

Международный союз электросвязи

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

M.3020

(07/2011)

СЕРИЯ М: УПРАВЛЕНИЕ ЭЛЕКТРОСВЯЗЬЮ,
ВКЛЮЧАЯ СУЭ И ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ
СЕТЕЙ

Сеть управления электросвязью

Методика определения интерфейсов управления

Рекомендация МСЭ-Т M.3020

ITU-T

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ М

УПРАВЛЕНИЕ ЭЛЕКТРОСВЯЗЬЮ, ВКЛЮЧАЯ СУЭ И ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ СЕТЕЙ

Введение и общие принципы технического обслуживания и организации технического обслуживания	M.10–M.299
Международные системы передачи	M.300–M.559
Международные телефонные каналы	M.560–M.759
Системы сигнализации по общему каналу	M.760–M.799
Международные системы телеграфной и фототелеграфной передачи	M.800–M.899
Международные арендованные первичные и вторичные групповые тракты	M.900–M.999
Международные арендованные каналы	M.1000–M.1099
Системы и службы подвижной электросвязи	M.1100–M.1199
Международная телефонная сеть общего пользования	M.1200–M.1299
Международные системы передачи данных	M.1300–M.1399
Обозначение и обмен информацией	M.1400–M.1999
Международная сеть транспортировки сообщений	M.2000–M.2999
Сеть управления электросвязью	M.3000–M.3599
Цифровые сети с интеграцией служб	M.3600–M.3999
Системы сигнализации по общему каналу	M.4000–M.4999

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

Рекомендация МСЭ-Т М.3020

Методика определения интерфейсов управления

Резюме

В Рекомендации МСЭ-Т М.3020 представлена методика определения интерфейсов управления (MISM). Описан процесс составления спецификаций интерфейсов на основе требований пользователя, анализа и проектирования (ТАП). Изложены руководящие указания в отношении ТАП с использованием нотации унифицированного языка моделирования (UML); однако не исключаются и другие методы определения интерфейсов. В настоящей Рекомендации МСЭ-Т приведено высокоуровневое описание Руководящих указаний по использованию UML.

Хронологическая справка

Издание	Рекомендация	Дата утверждения	Исследовательская комиссия
1.0	МСЭ-Т М.3020	05.10.1992 г.	
2.0	МСЭ-Т М.3020	27.07.1995 г.	4-я
3.0	МСЭ-Т М.3020	04.02.2000 г.	4-я
4.0	МСЭ-Т М.3020	22.07.2007 г.	4-я
5.0	МСЭ-Т М.3020	29.07.2008 г.	4-я
6.0	МСЭ-Т М.3020	14.05.2009 г.	2-я
7.0	МСЭ-Т М.3020	06.09.2010 г.	2-я
8.0	МСЭ-Т М.3020	14.07.2011 г.	2-я

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 с целью 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 извещения 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 2013

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 данной Рекомендации	2
4	Сокращения	3
5	Условные обозначения	4
6	Требования к методике и поддержка нотации	4
7	Методика	5
7.1	Общие положения	5
7.2	Применение и структура методики	5
7.3	Детальная методика	5
8	Спецификации интерфейсов управления	8
9	Возможность отслеживания в процессе MISM	8
10	Структура документации	8
Приложение А – Требования		9
A.1	Условные обозначения	9
A.2	Шаблон требований	12
A.3	Упрощенный шаблон требований	14
Приложение В – Анализ		16
B.1	Условные обозначения	17
B.2	Шаблон для анализа	19
B.3	Свойства ИОС и наследование	28
Приложение С – Набор символов UML MISM		30
C.1	Введение	30
C.2	Базовые элементы модели	30
C.3	Стереотипы	33
C.4	Классы ассоциации	40
C.5	Абстрактный класс	40
C.6	Применение <<InformationObjectClass>> и <SupportIOС>>	41
Приложение D – Проектирование		42
Приложение E – Определение типов информации – набор типов		43
E.1	Базовые типы	43
E.2	Тип перечислимых данных	43
E.3	Сложные/составные типы	43
E.4	Полезные типы	43
E.5	Ключевые слова	44

	Стр.
Приложение F – Руководящие указания по свойствам ИОС, наследованию и импорту объектов	45
F.1 Свойство ИОС	45
F.2 Наследование	46
F.3 Импорт объектов (интерфейсов, классов информационных объектов и атрибутов).....	47
Дополнение I – Пример требований	48
Дополнение II – Пример анализа	51
Дополнение III – Сравнение с Рекомендацией МСЭ-Т Z.601	59
Дополнение IV – Вопросы для дальнейшего изучения	60
IV.1 SOA	60
IV.2 UML	60
IV.3 Видимость	60
IV.4 Определение типов	60
Дополнение V – Дополнительные примеры использования UML	61
V.1 Прокси-класс	61
Дополнение VI – Руководящие указания по нумерации требований	63
Библиография	64

Рекомендация МСЭ-Т М.3020

Методика определения интерфейсов управления

1 Сфера применения

В данной Рекомендации описывается методика определения интерфейсов управления (MISM). Описана процедура формирования спецификаций межмашинного интерфейса на основании пользовательских требований, анализа и проектирования (ТАП). Руководящие указания приведены в рамках процесса ТАП с использованием нотации унифицированного языка моделирования (UML); однако не исключаются и другие методы определения интерфейсов. В настоящей Рекомендации описаны руководящие указания по использованию UML. Спецификация интерфейсов касается услуг управления, описанных в серии рекомендаций [ITU-T М.3200] и/или поддержки процессов управления, описанных в серии рекомендаций [ITU-T М.3050.x]. Такая спецификация может охватывать одну или несколько услуг управления или их частей. Услуги управления включают в себя функции управления. Это может касаться функций, описанных в Рекомендации [ITU-T М.3400] или процессов, описанных в серии рекомендаций [ITU-T М.3050.x], в зависимости от конкретной сферы управления. Кроме того, при необходимости могут определяться новые функции.

Данная методика может применяться как к традиционной модели интерфейсов управления "менеджер-агент" [ITU-T М.3010], так и к принципам сервисно-ориентированной архитектуры (АОУ), на которых строится архитектура управления сетями последующих поколений [ITU-T М.3060].

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 данной Рекомендации предлагается изучить возможность применения последнего издания Рекомендаций и других документов, приведенных ниже. Перечень действующих Рекомендаций МСЭ-Т публикуется регулярно. Ссылка на документ в данной Рекомендации, не придает ему как отдельному документу статус рекомендации.

- [ITU-T М.3010] Recommendation ITU-T М.3010 (2000), *Principles for a telecommunications management network.*
- [ITU-T М.3050.x] Рекомендация МСЭ-Т М.3050.x (2007 г.), *Расширенная карта бизнес-процессов оператора электросвязи (eТОМ).*
- [ITU-T М.3060] Рекомендация МСЭ-Т М.3060/Y.2401 (2006 г.), *Принципы управления сетями последующих поколений.*
- [ITU-T М.3200] Recommendation ITU-T М.3200 (1997), *TMN management services and telecommunications managed areas: Overview.*
- [ITU-T М.3400] Recommendation ITU-T М.3400 (2000), *TMN management functions.*
- [ITU-T Q.812] Рекомендация МСЭ-Т Q.812 (2004 г.), *Профили протоколов верхних уровней для интерфейсов Q и X.*
- [ITU-T X.680] Рекомендация МСЭ-Т X.680 (2008 г.) | ISO/IEC 8824-1:2008, *Информационная технология – Абстрактная синтаксическая нотация № 1 (ASN.1): спецификация базовой нотации.*
- [ITU-T X.681] Рекомендация МСЭ-Т X.681 (2008 г.) | ISO/IEC 8824-2:2008, *Информационная технология – Абстрактная синтаксическая нотация № 1 (ASN.1): спецификация информационных объектов.*

- [ITU-T X.722] Recommendation ITU-T X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.*
- [ITU-T Z.100] Рекомендация МСЭ-Т Z.100 (2007 г.), *Язык спецификации и описания (SDL).*
- [OMG UML] OMG: *Unified Modelling Language Specification, Version 1.5.*

Перечень ссылок на информативные справочные документы приведен в Библиографии.

3 Определения

3.1 Термины, определенные в других документах

В данной Рекомендации используются следующие термины из Рекомендации [ITU-T M.3010]:

- пользователь (user);
- услуга управления (management service);
- набор функций управления (management function set).

В данной Рекомендации используются следующие термины из [OMG UML]:

- диаграмма деятельности (activity diagram);
- актер (actor);
- ассоциация (association);
- класс (class);
- диаграмма класса (class diagram);
- классификатор (classifier);
- диаграмма кооперации (collaboration diagram);
- состав (composition);
- элемент модели (modelElement);
- диаграмма последовательности (sequence diagram);
- диаграмма состояний (state diagram);
- стереотип (stereotype);
- вариант использования (use case).

В данной Рекомендации используются следующие термины из Рекомендации [ITU-T M.3060]:

- эталонная точка (reference point).

3.2 Термины, определенные в данной Рекомендации

В данной Рекомендации определяются следующие термины:

3.2.1 агент (agent): Включает четко определенную подгруппу функций управления. Взаимодействует с менеджерами при помощи интерфейса управления. С точки зрения менеджера, поведение агента отслеживается только посредством интерфейса управления.

ПРИМЕЧАНИЕ. – Считается эквивалентом агента IRP [b-3GPP TS 32.150].

3.2.2 класс информационных объектов (information object class): Описывает информацию, которая может передаваться/использоваться в интерфейсах управления и моделируется при помощи стереотипа "Class" в метамодели UML. Формальное определение понятия класса информационных объектов и структура его спецификации приводятся в Приложении В.

3.2.3 информационное обслуживание (information service): Описывает информацию об управляемых объектах (сетевых ресурсах или объектах поддержки) и способе управления информацией в той или иной функциональной области. Информационное обслуживание определено для всех IRP.

ПРИМЕЧАНИЕ. – Считается аналогичным определению информационного обслуживания, приведенному в документе [b-3GPP TS 32.150].

3.2.4 тип информации (information type): Определение типа входных параметров операций.

3.2.5 объединенная эталонная точка (integration reference point): Архитектурная концепция, которая может быть описана при помощи набора спецификаций для определения того или иного аспекта интерфейса управления, в который входит спецификация требований, спецификация информационного обслуживания и одна или несколько спецификаций наборов решений.

ПРИМЕЧАНИЕ. – Считается аналогичным определению объединенной эталонной точки, приведенному в документе [b-3GPP TS 32.150].

3.2.6 цели управления (management goals): Цели высокого уровня для пользователя, осуществляющего операции в сфере управления.

3.2.7 интерфейс управления (management interface): Реализация возможностей управления между менеджером и агентом, что позволяет одному менеджеру использовать нескольких агентов, а одному агенту – поддерживать работу нескольких менеджеров.

ПРИМЕЧАНИЕ. – Примерами интерфейсов управления являются Q, C2B/B2B и Itf-N (3GPP).

3.2.8 роль управления (management role): Определяет направления деятельности, которые должны осуществляться обслуживающим персоналом и системами, обеспечивающими управление электросвязью. Роли управления определяются независимо от других компонентов, таких как ресурсы электросвязи и функции управления.

3.2.9 сценарий управления (management scenario): Сценарий управления представляет собой пример взаимодействий управления в отношении услуги управления.

3.2.10 менеджер (manager): Моделирует пользователя агента(ов) и напрямую взаимодействует с агентом(ами) с использованием интерфейсов управления.

Поскольку менеджер представляет пользователя агента, он дает четкое представление о том, что должен делать агент. С точки зрения агента, поведение менеджера отслеживается только посредством интерфейса управления.

ПРИМЕЧАНИЕ. – Считается эквивалентом менеджера IRP [b-3GPP TS 32.150].

3.2.11 сведения о соответствии (matching information): Спецификация типа параметра (например, ссылка на класс информационных объектов или атрибут класса информационных объектов).

3.2.12 спецификация, нейтральная в отношении протокола (protocol-neutral specification): Определяет интерфейсы управления, поддерживающие возможности управления независимо от протокола и формы представления информации, требуемой или предполагаемой например, CORBA и XML.

3.2.13 спецификация, определяемая протоколом (protocol-specific specification): Определяет интерфейсы управления, поддерживающие возможности управления для определенной технологии управления (например, CORBA).

ПРИМЕЧАНИЕ. – Считается эквивалентом набора решений [b-3GPP TS 32.150].

3.2.14 ресурсы электросвязи (telecommunications resources): Ресурсы электросвязи представляют собой физические или логические объекты, требующие управления с использованием услуг управления.

4 Сокращения

В данной	Рекомендации	используются	следующие	сокращения:
3GPP	3rd Generation Partnership Project			Проект партнерства третьего поколения
ADM	Administrative (usage: requirements category)			Административные (применение: категория требований)
ASN.1	Abstract Syntax Notation One			Абстрактная синтаксическая нотация версии 1
CM	Conditional-Mandatory			Условно обязательно
CO	Conditional-Optional			Условно необязательно
CON	Conceptual (usage: requirements category)			Концептуальные (применение: категория требований)

CORBA	Common Object Request Broker Architecture		Обобщенная архитектура посредника объектных запросов
FUN	Functional (usage: requirements category)		Функциональные (применение: категория требований)
GDMO	Guidelines for the Definition of Managed Objects		Руководящие указания по определению объектов управления
IDL	Interface Definition Language		Язык определения интерфейсов
IOC	Information Object Class		Класс информационных объектов
IRP	Integration Reference Point		Объединенная эталонная точка
IS	Information Service		Информационное обслуживание
MISM	Management Interface Specification Methodology		Методика определения интерфейсов управления
NA	Not Applicable	н/п	Не применяется
NE	Network Element	СЭ	Сетевой элемент
NON	Non-functional (usage: requirements category)		Нефункциональные (применение: категория требований)
OMG	Object Management Group		Рабочая группа по управлению объектами
OO	Object Oriented	ОО	Объектно-ориентированный
OSI	Open Systems Interconnection	ВОС	Взаимосвязь открытых систем
SDL	Specification and Description Language		Язык спецификации и описания
SOA	Service Oriented Architecture	АОУ	Сервисно-ориентированная архитектура
SS	Solution Set		Набор решений
TS	Technical Specification		Техническая спецификация
UML	Unified Modelling Language		Унифицированный язык моделирования
XML	eXtensible Markup Language		Расширяемый язык разметки

5 Условные обозначения

В пункте А.1 содержатся условные обозначения, применимые на стадии требований.

В пункте В.1 содержатся условные обозначения, применимые на стадии анализа.

6 Требования к методике и поддержка нотации

К разработке методике и выбору нотации применяются следующие требования:

- 1) В методике, включая выбор нотации, должны учитываться все соответствующие требования рассматриваемой области, а именно управления электросвязью.
- 2) Методика способствует разработке требований, определяющих Анализ | Информационное обслуживание и соответствующе Спецификации проектирования | Наборы решений.
- 3) Нотация должна способствовать недвусмысленному формированию спецификации в целевом профиле протокола управления. Методика не определяет возможный выбор службы протокола (например, CORBA Security Service).
ПРИМЕЧАНИЕ. – Протоколы управления, применяемые МСЭ-Т, указаны в документе [ITU-T Q.812].
- 4) Методика должна позволять формировать спецификации обязательных и необязательных объектов на всех трех стадиях. Кроме того, она определяет взаимосвязи обязательных и необязательных объектов в разрезе трех стадий.

- 5) Следует предусмотреть возможность формулирования, на основании спецификации, нейтральной в отношении протокола (Анализ | IS), взаимодействующих определений, ориентированных на конкретный язык, т. е. Проектирование | SS (например, UML для IDL, UML для GDMO/ASN.1).

7 Методика

7.1 Общие положения

Целью данной методики является описание процедур, которые позволяют определить интерфейсы межмашинного управления.

7.2 Применение и структура методики

Методика определения интерфейсов управления (MISM) определяет процедуру, включающую в себя три стадии, со свойствами, позволяющими обеспечить возможность отслеживания на всех трех стадиях. На трех стадиях используются принятые в отрасли методы с применением принципов объектно-ориентированного анализа и проектирования. Три стадии – это стадии требований, анализа и проектирования. Методы должны позволять использовать или разрабатывать серийно выпускаемые инструменты поддержки. В зависимости от характера проблемы на разных стадиях могут использоваться разные методы.

7.3 Детальная методика

7.3.1 Общие положения

На стадиях требований и анализа результатом работы является проектирование спецификаций UML. На стадии проектирования используется нотация, основанная на парадигме управления сетями. Результаты завершения каждой из трех стадий приведены ниже:

- Стадия требований – требования.
- Стадия анализа – независимая от реализации спецификация.
- Стадия проектирования – спецификация, основанная на конкретной технологии.

Первоначально стадия проектирования проводится вручную или же для ее выполнения используется индивидуальный подход. Когда взаимодействующее определение, ориентированное на определенный протокол, может быть сформировано посредством инструментов, нотация UML может применяться к стадии проектирования.

Ниже три указанных стадии описаны более подробно.

7.3.2 Требования

Требования в отношении решаемой проблемы могут быть разделены на два класса. Первый класс требований далее по тексту мы будем называть коммерческими требованиями. Эксперт в соответствующей отрасли должен быть в состоянии установить, что требования должным образом отражают потребности в решении конкретной проблемы управления. Второй класс обозначается как требования спецификации. Данные требования должны быть достаточно подробными для формирования определения интерфейсов на стадиях анализа и проектирования. Поскольку между окончательными определениями интерфейса и требованиями должна прослеживаться взаимосвязь, может потребоваться обеспечение взаимодействия между тремя стадиями. В ходе такого взаимодействия следует устранить любую неопределенность в требованиях для обеспечения возможности формирования реализуемой спецификации.

Данные интерфейса "человек-машина" могут конкретизироваться в числе требований второго класса. Такие требования могут оказывать существенное влияние на концепции и данные, разрабатываемые на последующих стадиях. Более подробную информацию можно найти в Дополнении III, а также в Рекомендациях МСЭ-Т серии М.1400 по разработке данных для интерфейсов "человек-машина".

Для определения двух указанных классов требований можно использовать различные методы. Независимо от применяемого метода важнейшую роль играет удобочитаемость требований. Сами требования не обязательно должны быть изложены в машиночитаемой форме, при условии соблюдения удобочитаемости и возможности отслеживания требований. Рекомендуется производить нумерацию требований с целью разграничения различных требований для обеспечения возможности их отслеживания.

На стадии требований происходит определение таких аспектов, как политика безопасности, границы проблемной области в разрезе возможностей применения, ресурсов и ролей, которые берут на себя ресурсы. В требованиях указываются роли, сферы ответственности и связи между составляющими объектами рассматриваемой области. Для обозначения требований коммерческого уровня могут применяться различные методы, в том числе их текстовое обозначение. С целью обеспечения возможности отслеживания этих требований на стадиях проектирования и внедрения рекомендуется выполнять их нумерацию.

Проблема должна быть ограничена определенной областью. Одним из способов определения такой области является использование услуг управления, определенных в Рекомендации [ITU-T M.3200], и наборов функций, определенных в Рекомендации [ITU-T M.3400]. Требования указаны с использованием управляемых ресурсов и функций управления. Альтернатива подходу, основанному на услугах управления, описана в Рекомендации [ITU-T M.3050.x] "Расширенная карта бизнес-процессов оператора электросвязи (еТОМ)", в которой представлен подход, основанный на бизнес-процессах.

Взаимосвязь между подходами, представленными в Рекомендациях [ITU-T M.3200] и [ITU-T M.3050], описана в Рекомендациях [ITU-T M.3050.x].

Функции управления должны группироваться и поддерживаться в рамках приложений, отвечающих конкретным коммерческим потребностям, поэтому связь между процессами еТОМ, услугами управления [ITU-T M.3200], наборами функций управления [ITU-T M.3400] и функциями управления важна для обеспечения их четкой и эффективной группировки. Для выполнения коммерческих требований той или иной проблемы может потребоваться дополнение [ITU-T M.3400].

При взаимодействии с отраслевыми экспертами для сбора данных по требованиям коммерческого уровня должны применяться варианты и сценарии использования UML. В требованиях также должны определяться условия возникновения сбоя, видимые в рамках бизнес-процесса.

ПРИМЕЧАНИЕ. – Требования не обязательно должны быть выражены в форме вариантов использования.

Разработанные требования должны быть полными и подробными. Рекурсивный характер данной методики необходим для обеспечения полноты требований. Полнота требований (которые должны быть четкими и подтвержденными документально) является определяющим фактором на стадиях анализа и проектирования.

Руководящие указания и шаблон структуры и идентификации требований приводятся в пункте А.1.2.

Варианты использования представляют собой цели, которые достигаются посредством реализации последовательности шагов. Каждый из шагов может рассматриваться как подцель варианта использования. Каждый такой шаг представляет собой или отдельный вариант использования (нижнего уровня), либо независимое действие, которое отвечает самому нижнему уровню разбивки такого варианта.

Руководящие указания и шаблон вариантов использования приводятся в пункте А.1.2.

Пример определения требований приводится в Дополнении I.

