атрибуты, например `userLabel`, как показано в таблице 3.

Таблица 3 – Тип данных `Equipment_C` по расширению

```
<xsd:complexType name="Equipment_C">
  <xsd:complexContent>
    <xsd:extension base="x782:ManagedObject_C">
      <xsd:sequence>
        <xsd:element name="userLabel" type="xsd:string"/>
        ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Некоторым МО может потребоваться множественное наследование, но XML-схема поддерживает только одиночное наследование для типов данных и использование множественного наследования не рекомендуется. Если требуется семантика множественного наследования, то производный МОС наследует только от одного из вышестоящих МОС, а атрибуты другого вышестоящего класса (других вышестоящих классов) добавляются вручную.

8.2.3 Пакетирование

Для группирования определенных возможностей (например, связанных атрибутов) или предоставления возможностей обусловленной поддержки могут использоваться пакеты. Пакет можно определить как XSD `complexType` с суффиксом имени "_P". Когда он используется, элемент может иметь тип `complexType`, представляющий этот пакет, с квалификаторами `minOccurs = "0"` `maxOccurs`

= "1", что указывает на то, что этот пакет может быть условным или необязательным. Пример определения и использования пакета приведен в таблице 4.

Таблица 4 – Тип данных оборудования по расширению

```

<!-- определение пакета -->
<xsd:complexType name="StatePackage_P">
  <xsd:sequence>
    <xsd:element name="administrativeState"
type="x782:AdministrativeStateType"/>
    <xsd:element name="operationalState" type="x782:OperationalStateype"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Equipment_C">
  <xsd:complexContent>
    <xsd:extension base="x782:ManagedObject_C">
      <xsd:sequence>
        <xsd:element name="equipmentId" type="xsd:string"/>
        <xsd:element name="userLabel" type="xsd:string"/>
        ...
      <!-- использование пакета -->
      <xsd:element name="statePackage" type="x782:StatePackage_P" minOccurs="0"
maxOccurs="1" /></xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

8.2.4 Общие атрибуты и типы данных

В следующей таблице приведены некоторые общие атрибуты, а также некоторые общие типы данных, которые могут совместно использоваться в рамках данной платформы.

Таблица 5 – Стандартные атрибуты и типы данных

Имя атрибута	Тип данных	Описание
administrativeState	AdministrativeStateType	Подробнее см. в [ITU-T M.3701]
availabilityStatus	AvailabilityStatusSetType	Подробнее см. в [ITU-T M.3701]
backedUpStatus	BackedUpStatusType	Подробнее см. в [ITU-T M.3701]
controlStatus	ControlStatusSetType	Подробнее см. в [ITU-T M.3701]
creationSource (Note)	SourceIndicatorType	Подробнее см. в [ITU-T M.3701]
externalTime	ExternalTimeType	
objectClass (Note)	ObjectClassType	Указывает МОС
objectInstance (Note)	NameType	Указывает экземпляр МО
operationalState	OperationalStateType	Подробнее см. в [ITU-T M.3701]
packages (Note)	StringSetType	Указывает пакеты, поддерживаемые экземпляром МО
proceduralStatus	ProceduralStatusSetType	Подробнее см. в [ITU-T M.3701]

Таблица 5 (окончание)

Имя атрибута	Тип данных	Описание
standbyStatus	StandbyStatusType	Подробнее см. в [ITU-T M.3701]
systemLabel	SystemLabelType	Указывает метку системы
unknownStatus	UnknownStatusType	Подробнее см. в [ITU-T M.3701]
usageState	UsageStateType	Подробнее см. в [ITU-T M.3701]
ПРИМЕЧАНИЕ. – Эти атрибуты наследуются всеми управляемыми объектами.		

Подробные XSD-определения вышеуказанных типов данных приведены в разделе А.1.

9 Методы доступа к управляемым объектам

В этом разделе описывается платформа сетевого управления на основе веб-услуг, которая должна предоставлять набор методов управления сетевыми ресурсами. Эти методы предоставляют базовые возможности для управления МО, в связи с чем их называют универсальными методами доступа; см. таблицу 6. На рисунке 1 показана процедура доступа, и в рамках платформы используются технологии на основе веб-услуг для обмена информацией МО. Веб-услуги разделяют состояние и поведение МО и демонстрируют свое поведение через интерфейс веб-услуг. Поскольку веб-услуга относится к сервисно-ориентированным технологиям, в данной платформе все МО сконструированы так, чтобы доступ к ним осуществлялся через один интерфейс, и этот интерфейс должен "знать", какой МО является фактической целью операции, а в каждом запросе доступа должен содержаться уникальный идентификатор целевого МО. В таблице 1 указаны некоторые необходимые универсальные методы доступа, соответствующие вышеуказанным требованиям.

Таблица 6 – Универсальные методы доступа

Наименование операции	Входной параметр	Выходной параметр
getMOAttributes	– objectInstance : Name – attributeNameList : ПОСЛЕДОВАТЕЛЬНОСТЬ строк	– attributeNameAndValueList : ПОСЛЕДОВАТЕЛЬНОСТЬ {attributeName, attributeType, attributeValue} – status : ПЕРЕЧИСЛЯЕМЫЙ ТИП
setMOAttributes	– objectInstance : DN – attributeNVMLList : ПОСЛЕДОВАТЕЛЬНОСТЬ {attributeName, attributeType, attributeValue, modifyOption}	– status
createMO	– objectclass – objectClassInstance – attributeNameAndValueList	– status
deleteMO	– objectInstance	– status
getPackages	– objectInstance	– packages : список строк – status

где:

- 1) getMOAttributes – получить все или любое подмножество значений атрибута МО за одну операцию. В качестве первого параметра используется DN для однозначной идентификации МО и список имен запрашиваемых атрибутов. Возвращаемый результат состоит из значений атрибута и статуса операции. Параметр attributeNameAndValueList – это список троек значений attributeName, attributeType и attributeValue. Параметр attributeType указывает исходный тип attributeValue, а значения атрибутов возвращаются через элемент "any" XML-схемы для значений произвольного типа. Параметр status указывает, выполнена ли операция успешно или не выполнена. Поскольку тип данных значений возвращаемого атрибута

определен как "any", при получении такого запроса от клиента сервер возвращает запрошенные атрибуты в выходном параметре `attributeNameAndValueList`, где поле `attributeValue` закодировано из переменного элемента во фрагмент XML-текста, который может быть декодирован клиентским приложением с помощью параметра `attributeType`.