7.3.3 Анализ

На стадии анализа требования используются для выявления взаимодействующих объектов, их свойств и взаимосвязи между ними. Это позволяет определить интерфейсы, предлагаемые такими объектами. В нотации UML такие объекты становятся классами. Должна проследиваться связь между описанием классов, а также сопоставимыми интерфейсами и требованиями. Отклонения между классами, определенными в спецификации на стадии анализа, и классами, указанными в спецификации на стадии проектирования, не обязательно должны точно совпадать.

На данной стадии следует принимать во внимание потребности данного интерфейса "человек-машина" (т. е. информационная модель должна содержать в себе достаточно информации для обеспечения проектирования спецификаций на основании результатов анализа).

В данной Рекомендации предлагаются руководящие указания высокого уровня по использованию нотации UML для поддержки определения интерфейсов управления; вместе с тем в качестве дополнения определений UML может использоваться SDL [ITU-T Z.100].

На стадии анализа не должно применяться никаких ограничений по разработке. Например, результаты анализа могут отражаться в документах на основании принципов объектно-ориентированной технологии, при этом само проектирование может на ней не базироваться.

Информация, которая конкретизируется на стадии анализа, включает описание классов, определения данных, взаимосвязи классов, диаграммы взаимодействий (диаграммы последовательности и/или диаграммы кооперации), диаграммы перехода состояний и диаграммы деятельности. Определения классов включают в себя спецификации операций, уведомлений, атрибутов и поведения, отображаемые в виде примечаний или текстовых описаний.

Общие услуги управления, нейтральные в отношении протокола (при наличии) – или другие имеющиеся услуги – должны повторно использоваться на стадии анализа с целью поддержки гармонизации интерфейса управления.

Руководящие указания и шаблон вариантов использования приводятся в Приложении А.

В шаблоне анализа используется тип информации как одна из характеристик для описания атрибутов класса информационных объектов и параметров операции/уведомления. Действительные типы информации, которые могут быть использованы и их семантика определяются в Приложении Е.

7.3.4 Проектирование

7.3.4.1 Общие положения

На стадии проектирования осуществляется подготовка реализуемой спецификации межоперационного интерфейса. Это предполагает выбор целевого языка спецификации. Спецификации на стадии проектирования зависят от конкретной парадигмы управления (например, IDL для интерфейсов CORBA).

На данной стадии различают три вида спецификаций данных: проектирование данных в зависимости от парадигмы управления (например, XML) для передачи по различным интерфейсам (например, в отношении неисправностей и рабочих характеристик), сообщений (например, аварийных отчетов) для передачи по каждому конкретному интерфейсу и метод кодирования данных (например, сжатый язык XML), соответствующий той или иной парадигме.

Вопрос выбора той или иной парадигмы управления рассматривается в других Рекомендациях МСЭ-Т. Обзор данного вопроса приводится в последующих пунктах.

На стадии проектирования рекомендуется обращаться к описаниям UML, полученным на стадиях требований и анализа, для дополнения спецификаций поведения. Например, для определения поведения GDMO можно обращаться к схемам состояния, диаграммам последовательности и определению класса, полученному на стадии анализа. При необходимости могут также применяться дополнительные диаграммы UML с описанием взаимодействия между объектами в соответствии с парадигмой того или иного протокола.

По мере принятия дополнительных парадигм для использования в управлении начинают применяться нотации/языки, определяемые такими парадигмами.

7.3.4.2 CORBA

В контексте управления на базе CORBA информационная модель определяется с использованием IDL.

7.3.4.3 GDMO

В контексте парадигмы на базе управления системами OSI [ITU-T X.722] спецификация проектирования представляет собой спецификацию информационной модели с использованием шаблонов GDMO для управляемых классов объектов, атрибутов, поведения, уведомлений, действий, примеров названий классов и спецификаций ошибок/исключений. Синтаксис информации указывается с использованием нотации ASN.1 [ITU-T X.680].

В GDMO в иерархии классов объектов указываются свойства классов объектов, необходимые для управления. Получение максимальной пользы от повторного применения спецификаций требует широкого использования наследования (суперклассов и подклассов). Классы объектов указываются с применением шаблонов, содержащихся в Рекомендации [ITU-T X.722]. Шаблоны, определяющие информационную модель, должны регистрироваться (в соответствии с положениями Рекомендации [ITU-T X.722]) со значением для идентификатора объекта ASN.1. Для тех классов объектов, которые уже определены в других Рекомендациях МСЭ-Т и в стандартах ИСО, необходимо указывать только ссылку на такую Рекомендацию и класс объектов. Присвоение наименования не является составляющей иерархии классов объектов или же ее целью.

7.3.4.4 XML

Для дальнейшего изучения.

8 Спецификации интерфейсов управления

Спецификация интерфейса управления включает в себя спецификации требований, анализа и проектирования, описанные в пункте 7. Структура формирования таких спецификаций приводится в Приложениях А, В и С.

Такие методы и соответствующие нотации также могут применяться при разработке системы спецификации интерфейсов управления, хотя проектирование систем и не входит в Рекомендации по управлению МСЭ-Т. Они помогают описать, каким образом спецификации интерфейса применяются к управлению ресурсами в рамках системы, такой как СЭ.

9 Возможность отслеживания в процессе MISM

Обеспечение возможности отслеживания связей между требованиями, анализом и проектированием предусматривает присвоение соответствующих идентификаторов. Возможность отслеживания обеспечивается путем использования ссылок между объектами на каждой стадии, а также между стадиями. Возможность отслеживания обеспечивается от компонента проектирование | набор решений к компоненту анализ | информационное обслуживание, а также от компонента анализ | информационное обслуживание к требованиям. Отслеживание также применяется между артефактами спецификации требований, а также между артефактами анализа | информационного обслуживания, например, между вариантами использования и требованиями в текстовой форме. Требования должны определяться в соответствии с пунктом 7.3.2. Результаты стадии анализа определяют дальнейшие детальные требования к информации для различных вариантов использования. На стадии проектирования должны использоваться указания на различные диаграммы и текстовую информацию, полученную по результатам стадии анализа. Такое указание может принимать форму ссылок на соответствующие положения.

Отслеживание от стадии проектирования к уровню фактических требований обычно осуществляется в непрямой форме. Это необходимо, поскольку результаты данных стадий могут иметь различный уровень детализации.

Руководящие указания в отношении отслеживания между стадиями требований и анализа приводятся в Приложении В.

Рекомендуется применять следующий механизм отслеживания в отношении требований и т. п., указанных в других документах (возможно отступление от рекомендованной схемы идентификации):

forum/body "::<" document ID "::<" id,

где "id" может означать один из приведенных ниже вариантов:

- 1) идентификатор требования;
- 2) идентификатор варианта использования;
- 3) заголовок/текст требования;
- 4) заголовок варианта использования;
- 5) подпункт документа, который содержит уникальную идентификацию требования или варианта использования.

Примеры:

3GPP::32.111-1::getAlarmList

ITU-T::M.3016::1.5.1.2.

10 Структура документации

Несмотря на наличие трех стадий, результаты, полученные на таких стадиях, могут приводиться в одном или нескольких документах по тому или иному интерфейсу. Рекомендуется объединять документы, разработанные на стадиях требований и анализа, а также готовить отдельные документы на стадии проектирования по каждой конкретной парадигме протоколов управления сетями.

Приложение А

Требования

(Данное приложение является неотъемлемой частью данной Рекомендации.)

- A.1 Условные обозначения*
 - A.1.1 Использование нотации UML в требованиях*
 - A.1.2 Шаблон варианта использования*
 - A.1.3 Категории требований*
- A.2 Шаблон требований*
 - 1 Концепции и базовая информация*
 - 2 Требования коммерческого уровня*
 - 2.1 Требования*
 - 2.2 Роли актеров*
 - 2.3 Ресурсы электросвязи*
 - 2.4 Варианты использования высокого уровня*
 - 3 Требования уровня спецификаций*
 - 3.1 Требования*
 - 3.2 Роли актеров*
 - 3.3 Ресурсы электросвязи*
 - 3.4 Варианты использования*
- A.3 Упрощенный шаблон требований*
 - 1 Концепции и базовая информация*
 - 2 Требования*

Ниже приведены руководящие указания для спецификации требований. Вариант использования этого шаблона можно найти в Дополнении I.

Стандартную (или полную) версию шаблона требований можно найти в пункте А.2. Кроме того, в пункте А.3 приводится и определяется упрощенная версия шаблона требований.

A.1 Условные обозначения

A.1.1 Использование нотации UML в требованиях

В таблице А.1 приведено соответствие между концепциями управления и нотацией UML. В данной Рекомендации определяются концепции высокого уровня и нотации, которые должны использоваться на различных стадиях. Для расширения нотации UML применяются стереотипы. Стереотипы, утвержденные для использования в среде управления, включены в данную Рекомендацию (см. Приложение С).

Таблица А.1 – Концепции требований

Концепция управления	Нотация UML	Комментарии
Пользователь	Актер	Пользователь моделируется как актер
Роль управления	Актер	Актер играет роль. Как правило, рекомендуется моделирование одной роли для каждого актера
Функция управления	Вариант использования	Функция управления моделируется одним или более вариантами использования
Набор функций управления	Вариант использования	Набор функций управления представляет собой составной вариант использования, в рамках которого каждая функция управления (теоретически) моделируется в виде отдельного варианта использования
Услуга управления	Вариант использования	Услуга управления моделируется как вариант использования высокого уровня
Сценарий управления	Диаграмма последовательности	Диаграммы последовательности являются предпочтительными по сравнению с диаграммами кооперации
Тип ресурсов электросвязи	Класс	Диаграммы класса отражают детальные свойства типа ресурсов электросвязи, при этом уровень детализации соответствует стадии методики
Цели управления	–	Цели управления фиксируются в форме текстовых описаний за неимением соответствующей нотации UML

А.1.2 Шаблон варианта использования

Если имеются варианты использования, следует применять приведенные ниже условные обозначения и шаблоны.

Таблица А.2 – Шаблон варианта использования

Этап вариантов использования	Эволюция/Спецификация	<<Uses>> Соответствующее использование
Цель(*)	<p>Это задача/конечный результат, который должен быть получен в данном варианте использования. Цель должна быть представлена в форме краткого утверждения о том, что должно быть достигнуто в случае успешной реализации варианта использования.</p> <p>Может быть сформулировано утверждение в отношении приоритета по сравнению с другими вариантами использования и требуемой эффективности реализации варианта использования, например:</p> <ul style="list-style-type: none"> • В режиме реального времени. • Практически в режиме реального времени. • Не в режиме реального времени. 	
Актеры и роли(*)	Названия актеров/ролей, задействованных в варианте использования, включая характеристику роли каждого актера.	
Ресурсы электросвязи	Названия ресурсов электросвязи, задействованных в варианте использования.	
Допущения	<p>Описание среды, формирующей контекст того или иного варианта использования.</p> <p>Допущения и предварительные условия являются взаимоисключающими.</p> <p>Допущения имеют отношение к статическим свойствам.</p>	
Предварительные условия	<p>Перечень всех условий, которые должны соблюдаться в системе и в окружающей ее среде, перед началом реализации варианта использования.</p> <p>Предварительные условия и допущения являются взаимоисключающими.</p> <p>Предварительные условия имеют отношения к динамическим свойствам и могут формировать исключения. С допущениями этого никогда не происходит.</p>	

Таблица А.2 – Шаблон варианта использования

Этап вариантов использования	Эволюция/Спецификация	<<Uses>> Соответствующее использование
Когда начинается	Название конкретного события, которое служит толчком для начала реализации варианта использования. Данный пункт является необязательным и обычно не используется для обозначения таких событий, как "когда менеджеру необходимо извлечь информацию".	
Шаг 1(*) (М О)	В варианте использования приводится перечень шагов (совершаемых вручную или автоматически), которые необходимы для достижения цели, поставленной в данном варианте использования. Шаги могут вызывать реализацию других вариантов использования. Шаги нумеруются для обеспечения возможности отслеживания. Каждый шаг определяется как обязательный (М) или необязательный (О). Промежуточные шаги определяются в привязке к основному шагу, например: Шаг n Шаг n.1 Шаг n.2 где n.1 и n.2 – это промежуточные шаги в рамках шага n.	Ссылка на применяемый вариант использования
Шаг n (М О)	Шаги добавляются по мере необходимости, в логической последовательности.	
Когда заканчивается(*)	Перечень событий, которые знаменуют завершение реализации варианта использования. ПРИМЕЧАНИЕ. – В данном контексте термин "событие" может толковаться в самом широком смысле и не ограничивается, например, уведомлениями, направляемыми в рамках интерфейса управления. Например, завершение процесса обработки может считаться событием, указывающим на завершение реализации варианта использования.	
Исключения	Итоговый перечень исключений и неисправностей, выявленных в результате реализации варианта использования.	
Заключительные условия	Перечень всех условий, которые должны соблюдаться в системе и окружающей ее среде после завершения реализации варианта использования. Перечень заключительных условий указывает на то, должен ли вариант использования быть полностью успешным, частично успешным или же должен оказаться неудачным для того, чтобы реализация варианта использования считалась завершенной.	
Отслеживание(*)	Требования или варианты использования, на которые ссылается данный вариант использования.	
ПРИМЕЧАНИЕ. – Поля, помеченные символом "*" являются обязательными для всех спецификаций вариантов использования. Другие поля являются обязательными только для отдельных вариантов использования.		

А.1.3 Категории требований

Целесообразно классифицировать требования по различным категориям. Считается, что для MISM могут использоваться следующие категории требований:

- Концептуальные (CON) – определяют концепцию, тип данных, взаимосвязь, формат или структуру.
- Функциональные (FUN) – определяют функциональные возможности, динамическую ситуацию, последовательность, временные параметры или взаимодействие.
- Нефункциональные (NON) – нефункциональные требования, в том числе в отношении аварийного состояния, неисправного состояния и диапазон характеристик.
- Административные (ADM) – требования в отношении системного администрирования или функциональности, не связанные с нормальной реализацией вариантов использования.

Требования должны оформляться в письменной форме с использованием следующего шаблона:

REQ-Label-Category-Number {Category, number} Details {Source Citation},

где "Label" – это сокращение для Рекомендации (или ее части). Набор меток не ограничен и не подлежит стандартизации.

Руководящие указания в отношении нумерации требований приведены в Дополнении VI.

A.2 Шаблон требований

1 Концепции и базовая информация

Определить основные цели, задачи и применимые интерфейсы управления (и эталонные точки) для данной спецификации. Использовать разбивку на категории, приведенную в Рекомендации [ITU-T M.3200] в качестве источника определения услуг управления, которые поддерживаются интерфейсом.

В данном подпункте должно приводиться четкое описание преимуществ для пользователей, т. е. основания для реализации такой услуги управления. В случае необходимости может добавляться справочная информация и контекст, однако при этом пояснительная и описательная части должны быть разделены. Необходимая справочная информация, если она требуется, должна приводиться в приложении.

1.a SubClauseTitle

SubClauseTitle представляет собой название подпункта.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового подпункта.

Использование подпунктов является необязательным.

2 Требования коммерческого уровня

2.1 Требования

2.1.a SubSetTitle

SubSetTitle представляет собой название поднабора требований коммерческого уровня.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового поднабора.

Использование поднаборов является необязательным, и все требования коммерческого уровня могут быть указаны в подпункте 2.1 (требования).

Указать основные требования в тексте и определить варианты использования с указанием актера/роли и ресурсов. Варианты использования высокого уровня (подпункт 2.4, ниже) должны соответствовать требованиям коммерческого уровня и отличаются от требований спецификации, поскольку они не детализируются до нижних уровней. В пункте 2.4 содержится много примеров того, что представляют собой варианты использования высокого уровня. Информация, касающаяся политики (например, в отношении безопасности, сохранности) может относиться к такому уровню. Для обеспечения возможности отслеживания необходимо использовать нумерацию требований.

Требования должны определяться, как указано в пункте A.1.3. В рамках спецификации требований предлагается указывать требования в последовательности, определенной в пункте A.1.3 (для всей спецификации или для каждого поднабора).

Использование категорий требований является необязательным, и – в случае их использования – могут применяться поднаборы категорий.

Например, концептуальное требование номер 23 в рекомендации с меткой "SM" должно быть указано следующим образом:

Идентификатор	Определение
REQ-SM-CON-23	Служебный приказ содержит название, адрес, номер телефона, описание услуги и (необязательно) номер факса для контактов {T1M1.5 Document 246 11/96}

При необходимости может использоваться одна или несколько таблиц с сопроводительным текстом между такими таблицами.

2.2 Роли актеров

Приводится текстовое описание актера (см. пункт 3).

2.3 Ресурсы электросвязи

Приводится текстовое описание соответствующих ресурсов (см. пункт 3), необходимых для поддержки реализации вариантов использования.

2.4 Варианты использования высокого уровня

Может быть представлена диаграмма вариантов использования высокого уровня. Для понимания отраслевыми экспертами вариантов использования, каждый из таких вариантов должен быть дополнен

соответствующим текстовым описанием. Такое описание должно выполнять две задачи: обобщение знаний отраслевых экспертов и подтверждение моделей, используемых на стадиях анализа и проектирования, в контексте требований. Пример диаграммы вариантов использования высокого уровня приводится в Дополнении I.

2.4.a UseCaseName

UseCaseName представляет собой название варианта использования.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового определения варианта использования.

Данный подпункт повторяется для каждого варианта использования высокого уровня, определяемого для требований спецификации интерфейса.

Варианты использования высокого уровня могут определять различные наборы функций, указанные в Рекомендации [ITU-T M.3400], или процессы управления, указанные в Рекомендации [ITU-T M.3050.x]. Такие варианты использования могут быть представлены более детально, как описано в приведенном ниже подпункте в отношении требований уровня спецификации, с использованием стереотипов, таких как "include" и "extend".

При необходимости могут использоваться диаграммы последовательности. Однако применительно к требованиям высокого уровня такие диаграммы обычно не используются. Когда варианты использования этого уровня более подробно детализируются в разрезе требований следующего уровня, применение таких диаграмм может быть более целесообразно.

Отслеживание следующего уровня требований с этого уровня определяется тем, каким образом каждый набор функций дополнительно детализируется новыми вариантами использования.

Набор таблиц с вариантами использования, составленный при помощи шаблона, приведенного в таблице A.2, может использоваться для представления существенных возможностей, изучаемых на уровне абстракции, соответствующем анализируемой проблеме.

Уровень детализации, а также степень охвата, предусмотренная вариантами использования, зависит от того, насколько команда разработчиков знакома с существом вопроса и, следовательно, насколько субъективны ее суждения. Более низкие уровни детализации с большей вероятностью указывают на проведение анализа, чем на сбор информации о требованиях.

Допускается проведение более детального последовательного анализа каждого шага варианта использования более высокого уровня абстракции путем ссылки на более детальный вариант использования в ячейке таблицы, предназначенной для этой цели. Следует подчеркнуть, что эти действия не являются необходимыми, и их выполнение зависит от потребностей автора/группы.

Приведенный ниже список предназначен для того, чтобы помочь первоначальной идентификации подходящих вариантов использования:

- Каково основное предназначение системы?
- Какие типы людей/систем должны взаимодействовать с системой?
- Каким образом эти люди/системы могут быть сгруппированы или привязаны к ролям?
- Какие аспекты системы связаны с ее запуском, нормальным функционированием, сбоями в работе и восстановлением?
- Получение каких типов отчетов или данных может потребоваться от системы?
- Какие потребуются специальные действия (например, исходя из времени суток и загрузки сети)?

Целесообразно применять общую процедуру для документального оформления вариантов использования.

Предлагается следующая структура:

- <use case table> (см. таблицу A.2)
- <optional sequence diagram(s)>
- <optional state chart(s)>

3 Требования уровня спецификаций

3.1 Требования

В данном разделе производится дальнейшая детализация требований коммерческого уровня в разрезе функций управления, приведенных в Рекомендации [ITU-T M.3400]. Поскольку Рекомендация [ITU-T M.3400] не является исчерпывающей и не охватывает все услуги управления во всех сферах, ожидается, что перечень функций будет расширяться. Новые функции должны включаться в требования, как описано ниже.

3.1.a SubSetTitle

SubSetTitle представляет собой название поднабора требований уровня спецификации.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового поднабора.

Использование поднаборов является необязательным, и все требования уровня спецификации могут быть указаны в подпункте 3.1 (требования).

Указать основные детальные и конкретные требования в тексте и определить варианты использования с указанием актера/роли и ресурсов. Варианты использования, указанные в подпункте 3.4, должны соответствовать требованиям уровня спецификации с указанием деталей нижнего уровня и должны быть в большей степени ориентированы на реализацию по сравнению с требованиями к вариантам использования коммерческого уровня. Для обеспечения возможности отслеживания необходимо использовать нумерацию требований.

Требования должны определяться, как указано в пункте А.1.3. В рамках спецификации требований предлагается указывать требования в последовательности, определенной в пункте А.1.3 (для всей спецификации или для каждого поднабора).