- 2) `setMOAttributes` – изменить значения атрибутов существующего МО. Помимо использования `objectInstance` для указания целевого МО, значения которого должны быть изменены, для задания атрибутов МО операция также использует список четверок, включающих `attributeName`, `attributeType`, `attributeValue` и `modifyOption`. Первые три атрибута – те же, что описаны выше; `modifyOption` указывает, как установить соответствующие значения атрибутов МО. Это перечисляемый тип, состоящий из значений "REPLACE", "ADDValues", "REMOVEValues" и "SETToDefault". "REPLACE" означает, что указанные значения атрибутов должны использоваться для замены текущих значений. "ADDValues" означает, что указанные значения атрибутов должны добавляться к текущим значениям. "REMOVEValues" означает, что указанные значения атрибутов должны быть удалены из текущих значений. "SETToDefault" означает, что значение атрибута должно быть установлено по умолчанию. Параметр `modifyOption` является необязательным, и если он не указан, то предполагается "REPLACE".
- 3) `createMO` – создать МО в управляемой системе. Должны быть указаны класс и имя создаваемого МО. Параметр `attributeNameAndValueList` используется для задания значений атрибутов, но его можно опустить, и если он не указан, то атрибутам присваиваются значения по умолчанию.
- 4) `deleteMO` – освободить все ресурсы, связанные с МО, и удалить его. Для определения целевого МО используется DN, а затем возвращается статус операции. Если целевой МО или какие-либо из содержащихся в нем МО не могут быть удалены, то операция возвращает статус `OperationFailed`.
- 5) `getPackages` – определить возможности целевого МО (группа атрибутов и/или операций). Экземпляр МО может поддерживать или не поддерживать все группы возможностей, определенные в МОС, и эта операция используется клиентом для получения фактических поддерживаемых возможностей экземпляра МО.

Для обеспечения компактного и эффективного представления потенциально сложных данных, которые должны передаваться как единое целое, широко используются списки и табличные структуры. В то же время могут возникать такие проблемы, как трудности с проверкой и интерпретацией данных. Поскольку веб-услуги являются веб-ориентированными и наследование услуг нарушает их слабые взаимосвязи, его использование не рекомендуется. Для обеспечения согласованности данных МО можно рассмотреть модель одноэлементной конструкции.

Все упомянутые выше методы работают только с одним МО. Если понадобится управлять миллионами объектов, необходимо, чтобы платформа поддерживала операции над множеством объектов с единственным методом инициирования или хотя бы с небольшим числом инициирований. Эту возможность обеспечивает услуга операций над множеством объектов (МОО), которая описана в [ITU-T Q.818].

Полная XML-схема и WSDL-определение интерфейса универсальных методов доступа МО приведены в разделе А.2.

(R) ОБЪЕКТ-2. Реализация методов доступа к МО должна поддерживать все операции, описанные выше, а также те, WSDL-описание которых приведено в разделе А.2.

10 Наследование управляемых объектов и операций интерфейса

10.1 Наследование атрибутов управляемых объектов

Используя наследование, один класс управляемых объектов можно определить, как специализацию другого класса управляемых объектов. Специализация класса управляемых объектов подразумевает, что все методы и атрибуты, определенные в суперклассе, будут поддерживаться также и подклассом. В случае сервисно-ориентированных интерфейсов на основе веб-услуг поддерживаются только атрибуты классов управляемых объектов. Введение наследования операций приведет к ухудшению таких характеристик, как хорошая функциональная совместимость и слабая взаимосвязь.

Хотя атрибуты управляемых объектов описываются XML-схемой, для наследования атрибутов можно использовать пункт 4.2 [W3C Primer] "Получение типов посредством расширения". Поскольку типы данных атрибутов базовых управляемых объектов определены в сложном типе XML-схемы, в подклассе можно расширить тип данных базовых управляемых объектов посредством значения атрибута *base* в элементе *extension* XML-схемы.

Когда сложный тип получается с помощью расширения, его эффективная модель содержимого представляет собой модель содержимого базового типа, к которой прибавляется модель содержимого, указанная при получении типа. Кроме того, две модели содержимого рассматриваются как две дочерние модели последовательной группы. В случае UKAddress модель содержимого UKAddress представляет собой модель содержимого Address, к которой прибавляются декларации элемента *postcode* и атрибута *exportCode*.

```
<complexType name="Address">
  <sequence>
    <element name="name" type="string"/>
    <element name="street" type="string"/>
    <element name="city" type="string"/>
  </sequence>
</complexType>
<complexType name="USAddress">
  <complexContent>
    <extension base="ipo:Address">
      <sequence>
        <element name="state" type="ipo:USState"/>
        <element name="zip" type="positiveInteger"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Соответственно для USAddress:

```
<sequence>
  <element name="name" type="string"/>
  <element name="street" type="string"/>
  <element name="city" type="string"/>
  <element name="state" type="ipo:USState"/>
  <element name="zip" type="positiveInteger"/>
</sequence>
</complexType>
```

Вышеупомянутый подход можно использовать для наследования элемента, который поддерживает семантику наследования атрибутов управляемого объекта.

10.2 Соображения, касающиеся наследования операций интерфейса

Хотя более поздняя версия WSDL (версия 2.0) поддерживает синтаксис наследования операций, она не нашла широкой поддержки в отрасли. Более ранние версии WSDL (до версии 1.1) не поддерживают этот синтаксис, но полностью поддерживаются отраслью. Соответственно в настоящей Рекомендации выражение наследования операций не допускается. Эта функция может поддерживаться семантически путем повторения определений операций интерфейса, унаследованных от интерфейсов более высокого уровня.

11 Указания по моделированию информации для интерфейсов на основе веб-услуг

11.1 Пространство имен (namespace)

Для целевого пространства имен в данной платформе используется следующий URI.

Таблица 5. Целевое пространство имен в данной платформе

Рекомендация	Целевое пространство имен
ITU-T X.782	http://www.itu.int/xml-namespace/itu-t/x.782
ITU-T Q.818	http://www.itu.int/xml-namespace/itu-t/q.818
Другие, такие как X.nnnn	http://www.itu.int/xml-namespace/itu-t/x.nnnn

При разработке других рекомендаций номер рекомендации X.nnnn в последней строке вышеуказанной таблицы должен быть заменен фактическим номером соответствующей рекомендации.

11.2 Сложный тип (complexType)

При использовании XML-схемы для определения содержимого МОС в целях моделирования МОС используется тип XSD complexType. Тип XSD complexType содержит последовательность, в которую могут входить один или несколько элементов XSD. Каждый тип complexType, соответствующий МОС, должен иметь суффикс имени "_C".

Другие общие типы данных атрибутов также можно определить как типы complexType, но в их имени типа должен использоваться суффикс "Type".

11.3 Атрибут (attribute)

Атрибуты и состояния МОС определяются как элементы типа complexType, соответствующие МОС. Следует отметить, что атрибут – это концептуальное свойство объекта, который моделируется как МОС; он не совпадает с атрибутом ключевого слова в XSD. В данной платформе атрибут ключевых слов XSD-элемента не используется.

11.4 Запрос (request)

Запрос используется для определения сообщения в целях взаимодействия между клиентом и серверным приложением веб-услуги. Сообщение запроса направляется клиентом веб-услуги серверу. Запрос содержит все входные параметры операции веб-услуги. Входные параметры могут состоять из нескольких частей или же в запросе может быть определена только одна часть, которая содержит все входные параметры в качестве своих внутренних элементов, входящих в состав соответствующего complexType.

11.5 Ответ (response)

Ответ используется для определения сообщения, возвращаемого сервером веб-услуги клиенту. Ответ содержит все выходные параметры, а также возвращаемое значение операции веб-услуги. Выходные параметры могут состоять из нескольких частей или же в ответе может быть определена только одна часть, которая содержит все выходные параметры в качестве своих внутренних элементов, входящих в состав соответствующего complexType.

11.6 Уведомление (notification)

Формат любых уведомлений, отправляемых через интерфейс управления, должен соответствовать формату из спецификации [OASIS WSN]. В [ITU-T Q.818] представлены общий заголовок для всех уведомлений и содержание уведомлений для некоторых часто используемых типов уведомлений, в том числе:

objectCreation, objectDeletion, attributeValueChange, stateChange, communicationAlarm, environmentalAlarm, equipmentAlarm, processingErrorAlarm, qualityOfServiceAlarm, Violation, integrityViolation, operationalViolation, physicalViolation, securityViolation, timeDomainViolation, relationshipChange, а также сообщения heartbeat.

Для известных типов уведомлений разработчики данной платформы должны использовать определения уведомлений, содержащиеся в [ITU-T Q.818]. Любые новые определения уведомлений должны быть увязаны с этим подходом, а также с заголовком уведомления.

11.7 Соглашения по именованию для МОС, пакетов, атрибутов и типов данных

Для моделирования на основе XML-схемы применяются следующие соглашения по именованию:

- Все атрибуты МОС определяются как XSD complexType, имя МОС должно иметь суффикс имени "_C", первая буква должна быть прописной, так чтобы это определение complexType отличалось от других, обычных определений типа данных. Например: ManagedObject_C, Equipment_C.
- Атрибут МОС определяется как элемент типа complexType, представляющий МОС, и первая буква имени атрибута должна быть строчной.
- Пакет можно определить как XSD complexType с суффиксом имени "_P", первая буква должна быть прописной.
- Обычное определение типа данных должно иметь суффикс имени "Type" и начинаться с прописной буквы, чтобы сделать его более удобочитаемым. Например: AdministrativeStateType.
- Для элементов следует использовать lowerCamelCase, например "personName".
- Для определения имен simpleType и complexType следует использовать UpperCamelCase.
- Тип установки заданного значения (неупорядоченный набор) должен иметь суффикс имени SetType, а списочный тип (упорядоченная последовательность) – суффикс имени ListType.

12 Идиомы проектирования для спецификаций XML-схем¹

12.1 Модель данных с использованием XML-схемы

12.1.1 Обзор использования XML-схемы

XML допускает произвольные определения данных с использованием специальных меток. XML-документ становится практически применимым, когда он ограничен четко определенной структурой. XML-схема служит средством определения правил, семантики и структуры XML-документов. Схема обеспечивает программную проверку структурированного XML-документа. XML-схема определяется с использованием формата XML в файле с расширением .xsd.

12.1.2 Версия спецификации схемы

Пространство имен XML-схемы версии 1.1 <http://www.w3.org/2001/XMLSchema> то же, что и для документа XML-схемы версии 1.0, а пространством имен экземпляра документа XML остается <http://www.w3.org/2001/XMLSchema-instance>. Это позволяет разработчикам схем создавать документы XML-схемы версии 1.0, не беспокоясь об обновлении пространств имен при добавлении функций версии 1.1. В XML-схеме версии 1.1 вводится новое пространство имен для управления версиями (<http://www.w3.org/2007/XMLSchemaversioning>).

В настоящей Рекомендации используется XML-схема версии 1.1, как указано в пункте 7.2 [ITU-T Q.818].

12.2 Вопросы проектирования XML-схемы

12.2.1 Разработка единственной схемы или набора взаимосвязанных схем

Язык описания схемы позволяет импортировать схему из других документов. При разработке набора взаимосвязанных схем соображения, касающиеся пространства имен, играют важную роль при определении того, как представлять другие схемы в итоговом экземпляре XML-документа.

- Ссылка с использованием тега `<xsd:import>`: элемент `import` позволяет ссылаться на компоненты схемы из документов схемы с разными целевыми пространствами имен. Это наиболее распространенный подход к разработке схемы, также именуемый неоднородным подходом к проектированию пространства имен.
- Ссылка с использованием тега `<xsd:include>`: элемент `include` добавляет во включающую схему компоненты из других документов схем с тем же целевым пространством имен (или без указанного пространства имен). Таким образом, элемент `include` позволяет добавить во включающую схему все компоненты включенной схемы. Это приводит к двум разным подходам к проектированию.
- Однородный подход к проектированию пространства имен: этот подход предусматривает, что все разрабатываемые связанные схемы имеют одно и то же целевое пространство имен.
- Подход к проектированию пространства имен на основе заимствования: этот подход предусматривает существование одной или нескольких вспомогательных схем, не имеющих определенного целевого пространства имен, причем основная схема включает в себя вспомогательную схему (вспомогательные схемы). Вспомогательные схемы принимают пространство имен основной схемы.

Однородный подход к проектированию и подход на основе заимствования позволяют пересмотреть определения типа, группы и группы атрибутов, данные во вспомогательной схеме. Переопределение типа влияет на элементы включающей схемы, а также на элементы включенной схемы. Таким образом, переопределенные типы могут взаимодействовать с производными типами и вызывать конфликты.

В настоящей Рекомендации и [ITU-T Q.818] используется только подход [`xsd:import`].

¹ Текст этого раздела перенесен из [ATIS-I-000002] с некоторыми изменениями, соответствующими данной платформе управления на основе веб-услуг.

12.2.2 Шаблоны проектирования схемы

В отношении проектирования XML-схемы обычно упоминаются следующие четыре шаблона структурного проектирования:

- "Матрешка": этот шаблон проектирования предусматривает, что в файле схемы содержится единый глобальный элемент. Все дочерние элементы определяются в рамках этой единой иерархии определения элементов. Данный шаблон проектирования не способствует повторному использованию элементов и требует определения элементов в одном файле схемы.
- "Колбасная нарезка": при этом подходе к проектированию определены несколько корневых элементов, повторное использование отдельных элементов.
- "Райский сад": подход к проектированию схемы, при котором все элементы и типы отображаются глобально; сложные типы определяются посредством ссылки на глобальные элементы повторного использования. Характерной чертой этого шаблона проектирования является существование большого количества возможных корневых элементов.
- "Жалюзи": подход к проектированию схемы, при котором сначала создаются типы, из которых затем строятся элементы. Чтобы обеспечить максимально возможное повторное использование элементов, создаются типы `simpleType` и `complexType`. Особенностью этого подхода является существование единственного корневого элемента.

В данной платформе используется подход "Райский сад".

12.2.3 Проектирование с учетом устойчивости

Одна из целей, преследуемых при разработке схемы, – сделать ее устойчивой к изменениям и обеспечить гибкость для удовлетворения будущих потребностей пользователей этой схемы. В этом разделе обсуждаются некоторые механизмы обеспечения гибкости и расширяемости модели XML-схемы.

12.2.3.1 Использование метасимволов

Схема W3C позволяет создавать такие конструкции, как `<xsd:any>` и `<xsd:anyAttribute>`, чтобы экземпляр документа мог содержать XML-данные, не ограниченные напрямую моделью содержимого определяющей XML-схемы. Это позволяет механизму расширяемости обеспечить определенный контроль, уровень которого можно задать с использованием атрибутов `namespace` и `processContents` (см. [W3C XS-P1] и [W3C XS-P2]). Например, атрибут `namespace` можно использовать для ограничения XML-данных набором предопределенных пространств имен, тем самым обеспечив возможность определенной проверки этих расширенных XML-данных в экземпляре документа. Атрибут `processContents` определяет, как проверяются эти расширенные XML-данные с помощью XML-валидатора.

Метасимвол `<xsd:any>` позволяет поставщикам разрабатывать конкретные функции, отсутствующие в определяющей схеме. Продуманное использование метасимволов `<xsd:any>` и `<xsd:anyAttribute>` помогает строить модели содержимого схемы, более устойчивые к изменениям и в меньшей степени подверженные отторжению схемы. Однако следует помнить, что, используя `<xsd:any>`, можно непреднамеренно разрешить создание недетерминированных моделей содержимого, способных вызывать проблемы при работе с некоторыми синтаксическими анализаторами XML. Разработчик схемы должен учитывать это при принятии решения об использовании `<xsd:any>`.

Данная платформа допускает использование `<xsd:any>`, но в этом случае также следует дать соответствующее пояснение.

12.2.3.2 Использование `substitutionGroup`

Модели содержимого элементов группы подстановки связаны друг с другом получением типа. Сменный элемент называется головным элементом и определяется в глобальной области схемы. По сути, конструкция `substitutionGroup` помогает построить набор элементов, которые можно указывать с помощью общего элемента.

Конструкция `substitutionGroup` может быть полезной в следующих случаях:

- Разработка иерархий связанных классов: создание иерархии классов позволяет использовать преимущества такого наследования в объектно-ориентированных языках программирования.
- Настройка внешней схемы под конкретные нужды: если элемент импортированной схемы определен глобально, то для этого элемента можно создать substitutionGroup, построив производный тип и разрешив подстановку этого производного типа в сложной структуре данных.

Иерархии классов и substitutionGroup приводят к тесной связи между структурами данных и могут сделать конструкцию хрупкой и неизменяемой. Вместо этого для создания составной модели содержимого можно использовать такие конструкции, как <xsd:choice>. Составная конструкция может упростить проект и ослабить взаимосвязи.

В настоящей Рекомендации используется расширение complexType для представления иерархии классов, а использование substitutionGroup в данной платформе не допускается.

12.3 Рекомендации для разработчиков схем

12.3.1 Рекомендации по разработке XML-схемы

Разработчику XML-интерфейса следует:

- по возможности создавать схемы общего пользования.

Разработчик должен:

- для схем, создаваемых МСЭ-Т, использовать следующий формат пространства имен:
http://www.itu.int/xml-namespace/itu-t/<номер документа МСЭ-Т> или
http://www.itu.int/xml-namespace/itu-t/<номер документа МСЭ-Т>/<идентификатор модели данных>;
- для имен пространства имен использовать только строчные буквы.
- Возможно, что документ будет изменен с небольшим изменением схемы или без него (например, обновление ссылок).

По этой причине в <номер документа МСЭ-Т> не включается номер версии документа. Включение номера версии документа приведет к изменению пространства имен при каждом обновлении документа.

Разработчику следует:

- Объявить все простые и сложные типы глобально.
- Объявить элементы и атрибуты локально; один из основных элементов инкапсулирует все остальные.
- Обеспечить присутствие атрибута версии схемы (младший номер версии схемы) и увеличивать его с каждым изменением в схеме; начальное значение равно "0".
- Обеспечить присутствие во всех схемах не менее двух пространств имен: пространства имен XML-схемы W3C и пространства имен, относящегося к сопутствующему стандарту.
- Указать пример атрибутов схемы для версии 1.0 схемы услуги доступа к МО (moas).

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  targetNamespace="http://www.itu.int/xml-namespace/itu-t/x.782"
```

```
  xmlns:moas="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"
```

```
  version="1.0">
```

```
.....
```

```
</schema>
```

12.3.2 Рекомендации по кодированию XML-схемы

Разработчику надлежит руководствоваться следующими рекомендациями по названиям меток типов, элементов и атрибутов:

- По возможности используйте общепринятые отраслевые или коммерческие названия, чтобы обеспечить согласованность между отраслевой или деловой документацией и схемами.
- Делайте названия описательными; избегайте использования излишних сокращений.
- Не используйте одно и то же название дважды в одной схеме.
- Избегайте сокращений, но, если они используются, сохраняйте написание с прописной буквы.
- Избегайте точек и прочерков.
- Для определения названий элементов и типов используйте только буквенно-цифровые символы.
- Используйте названия в единственном числе, если само понятие не предполагает множественного.

Разработчику следует рассматривать элемент как тип модели по умолчанию и придерживаться следующих рекомендаций:

- Если компонент можно считать независимым объектом, сделайте его элементом.
- Если компонент должен использоваться многократно, сделайте его глобальным типом.

Разработчику надлежит руководствоваться следующими общими рекомендациями:

- Объявляйте элементы как необязательные, если нет абсолютной необходимости в обратном.
- Вместо `nillable="True"` используйте `minOccurs="0"`.
- Используйте `maxOccurs`. Если размерность больше одного, определите ее; в противном случае используйте `maxOccurs="unbounded"`.
- Создавайте как можно больше `simpleTypes`.
- Если элемент должен использоваться многократно, создавайте глобальный тип (глобальные типы определяются сразу под элементом схемы).
- Все типы должны определяться глобально.
- Используйте `elementFormDefault = "qualified"`.
- Используйте `attributeFormDefault = "unqualified"`.
- Используйте кодировку символов UTF-8: `<?xml version="1.0" encoding="UTF-8"?>`.
- Везде, где возможно повторное использование и требуется ясность, следует использовать элементы `<documentation>` (для указания языка следует использовать атрибут `xsd:lang`).
- Для описания определений всех типов используйте аннотации; название типа включайте в минимальной степени.
- Для компонентов даты и времени используйте только элемент `xsd:dateTime`.
- Там где требуется компонент `compositor`, используйте только `<sequence>` или `<choice>`.

При добавлении ограничений разработчику надлежит руководствоваться следующими рекомендациями:

- При разработке новой схемы ограничивайте новые простые типы с применением по мере возможности соответствующих запретов.
- Определите границы диапазонов значений.
- Выберите подходящий простой тип. Например, когда требуется число, использование простого типа `nonNegativeInteger` или `positiveInteger`, если это возможно, предпочтительнее выбора типа `integer`.
- При определении типа `string` по возможности используйте `maxLength` и шаблоны для дополнительных ограничений. Например, шаблон определения 15-значного международного номера телефона может быть следующим ([ITU-T E.164]):

```
<xsd:restriction base="xsd:string">
  <xsd:pattern value="([1-9][0-9]{0,2})(-[1-9][0-9]{0,4})?(-[1-9][0-9]{0,13})(-[0-9]+)?"/>
</xsd:restriction>
```

- Когда простой тип принимает только предопределенный набор значений, используйте границы перечисления, чтобы ограничить допустимые значения простого типа.
- Используйте типы union, если типы данных принимают два или более разных набора значений, которые можно ограничить независимо друг от друга. Например, код штата или почтовый индекс.