Использование категорий требований является необязательным, и – в случае их использования – может применяться подбор категорий.

Например, функциональное требование номер 33 в рекомендации с меткой "ОМ" должно быть указано следующим образом:

Идентификатор	Определение
REQ-OM-FUN-33	Отложенная операция может быть отменена ее инициатором.

При необходимости может использоваться одна или несколько таблиц с сопроводительным текстом между такими таблицами.

Для требований уровня спецификации должны применяться условные обозначения и шаблоны, приведенные в пункте А.1.

3.2 Роли актеров

Данный раздел содержит перечень всех актеров и текстовое описание актеров, приведенное в отношении требований коммерческого уровня.

3.3 Ресурсы электросвязи

Данный раздел содержит перечень всех пассивных ресурсов и текстовое описание ресурсов, которые не были определены в отношении требований коммерческого уровня.

3.4 Варианты использования

Варианты использования высокого уровня более подробно детализируются в данном разделе посредством использования нескольких вариантов использования уровня спецификации, каждый из которых далее объясняется в отдельном подпункте, как описано ниже.

3.4.a UseCaseName

UseCaseName представляет собой название варианта использования.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового определения варианта использования.

При необходимости могут использоваться диаграммы последовательности и состояния.

ПРИМЕЧАНИЕ. – Руководящие указания и критерии использования диаграмм последовательности и состояния будут разработаны в дальнейшем.

Для спецификаций вариантов использования должны применяться условные обозначения и шаблоны, приведенные в пункте А.1.

А.3 Упрощенный шаблон требований

Упрощенный шаблон требований представляет собой альтернативный шаблон, который должен применяться в случаях, когда необходимы только требования в текстовой форме. Для этих целей разработан отдельный шаблон, что позволяет избежать неоднозначности, которая возникла бы в результате добавления дополнительных вариантов в стандартный шаблон, приведенный в пункте А.2.

1 Концепции и базовая информация

Определить основные цели, задачи и применимые интерфейсы управления (и эталонные точки) для данной спецификации. Использовать разбивку на категории, приведенную в Рекомендации [ITU-T M.3200] в качестве источника определения услуг управления, которые поддерживаются таким интерфейсом.

В данном пункте должно приводиться четкое описание выгоды пользователей, т. е. основания для реализации такой услуги управления. В случае необходимости может добавляться справочная информация и контекст, однако при этом пояснительная и описательная части должны быть разделены. Необходимая справочная информация, если она требуется, должна приводиться в приложении.

1.a SubClauseTitle

SubClauseTitle представляет собой название подпункта.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового подпункта.

Использование подпунктов является необязательным.

2 Требования

2.a SubSetTitle

SubSetTitle представляет собой название поднабора требований коммерческого уровня.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового поднабора.

Использование поднаборов является необязательным, и все требования коммерческого уровня могут быть указаны в пункте 2 (требования).

Указать основные требования в тексте и определить варианты использования с указанием актера/роли и ресурсов. Варианты использования должны соответствовать требованиям высокого уровня и отличаются от требований спецификации, поскольку они не детализируются до нижних уровней. Информация, касающаяся политики (например, в отношении безопасности, сохранности) может относиться к такому уровню. Для обеспечения возможности отслеживания, необходимо использовать нумерацию требований.

Требования должны определяться, как указано в подпункте A.1.3. В рамках спецификации требований предлагается указывать требования в последовательности, определенной в подпункте A.1.3 (для всей спецификации или для каждого поднабора).

Использование категорий требований является необязательным, и – в случае их использования – могут применяться поднаборы категорий.

Например, концептуальное требование номер 23 в Рекомендации с меткой "SM" должно быть указано следующим образом:

Идентификатор	Определение
REQ-SM-CON-23	Служебный приказ содержит название, адрес, номер телефона, описание услуги и (необязательно) номер факса для контактов {T1M1.5 Document 246 11/96}

При необходимости может использоваться одна или несколько таблиц с сопроводительным текстом между такими таблицами.

Приложение В

Анализ

(Данное приложение является неотъемлемой частью данной Рекомендации.)

- В.1 Условные обозначения*
 - В.1.1 Обязательные, необязательные и условные квалификаторы*
- В.2 Шаблон для анализа*
 - 1 Концепции и базовая информация*
 - 2 Классы информационных объектов*
 - 2.1 Импортированные информационные объекты и локальные метки*
 - 2.2 Диаграмма класса*
 - 2.2.1 Атрибуты и связи*
 - 2.2.2 Наследование*
 - 2.3 Определения классов информационных объектов*
 - 2.3.a InformationObjectClassName*
 - 2.4 Определение информационных связей*
 - 2.4.a InformationRelationshipName (supportQualifier)*
 - 2.5 Определение атрибутов информации*
 - 2.5.1 Определение и допустимые значения*
 - 2.5.2 Ограничения*
 - 2.6 Общие уведомления*
 - 2.7 Модель состояния системы*
 - 3 Определение интерфейса*
 - 3.1 Диаграмма класса представляющая интерфейс*
 - 3.2 Общие правила*
 - 3.b Интерфейс InterfaceName (supportQualifier)*
 - 3.b.a Операция OperationName (supportQualifier)*
 - 3.b.b Уведомление NotificationName (supportQualifier)*
 - 3.c Сценарий*
- В.3 Свойства ИОС, наследование и импорт*
 - В.3.1 Свойства*
 - В.3.2 Наследование*
 - В.3.3 Импорт*

Ниже приведены руководящие указания в отношении спецификации результатов стадии анализа.

Шаблон анализа базируется на информационном обслуживании 3GPP [b-3GPP TS 32.151] и доработан для соответствия дополнительным требованиям согласно методике (например, отслеживания).

Для спецификации интерфейса управления должны использоваться оба подпункта 2.2 и 2.3 шаблона "Анализ", предусмотренного в пункте В.2. Для информационной модели (например, модели сетевых ресурсов) необходимо применять только подпункт 2.2.

В шаблоне анализа используется Тип информации (Information Type) как одна из характеристик для описания атрибутов класса информационных объектов и параметров операции/уведомления. Могут быть использованы разрешенные Типы информации, а их семантика определяется в Приложении Е.

Вариант использования данного шаблона приведен в Дополнении II.

Конструкции "Анализ | Информационное обслуживание" и наборы "Проектирование | Решение" используются для обозначения аналогичных спецификаций, разработанных МСЭ-Т и 3GPP, которые отличаются названиями.

В.1 Условные обозначения

В.1.1 Обязательные, необязательные и условные квалификаторы

В данном подпункте определяется ряд терминов, которые используются для определения связей между компонентами Анализ | Информационное обслуживание, Проектирование | Наборы решений и их влияния на реализацию интерфейса. Квалификаторы, определенные в данном пункте, используются исключительно для обозначения поведения агента. Считается, что для спецификации интерфейсов управления этого достаточно.

Спецификации анализа | спецификации информационного обслуживания определяют атрибуты ИОС, интерфейсы, операции, уведомления, параметры операций и параметры уведомлений. Им могут присваиваться следующие квалификаторы поддержки/чтения/записи: М, О, СМ, СО, С.

Определение квалификатора М (Обязательно):

- Применяется для объектов, которым должна предоставляться поддержка.

Определение квалификатора О (Необязательно):

- Применяется для объектов, которым может предоставляться, а может не предоставляться поддержка.

Определение квалификатора СМ (Условно Обязательно):

- Применяется для объектов, которые являются обязательными при выполнении определенных условий, а именно:
 - Все объекты, сопровождающиеся квалификатором поддержки СМ, должны иметь соответствующие ограничения, которые определяются в Рекомендации | Спецификации информационного обслуживания. Если соответствующее ограничение выполняется, объекты должны поддерживаться.

Определение квалификатора СО (Условно Необязательно):

- Применяется для объектов, которые являются необязательными при выполнении определенных условий, а именно:
 - Все объекты, сопровождающиеся квалификатором поддержки СО, должны иметь соответствующие ограничения, которые определяются в Рекомендации | Спецификации информационного обслуживания. Если соответствующее ограничение выполняется, объекты могут поддерживаться.

Определение квалификатора С (Условно-SS):

- Применяется для объектов, которые используются в некоторых, но не во всех компонентах Проектирования | Наборах решений (SS).

Спецификации Проектирования | SS определяют эквиваленты SS для атрибутов ИОС, операций, уведомлений, параметров операций и параметров уведомлений. Таким эквивалентам SS могут присваиваться следующие квалификаторы поддержки/чтения/записи: М, О, СМ и СО.

Соответствия между квалификаторами конструкций, определяющихся на основании Анализа | IS, и квалификаторами соответствующих конструкций SS, определяется следующим образом:

- Для квалификаторов М, О, СМ и СО должно быть установлено соответствие между каждым объектом, определяемым IS (операции и уведомления, входные и выходные параметры операций, входные параметры уведомлений, информационные связи и атрибут информации) и его эквивалентом(ами) во всех наборах решений. Установленный(ые) эквивалент(ы) должен(ы) иметь такой же квалификатор, как и квалификатор, определяемый IS.

- Что касается квалификатора С, должно быть установлено соответствие каждого объекта, определяемого IS, и его эквивалента(ов), по меньшей мере, из одного набора решений. Установленный(ые) эквивалент(ы) может/могут иметь квалификатор поддержки М или О.

В таблице В.1 определена семантика квалификаторов эквивалентов с точки зрения поддержки со стороны агента.

**Таблица В.1 – Семантика квалификаторов, применяемых в Проектирование |
Наборах решений**

Сопоставленный эквивалент SS	Обязательно	Необязательно	Условно обязательно (СМ)	Условно необязательно (СО)
Сопоставленный эквивалент уведомления	Агент должен создать уведомление.	Агент может создать или не создавать уведомление.	Агент должен создать такое уведомление при условии выполнения ограничения для данного объекта.	Агент может создать или не создавать уведомление на свой выбор. Если агент решит создать уведомление, должно быть выполнено ограничение для данного уведомления.
Сопоставленный эквивалент операции	Агент должен поддержать операцию.	Агент может поддержать или не поддержать операцию. Если агент не поддержит операцию, он должен отклонить инициирование операции по причине того, что агент не поддерживает такую операцию. Отклонение, вместе с указанием причины, должно быть направлено менеджеру.	Агент должен поддержать такую операцию при условии выполнения ограничения для данного объекта.	Агент может поддержать такую операцию при условии выполнения ограничения для данного объекта.
Входной параметр сопоставленного эквивалента операции	Агент должен принять значение, от которого будет зависеть его поведение.	Агент может поддержать или не поддержать такой входной параметр. Если агент не поддерживает такой входной параметр и если он несет в себе смысл (т. е. не несет неинформативной семантики), агент отклоняет инициирование по причине (того, что он не поддерживает параметр). Отклонение вместе с указанием причины должно быть направлено менеджеру.	Агент должен принять значение, от которого будет зависеть его поведение, при условии выполнения ограничения для данного объекта.	Агент может принять значение, от которого будет зависеть его поведение, при условии выполнения ограничения для данного объекта.
Входной параметр сопоставленного эквивалента уведомления AND выходной параметр сопоставленного эквивалента операции	Агент должен предоставить этот параметр.	Агент может предоставить этот параметр.	Агент должен предоставить такой параметр при условии выполнения ограничения для данного объекта.	Агент может предоставить такой параметр при условии выполнения ограничения для данного объекта.
Сопоставленный эквивалент атрибута ИОС	Агент должен поддержать атрибут.	Агент может поддержать атрибут.	Агент должен поддержать такой атрибут при условии выполнения ограничения для данного объекта.	Агент может поддержать такой атрибут при условии выполнения ограничения для данного объекта.

В.2 Шаблон для анализа

1 Концепции и базовая информация

В данном пункте содержится введение в анализ определения интерфейсов управления.

1.a SubClauseTitle

SubClauseTitle представляет собой название подпункта.

"a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового подпункта.

Использование подпунктов является необязательным.

2 Классы информационных объектов

Данный пункт используется для всех спецификаций (как для определения интерфейсов управления, так и для определения исключительно информационной модели).

2.1 Импортированные информационные объекты и локальные метки

В данном подпункте определяется перечень информационных объектов (например, класс информационных объектов, интерфейс, информационные связи, атрибут информации), которые были определены в других спецификациях и импортированы в текущий документ. Все импортированные объекты должны восприниматься так, как если бы они были определены в текущей спецификации. Одна из целей импортирования объектов – обеспечение наследования. Каждый элемент в данном перечне представляет собой пару (ссылка на метку, локальная метка). Ссылка на метку содержит название спецификации, где она определяется, тип информационного объекта и его название. Локальные метки импортированных информационных объектов могут в дальнейшем использоваться в тексте спецификации вместо ссылок на метки.

Эта информация предоставляется в таблице.

Ссылка на метку	Локальная метка

Импортированные элементы должны происходить из определений, нейтральных в отношении протокола и основанных на данной методике, однако при необходимости могут быть импортированы элементы из других спецификаций, с целью миграции спецификаций, ориентированных на тот или иной протокол, во времени.

Руководящие указания в отношении импорта объектов, а также свойств ИОС и наследования, приводятся в Приложении F.

2.2 Диаграмма класса

2.2.1 Атрибуты и связи

В этом первом наборе диаграмм представлены все классы информационных объектов, определенные в рамках информационного обслуживания, со всеми связями и всеми атрибутами, включая связи с импортированными классами информационных объектов (при наличии). В таких диаграммах должно содержаться число элементов классов информационных объектов (для ассоциаций, а также отношений включения), а также могут содержаться названия ассоциаций и названия ролей. Это должны быть диаграммы класса, соответствующие UML (см. также Приложение C).

Характеристики (связи) импортированных классов информационных объектов не должны повторяться в диаграмме. Классы информационных объектов должны определяться с использованием стереотипа <<InformationObjectClass>>.

2.2.2 Наследование

Этот второй набор диаграмм отражает иерархию наследования всех классов информационных объектов, определенных в рамках информационного обслуживания. Эти диаграммы не обязательно должны содержать полную иерархию наследования, однако должны, по меньшей мере, содержать исходные классы информационных объектов для всех классов информационных объектов, определенных в текущем документе. По умолчанию предполагается наследование между тем или иным классом информационных объектов и классом информационных объектов "top". Это должны быть диаграммы класса, соответствующие UML.

Характеристики (атрибуты, связи) импортированных классов информационных объектов не должны повторяться в диаграмме. Классы информационных объектов должны определяться с использованием стереотипа <<InformationObjectClass>>.

ПРИМЕЧАНИЕ 1. – Некоторые отношения наследования, представленные в подпункте 2.2.2, могут повторяться в подпункте 2.2.1 для улучшения удобочитаемости.

ПРИМЕЧАНИЕ 2. – Наследование интерфейса показано в подпункте 3.1, а не в данном подпункте.

2.3 Определение классов информационных объектов

Каждый класс информационных объектов определяется в соответствии с указанной ниже структурой. Унаследованные объекты (атрибуты и т. д.) не требуют отображения, поскольку они определены в родительском(их) классе(ах) информационных объектов, а потому применяются ко всем подпунктам.

2.3.a InformationObjectClassName

InformationObjectClassName представляет собой название класса информационных объектов.

"a" представляет собой число, начиная с 1, увеличиваясь на 1 для каждого нового определения класса информационных объектов.

2.3.a.1 Определение

Подпункт <Definition> должен быть написан на естественном языке. Подпункт <Definition> относится к самому классу информационных объектов. В определении не приводятся характеристики, относящиеся к связям, которые могут иметься между классом объектов и другими классами объектов. Информацию такого рода читатель может получить, обратившись к определению связей. В данном подпункте указывается информация, относящаяся к наследованию.

Информация по возможностям отслеживания относительно одного или нескольких требований, которые поддерживаются данным классом информационных объектов, также указывается в данном подпункте в такой форме:

Ссылка	Метки требований	Комментарий

2.3.a.2 Атрибуты

В подпункте <Attributes> приводится перечень атрибутов, которые представляют собой управляемые свойства класса объектов. Каждый элемент представляет собой кортеж (*attributeName*, *supportQualifier*, *readQualifier*, *writeQualifier*):

- Квалификатор *supportQualifier* указывает на то, является ли элемент Обязательным (M), Необязательным (O), Условно Обязательным (CM), Условно Необязательным (CO), SS-Условным (C) или же Не поддерживаемым (-). Допустимые значения: Обязательный, Необязательный, Условный и Не поддерживается (соответственно, "M", "O", "C" или "-").
- Квалификатор *readQualifier* указывает на то, должен ли атрибут считываться менеджером. Возможные значения: Обязательный (M), Необязательный (O), Условно Обязательный (CM), Условно Необязательный (CO), SS-Условный (C) или Не поддерживается (-). Допустимые значения: Обязательный (M), Необязательный (O) и Не поддерживается (-).
- Квалификатор *writeQualifier* указывает на то, должен ли атрибут быть доступным для записи менеджером. Семантика квалификатора *writeQualifier* аналогична семантике квалификатора *supportQualifier* для "M", "O" и "-". Допустимые значения: Обязательный (M), Необязательный (O) и Не поддерживается (-).

Между квалификаторами *supportQualifier*, *readQualifier* и *writeQualifier* существуют отношения зависимости. Квалификатор *supportQualifier* обозначает требования для поддержки атрибута. Для любого заданного атрибута, независимо от значения квалификатора *supportQualifier*, по крайней мере, один из квалификаторов *readQualifier* и *writeQualifier* должен иметь значение "M". Если квалификатор *supportQualifier* имеет значение "O", это означает, что атрибут является необязательным; вместе с тем квалификаторы *Read* и *Write* указывают на то, каким образом должен быть поддержан атрибут в случае его поддержки.

Закрытые атрибуты и внутренние атрибуты агента по своему определению не доступны для записи Менеджером IRP. Следовательно, значение *writeQualifier* у таких атрибутов всегда "-".

Значения обоих квалификаторов *readQualifier* и *writeQualifier* для поддерживаемого атрибута не могут быть "-".

Использование значения "-" для квалификатора *supportQualifier* зарезервировано для отображения поддержки атрибутов, определяемых классом информационных объектов "Archetype" (см. подпункт С.3.5). Атрибуты, у которых значение квалификатора *supportQualifier* "-", не выполняются классом информационных объектов, который реализует поднабор атрибутов, определенных "Archetype". В данном случае квалификаторы *readQualifier* и *writeQualifier* не имеют значения. Однако, не поддерживаемый атрибут не доступен ни для чтения, ни для записи. Поэтому значения квалификаторов *readQualifier* и *writeQualifier* для неподдерживаемых атрибутов должны быть "-".

Для всех классов информационных объектов, которые используют один или несколько атрибутов из "Archetype", необходимо использовать отдельную таблицу для указания поддерживаемых атрибутов. Если ни один атрибут "Archetype" не поддерживается, такая таблица не используется. Например, если тот или иной класс информационных объектов имеет определенные атрибуты (т. е. атрибуты, которые не определяются "Archetype"), а также включает в себя атрибуты из двух объектов "Archetype", то все атрибуты указанного класса информационных объектов разбиваются на три отдельные таблицы.

Такая информация приводится в виде таблицы.

Название атрибута	Квалификатор Support	Квалификатор Read	Квалификатор Write	Идентификаторы требований

2.3.a.3 Ограничения атрибутов

В подпункте <Attribute constraints> представлены ограничения для атрибутов, которые всегда выполняются. Такие свойства сохраняются на протяжении времени существования атрибутов и, в частности, нет необходимости повторять их в предварительных и заключительных условиях операций и уведомлений.

ПРИМЕЧАНИЕ. – Данный подпункт не используется в случае, если нет необходимости указывать какие-либо ограничения атрибутов.

2.3.a.4 Связи

В подпункте <Relationship> представлен перечень связей, в которых принимает участие тот или иной класс. Каждый элемент представляет собой relationshipName.

Связи приводятся в таблице, как указано ниже:

Связи	Идентификаторы требований

При этом каждое имя связи должно представлять собой ссылку (и желательно также гиперссылку) на соответствующий подпункт пункта 2 (Классы информационных объектов).

ПРИМЕЧАНИЕ. – Использование данного подпункта является необязательным. Такой подпункт можно пропустить, поскольку все связи представлены в диаграмме классов в подпункте 2.2.1.

2.3.a.5 Диаграмма состояний

В подпункте <State diagram> приводятся диаграммы состояний. Диаграмма состояний класса информационных объектов определяет допустимые состояния такого класса информационных объектов и переход между такими состояниями. Состояние выражается в виде отдельных значений атрибутов, комбинации таких значений или участия в связях класса информационных объектов. Диаграмма состояний должна быть построена в соответствии с UML.

ПРИМЕЧАНИЕ. – Данный подпункт не используется в случае, если нет необходимости приводить диаграммы состояний.

2.3.a.6 Уведомления

В подпункте <Notifications> представлены следующие положения для того или иного класса информационных объектов:

- необязательно – ссылка на общие уведомления, определенные в подпункте 2.6 как действительные для данного класса информационных объектов;
- необязательно – перечень уведомлений, которые необходимо исключить из перечня общих уведомлений (определенного в подпункте 2.6) для данного класса информационных объектов (обратите внимание на то, что уведомления, унаследованные от родительского(их) класса(ов) информационных объектов, не могут исключаться;

и

- необязательно – перечень уведомлений, применимых к данному классу информационных объектов, которые могут быть, а могут не быть определены в числе общих уведомлений в подпункте 2.6.