12.3.3 Конструкции XML-схемы, которых следует избегать

При проектировании XML-схем разработчику предлагается руководствоваться следующими рекомендациями:

- Не используйте <xsd:all> (вместо этого применяйте <sequence> или <choice>). Конструкции <xsd:sequence> и <xsd:choice> принудительно устанавливают фиксированный порядок дочерних элементов в экземпляре документа.
- Не используйте инструкции по обработке. Инструкции по обработке не являются частью документа, а передаются приложению для выполнения конкретных функций приложения. Инструкции по обработке могут не соответствовать внутренней структуре и поэтому имеют ограниченное применение в определениях схем.
- Не используйте комментарии style DTD или XML. Конструкцию для комментариев и информации обеспечивают элементы annotation и documentation. Комментарии style XML неудобны для обработки экземпляра документа.
- Не используйте группы или переопределение групп XML. Переопределение группы может привести к конфликту при обработке переопределенного типа и производного типа экземпляра данных.
- Не используйте группы замещения. Конструкция группы замещения создает жесткую связь и повышает сложность, что может сделать конструкцию хрупкой и неизменяемой.
- Не используйте значения по умолчанию/фиксированные значения. Такое использование атрибутов, как правило, вводит в заблуждение читателей документа схемы, поскольку оно не дает никакого эффекта.

12.4 Рекомендации по расширению схемы

Понятно, что разработчики схем не могут заранее предвидеть все будущие потребности схемы. Разработчику следует использовать такое определение схемы, которое обеспечит гибкость и позволит пользователям передавать частные или специальные данные. В общем случае разработчикам схем следует учесть необходимость поддержки частных данных, вводя конструкции, поддерживающие включение некоторых данных универсальным способом (например, пар "имя-значение").

Однако для более сложных потребностей, например, когда требуется проверка, установление правил или политики, такого включения может оказаться недостаточно.

Разработчик схем должен использовать один из двух возможных способов расширения поддержки частных данных в схеме:

- 1) прогнозирование необходимости расширения определенных частей схемы и разрешение пользователям добавлять частные данные OpenContent; или
- 2) наследование типа с использованием расширения <xsd:extension>.

Использование такого расширения не рекомендуется, так как это может привести к проблемам с интерпретируемостью. Например, если пользователь расширяет стандартную схему с использованием xsd:extension, чтобы добавить свои специальные элементы и атрибуты, то результирующие XML-данные могут некорректно анализироваться другими пользователями, которые строят свои приложения на основе исходной стандартной схемы.

Таких случаев можно избежать, если обе стороны используют одну и ту же XML-схему. В этой платформе использование расширений допускается, но расширенные типы должны быть также включены в качестве общедоступного стандарта.

13 Согласованность и соответствие

В этом разделе определяется критерий, которому должны удовлетворять другие документы по стандартам, претендующие на соответствие настоящим руководящим указаниям, и функции, которые должны быть реализованы в системах, претендующих на соответствие настоящей Рекомендации.

13.1 Согласованность документов по стандартам

Любая спецификация, претендующая на соответствие настоящим руководящим указаниям, должна:

- 1) Определять все классы, которые моделируют ресурсы как производные (прямые или косвенные) от объекта *ManagedObject_C*, описанного в пункте 8.2.1 и определенного в XML-схеме в разделе A.1.
- 2) Поддерживать наследование атрибутов с использованием механизма, указанного в разделе 10.1.
- 3) По возможности использовать определения общих типов атрибутов, указанных в пункте 8.2.4.
- 4) По возможности использовать общие типы данных, определенные в XML-схеме в разделе A.1.
- 5) Соблюдать руководящие принципы моделирования интерфейсов на основе веб-услуг, приведенные в разделе 11.
- 6) Придерживаться соглашений о проектировании XML-схемы, приведенных в разделе 12.

13.2 Соответствие системы

Реализация, претендующая на соответствие настоящей Рекомендации, должна:

- 1) Поддерживать все возможности методов доступа к МО, описанные в разделе 9, и соответствующий интерфейс WSDL, как указано в разделе A.2.

13.3 Рекомендации в отношении заявления о соответствии

В заявлении о соответствии должен быть указан документ и год публикации, чтобы гарантировать правильность определения версии XML-схемы и WSDL.

Приложение А

Общие определения WSDL и XML-схемы

(Это приложение является неотъемлемой частью настоящей Рекомендации.)

В этом приложении даны общие определения WSDL-интерфейсов, а также определены некоторые общие типы данных на основе XML-схемы.

А.1 Определения XML-схемы для общих типов данных и универсального управляемого объекта

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Определение XML-схемы для общих типов данных, используемых в данной платформе.
      Имя файла: x782.xsd -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
  targetNamespace="http://www.itu.int/xml-namespace/itu-t/x.782"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">

  <xsd:simpleType name="RDNTType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>

  <xsd:complexType name="NameType">
    <xsd:sequence>
      <xsd:element name="rdn" type="x782:RDNTType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="NameSetType">
    <xsd:sequence>
      <xsd:element name="dn" type="x782:NameType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="UIDType">
    <xsd:sequence>
      <!-- uri указывает пространство имен, в котором определена константа. -->
      <xsd:element name="uri" type="xsd:string"/>
      <!-- value указывает постоянное значение для этого компонента
в вышеуказанном пространстве имен. -->
      <xsd:element name="value" type="xsd:unsignedLong"/>
    </xsd:sequence>
  </xsd:complexType>
```

```

<xsd:complexType name="UIDSetType">
  <xsd:sequence>
    <xsd:element name="uid" type="x782:UIDType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MOClassListType">
  <xsd:sequence>
    <xsd:element name="moClass" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="AdministrativeStateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="locked"/>
    <xsd:enumeration value="unlocked"/>
    <xsd:enumeration value="suttingDown"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="OperationalStateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="disabled"/>
    <xsd:enumeration value="enabled"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="AvailabilityStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="inTest"/>
    <xsd:enumeration value="failed"/>
    <xsd:enumeration value="powerOff"/>
    <xsd:enumeration value="offLine"/>
    <xsd:enumeration value="offDuty"/>
    <xsd:enumeration value="dependency"/>
    <xsd:enumeration value="degraded"/>
    <xsd:enumeration value="notInstalled"/>
    <xsd:enumeration value="logFull"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:complexType name="AvailabilityStatusSetType">
  <xsd:sequence>
    <xsd:element name="availableState" type="x782:AvailabilityStatusType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="BackedUpStatusType">
  <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>
<xsd:simpleType name="ControlStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="subjectToTest"/>
    <xsd:enumeration value="partOfServicesLocked"/>
    <xsd:enumeration value="reservedForTest"/>
    <xsd:enumeration value="suspended"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ControlStatusSetType">
  <xsd:sequence>
    <xsd:element name="controlState" type="x782:ControlStatusType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="ExternalTimeType">
  <xsd:restriction base="xsd:dateTime"/>
</xsd:simpleType>

<xsd:simpleType name="ObjectClassType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:simpleType name="ProceduralStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="initializationRequired"/>
    <xsd:enumeration value="notInitialized"/>
    <xsd:enumeration value="initializing"/>
    <xsd:enumeration value="reporting"/>
    <xsd:enumeration value="terminating"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:complexType name="ProceduralStatusSetType">
  <xsd:sequence>
    <xsd:element name="proceduralState" type="x782:ProceduralStatusType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SourceIndicatorType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="resourceOperation"/>
    <xsd:enumeration value="managementOperation"/>
    <xsd:enumeration value="unknown"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="StandbyStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="hotStandby"/>
    <xsd:enumeration value="coldStandby"/>
    <xsd:enumeration value="providingService"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="StringSetType">
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SystemLabelType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:simpleType name="UsageStateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="idle"/>
    <xsd:enumeration value="active"/>
    <xsd:enumeration value="busy"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="UnknownStatusType">
  <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>

```

```

<xsd:complexType name="AttributeValueType">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AttributeNameAndValueType">
  <xsd:sequence>
    <xsd:element name="attributeName" type="xsd:string"/>
    <xsd:element name="attributeType" type="xsd:string"/>
    <xsd:element name="attributeValue" type="x782:AttributeValueType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AttributeNameAndValueSetType">
  <xsd:sequence>
    <xsd:element name="attributeNameAndValue"
type="x782:AttributeNameAndValueType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="AdditionalTextType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:complexType name="AnyValueType">
  <xsd:sequence>
    <xsd:element name="typeURI" type="xsd:string"/>
    <xsd:element name="value" type="x782:AttributeValueType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AdditionalInformationSetType">
  <xsd:sequence>
    <xsd:element name="additionalInfo" type="x782:AnyValueType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="NotificationIDType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:complexType name="NotificationIDSetType">
  <xsd:sequence>

```

```

        <xsd:element name="source" type="x782:NameType" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CorrelatedNotificationType">
    <xsd:sequence>
        <xsd:element name="source" type="x782:NameType"/>
        <xsd:element name="notifIDs" type="x782:NotificationIDSetType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CorrelatedNotificationSetType">
    <xsd:sequence>
        <xsd:element name="notifications" type="x782:CorrelatedNotificationType"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType><xsd:complexType name="AttributeChangeType">
    <xsd:sequence>
        <xsd:element name="attribugteName" type="xsd:string"/>
        <xsd:element name="attributeTypeURI" type="xsd:string"/>
        <xsd:element name="oldValue" type="x782:AttributeValueType"/>
        <xsd:element name="newValue" type="x782:AttributeValueType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AttributeChangeSetType">
    <xsd:sequence>
        <xsd:element name="attributeChange" type="x782:AttributeChangeType"
minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ProbableCauseType">
    <xsd:complexContent>
        <xsd:extension base="x782:UIDType"/>
    </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name="PerceivedSeverityType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="indeterminate"/>
        <xsd:enumeration value="critical"/>
        <xsd:enumeration value="major"/>
        <xsd:enumeration value="minor"/>
        <xsd:enumeration value="warning"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

        <xsd:enumeration value="cleared"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TrendIndicationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="lessSevere"/>
        <xsd:enumeration value="noChange"/>
        <xsd:enumeration value="moreSevere"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ThresholdIndicationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="up"/>
        <xsd:enumeration value="down"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="ThresholdLevelIndType">
    <xsd:sequence>
        <xsd:element name="indication" type="x782:ThresholdIndicationType"/>
        <!-- исследуемое значение -->
        <xsd:element name="low" type="xsd:float" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="high" type="xsd:float"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ThresholdInfoType">
    <xsd:sequence>
        <xsd:element name="attributeID" type="xsd:string"/>
        <xsd:element name="observedValue" type="xsd:float"/>
        <xsd:element name="thresholdLevel" type="x782:ThresholdLevelIndType"/>
        <xsd:element name="armTime" type="xsd:dateTime"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ProposedRepairActionSetType">
    <xsd:complexContent>
        <xsd:extension base="x782:UIDType"/>
    </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:complexType name="SuspectObjectType">
  <xsd:sequence>
    <xsd:element name="moClass" type="xsd:string"/>
    <xsd:element name="suspectedMOInstance" type="x782:NameType"/>
    <xsd:element name="failureProbability" type="xsd:unsignedShort"
minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SuspectObjectSetType">
  <xsd:sequence>
    <xsd:element name="suspectedMO" type="x782:SuspectObjectSetType"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SecurityAlarmCauseType">
  <xsd:complexContent>
    <xsd:extension base="x782:UIDType"/>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="SecurityAlarmDetectorType">
  <xsd:sequence>
    <xsd:element name="mechanism" type="x782:UIDType"/>
    <xsd:element name="obj" type="x782:NameType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ServiceUserType">
  <xsd:sequence>
    <xsd:element name="typeURI" type="xsd:string"/>
    <xsd:element name="value" type="x782:AttributeValueType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ServiceProviderType">
  <xsd:sequence>
    <xsd:element name="typeURI" type="xsd:string"/>
    <xsd:element name="value" type="x782:AttributeValueType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SpecificProblemSetType">
  <xsd:complexContent>
    <xsd:extension base="x782:UIDSetType"/>
  </xsd:complexContent>
</xsd:complexType>

```

```

<!-- Определение XML-схемы универсального управляемого объекта -->
<xsd:complexType name="ManagedObject_C">
  <xsd:sequence>
    <xsd:element name="objectClass" type="xsd:string"/>
    <xsd:element name="objectInstance" type="x782:NameType"/>
    <xsd:element name="packages" type="x782:StringSetType"/>
    <xsd:element name="creationSource" type="x782:SourceIndicatorType"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

A.2 Определение WSDL и XML-схемы для общих методов доступа к объектам

1) Определение XML-схемы услуги доступа к МО МСЭ

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Определение XML-схемы для типов данных, используемых в услуге доступа к МО,
описанной в настоящей Рекомендации.
Имя файла : x782_MOAccessService.xsd -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
  xmlns:moas="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"
  targetNamespace="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">
<xsd:import namespace="http://www.itu.int/xml-namespace/itu-t/x.782"
  schemaLocation="x782.xsd"/>
  <xsd:complexType name="AttributeNameListType">
    <xsd:sequence>
      <xsd:element name="attributeName" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="GetMOAttributesRequestType">
    <xsd:sequence>
      <xsd:element name="objectInstance" type="x782:NameType"/>
      <xsd:element name="attributeNameList"
type="moas:AttributeNameListType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="StatusType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="OperationSucceed"/>
      <xsd:enumeration value="OperationFailed"/>
    </xsd:restriction>
  </xsd:simpleType>

```