Уведомления, определенные в данном подпункте, представляют собой уведомления, которые могут передаваться в рамках интерфейса управления, при этом параметры заголовка уведомления "object class" и "object instance" (см. Примечание 2) для таких уведомлений представляют экземпляр класса информационных объектов, определенный путем инкапсуляции подпункта (т. е. подпункта 2.3.a).

Уведомления, определенные в данном подпункте, могут создаваться объектом(ами) реализации, идентификатор которых сопоставлен в ходе реализации с идентификатором экземпляра объекта, использованного в интерфейсе управления. Таким образом, наличие уведомлений в данном подпункте (т. е. в подпункте 2.3.a.6) не подразумевает и не означает, что такие уведомления были созданы экземпляром ИОС, определенным путем инкапсуляции подпункта (т. е. подпункта 2.3.a).

Информация, связанная с указанным выше вариантом с) приводится в таблице. Пример такой таблицы приводится ниже:

Название	Квалификатор	Идентификаторы требований	Примечания

ПРИМЕЧАНИЕ 1. – Данный подпункт и таблица являются необязательными.

ПРИМЕЧАНИЕ 2. – Заголовок уведомления определяется в Информационной службе уведомлений IRP [b-3GPP TS 32.302].

ПРИМЕЧАНИЕ 3. – Квалификатор уведомления, представленный в Таблице уведомлений, указывает на то, может ли содержаться в таком уведомлении номер DN экземпляра. Квалификатор уведомления, указанный в спецификации управления, указывает на уровень поддержки в отношении создания субъектного уведомления. Менеджер может получить уведомление notification-XYZ, которое содержит DN экземпляра класса ABC если и только если:

- 1) Уведомление notification-XYZ определено в Таблице уведомлений класса class-ABC и
- 2) Реализация экземпляра класса class-ABC поддерживает такое уведомление notification-XYZ и
- 3) Уведомление notification-XYZ определено в интерфейсе управления и
- 4) Реализация интерфейса управления поддерживает такое уведомление notification-XYZ.

2.4 Определение информационных связей

В первую очередь, в данном подпункте приводятся все связи, которые поддерживаются данной Рекомендацией | Спецификацией (см. таблицу, ниже). Квалификатор Support определяется для атрибутов в пункте В.1.

Связи	Квалификатор Support	Идентификаторы требований

Все информационные связи определяются в соответствии с указанной ниже структурой.

Унаследованные связи не требуют отображения, поскольку они определены в родительском(их) ИОС, а потому применяются ко всем подпунктам.

2.4.a InformationRelationshipName (supportQualifier)

InformationRelationshipName представляет собой название информационной связи, за которым следует квалификатор (см. пункт В.1).

"a" представляет собой число, начиная с 1, увеличиваясь на 1 для каждой новой информационной связи.

2.4.a.1 Определение

Подпункт <Definition> должен быть написан на естественном языке.

2.4.a.2 Роли

В подпункте <Roles> определяются роли, которые играют классы объектов в рамках связей. Каждый элемент представлен в виде пары (roleName, roleDefinition).

Такая информация приводится в виде таблицы.

Название	Определение

2.4.a.3 Ограничения

В подпункте <Constraints> содержится перечень свойств с указанием семантических инвариантов, которые необходимо сохранить в рамках связей. Каждый элемент представлен в виде пары (propertyName, propertyDefinition). Такие свойства сохраняются на протяжении времени существования связей, и нет необходимости повторять их в предварительных и заключительных условиях операций или уведомлений.

Такая информация приводится в виде таблицы.

Связи	Квалификатор Support	Идентификаторы требований

2.5 Определение атрибутов информации

Все атрибуты информации определяются в соответствии с указанной ниже структурой.

Унаследованные атрибуты не требуют отображения, поскольку они определены в родительском(их) ИОС, а потому применяются ко всем подпунктам.

2.5.1 Определение и допустимые значения

В данном подпункте содержатся, для каждого определяемого атрибута, *Attribute Name* (Имя атрибута), его *Definition* (Определение), написанное на естественном языке, *Information Type* (Тип информации) (см. Приложение Е) и (необязательно) перечень *Legal Value* (Допустимые значения), которые поддерживаются атрибутом.

В случае, когда *Legal Value* (Допустимые значения) являются перечислимыми, каждый элемент представляется в виде пары (*Legal Value Name*, *Legal Value Semantics*), за исключением случаев, когда *Legal Value Semantics* применяется к нескольким значениям – в таком случае *Semantics* указывается только один раз. Когда *Legal Value* (Допустимые значения) не являются перечислимыми, перечень *Legal Value* (Допустимых значений) определяется при помощи одного определения.

Такая информация приводится в виде таблицы.

Имя атрибута	Определение	Тип информации/ Допустимые значения

2.5.2 Ограничения

В подпункте <*Constraints*> указывается наличие каких-либо ограничений применительно к атрибутам.

Каждое ограничение определяется при помощи кортежа (*propertyName*, затронутые атрибуты, *propertyDefinition*). *PropertyDefinitions* выражаются посредством естественного языка.

Такая информация приводится в виде таблицы.

Название	Затронутый(ые) атрибут(ы)	Определение

2.6 Общие уведомления

В подпункте <*Common Notifications*> приводится перечень уведомлений, на которые может ссылаться любой класс информационных объектов, определенный в данной спецификации интерфейса управления. Такие уведомления применяются только к классам информационных объектов, которые ссылаются на данный подпункт в подпункте 2.3.а.б.

Такая информация приводится в виде таблицы.

Название	Квалификатор	Примечания

ПРИМЕЧАНИЕ. – Данный подпункт может отсутствовать, если общие уведомления не используются.

2.7 Модель состояния системы

Некоторые конфигурации информации являются настолько особенными или сложными, что для их объяснения требуется использование диаграммы состояний. В данном подпункте диаграмма состояний определяет допустимые состояния системы и переход между такими состояниями. Состояние выражается в виде комбинации ограничений значений атрибутов или участия в отношениях одного или нескольких классов информационно-объектов.

3 Определение интерфейса

Данный пункт должен использоваться во всех спецификациях интерфейса управления и является необязательным для спецификаций, в которых описывается только информационная модель.

3.1 Диаграмма класса представляющая интерфейс

Каждый интерфейс определяется в диаграмме. Это должна быть диаграмма классов, соответствующая UML (см. также Приложение С).

Интерфейсы определяются посредством стереотипа <<Interface>>. Каждый интерфейс содержит или набор операций и уведомлений, которые являются обязательными, или одну операцию или одно уведомление, которые являются необязательными. Для определения необязательных и обязательных интерфейсов применяются стереотипы (см. Приложение С). На диаграмме классов каждая операция и уведомление в рамках интерфейса должны быть квалифицированы как "public" путем добавления символа "+" перед каждой операцией и уведомлением.

ПРИМЕЧАНИЕ. – В данном подпункте может отображаться наследование интерфейса.

3.2 Общие правила

Приведенные ниже правила применяются ко всем спецификациям. Их необходимо копировать и включать в спецификации.

Правило 1: Каждая операция, по меньшей мере, с одним входным параметром поддерживает предварительное условие `valid_input_parameter`, которое указывает на то, что все входные параметры должны быть действительными применительно к используемому типу информации. Кроме того, каждая такая операция поддерживает исключение `operation_failed_invalid_input_parameter`, которое применяется, когда указанный в предварительных условиях `valid_input_parameter` является ложным. Данное исключение характеризуется одинаковым состоянием на входе и на выходе.

Правило 2: Каждая операция, по меньшей мере, с одним необязательным входным параметром, поддерживает набор предварительных условий `supported_optional_input_parameter_xxx`, где "xxx" представляет собой название необязательного входного параметра, а предварительное условие состоит в том, что операция поддерживает названный необязательный входной параметр. Кроме того, каждая такая операция поддерживает исключение `operation_failed_unsupported_optional_input_parameter_xxx`, которое применяется, когда (a) предварительное условие `supported_optional_input_parameter_xxx` является ложным и (b) названный необязательный входной параметр несет в себе информацию. Данное исключение характеризуется одинаковым состоянием на входе и на выходе.

Правило 3: Каждая операция должна поддерживать общее исключение `operation_failed_internal_problem`, которое выполняется в случае возникновения внутренней проблемы и невозможности завершения операции. Данное исключение характеризуется одинаковым состоянием на входе и на выходе.

ПРИМЕЧАНИЕ. – Требования обеспечения безопасности и соответствующие общие правила подлежат дальнейшему изучению.

3.b Название интерфейса `InterfaceName (supportQualifier)`

`InterfaceName` представляет собой название интерфейса, за которым следует квалификатор (см. пункт В.1). "b" представляет собой число, начиная с 3, увеличиваясь на 1 для каждого нового определения интерфейсов. Каждый интерфейс определяется названием и последовательностью операций или уведомлений, как определено ниже.

Все операции определяются в соответствии с указанной ниже структурой.

ПРИМЕЧАНИЕ. – Вопросы группировки операций/разбивки содержимого интерфейса и наименования интерфейсов подлежат дальнейшему изучению.

3.b.a Название операции `OperationName (supportQualifier)`

`OperationName` представляет собой название операции, за которым следует квалификатор (см. пункт В.1). "a" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового определения операции.

3.b.a.1 Определение

Подпункт <Definition> (определение) должен быть написан на естественном языке.

Информация по возможностям отслеживания относительно одного или нескольких требований, которые поддерживаются данной операцией, также указывается в данном подпункте в такой форме:

Ссылка	Метка требования	Комментарий

3.б.а.2 Входные параметры

Перечень входных параметров операции. Каждый элемент представляет собой кортеж *Parameter Name* (имя параметра), *Support Qualifier* (квалификатор *Support*), *Information Type* (тип информации) (см. Приложение E и примечания в пункте E.2), (необязательно) перечень *Legal Values* (допустимых значений), поддерживаемых параметром, *Comments* (комментарии)). Допустимые значения для квалификатора *Support* приведены в пункте B.1.

Такая информация приводится в виде таблицы.

Имя параметра	Квалификатор <i>Support</i>	Тип информации/ Допустимые значения	Комментарии

ПРИМЕЧАНИЕ. – В поле "*Information Type*" уточняется значение "*Parameter Name*". В случае, когда *Legal Values* являются перечислимыми, каждый элемент представляется в виде пары (*Legal Value Name*, *Legal Value Semantics*), за исключением случаев, когда *Legal Value Semantics* применяется к нескольким значениям – в таком случае определение указывается только один раз. Когда *Legal Values* не являются перечислимыми, перечень *Legal Values* определяется при помощи одного определения.

3.б.а.3 Выходные параметры

Перечень выходных параметров операции. Каждый элемент представляет собой кортеж (*Parameter Name* (имя параметра), *Support Qualifier* (квалификатор *Support*), *Matching Information/Information Type* (соответствующая информация / тип информации) (см. Приложение E и примечания в пункте E.2), (необязательно) перечень *Legal Values* (допустимых значений), поддерживаемых параметром, *Comment* (комментарии)). Допустимые значения для квалификатора *Support* приведены в пункте B.1.

Такая информация приводится в виде таблицы.

Имя параметра	Квалификатор <i>Support</i>	Соответствующая информация/ Тип информации/ Допустимые значения	Комментарии

ПРИМЕЧАНИЕ. – В поле "*Тип информации*" уточняется значение "*Имя параметра*". В случае, когда *Допустимые значения* являются перечислимыми, каждый элемент представляется в виде пары (*Legal Value Name*, *Legal Value Semantics*), за исключением случаев, когда *Legal Value Semantics* применяется к нескольким значениям – в таком случае определение указывается только один раз. Когда *Legal Values* (*Допустимые значения*) не являются перечислимыми, перечень *Legal Values* (*Допустимых значений*) определяется при помощи одного определения.

В данной таблице должен также указываться специальный "*status*" параметра для обозначения статуса завершения операции (выполнено, частично выполнено, причина сбоя и т. д.).

3.б.а.4 Предварительное условие

Предварительное условие представляет собой набор утверждений, которые соединены между собой логическими операторами *AND*, *OR* и *NOT*. Для выполнения операции предварительное условие должно быть выполнено.

Каждое утверждение представлено в виде пары (*propertyName*, *propertyDefinition*). Все утверждения, составляющие предварительное условие, вносятся в таблицу.

Название утверждения	Определение

3.в.а.5 Заключительное условие

Заключительное условие представляет собой набор утверждений, которые соединены между собой логическими операторами AND, OR и NOT. Заключительное условие должно выполняться после завершения операции. Если в заключительном условии ничего не говорится о том или ином информационном объекте, предполагается, что такой информационный объект не изменился по сравнению с его описанием в предварительном условии.

Каждое утверждение представлено в виде пары (*propertyName*, *propertyDefinition*). Все утверждения, составляющие заключительное условие, вносятся в таблицу.

Название утверждения	Определение

3.в.а.6 Исключения

Перечень исключений, которые могут возникать в ходе операции. Каждый элемент представляет собой кортеж (*exceptionName*, *condition*, *ReturnedInformation*, *exitState*).

3.в.а.6.с exceptionName

ExceptionName представляет собой название исключения.

"с" представляет собой число, начиная с 1, увеличиваясь на 1 для каждого нового определения исключения.

Такая информация приводится в виде таблицы.

Название исключения	Определение	
	Условие	
	Возвращаемая информация	
	Состояние на выходе	
	Условие	
	Возвращаемая информация	
	Состояние на выходе	

3.в.а.7 Ограничения

В подпункте <Constraints> содержатся ограничения для операции или параметров операции.

ПРИМЕЧАНИЕ. – Данный подпункт не используется в случае, если нет необходимости указывать какие-либо ограничения.

3.в.б Название уведомления NotificationName (supportQualifier)

NotificationName представляет собой название уведомления, за которым следует квалификатор (см. пункт В.1).

"b" представляет собой число, начиная с 1 и увеличиваясь на 1 для каждого нового определения уведомления.

3.в.б.1 Определение

Подпункт <Definition> должен быть написан на естественном языке.

Информация по возможностям отслеживания относительно одного или нескольких требований, которые поддерживаются данным уведомлением, также указывается в данном подпункте в такой форме:

Ссылка	Метка требования	Комментарии

3.в.б.2 Входные параметры

Перечень входных параметров уведомления. Каждый элемент представляет собой кортеж *Parameter Name* (имя параметра), *Qualifiers* (квалификаторы), *Matching Information/Information Type* (соответствующая информация /тип информации) (см. Приложение Е и примечания в пункте Е.2), (необязательно) перечень *Legal Values* (допустимых значений), поддерживаемых параметром, *Comments* (комментарии)).

В столбце "Квалификаторы" приводятся два квалификатора, *Support Qualifier* (см пункт B.1) и *Filtering Qualifier*, которые отделяются друг от друга запятой. *Filtering Qualifier* указывает на то, могут ли параметры уведомления быть отфильтрованы. Значения могут быть Да (Y) и Нет (N).

Такая информация приводится в виде таблицы.

Название параметра	Квалификаторы	Соответствующая информация/ Тип информации/ Допустимые значения	Комментарии

ПРИМЕЧАНИЕ. – В поле "Тип информации" уточняется значение "Имя параметра". В случае, когда Допустимые значения являются перечислимыми, каждый элемент представляется в виде пары (*Legal Value Name*, *Legal Value Semantics*), за исключением случаев, когда *Legal Value Semantics* применяется к нескольким значениям – в таком случае определение указывается только один раз. Когда Допустимые значения не являются перечислимыми, перечень Допустимых значений определяется при помощи одного определения.

3.b.b.3 Иницирующее событие

Событием, иницирующим отправку уведомления, является переход из состояния информации, определенного в подпункте "из состояния" в состояние информации, определенное в подпункте "в состоянии".

3.b.b.3.1 Из состояния

Данный подпункт представляет собой набор утверждений, которые соединены между собой логическими операторами AND, OR и NOT.

Каждое утверждение представлено в виде пары (*propertyName*, *propertyDefinition*). Все утверждения, включаемые в такое состояние, вносятся в таблицу.

Название утверждения	Определение

3.b.b.3.2 В состоянии

Данный подпункт представляет собой набор утверждений, которые соединены между собой логическими операторами AND, OR и NOT. Если в данном подпункте ничего не говорится о том или ином информационном объекте, предполагается, что такой информационный объект не изменился по сравнению с его описанием в предыдущем подпункте.

Каждое утверждение представлено в виде пары (*propertyName*, *propertyDefinition*). Все утверждения, включаемые в такое состояние, вносятся в таблицу.

Название утверждения	Определение

3.b.b.4 Ограничения

В подпункте <Constraints> содержатся ограничения для уведомления или параметров уведомления.

ПРИМЕЧАНИЕ. – Данный подпункт не используется в случае, если нет необходимости указывать какие-либо ограничения.

3.c Сценарий

В данном подпункте приводятся одна или несколько диаграмм последовательности, в каждой из которых описывается возможный сценарий. Это должны быть диаграммы последовательности, соответствующие UML. Использование данного подпункта является необязательным.

В.3 Свойства ИОС и наследование

В.3.1 Свойство

Свойства класса информационных объектов (за исключением Support ИОС) определяются с учетом следующих положений:

a) Атрибут(ы) класса информационных объектов, включая семантику и синтаксис, диапазон допустимых значений и дополнительные квалификаторы. Атрибуты ИОС не ограничиваются Configuration Management (управлением конфигурацией), но также включают атрибуты, относящиеся, например, к: 1) Performance Management (управлению производительностью) (т. е. типам измерений); 2) Trace management (управлению трассировкой); и 3) Accounting Management (управлению счетом).

b) Поведение, не зависящее от атрибутов, связанное с ИОС (см. Примечание 1).

ПРИМЕЧАНИЕ 1. – Например, Связь между А и В является необязательной. Это обязательно в случае, если экземпляр А принадлежит к одному экземпляру ManagedElement, а экземпляр В принадлежит к другому экземпляру ManagedElement. Поведение этой Связи не зависит от атрибутов. Ожидается, что в таком поведении, как и в других случаях, будет прослеживаться наследование.

c) Связь(и) класса информационных объектов с другим(ими) классом(ами) информационных объектов.

d) Тип(ы) уведомлений ИОС и их квалификаторы.

e) Связи класса информационных объектов с родительскими объектами (см. Примечание 2). Существуют три взаимоисключающих случая:

1) Класс информационных объектов является абстрактным, и для него не были еще назначены родительские объекты.

2) Класс информационных объектов является абстрактным, все возможные родительские объекты были назначены, и было определено, могут ли подклассы классов информационных объектов назначаться в качестве корневого класса информационных объектов.

3) Класс информационных объектов не является абстрактным, все возможные родительские объекты были назначены, и было определено, могут ли подклассы классов информационных объектов назначаться в качестве корневого класса информационных объектов.

Экземпляр класса информационных объектов или является корневым классом информационных объектов, или имеет только один родительский объект.

ПРИМЕЧАНИЕ 2. – Связи между родительскими и дочерними объектами в данном подпункте представляют собой отношения включения между родительским и дочерним объектом.

f) Связи класса информационных объектов с дочерними элементами. Существуют три взаимоисключающих случая:

1) Класс информационных объектов не должен иметь дочерних классов информационных объектов (отношения включения).

2) Класс информационных объектов может иметь дочерний(ие) класс(ы) информационных объектов. Может указываться максимальное число экземпляров для дочернего класса информационных объектов. Для класса информационных объектов может определяться, что зависящие от поставщика объекты не могут выступать в роли дочерних классов информационных объектов.

3) Класс информационных объектов может иметь только конкретный(ые) дочерний(ие) класс(ы) информационных объектов (или их подклассы). Может указываться максимальное число экземпляров для дочернего класса информационных объектов. Для класса информационных объектов может определяться, что зависящие от поставщика объекты не могут выступать в роли дочерних классов информационных объектов.

g) Могут ли создаваться экземпляры класса информационных объектов (т. е. является ли класс информационных объектов абстрактным).

h) Атрибут с целью присвоения имен.

В.3.2 Наследование

Когда класс информационных объектов (подкласс) наследует другой класс информационных объектов (суперкласс), это означает, что подкласс обладает всеми свойствами суперкласса.

Подкласс может изменять необязательные унаследованные квалификаторы поддержки на обязательные, но не наоборот. Подкласс может изменять условно необязательные унаследованные квалификаторы поддержки на условно обязательные, но не наоборот.

Класс информационных объектов может выступать в роли суперкласса для многих классов информационных объектов. В то же время, подкласс не может иметь более одного суперкласса.