```

<xsd:complexType name="GetMOAttributesResponseType">
  <xsd:sequence>
    <xsd:element name="attributeNameAndValueList"
type="x782:AttributeNameAndValueSetType"/>
    <xsd:element name="status" type=" moas:StatusType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="ModifyOptionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="REPLACE"/>
    <xsd:enumeration value="ADDValues"/>
    <xsd:enumeration value="REMOVEValues"/>
    <xsd:enumeration value="SETToDefault"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AttributeNVMTType">
  <xsd:sequence>
    <xsd:element name="attributeName" type="xsd:string"/>
    <xsd:element name="attributeType" type="xsd:string"/>
    <xsd:element name="attributeValue" type="x782:AttributeValueType"/>
    <xsd:element name="modifyOption" type="moas:ModifyOptionType"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AttributeNVMLListType">
  <xsd:sequence>
    <xsd:element name="attributeNVM" type=" moas:AttributeNVMTType"
minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SetMOAttributesRequestType">
  <xsd:sequence>
    <xsd:element name="objectInstance" type="x782:NameType"/>
    <xsd:element name="attributeNVMLList" type="
moas:AttributeNVMLListType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CreateMORequestType">
  <xsd:sequence>
    <xsd:element name="objectClass" type="xsd:string"/>
    <xsd:element name="objectInstance" type="x782:NameType"/>
  </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="attributeNameAndValueList"
type="x782:AttributeNameAndValueSetType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="GetPackagesResponseType">
    <xsd:sequence>
        <xsd:element name="status" type="moas:StatusType"/>
        <xsd:element name="packages" type="x782:StringSetType"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

2) WSDL-определение услуги доступа к МО МСЭ

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- WSDL-определение работы услуги доступа к МО, описанной в настоящей
Рекомендации.

    Имя файла : x782_MOAccessService.wsdl -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:x782="http://www.itu.int/xml-namespace/itu-t/x.782"
xmlns:moas="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"
name="MOAccessService"
targetNamespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService">
<import namespace="http://www.itu.int/xml-namespace/itu-t/x.782"
location="x782.xsd"/>
<import namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"
location="x782_MOAccessService.xsd"/>

    <wsdl:message name="getMOAttributesRequest">
        <wsdl:part name="getMOAttributesInput"
type="moas:GetMOAttributesRequestType"/>
    </wsdl:message>
    <wsdl:message name="getMOAttributesResponse">
        <wsdl:part name="getMOAttributesOutput"
type="moas:GetMOAttributesResponseType"/>
    </wsdl:message>
    <wsdl:message name="setMOAttributesRequest">
        <wsdl:part name="setMOAttributesInput"
type="moas:SetMOAttributesRequestType"/>
    </wsdl:message>
    <wsdl:message name="setMOAttributesResponse">
        <wsdl:part name="status" type="moas:StatusType"/>
    </wsdl:message>
    <wsdl:message name="createMORequest">
        <wsdl:part name="createMOInput" type="moas:CreateMORequestType"/>
    </wsdl:message>
    <wsdl:message name="createMOResponse">

```

```

    <wsdl:part name="status" type="moas:StatusType"/>
</wsdl:message>
<wsdl:message name="deleteMORequest">
    <wsdl:part name="objectInstance" type="x782:NameType"/>
</wsdl:message>
<wsdl:message name="deleteMOResponse">
    <wsdl:part name="status" type="moas:StatusType"/>
</wsdl:message>
<wsdl:message name="getPackagesRequest">
    <wsdl:part name="objectInstance" type="x782:NameType"/>
</wsdl:message>
<wsdl:message name="getPackagesResponse">
    <wsdl:part name="getPackageOutput" type="moas:GetPackagesResponseType"/>
</wsdl:message>
<wsdl:portType name="MOAccessServicePortType">
    <wsdl:operation name="getMOAttributes">
        <wsdl:input message="moas:getMOAttributesRequest"/>
        <wsdl:output message="moas:getMOAttributesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="setMOAttributes">
        <wsdl:input message="moas:setMOAttributesRequest"/>
        <wsdl:output message="moas:setMOAttributesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="createMO">
        <wsdl:input message="moas:createMORequest"/>
        <wsdl:output message="moas:createMOResponse"/>
    </wsdl:operation>
    <wsdl:operation name="deleteMO">
        <wsdl:input message="moas:deleteMORequest"/>
        <wsdl:output message="moas:deleteMOResponse"/>
    </wsdl:operation>
    <wsdl:operation name="getPackages">
        <wsdl:input message="moas:getPackagesRequest"/>
        <wsdl:output message="moas:getPackagesResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="MOAccessServiceBinding"
type="moas:MOAccessServicePortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getMOAttributes">
        <soap:operation soapAction="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService/getMOAttributes"/>
        <wsdl:input>

```

```

        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="
http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="
http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="setMOAttributes">
    <soap:operation soapAction=" http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService/setMOAttributes"/>
    <wsdl:input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="
http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="createMO">
    <soap:operation soapAction="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService/createMO"/>
    <wsdl:input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="deleteMO">
    <soap:operation soapAction="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService/deleteMO"/>
    <wsdl:input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>

```

```

        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getPackages">
        <soap:operation soapAction="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService/getPackages"/>
        <wsdl:input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://www.itu.int/xml-namespace/itu-t/x.782/MOAccessService"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="MOAccessService">
    <wsdl:port name="MOAccessService" binding="moas:MOAccessServiceBinding">
        <soap:address location="http://www.itu.int/xml-namespace/itu-
t/x.782/MOAccessService"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Дополнение I

Обзор технологий и прикладных сценариев на основе веб-услуг в интерфейсах управления сетями

(Это дополнение не является неотъемлемой частью настоящей Рекомендации.)

В настоящем дополнении содержится обзор технологий на основе веб-услуг.

I.1 Характеристики технологий на основе веб-услуг

Веб-услуги обеспечивают упрощенный механизм подключения к приложениям независимо от используемых технологий или устройств, а также от местоположения. Они основаны на отраслевых стандартных протоколах с универсальной поддержкой поставщиков, которые могут использовать интернет для организации недорогой связи, а также другие транспортные механизмы. Подход, основанный на использовании сообщений со слабой связью, поддерживает множество сценариев взаимодействия и обмена информацией через услуги с самоописанием, которые устанавливаются автоматически.

В отличие от традиционных распределенных сред, веб-услуги делают упор на функциональную совместимость. Веб-услуги не зависят от конкретного языка программирования, тогда как традиционные среды, как правило, привязаны к тому или иному языку. Аналогичным образом, веб-услуги, поскольку их легко привязать к различным транспортным механизмам, предлагают большую гибкость при выборе этих механизмов. Кроме того, в отличие от традиционных сред, веб-услуги часто не увязаны с конкретными клиентскими или серверными платформами. В целом веб-услуги лучше подходят для взаимоотношений со слабой связью и низкими уровнями детализации. Использование XML дает веб-услугам дополнительное преимущество, поскольку XML позволяет использовать документы в неоднородной среде.

Веб-услуги – это основанные на XML-схеме технологии, которые обеспечивают следующие преимущества в программных приложениях.