Подкласс может:

- a) Добавлять (по сравнению с суперклассом) уникальные атрибуты, в том числе поведение, диапазоны допустимых значений и квалификаторы поддержки. Каждому дополнительному атрибуту должно быть присвоено свое уникальное имя (для всех добавленных и унаследованных атрибутов).
- b) Добавлять безатрибутное поведение на базе ИОС. Такое поведение не может противоречить унаследованному поведению суперкласса.
- c) Добавлять связь(и) с классом(ами) информационных объектов. Каждой дополнительной связи должно быть присвоено свое уникальное название (для всех добавленных и унаследованных связей).
- d) Добавлять дополнительные типы уведомлений и их квалификаторы.
- e) Назначать все возможные родительские элементы (и их подклассы), если суперкласс обладает свойством Property-e-1, так что класс информационных объектов будет обладать свойствами Property-e-2 или Property-e-3. Ограничивать возможные родительские элементы (и их подклассы) и/или ограничивать возможность для подкласса выступать в качестве корневого класса информационных объектов, если суперкласс обладает свойствами Property-e-2 или Property-e-3.
- f) Добавлять дочерние классы информационных объектов, если суперкласс обладает свойством Property-f-2, так что класс информационных объектов будет обладать свойством Property-f-3. Ограничивать допустимые дочерние классы информационных объектов (или их подклассы), если суперкласс обладает свойством Property-f-3.
- g) Указывать, могут ли создаваться экземпляры класса информационных объектов (т. е. является ли класс информационных объектов абстрактным).
- h) Ограничивать диапазон допустимых значений атрибута суперкласса, имеющего диапазон допустимых значений.

В.3.3 Импорт

Чтобы способствовать повторному использованию определений класса информационных объектов в спецификациях IRP, спецификацией IRP (которая называется субъектной IRP) используется механизм импорта с целью повторного использования определения класса информационных объектов, определенного в другой спецификации IRP. При импорте класса информационных объектов спецификацией субъектной IRP не могут изменяться свойства импортированного класса информационных объектов. Если необходимо внести изменения в импортированный класс информационных объектов, необходимо использовать наследование для определения собственного нового класса.

Приложение С

Набор символов UML MISM

(Данное приложение является неотъемлемой частью данной Рекомендации.)

Ниже приведены руководящие указания для спецификации результатов стадии анализа на базе набора символов унифицированного языка моделирования (UML) 3GPP [b-3GPP TS 32.152].

С.1 Введение

UML предлагает широкий выбор концепций, нотаций и элементов модели для моделирования распределенных систем. Нотации и элементы модели UML не обязательно используются с целью формирования анализа спецификаций. В данном дополнении приводится необходимый и достаточный набор нотаций и элементов модели UML, в том числе построенных при помощи применяемого в UML механизма расширения <<stereotype>> для использования в процессе проектирования спецификаций, нейтральных в отношении протокола. Такой набор нотаций и элементов модели называется набором символов UML.

В рекомендациях, соответствующих данной методике, должны применяться нотации и элементы модели UML из такого набора символов, а также при необходимости могут применяться другие нотации и элементы модели UML.

С.2 Базовые элементы модели

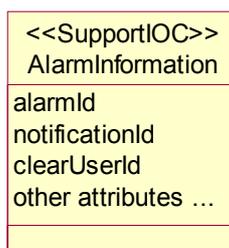
С.2.1 Общие положения

В UML определен ряд базовых элементов модели. В данном подпункте приводятся избранные элементы для использования в рамках набора символов. Семантика таких избранных элементов определяется в документе [OMG UML].

С.2.2 Атрибут

См. подпункт 3.25 [OMG UML].

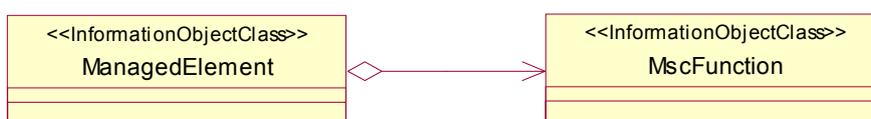
В данном примере показан ряд атрибутов, которые перечислены в виде строк в секции атрибута класса AlarmInformation.



С.2.3 Агрегирование

См. подпункт 3.43.2.5 [OMG UML].

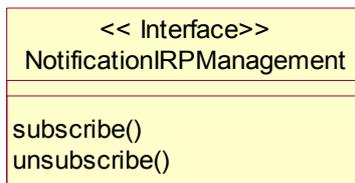
В данном примере показан пустой маркер-ромб в конце пути для обозначения агрегирования. Маркер-ромб присоединен к классу, который является агрегатным.



С.2.4 Операция

См. подпункт 3.26 [OMG UML].

В данном примере показаны две операции, отображаемые в виде строк в секции операций класса NotificationIRPManagement, запрос на выполнение которых может быть получен экземпляром NotificationIRPManagement. Каждая операция имеет название, например, "subscribe" и перечень аргументов (не показан на рисунке).

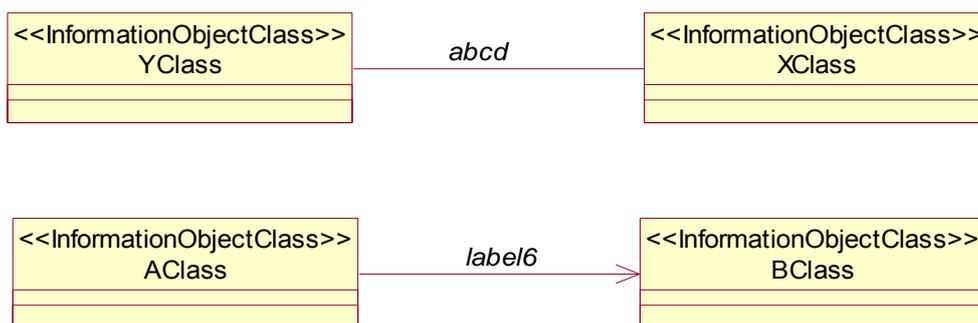


С.2.5 Ассоциация и название ассоциации

См. подпункт 3.41 [OMG UML].

В двух примерах ниже показана бинарная ассоциация между исключительно двумя элементами модели. Ассоциация может предусматривать возможность установления связи элемента модели с самим таким элементом. В первом примере показана двунаправленная ассоциация, в которой каждому элементу модели известно о другом элементе. Во втором примере показана однонаправленная ассоциация (изображенная в виде открытой стрелки со стороны целевого элемента модели), при которой только исходному элементу модели известно о целевом элементе модели, а не наоборот.

Ассоциации могут иметь названия, такие как "abcd" и "label6" в приведенных ниже примерах.



С.2.6 Реализация связи

См. подпункт 2.5.2.1 [OMG UML].

В данном примере показаны реализация связи между элементом модели AlarmIRPOperations_1 и другим элементом модели AlarmIRP. Последний (целевой элемент модели) реализует предшествующий элемент. Целевым элементом модели должен быть <<Interface>>.



С.2.7 Обобщение связи

См. подпункт 3.50 [OMG UML].

В данном примере показаны обобщения связей между более общим элементом (агентом) и более специфическим элементом (Agent_vendor_A), который полностью соответствует первому элементу, и при этом несет в себе дополнительную информацию.



С.2.8 Отношение зависимости

См. подпункт 3.51 [OMG UML].

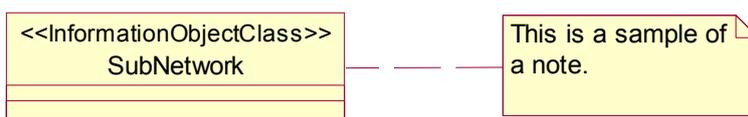
В данном примере показана семантическая связь между экземплярами BClass и экземплярами AClass. Такие отношения характеризуют ситуацию, при которой изменение целевого элемента зависимости требует изменения исходного элемента.



С.2.9 Примечание

См. подпункт 3.11 [OMG UML].

В данном примере показано примечание в виде прямоугольника с "загнутым" правым верхним углом. Примечание содержит произвольный текст. Оно отображается для конкретной диаграммы и может соединяться пунктирными линиями с нулевым или большим количеством элементов моделирования.



С.2.10 Кратность или количество элементов

См. подпункт 3.44 [OMG UML].

В данном примере показано изображение кратности в конце пути ассоциации. Такая кратность имеет значение "один ко многим". Экземпляры сети могут быть связаны с некоторым количеством экземпляров SubNetwork, при этом такое количество может быть равно нулю, единице или большему числу.

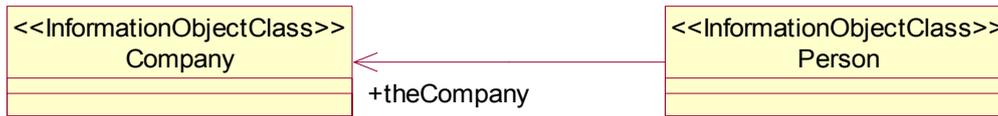
В предыдущих версиях [b-3GPP TS 32.152] кратность, равная нулю, могла означать, что классу информационных объектов присуща характеристика так называемого "переходного состояния". Например, она указывает на то, что экземпляр еще не создан, но находится в процессе создания. В данной версии методики нулевая кратность не будет использоваться для указания такой характеристики, поскольку данная характеристика присуща всем классам информационных объектов. Считается, что все определяемые классы информационных объектов имеют такие характеристики "переходного состояния".



С.2.11 Ролевое имя

См. подпункт 3.43.2.6 [OMG UML].

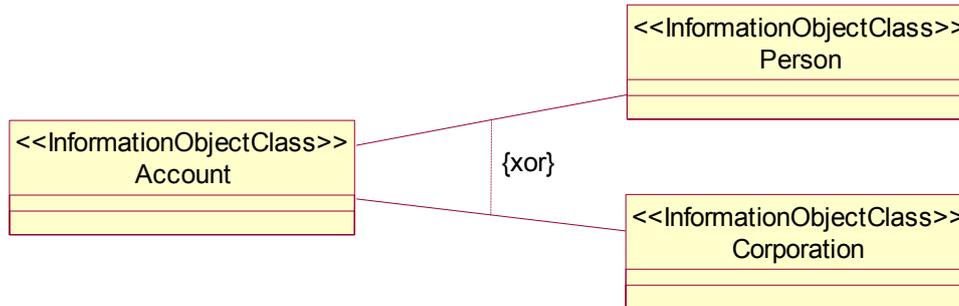
В данном примере показана ассоциация класса Person (например, экземпляр Джон) с классом Company (например, с номером DN "Company=XYZ"). Мы осуществляем навигацию в рамках ассоциации с использованием противоположного конца ассоциации, так что для Джона Person.theCompany будет использоваться номер DN, т. е. "Company=XYZ". Для ролевого имени следует использовать имя существительное.



С.2.12 Ограничение Хор

См. подпункты 2.5.2.3 и 3.42.5.1 [OMG UML].

В данном примере показана ассоциация класса Account (например, счета 0960) с классом Person (например, Джон Смит) или классом Corporation (например, ABC Inc).



С.3 Стереотипы

С.3.1 Общие положения

В данном подпункте приведены все допустимые стереотипы, которые могут использоваться в спецификациях интерфейса управления. Один стереотип <<Interface>> определяется в [OMG UML]. В данной Рекомендации такой стереотип приводится для удобства пользования и полноты. Другие стереотипы определяются в данной Рекомендации.

Таблица С.3-1 – Стереотипы объектов

Стереотип	Базовый класс	Затронутые элементы метамодели
Interface	Class [Класс]	
ProxyClass	Class [Класс]	
Notification	Class [Класс]	
Archetype	Classifier [Классификатор] (подпункт 2.5.2.10 [OMG UML])	
InformationObjectClass	Classifier [Классификатор]	
SupportIOC	Classifier [Классификатор]	
use	Association [Ассоциация]	
may use	Association [Ассоциация]	
may realize	Association [Ассоциация]	
names	Composition [Состав]	

С.3.2 <<Interface>>

Подпункт 2.5.2.25 [OMG UML]

"Интерфейс представляет собой именованный набор операций, которые характеризуют поведение элемента. В метамодели Интерфейс содержит набор Операций, в совокупности определяющих услугу, которая предлагается Классификатором, реализующим Интерфейс. Классификатор может предлагать несколько услуг, что означает, что он может реализовать несколько Интерфейсов, в то время как несколько Классификаторов могут реализовать один и тот же Интерфейс.

У интерфейсов [могут иметься или] могут не иметься Атрибуты, Ассоциации и Методы. Интерфейс может принимать участие в Ассоциациях, при условии, что такая Ассоциация не является видимой для Интерфейса; таким образом, Классификатор (отличный от Интерфейса) может иметь Ассоциацию с Интерфейсом, которая подлежит управлению со стороны Классификатора, но не со стороны Интерфейса".

См. подпункт 2.5.4.6 [OMG UML]

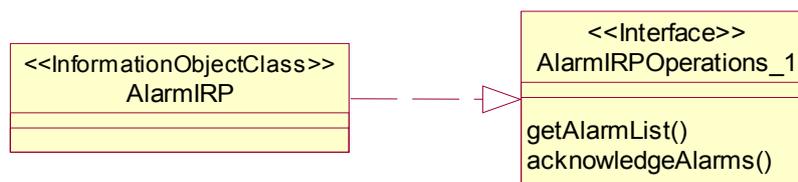
"Цель интерфейса заключается в том, чтобы собрать набор операций, составляющих связанные услуги, предлагаемые классификаторами. Интерфейсы предоставляют возможности разделения и характеризуют группы операций. Интерфейс представляет собой исключительно набор операций, которому присвоено имя. Непосредственно из интерфейса невозможно создавать объекты".

См. подпункт 2.5.4.6 [OMG UML]

"Несколько классификаторов могут осуществлять реализацию одного и того же интерфейса. Все они должны содержать, по меньшей мере, операции, соответствующие операциям, содержащимся в интерфейсе. Спецификация операции содержит подпись операции (т. е. название операции, типы параметров и тип возврата). Интерфейс не определяет внутреннюю структуру реализующего его классификатора. Например, интерфейсом не предусматривается алгоритм, который должен использоваться для выполнения операции. Вместе с тем операция может содержать спецификацию последствий [например, с предварительными и заключительными условиями] ее использования".

С.3.2.1 Пример

В данном примере показан класс AlarmIRPOperations_1 <<Interface>>, имеющий две операции. Входные и выходные параметры операций скрыты (т. е. не отображаются). Класс AlarmIRP имеет однонаправленные обязательные отношения реализации с элементом <<Interface>>.



Нотация <<Interface>>

С.3.3 <<ProxyClass>>

С.3.3.1 Общие положения

Данный элемент представляет собой номер класса <<InformationObjectClass>>. Он инкапсулирует атрибуты, ссылки, методы (или операции) и взаимосвязи, присутствующие в представленном классе <<InformationObjectClass>>.

Семантика <<ProxyClass>> предполагает, что поведение <<ProxyClass>> в полном объеме присутствует в <<InformationObjectClass>>. Поскольку данный класс является только представлением других классов, он не может определять собственное поведение, отличное от поведения, определенного для представляемого <<InformationObjectClass>>.

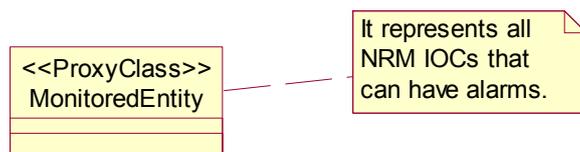
Тот или иной <<InformationObjectClass>> может быть представлен ни одним, одним или несколькими элементами <<ProxyClass>> или <<Archetype>>. Например, ManagedElement <<InformationObjectClass>> может иметь <<ProxyClass>> для MonitoredEntity и <<ProxyClass>> для ManagedEntity.

Атрибуты <<ProxyClass>> доступны для исходного объекта, у которого имеется ассоциация с <<ProxyClass>>.

С.3.3.2 Пример

В данном примере показан <<ProxyClass>> под названием MonitoredEntity. Он представляет все NRM <<InformationObjectClass>> (например, GgsnFunction <<InformationObjectClass>>), экземпляры которых проходят мониторинг на аварийное состояние.

Обратите внимание, что <<MonitoredEntity>> не определяет никаких атрибутов. Атрибуты уже определены всеми классами <<InformationObjectClass>>, которые представлены в <<MonitoredEntity>>.



Нотация <<ProxyClass>>

Больше примеров использования <<ProxyClass>> можно найти в Дополнении V.

С.3.4 <<Archetype>>

С.3.4.1 Общие положения

Данный элемент представляет ряд общих свойств класса (например, атрибуты, ссылки, операции и взаимодействия, характерные для представляемого класса <<InformationObjectClass>>).

Семантика <<Archetype>> предполагает, что все атрибуты, ссылки, операции и взаимодействия, которые инкапсулированы в <<Archetype>>, могут быть представлены или не представлены в классе <<InformationObjectClass>>. <<Archetype>> представляет собой прототип класса/замещающий класс, который является наиболее эффективным при использовании в моделях анализа, независимых от технологий, требующих дальнейшей спецификации и/или сопоставления в рамках более полной модели проектирования.

С.3.4.2 Пример

В данном примере показан <<Archetype>> под названием StateManagement. Также в нем показан Агент класса <<InformationObjectClass>>, который зависит от элемента StateManagement. Обратите внимание, что StateManagement определяет ряд атрибутов (которые не отображаются в диаграмме UML). Классы, зависящие от StateManagement, могут использовать, а могут не использовать все атрибуты StateManagement. Другими словами, в Агенте представлен, по меньшей мере, один из атрибутов StateManagement. Точный набор атрибутов StateManagement, используемых Агентом, указывается в спецификации Агента.



Нотация <<Archetype>>

С.3.5 <<InformationObjectClass>>

С.3.5.1 Общие положения

В данном подпункте представлен класс информационных объектов. Каждый класс <<InformationObjectClass>> представляет набор экземпляров со схожей структурой, поведением и связями.

Данный класс <<InformationObjectClass>> и другие классы информации, такие как <<Interface>>, сопоставляются с элементами моделей, определяемых технологией, например, GDMO Managed Object Class для технологии CMIP. Сопоставление конструкций языка моделирования, независимых от протокола, с конструкциями языка моделирования, определяемого технологией, описывается в соответствующих спецификациях, которые определяются протоколом.

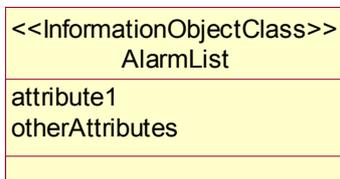
Сфера применения имени <<InformationObjectClass>> распространяется на Рекомендацию, в которой указан такой класс объектов, и такое имя должно быть уникальным в рамках всех имен классов <<InformationObjectClass>> в такой Рекомендации. Название Рекомендации рассматривается таким же образом, как и имя Пакета UML.

Класс объектов <<InformationObjectClass>> идентичен классу UML, за исключением того, что в него не включены/в нем не определены методы и операции.

Подпункт 3.22.1 [OMG UML]: "Класс представляет собой концепцию в рамках моделируемой системы. Классы обладают структурой данных, поведением и связями с другими элементами".

С.3.5.2 Пример

В данном примере показан класс объектов <<InformationObjectClass>> AlarmList.



Нотация <<InformationObjectClass>>

С.3.6 <<use>> и <<may use>>

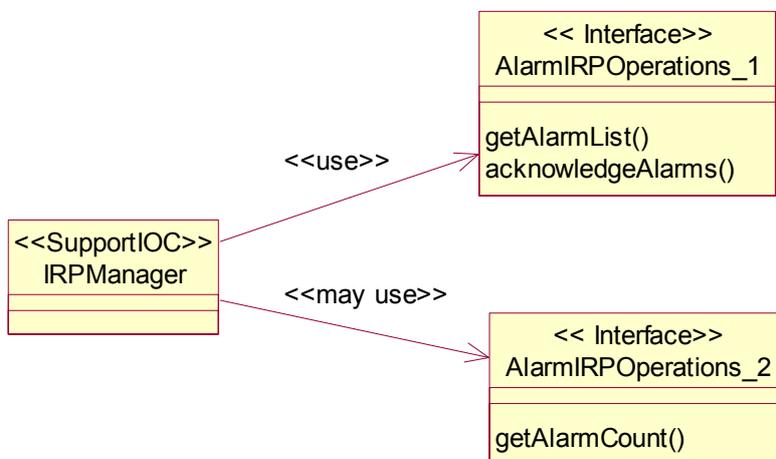
<<use>> и <<may use>> представляют собой однонаправленные ассоциации. Целевым элементом должен быть <<Interface>> или <<Notification>>.

Если целевым элементом является <<Interface>>, <<use>> указывает на то, что исходный класс должен иметь возможность использовать целевой элемент <<Interface>>, т. е. может вызывать операции, определенные элементом <<Interface>>. Поддержка такой возможности исходным объектом обязательна. <<may use>> указывает на то, что исходный класс может иметь возможность использовать целевой элемент <<Interface>>, т. е. может вызывать операции, определенные элементом <<Interface>>. Поддержка такой возможности исходным объектом необязательна.

Если целевым элементом является <<Notification>>, <<use>> указывает на то, что исходный класс должен выступать в качестве отправителя уведомлений, определенных целевым элементом <<Notification>>. Поддержка такой возможности исходным объектом обязательна. <<may use>> указывает на то, что исходный класс может выступать в качестве отправителя уведомлений, определенных целевым элементом <<Notification>>. Поддержка такой возможности исходным объектом необязательна.

С.3.6.1 Пример для целевого элемента <<Interface>>

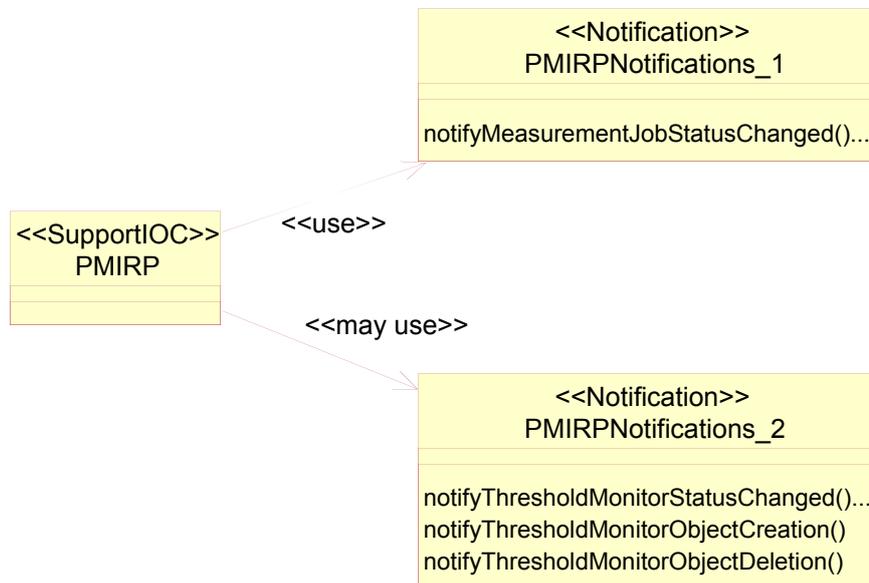
В данном примере показано, что Менеджер IRP должен использовать операции, определенные AlarmIRPOperations_1 и может использовать операции, определенные AlarmIRPOperations_2.



Нотация <<use>> и <<may use>> для целевого элемента <<Interface>>

С.3.6.2 Пример для целевого элемента <<Notification>>

В данном примере показано, что эталонная точка PMIRP должна иметь возможность создавать или отправлять уведомления, определенные PMIRPNotifications_1 и может иметь возможность создавать или отправлять уведомления, определенные PMIRPNotifications_2.



Нотация <<use>> и <<may use>> для целевого элемента <<Notification>>

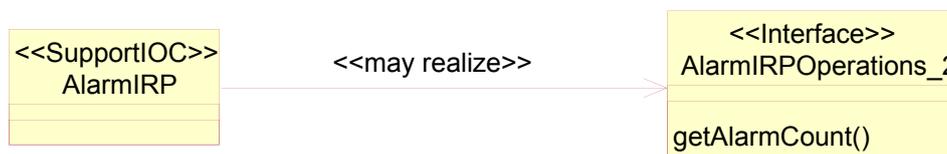
С.3.7 <<may realize>>

<<may realize>> представляет собой однонаправленную ассоциацию. Целевым элементом должен быть <<Interface>>. <<may realize>> показывает, что исходный объект может выполнять операции, определенные целевым элементом <<Interface>>.

Обратите внимание, что базовым элементом UML была определена ассоциация реализации (и, таким образом, нет необходимости в определении стереотипа такой ассоциации). Ассоциация реализации показывает, что исходный объект должен реализовывать (или выполнять) операции, определенные целевым элементом <<Interface>>.

С.3.7.1 Пример

В данном примере показано, что AlarmIRP может реализовывать операции AlarmIRPOperations_2.



Нотации <<may realize>>

С.3.8 <<names>>

Указывает на однонаправленный состав. Целевой экземпляр является однозначно определяемым в рамках пространства имен исходного объекта, среди всех других целевых экземпляров одного и того же целевого классификатора и среди других целевых экземпляров других классификаторов, имеющих одинаковый состав <<names>> с исходным элементом.

И исходный классификатор, и целевой классификатор должны иметь атрибут именованя.

Состав, используемый в качестве действия включения между именами, обеспечивает семантику отношений "целое-часть" между доменом и включенными именованными элементами, даже если только в части имен. С точки зрения управления, доступ к части можно получить через целое. Кратность должна быть указана с обеих сторон связи.

Целевой экземпляр не может иметь несколько <<names>> с несколькими источниками, т. е. целевой экземпляр не может участвовать или принадлежать нескольким пространствам имен.

С.3.8.1 Пример

В данном примере показано, что все экземпляры MscFunction являются однозначно определяемыми в рамках пространства имен экземпляра ManagedElement.



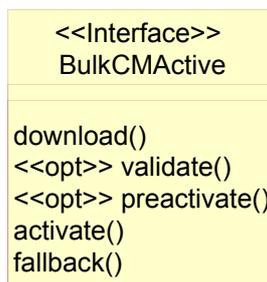
Нотация <<names>>

С.3.9 <<opt>>

<<opt>> (или <<optional>>) дает возможность указать на необязательность атрибутов, параметров и операций (соответственно) в диаграммах UML.

Если данный стереотип отсутствует, соответствующий атрибут, параметр или операция являются обязательными.

С.3.9.1 Пример



Нотация <<opt>> для операций

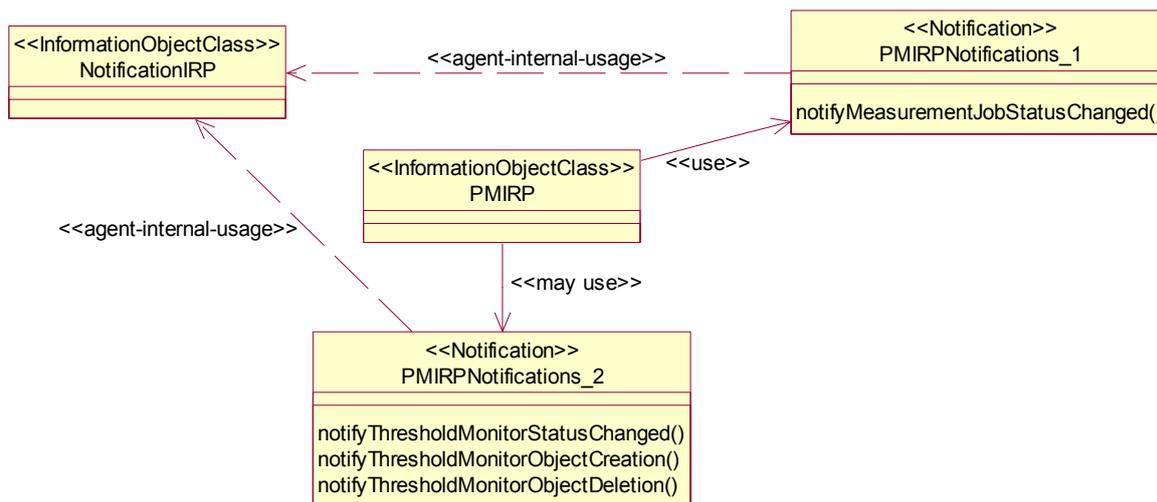
С.3.10 <<Notification>>

С.3.10.1 Общие положения

<<Notification>> представляет собой именованный набор уведомлений.

С.3.10.2 Пример

В данном примере показывается <<Notification>> с именем "PMIRPNotifications_1" и <<Notification>> с именем "PMIRPNotifications_2". Оба этих элемента имеют уведомление(я). Примером уведомления может служить notifyMeasurementJobStatusChanged().



Нотация <<Notification>>

С.3.11 <<agent-internal-usage>>

Это однонаправленная ассоциация. Исходный объект передает информацию о сетевом управлении целевому объекту. Исходный и целевой объекты представляют собой объекты или процессы, которые выполняются в различных экземплярах IRP, таких как AlarmIRP, PMIRP. Экземпляры могут иметь отношения включения между именами с одним и тем же Агентом IRP или различными экземплярами Агента IRP. Конкретная передаваемая информация об управлении сетью и механизм передачи информации не стандартизированы и зависят от поставщика.

С.3.11.1 Пример

В данном примере показано, что NLIRP (NotificationLog IRP) может передавать некоторую информацию об управлении сетью FTIRP (FileTransferIRP).



Нотация <<agent-internal-usage>>

С.3.12 <<SupportIOC>>

Это дескриптор набора возможностей управления.

<<SupportIOC>> идентичен классу UML, за исключением того, что в него не включены/в нем не определены методы и операции.

Подпункт 3.22.1 [OMG UML]: "Класс представляет собой концепцию в рамках моделируемой системы. Классы обладают структурой данных, поведением и связями с другими элементами".

С.3.12.1 Пример

В данном примере показан <<SupportIOC>> AlarmList.



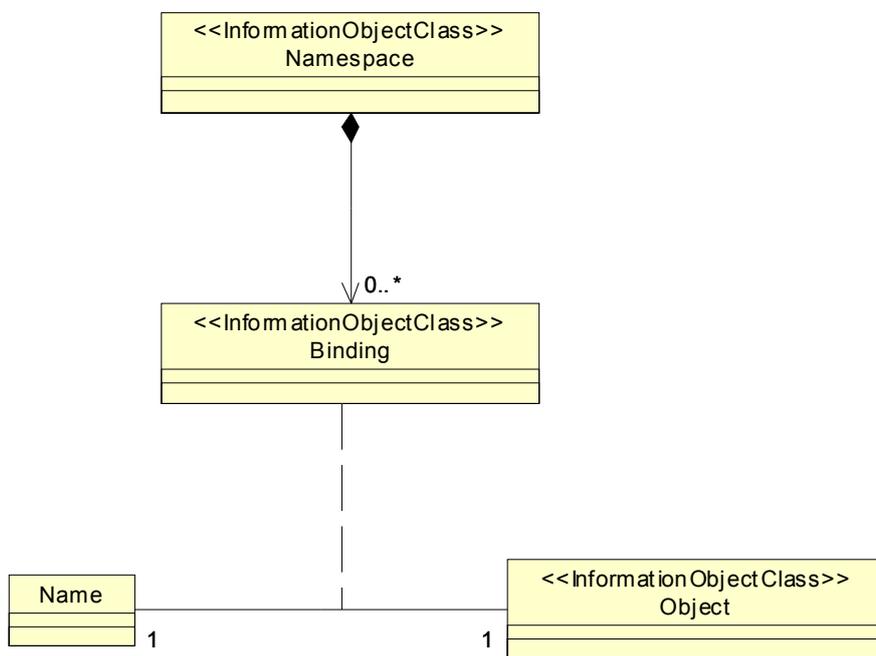
Нотация <<SupportIOC>>

С.4 Классы ассоциации

В подпункте 3.46 [OMG UML] класс ассоциации определяется следующим образом:

"Класс ассоциации представляет собой ассоциацию, которая также обладает свойствами класса (или класс, который обладает свойствами ассоциации). Несмотря на то, что он изображается в виде ассоциации и класса, на самом деле это один элемент модели".

Целесообразно использовать классы ассоциации тогда, когда классу "InformationObjectClass" необходимо поддерживать ассоциации с несколькими другими классами "InformationObjectClass", и при этом существуют связи между участниками ассоциаций в рамках "включающего" "InformationObjectClass". Например, пространство имен поддерживает набор привязок, привязка соединяет имя с объектом. Привязка "ИОС" может быть моделирована в виде Класса ассоциации, обеспечивающего семантику привязки для связи между именем и каким-либо другим классом "InformationObjectClass". Это показано на рисунке ниже (приводится в качестве примера, не взято из другой Рекомендации).



Пример Класса ассоциации

С.5 Абстрактный класс

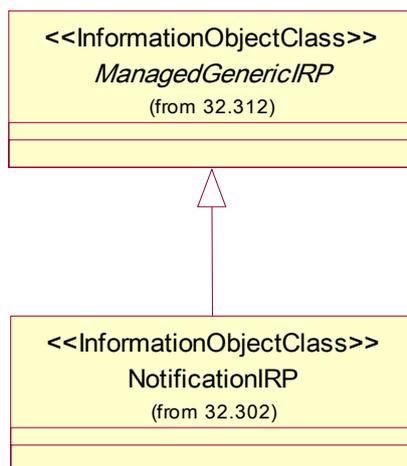
С.5.1 Общие положения

Указывает на <<InformationObjectClass>> как на базовый класс, который должен наследоваться подклассами. Непосредственно из абстрактного класса невозможно создавать объекты.

Нотацией абстрактного класса является использование курсива для написания имени класса соответствующего <<InformationObjectClass>> в диаграмме.

С.5.2 Пример

В данном примере показано, что ManagedGenericIRP представляет собой абстрактный класс <<InformationObjectClass>>.



Нотация абстрактного класса

С.6 Применение <<InformationObjectClass>> и <SupportIOC>>

<<InformationObjectClass>> и <SupportIOC>> представляют собой стереотипы. Два этих стереотипа служат аналогичной цели, поскольку каждый из них представляет собой именованный набор свойств управления. Вместе с тем области их применения в контексте поддержки управления интерфейсом управления могут отличаться. В данном пункте рассмотрены общие черты и различия применения таких стереотипов.

	<<InformationObjectClass>>	<<SupportIOC>>
Может ли представлять собой абстрактный класс?	Да	Да
Может ли представлять собой конкретный класс?	Да	Да
Может ли наследовать <<InformationObjectClass>>?	Да	Нет
Может ли наследовать <<SupportIOC>>?	Нет	Да
Может ли быть включен в <<InformationObjectClass>>?	Да	Да
Может ли быть включен в <<SupportIOC>>?	Нет	Да
Может ли экземпляр иметь номер DN?	<<InformationObjectClass>> должен представлять собой класс дерева имен, что означает, что все его экземпляры должны иметь номер DN.	<<SupportIOC>> может использоваться автором спецификации для класса в рамках дерева имен. В таком случае это означает, что все экземпляры должны иметь номер DN.
Может ли Менеджер получать информацию посредством уведомлений, для которых параметры objectClass и objectInstance несут в себе номер DN экземпляра?	Да. Типы создаваемых уведомлений показаны в Таблице уведомлений, связанной с определением класса.	Да, если <<SupportIOC>> представляет собой класс дерева имен. Типы создаваемых уведомлений показаны в Таблице уведомлений, связанной с определением класса. Нет, если <<SupportIOC>> не является классом дерева имен.

Приложение D

Проектирование

(Данное приложение является неотъемлемой частью данной Рекомендации.)

В данном приложении приводятся руководящие указания для разработок, определяемых протоколом. Данный вопрос подлежит дальнейшему изучению.

Приложение Е

Определение типов информации – набор типов

(Данное приложение является неотъемлемой частью данной Рекомендации.)

В данном приложении определен набор типов, которые должны использоваться для указания информации о типах в концептуальной модели (аналитической модели/информационном обслуживании).

Данный набор представляет собой поднабор типов, определенных в Рекомендации [ITU-T X.680] в сочетании с типами, полученными из типов, определенных ASN.1 (пункт Е.4).

Ключевые слова, которые должны использоваться для каждого типа, приведены в таблице Е.1.

Е.1 Базовые типы

Базовые типы – это те типы, которые могут быть использованы для непосредственного определения атрибутов и параметров. Базовые типы могут также использоваться для построения составных типов. К базовым типам относятся следующие типы ASN.1:

Е.1.1 целочисленный тип пункт 19 [ITU-T X.680]

Е.1.2 действительный тип пункт 21 [ITU-T X.680]

Е.1.3 логический тип пункт 18 [ITU-T X.680]

Е.1.4 тип битовых строк пункт 22 [ITU-T X.680]

Е.1.5 тип null пункт 24 [ITU-T X.680]

Е.1.6 обобщенный временной тип пункт 38 [ITU-T X.680]

Е.2 Тип перечислимых данных

В подпункте 20 Рекомендации [ITU-T X.680] о типе перечислимых данных представлены перечислимые значения. Все значения, которые могут использоваться тем или иным атрибутом или параметром, должны быть перечислены в столбцах допустимых значений. К концептуальной модели применяется только стиль перечислимых значений, т. е. определение конкретных значений (цифр или строк) зависит от конкретных моделей проектирования.

ПРИМЕЧАНИЕ. – Если число таких значений превышает 50, рекомендуется приводить их в приложении или в отдельном документе.

Е.3 Сложные/составные типы

Сложные типы могут быть определены при помощи следующих концепций:

Е.3.1 типы последовательности пункт 25 [ITU-T X.680]

Е.3.2 типы выбора пункт 29 [ITU-T X.680]

Е.3.3 типы набора пункт 27 [ITU-T X.680]

Кроме того, списки и наборы сложных типов поддерживаются с использованием:

Е.3.4 типов последовательности данных пункт 26 [ITU-T X.680]

Е.3.5 типов наборов данных пункт 28 [ITU-T X.680]

Е.4 Полезные типы

Е.4.1 Тип строки

Строка представляет собой строку символов, набор символов не ограничен, т. е.:

String ::= UnrestrictedCharacterStringType пункт 44 Рекомендации [ITU-T X.680]

Е.4.2 Тип имени

Name представляет собой уникальное имя экземпляра объекта в пространстве имен. В него может включаться информация о дереве иерархии включения объектов, однако оно зависит от выполнения и не входит в сферу действия данной Рекомендации. Формально тип имени определяется таким образом:

Name ::= TYPE-IDENTIFIER Приложение А Рекомендации [ITU-T X.681]

Е.5 Ключевые слова

В таблице Е.1 определен список ключевых слов, которые должны использоваться в примере анализа (см. Приложение В) для определения типа информации, например:

Имя параметра	Квалификатор Support	Тип информации/ Допустимые значения	Комментарий
...			
eventIdList	M	SET OF INTEGER/-	Перечень сигналов, которые должны быть подтверждены.

Таблица Е.1 – Ключевые слова

Тип	Ключевое слово
integer type [целочисленный тип]	INTEGER
real type [действительный тип]	REAL
boolean type [логический тип]	BOOLEAN
bitstring type [тип битовых строк]	BIT STRING
null type [тип, допускающий значение null]	NULL
generalized time type [обобщенный временной тип]	GeneralizedTime
enumerated type [тип перечислимых данных]	ENUMERATED
sequence type [тип последовательности]	SEQUENCE
choice type [тип выбора]	CHOICE
set type [тип набора]	SET
sequence-of type [тип последовательности данных]	SEQUENCE OF
set-of type [тип наборов данных]	SET OF
string type [тип строки]	String
name type [тип имени]	Name

Приложение F

Руководящие указания по свойствам ИОС, наследованию и импорту объектов

(Данное приложение является неотъемлемой частью данной Рекомендации.)

Приведенные ниже руководящие указания базируются на документе [b-3GPP TS 32.150].

F.1 Свойство ИОС

Свойства класса информационных объектов (включая Support ИОС) определяются с учетом следующих положений:

- a) Атрибут(ы) класса информационных объектов, включая семантику и синтаксис, диапазон допустимых значений и поддерживаемые квалификаторы. Атрибуты ИОС не ограничиваются Configuration Management (управлением конфигурацией), но также включают атрибуты, относящиеся, например, к: 1) Performance Management (управлению производительностью) (т. е. типам измерений); 2) Trace Management (управлению трассировкой); и 3) Accounting Management (управлению счетом).
- b) Поведение, не зависящее от атрибутов, связанное с ИОС.
ПРИМЕЧАНИЕ 1. – Например, Связь между MscServerFunction и CsMgwFunction является необязательной. Это обязательно в случае, если экземпляр MscServerFunction принадлежит к одному экземпляру ManagedElement, а экземпляр CsMgwFunction принадлежит к другому экземпляру ManagedElement. Поведение этой Связи не зависит от атрибутов. Ожидается, что в таком поведении, как и в других случаях, будет прослеживаться наследование.
- c) Связь(и) класса информационных объектов с другим(и) классом(ами) информационных объектов.
- d) Тип(ы) уведомлений ИОС и их квалификаторы.
- e) Связи класса информационных объектов с родительскими объектами (см. Примечание 2). Существуют три взаимоисключающих случая:
 - 1) Класс информационных объектов может иметь любой родительский объект. В диаграмме UML класс имеет родительский объект Any.
 - 2) Класс информационных объектов является абстрактным, все возможные родительские объекты были назначены, и было определено, могут ли подклассы классов информационных объектов назначаться в качестве корневого класса информационных объектов. В диаграмме UML класс имеет нулевое или большее число возможных родительских элементов конкретных классов (за исключением Any).
 - 3) Класс информационных объектов является конкретным, все возможные родительские объекты были назначены, и было определено, может ли класс информационных объектов назначаться в качестве корневого класса информационных объектов. В диаграмме UML класс имеет один или большее число возможных родительских объектов конкретных классов (за исключением Any).Экземпляр класса информационных объектов или является корневым классом информационных объектов, или имеет только один родительский объект. Только 3GPP SA5 может назначать класс ИОС как возможный корневой ИОС. В данный момент только такие классы информационных объектов, как SubNetwork, ManagedElement и MeContext, могут быть корневыми классами информационных объектов.
ПРИМЕЧАНИЕ 2. – Связи между родительскими и дочерними объектами в данном подпункте представляют собой отношения включения между родительским и дочерним объектом.
- f) Связи класса информационных объектов с дочерними объектами. Существуют три взаимоисключающих случая:
 - 1) Класс информационных объектов не должен иметь дочерних классов информационных объектов (отношения включения). В диаграмме UML класс не имеет дочернего объекта.

- 2) Класс информационных объектов может иметь дочерний(ие) класс(ы) информационных объектов. Может указываться максимальное число экземпляров для дочернего класса информационных объектов. Для класса информационных объектов может определяться, что зависящие от поставщика объекты не могут выступать в роли дочерних классов информационных объектов. В диаграмме UML класс имеет дочерний объект Any.
 - 3) Класс информационных объектов может иметь только конкретный(ые) дочерний(ие) класс(ы) информационных объектов (или их подклассы). Может указываться максимальное число экземпляров для дочернего класса информационных объектов. Для класса информационных объектов может определяться, что зависящие от поставщика объекты не могут выступать в роли дочерних классов информационных объектов. В диаграмме UML класс имеет один или большее число дочерних элементов конкретных классов (за исключением Any).
- g) Могут ли создаваться экземпляры класса информационных объектов (т. е. является ли класс информационных объектов абстрактным).
- h) Атрибут с целью присвоения имен.

Ф.2 Наследование

Когда класс информационных объектов (подкласс) наследует другой класс информационных объектов (суперкласс), это означает, что подкласс обладает всеми свойствами суперкласса.

Подкласс может изменять необязательные унаследованные поддерживаемые квалификаторы на обязательные, но не наоборот. Подкласс может изменять условно необязательные унаследованные поддерживаемые на условно обязательные, но не наоборот.

Класс информационных объектов может выступать в роли суперкласса для многих классов информационных объектов. В то же время, подкласс не может иметь более одного суперкласса.

Подкласс может:

- a) Добавлять (по сравнению с суперклассом) уникальные атрибуты, в том числе поведение, диапазоны допустимых значений и поддерживаемых квалификаторов. Каждому дополнительному атрибуту должно быть присвоено свое уникальное имя (для всех добавленных и унаследованных атрибутов).
- b) Добавлять безатрибутное поведение на базе ИОС. Такое поведение не может противоречить унаследованному поведению суперкласса.
- c) Добавлять связь(и) с другим(и) классом(ами) информационных объектов. Каждой дополнительной связи должно быть присвоено свое уникальное название (для всех добавленных и унаследованных связей).
- d) Добавлять дополнительные типы уведомлений и их квалификаторы.
- e) Назначать все возможные родительские объекты (и их подклассы), если суперкласс обладает свойством Property-e-1, так что класс информационных объектов будет обладать свойствами Property-e-2 или Property-e-3. Ограничивать возможные родительские элементы (и их подклассы) и/или ограничивать возможность для подкласса выступать в качестве корневого класса информационных объектов, если суперкласс обладает свойствами Property-e-2 или Property-e-3.
- f) Добавлять дочерние классы информационных объектов, если суперкласс обладает свойством Property-f-2, так что класс информационных объектов будет обладать свойством Property-f-3. Ограничивать допустимые дочерние классы информационных объектов (или их подклассы), если суперкласс обладает свойством Property-f-3.
- g) Указывать, могут ли создаваться экземпляры класса информационных объектов (т. е. является ли класс информационных объектов абстрактным).
- h) Ограничивать диапазон допустимых значений атрибута суперкласса, имеющего диапазон допустимых значений.

Г.3 Импорт объектов (интерфейсов, классов информационных объектов и атрибутов)

Спецификации интерфейсов управления определяют объекты (например, классы информационных объектов, интерфейсы и атрибуты). Чтобы способствовать повторному использованию определений объектов в спецификациях интерфейса, используется механизм импорта. Когда спецификацией интерфейса управления (субъектной спецификацией) импортируется объект, определенный в другой спецификации интерфейса управления, считается, что субъектная спецификация определяет импортированный объект. Кроме того, субъектная спецификация не может изменять свойства такого импортированного объекта. В случае если требуется объект, не идентичный импортированному объекту, но сходный с ним, необходимо определить новый объект, который бы наследовал импортированный объект, и внести изменения в определение такого нового объекта.

Дополнение I

Пример требований

(Данное дополнение не является неотъемлемой частью данной Рекомендации.)

ПРИМЕЧАНИЕ. – Приведенный ниже пример базируется на управлении аварийными сигналами, при этом он приведен исключительно с иллюстративной целью и не предназначен для предоставления полного или правильного набора требований по управлению аварийными сигналами.

1 Концепции и базовая информация

Любая оценка СЭ и общего состояния работоспособности сети требует обнаружения сбоев в сети и, соответственно, отправки аварийных сигналов OS (EM и/или NM).

2 Требования коммерческого уровня

2.1 Требования

Сбои, возникающие в сети, могут быть сгруппированы по следующим категориям:

- Сбои оборудования, т. е. неисправность какого-либо физического ресурса СЭ.
- Неполноценности программного обеспечения, например, ошибки программного обеспечения, несовместимость баз данных.

2.1.1 Обнаружение неисправностей

REQ-FM-FUN-01 Для большинства неисправностей должны иметься четко определенные условия определения их наличия или отсутствия, т. е. условия появления и устранения неисправностей. Любой такой инцидент в данном приложении должен именоваться как неисправность ADAC. Сетевые объекты должны быть способны распознавать то, когда ранее выявленная неисправность ADAC больше не присутствует, т. е. распознавать устранение неисправности, с использованием тех же методов, которые используются ими для обнаружения неисправности.

2.1.2 Сброс аварийных сигналов

Аварийные сигналы, созданные в результате неисправностей, должны быть сброшены. Для сброса аварийного сигнала, как правило, необходимо устранить соответствующую неисправность.

...

REQ-FM-FUN-02 Каждый раз при сбросе аварийного сигнала Агент должен сформировать соответствующее событие сброса аварийного сигнала. Устранение аварийного сигнала определяется, как аварийный сигнал.

2.1.3 Переадресация и фильтрация аварийных сигналов

REQ-FM-FUN-03 Для каждой выявленной неисправности сетевым объектом, в котором возникла такая неисправность, должны быть сформированы соответствующие аварийные сигналы (уведомления о неисправности).

...

2.2 Роли актеров

Управляемая система Объект, выполняющий роль агента.

Управляющая система Объект, выполняющий роль менеджера.

2.3 Ресурсы электросвязи

В данной Рекомендации управляемое сетевое оборудование рассматривается как часть ресурсов электросвязи.

2.4 Диаграммы вариантов использования высокого уровня

2.4.1 Аварийный отчет

На первой обзорной диаграмме вариантов использования на рисунке I.1 показаны общие взаимодействия интерфейса аварийных сигналов.

На первой обзорной диаграмме вариантов использования показаны взаимодействия, которые происходят при формировании отчетов о выявленных неисправностях.



M.3020(11)_F.1.1

Рисунок I.1 – Аварийный отчет

3 Требования уровня спецификаций

3.1 Требования

Требования уровня спецификации отсутствуют.

3.2 Роли актеров

См. подпункт 2.2 данного примера.

3.3 Ресурсы электросвязи

См. подпункт 2.3 данного примера.

3.4 Варианты использования

3.4.1 Уведомления об отказах

Этап вариантов использования	Эволюция/Спецификация	<<Uses>> Соответствующее использование
Цель (*)	При выявлении сбоя управляемая система направляет управляющей системе аварийное уведомление соответствующего типа посредством интерфейса Q.	
Актеры и Роли (*)	Управляющая система выступает получателем уведомлений от управляемой системы.	
Ресурсы электросвязи	Любой управляемый объект.	
Допущения	Выявлено условие возникновения сбоя.	
Предварительные условия	Имеется открытый канал связи между управляющей системой и управляемой системой.	
Когда начинается	Выявление условия возникновения сбоя.	
Шаг 1 (*)	В случае выявления условия возникновения сбоя создается аварийный отчет или аварийный отчет системы защиты.	
Когда заканчивается	Агентом создается аварийный отчет или аварийный отчет системы защиты.	

Этап вариантов использования	Эволюция/Спецификация	<<Uses>> Соответствующее использование
Исключения	Сбой связи или процесса может привести к тому, что аварийный отчет не будет доставлен управляющей системе. Такая ситуация может быть разрешена при помощи варианта использования аварийной синхронизации.	
Заключительные условия	Управляющая система проинформирована о сбое в управляемой системе.	
Возможность отслеживания (*)	REQ-FM-FUN-01, REQ-FM-FUN-02 и т. д.	

3.4.2 Сброс аварийного сигнала

...

3.4.3 Подтверждение приема аварийного сигнала

...

Дополнение II

Пример анализа

(Данное дополнение не является неотъемлемой частью данной Рекомендации.)

ПРИМЕЧАНИЕ. – Приведенный ниже пример базируется на управлении аварийными сигналами, при этом он приведен исключительно с иллюстративной целью и не предназначен для предоставления полного или правильного набора требований по управлению аварийными сигналами.

1 Концепции и базовая информация

Любая оценка СЭ и общего состояния работоспособности сети требует обнаружения сбоев в сети и, соответственно, отправки аварийных сигналов OS (EM и/или NM).

...

2 Классы информационных объектов

2.1 Импортированные информационные объекты и локальные метки

Ссылка на метку	Локальная метка
3GPP TS 32.302, information object class, NotificationIRP	NotificationIRP
3GPP TS 32.302, interface, notificationIRPNotification	NotificationIRPNotification
3GPP TS 32.622, information object class, IRPAgent	IRPAgent
3GPP TS 32.312, information object class, ManagedGenericIRP	ManagedGenericIRP

2.2 Диаграмма классов

В данном подпункте представлен набор классов информационных объектов, которые несут в себе информацию в рамках агента. Цель состоит в определении информации, которая требуется AlarmAgent для выполнения операций и создания уведомлений. В данном подпункте предлагается обзор всех вспомогательных классов объектов в UML. Следующие подпункты содержат более подробное описание различных объектов таких вспомогательных классов объектов.

2.2.1 Атрибуты и связи

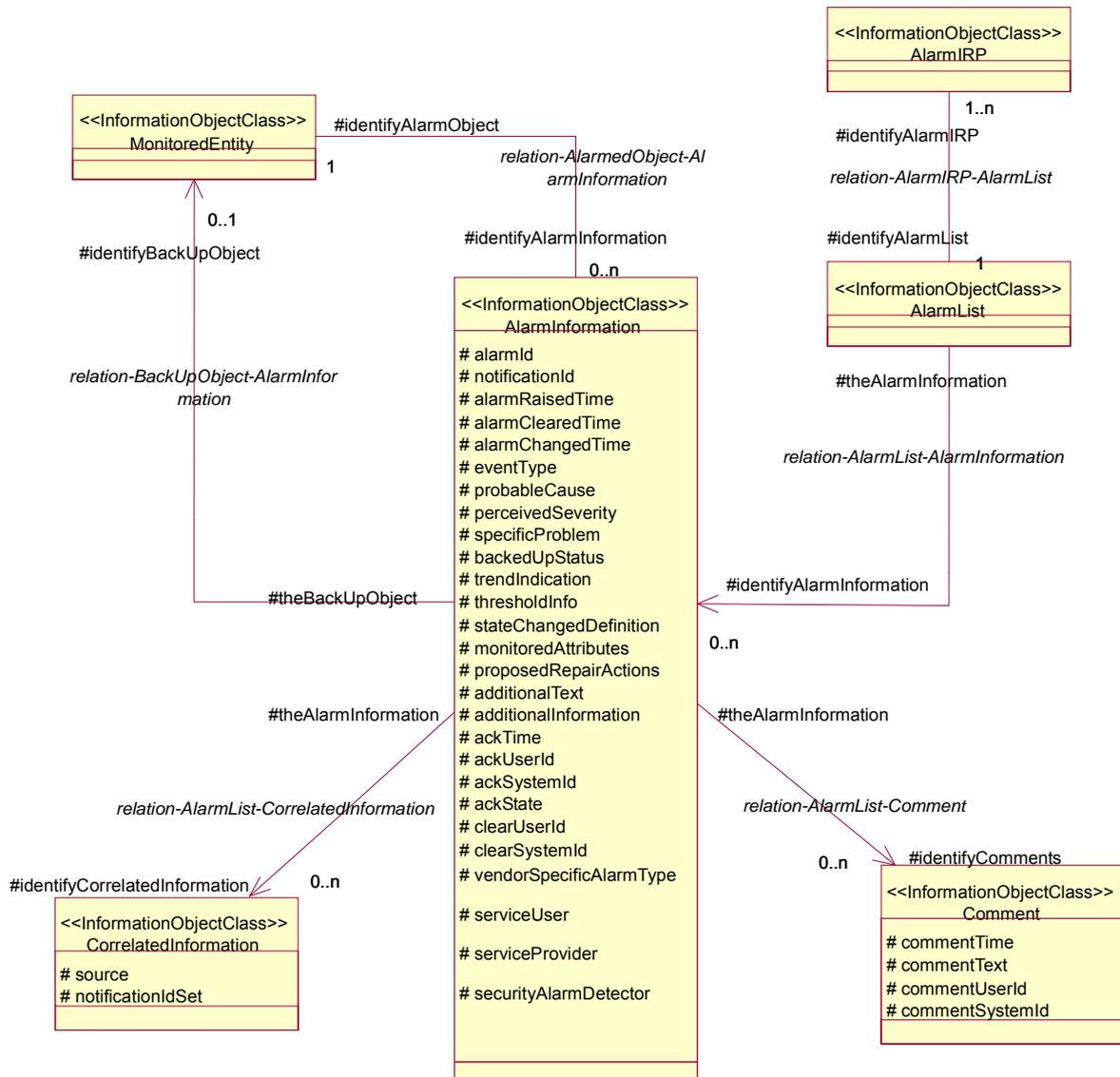


Рисунок П.1 – Классы информационных объектов управления аварийными сигналами

2.2.2 Наследование

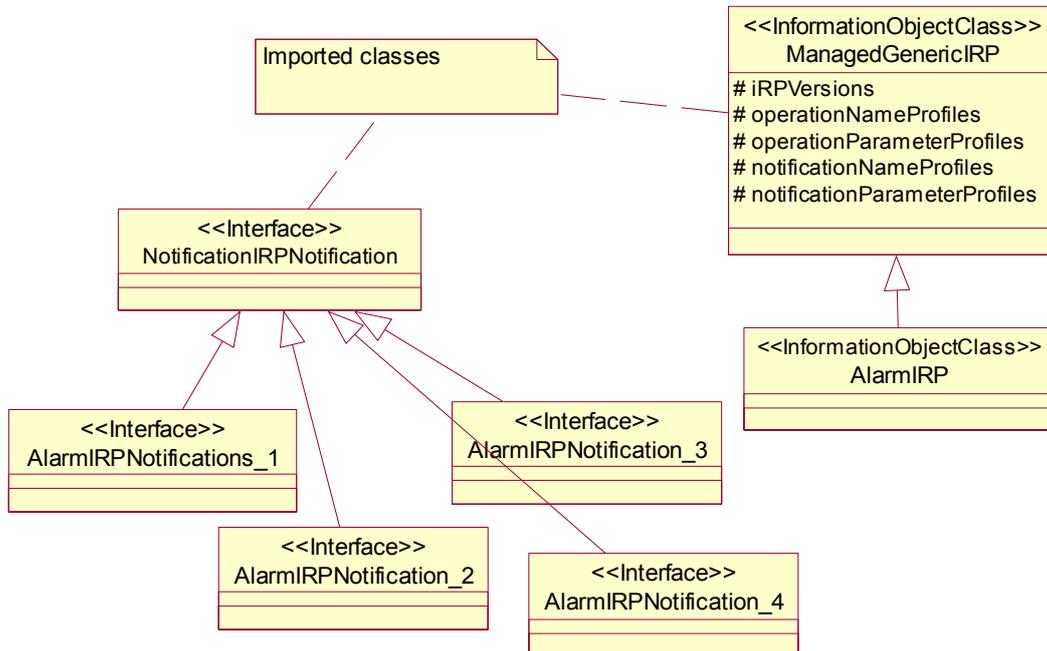


Рисунок П.2 – Наследование классов информационных объектов управления аварийными сигналами

2.3 Определения классов информационных объектов

Имя класса	Квалификатор	Идентификатор требований
AlarmInformation	M	REQ-FM-FUN-01, REQ-FM-FUN-02 и др.
AlarmList	M	REQ-FM-FUN-n
...		

2.3.1 Класс AlarmInformation

2.3.1.1 Определение

Класс AlarmInformation содержит информацию об аварийном состоянии объекта MonitoredEntity, подающего аварийные сигналы.

...

2.3.1.2 Атрибуты

Имя атрибута	Квалификатор Support	Квалификатор Read	Квалификатор Write	Идентификаторы требований
alarmed	M	M	M	
probableCause	C	M	C	
structuredProbableCause	C	M	C	
perceivedSeverity	M	M	M	
specificProblem	O	O	O	
...				
...				

2.3.1.3 Диаграмма состояний

Аварийные сигналы могут иметь разные состояния.

...

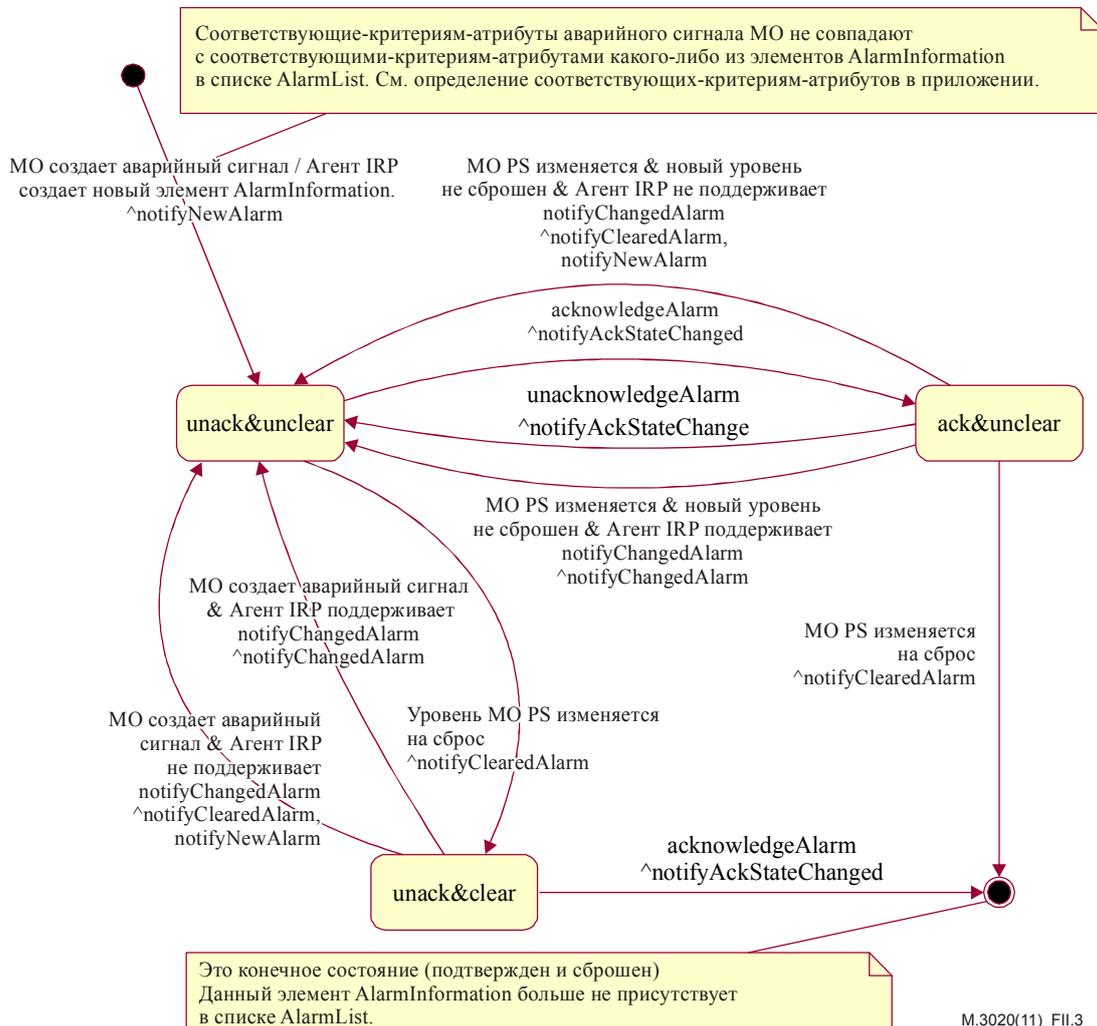


Рисунок П.3 – Диаграмма состояний аварийных сигналов

2.3.2 Класс AlarmList

2.4 Определение информационных связей

Связь	Квалификатор Support	Идентификатор требований
relation-AlarmIRP-AlarmList	M	REQ-FM-FUN-x
...		

2.4.1 Класс relation-AlarmIRP-AlarmList (M)

2.4.1.1 Определение

Представляет связь между AlarmIRP и AlarmList.

2.4.1.2 Роли

Имя	Определение
identifyAlarmIRP	Отражает возможность получения идентификаторов одной или нескольких AlarmIRP.
identifyAlarmList	Отражает возможность получения идентификатора одного AlarmList.

2.4.1.3 Ограничения

Для данных связей ограничения отсутствуют.

2.4.2 Класс relation-AlarmList-AlarmInformation (M)

...

2.5 Определение атрибутов информации

2.5.1 Определение и допустимые значения

Имя	Определение	Тип информации/ Допустимые значения
alarmed	Определяет один элемент AlarmInformation в AlarmList.	INTEGER
notificationId	Определяет уведомление, которое содержит AlarmInformation.	INTEGER
ntfSubscriptionState	Указывает состояние активации подписки	ENUMERATED/"suspended": подписка приостановлена. "notSuspended": подписка активна.

2.5.2 Ограничения

Имя	Затронутый(ые) атрибу(т)ы	Определение
inv_notificationId	notificationId	Идентификаторы NotificationId должны выбираться таким образом, чтобы они были уникальными для всех уведомлений в рамках того или иного управляемого объекта (представляющего СЭ) в течение всего времени, пока корреляция аварийных сигналов является существенной. Алгоритм обеспечения корреляции аварийных сигналов выходит за рамки данной IRP.

3 Определение интерфейса

3.1 Диаграмма классов с представлением интерфейса

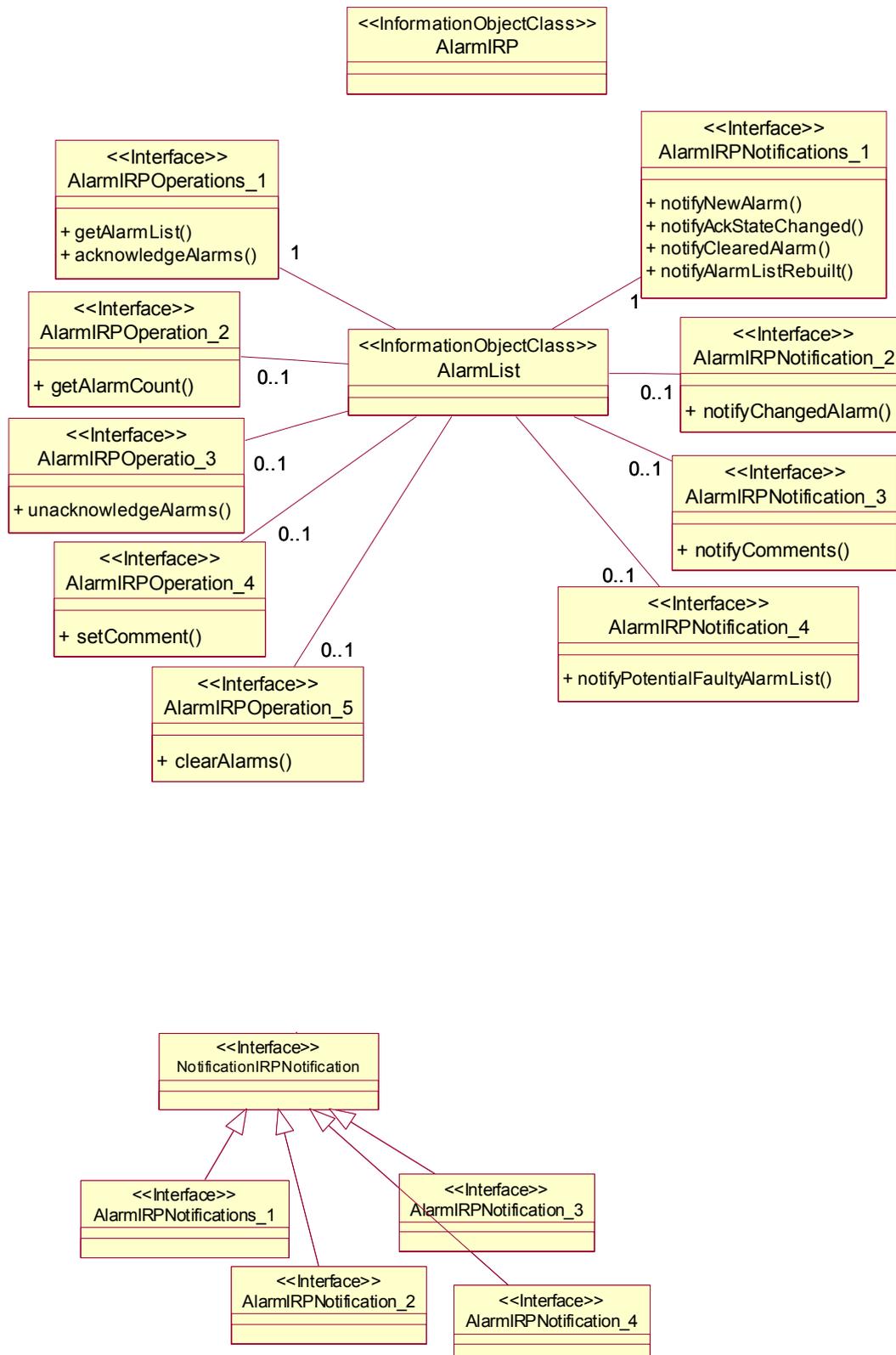


Рисунок П.4 – Диаграмма классов управления аварийными сигналами IRP

3.2 Общие правила

Правило 1: Каждая операция, по меньшей мере, с одним входным параметром поддерживает предварительное условие `valid_input_parameter`, которое указывает на то, что все входные параметры должны быть действительными применительно к используемому типу информации. Кроме того, каждая такая операция поддерживает исключение `operation_failed_invalid_input_parameter`, которое применяется, когда указанный в предварительных условиях `valid_input_parameter` является ложным. Данное исключение характеризуется одинаковым состоянием на входе и на выходе.

Правило 2: Каждая операция, по меньшей мере, с одним необязательным входным параметром, поддерживает набор предварительных условий `supported_optional_input_parameter_xxx`, где "xxx" представляет собой название необязательного входного параметра, а предварительное условие состоит в том, что операция поддерживает названный необязательный входной параметр. Кроме того, каждая такая операция поддерживает исключение `operation_failed_unsupported_optional_input_parameter_xxx`, которое применяется, когда:

- a) `pre-condition supported_optional_input_parameter_xxx` является ложным; и
- b) названный необязательный входной параметр несет в себе информацию.

Данное исключение характеризуется одинаковым состоянием на входе и на выходе.

Правило 3: Каждая операция должна поддерживать общее исключение `operation_failed_internal_problem`, которое выполняется в случае возникновения внутренней проблемы и невозможности завершения операции. Данное исключение характеризуется одинаковым состоянием на входе и на выходе.

3.3 Операции интерфейса AlarmIRPOperations_1 (O)

Имя операции	Квалификатор	Идентификатор требований
<code>acknowledgeAlarms</code>	M	REQ-FM-FUN-x, REQ-FM-FUN-y
<code>getAlarmList</code>	M	...

3.3.1 Операция `acknowledgeAlarms` (M)

3.3.1.1 Определение

Менеджер запускает данную операцию с целью подтверждения одного или более аварийных сигналов.

3.3.1.2 Входные параметры

Имя параметра	Квалификатор Support	Тип информации/ Допустимые значения	Комментарий
...			
<code>eventIdList</code>	M	SET OF INTEGER/–	Список аварийных сигналов, которые должны быть подтверждены.

3.3.1.3 Выходные параметры

Имя параметра	Квалификатор Support	Соответствующая информация/ Тип информации/ Допустимые значения	Комментарий
...			
<code>Status</code>	M	-- / ENUM / "OperationSucceeded": Если выполняется условие <code>allAlarmsAcknowledged</code> , "OperationPartiallySucceeded": Если выполняется условие <code>someAlarmAcknowledged</code> , "OperationFailed": Если выполняется условие <code>operationFailed</code> .	

3.3.1.4 Предварительное условие

atLeastOneValidId.

Название утверждения	Определение
atLeastOneValidId	Список AlarmInformationReferenceList содержит, по меньшей мере, один идентификатор, определяющий один элемент AlarmInformation в списке AlarmList, и такой определенный элемент AlarmInformation должен иметь статус ackState, в котором бы указывалось "unacknowledged" и, при наличии, соответствующую степень серьезности (perceivedSeverity).

3.3.1.5 Заключительное условие

someAlarmAcknowledged OR allAlarmsAcknowledged.

Название утверждения	Определение
someAlarmAcknowledged	...
allAlarmsAcknowledged	...

3.3.1.6 Исключения

Название	Определение
operation_failed	Условие: Предварительное условие является ложным или заключительное условие является ложным. Возврат информации: Статус выходного параметра. Состояние на выходе: Состояние на входе.

3.3.2 Операция getAlarmList (M)

...

Дополнение III

Сравнение с Рекомендацией МСЭ-Т Z.601

(Данное дополнение не является неотъемлемой частью данной Рекомендации.)

В данном дополнении предоставлена информация с целью сопоставления данной Рекомендации и Рекомендации [b-ITU-T Z.601], которая используется для разработки Рекомендаций МСЭ-Т серии M.1400.

В то время как в данной Рекомендации представлена методика спецификации интерфейсов управления между двумя физическими системами, в Рекомендации [b-ITU-T Z.601] предлагается платформа для разработки одной системы. Такая архитектура данных определяет возможные интерфейсы в рамках одной системы, а также интерфейсы на границе данной системы. Такие граничные интерфейсы существуют между системами.

Методика, представленная в данной Рекомендации, в первую очередь направлена на разработку набора Рекомендаций по вопросам интерфейса управления, чем на разработку отдельных систем. Архитектура данных не предписывает определения требований, как и на стадии требований, поскольку такая архитектура предписывает только спецификацию отдельных систем, а не их целей в отношении организации.

В Рекомендации [b-ITU-T Z.601] акцент сделан на спецификации внешней терминологии и грамматики в восприятии конечных пользователей. Данная Рекомендация направлена на спецификацию интерфейсов управления, которые не могут восприниматься конечными пользователями.

В данной Рекомендации требования в отношении решения проблемы могут быть разделены на два класса. Первый класс требований мы будем называть коммерческими требованиями, второй класс – требованиями спецификации. Требования спецификации могут включать требования по поддержке взаимодействий конечных пользователей в рамках интерфейсов "человек-машина". Некоторые из этих требований могут содержать синтаксические требования, которые должны поддерживаться любым интерфейсом управления. Синтаксические требования соответствуют схеме внешней терминологии архитектуры данных, как описано в Рекомендации [b-ITU-T Z.601].

Результатом стадии анализа должна стать информационная модель. Это соответствует концептуальной схеме архитектуры данных, как описано в Рекомендации [b-ITU-T Z.601]. Если информационные модели, полученные на стадии анализа, не передают всей необходимой информации из синтаксических требований, может быть необходимо включить в проектирование реализации отображение из синтаксических требований.

Документация на стадии проектирования состоит из двух частей:

- 1) Зависимая от технологии спецификация данных, общая для нескольких интерфейсов, например, с использованием GDMO или CORBA IDL, в соответствии со схемой внутренней терминологии согласно архитектуре данных, определенной в Рекомендации [b-ITU-T Z.601].
- 2) Зависимая от технологии спецификация для каждого интерфейса, например, с использованием SMIP или CORBA IDL, в соответствии со схемой распределения согласно архитектуре данных, определенной в Рекомендации [b-ITU-T Z.601].

Дополнение IV

Вопросы для дальнейшего изучения

(Данное дополнение не является неотъемлемой частью данной Рекомендации.)

В данном дополнении определены известные вопросы, подлежащие дальнейшему изучению.

IV.1 SOA

Утверждение Рекомендации [ITU-T M.3060] (Принципы управления сетями последующих поколений) ознаменовало переход от объектно-ориентированного к сервисно-ориентированному подходу к управлению. Влияние таких изменений подлежит дальнейшему изучению для определения необходимых изменений, которые потребуется внести при пересмотре данной Рекомендации в будущем.

IV.2 UML

Данное издание Рекомендации МСЭ-Т М.3020 ссылается на версию UML 1.5 с целью обеспечения согласованности с соответствующими спецификациями 3GPP. Пересмотренные издания Рекомендации МСЭ-Т М.3020 должны ссылаться на более поздние версии UML:

- В метамодели OMG MOF UML 2.x интегрирован как метамодель, которую поддерживают ведущие поставщики инструментов в отрасли. До появления UML 2.0 не существовало общей метамодели, а сам язык UML не был стандартизирован. MOF поддерживает добавление и создание новых метамodelей, точно определенных посредством OCL, который представляет собой язык исчисления предикатов.
- Как компании (предприятия электросвязи, правительственные и военные организации), так и поставщики инструментов сходятся на использовании модели OMG MOF.
- Преимуществами метамодели MOF состоят в том, что она поддерживает семейство метамodelей, которые могут использоваться для определения моделей объектов, связей ЧМИ, различных реализаций, определяемых технологиями, а также допускает реализацию стандартных методов перехода между моделями. Это не может быть реализовано в UML 1.5, поскольку UML 1.5 существует в изоляции от метамодели более высокого уровня.

IV.3 Видимость

Предлагается, чтобы по умолчанию параметры видимости звучали как "закрытый" (private) для атрибутов и "открытый" (public) для операций с целью обеспечения инкапсуляции данных и сокращения времени и усилий на определение модели реализации.

IV.4 Определение типов

При написании новой спецификации на базе данной методики необходимо указывать типы параметров и атрибутов. В данном издании Рекомендации не приводятся формальные определения типов, поэтому определение типов может быть различным и неоднородным для одних и тех же значений в разных спецификациях, например, массив целых чисел может быть определен как список целых чисел, последовательность целых чисел или набор целых чисел.

В Приложении E определены типы, которые могут использоваться в концептуальной модели.

Дополнение V

Дополнительные примеры использования UML

(Данное дополнение не является неотъемлемой частью данной Рекомендации.)

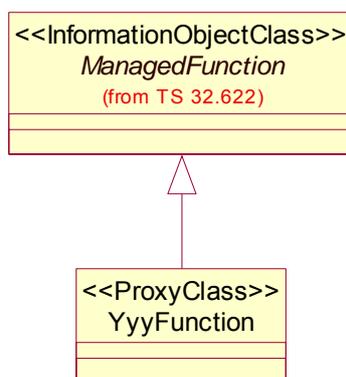
В данном дополнении приводятся дополнительные примеры использования UML, описанные в Приложении С.

V.1 Прокси-класс

V.1.1 Первый пример

В данном примере показан <<ProxyClass>> под названием YyyFunction. В нем представлены все классы информационных объектов, приведенные в Примечании под диаграммой UML. Все приведенные классы информационных объектов, в контексте данного примера, наследуют класс информационных объектов ManagedFunction IOC.

Использование <<ProxyClass>> снимает потребность изображать несколько ячеек <<InformationObjectClass>> UML, а именно тех, имена которых перечислены в приложении в диаграмме UML.



ПРИМЕЧАНИЕ. – Функция <<ProxyClass>> YyyFunction представляет функции AsFunction, AucFunction, BgFunction и др.

Пример Нотации <<ProxyClass>> V.1

V.1.2 Второй пример

В данном примере показан <<ProxyClass>> под названием YyyFunction. В нем представлены все классы информационных объектов, приведенные в Примечании под диаграммой UML. В контексте данного примера все перечисленные классы информационных объектов имеют отношения связи (внутренние и внешние).

Подлинные имена классов информационных объектов, представленные InternalYyyFunction <<ProxyClass>> и ExternalYyyFunction <<ProxyClass>>, перечислены согласно подпункту X.Y соответствующей функции YyyFunction. Например, согласно X.Y.1 для AsFunction добавляются два абзаца для перечисления всех одноранговых внутренних объектов и внешних объектов, которые связаны с AsFunction. В примере ниже AsFunction приводится как пример YyyFunction.

Подлинные имена классов информационных объектов, представленные Link_a_z <<ProxyClass>> и ExternalLink_a_z <<ProxyClass>>, перечислены согласно подпункту X.Y соответствующей функции YyyFunction. Например, согласно X.Y.1 для AsFunction добавляются два абзаца для перечисления имен классов информационных объектов, представленных Link_a_z и ExternalLink_a_z. В цитируемом ниже тексте AsFunction используется как пример для YyyFunction.

Х.У.1 AsFunction

Х.У.1.1 Определение

Данный класс информационных объектов представляет функциональность As. Дополнительную информацию о функции As можно найти в документе [b-3GPP TS 23.002].

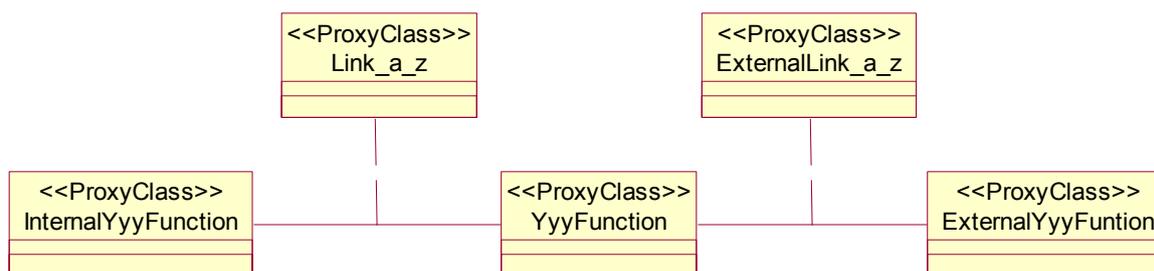
Связанная функция InternalYyyFunction <<ProxyClass>> представляет SlsFunction, CscfFunction, HlrFunction...

функция ExternalYyyFunction <<ProxyClass>> представляет...

Link_a_z <<ProxyClass>> представляет Link_As_Scscf, Link_Bgcf_Scscf...

ExternalLink_a_z <<ProxyClass>> представляет...

"



ПРИМЕЧАНИЕ. – Элемент "Yyy" функции YyyFunction <<ProxyClass>> представляет AsFunction, AucFunction и др.

Пример Нотации <<ProxyClass>> V.2

Дополнение VI

Руководящие указания по нумерации требований

(Данное дополнение не является неотъемлемой частью данной Рекомендации.)

Для нумерации требований применяется следующий формат:

REQ-метка-категория-номер,

где "метка" – это сокращение для Рекомендации (или ее части). Набор меток не ограничен и не подлежит стандартизации. Набор категорий определен в данной Рекомендации.

Некоторые вопросы:

- Каким образом должны быть структурированы метки в больших спецификациях требований?
- Каким образом должно осуществляться удаление и добавление требований?

Могут оказаться полезными следующие руководящие указания:

- Никогда не следует изменять нумерацию требований. Единственным исключением из этого правила является первое издание спецификации, однако даже в этом случае может быть лучше избежать изменения нумерации, поскольку спецификация могла использоваться и в предварительной редакции.
- Поскольку не допускается изменение нумерации требований, не предполагается последовательная нумерация требований в тексте спецификации.
- Метки могут использоваться для установления логических разделов нумерации. Например, для определения "B" как логической составляющей "A" рекомендуется использовать стиль "A_B". Вместе с тем могут использоваться и другие стили при условии сохранения структуры с разделителем "-" между полями нумерации требований.
- Не рекомендуется применение постфиксных или префиксных нотаций, т. е. добавление каких-либо записей перед или после поля "Number", поскольку поле "Number" не предназначено для отображения семантической информации.
- В качестве альтернативы стилю "A_B" авторы спецификации могут принять решение о присвоении группе требований диапазона номеров. Такой подход является допустимым.

Библиография

- [b-ITU-T M.1401] Рекомендация МСЭ-Т М.1401 (2006 г.), *Формализация обозначений присоединений между сетями электросвязи операторов.*
- [b-ITU-T M.1403] Рекомендация МСЭ-Т М.1403 (2007 г.), *Формализация общих заказов.*
- [b-ITU-T M.1404] Рекомендация МСЭ-Т М.1404 (2007 г.), *Формализация заказов на присоединения между сетями операторов.*
- [b-ITU-T Z.601] Рекомендация МСЭ-Т Z.601 (2007 г.), *Архитектура данных одной системы программного обеспечения.*
- [b-3GPP TS 23.002] 3GPP TS 23.002 (in force), *Network architecture.*
- [b-3GPP TS 32.101] 3GPP TS 32.101 V10.0.0 (2010), *Telecommunication management; Principles and high level requirements.*
- [b-3GPP TS 32.150] 3GPP TS 32.150 V10.2.0 (2011), *Telecommunication management; Integration Reference Point (IRP) Concept and definitions.*
- [b-3GPP TS 32.151] 3GPP TS 32.151 V10.1.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Information Service (IS) template.*
- [b-3GPP TS 32.152] 3GPP TS 32.152 V10.0.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire.*
- [b-3GPP TS 32.302] 3GPP TS 32.302 V10.0.0 (2010), *Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP); Information Service (IS).*

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