1) Высокая функциональная совместимость

Веб-услуги универсально совместимы, поскольку используют протоколы, не зависящие от платформы и языка программирования, такие как SOAP. Технологии на основе веб-услуг обеспечивают новый уровень функциональной совместимости между программными приложениями. Многие поставщики программных платформ, разработчики программного обеспечения и поставщики утилит наделяют свое программное обеспечение возможностями SOAP, WSDL и UDDI.

2) Слабые связи

Веб-услуги представляют собой программные модули с самоописанием, инкапсулирующие отдельные функциональные возможности. Веб-услуги доступны через стандартные связи по протоколу Интернет, такие как XML и SOAP. Такие веб-услуги можно разрабатывать на многочисленных языках реализации, и доступ к ним могут получить любые другие приложения или веб-услуги. Таким образом, веб-услуги представляют собой приложения со слабой связью.

3) Широкое применение

В настоящее время веб-услуги широко применяются в индустрии ИТ-услуг, например в приложениях электронного бизнеса и приложениях "компания – компания". Уже создано несколько устойчивых платформ для поддержки разработки приложений на основе веб-услуг.

4) Повторное использование программного обеспечения и данных

Веб-услуги поддерживают компонентную модель разработки программного обеспечения, которая позволяет разработчикам повторно использовать конструктивные блоки, созданные другими, для сборки сложных приложений и их расширения новыми способами.

Это позволяет повторно использовать не только исходный код, но и данные, стоящие за исходным кодом повторного использования веб-услуг. Другим видом повторного использования программного

обеспечения в рамках веб-услуг является интеграция их функций в несколько соответствующих приложений и их предоставление через интерфейсы веб-услуг.

5) Простота компоновки услуг

Веб-услуги позволяют определять все более сложные приложения путем постепенного объединения компонентов на более высоких уровнях абстракции. Клиент, вызывающий составную услугу, может сам выступать в качестве веб-услуги. Объединяя функциональные возможности нескольких веб-услуг, можно создавать новые веб-услуги, что называется компоновкой услуг. Компоновку услуг можно осуществлять путем составления элементарных или составных услуг.

Веб-услуги обеспечивают стандартизованный способ предоставления функций приложений в качестве услуг и доступа к ним, и существуют специальные языки для определения и исполнения бизнес-процессов (такие, как BPEL).

б) Низкая стоимость

В настоящее время существует множество инструментов, продуктов и технологий, поддерживающих стандарты веб-услуг. Это предоставляет организациям широкий выбор вариантов, что помогает снизить затраты на разработку новых приложений, текущие расходы, затраты на охрану окружающей среды и стоимость интеграции.

I.2 Прикладные сценарии, рассчитанные и не рассчитанные на применение веб-услуг в интерфейсах управления сетями

1) Общие сведения о подходящих прикладных сценариях

В общем случае, исходя из характеристик веб-услуг, можно утверждать, что они подходят для следующих прикладных сценариев.

– Связь через защитные системы

Поскольку в качестве протокола передачи веб-услуги используют стандартный протокол SOAP, они могут легко проходить через защитные системы и прокси-серверы, расположенные между различными взаимосвязанными приложениями. При использовании другого межплатформенного программного обеспечения это обычно дается сложнее.

– Интеграция приложений

Известно, что приложениям часто приходится запускать программы на одной платформе и обмениваться данными с приложениями, работающими на других платформах. Даже на одной и той же платформе часто требуется интегрировать разнообразное программное обеспечение от разных производителей. Веб-услуги позволяют приложениям предоставлять свои функции и данные другим приложениям для их использования стандартными способами.

– B2B-интеграция интерфейсов

Интеграцию бизнес-операций между предприятиями обычно называют B2B-интеграцией. Используя веб-услуги, предприятие может интегрировать важнейшие бизнес-приложения и предоставлять их определенным поставщикам и клиентам, а самое большое преимущество использования веб-услуг для внедрения B2B-интеграции заключается в том, что они могут легко обеспечить достижение функциональной совместимости.

– Открытый интерфейс, поддерживающий видоизменяемость

Интерфейсы веб-услуг определяются на языке описания веб-услуг (WSDL). WSDL определяет услуги как наборы конечных точек или портов сети. Для этой цели спецификация WSDL предоставляет формат XML-документов. Абстрактное определение портов и сообщений отделено от случаев их конкретного использования, или экземпляров, что позволяет повторно использовать эти определения. Таким образом, WSDL описывает открытый интерфейс веб-услуг. Благодаря разделению определения данных и определения интерфейсов клиентские и серверные приложения на основе веб-услуг могут разрабатываться отдельно и взаимодействовать через этот открытый интерфейс, а изменение интерфейсов окажет слабое влияние на разработку таких приложений.

2) Общие сведения о неподходящих прикладных сценариях

Поскольку веб-услуги – это технологии, основанные на использовании слабо связанных компонентов, в основном предназначенные для взаимодействия и интеграции приложений в неоднородной среде, они могут иметь недостатки, такие как пониженная скорость исполнения и меньшая эффективность кодирования по сравнению с некоторыми другими технологиями (такими, как CORBA или DCOM). В некоторых случаях использование веб-услуг может оказаться неоптимальным выбором. К их числу относятся, в частности:

- системы на базе одной машины;
- изоморфные приложения локальной сети (LAN) (приложения, взаимодействующие в одной среде LAN на одной и той же платформе и с одним и тем же базовым межплатформенным программным обеспечением);
- интерактивное взаимодействие с большим количеством данных.

3) Соображения, касающиеся прикладных сценариев на основе веб-услуг в контексте управления сетями

Исходя из вышеприведенного анализа, наиболее подходящими для приложений на основе веб-услуг в контексте управления сетями можно считать следующие интерфейсы:

- интерфейс B2B/C2B;
- интерфейс F;
- высокоуровневый интерфейс ОС–ОС (уровень управления услугами или уровень управления предприятием) и т.п.

В указанных выше случаях можно эффективно использовать преимущества веб-услуг, такие как слабо связанная интеграция приложений между предприятиями, веб-доступ и достаточная расширяемость для внедрения новых приложений.

Возможно, не лучшим выбором является использование веб-услуг в интерфейсах управления EMS-NE, поскольку они обычно предоставляются одним и тем же поставщиком и могут не являться открытыми интерфейсами.

Веб-услуги можно использовать для интерфейсов управления NMS-EMS, но в некоторых случаях это может оказаться не самым удачным выбором. Например, CORBA обеспечивает большую эффективность кодирования и более высокую скорость исполнения в среде LAN.

Библиография

- [b-ITU-T X.780] Recommendation ITU-T X.780 (2001), *TMN guidelines for defining CORBA managed objects*.
- [b-ITU-T X.780.2] Recommendation ITU-T X.780.2 (2007), *TMN guidelines for defining service-oriented CORBA managed objects and façade objects*.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Оконечное оборудование, субъективные и объективные методы оценки
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных, взаимосвязь открытых систем и безопасность
Серия Y	Глобальная информационная инфраструктура, аспекты протокола Интернет и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи