МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МККТТ

МЕЖДУНАРОДНЫЙ
КОНСУЛЬТАТИВНЫЙ КОМИТЕТ
ПО ТЕЛЕГРАФИИ И ТЕЛЕФОНИИ

СИНЯЯ КНИГА

ТОМ VIII – ВЫПУСК VIII.8

СЕТИ ПЕРЕДАЧИ ДАННЫХ
СПРАВОЧНАЯ СЛУЖБА

РЕКОМЕНДАЦИИ X.500–X.521

IX ПЛЕНАРНАЯ АССАМБЛЕЯ
МЕЛЬБУРН, 14 – 25 НОЯБРЯ 1988 ГОДА

МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МККТТ

МЕЖДУНАРОДНЫЙ
КОНСУЛЬТАТИВНЫЙ КОМИТЕТ
ПО ТЕЛЕГРАФИИ И ТЕЛЕФОНИИ

СИНЯЯ КНИГА

ТОМ VIII – ВЫПУСК VIII.8

СЕТИ ПЕРЕДАЧИ ДАННЫХ
СПРАВОЧНАЯ СЛУЖБА

РЕКОМЕНДАЦИИ X.500–X.521

IX ПЛЕНАРНАЯ АССАМБЛЕЯ
МЕЛЬБУРН, 14 – 25 НОЯБРЯ 1988 ГОДА

ISBN 92-61-03734-8

СОДЕРЖАНИЕ КНИГИ МККТТ, ДЕЙСТВУЮЩЕЙ ПОСЛЕ IX ПЛЕНАРНОЙ АССАМБЛЕИ (1988 г.)

СИНЯЯ КНИГА

ВЫПУСК III.6 — Передача по линии нетелефонных сигналов. Передача сигналов звукового и телевизионного вещания. Рекомендации серий H и J (Исследовательская комиссия XV).

ВЫПУСК III.7 — Цифровая сеть с интеграцией служб (ЦСИС) — Общая структура и возможности служб. Рекомендации I.110—I.257 (Исследовательская комиссия XVIII).

ВЫПУСК III.8 — Цифровая сеть с интеграцией служб (ЦСИС) — Общесетевые аспекты и функции, стыки пользователь—сеть ЦСИС. Рекомендации I.310—I.470 (Исследовательская комиссия XVIII).

ВЫПУСК III.9 — Цифровая сеть с интеграцией служб (ЦСИС) — Межсетевые стыки и принципы технической эксплуатации. Рекомендации I.500—I.605 (Исследовательская комиссия XVIII).

Том IV

ВЫПУСК IV.1 — Общие принципы технической эксплуатации; техническая эксплуатация международных систем передачи и международных телефонных каналов. Рекомендации M.10—M.782 (Исследовательская комиссия IV).

ВЫПУСК IV.2 — Техническая эксплуатация международных телеграфных, фототелеграфных и арендованных каналов. Техническая эксплуатация международной телефонной сети общего пользования. Техническая эксплуатация морских спутниковых систем и систем передачи данных. Рекомендации M.800—M.1375 (Исследовательская комиссия IV).

ВЫПУСК IV.3 — Техническая эксплуатация международных каналов звукового и телевизионного вещания. Рекомендации серии N (Исследовательская комиссия IV).

ВЫПУСК IV.4 — Требования к измерительному оборудованию. Рекомендации серии O (Исследовательская комиссия IV).

Том V — Качество телефонной передачи. Рекомендации серии P (Исследовательская комиссия XII).

Том VI

ВЫПУСК VI.1 — Общие Рекомендации по телефонной коммутации и сигнализации. Функции и информационные потоки для служб в ЦСИС. Дополнения. Рекомендации Q.1—Q.118 bis (Исследовательская комиссия XI).

ВЫПУСК VI.2 — Требования к системам сигнализации № 4 и № 5. Рекомендации Q.120—Q.180 (Исследовательская комиссия XI).

ВЫПУСК VI.3 — Требования к системе сигнализации № 6. Рекомендации Q.251—Q.300 (Исследовательская комиссия XI).

ВЫПУСК VI.4 — Требования к системам сигнализации R1 и R2. Рекомендации Q.310—Q.490 (Исследовательская комиссия XI).

ВЫПУСК VI.5 — Цифровые местные, транзитные, комбинированные и международные станции в интегральных цифровых сетях и смешанных аналого-цифровых сетях. Дополнения. Рекомендации Q.500—Q.554 (Исследовательская комиссия XI).

ВЫПУСК VI.6 — Взаимодействие систем сигнализации. Рекомендации Q.601—Q.699 (Исследовательская комиссия XI).

ВЫПУСК VI.7 — Требования к системе сигнализации № 7. Рекомендации Q.700—Q.716 (Исследовательская комиссия XI).

ВЫПУСК VI.8 — Требования к системе сигнализации № 7. Рекомендации Q.721—Q.766 (Исследовательская комиссия XI).

ВЫПУСК VI.9 — Требования к системе сигнализации № 7. Рекомендации Q.771—Q.795 (Исследовательская комиссия XI).

ВЫПУСК VI.10 — Цифровая абонентская система сигнализации № 1 (ЦАС 1), уровень звена данных. Рекомендации Q.920 и Q.921 (Исследовательская комиссия XI).

ВЫПУСК VI.11 — Цифровая абонентская система сигнализации № 1 (ЦАС 1), сетевой уровень, управление пользователь—сеть. Рекомендации Q.930—Q.940 (Исследовательская комиссия XI).

ВЫПУСК VI.12 — Сухопутная подвижная сеть общего пользования. Взаимодействие с ЦСИС и коммутируемой телефонной сетью общего пользования. Рекомендации Q.1000—Q.1032 (Исследовательская комиссия XI).

ВЫПУСК VI.13 — Сухопутная подвижная сеть общего пользования. Подсистема подвижного применения и стыки. Рекомендации Q.1051—Q.1063 (Исследовательская комиссия XI).

ВЫПУСК VI.14 — Взаимодействие со спутниковыми подвижными системами. Рекомендации Q.1100—Q.1152 (Исследовательская комиссия XI).

Том VII

ВЫПУСК VII.1 — Телеграфная передача. Рекомендации серии R. Оконечное оборудование телеграфных служб. Рекомендации серии S (Исследовательская комиссия IX).

ВЫПУСК VII.2 — Телеграфная коммутация. Рекомендации серии U (Исследовательская комиссия IX).

ВЫПУСК VII.3 — Оконечное оборудование и протоколы для телематических служб. Рекомендации T.0—T.63 (Исследовательская комиссия VIII).

ВЫПУСК VII.4 — Процедуры испытания на соответствие Рекомендациям по службе телетекс. Рекомендация T.64 (Исследовательская комиссия VIII).

ВЫПУСК VII.5 — Оконечное оборудование и протоколы для телематических служб. Рекомендации T.65—T.101, T.150—T.390 (Исследовательская комиссия VIII).

ВЫПУСК VII.6 — Оконечное оборудование и протоколы для телематических служб. Рекомендации T.400—T.418 (Исследовательская комиссия VIII).

ВЫПУСК VII.7 — Оконечное оборудование и протоколы для телематических служб. Рекомендации T.431—T.564 (Исследовательская комиссия VIII).

Том VIII

ВЫПУСК VIII.1 — Передача данных по телефонной сети. Рекомендации серии V (Исследовательская комиссия XVII).

ВЫПУСК VIII.2 — Сети передачи данных: службы и услуги, стыки. Рекомендации X.1—X.32 (Исследовательская комиссия VII).

ВЫПУСК VIII.3 — Сети передачи данных: передача, сигнализация и коммутация, сетевые аспекты, техническая эксплуатация и административные положения. Рекомендации X.40—X.181 (Исследовательская комиссия VII).

ВЫПУСК VIII.4 — Сети передачи данных: взаимосвязь открытых систем (ВОС) — Модель и система обозначений, определение служб. Рекомендации X.200—X.219 (Исследовательская комиссия VII).

ВЫПУСК VIII.5 — Сети передачи данных: взаимосвязь открытых систем (ВОС) — Требования к протоколам, аттестационные испытания. Рекомендации X.220—X.290 (Исследовательская комиссия VII).

ВЫПУСК VIII.6 — Сети передачи данных: взаимодействие между сетями, подвижные системы передачи данных, межсетевое управление. Рекомендации X.300—X.370 (Исследовательская комиссия VII).

ВЫПУСК VIII.7 — Сети передачи данных: системы обработки сообщений. Рекомендации X.400—X.420 (Исследовательская комиссия VII).

ВЫПУСК VIII.8 — Сети передачи данных: справочная служба. Рекомендации X.500—X.521 (Исследовательская комиссия VII).

Том IX — Защита от мешающих влияний. Рекомендации серий K (Исследовательская комиссия V). Конструкция, прокладка и защита кабелей и других элементов линейных сооружений. Рекомендации серии L (Исследовательская комиссия VI).

Том X

ВЫПУСК X.1 — Язык функциональных спецификации и описания (SDL). Критерии применения формальных методов описания (FDT). Рекомендация Z.100 и приложения A, B, C и E, Рекомендация Z.110 (Исследовательская комиссия X).

ВЫПУСК X.2 — Приложение D к Рекомендации Z.100: руководство для пользователей языка SDL (Исследовательская комиссия X).

ВЫПУСК X.3 — Приложение F.1 к Рекомендации Z.100: формальное определение языка SDL. Введение (Исследовательская комиссия X).

ВЫПУСК X.4 — Приложение F.2 к Рекомендации Z.100: формальное определение языка SDL. Статическая семантика (Исследовательская комиссия X).

ВЫПУСК X.5 — Приложение F.3 к Рекомендации Z.100: формальное определение языка SDL. Динамическая семантика (Исследовательская комиссия X).

ВЫПУСК X.6 — Язык МККТТ высокого уровня (CHILL). Рекомендация Z.200 (Исследовательская комиссия X).

ВЫПУСК X.7 — Язык человек—машина (MML). Рекомендации Z.301—Z.341 (Исследовательская комиссия X).

СОДЕРЖАНИЕ ВЫПУСКА VIII.8 СИНЕЙ КНИГИ

ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

1 Вопросы, порученные каждой Исследовательской комиссии на исследовательский период 1989—1992 годов, содержатся во Вкладе № 1 для данной Исследовательской комиссии.

2 В данном выпуске для краткости термин "Администрация" используется для обозначения как Администрации связи, так и признанной частной эксплуатационной организации.

3 Если не оговорено иное, то статус обязательных приложений (Annex) и необязательных приложений (Appendix) к Рекомендациям серии X следует трактовать следующим образом:

— *обязательное приложение* является неотъемлемой частью данной Рекомендации;

— *необязательное приложение* не является неотъемлемой частью данной Рекомендации и представляет только некоторые дополнительные пояснения или информацию, специфичную для данной Рекомендации.

4 Рекомендации Серии X, содержащиеся в настоящем выпуске, были разработаны совместно в сотрудничестве с МОС/МЭК (Международной электротехнической комиссией). Перекрестные ссылки между этими Рекомендациями и соответствующими стандартами МОС/МЭК приведены в нижеследующей таблице.

Рекомендация МККТТ	Стандарт МОС/МЭК
X.500	ISO 9594-1, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть I: Обзор концепций, моделей и служб[a)]
X.501	ISO 9594-2, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 2: Модели[a)]
X.509	ISO 9594-8, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 8: Структура аутентификации[a)]
X.511	ISO 9594-3, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 3: Определение абстрактной службы[a)]
X.518	ISO 9594-4, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 4: Процедуры распределенных операций[a)]
X.519	ISO 9594-5, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 5: Спецификация протокола[a)]
X.520	ISO 9594-6, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 6: Избранные типы атрибутов[a)]
X.521	ISO 9594-7, Системы обработки информации – Взаимосвязь открытых систем – Справочник – Часть 7: Избранные классы объектов[a)]

a) В настоящее время в стадии проекта Международного стандарта (ПМС).

ВЫПУСК VIII.8

РЕКОМЕНДАЦИИ X.500—X.521

СЕТИ ПЕРЕДАЧИ ДАННЫХ: СПРАВОЧНИК

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

СПРАВОЧНИК — ОБЗОР КОНЦЕПЦИЙ, МОДЕЛЕЙ И СЛУЖБ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

---

1) Рекомендация X.500 и ISO 9594-1, "Справочник – Обзор концепций моделей и служб", были разработаны в тесном сотрудничестве и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой ими справочной информацией может рассматриваться как объединенное целое, называемое *Справочником*. Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные процессы, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем (ВОС); его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых разными производителями:

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 В настоящей Рекомендации вводятся и моделируются концепции Справочника и ИБС, а также приводится обзор обеспечиваемых ими служб и возможностей. Другие Рекомендации используют эти модели для абстрактного определения службы, обеспечиваемой Справочником, а также для спецификации протоколов, обеспечивающих доступ к этим службам и их распространение.

1 Предмет рассмотрения и область применения

1.1 Справочник поставляет справочные возможности, требуемые приложениям ВОС, управляющим процессам ВОС, элементам прочих уровней ВОС, и службам телекоммуникации. В числе возможностей, поставляемых Справочником, имеется возможность "удобного для пользователя именования", позволяющая обращаться к объектам по именам, удобным для пользователя-человека (хотя не все объекты должны иметь такие имена); имеется также возможность "отображения имени-на-адрес", обеспечивающая динамическую привязку объектов к их местоположению. Последнее свойство позволяет, например, сетям ВОС быть "самоконфигурируемыми" в том смысле, что добавление, исключение или изменение местоположения объекта не оказывает влияния на функционирование сети.

1.2 Справочник не рассчитан быть общецелевой системой базы данных, хотя он и может быть создан на основе таких систем. Предполагается, в частности, что, как это обычно имеет место со справочниками связи, обращение за справками к Справочнику будет происходить значительно чаще, чем обновление его данных. Предполагается, что уровень обновлений будет зависеть от динамичности людей и организаций, а не от, например, динамичности сетей. Нет также нужды в немедленном полном осуществлении обновлений: вполне допустимо переходное состояние, когда доступны одновременно старая и новая версии одной и той же информации.

1.3 Справочник обладает тем свойством, что результаты обращения к нему не зависят от того, кто обращался к нему за справкой и где запрашивающий в этот момент находился; исключение составляет тот случай, когда пользователи имеют различные права доступа или когда в Справочнике предусмотрено ограничение на распространение обновлений. Это свойство делает Справочник не совсем подходящим для некоторых телекоммуникационных приложений, например для некоторых типов маршрутизации.

2 Библиография

Рекомендация X.200 "Взаимосвязь открытых систем — Базовая эталонная модель".

Рекомендация X.208 "Взаимосвязь открытых систем — Спецификация нотации абстрактного синтаксиса номер один (НАС.1)".

Рекомендация X.501 "Справочник — Модели".

Рекомендация X.509 "Справочник — Структура аутентификации".

Рекомендация X.511 "Справочник — Абстрактное определение служб".

Рекомендация X.518 "Справочник — Процедуры распределенных операций".

Рекомендация X.519 "Справочник — Спецификация протоколов".

Рекомендация X.520 "Справочник — Избранные типы атрибутов".

Рекомендация X.521 "Справочник — Избранные классы объектов".

Рекомендация X.219 "Удаленные операции — Модель, нотация и определение услуг".

Рекомендация X.229 "Удаленные операции — Спецификация протокола".

3 Определения

Определения, приводимые в этом параграфе, используют сокращения, определенные в § 4.

3.1 *Определения эталонной модели ВОС*

Настоящая Рекомендация опирается на концепции, развитые в Рекомендации X.200, и использует определенные в ней нижеследующие термины:

a) *элемент прикладного уровня;*

b) *прикладной уровень;*

c) *прикладной процесс;*

d) *блок данных прикладного протокола;*

e) *элемент прикладной службы.*

3.2 *Основные определения Справочника*

a) *Справочник:* совокупность открытых систем, кооперирующихся в целях предоставления справочных услуг;

b) *информационная база Справочника (ИБС):* совокупная информация, управляемая Справочником;

c) *пользователь (Справочника)*: оконечный пользователь Справочника, то есть элемент или человек, имеющий доступ к Справочнику.

3.3 *Определения модели Справочника*

Настоящая Рекомендация использует определенные в Рекомендации X.501 нижеследующие термины:

a) *административная область управления Справочником;*

b) *псевдоним;*

c) *атрибут;*

d) *тип атрибута;*

e) *значение атрибута;*

f) *информационное дерево Справочника (ИДС);*

g) *область управления Справочником (ОУС);*

h) *системный агент Справочника (САС);*

i) *агент пользователя Справочника (АПС);*

j) *выделенное имя;*

k) *статья;*

l) *имя;*

m) *объект (представляющий интерес)*;

n) *частная область управления Справочником;*

o) *относительно выделенное имя;*

p) *корень;*

q) *схема;*

r) *последующий объект;*

s) *предшествующая статья;*

t) *предшествующий объект;*

u) *дерево.*

3.4 *Определения распределенной операции*

Настоящая Рекомендация использует определенные в Рекомендации X.519 нижеследующие термины:

a) *сцепление;*

b) *многоадресная рассылка;*

c) *отсылка.*

4 Сокращения

АОУС — административная область управления Справочником

ПДС — протокол доступа к Справочнику

ИБС — информационная база Справочника

ИДС — информационное дерево Справочника

ОУС — область управления Справочником

САС — системный агент Справочника

СПС — системный протокол Справочника

АПС — агент пользователя Справочника

ВОС — взаимосвязь открытых систем

ЧОУС — частная область управления Справочником

ОВИ — относительно выделенное имя

5 Общее описание Справочника

5.1 *Справочник* является совокупностью открытых систем, кооперирующихся в целях хранения логической базы данных, содержащей информацию о множестве объектов реального мира. *Пользователи* Справочника, включая людей и программы, могут читать или модифицировать информацию или некоторые ее части при условии, что они обладают соответствующим правом. При обращениях к Справочнику каждый пользователь представлен агентом пользователя Справочника (АПС), который рассматривается как прикладной процесс. Эти концепции проиллюстрированы на рис. 1/X.500.

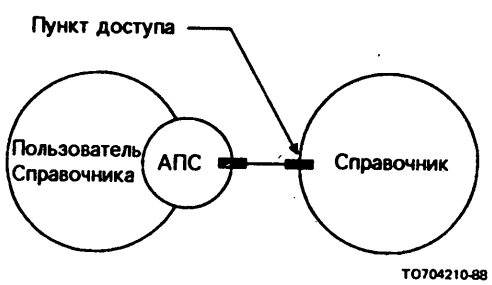


РИСУНОК 1/X.500

Доступ к Справочнику

Примечание. — В настоящей серии Рекомендаций термин *Справочник* используется в единственном числе, что отражает намерение создать, используя единственное, унифицированное, поименованное пространство, один единственный логический справочник, состоящий из многих систем и обслуживающий много приложений. Захотят ли эти системы взаимодействовать или нет, это будет зависеть от нужд тех приложений, которые они обеспечивают. Приложения, касающиеся не пересекающихся между собой миров и объектов, могут такой нужды не испытывать. Единое название пространства облегчает в дальнейшем взаимодействие, если изменяются потребности.

5.2 Информация, хранящаяся в Справочнике, в своей совокупности называется *информационной базой Справочника* (ИБС). В пункте 6 настоящей Рекомендации дается обзор ее структуры.

5.3 Справочник обеспечивает своим пользователям совокупность четко очерченных возможностей доступа; эта совокупность называется абстрактной службой Справочника. Эта служба, описанная в § 7 настоящей Рекомендации, обеспечивает средства простой модификации и извлечения информации. Это достигается с помощью функций локального АПС; тем самым обеспечиваются возможности, в которых нуждаются оконечные пользователи.

5.4 Весьма вероятно, что Справочник будет распределенным, и при том весьма широко, как в аспекте его функционирования, так и в аспекте его организации. В § 8 рассматриваются соответствующие модели Справочника. Эти модели были разработаны с целью обеспечения рамок кооперирования различных компонент для создания единого целого.

5.5 Обеспечение и использование служб Справочника требуют, чтобы пользователи (а фактически АПС) и различные функциональные компоненты Справочника кооперировались друг с другом. Во многих случаях это потребует кооперирования прикладных процессов в различных открытых системах. Это, в свою очередь, требует наличия стандартных протоколов приложений, рассматриваемых в § 9. Назначение протоколов — руководство кооперированием.

5.6 Справочник был спроектирован таким образом, чтобы обеспечить большое число приложений, извлеченных из широкого диапазона возможностей. Характер обеспечиваемого приложения обусловливает состав объектов, сведенных в Справочнике, состав пользователей, имеющих доступ к информации, и какими видами доступа они будут пользоваться. Приложения могут быть весьма специфичными, такими как представление списков рассылки для электронной почты, или, наоборот, достаточно общими, как, например, приложение "справочник по межперсональным связям". Справочник позволяет использовать общие свойства приложений:

— один отдельный объект может касаться более чем одного приложения; возможно даже, что одна и та же порция информации об одном и том же объекте будет касаться более чем одного приложения.

Для обеспечения этого определяется некоторое число таких классов объектов и таких типов атрибутов, которые будут полезными для целого ряда приложений из некоторого диапазона. Эти определения содержатся в Рекомендациях X.520 и X.521:

— некоторое число шаблонов использования Справочника будут общими для какого-то диапазона приложений; этот вопрос рассматривается ниже,в Приложении А.

6 Информационная база Справочника (ИБС)

Примечание. — ИБС и ее структура описаны в Рекомендации X.501.

6.1 ИБС составлена из информации об объектах. Она включает *статьи* (*Справочника*), каждая из которых состоит из совокупности информации об одном объекте. Каждая статья составлена из *атрибутов,* каждый со своим типом и одним или более значениями. Типы атрибутов, содержащиеся в некоторой конкретной статье, зависят от *класса* того объекта, который представлен этой статьей.

6.2 Статьи ИБС расположены в виде дерева, информационного дерева Справочника (ИДС), в котором вершины изображают статьи. Статьи, расположенные выше (ближе к корню), будут зачастую изображать такие объекты, как страны или организации, в то время как вершины, расположенные в дереве ниже, будут изображать людей или прикладные процессы.

Примечание. — Службы, определенные в настоящей Рекомендации, оперируют только с древовидной информацией ИДС. Настоящая Рекомендация не исключает в дальнейшем (по мере появления необходимости) наличия информации другой структуры.

6.3 Каждая статья имеет *выделенное имя,* которое однозначно и недвусмысленно идентифицирует статью. Это свойство выделенных имен является следствием древовидной структуры информации. Выделенное имя статьи составлено из выделенного имени статьи, предшествующей данной, в сочетании со специальным способом обозначенными значениями атрибутов (*выделенными* значениями) из данной статьи.

6.4 Некоторые статьи дерева, являющиеся листьями, представляют из себя статьи псевдонимов, в то время как все остальные статьи являются статьями объектов. Статьи псевдонимов указывают на статьи объектов и являются основой альтернативных имен соответствующих объектов.

6.5 Справочник обусловливает набор правил, обеспечивающих устойчивость ИБС перед лицом возникающих во времени модификаций. Эти правила, известные как *схема Справочника,* предохраняют статьи от возникновения типов атрибутов, не соответствующих тому классу, которому принадлежит объект; возникновения значений, не соответствующих типу атрибута; и даже возникновения у данной статьи последующих статей, принадлежащих недопустимым классам.

6.6 Рис. 2/X.500 иллюстрирует вышеописанные концепции ИДС и его компонентов.

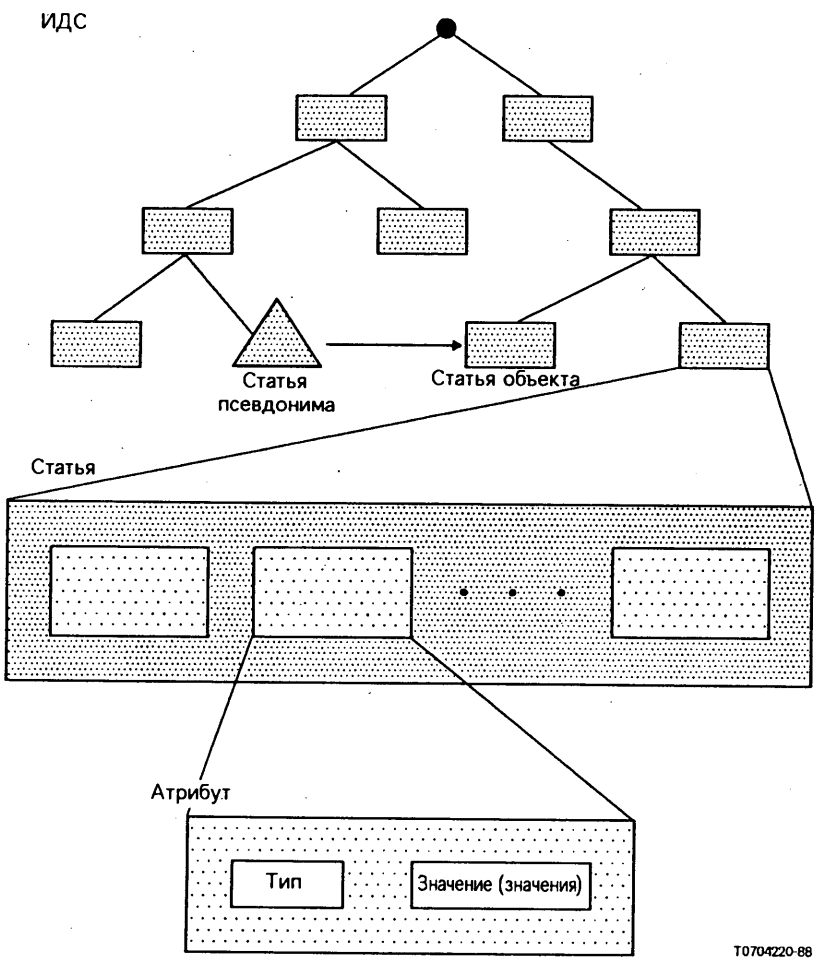


РИСУНОК 2/X.500

Структура ИДС и его статей

6.7 На рис. 3/X.500 приведен пример гипотетического ИДС. В этом дереве приведены примеры типов атрибутов, используемых для идентификации различных объектов. Например, имя:

{С = ВБ, М = Уинслоу, О = Графические службы, ОИ = Лазерный принтер},

идентифицирует прикладной элемент "Лазерный принтер", выделенное имя которого содержит географический атрибут "местность". Джон Джонс, проживающий в этом же районе, имя которого ВБ

{С = ВБ, М = Уинслоу, ОИ = Джон Джонс},

содержит тот же географический атрибут в своем выделенном имени.

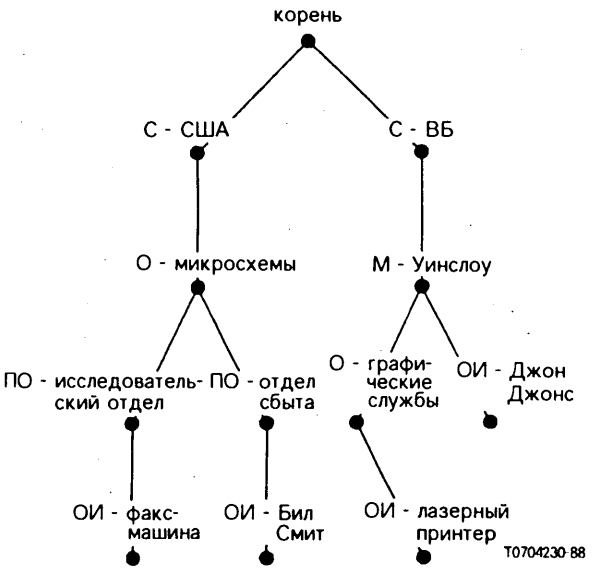


РИСУНОК 3/X.500

Гипотетическое информационное дерево Справочника

6.8 Рост и структура ИДС, определение схемы Справочника и выбор выделенных имен статей по мере их добавления — все это относится к сфере компетенции различных руководящих органов, иерархические взаимоотношения которых отражаются в очертаниях дерева Эти органы должны обеспечить, например, чтобы все статьи, находящиеся под их юрисдикцией, обладали недвусмысленными выделенными именами; для этого они должны с большой осторожностью обращаться с типами атрибутов и значениями, входящими в эти имена. Ответственность передается вниз по дереву от старших руководящих органов к подчиненным; управление этим осуществляется с помощью схемы Справочника.

7 Служба, обеспечиваемая Справочником .

Примечание. — Определение абстрактной службы можно найти в Рекомендации X.511.

7.1 *Введение*

7.1.1 В настоящем параграфе дается общее описание службы, обеспечиваемой Справочником своим пользователям; при этом сами пользователи представлены своими АПС. Все службы обеспечиваются Справочником только в ответ на запросы АПС. Существуют запросы, которые позволяют обращаться только за справками, как это описано в § 7.3, и такие, которые требуют модификации, как это описано в § 7.4. Кроме того, запросы могут быть квалифицированы, как это описано в § 7.2. Справочник всегда выдает извещение в ответ на поступивший к нему запрос. Формат нормального ответа специфичен для каждого вопроса и очевиден из описания запроса. Большинство ненормальных ответов является общим для нескольких запросов. Возможные варианты описаны в § 7.5.

7.1.2 Некоторое число аспектов эвентуальной справочной службы в настоящее время не обеспечиваются стандартами, специфицированными в настоящей серии Рекомендаций. Поэтому соответствующие средства должны быть обеспечены в порядке локальных функций до тех пор, пока не станут доступны стандартные решения. В эти возможности входят:

- добавление и удаление произвольных статей, что обеспечит возможность создания Справочника;
- право управления доступом (то есть право предоставлять конкретному пользователю возможность выполнять конкретный вид доступа к конкретной порции информации, или лишение его таких возможностей);
- управление схемой Справочника;
- управление информацией о знаниях;
- выполнение копий частей ИДС.

Примечание. — Этот список может не быть исчерпывающим.

7.1.3 Справочник обеспечивает следующее свойство: изменения в ИБС, связанные как с запросами к службам Справочника, так и с какими-то другими (локальными) средствами, приводят к ИБС, которая продолжает удовлетворять правилам схемы Справочника.

7.1.4 Пользователь и Справочник привязаны друг к другу на некоторый период времени в пункте доступа к Справочнику. В момент привязывания Справочник и пользователь могут (но не обязательно) проверить подлинность один другого.

7.2 *Квалификация службы*

7.2.1 *Ограничения службы*

На некоторые запросы могут быть наложены ограничения, главным образом для того, чтобы пользователь мог оговорить такие условия использования ресурсов, которые Справочник не вправе нарушить. Такие ограничения, среди прочих, накладываются на время получения ответа, размеры результата, область поиска, режимы взаимодействия и приоритетность запросов.

7.2.2 *Параметры безопасности*

Каждый запрос может сопровождаться информацией, обеспечивающей механизмы безопасности, для защиты информации Справочника. Эта защитная информация может содержать запрос пользователя на различные виды защиты; цифровую подпись под запросом в сочетании с информацией, которая поможет другой стороне проверить подпись, если эта сторона имеет санкцию на такую проверку.

7.2.3 *Фильтры*

Некоторые запросы, ответ на которые включают информацию из нескольких статей или затрагивают несколько статей, могут нести с собой фильтры. Фильтры выражают одно или несколько условий, которым должна удовлетворять статья, чтобы быть возвращенной в качестве части ответа. Это позволяет включить в возвращаемый ответ только те статьи, которые относятся к этому ответу.

7.3 *Обращения к Справочнику*

7.3.1 *Чтение*

Запрос на чтение относится только к конкретной статье и вызывает возврат значений всех или некоторых атрибутов статьи. Если требуется выдать только некоторые из атрибутов, то АПС предоставляет список типов интересующих атрибутов.

7.3.2 *Сравнение*

Запрос на сравнение относится только к конкретному атрибуту конкретной статьи и вызывает Справочник проверить, соответствует ли предоставленное значение значению этого атрибута.

Примечание. — Это может быть использовано, например, для проверки пароля, при которой пароль, хранящийся в Справочнике, может быть недоступен для чтения, но доступен для сравнения.

7.3.3 *Список*

Запрос на список побуждает Справочник выдать список всех статей, непосредственно следующих (в ИДС) за некоторой статьей, имя которой указано в запросе.

7.3.4 *Поиск*

Запрос на поиск побуждает Справочник выдать информацию из всех статей некоторой порции ИДС, удовлетворяющих некоторому фильтру. Информация, выданная из каждой статьи, состоит из некоторых или всех атрибутов этой статьи, как это имеет место при чтении.

7.3.5 *Отказ*

Запрос на "отказ" применительно к предшествующему обращению сообщает Справочнику, что автора запроса более не интересует выполняемый запрос. Справочник может например прекратить обработку запроса и аннулировать уже полученные результаты.

7.4 *Модификация Справочника*

7.4.1 *Добавление статьи*

В результате запроса на добавление статьи в ИДС вводится новая статья (как статья объекта, так и статья псевдонима), являющаяся листом дерева.

Примечание. — В своей настоящей форме эта служба рассчитана на добавление листьев, которые таковыми пребудут, как, например, статьи о людях или прикладных элементах, а не на добавление целого поддерева за счет многократного применения этой службы. Предвидится расширение этой службы в дальнейшем, с тем чтобы она удовлетворяла более общему случаю.

7.4.2 *Удаление статьи*

Запрос на удаление статьи вызывает удаление из ИДС статьи, являющейся листом дерева.

Примечание. — Так же как и для случая "добавление статьи" эта служба в своей настоящей форме рассчитана на удаление "подлинных" листьев и будет в дальнейшем расширена, с тем чтобы она удовлетворяла более общему случаю.

7.4.3 *Модификация статьи*

Запрос на модификацию статьи вызывает Справочник выполнить целый ряд изменений в конкретной статье. Независимо от того, выполнены ли все изменения или ни одно из них, ИБС всегда остается в состоянии, соответствующем схеме. Допустимые изменения включают добавление, удаление или замену атрибутов или значений атрибутов.

7.4.4 *Модификация относительно выделенного имени*

Запрос на модификацию относительно выделенного имени (ОВИ) вызывает модификацию относительно выделенного имени статьи, являющейся листом дерева (как статьи объекта, так и статьи псевдонима), за счет назначения других выделенных значений атрибутов.

7.5 *Прочие ответы*

7.5.1 *Ошибки*

Услуга может оказаться невыполненной, например в связи с трудностями, возникшими в отношении поставленных пользователем параметров; в этом случае поступает извещение об ошибке. Ответ возвращается вместе с ошибкой, чтобы, по возможности, способствовать исправлению ошибки. Однако, как правило, выдается извещение только о первой ошибке, обнаруженной Справочником. Помимо указанной выше ошибки, вызванной трудностями в представленных пользователем параметрах (в особенности, недействительных имен статей или недействительных типов атрибутов), ошибки могут быть связаны с нарушением безопасности, правил, описывающих схему, и требований на управление службами.

7.5.2 *Отсылки*

Услуга может оказаться невыполненной ввиду того, что пункт доступа, к которому привязан АПС, не является достаточно подходящим для выполнения запроса. Эта ситуация может возникнуть в силу, например, того, что требующаяся в связи с запросом информация (логически) расположена на значительном расстоянии от пункта доступа. В этом случае Справочник может выдать отсылку, указывающую альтернативный пункт доступа, в котором АПС может повторить свой запрос.

Примечание. — Справочник, так же как и АПС, может предпочесть один из двух методов: отсылку или *сцепление* запросов (см. § 8.3.3.2). АПС может высказать свое мнение с помощью параметров службы. Справочник принимает окончательное решение по поводу того, какой из двух способов ему предпочесть.

8 Распределенный Справочник

Примечание. — Модели Справочника определены в Рекомендации X.501, а процедуры выполнения операций распределенным Справочником специфицированы в Рекомендации X.518.

8.1 *Функциональная модель*

Функциональная модель Справочника изображена на рис. 4/X.500.

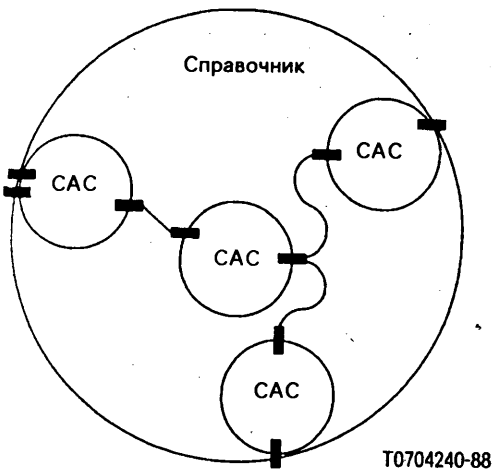


РИСУНОК 4/X.500

Функциональная модель Справочника

Системный агент Справочника (САС) является прикладным процессом ВОС, составляющим часть Справочника. Его назначение состоит в том, чтобы обеспечить доступ к ИБС как для АПС, так и для других САС. Для выполнения запроса САС может использовать информацию, хранящуюся в его локальной базе данных, или же вступить во взаимодействие с другими САС. Альтернативно САС может направить запрашивающего к другому САС, который поможет выполнить запрос. Локальные базы данных полностью зависят от конкретной реализации.

8.2 *Организационная модель*

8.2.1 Совокупность одного или более САС и ноль или более АПС, руководимая одной какой-то организацией, образует область управления Справочником (ОУС). Рассматриваемая Организация может решить, будет или не будет она руководствоваться настоящей серией Рекомендаций для обеспечения связи между функциональными компонентами внутри ОУС.

8.2.2 Последующие Рекомендации специфицируют некоторые аспекты поведения САС. В этих целях некоторая группа САС, входящая в один какой-то ОУС, может, по усмотрению организации, управляющей данным ОУС, вести себя как один САС.

8.2.3 ОУС может быть либо административным ОУС (АОУС), либо частным ОУС (ЧОУС) в зависимости от того, управляется ли он или нет телекоммуникационной организацией общего пользования.

Примечание. — Необходимо учитывать, что вопрос о поддержке частных справочных систем членами МККТТ зависит от характера национальных соглашений. В частности, описанные технические возможности могут предоставляться или не предоставляться Администрацией, обеспечивающей справочную службу. Внутреннее функционирование и конфигурация частных ОУС не входят в предмет рассмотрения Рекомендаций МККТТ.

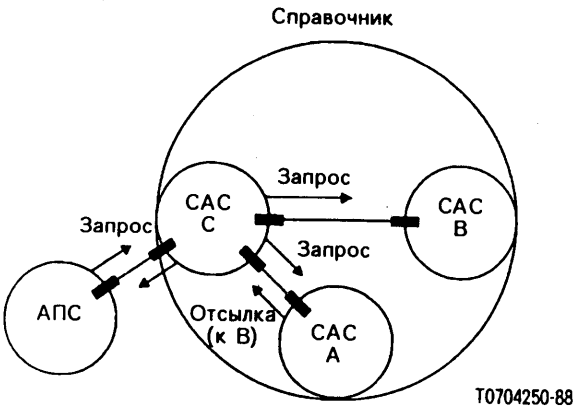
8.3 *Функционирование модели*

8.3.1 Чтобы взаимодействовать со Справочником, АПС вступает в контакт с одним или несколькими САС. Нет необходимости в привязывании АПС к какому-то одному конкретному САС. Для выдачи своих запросов АПС может взаимодействовать напрямую с различными САС. Может оказаться, что в силу каких-то организационных причин АПС будет лишен возможности вступить в непосредственный контакт с тем САС, который должен выполнить запрос, например выдать некоторую справочную информацию. Возможна также ситуация, при которой АПС будет иметь доступ к Справочнику посредством одного САС. Поэтому необходимо обеспечить возможность взаимодействия всех САС между собой.

8.3.2 САС отвечает за обработку запросов АПС и за получение информации, если он сам необходимой информацией не владеет. Он может взять на себя ответственность за получение информации, взаимодействуя для этого от имени АПС с другими САС.

8.3.3 Ниже описывается несколько случаев обработки запросов, изображенных на рис. 5—7/X.500.

8.3.3.1 На рис. 5a/X.500 САС С получает отсылку от САС А; САС С ответственен либо за отсылку запроса к САС В (названному в отсылке от САС А), либо за возвращение отсылки обратно к исходному АПС



Примечание. – Если САС С возвращает отсылку агенту пользователя Справочника, то "запроса (к В)" не возникает. Подобно этому, если САС С передаст запрос САС В, то этот последний не станет возвращать отсылку АПС.

РИСУНОК 5a/X.500

Отсылки

На рис. 5b/X.500 АПС получает отсылки от САС С; он сам несет ответственность за передачу запроса непосредственно САС А (указанному в отсылке от САС А).

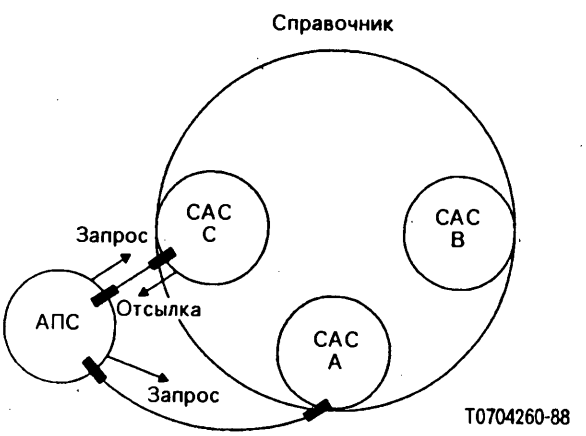


РИСУНОК 5b/X.500

Отсылки

8.3.3.2 Рис. 6/X.500 иллюстрирует сцепление, при котором запрос может передаваться сквозь несколько САС, прежде чем будет возвращен ответ.

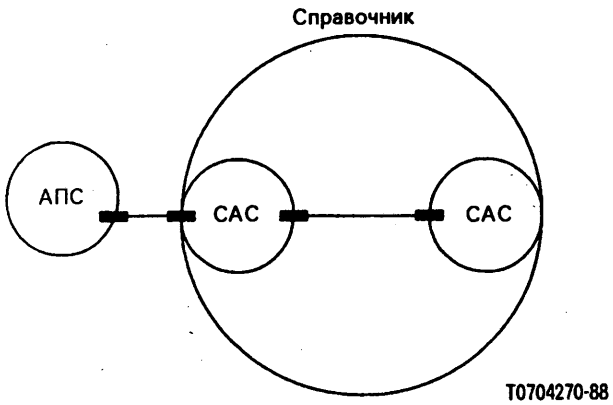


РИСУНОК 6/X.500

Сцепление

8.3.3.3 Рис. 7/X.500 иллюстрирует многоадресную рассылку, при которой САС, ассоциированный с АПС, для выполнения запроса пересылает его двум (или более) другим САС; при этом все посылаемые различным САС запросы идентичны.

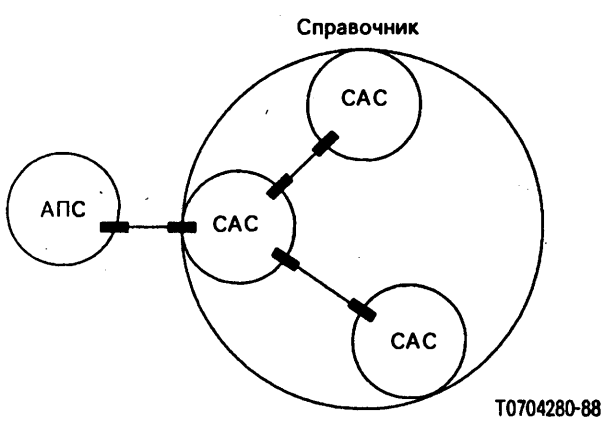


РИСУНОК 7/X.500

Многоадресная рассылка

8.3.4 Каждый из описанных подходов имеет свои достоинства. Например, подход, изображенный на рис. 5/X.500, может быть использован в тех случаях, когда желательна разгрузка местного САС. При других обстоятельствах может потребоваться некоторый гибридный подход, включающий более сложный набор функциональных взаимодействий; таким способом может быть удовлетворен исходный запрос, как это изображено на рис. 8/X.500.

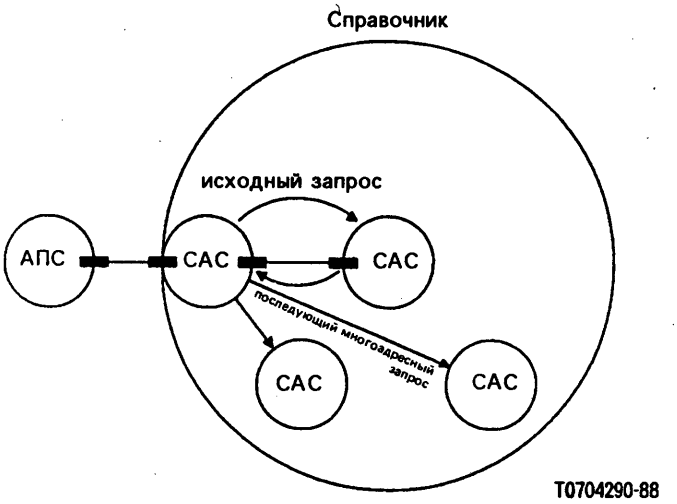


РИСУНОК 8/X.500

Комбинирование методов при гибридном подходе

9 Протоколы Справочника

Примечание. — Протоколы прикладного уровня ВОС, обеспечивающие возможность кооперирования между несколькими САС и несколькими АПС, находящимися в различных открытых системах, специфицированы в Рекомендации X.519.

9.1 Существует два протокола Справочника:

— протокол доступа к Справочнику (ПДС), в котором определен обмен запросами и ответами между АПС и САС;

— системный протокол Справочника (СПС), в котором определен обмен запросами и ответами между двумя САС.

9.2 Каждый протокол определяется как прикладной контекст, содержащий комплект элементов протокола. Например, ПДС содержит элементы протокола, связанные с обращением и модификацией Справочника.

9.3 Каждый прикладной контекст составлен из элементов прикладной службы. Эти элементы прикладной службы определены так, чтобы они могли использовать службу удаленных операций (СУО), описанную в Рекомендации X.219; эти СУО структурируют и обеспечивают возможность взаимодействия между элементами прикладной службы. Таким образом ПДС и СПС определены как совокупность удаленных операций и ошибок; для их описания используется нотация СУО.

ПРИЛОЖЕНИЕ A

(к Рекомендации X.500)

Использование Справочника

Данное Приложение не является составной частью настоящей Рекомендации.

A.1 *Среда Справочника*

Примечание. — В данном параграфе термин "сеть" используется в своем общем значении и означает совокупность взаимосвязанных систем и процессов, имеющих отношение к любой телекоммуникационной службе, а не только к сетевому уровню ВОС.

Справочник существует и обеспечивает службы в нижеследующей среде:

a) большое число телекоммуникационных сетей будут широкомасштабными и будут непрерывно подвергаться изменениям:

   1) объекты различной природы будут включаться в сеть и покидать ее без предупреждения об этом и могут поступать как в одиночку, так и группами;

   2) в связи с удалением и добавлением соединений между объектами их связанность (и, в частности, связанность вершин сети) будет меняться;

   3) некоторые характеристики объектов, такие как адреса, доступность, физическое расположение, могут меняться в любой момент времени;

b) хотя общая частота изменений в Справочнике достаточно высока, тем не менее полезный срок жизни каждого отдельного объекта довольно продолжителен; объект будет значительно чаще принимать участие в коммуникационных операциях, чем будут меняться его адрес, доступность, физическое местоположение и т.д.;

c) объекты, вовлеченные в текущие телекоммуникационные службы, индентифицируются, как правило, номерами или другими цепочками символов, которые выбираются потому, что с их помощью легко выполнять разметку и обработку, а не потому, что с ними легко работать людям.

A.2 *Характеристики службы, обеспечиваемой Справочником*

Потребность в возможностях Справочника определяется следующими обстоятельствами:

a) желанием изолировать (насколько это возможно) пользователя сети от частых изменений в Справочнике; это может быть достигнуто установлением допустимого "уровня обходных путей" между пользователями и объектами, с которыми общаются пользователи; последнее означает, в частности, что пользователи должны обращаться к объектам по именам, а не, например, по адресам; Справочник обеспечивает необходимую службу отображения одних на другие;

b) желанием предоставить взгляд на сеть, как можно более "удобный для пользователя"; например, использование псевдонимов, обеспечение "желтых листов" (см. A.3.5) и т.д. помогают пользователю в отыскании и использовании сетевой информации.

Справочник разрешает пользователям получать различную информацию о самой сети и обеспечивает поддержание, распространение и безопасность этой информации.

A.3 *Шаблоны использования Справочника*

Примечание. — В настоящем подпункте рассматривается только получение информации из Справочника: предполагается, что службы модификации Справочника используются (в течение всего времени) только для поддержания ИБС в форме, требующейся для приложений.

A.3.1 *Введение*

В настоящих стандартах службы Справочника определены в терминах конкретных запросов, которые могут поступить от АПС, и параметров, связанных с этими запросами. Между тем, разработчик некоторого приложения при рассмотрении требований к получению информации из Справочника будет, по-видимому, мыслить в терминах более проблемно-ориентированных к его приложению. Соответственно с этим в настоящем пункте рассматривается несколько шаблонов высокого уровня использования служб Справочника. Можно предположить, что эти шаблоны окажутся подходящими для многих приложений.

A.3.2 *Выборка*

Прямая выборка из Справочника будет, по-видимому, наиболее частым типом опроса Справочника. Она включает предоставление агентом пользователя Справочника выделенного имени объекта и вместе с ним типа атрибута. Справочник выдает значение(я), соответствующее(ие) этому типу. Это является обобщением классической функции Справочника, при которой запрошенный тип атрибута соответствует конкретному типу адреса. Типы атрибутов для адресов различного типа стандартизированы, включая адреса ПрПДС ВОС, телефонных и телексных номеров, а также адресов в Системе обработки сообщений.

Выборка обеспечивается службой чтения, которая, кроме того, обеспечивает нижеследующие дальнейшие обобщения:

— поиск может быть основан на именах, которые отличаются от выделенных имен, например на псевдонимах;

— в одном обращении могут быть запрошены значения нескольких атрибутов; максимальным расширением этого запроса будет запрос на выдачу значений всех атрибутов статьи.

А.3.3 *Наименование, удобное для пользователя*

Имена объектам могут подбираться таким образом, чтобы пользователь мог с наибольшей степенью вероятности предсказать (или, быть может, запомнить) эти имена. Имена, обладающие этим свойством, будут, как правило, составлены из атрибутов, в каком-то смысле присущих объектам, а не атрибутов, специально созданных для каких-то целей. Имя объекта будет одним и тем же для всех приложений, обращающихся к нему.

А.3.4 *Просмотр*

Во многих человеко-ориентированных использованиях Справочника пользователь (или АПС) не сможет выдать непосредственно имя (удобное для пользователя или нет) того объекта, информацию о котором он ищет. Вместе с тем, возможно, что пользователь "опознает имя, когда увидит его". Возможность просмотра позволит пользователю-человеку ознакомиться с ИБС, выискивая подходящие ему статьи.

Просмотр осуществляется сочетанием служб "список" и "поиск", возможно, в сочетании со службой "чтение" (хотя служба "поиска" включает и возможность "чтения").

А.3.5 *"Желтые страницы"*

Существует несколько способов обеспечения средства типа "Желтые страницы". Простейший из них основан на фильтрации, использующей утверждения относительно некоторых атрибутов, чьи значения являются категориями (например, тип атрибута "категория бизнеса", определенный в Рекомендации X.520). Этот подход не требует включения в ИДС какой-либо специальной информации, за исключением требования, чтобы в нем имелись атрибуты, входящие в данную категорию. Однако, как правило, там, где имеется большое скопление населения, фильтрация может оказаться дорогостоящим способом, так как она требует создания универсального множества, подлежащего фильтрации.

Возможен и альтернативный подход, основанный на создании специальных поддеревьев, структура именования которых подобрана специально для поиска типа "Желтые страницы". На рис. А-1/X.500 приведен пример поддеревьев "Желтые страницы", содержащих только статьи псевдонимов. На практике статьи, входящие в поддеревья "Желтых страниц", могут быть как статьями объектов, так и статьями псевдонимов, если для каждого объекта, хранящегося в Справочнике, в ИДС имеется всего одна статья объекта.

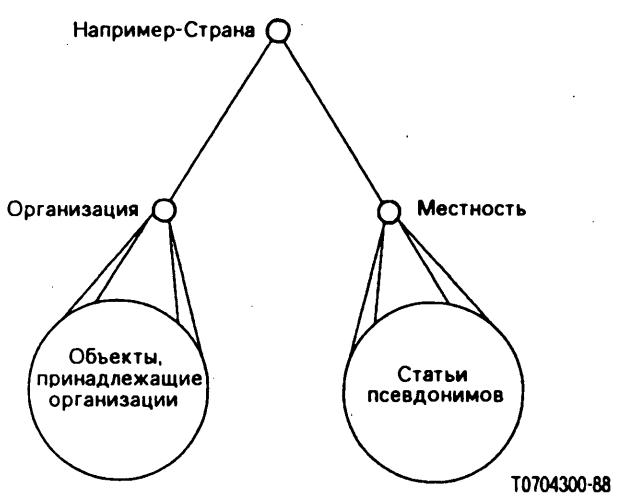


РИСУНОК А-1/X.500

Подход к "Желтым страницам"

A.3.6 *Группы*

Группой является множество, членство в котором может со временем меняться за счет явного добавления или исключения членов. Группа, так же как и ее члены, является объектом. К Справочнику можно обратиться, с тем чтобы он

— указал, является ли некоторый конкретный объект членом группы;

— перечислил членов группы.

Для обеспечения этого средства Справочник содержит для каждой группы отдельную статью, содержащую многозначный атрибут "Член" (атрибут такого типа определен в Рекомендации X. 520). Две указанные выше возможности могут быть в таком случае реализованы с помощью сравнения и чтения соответственно.

Член группы может в свою очередь быть группой, если только это имеет содержательный смысл для данного приложения. Как бы то ни было, но необходимые службы рекурсивной верификации и расширения должны быть осуществлены АПС, который может использовать для этого обеспечиваемые нерекурсивные версии.

A.3.7 *Аутентификация*

Во многих приложениях участвующие в них объекты должны представить некоторые доказательства, подтверждающие их подлинность, прежде чем они будут допущены к выполнению каких-либо действий. Справочник обеспечивает возможность такой аутентификации. (Кроме того, в порядке осуществления контроля за доступом Справочник требует от своих пользователей установления ими их подлинности.)

Самый непосредственный подход к аутентификации, называемый "простой аутентификацией", заключается в том, что Справочник обеспечивает наличие атрибута "Пароль пользователя" в каждой статье пользователя, желающего иметь возможность установить свою подлинность. При запросе на услугу Справочник либо подтверждает, что представленное значение является паролем, либо скажет, что это не так. Благодаря этому пользователь может не иметь различных паролей для каждой службы. В некоторых случаях обмен паролями в локальной среде, использующей простую аутентификацию, считается нежелательным. В этих случаях Справочник может обеспечить средства защиты паролей от их "розыгрыша" или недопустимого использования; для этого Справочник может использовать функции, действующие только в одном направлении.

Более сложный подход, называемый "строгой аутентификацией", основан на криптографии с общедоступными ключами, успешно защищающей от вмешательства. При этом подходе Справочник выступает в роли хранителя общедоступных криптографических ключей пользователей. Шаги, которые должны предпринять пользователи для получения общедоступного ключа друг друга, а потом, используя этот ключ, аутентифицировать один другого, детально описаны в Рекомендации X.509.

A.4 *Общие приложения*

A.4.1 *Введение*

Существует несколько общих приложений, на которые можно смотреть как на неявно обеспечиваемые Справочником; это приложения, которые не являются специфичными для одной какой-то специальной телекоммуникационной службы. Ниже описываются два таких приложения: справочник по межперсональным связям и справочник по взаимосвязи между системами (для ВОС).

Примечание. — Аутентификация, которая в предыдущем подпункте была описана в качестве одного из "шаблонов доступа", может в то же время рассматриваться и как "общее приложение" Справочника.

A.4.2 *Межперсональная связь*

Это приложение преследует цель обеспечить людей или их агентов информацией о том, как им взаимодействовать с другими людьми или группами людей.

Здесь наверняка используются следующие классы объектов: человек, служебное положение и группа. Кроме того, потребуется использование многих других классов, хотя, возможно, и не столь непосредственным образом; к этим классам относятся, например, страны, организации, подразделения организаций.

В эту службу будут вовлечены, помимо атрибутов, связанных с именованиями, в основном атрибуты, связанные с адресацией. На практике, статья отдельного конкретного человека будет содержать адреса, соответствующие каждому из тех методов связи, по которым имеется доступ к этому человеку. Эти адреса извлекаются из открытого снизу списка, который содержит как минимум телефон, электронную почту, телекс, ЦСИС, метод физической доставки (например, почтовую систему), факсимиле. В некоторых случаях, например, для электронной почты, статья будет содержать некоторую дополнительную информацию, такую, скажем, как тип информации, обеспечиваемой оборудованием пользователя. Если должна быть обеспечена аутентификация, то потребуется наличие пароля и/или удостоверения пользователя.

Метод именования, используемый для различных классов объектов, должен быть "удобным для пользователя". Кроме того, должны быть введены подходящие псевдонимы, которые обеспечивали бы альтернативные имена, преемственность при смене имен и т.д.

В настоящем приложении заявляются следующие шаблоны доступа: выборка, удобное для пользователя именование, просмотр, "Желтые страницы" и группы. До некоторой степени будет использоваться еще и аутентификация.

А.4.3 *Взаимосвязь между системами (для ВОС)*

Эталонная модель взаимосвязи открытых систем нуждается в двух функциях Справочника. Одна из них функционирует в прикладном уровне; ее назначение — отображение титулов приложений на адреса уровня представлений. Другая, функционирующая в сетевом уровне, отображает адреса СтПДС на адреса ПСПП (ПСПП= подсетевому пункту присоединения).

Примечание. — В остальной части настоящего параграфа рассматривается только случай прикладного уровня.

Если информация, требующаяся для выполнения этой функции, недоступна обычными средствами, то для выполнения функции требуется обращение к Справочнику.

Пользователями являются прикладные элементы; интересующие классы объектов также являются прикладными элементами или некоторыми их подклассами.

Основным требующимся типом атрибута, помимо атрибутов, используемых для именования, является атрибут, связанный с пунктом доступа к уровню представлений. Другими типами атрибутов, не считающимися обязательными для самой функции Справочника, являются типы, обеспечивающие верификацию или выявление типа прикладного элемента, а также списки прикладных контекстов, абстрактных синтаксисов и т.п. Могут пригодиться также и типы атрибутов, требующихся для аутентификации.

Основным объявляемым шаблоном доступа в этом случае является "выборка".

Рекомендация X.501

СПРАВОЧНИК — МОДЕЛИ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

[1)] Рекомендации X.501 и ISO 9594-2, "Справочник – Модели", были разработаны в тесном сотрудничестве и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой ими справочной информацией может рассматриваться как объединенное целое, называемое *Справочником*. Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные процессы, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых разными производителями;

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 В настоящей Рекомендации описываются несколько различных моделей Справочника в качестве рамок для других Рекомендаций. Этими моделями являются: общая (функциональная) модель; организационная модель; модель безопасности; структура информации. Последняя модель описывает способ организации в Справочнике хранимой в нем информации. Она описывает, к примеру, каким образом информация об объектах объединяется так, чтобы образовать статьи Справочника об этих объектах, и каким образом эта информация обеспечивает имена объектов.

0.4 В Приложении А кратко изложена математическая терминология, относящаяся к структуре деревьев.

0.5 В Приложении В кратко изложено, как НАС.1-идентификаторы объектов используются в настоящей серии Рекомендаций.

0.6 В Приложении С представлен НАС.1-модуль, содержащий все определения, связанные со структурой информации.

0.7 В Приложении D в алфативном порядке перечислены термины, определенные в данном документе.

0.8 В Приложении E описаны некоторые критерии, которые можно учесть при выработке имен.

0.9 В Приложении F описываются основные принципы управления доступом.

1 Предмет рассмотрения и область применения

1.1 Модели, определяемые в настоящей Рекомендации, обеспечивают концептуальные и терминологические рамки для других Рекомендаций, в которых определяются различные аспекты Справочника.

1.2 Функциональная и организационная модели определяют те способы, которыми может быть распределен Справочник, как функционально, так и административно.

1.3 Модель безопасности определяет рамки, в которых такие средства безопасности, как, например, управление доступом, могут быть обеспечены в Справочнике.

1.4 Информационная модель описывает логическую структуру ИБС. Эта точка зрения скрывает то обстоятельство, что Справочник является распределенным, а не централизованным. Другие Рекомендации этой серии пользуются концепциями структуры информации, а именно:

a) служба, обеспечиваемая Справочником, описана (в Рекомендации X.511) в терминах концепций структуры информации; в силу этого обеспечиваемая служба является в некотором смысле независимой от физической распределенности ИБС;

b) распределенное функционирование Справочника специфицировано (в Рекомендации X.518) так, чтобы обеспечить эту службу; вследствие этого сохраняется логическая структура информации, учитывающая, что фактически ИБС в высокой степени распределена.

2 Библиография

Рекомендация X.200 "Взаимосвязь открытых систем — Базовая эталонная модель".

Рекомендация X.500 "Справочник — Обзор концепций, моделей и служб".

Рекомендация X.509 "Справочник — Структура аутентификации".

Рекомендация X.511 "Справочник — Определение доступа и системных служб".

Рекомендация X.518 "Справочник — Процедуры распределенной операции".

Рекомендация X.519 "Справочник — Спецификация доступа и системных протоколов".

Рекомендация X.520 "Справочник — Избранные типы атрибутов".

Рекомендация X.521 "Справочник — Избранные классы объектов".

3 Определения

Определения терминов приводятся в начале тех пунктов, где они уместны. Для удобства ссылок в Приложении D приведен алфавитный перечень этих терминов.

4 Сокращения

АОУС	Административная область управления Справочником
ПЗА	Проверка значения атрибута
ИБС	Информационная база Справочника
ИДС	Информационное дерево Справочника
ОУС	Область управления Справочником
САС	Системный агент Справочника
АПС	Агент пользователя Справочника

ЧОУС Частная область управления Справочником

ОВИ Относительно выделенное имя

РАЗДЕЛ 1 — *Модель Справочника*

5 Модель Справочника

5.1 *Определения*

a) *пункт доступа:* место, в котором обеспечивается абстрактная служба;

b) *административная область управления Справочником (АОУС):* ОУС, которая находится под управлением Администрации.

Примечание. — Термин "Администрация" обозначает администрацию телекоммуникационных служб общего пользования или другую организацию, обеспечивающую телекоммуникационную службу общего пользования;

c) *административный руководящий орган* — элемент, осуществляющий административное управление всеми статьями, хранящимися у одного системного агента Справочника;

d) *Справочник* — хранилище информации об объектах, предоставляющее пользователям справочные службы, обеспечивающие им доступ к информации;

e) *область управления Справочником (ОУС)* — совокупность одного или более САС и ноль или более АПС, управляемых одной организацией;

f) *системный агент Справочника (САС)* — прикладной процесс ВОС, являющийся частью Справочника;

g) *пользователь (Справочника)* — оконечный пользователь Справочника, то есть элемент или человек, имеющий доступ к Справочнику;

h) *агент пользователя Справочника (АПС)* — прикладной процесс ВОС, который представляет пользователя при доступе к Справочнику.

Примечание. — Кроме того, АПС могут предоставлять целый ряд локальных средств, помогающих пользователям в составлении опросов и интерпретации ответов;

i) *частная область управления Справочником (ЧОУС):* ОУС, управляемый организацией, иной, чем Администрация.

5.2 *Справочник и его пользователи*

5.2.1 Чтобы пользователь Справочника (то есть лицо или прикладной процесс) мог пользоваться службами Справочника, он должен осуществить доступ к *Справочнику.* Точнее, это *Агент пользователя Справочника* (АПС), который осуществляет фактически доступ к Справочнику и взаимодействует с ним для обращения к службам в интересах конкретного пользователя. Справочник обеспечивает один или более *пунктов доступа,* в которых и может осуществляться этот доступ. Эти концепции иллюстрируются на рис. 1/X.501.

5.2.2 Службы, предоставляемые Справочником, определены в Рекомендации X.511.

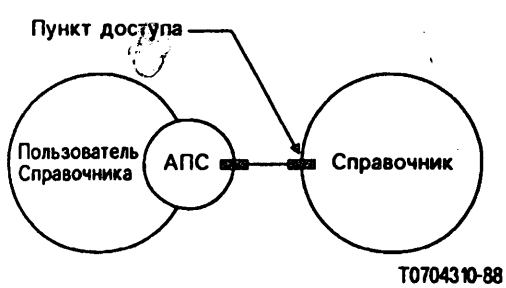


РИСУНОК 1/X.501

Доступ к Справочнику

5.2.3 Справочник является хранилищем информации об объектах, а службы Справочника, обеспечиваемые им для пользователя, связаны с различными типами доступа к этой информации. В своей сопокупности эта информация известна как *Информационная база Справочника (ИБС)*. Модель ИБС определена в разделе 2 данной Рекомендации.

5.2.4 АПС выступает в качестве прикладного процесса. Каждый АПС представляет именно одного пользователя Справочника.

Примечание 1. — Некоторые открытые системы могут обеспечивать функцию централизованного АПС, извлекая информацию для фактических пользователей (прикладных-процессов, людей и т.д.). Для Справочника это прозрачно.

Примечание 2. — Функции АПС и САС (см. § 5.3.1) могут находиться в одной и той же открытой системе; на усмотрение реализатора оставляется сделать одного или более АПС видимыми в среде ВОС в роли прикладных-элементов.

Примечание 3. — По всей вероятности, АПС будет проявлять локальные режимы работы и структуру, что лежит вне предмета рассмотрения данных Рекомендаций. Например, АПС, представляющий интересы человека-пользователя Справочника, может обеспечивать ряд локальных средств, помогающих пользователям в составлении опросов и интерпретации ответов.

5.3 *Функциональная модель*

5.3.1 Справочник выступает в качестве набора одного или более прикладных-процессов, известных как *Системный агент Справочника (САС)*, каждый из которых обеспечивает нуль, один или более пунктов доступа. Сказанное иллюстрируется на рис. 2/X.501. Если Справочник состоит из более чем одного САС, то о нем говорят, что он *распределенный*. Процедуры функционирования Справочника, являющегося распределенным, специфицированы в Рекомендации X.518.

Примечание. — По всей вероятности, САС будет проявлять локальные режим работы и структуру, что лежит вне предмета рассмотрения данных Рекомендаций. Например, САС, который ответствен за хранение у себя части или всей информации из ИБС, обычно выполняет это средствами базы данных, интерфейс с которой является локальным вопросом.

5.3.2 Некоторая конкретная пара прикладных процессов, которым необходимо взаимодействие для обеспечения служб Справочника (или АПС и САС или два САС) может быть расположена в различных открытых системах. Такое воздействие происходит в соответствии с протоколами ВОС Справочника, как это специфицировано в Рекомендации X.519.

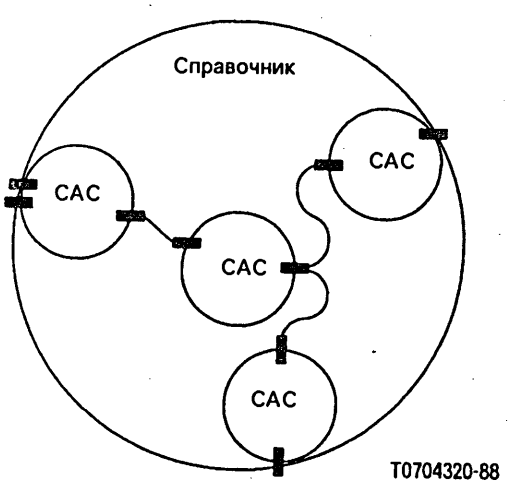


РИСУНОК 2/X.501

Справочник, обеспечиваемый несколькими САС

5.4 *Организационная модель*

5.4.1 Совокупность одного или более САС и нуль или более АПС, управляемых одной организацией, может образовывать *область управления Справочником (ОУС)*.

Примечание. — Организация, которая управляет ОУС, может быть Администрацией (то есть администрацией телекоммуникационных служб общего пользования или другой организацией, обеспечивающей телекоммуникационные службы общего пользования), и в этом случае ОУС называется Административным ОУС (АОУС); в противном случае, это — частный ОУС (ЧОУС). Следует учесть, что обеспечение поддержки частных справочных систем членами МККТТ входит в рамки национальных соглашений. Таким образом, описанные технические возможности могут обеспечиваться, а могут и не обеспечиваться Администрацией, которая поставляет службы Справочника. Внутреннее функционирование и конфигурация частного ОУС находятся вне предмета рассмотрения настоящих Рекомендаций МККТТ.

5.4.2 Управление АПС со стороны ОУС включает ответственность ОУС за обслуживание АПС, например текущее обеспечение, а в некоторых случаях — владение АПС.

5.4.3 Некоторая организация может как воспользоваться, так и не воспользоваться настоящей серией Рекомендаций для определения взаимодействия между АПС и САС, что является полностью внутренним вопросом ОУС.

5.4.4 Каждым САС распоряжается Административный руководящий орган. Этот элемент управляет всеми статьями объектов и статьями псеводнимов, загруженных этим САС. В это входит ответственность схемы Справочника, используемой для создания и модификации статей (см. § 9). Структура и распределение имен является обязанностью органа по присвоению имен [см. § 8.1, *f*)], а ролью Административного органа является реализация этих именных структур в схеме.

РАЗДЕЛ 2 — *Информационная модель*

6 Информационная база Справочника

6.1 *Определения*

a) *статья псевдонима* — статья, принадлежащая классу "псевдоним", содержащая информацию, используемую для обеспечения объекта альтернативным именем;

b) *информационная база Справочника (ИБС)* — полное множество информации, к которой Справочник обеспечивает доступ и которая включает все порции информации, которые можно читать, или с которыми можно производить действия, используя операции Справочника;

c) *информационное дерево Справочника (ИДС)* — ИБС, рассматриваемая в виде дерева, вершины которого (отличные от корня) являются статьями Справочника;

Примечание. — Термин ИДС употребляется взамен ИБС только в контексте, в котором древовидная структура информации существенна.

d) *статья (Справочника)* — часть ИБС, содержащая информацию об объекте;

e) *непосредственный предшественник* (имя существительное) по отношению к конкретной статье или объекту (из контекста должно быть ясно, что именно имеется в виду) — непосредственно предшествующая статья или объект;

f) *непосредственно предшествующая*

(статья) — по отношению к конкретной статье — статья, являющаяся начальной вершиной дуги ИДС, конечной вершиной которой является данная статья;

(объект) — по отношению к конкретному объекту — объект, статья *объекта* которого является непосредственно предшествующей *любой* из статей (объекта или псевдонима) второго объекта;

g) *объект (представляющий интерес)* — любой объект из некоторого "мира", обычно из мира телекоммуникаций и обработки информации или их некоторой части, который может быть идентифицирован (назван) и который представляет достаточный интерес, чтобы хранить информацию о нем в ИБС;

h) *класс объектов* — идентифицируемое семейство объектов (или мыслимых объектов), обладающих некоторыми общими характеристиками;

i) *статья объекта* – *статья,* являющаяся хранимой в ИБС первичной совокупностью информации об объекте, и о которой вследствие этого можно сказать, что она представляет этот объект в ИБС;

j) *подкласс* — по отношению к надклассу — класс объектов, выделенных из надкласса. Члены подкласса обладают всеми характеристиками другого класса объектов (надкласса), а также дополнительными характеристиками, которыми ни один член этого класса объектов (надкласса) не обладает;

k) *подчиненный/последующий* — обратный к предшествующему;

l) *надкласс* — по отношению к подклассу — класс объектов, из которого выделен подкласс;

m) *предшествующий* — (в применении к статье или объекту) непосредственно предшествующий или предшествующий непосредственно предшествующему (рекурсивно).

6.2 *Объекты*

6.2.1 Назначением Справочника является хранение и предоставление доступа к информации об *объектах, представляющих интерес (объектах)*, которые существуют в некотором "мире". Объект — это нечто в этом мире, что может быть идентифицировано (снабжено именем).

Примечание 1. — "Мир" — это обычно мир телекоммуникаций и обработки информации или часть указанного.

Примечание 2. — Объекты, известные Справочнику, не обязательно точно соответствуют "реальным" вещам в мире. Например, человек из реального мира, с точки зрения Справочника, может рассматриваться как два различных объекта: деловое лицо и житель. Отображение в данной Рекомендации не определено; оно является делом пользователей и поставщиков Справочника в контексте их приложений.

6.2.2 Полный набор информации, к которой Справочник обеспечивает доступ, известен как *Информационная база Справочника* (ИБС). Все порции информации, которые можно прочитать и с которыми можно производить действия посредством операций Справочника, считаются включенными в ИБС.

6.2.3 *Класс объектов* является идентифицируемым семейством объектов (или мыслимых объектов), обладающих некоторыми общими конкретными характеристиками. Каждый объект принадлежит по крайней мере одному классу. Класс объектов может быть *подклассом* другого класса объектов; в этом случае члены первого класса (подкласса) рассматриваются также как члены последнего (надкласса). Могут существовать подклассы подклассов и т.д. на произвольную глубину.

6.3 *Статьи Справочника*

6.3.1 ИБС состоит из *статей Справочника (статей)*, каждая из которых содержит информацию (описание) об отдельном объекте.

6.3.2 Для любого конкретного объекта имеется определенно одна *статья объекта*, составляющая первичную совокупность информации об этом объекте в ИБС. Говорят, что статья объекта представляет объект.

6.3.3 Для любого конкретного объекта может иметься в дополнение к статье объекта одна или более статей псевдонимов этого объекта, которые используются для обеспечения альтернативных имен (см. § 8.5).

6.3.4 Структура статьи Справочника изображена на рис. 3/X.501 и описана в § 7.2.

6.3.5 Каждая статья содержит указание класса объектов и надклассов данного класса объектов, с которыми связана эта статья. В случае статьи объекта — это указание класса (классов), к которому принадлежит объект, а в случае статьи псевдонима — это указание, посредством особого класса объектов "псевдоним" (определенного в § 9.4.8.2), на то, что на самом деле статья является статьей псевдонима; кроме того, в статье может содержаться указание на то, какому подклассу (подклассам) класса объектов "псевдоним" принадлежит данная статья.

6.4 *Информационное дерево Справочника (ИДС)*

6.4.1 Для того чтобы удовлетворить требования распределенности и управления потенциально очень большой ИБС и обеспечить требования, чтобы объекты назывались однозначно определенными именами (см § 8), а их статьи отыскивались, плоское расположение статей, по всей видимости, неудобно. Поэтому можно использовать обычно наблюдаемую иерархическую зависимость между объектами (например, человек работает в отделе, который принадлежит организации, штаб-квартира которой находится в стране), размещая статьи в форме дерева, известного как *Информационное дерево Справочника (ИДС)*.

Примечание. — Введение в концепции и терминологию древовидных структур можно найти в Приложении А.

6.4.2 Компоненты ИДС интерпретируются следующим образом:

a) вершины являются статьями. Статья объекта может быть как листом, так и не листом, причем статьи псевдонима всегда являются листьями. Корень не является статьей как таковой, но если это удобно (например, в нижеследующих определениях b) и c)), может рассматриваться как статья нулевого объекта [см. d), ниже]:

b) дуга определяет зависимость между вершинами (и, следовательно, статьями). Дуга из вершины A в вершину B означает, что статья A является *непосредственно предшествующей статьей (непосредственным предшественником)* для статьи B, и, наоборот, что статья B является *непосредственно последующей статьей (непосредственным подчиненным)* для статьи A. *Предшествующими статьями (предшественниками)* для некоторой статьи является ее непосредственно предшествующая вместе со своими предшественниками (рекурсивно). *Последующими статьями (подчиненными)* некоторой статьи являются ее непосредственно последующая вместе со своими последующими (рекурсивно);

c) объект, представленный статьей, для своих непосредственно последующих является органом распределения имен или непосредственно связан с этим органом (см. § 8);

d) корень представляет наивысший уровень органов распределения имен для ИБС.

6.4.3 Отношения предшествования/следования между объектами могут быть выведены из таких же отношений между статьями. Объект является *непосредственно предшествующим объектом (непосредственным предшественником)* другому объекту тогда и только тогда, когда статья объекта для первого объекта является непосредственно предшествующей любой из статей для второго объекта. Термины *непосредственно последующий объект, непосредственный подчиненный, предшествующий* и *последующий* (в приложении к объектам) имеют аналогичные значения.

6.4.4 Допустимые отношения предшествования/следования между объектами регулируются определениями структуры ИДС (см. § 9.2).

7 Статьи Справочника

7.1 *Определения*

a) *атрибут* — информация о конкретном типе, относящемся к объекту, и появляющаяся в статье, описывающей этот объект в ИБС;

b) *тип атрибута* — компонент атрибута, указывающий на класс информации, заданный данным атрибутом;

c) *значение атрибута* — конкретный экземпляр класса информации, заданной типом атрибута;

d) *проверка значения атрибута* — высказывание, которое может быть истинно, ложно или неопределенно и зависит от значений (или, может быть, только от выделенных значений) статьи;

Примечание. — В данном документе обозначения типа "цепочка1 = цепочка2" используются в качестве примеров проверки значения атрибута. В этом обозначении "цепочка1" является аббревиатурой для "имени" типа атрибута, а "цепочка2" является текстуальным представлением подходящего значения. Хотя типы атрибутов в примерах часто основаны на реальных типах, таких как определенные в Рекомендации X.520 (то есть "C" заменяет "страна", ОИ — "обычное имя"), это не является обязательным для целей данного документа, так как обычно Справочник не осведомлен о значениях используемых типов атрибутов.

e) *выделенное значение* — значение атрибута в статье, которое предназначено для появления в относительно выделенном имени статьи.

7.2 *Общая структура*

7.2.1 Статья состоит из набора *атрибутов*, как это показано на рис. 3/X.501.

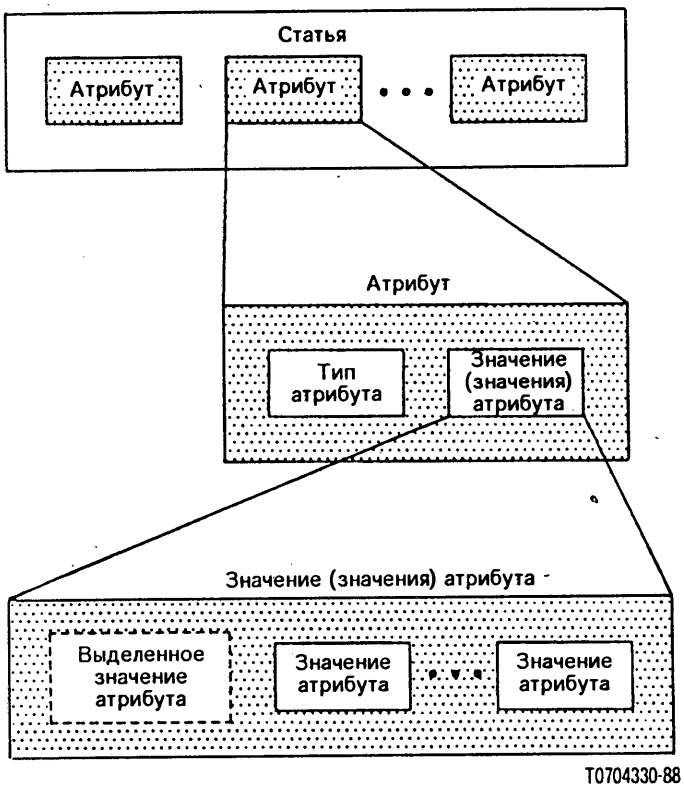


РИСУНОК 3/X.501

Структура статьи

7.2.2 Каждый атрибут обеспечивает часть информации об объекте или описывает конкретные характеристики объекта, которому соответствует статья.

Примечание. — Примеры атрибутов, которые могут быть представлены в статье, включают именующую информацию, например, личное имя объекта, и адресную информацию, такую как телефонный номер.

7.2.3 Атрибут состоит из *типа атрибута,* который идентифицирует класс информации, заданной атрибутом, и соответствующее (щие) *значение(ния) атрибута,* которые являются конкретными появляющимися в статье экземплярами этого класса.

```
Атрибут ::=
    SEQUENCE {
        тип             ТипАтрибута
        значения        SET OF ЗначениеАтрибута
        -- требуется хотя бы одно значение --}
```

7.3 *Типы атрибутов*

7.3.1 Некоторые типы атрибутов будут иметь международные стандарты. Другие типы будут определены национальными административными органами и частными организациями. Отсюда следует, что несколько отдельных органов должно быть ответственно за выработку типов и притом так, чтобы гарантировать отличие каждого типа от всех других типов. Это достигается идентифицированием (при определении типа) каждого типа атрибута идентификатором объекта (как это описано в § 9.5).

```
ТипАтрибута ::= OBJECT IDENTIFIER
```

7.3.2 Все атрибуты в статье должны быть атрибутами различных типов.

7.3.3 Существует несколько типов атрибутов, которые известны Справочнику и которые он использует в своих целях. К ним относятся:

a) **КлассОбъектов.** Атрибут этого типа появляется в каждой статье и указывает тот класс объектов и тот (те) надкласс (ы), к которым принадлежит объект.

b) **ИмяОбъектаПсевдонима**. Атрибут этого типа появляется в каждой статье псевдонима и содержит выделенное имя (см. § 8.5) того объекта, который описывается этим псевдонимом.

Эти атрибуты определены (частично) в § 9.5.4.

7.3.4 Типы атрибутов, которые должны или могут появляться в статье (отличные от типов, описанных в § 7.3.3), регулируются правилами, относящимися к классу индицируемого (мых) объекта (тов).

7.4 *Значения атрибутов*

7.4.1 Определение типа атрибута (см. § 9.5) включает в себя, кроме того, спецификацию синтаксиса, а следовательно, типа данных, которому должно соответствовать каждое значение атрибута. Это может быть любой тип данных:

ЗначениеАтрибута :: = **ANY**

7.4.2 Максимум одно из значений атрибута может быть указано как *выделенное значение;* в этом случае значение атрибута фигурирует в относительно выделенном имени (см. § 8.3) статьи.

7.4.3 *Проверка значения атрибута (ПЗА)* является высказыванием, которое может быть истинным, ложным или неопределенным; это высказывание зависит от значений (или, возможно, только от выделенных значений), входящих в статью. Оно содержит тип атрибута и значение атрибута:

ПроверкаЗначенияАтрибута :: =
SEQUENCE {**ТипАтрибута, ЗначениеАтрибута**}

и является:

a) неопределенным, если выполнено хотя бы одно из следующих утверждений:

i) тип атрибута неизвестен;

ii) синтаксис атрибута данного типа не имеет правил сопоставления на равенство;

iii) значение не соответствует типу данных синтаксиса атрибута;

Примечание. — Как правило, случаи ii) и iii) свидетельствуют об ошибочном ПЗА, однако могут возникнуть и как локальная ситуация (например, некоторый конкретный САС не зарегистрировал этот конкретный тип атрибута).

b) истинным, если статья содержит атрибут этого типа, одно из значений которого сопоставимо с этим значением (если проверка связана только с выделенными значениями, то сопоставляемое значение также должно быть выделенным);

Примечание. — Сопоставление значений осуществляется на равенство и содержит правило сопоставления, связанное с синтаксисом атрибута.

c) ложным, в противном случае.

8 Имена

8.1 *Определения*

a) *псевдоним, имя псевдонима* — имя объекта, обеспечиваемое использованием одной или более статей псевдонимов в ИДС;

b) *переименование* — замена фиктивного имени объекта выделенным именем объекта;

c) *выделенное имя* (объекта) — одно из имен объекта, формируемое из последовательности ОВИ статьи объекта и ОВИ каждой из ее предшествующих статей;

d) *имя (в Справочнике)* — конструкция, которая выделяет индивидуальный объект из всех других объектов. Имя должно быть недвусмысленным (то есть обозначать ровно один объект), хотя оно и не должно быть уникальным (то есть может быть единственным именем, однозначно обозначающим объект);

e) *потенциальное имя* — конструкция, которая синтаксически является именем, однако (пока) не установлено, что оно является действительным именем;

f) *орган распределения имен* — орган, ответственный за распределение имен. Каждый объект, статья объекта которого расположена в вершине ИДС, не являющейся листом, является органом распределения имен или непосредственно связан с органом распределения имен;

g) *относительно выделенное имя (ОВИ)* — множество проверок значений атрибутов, каждое из которых истинно, связанных с выделенными значениями индивидуальной статьи.

8.2 *Общее описание имен*

8.2.1 *Имя (в Справочнике)* является конструкцией, которая идентифицирует индивидуальный объект в множестве всех объектов. Имя должно быть недвусмысленным, то есть точно обозначать один объект. Однако имя не обязано быть уникальным, быть единственным именем, однозначно обозначающим объект.

8.2.2 Синтаксически каждое имя объекта является упорядоченной последовательностью относительно выделенных имен (см. § 8.3).

```
ИМЯ ::=
    CHOICE { -- сейчас только одна возможность --
        ПоследовательностьОВИ}

ПоследовательностьОВИ ::= SEQUENCE OF ОтносительноВыделенноеИмя
ВыделенноеИмя ::= ПоследовательностьОВИ
```

Примечание. — Имена, сформированные иначе чем описано выше, возможно, будут использоваться в будущем.

8.2.3 Пустая последовательность является именем корня дерева.

8.2.4 Каждая начальная подпоследовательность имени объекта является, в свою очередь, именем объекта. Последовательность идентифицированных таким образом объектов, начинающаяся с корня и заканчивающаяся именованным объектом, такова, что каждый объект является непосредственно предшествующим тому, который следует за ним в этой последовательности.

8.2.5 *Потенциальное имя* — это конструкция, которая синтаксически является именем, однако (пока) не установлено, что она является действительным именем.

8.3 *Относительно выделенное имя*

8.3.1 Каждая статья обладает уникальным *относительно выделенным именем (ОВИ)*. ОВИ состоит из множества проверок значений атрибутов, каждое из которых истинно, связанных с выделенными значениями статьи.

```
ОтносительноВыделенноеИмя ::=
    SET OF ПроверкаЗначенийАтрибута
```

Множество содержит в точности одну проверку, связанную с каждым выделенным значением в статье.

8.3.2 ОВИ всех статей, каждая из которых является непосредственно предшествующей следующей, различны между собой. Соответствующий орган распределения имен этого объекта несет ответственность за то, чтобы было выполнено это требование; это достигается подходящим присвоением выделенных значений атрибута.

Примечание. — Часто статья содержит единственное выделенное значение (и ОВИ поэтому содержит единственную ПЗА); однако при определенных обстоятельствах (как правило, с целью различения) могут использоваться дополнительные значения (и, следовательно, дополнительные ПЗА).

8.3.3 ОВИ для статьи выбирается, когда она создается. Отдельный экземпляр значения любого типа атрибута может быть использован в качестве части ОВИ в зависимости от природы класса именуемого объекта. Распределение ОВИ осуществляется административно с учетом того, что переговоры между заинтересованными организациями или администрациями могут потребоваться или не потребоваться. Настоящие Рекомендации не обеспечивают механизма таких переговоров и не делают никаких предположений относительно того, как их проводить. ОВИ может быть модифицировано, если это необходимо с помощью полной замены.

Примечание. — Предполагается, что у ОВИ будет долгая жизнь, так чтобы пользователи Справочника могли сохранять выделенные имена объектов (например, в самом Справочнике), не беспокоясь о том, что они могут быть отменены. Поэтому ОВИ должны изменяться осторожно.

8.4 *Выделенные имена*

8.4.1 *Выделенное имя* данного объекта является такой последовательностью ОВИ статьи, которая представляет объект и ОВИ всех ее предшествующих статей (в порядке убывания). Вследствие взаимооднозначного соответствия между объектами и статьями объектов выделенное имя объекта может рассматриваться, кроме того, и как идентифицирующее статью объекта.

Примечание 1. — Желательно, чтобы выделенные имена объектов, с которыми имеют дело люди, были бы удобными для пользователя.

Примечание 2. — В стандарте ISO 7498/3 определяется концепция примитивного имени. Выделенное имя может использоваться как примитивное имя объекта, который оно идентифицирует, потому что a) оно недвусмысленно; b) оно уникально; и c) его внутренняя структура (последовательность ОВИ) не обязана (однако, конечно, может) быть понятной пользователю Справочника.

Примечание 3. — Так как используются только статьи объектов и их предшествующие, выделенное имя объекта никогда не сможет использовать статьи псевдонима.

8.4.2 Для удобства определяются также "выделенные имена" корня и статьи псевдонима, хотя имя никогда не бывает еще и выделенным именем статьи. Выделенное имя корня определяется как пустая последовательность. Выделенное имя статьи псевдонима определяется как последовательность ОВИ статьи псевдонима и ОВИ каждой из ее предшествующих статей (в порядке убывания).

8.4.3 На рис. 4/X.501 изображен пример, иллюстрирующий концепции ОВИ и выделенного имени.

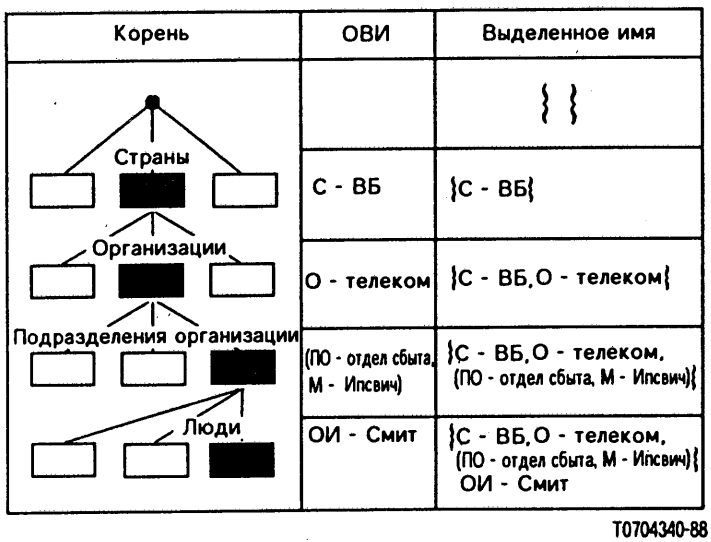


РИСУНОК 4/X.501

Определение выделенных имен

8.5 *Псевдоним*

8.5.1 *Псевдоним* или *имя псевдонима* объекта — это имя, по крайней мере одно из ОВИ которого является ОВП псевдонима. Псевдонимы позволяют статьям объектов иметь как бы несколько непосредственно предшествующих статей, обеспечивая тем самым основание для альтернативного имени.

8.5.2 Точно так же, как выделенное имя объекта выражает его основные соотношения в некоторой иерархии объектов, так и псевдоним выражает (в общем случае) альтернативные соотношения в другой иерархии объектов.

8.5.3 Объект, представленный в ИБС статьей, может иметь нуль или более псевдонимов. Следовательно, на одну и ту же статью объекта могут указывать различные статьи псевдонимов. Такие статьи могут указывать на статью объекта, не являющуюся листом. Псевдонимы могут иметь только статьи объектов; таким образом, псевдонимы псевдонимов не допускаются.

8.5.4 Псевдоним не имеет подчиненных, и следовательно, является листом.

8.5.5 Справочник пользуется атрибутом "имя объекта псевдоним" в статье псевдонима для идентификации и отыскивания соответствующей статьи объекта.

9 Схема Справочника

9.1 *Определения*

a) *Схема Справочника* — набор правил и ограничений, относящийся к структуре ИДС, определениям классов объектов, типов атрибутов и синтаксисов, характеризующих ИБС;

b) *правило структуры ИДС* — правило, являющееся частью схемы Справочника, которое соотносит класс объектов (последующий) с другим классом объектов (предшествующим) и которое позволяет статье из второго класса быть в ИДС непосредственно последующей статьей из первого класса. Кроме того, правило определяет тот (те) тип (ы) атрибута (ов), который (ые) может (могут) фигурировать в ОВИ (последующих) статей и может налагать дополнительные условия. Схема может содержать много таких правил.

9.2 *Общее описание*

9.2.1 Схема Справочника является набором определений и ограничений, касающихся структуры ИДС и допустимых способов именования статей, а также касающихся информации, которая может храниться в статьях, и атрибутов, используемых для представления этой информации.

Примечание 1. — Схема позволяет системе Справочника, например:

— предотвращать построение последующих статей из ошибочных классов объектов (например, страны как следующей за человеком);

— предотвращать добавление к статье таких типов атрибутов, которые несовместимы с классом объекта (например, серийный номер к статье человека);

— предотвращать добавление к атрибуту значения, синтаксис которого не согласуется с определенным для данного типа атрибута (например, печатаемой цепочки к цепочке битов).

Примечание 2. — Динамические механизмы для управления схемой Справочника в настоящее время этой серией Рекомендаций не обеспечиваются.

9.2.2 Формально Схема Справочника является множеством, состоящим из:

a) определений (правил) *структуры ИДС,* определяющих выделенные имена, которые могут иметь статьи, и способы, которыми они могут быть соотнесены друг с другом посредством ИДС;

b) определений *классов объектов,* описывающих множество обязательных и дополнительных атрибутов. которые, соответственно, должны быть и могут быть представлены в статье данного класса (см. § 6.2.3 настоящей Рекомендации);

c) определений *типов атрибутов,* идентифицирующих идентификаторы объектов, под которыми атрибуты известны, их синтаксис и могут ли атрибуты иметь по несколько значений;

d) определений *синтаксиса атрибутов,* описывающих для каждого атрибута тип соответствующих данных и правила сопоставления в форме НАС.1.

На рис. 5/X.501 суммируются отношения между определениями схемы, с одной стороны, и ИДС, статьями Справочника, атрибутами и значениями атрибутов — с другой.

9.2.3 Схема Справочника, как и сама ИБС, является распределенной. Каждый Административный орган сам определяет схему, применяемую к тем порциям ИБС, которое обеспечивает это ведомство.

Примечание. — Распределение информации о схемах между различными САС, руководимыми различными административными органами, не представлено в настоящей серии Рекомендаций. Такое распределение производится административным способом путем двусторонних соглашений.

9.2.4 Спецификации, которые описывают, что входит в определения структуры ИДС, классов объектов, типов атрибутов и синтаксиса атрибутов, можно найти в § 9.3 — 9.6 соответственно.

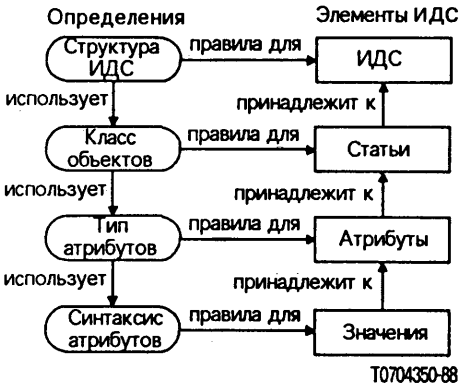


РИСУНОК 5/X.501

Обзор схемы Справочника

9.3 *Определение структуры ИДС*

9.3.1 Определение правила структуры ИДС содержит:

— идентификацию последующего и предшествующего классов объектов;

— идентификацию типов атрибутов, которые могут содержаться в ОВИ последующих статей; и

— (не обязательную) дополнительную информацию.

9.3.2 Справочник позволяет некоторой статье являться непосредственно следующей за другой статьей (ей непосредственно предшествующей) только в том случае, если существует определение структуры ИДС, входящее в схему (см. § 9.2.3), применимую к той порции ИДС, в которой будет храниться данная статья, согласно которому:

— данная статья принадлежит классу последующих объектов;

— статья, непосредственно предшествующая данной, принадлежит предшествующему классу объектов;

— тип (ы) атрибута (тов) ОВИ статьи принадлежит (жат) допустимому (мым);

и

— удовлетворены все условия, налагаемые элементом дополнительного множества информации.

Примечание 1. — Техническая сторона документирования структуры ИДС или представления правил структуры ИДС в данной серии Рекомендаций в настоящее время не обеспечивается.

Примечание 2. — Если определение структуры ИДС позволяет некоторому предшественнику или подчиненному принадлежать какому-то конкретному классу, то тем самым неявно допускается (если только явно не оговорено противное) вхождение предшественника или подчиненного в любой класс объектов, выводимый из данного класса (см. § 9.4).

9.3.3 Справочник требует реализации определения правила структуры для каждой статьи в ИДС. Любая попытка модификации ИДС, нарушающая соответствующие структурные правила, окажется безуспешной.

9.3.4 Правило структуры ИДС, в котором класс объектов является последующим, называется привязыванием имени этого класса объектов.

9.3.5 Чтобы класс объектов мог быть представлен статьями в некоторой части ИБС, в схеме, касающейся этой части, должно быть указано хотя бы одно привязывание имени этого класса объектов. Эта схема по мере необходимости должна содержать дополнительные привязывания имен.

Примечание. — Предполагается, что класс объектов, входящий в две различные схемы, может иметь различные привязывания имени в каждой схеме.

9.4 *Определение Класса Объектов*

9.4.1 Определение класса объектов содержит:

а) необязательное присвоение идентификатора объектов классу объектов;

b) указания, для каких классов данный класс будет подклассом;

c) список *обязательных* типов атрибутов, которые должна содержать статья данного класса объектов, в дополнение к обязательным типам атрибутов всех надклассов;

d) список *необязательных* типов атрибутов, которые может содержать статья данного класса объектов, в дополнение к необязательным атрибутам всех надклассов.

Примечание. — Класс объектов без присвоенного идентификатора объектов предназначен для локального использования как средство удобного добавления новых типов атрибутов к определенному ранее надклассу. "Такое добавление предусмотрено для нескольких возможностей. Например, Административный орган может определить незарегистрированный Класс Объектов таким образом, чтобы разрешить пользователю добавлять некоторые зарегистрированные атрибуты к статье. Административный орган может ограничить атрибуты статьи некоторого класса объектов теми атрибутами, которые имеются в локальном списке. Кроме того, оно может сделать некоторые атрибуты обязательными для некоторого класса объектов, сверх требуемых зарегистрированным определением класса объектов".

9.4.2 Имеется особый класс объектов, для которого любой другой класс является подклассом. Этот класс объектов называется "Вершина" и определяется в § 9.4.8.1.

9.4.3 Каждая статья должна содержать атрибут типа КлассОбъектов для индентификации класса объекта и надкласса, к которому принадлежит данная статья. Определение этого атрибута приведено в § 9.5.4. Этот атрибут является многозначным. Должно иметься только по одному значению атрибута для "класса объектов" и для каждого из его надклассов, для которых определен идентификатор объектов, за исключением того, что присутствие значения класса "Вершина" не обязательно, если присутствуют хоть какие-нибудь другие значения.

Примечание 1. — Требование того, чтобы атрибут КлассОбъектов был представлен в каждой статье, отражено в определении "Вершина".

Примечание 2. — Так как класс объектов рассматривается как принадлежащий всем его надклассам, то каждый член цепочки надклассов вплоть до вершины представлен значением в атрибуте "класс объектов" (и каждое значение в цепочке может быть сопоставлено с некоторым фильтром).

Атрибут КлассОбъектов находится под управлением Справочника, то есть не может быть модифицирован пользователем.

9.4.4 Справочник требует, чтобы каждая статья ИБС принадлежала некоторому определенному классу объектов. Поэтому попытка модификации статьи, нарушающей класс объектов статьи, будет безуспешной.

Примечание. — В частности, Справочник предотвратит:

a) добавление к статье некоторого класса объектов типов атрибутов, которые отсутствуют в определении данного класса объектов;

b) создание статьи без одного или более типов атрибутов, обязательных для класса объектов данной статьи;

c) удаление из статьи типов атрибутов, обязательных для класса объектов данной статьи.

9.4.5 Специальный класс объектов Псевдоним определен в § 9.4.8.2. Класс объектов каждой статьи "псевдоним" является подклассом этого класса.

Примечание. — Переименование Справочником статей псеводонимов гарантирует, что значения атрибута КлассОбъектов статьи псевдонима обнаруживаются очень редко. Рекомендуется, чтобы требующиеся классы объектов псевдонимов получались из "Псевдонима" без присвоения идентификатора объектов.

9.4.6 Следующий макрос на НАС.1 может (но не обязательно) использоваться для определения класса объектов. Пустая продукция для ПодклассИз допускается только в определении Вершины:

```
OBJECT-CLASS MACRO ::=
BEGIN

TYPENOTATION ::= ПодклассИз
                 ОбязательныеАтрибуты
                 НеобязательныеАтрибуты
```

```
VALUE NOTATION  ::=
    value (VALUE OBJECT IDENTIFIER)

ПодклассИз ::=
    "SUBCLASS OF   Подклассы |
    empty
Подклассы ::=      Подкласс | подкласс","
                   Подклассы
Подкласс  ::=      value (OBJECT-CLASS)

ОбязательныеАтрибуты ::=
    "MUST CONTAIN {"Атрибуты"}" | empty

НеобязательныеАтрибуты ::=
    "MAY CONTAIN  {"Атрибуты" } "  | empty

Атрибуты           ::= ТермАтрибута | ТермАтрибута", "Атрибуты

ТермАтрибута       ::= Атрибут | КомплектАтрибутов

Атрибут            ::= value (ATTRIBUTE)

КомплектАтрибутов  ::= value (ATTRIBUTE-SET)

END
```

Соответствие между частями определения, перечисленными в § 9.4.1, и различными частями нотации, вводимой в макросе, следующее:

a) идентификатором объекта класса объектов является значение, поставляемое присвоением value в макросе;

b) надклассами, для которых данный класс является подклассом, являются классы, идентифицируемые продукцией ПодклассИз, то есть те, которые следуют за "SUBCLASS OF";

c) обязательными атрибутами являются те, которые идентифицируются списком идентификаторов объектов, вырабатываемых продукцией ОбязательныеАтрибуты, то есть те, которые следуют за "MUST CONTAIN".

d) необязательными атрибутами являются те, которые идентифицируются списком идентификаторов объектов, вырабатываемых продукцией НеобязательныеАтрибуты, то есть те, которые следуют за "MAY CONTAIN".

Примечание 1. — Идентификаторы объектов в c) и d) идентифицируют как отдельные атрибуты, так и наборы атрибутов (см. § 9.4.7). Эффективным списком в обоих случаях является объединенный набор из них. Если атрибут появляется как в наборе обязательных, так и необязательных атрибутов, то он должен рассматриваться как обязательный.

Примечание 2. — Этот макрос используется в определении выделенных классов объектов в Рекомендации X.521.

Если все части нотации, введенные макросом и описанные в b), c) и d),выше, пусты, то результирующая нотация ("OBJECT-CLASS") может быть использована для обозначения любого возможного класса объектов.

9.4.7 *Набором атрибутов* является набор атрибутов, идентифицируемых некоторым идентификатором объектов. Определение набора атрибутов состоит из:

a) присвоения набору идентификатора объекта;

b) перечня идентификаторов объектов атрибутов и других наборов атрибутов, члены которых совокупно образуют набор.

Следующий макрос на НАС.1 может (но не обязательно) использоваться для определения набора атрибутов в целях использования совместно с Макросом OBJECT-CLASS:

```
ATTRIBUTE-SET-MACRO ::=

    BEGIN

    TYPE NOTATION ::= "CONTAINS" {"Атрибуты" } " | empty

    VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)
```

```
Атрибуты ::=
    ТермАтрибута | ТермАтрибута", "Атрибуты

ТермАтрибута        ::= Атрибут | КомплектАтрибутов

Атрибут             ::= value (ATTRIBUTE)

КомплектАтрибутов   ::= value (ATTRIBUTE-SET)

END
```

Соответствие между частями определения набора атрибутов и нотацией, вводимой макросом, следующее:

a) идентификатор объекта; присваиваемый набору атрибутов, является значением, поставляемым присвоением в макросе;

b) множество тех атрибутов, которые образуют набор атрибутов, совпадает с тем объединением атрибутов и наборов атрибутов, которые идентифицируются продукцией "Атрибуты", то есть тем, что следует за "CONTAINS".

Если в нотации выбирается альтернатива "empty" ("пусто"), то результирующая нотация ("ATTRIBUTE-SET") может быть использована для обозначения любого возможного набора атрибутов.

9.4.8 Классы объектов, о которых шла речь выше, определяются в § 9.4.8.1, § 9.4.8.2.

Примечание. — Это — частичные определения; фактически идентификаторы объектов для этих классов объектов помещены в Рекомендации X.521, с тем чтобы обеспечить единственное место, в котором в настоящей серии Рекомендаций сгруппированы все идентификаторы объектов.

9.4.8.1 Класс объектов "Вершина" определяется следующим образом:

```
Вершина :: =
    OBJECT-CLASS
        MUST CONTAIN  {КлассОбъектов}
```

9.4.8.2 Класс объектов "Псевдоним" определяется следующим образом:

```
Псевдоним :: =
    OBJECT-CLASS
        SUBCLASS OF   Вершина
        MUST CONTAIN  {имяобъектаПсевдонима}
```

Примечание 1. — Класс объектов "Псевдоним" не специфицирует подходящих типов атрибутов для ОВИ статьи псевдонима. Административный орган может специфицировать подклассы класса "Псевдоним", которые специфицируют типы атрибутов, полезные для ОВИ статей псевдонимов (см. Рекомендацию X.521).

Примечание 2. — Статьи подклассов класса "Псевдоним" являются статьями псевдонимов.

9.5 *Определение типа атрибута*

9.5.1 Определение типа атрибута содержит:

a) присвоение идентификатора объекта типу атрибута;

b) указание или определение синтаксиса атрибута для типа атрибута;

c) указание того, может ли атрибут этого типа иметь только одно значение или более одного значения (рекурсия).

9.5.2 Справочник гарантирует, что указанный синтаксис атрибута используется для каждого атрибута этого типа. Кроме того, Справочник гарантирует, что атрибут данного типа будет иметь одно и только одно значение в статье, если атрибут этого типа определен как имеющий только одно значение.

9.5.3 Следующий макрос на НАС.1 может (но не обязательно) использоваться для определения типа атрибута:

```
ATTRIBUTE MACRO   ::=
BEGIN

TYPENOTATION      ::= СинтаксисАтрибута Многозначный | empty
```

```
VALUENOTATION               ::= value (VALUE OBJECT IDENTIFIER)

СинтаксисАтрибута           ::=
    "WITH ATTRIBUTE-SYNTAX" ВыборСинтаксиса

Многозначный                ::= "SINGLE VALUE"
                                | "MULTIVALUE" | empty

ВыборСинтаксиса             ::= value (ATTRIBUTE-SYNTAX)
                                Ограничение|type СопоставимыеТипы

Ограничение                 ::= "("АльтернативноеОграничение")" | empty

АльтернативноеОграничение   ::= ОграничениеЦепочки | ОграничениеЦелого

ОграничениеЦепочки          ::= "SIZE" "("ОграничениеДлины")"

ОграничениеДлины            ::= ЕдинственноеЗначение | Диапазон

ЕдинственноеЗначение        ::= value (INTEGER)

Диапазон                    ::= value(INTEGER) ". ." value
                                (INTEGER)

ОграничениеЦелого           ::= Диапазон

СопоставимыеТипы            ::= "MATCHES FOR" Сопоставление | empty

Сопоставление               ::= Сопоставим Сопоставление | Сопоставим

Сопоставим                  ::= "EQUALITY" | "SUBSTRINGS" |
                                "ORDERING"

END
```

Соответствие между отдельными частями определения, перечисленными в § 9.5.1, и различными частями нотации, введенными макросом, следующее:

a) идентификатор объекта, присваиваемый типу атрибута, является значением, поставляемым присвоением value в макросе;

b) синтаксис атрибута для типа атрибута совпадает с идентифицируемым продукцией СинтаксисАтрибута. Эта продукция либо задает отдельно определенный синтаксис атрибута, либо явно определяет синтаксис атрибута, задав его тип на HAC.1, и правила сопоставления (см. § 9.6). Если используется отдельно определенный синтаксис атрибута, то ограничение длины для соответствующих типов цепочек или значение диапазона для соответствующих целых типов может быть при желании указано;

c) атрибут имеет одно значение, если продукция Многозначный является "SINGLE VALUE", и может иметь одно или более значений, если эта продукция есть "MULTI VALUE" или empty.

Примечание. — Макрос используется в определении выделенных типов атрибутов в Рекомендации X.520.

Если в нотации типа выбирается альтернатива "empty", то результирующая нотация ("ATTRIBUTE") может быть использована для обозначения любого возможного типа атрибута.

9.5.4 Типы атрибутов, идентифицированные в § 7.3.3, которые известны Справочнику и используются им для своих нужд, определяются следующим образом:

```
КлассОбъектов ::= ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX синтаксисИдентификаторовОбъектов

ИмяОбъектаПсевдонима ::= ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX синтаксисВыделенногоИмени
    SINGLE VALUE
```

Примечание 1. — Это — частичное определение; фактически идентификаторы объектов для этих классов объектов размещены в Рекомендации X.520, с тем чтобы обеспечить единственное место, в котором в настоящей серии Рекомендаций сгруппированы все определения.

Примечание 2. — Синтаксисы атрибутов, на которые делается ссылка в этих определениях, в свою очередь определены в § 9.6.5.

9.6 *Определение синтаксиса атрибутов*

9.6.1 Определение синтаксиса атрибутов содержит:

a) необязательное присвоение идентификатора объекта синтаксису атрибутов;

b) указание типа данных в НАС.1 — определении синтаксиса атрибутов;

c) определение подходящих правил для сопоставления предъявляемых значений с целевыми значениями атрибутов, хранящимися в ИБС. Для отдельного синтаксиса атрибутов может быть определено ни одного, некоторые или все из следующих правил сопоставлений:

i) равенство. Применимо к любому синтаксису атрибутов. Предъявленное значение должно соответствовать типу данных синтаксиса атрибута;

ii) подцепочка. Применимо к любому синтаксису атрибутов с типом данных, имеющих вид цепочки. Предъявленное значение должно быть последовательностью ("SEQUENCE OF"), каждый из элементов которой должен соответствовать типу данных;

iii) упорядочение. Применимо к любому синтаксису атрибутов, для которого может быть определено правило, позволяющее описать предъявленное значение по отношению к целевому значению в терминах "меньше чем", "равно", "больше чем". Предъявленное значение должно соответствовать типу данных синтаксиса атрибута.

9.6.2 Если правило сопоставления на равенство не определено, то Справочник:

a) трактует значения, рассматриваемые как атрибуты данного синтаксиса атрибутов, как имеющие тип ANY, то есть Справочник не проверяет, соответствуют ли эти значения типу данных, указанному для синтаксиса атрибутов;

b) не пытается сопоставить предъявленные значения целевым значениям такого типа атрибутов.

Примечание. — Отсюда следует, что Справочник не позволит ни использовать такие атрибуты в качестве выделенного имени, ни модифицировать такое специфическое значение.

9.6.3 Если правило сопоставления на равенство определено, то Справочник:

a) трактует значения атрибутов этого синтаксиса атрибутов как имеющие тип ANY DEFINED BY, то есть определенного типом данных, указанным в определении синтаксиса атрибутов;

b) выполняет сопоставление только в соответствии с правилами сопоставления, определенными для данного синтаксиса атрибутов;

c) выполняет сопоставление предъявленного значения, только имеющего подходящий тип данных, как это специфицированно в § 9.6.1 c).

9.6.4 Следующий макрос на НАС.1 может (но не обязательно), использоваться для определения синтаксисов атрибутов:

```
ATTRIBUTE-SYNTAX MACRO ::=
BEGIN

TYPE NOTATION  ::= Синтаксис
                   СопоставимыеТипы | empty

VALUE NOTATION ::=
    value (VALUE OBJECT IDENTIFIER)

Синтаксис ::= type

СопоставимыеТипы ::= "MATCHERS FOR" "Сопоставление | empty

Сопоставление ::= Сопоставим Сопоставление | Сопоставим

Сопоставим ::= "EQUALITY" | "SUBSTRINGS" | "ORDERING"

END
```

Соответствие между отдельными частями определения, перечисленными в § 9.6.1, и различными частями нотации, введенными макросом, следующее:

a) идентификатор объекта, присваиваемый синтаксису атрибутов, является значением, поставляемым присвоением value в макросе;

b) тип данных синтаксиса атрибута идентифицирован продукцией Синтаксис, то есть тем, что следует за именем макроса;

c) определяемые правила сопоставления суть "равенство", если в продукции СопоставимыеТипы присутствует "EQUALTY", "подцепочка", если присутствует "SUBSTRINGS", и "упорядочение", если присутствует "ORDERING". Если продукция пуста, то правила сопоставления не определяются.

Если в нотации выбирается альтернатива "empty", то результирующая нотация ("ATTRIBUTE-SYNTAX") может быть использована для обозначения любого возможного синтаксиса атрибутов.

Примечание 1. — Для фактического определения самих правил сопоставления не предусмотрено никаких средств в макросе. Это должно быть сделано на естественном языке или другим способом.

Примечание 2. — Макрос используется в определении выделенных синтаксисов атрибутов в Рекомендации X.520.

9.6.5 Синтаксис атрибутов, используемый в § 9.5.4, определен в § 9.6.5.1 и 9.6.5.2.

Примечание. — Это — частичное определение; фактически идентификаторы объектов для этих классов объектов размещены в Рекомендации X.520, с тем чтобы обеспечить единственное место, в котором в настоящей серии Рекомендации сгруппированы все определения.

9.6.5.1 СинтаксисИдентификатораОбъекта определяется следующим образом:

```
СинтаксисИдентификатораОбъекта ::=
    ATTRIBUTE-SYNTAX
       OBJECT IDENTIFIER
       MATCHES FOR EQUALITY
```

Правило сопоставления на равенство присуще определению идентификатора объекта типа на НАС.1.

9.6.5.2 СинтаксисВыделенногоИмени определяется следующим образом:

```
СинтаксисВыделенногоИмени ::=
    ATTRIBUTE-SYNTAX
    ВыделенноеИмя
    MATCHES FOR EQUALITY
```

Предъявленное значение выделенного имени равно целевому значению выделенного имени тогда и только тогда, когда истинны все следующие высказывания:

a) число ОВИ в обоих выделенных именах одинаково;

b) соответствующие ОВИ имеют одно и то же число ПЗА;

c) соответствующие ПЗА (то есть те, у которых идентичны типы атрибутов) имеют значения атрибутов, сопоставимые на равенство (в таком сопоставлении значения атрибутов выступают в той же роли — то есть как предъявленное, так и целевое значение, — что и содержащее их выделенное имя в общем сопоставлении).

РАЗДЕЛ 3 — *Модель безопасности*

10 Безопасность

10.1 Справочник существует в среде, в которой различные руководящие органы обеспечивают доступ к их фрагментам ИБС. Этот доступ должен быть согласован со стратегией безопасности (см. Рекомендацию X.509) той области безопасности, в которой находится фрагмент ИБС.

10.2 Здесь пойдет речь о двух специфических компонентах стратегии безопасности:

a) определении стратегии санкционирования;

b) определении стратегии аутентификации.

10.3 Определение санкционирования в контексте Справочника включает методы:

a) спецификации права доступа;

b) осуществления права доступа (управления доступом);

c) обслуживания права доступа.

10.4 Определение аутентификации в контексте Справочника включает методы проверки:

a) подлинности САС пользователя Справочника;

b) подлинности источника информации, полученной в пункте доступа.

Целостность полученной информации является локальным вопросом и должна быть согласована с осуществляемой стратегией безопасности.

10.5 Эта Рекомендация не определяет стратегии безопасности.

10.6 В Приложении F описаны общие подходы к спецификации права доступа.

10.7 В Рекомендации X.509 определены процедуры аутентификации. ПДС и СПС могут обеспечить строгую аутентификацию инициатора запроса по подписи запроса, целостность данных запроса путем подписывания запроса, а также строгую аутентификацию респондера и целостность данных результата путем подписывания результата. ПДС может обеспечить простую аутентификацию между АПС и САС. СПС может обеспечить простую аутентификацию между двумя САС.

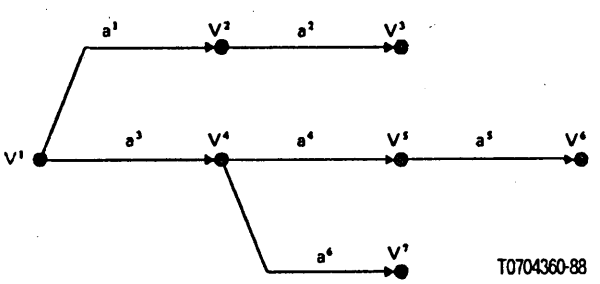
10.8 Административные органы прикладных задач, пользующиеся Справочником, могут применить свою собственную стратегию безопасности. Справочник может обеспечить прикладные задачи посредством хранения информации об аутентификации (например, выделенные имена, пароли, сертификаты) о взаимодействующих элементах. Подробнее это описано в Рекомендации X.509.

ПРИЛОЖЕНИЕ A

(к Рекомендации X.501)

Математическое описание деревьев

Настоящее Приложение не является составной частью стандарта.



Деревом является множество точек, называемых *вершинами,* и множество направленных отрезков, называемых *дугами;* каждая дуга ведет из вершины V в вершину V^1. Например, дерево на рисунке имеет семь вершин, помеченных V^1, .., V^7, и шесть дуг, помеченных a^1, .., a^6.

Говорят, что две вершины V и V^1 являются соответственно *начальной* и *конечной* вершинами дуги a, идущей из V в V^1. Например, V^2 и V^3 являются соответственно начальной и конечной вершинами дуги a^2. Несколько различных дуг могут иметь одну и ту же начальную вершину, но не одну и ту же конечную вершину. Например, дуги a^1 и a^3 имеют одну и ту же начальную вершину V^1, но никакие две дуги на рисунке не имеют одной и той же конечной вершины.

Вершину, не являющуюся конечной вершиной никакой дуги, часто называют *корневой* вершиной, или, не менее официально, "корнем" дерева. Например, на рисунке корнем является V^1.

Вершину, не являющуюся начальной вершиной никакой дуги, часто неофициально называют *листовой* вершиной, или, еще менее официально, "листом" древовидного графа. Например, вершины V^3, V^6, V^7 являются листьями.

Ориентированный путь из вершины V в вершину V^1 — это множество дуг $(a^1, a^2, .., a^n)$ $(n \geqslant 1)$, такое, что V является начальной вершиной дуги a^1, V^1 — конечной вершиной дуги a^n, а конечная вершина дуги a^k является также начальной вершиной дуги a^{k+1} для $1 \leqslant k < n$. Например, ориентированный путь из вершины V^1 в вершину V^6 представляет собой множество дуг (a^3, a^4, a^5). Термин "путь" следует понимать как обозначение ориентированного пути от корня к вершине.

ПРИЛОЖЕНИЕ B

(к Рекомендации X.501)

Использование идентификаторов объектов

Данное Приложение является составной частью стандарта.

Данное Приложение документирует верхние пределы тех поддеревьев идентификаторов объектов, в котором находятся все идентификаторы объектов, присвоенные в настоящей серии Рекомендаций. Это делается посредством введения НАС.1-модуля, называемого "ПолезныеОпределения", в котором всем узлам, не являющимся листьями поддерева, присвоены имена.

```
ПолезныеОпределения { joint-iso-ccitt ds (5) modules (1)
                      usefulDefinitions (0)}

DEFINITIONS ::=
BEGIN

EXPORTS
    модуль, элементСлужбы, прикладнойКонтекст, типАтрибута, синтаксисАтрибута, классОбъектов,
    алгоритм, абстрактныйСинтаксис, комплектАтрибутов,

    полезныеОпределения, структураИнформации, абстрактнаяСлужбаСправочника,
    идентификаторыОбъектовСправочника, идентификаторыОбъектовАлгоритма,
    распределенныеОперации, идентификаторыОбъектовПротокола, избранныеТипыАтрибутов,
    избранныеКлассыОбъектов, структураАутентификации, верхниеГраницы,
    пдс, спс

    ид-кп, ид-эпс, ид-ас, ид-от, ид-пт;

ds              OBJECT IDENTIFIER ::= { joint-iso-ccitt ds (5)}

-- категории информационных объектов --

модуль                  OBJECT IDENTIFIER ::= { ds 1}
элементСлужбы           OBJECT IDENTIFIER ::= { ds 2}
прикладнойКонтекст      OBJECT IDENTIFIER ::= { ds 3}
типАтрибута             OBJECT IDENTIFIER ::= { ds 4}
синтаксисАтрибута       OBJECT IDENTIFIER ::= { ds 5}
классОбъектов           OBJECT IDENTIFIER ::= { ds 6}
комплектАтрибутов       OBJECT IDENTIFIER ::= { ds 7}
алгоритм                OBJECT IDENTIFIER ::= { ds 8}
абстрактныйСинтаксис    OBJECT IDENTIFIER ::= { ds 9}
объект                  OBJECT IDENTIFIER ::= { ds 10}
порт                    OBJECT IDENTIFIER ::= { ds 11}
```

– – модули – –

```
полезныеОпределения                                  OBJECT IDENTIFIER ::= { module 0}
структураИнформации                                  OBJECT IDENTIFIER ::= { module 1}
абстрактнаяСлужбаСправочника                         OBJECT IDENTIFIER ::= { module 2}
распределенныеОперации                               OBJECT IDENTIFIER ::= { module 3}
идентификаторыОбъектовПротокола                      OBJECT IDENTIFIER ::= { module 4}
избранныеТипыАтрибутов                               OBJECT IDENTIFIER ::= { module 5}
избранныеКлассыОбъектов                              OBJECT IDENTIFIER ::= { module 6}
структураАутентификации                              OBJECT IDENTIFIER ::= { module 7}
идентификаторыОбъектовАлгоритма                      OBJECT IDENTIFIER ::= { module 8}
идентификаторыОбъектовСправочника                    OBJECT IDENTIFIER ::= { module 9}
верхниеГраницы                                       OBJECT IDENTIFIER ::= { module 10}
пдс                                                  OBJECT IDENTIFIER ::= { module 11}
спс                                                  OBJECT IDENTIFIER ::= { module 12}
идентификаторыОбъектовРаспределенногоСправочника     OBJECT IDENTIFIER ::= { module 13}
```

– – синонимы – –

```
ид-кп     OBJECT IDENTIFIER ::= контекстПрименения
ид-эпс    OBJECT IDENTIFIER ::= элементСлужбы
ид-ас     OBJECT IDENTIFIER ::= абстрактныйСинтаксис
ид-от     OBJECT IDENTIFIER ::= объект
ид-пт     OBJECT IDENTIFIER ::= порт

END
```

ПРИЛОЖЕНИЕ С

(к Рекомендации X. 501)

Структура информации на НАС.1

Данное Приложение является составной частью стандарта.

Данное Приложение содержит НАС.1-модуль "СтруктураИнформации", в который включены все НАС.1-определения типов, значений и макросов, введенные в настоящей Рекомендации.

```
СтруктураИнформации  { joint-iso-ccitt ds(5) modules(1)
                       informationFramework(1)}
DEFINITIONS ::=
BEGIN

EXPORTS
    Атрибут, ТипАтрибута, ЗначениеАтрибута, ПроверкаЗначенияАтрибута,
    ВыделенноеИмя, Имя, ОтносительноВыделенноеИмя,
    OBJECT-CLASS,ATTRIBUTE,ATTRIBUTE-SET,ATTRIBUTE-SYNTAX,
    Вершина, Псевдоним,
    КлассОбъектов, ИмяОбъектаПсевдонима,
    СинтаксисИдентификаторовОбъектов, СинтаксисВыделенногоИмени;

IMPORTS
    избранныеТипыАтрибутов, избранныеКлассыОбъектов
        FROM  ПолезныеОпределения { joint-iso-ccitt ds(5) modules (1)
              usefulDefinitions (0)}

    вершина
        FROM  ИзбранныеКлассыОбъектов  избранныеКлассыОбъектов
    синтаксисИдентификатораОбъекта,  синтаксисВыделенногоИмени, классОбъектов,
    имяОбъектаПсевдонима
        FROM  ИзбранныеТипыАтрибутов  избранныеТипыАтрибутов;
```

```
-- типы данных атрибутов --

Атрибут                    ::= SEQUENCE{
                                   type          ТипАтрибута
                                   value SET OF ЗначениеАтрибута
                                   -- требуется хотя бы одно значение --}

ТипАтрибута                ::= OBJECT IDENTIFIER

ЗначениеАтрибута           ::= ANY

ПроверкаЗначенияАтрибута   ::= SEQUENCE {ТипАтрибута, ЗначениеАтрибута}

-- именующие типы данных --

Имя                        ::= CHOICE {-- сейчас только одна возможность --
                                   ПоследовательностьОВИ}

ПоследовательностьОВИ      ::= SEQUENCE OF
                                   ОтносительноВыделенноеИмя

ВыделенноеИмя              ::= ПоследовательностьОВИ

ОтносительноВыделенноеИмя  ::= SET OF ПроверкаЗначенияАтрибута

-- макросы --

OBJECT-CLASS MACRO         ::=
BEGIN
    TYPENOTATION           ::= ПодклассИз ОбязательныеАтрибуты
                                   НеобязательныеАтрибуты
    VALUENOTATION          ::= value (VALUE OBJECT IDENTIFIER)

    ПодклассИз             ::= "SUBCLASS OF" Подклассы|empty
    Подклассы              ::= Подкласс|Подкласс "," Подклассы
    Подкласс               ::= value (OBJECT-CLASS)
    ОбязательныеАтрибуты   ::= "MUST CONTAIN {"Атрибуты"} " |empty
    НеобязательныеАтрибуты ::= "MAY CONTAIN {"Атрибуты"} " |empty

    Атрибуты               ::= ТермАтрибута | ТермАтрибута "," Атрибуты
    ТермАтрибута           ::= Атрибут | КомплектАтрибутов
    Атрибут                ::= value (ATTRIBUTE)
    Комплект Атрибутов     ::= value (ATTRIBUTE-SET)

END

ATTRIBUTE-SET MACRO        ::=

    BEGIN

    TYPE NOTATION          ::= "CONTAINS {"Атрибуты"} " |empty

    VALUE NOTATION         ::= value (VALUE OBJECT IDENTIFIER)

    Атрибуты               ::= ТермАтрибута | ТермАтрибута "," Атрибуты

    ТермАтрибута           ::= Атрибут | КомплектАтрибутов

    Атрибут                ::= value(ATTRIBUTE)
    КомплектАтрибутов      ::= value (ATTRIBUTE-SET)

END

ATTRIBUTE MACRO            ::=
BEGIN
TYPENOTATION               ::= СинтаксисАтрибута Многозначный|empty
VALUENOTATION              ::= value (VALUE OBJECT IDENTIFIER)

СинтаксисАтрибута          ::= "WITH ATTRIBUTE-SYNTAX" ВыборСинтаксиса

Многозначный               ::= "SINGLE VALUE"|"MULTI VALUE"|empty

ВыборСинтаксиса            ::= value (ATTRIBUTE-SYNTAX)
                               Ограничение | type СопоставимыеТипы
```

```
Ограничение                    ::= "("АльтернативноеОграничение")" | empty

АльтернативноеОграничение      ::= ОграничениеЦепочки|ОграничениеЦелого

ОграничениеЦепочки             ::= "SIZE" "("ОграничениеДлины")"

ОграничениеДлины               ::= ЕдинственноеЗначение | Диапазон

ЕдинственноеЗначение           ::= value(INTEGER)

Диапазон                       ::= value(INTEGER) ".."value(INTEGER)

ОграничениеЦелого              ::= Диапазон

СопоставимыеТипы               ::= "MATCHES FOR" Сопоставление | empty

Сопоставление                  ::= Сопоставим Сопоставление/Сопоставим

Сопоставим                     ::= "EQUALITY" | "SUBSTRINGS" | "ORDERING"

END


ATTRIBUTE-SYNTAX MACRO         ::=
BEGIN

TYPENOTATION                   ::= Синтаксис СопоставимыеТипы|empty
VALUENOTATION                  ::= value(VALUE OBJECT IDENTIFIER)

  Синтаксис                    ::= type

  СопоставимыеТипы             ::= "MATCHES FOR" "Сопоставление | empty
  Сопоставление                ::= Сопоставим Сопоставление | Сопоставим
  Сопоставим                   ::= "EQUALITY"|"SUBSTRINGS"|"ORDERING"

END


-- классы объектов --

Вершина        ::=   OBJECT-CLASS
                       MUST CONTAIN {классОбъектов}

Псевдоним      ::=   OBJECT-CLASS
                       SUBCLASS OF   Вершина
                       MUST CONTAIN  {имяОбъектаПсевдонима}

-- типы атрибутов --

КлассОбъектов  ::= ATTRIBUTE
                       WITH ATTRIBUTE-SYNTAX синтаксисИдентификатораОбъекта

ИмяОбъектаПсевдонима     ::= ATTRIBUTE
                                 WITH ATTRIBUTE-SYNTAX синтаксисВыделенногоИмени
                                 SINGLE VALUE


-- синтаксисы атрибутов --

СинтаксисИдентификатораОбъекта ::=
    ATTRIBUTE-SYNTAX
    OBJECT IDENTIFIER
    MATCHES FOR EQUALITY

СинтаксисВыделенногоИмени ::=
    ATTRIBUTE-SYNTAX
    ВыделенноеИмя
    MATCHES FOR EQUALITY

END
```

ПРИЛОЖЕНИЕ D

(к Рекомендации X.501)

Алфавитный список определений

Настоящее Приложение не является составной частью стандарта.

В данном Приложении перечисляются в алфавитном порядке все термины, определенные в настоящей Рекомендации, вместе со ссылками на параграф, в котором они определены.

ПРИЛОЖЕНИЕ E

(к Рекомендации X.501)

Критерий выработки имен

Настоящее Приложение не является составной частью стандарта.

Структура информации Справочника очень широка и допускает большое разнообразие статей и атрибутов ИДС. Согласно определению, имена тесно связаны с путями в ИДС, в силу чего возможно большое разнообразие имен. В настоящем разделе предлагается некоторый критерий, который следует учитывать при выработке имен. Этот критерий был применен при формировании рекомендуемых имен, используемых в Рекомендации X.521. Предлагается использовать этот критерий, где это окажется подходящим, для выработки имен объектов, для которых рекомендованные формы имен не были применены.

В настоящее время предлагается только один критерий: "удобный для пользователя".

Примечание. — Не все имена должны быть удобными для пользователя.

E.1 *Удобство пользователя*

Имена, с которыми человек имеет дело непосредственно, должны быть удобными для него. Имя, удобное для пользователя, — это такое имя, которое отражает точку зрения человека, а не компьютера. Оно должно быть легко запоминающимся, легко выводимым и понятным, а не таким, которое легко интерпретируемо компьютером.

Удобство пользователя может быть несколько точнее сформулировано в терминах двух следующих принципов:

— человек должен быть в состоянии правильно угадать удобное для него имя объекта на основании информации об объекте, которой, он, естественно, обладает. Например, должно быть угадываемо имя делового человека, если о нем имеется информация, случайно полученная в силу обычной деловой связи;

— если имя объекта специфицировано не однозначно, Справочник должен установить именно этот факт, а не то, идентифицирует ли имя какой-то конкретный объект. Например, если два человека имеют одинаковые фамилии, то только фамилия рассматривается как неадекватная идентификация того или другого лица.

Из задачи удобства пользователя вытекают следующие подзадачи:

a) имена не должны искусственным образом устранять естественную неоднозначность. Например, если два человека имеют одну и ту же фамилию "Джонс", то нельзя требовать, чтобы они отвечали на запрос "УДжонс" или "Джонс2". Вместо этого договоренность о присвоении имен должна предусматривать удобные для человека средства различения элементов. Например, можно потребовать имя и отчество в дополнение к фамилии;

b) имена должны допускать общепринятые сокращения и обычные вариации в орфографии. Например, если некто работает в Конвей Стил Корпорейшн и если это наименование должно фигурировать в имени работника, то имена "Конвей Стил Корпорейшн", "Конвей Стил Корп.", "Конвей Стил" и "КСК" должны быть достаточны для идентификации этой организации;

c) в некоторых случаях при обращении к поиску индивидуальной статьи могут использоваться псевдонимы, для того чтобы быть более удобными для человека или чтобы сократить область поиска. Последующий пример демонстрирует использование в этих целях псевдонима, как показано на рис. E-1/X.501. Филиал предприятия, находящийся в городе Осака, может быть идентифицирован также именем С-Япония, М-Осака, О-АВС, ПО-Филиал-в-Осаке;

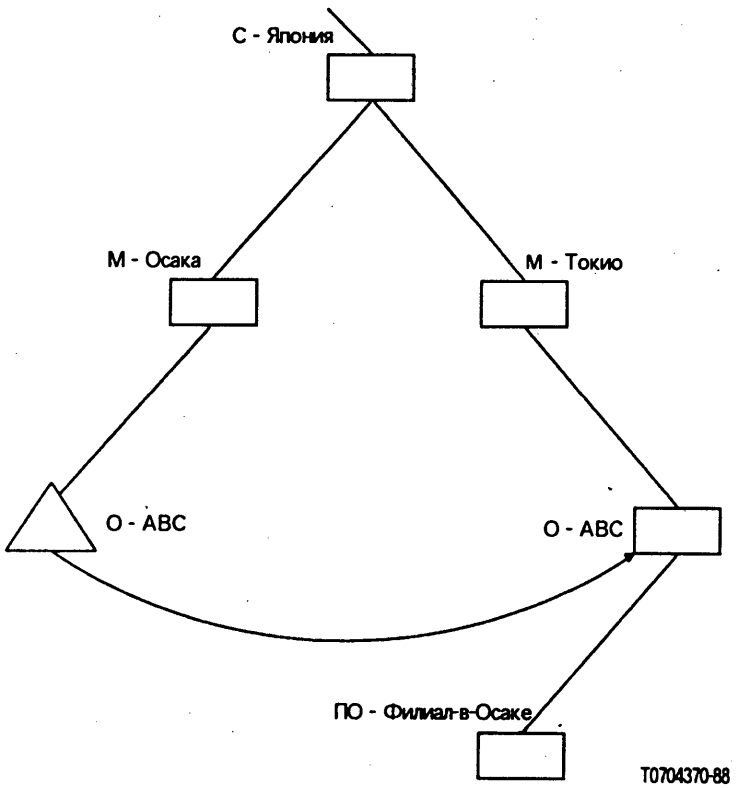


РИСУНОК Е-1/X.501

Пример использования псевдонима

d) если имя состоит из многих частей, то число как обязательных, так и необязательных частей должно быть относительно невелико и поэтому легко запоминающимся;

e) если имя состоит из нескольких частей, то, как правило, порядок их следования должен быть несущественен;

f) удобные для пользователя имена не должны содержать компьютерных адресов.

ПРИЛОЖЕНИЕ F

(к Рекомендации X.501)

Управление доступом

Настоящее Приложение не является составной частью стандарта.

F.1 *Введение*

Пользователи Справочника получают доступ к информации в ИБС на основе их права доступа к информации в соответствии с используемой стратегией управления доступом, защищающей эту информацию.

В настоящей серии Рекомендаций управление доступом оставлено на локальное усмотрение. Однако следует иметь в виду, что реализация Справочника требует каких-то средств управления доступом и что в последующих вариантах Рекомендаций этой серии средства создания, поддержания и применения информации, управляющей доступом, будут, по-видимому, стандартизованы. Поэтому в настоящем Приложении описываются принципы, лежащие в основе управления доступом, и выделяются два возможных подхода к управлению доступом.

F.2 *Принципы*

Двумя принципами, которыми следует руководствоваться в создании процедур управления доступом, являются следующие:

a) должны иметься средства защиты информации, имеющейся в Справочнике, от несанкционированного обнаружения, проверки и изменения, включая защиту ИДС от несанкционированных модификаций;

b) информация, необходимая для установления права пользователя на выполнение некоторой операции, должна быть доступна тем САС, которые участвуют в выполнении данной операции, чтобы избежать выполнения других удаленных операций, единственным назначением которых будет проверка этих прав.

F.3 *Защищаемые элементы*

В настоящее время определены следующие уровни защиты:

a) защита целых поддеревьев в ИДС;

b) защита отдельных статей в ИДС;

c) защита всего атрибута внутри одной статьи;

d) защита отдельных экземпляров значений атрибутов.

F.4 *Категории доступа*

Предусматривается необходимость в по крайней мере пяти категориях доступа. Если доступ к защищенному элементу не разрешен в любой из категорий, то Справочник, насколько это возможно, отвечает так, как будто защищенные элементы просто не существуют.

Эти категории доступа сведены в таблицу F-1/X.501. Колонка "элемент" указывает, является ли защищаемое средство статьей (C), атрибутом (A) или ими обоими (CA).

ТАБЛИЦА F-1/X.501

Категории доступа

Категория	Элемент	Описание
Обнаружение	A	Позволяет обнаружить защищаемый элемент.
Сравнение	A	Позволяет сравнивать предъявленное значение с защищаемым элементом.
Чтение	A	Позволяет прочесть защищаемый элемент.
Модификация	A	Позволяет обновить защищаемый элемент.
Добавление/удаление	CA	Позволяет создавать и удалять новые компоненты (атрибуты или значения атрибутов) в защищаемом элементе.
Именование	C	Позволяет модифицировать относительно выделенное имя, а также удалить и создать статью, непосредственно следующую за защищаемым элементом.

F.5 *Определение права доступа*

Одна из схем организации управления доступом явно или неявно связывает каждый защищаемый элемент со списком прав доступа. Каждая строка такого списка сводит в пары набор пользователей с набором категорий доступа.

Определение принадлежности пользователя к одному (или более) из указанных наборов должно быть возможным, исходя из информации, поставляемой вместе с запросом,— либо из установленной подлинности и удостоверения пользователя, поставляемых в BIND, либо из информации, находящейся в аргументе операции.

Имеются по крайней мере две возможности:

a) наборы описываются в терминах выделенных имен идентифицируемых ими пользователей — либо выделенных имен самих пользователей, либо выделенных имен непосредственно предшествующих объектов, флажок которых специфицирует включение всего поддерева;

b) наборы предоставляют только возможности и неявно включают всех пользователей, имеющих эту возможность. Эта схема требует, чтобы возможности пользователя были либо локально доступны, либо доставлялись в BIND, или находились в аргументе операции. Последнее может потребовать расширения определенных в настоящее время протоколов.

Рекомендация X.509

СПРАВОЧНИК — СТРУКТУРА АУТЕНТИФИКАЦИИ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

[1)] Рекомендация X.509 и Стандарт ISO-9594-8 "Системы обработки информации – Взаимосвязь открытых систем – Справочник – Структура аутентификации" были разработаны в тесном сотрудничестве и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой ими справочной информацией может рассматриваться как объединенное целое, называемое *Справочником*. Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника, обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные-элементы ВОС, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых разными производителями;

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 Многие приложения предъявляют требования к безопасности для защиты информации от опасностей во время передачи. Несколько распространенных опасностей совместно со службами безопасности и механизмами, которые могут быть использованы для защиты от этих опасностей, кратко описываются в Приложении А. Фактически все службы безопасности зависят от надежного распознавания тождественности взаимодействующих сторон, то есть основываются на аутентификации.

0.4 В настоящей Рекомендации устанавливается структура служб аутентификации, которые Справочник поставляет своим пользователям. В число этих пользователей включается сам Справочник, равно как другие приложения и службы. Справочник может быть с большой пользой вовлечен в удовлетворение потребностей этих пользователей в аутентификации и в прочих службах безопасности, так как Справочник является тем естественным местом, из которого взаимодействующие стороны могут извлечь сведения об аутентификации друг друга: знания, которые являются основой установления подлинности. Справочник является естественным местом, так как в нем хранится прочая информация, требующаяся для связи и извлекаемая еще до того, как осуществляется связь. Таким образом, при этом подходе извлечение из Справочника аутентифицирующей информации партнера по связи становится подобным извлечению его адреса. Учитывая широкое обращение к Справочнику для нужд связи, можно допустить, что описываемая структура аутентификации будет широко использоваться приложениями из большого диапазона.

1 Предмет рассмотрения и область применения

1.1 В настоящей Рекомендации:

- специфицируется форма аутентифицирующей информации, хранящейся в Справочнике;
- описывается, как эта аутентифицирующая информация может быть извлечена из Справочника;
- формулируются предположения о том, как формируется и помещается в Справочник аутентифицирующая информация;
- описываются три метода, согласно которым пользователи могут воспользоваться этой аутентифицирующей информацией для выполнения аутентификации; описывается также метод использования аутентификации для обеспечения прочих служб безопасности.

1.2 В настоящей Рекомендации описываются два уровня аутентификации: простая аутентификация, использующая пароль для подтверждения заявляемой подлинности, и строгая аутентификация, использующая удостоверения, созданные с помощью криптографической техники. Простая аутентификация обеспечивает сравнительно ограниченные средства защиты от несанкционированного доступа. Поэтому в качестве основы обеспечения служб безопасности требуется использование строгой аутентификации. Этот подход не рассматривается как образующий единственно возможную общую структуру средств аутентификации, однако он может быть общеполезным для приложений, считающих его адекватным их нуждам.

1.3 Аутентификация (как и прочие службы безопасности) может быть обеспечена только в контексте выработанной стратегии безопасности. Пользователи отдельных приложений могут выработать свою стратегию безопасности, которая, однако, может быть ограничена службами, обеспечиваемыми стандартом.

1.4 Если некоторое приложение использует предлагаемую структуру аутентификации, то стандарты, определяющие эти приложения, должны сами специфицировать протокол таких обменов, которые надлежит осуществить, с тем чтобы обеспечить аутентификацию, опирающуюся на аутентифицирующую информацию, извлекаемую из Справочника. Протоколом, используемым приложениями для извлечения удостоверений из Справочника, является Протокол доступа к Справочнику (ПДС), специфицированный в Рекомендации X.519.

1.5 Метод строгой аутентификации, специфицируемый в настоящей Рекомендации, основан на криптосистеме с общедоступными ключами. Огромным достоинством таких систем является то, что удостоверения могут храниться в самом Справочнике в качестве атрибутов и могут легко пересылаться внутри Справочника. Пользователи Справочника могут извлекать эти удостоверения тем же методом, что и прочую информацию, хранящуюся в Справочнике. Предполагается, что сертификаты пользователей формируются "офлайновыми" средствами; в Справочник их помещает тот, кто их формирует. Генерация сертификатов пользователей осуществляется некоторым офлайновым сертификатным руководящим органом, который полностью независим от всех САС Справочника. В частности, к поставщикам Справочника не предъявляется никаких требований по размещению и пересылке удостоверений безопасным способом.

Краткое введение в криптографию с общедоступными ключами может быть найдено в Приложении B.

1.6 Вообще говоря, структура криптографии не зависит от использования конкретного криптографического алгоритма при условии, что алгоритм удовлетворяет требованиям, описанным в §6.1. Потенциально может использоваться несколько различных алгоритмов. Однако, если два пользователя хотят иметь возможность устанавливать подлинность друг друга, то они должны обеспечивать один и тот же криптографический алгоритм, чтобы аутентификация выполнялась корректно. Таким образом, в контексте некоторого комплекта связанных приложений выбор одного какого-нибудь алгоритма способствует максимальному расширению сообщества пользователей, могущих убедиться в подлинности друг друга и безопасно обмениваться информацией. Один пример криптографического алгоритма с общедоступными ключами может быть найден в Приложении C.

1.7 Подобно этому, два пользователя, желающие аутентифицировать друг друга, должны обеспечивать одну и ту же хэш-функцию (см. § 3.3 f)), используемую при формировании удостоверений и аутентификационных мандатов. И здесь, в принципе, может использоваться несколько альтернативных хэш-функций, но в силу этого будет сужаться сообщество пользователей, которые могут убедиться в подлинности друг друга. Краткое введение в теорию хэш-функций и пример одной такой функции могут быть найдены в Приложении D.

2 Библиография

2.1 ISO 7498-2 "Системы обработки информации — Взаимосвязь открытых систем — Архитектура безопасности".

3 Определения

3.1 В настоящей Рекомендации используются определенные в эталонной модели ВОС, часть 2, "Безопасность", нижеследующие общие термины, касающиеся обеспечения безопасности:

a) *несимметричная* (шифровка);

b) *обмен аутентификацией;*

c) *аутентифицирующая информация;*

d) *секретность;*

e) *удостоверение;*

f) *криптография;*

g) *аутентификация источника данных;*

h) *дешифровка, расшифровка, расшифровывание;*

i) *шифровка, зашифровывание;*

j) *ключ;*

k) *пароль;*

l) *аутентификация равноуровневого элемента;*

m) *симметричная* (шифровка).

3.2 Нижеследующие термины, используемые в настоящей Рекомендации, определены в Рекомендации X.501:

a) *атрибут;*

b) *информационная база Справочника;*

c) *информационное дерево Справочника;*

d) *выделенное имя;*

e) *статья;*

f) *объект;*

g) *корень.*

3.3 Нижеследующие специфичные термины определены и используются в настоящей Рекомендации:

a) *аутентификационный мандат (мандат)* – информация, передаваемая в процессе обмена строгой аутентификацией; может быть использована для аутентификации ее отправителя;

b) *сертификат пользователя (сертификат)* – общедоступные ключи пользователя в сококупности с некоторой дополнительной информацией, вырабатываемой (без возможности подделки) в процессе шифровки с помощью секретного ключа, выданного сертификатным органом;

c) *сертификатный руководящий орган* – орган, которому один или несколько пользователей доверили разработку и выдачу сертификатов; возможен вариант, при котором сертификатный руководящий орган вырабатывает ключи пользователя;

d) *ветвь сертификации* – упорядоченная последовательность сертификатов объектов в ИДС, обладающая тем свойством, что, исходя из общедоступного ключа начального объекта этой ветви и в результате обработки этой ветви, может быть получен общедоступный ключ конечного объекта этой ветви;

e) *криптографическая система, криптосистема* – совокупность преобразований открытого текста в криптограмму и наоборот; при этом конкретное (ные) преобразование (ия), подлежащее (щие) использованию, выбирается ключом; обычно преобразования определяются с помощью математического алгоритма;

f) *хэш-функция* – (математическая) функция, отображающая значения из большой (возможно, очень большой) области в меньший диапазон значений; хэш-функция считается "хорошей", если результаты ее применения к (большому) набору значений будут равномерно (и, очевидно, случайным образом) распределены по диапазону;

g) *однонаправленная функция* — (математическая) функция f, которую легко вычислить, но для которой трудно найти такое x из области определения функции, для которого $f(x) = y$, где y — любое значение из области значений функции; допускается лишь небольшое число таких y, для которых указанное уравнение легко разрешимо;

h) *общедоступный ключ* — (в криптосистеме с общедоступными ключами) тот ключ из парных ключей пользователя, который общеизвестен;

i) *частный ключ (секретный ключ – отвергнут)* — (в криптосистеме с общедоступными ключами) тот ключ из парных ключей пользователя, который известен только этому пользователю;

j) *простая аутентификация* — аутентификация, основанная на простых соглашениях о паролях;

k) *стратегия безопасности* — совокупность правил, устанавливаемая руководящим органом безопасности, управляющим поставкой и обеспечением служб и средств безопасности;

l) *строгая аутентификация* — аутентификация, основанная на криптографически выведенных удостоверениях;

m) *доверие* — в общем случае говорят, что один элемент испытывает "доверие" к другому, если первый элемент считает, что второй будет вести себя именно так, как предполагает первый; это доверие может касаться какой-то специфической функции; ключевая роль доверия в структуре аутентификации заключается в описании взаимоотношений между аутентифицирующим элементом и сертификатным органом; аутентифицирующий элемент должен быть уверен в том, что сертификатный руководящий орган обеспечит его действительными и надежными сертификатами;

n) *порядковый номер сертификата* — целое число, уникальное в рамках некоторого CA, который однозначно ассоциируется с сертификатом, выработанным данным CA.

4 Обозначения и сокращения

4.1 Обозначения, используемые в настоящей Рекомендации, определены в таблице 1/X.509, ниже.

Примечание. — В этой таблице символы X, X_1, X_2 и т.д. используются в качестве имен пользователей, а символ I заменяет произвольную информацию.

4.2 В настоящей Рекомендации используются нижеследующие сокращения:

CA — сертификатный руководящий орган

ИБС — информационная база Справочника

ИДС — информационное дерево Справочника

КОДК — криптосистема с общедоступными ключами

ТАБЛИЦА 1/X.509

Обозначения

ОБОЗНАЧЕНИЕ	ЗНАЧЕНИЕ
Xp	Общедоступный ключ пользователя X.
Xs	Секретный ключ пользователя X.
Xp[I]	Шифровка некоторой информации I с помощью общедоступного ключа пользователя X.
Xs[I]	Шифровка некоторой информации I с помощью секретного ключа пользователя X.
X{I}	Подписывание информации I пользователем X. Состоит из I с присоединенным зашифрованным резюме.
CA (X)	Сертификатный руководящий орган пользователя X.
CA^n (X)	(где n>1) : CA (CA (... n раз ... (x))).
X_1«X_2»	Сертификат пользователя X_2, выданный сертификатным органом X_1.
X_1«X_2»X_2«X_3»	Цепочка сертификатов (может иметь произвольную длину), в которой каждый элемент является сертификатом того сертификатного органа, который выработал следующий сертификат. Функционально эта цепочка эквивалентна сертификату X_1«X_{n+1}». Например, обладание цепочкой A«B»B«C» обеспечивает те же возможности, что и A«C», а именно получение Cp по заданному Ap.
$X_{1p} \cdot X_1$«X_2»	Операция по развертыванию сертификата (или цепочки сертификатов) в целях извлечения общедоступного ключа. Это инфиксная операция, левым операндом которой является общедоступный ключ сертификатного органа, а правым – сертификат, выданный этим сертификатным органом. Результатом операции является общедоступный ключ пользователя, сертификатом которого является правый операнд. Например, выражение Ap · A«B»B«C» обозначает операцию использования общедоступного ключа пользователя A для получения общедоступного ключа Bp пользователя B из сертификата пользователя B, сопровождаемую использованием Bp для развертывания сертификата пользователя C. Результатом совокупной операции является общедоступный ключ Cp пользователя C.
A → B	Ветвь сертификации от A к B, образованная цепочкой сертификатов, начиная с CA (A) «CA^2 (A)» и кончая CA (B) «B».

5 Процедура простой аутентификации

5.1 Простая аутентификация предназначена для обеспечения локальной проверки санкционированности, основанной на выделенном имени пользователя, двусторонне согласованном (необязательном) пароле и двустороннем понимании средств использования и обработки этого пароля в пределах одной области. Применение простой аутентификации предполагает, в основном, только локальное использование, то есть аутентификацию равноуровневых элементов между одним АПС и одним САС или между одним САС и одним САС. Простая аутентификация может быть осуществлена различными способами:

a) открытой (незащищенной) передачей выделенного имени пользователя и (необязательного) пароля получателю для последующего распознавания;

b) передачей выделенного имени пользователя, пароля пользователя, случайного числа и/или штемпеля времени, причем все передаваемые данные защищены применением однонаправленной функции;

c) передачей защищенной информации, описанной в пункте b), совместно со случайным числом и/или штемпелем времени, причем все эти данные защищены применением однонаправленной функции.

Примечание 1. — Однонаправленные функции не обязаны быть различными.

Примечание 2. — Поступление сообщений о новых процедурах по защите паролей может стать предметом расширения настоящего Документа.

5.2 Если пароли не защищены, то предотвращение несанкционированного доступа обеспечивается минимальным уровнем безопасности, который не должен рассматриваться как основа служб безопасности. Защита выделенного имени и пароля пользователя обеспечивает более высокий уровень безопасности. Алгоритмами, используемыми в качестве механизмов защиты, являются обычно однонаправленные функции (а не методы шифровки), которые весьма просты в употреблении.

5.3 Общая процедура обеспечения простой аутентификации изображена на рис. 1/X.509.

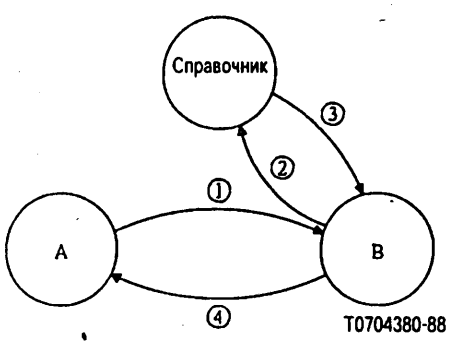


РИСУНОК 1/X.509

Процедура простой аутентификации без защиты

5.3.1 Предпринимаются следующие шаги:

1) начинающий пользователь A посылает свои выделенное имя и пароль получателю B;

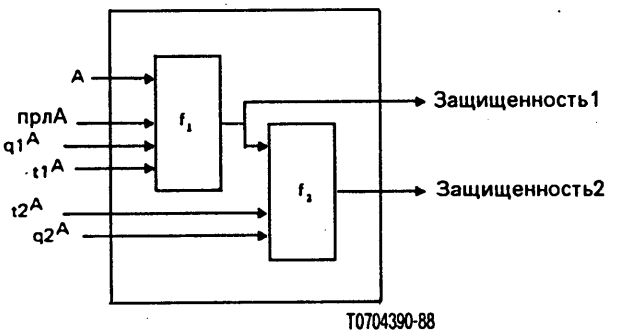
2) пользователь B посылает Справочнику предполагаемые выделенные имя и пароль пользователя A; в Справочнике пароль сопоставляется с тем паролем, который хранится в качестве атрибута Пароль пользователя статьи пользователя A в Справочнике (с помощью операции Справочника "Сравнение");

3) Справочник подтверждает (или отрицает) пользователю B действительность удостоверений;

4) успешность (или неуспешность) аутентификации может быть сообщена пользователю A.

5.3.2 Самая основная форма простой аутентификации содержит только шаг 1), и, после того, как B проверит выделенное имя и пароль, может содержать шаг 4).

5.4 Рис. 2/X.509 иллюстрирует два подхода к выработке защищенной идентифицирующей информации. Здесь f_1 и f_2 являются однонаправленными функциями (совпадающими или различными); штемпель времени и случайные числа не являются обязательными, и их использование зависит от двустороннего соглашения.



A — выделенное имя пользователя

t^A — штемпель времени

прл^A — пароль пользователя A

q^A — случайные числа, возможно, с включенным счетчиком.

РИСУНОК 2/X.509

Простая аутентификация с защитой

5.4.1 Рис. 3/X.509 иллюстрирует процедуру простой аутентификации с защитой.

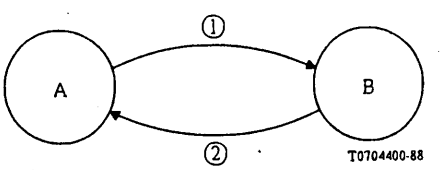


РИСУНОК 3/X.509

Процедура простой аутентификации с защитой

Предпринимаются следующие шаги (вначале используя только f_1):

1) Начинающий пользователь (пользователь A) посылает свою защищенную идентифицирующую информацию (Аутентификатор1) пользователю B. Защита обеспечивается применением однонаправленной функции (f_1), изображенной на рис. 2/X.509, где штемпель времени и/или случайное число (если они используются) должны минимизировать розыгрыш и скрыть пароль.

   Защита пароля пользователя A имеет форму:

   Защищенность1 = f_1 ($t1^A$, $q1^A$, прлA).

   Информация, посылаемая пользователю B, имеет форму:

   Аутентификатор1 = $t1^A$, $q1^A$, A, Защищенность1.

Пользователь В проверяет защищенную идентифицирующую информацию, представленную пользователем А. Для этого он генерирует (используя штемпель времени, выделенное имя и, возможно, дополнительные штемпель времени и/или случайное число, предоставленные пользователем А наряду с локальной копией пароля А) локальную защищенную копию пароля пользователя А (в форме Защищенность1). После этого пользователь В сравнивает (на равенство) предполагаемую идентифицирующую информацию (Защищенность1) с локально выработанным значением.

2) Пользователь В подтверждает (или отрицает) пользователю А успешность верификации защищенной идентифицирующей информации.

5.4.2 Процедура, описанная в § 5.4.1, может быть расширена, обеспечивая тем самым большую защищенность (использованием f_1 и f_2).

Основные различия заключаются в следующем:

1) А посылает свою (дополнительно) защищенную идентифицирующую информацию (Аутентификатор2) пользователю В. Дополнительная защищенность достигается дальнейшим применением однонаправленной функции f_2, как это показано на рис. 2/X.509. Эта дополнительная защищенность имеет форму:

Защищенность2 $= f_2(t2^A, q2^A,$ Защищенность1).

Информация, посылаемая пользователю В, имеет форму:

Аутентификатор2 $= t1^A, t2^A, q1^A, q2^A,$ А, Защищенность2.

Для сравнения пользователь В генерирует локальное значение дополнительно защищенного пароля пользователя А и сравнивает его (на равенство) со значением Защищенность2. (В принципе этот шаг аналогичен шагу 1) § 5.4.1.).

2) Пользователь В подтверждает (или отрицает) пользователю А успешность верификации защищенной идентифицирующей информации.

Примечание. — Процедуры, определенные в этом параграфе, специфицированы в терминах А и В. В применении к Справочнику (специфицированном в Рекомендации X.511 и X.518) в качестве А может выступать АПС, привязанный к САС, выступающему в роли В, или в качестве А может выступать САС, привязанный к другому САС, выступающему в роли В.

5.5 Тип атрибута пароль пользователя содержит пароль объекта. Значением пароля пользователя атрибута пароль пользователя является цепочка, специфицируемая объектом.

```
паральПользователя ::= ATTRIBUTE
                       WITH ATTRIBUTE-SYNTAX
                            OCTET STRING (SIZE (0.. вг-пароля-пользователя))
                            MATCHES FOR EQUALITY
```

5.6 Нижеследующий НАС.1-макрос может быть использован для определения типа данных, возникающих при применении однонаправленной функции к заданному другому типу данных.

```
PROTECTED MACRO ::= SIGNATURE
```

РАЗДЕЛ 2 — *Строгая аутентификация*

6 Основы строгой аутентификации

6.1 Подход к строгой аутентификации, принятый в настоящей Рекомендации, использует свойства семейства криптографических систем, известных под названием криптосистем с общедоступными ключами (КОДК). Эти криптосистемы, называемые также несимметричными в отличие от обычных криптосистем, использующих один ключ, используют парные ключи, из которых один — секретный, а другой — общедоступный. В Приложении В дается краткое введение в такие системы и описываются те их свойства, которые делают их полезными в аутентификации. В настоящее время КОДК может быть полезной для описываемой структуры установления подлинности, если она обладает нижеследующим свойством: оба ключа из ее парных ключей могут быть использованы в шифровальных целях, причем если шифровка осуществлялась общедоступным ключом, то дешифровка должна осуществляться секретным, и, наоборот, если шифровка осуществлялась секретным ключом, то дешифровка должна осуществляться общедоступным ключом. Иными словами, $X_P \cdot X_S = X_S \cdot X_P$, где X_P/X_S являются функциями шифровки/дешифровки, использующими общедоступный/секретный ключи пользователя X.

Примечание. — Альтернативные типы КОДК, то есть такие, для которых не требуется указанное свойство перестановки и которые могут быть обеспечены без больших изменений настоящей Рекомендации, являются предметом возможных дальнейших исследований системы.

6.2 Настоящая структура аутентификации не требует использования какой-то конкретной криптосистемы. Предполагается, что описываемая структура будет применима к любой подходящей криптосистеме с общедоступным ключом и, следовательно, сможет поддержать те изменения используемых методов, которые окажутся результатом будущих развитий в криптографии, в математической технике и в вычислительных возможностях. Однако если два пользователя хотят уметь аутентифицировать друг друга, то они должны использовать один и тот же криптографический алгоритм, чтобы аутентификация выполнялась корректно. Таким образом, в контексте некоторого набора связанных приложений выбор одного какого-нибуль алгоритма будет способствовать максимальному расширению того сообщества пользователей, которые смогут аутентифицировать друг друга и взаимодействовать безопасным образом. Один пример криптографической системы может быть найден в Приложении С.

6.3 Аутентификация опирается на наличие у каждого пользователя уникального имени, отличного от имен всех остальных пользователей. Присвоение различных имен находится в компетенции Руководящего органа по распределению имен. Поэтому каждый пользователь должен доверять этому органу в том, что оно не повторит дважды одного и того же выделенного имени.

6.4 Каждый пользователь идентифицируется тем, что он владеет своим секретным ключом. Другой пользователь имеет возможность определить, владеет ли его партнер по связи секретным ключом, и может использовать это в качестве подтверждения того, что его партнер по связи действительно является предполагаемым пользователем. Надежность этого подтверждения зависит от секретного ключа, который остается тайной пользователя.

6.5 Чтобы пользователь мог определить, владеет ли его партнер по связи секретным ключом некоторого третьего пользователя, он сам должен владеть общедоступным ключом этого третьего пользователя. Получение значения этого общедоступного ключа из статьи пользователя в Справочнике достаточно просто, однако проверка его корректности более проблематична. Существует много различных путей осуществления этого: § 7 описывает процесс, посредством которого общедоступный ключ пользователя может быть проверен обращением к Справочнику. Этот процесс может функционировать только в том случае, если в Справочнике существует цепь без разрывов из пользующихся доверием точек между пользователями, требующими аутентификации. Такая цепь может быть создана идентификацией общей пользующейся доверием точки. Эта общая пользующаяся доверием точка должна быть связана с каждым из пользователей цепочкой без разрывов из пользующихся доверием точек.

7 Извлечение общедоступного ключа пользователя

7.1 Чтобы пользователь мог доверять процессу аутентификации, он должен извлекать общедоступный ключ другого пользователя из источника, которому он доверяет. Такой источник, называемый сертификатным руководящим органом (CA), для выработки сертификата некоторого пользователя использует алгоритм общедоступного ключа, с помощью которого этот орган, вырабатывая сертификат, вырабатывает общедоступный ключ, включаемый в *сертификат.* Сертификат, форма которого специфицируется в § 7.2, обладает следующими свойствами:

— каждый пользователь, имеющий доступ к общедоступному ключу сертификатного органа, может выделить общедоступный ключ, включенный в сертификат;

— ни одна сторона, помимо сертификатного ведомства, не может изменить сертификат так, чтобы это не было обнаружено (сертификаты нельзя подделать).

Так как сертификаты не могут быть подделаны, то их можно опубликовать, поместив в Справочник, не требуя от последнего специальных усилий для защиты этих сертификатов.

Примечание. — Хотя все CA однозначно определены выделенными именами в ИДС, из этого отнюдь не следует, что существует какая бы то ни было связь между организацией CA и ИДС.

7.2 Сертификатный руководящий орган вырабатывает сертификат пользователя подписыванием (см. § 8) комплекта информации, включающего выделенное имя пользователя и общедоступный ключ. Конкретно сертификат, выработанный сертификатным органом CA для пользователя, выделенное имя которого A, имеет следующую форму:

$$CA\,\langle\langle A\rangle\rangle = CA\{SN, AI, CA, A, Ap, T^A\},$$

где через SN обозначен порядковый номер сертификата, через AI — идентификатор алгоритма, используемого для подписывания сертификата, а через T^A — период годности сертификата. Этот период состоит из двух дат: начала и конца периода годности сертификата. Так как предполагается, что T^A будет меняться по периодам,

не меньшим, чем 24 часа, то предполагается, что система будет использовать время от Гринвича в качестве основы эталонного времени. Годность подписи в сертификате может быть проверена любым пользователем, знающим CA. Следующий НАС.1-тип данных может быть использован для представления сертификата.

```
Сертификат  ::=        SIGNED SEQUENCE{
    версия                              [0] Версия  DEFAULT 1988,
    порядковыйНомер                     ПорядковыйНомер,
    подпись                             ИдентификаторАлгоритма,
    источник                            Имя,
    годность                            Годность,
    субъект                             Имя,
    инфоОбщедостКлючаСубъекта           ИнфоОбщедостКлючаСубъекта }

Версия              ::= INTEGER   1988(0)
ПорядковыйНомер     ::= INTEGER

Годность            ::=
    SEQUENCE{
        нераньше                    ВремяОтГринвича,
        непозже                     ВремяОтГринвича}

ИнфоОбщедостКлючаСубъекта   ::=
    SEQUENCE{
        алгоритм                    ИдентификаторАлгоритма,
        общедостКлючСубъекта        BIT STRING}

ИдентификаторАлгоритма      ::=
    SEQUENCE{
        алгоритм                    OBJECT IDENTIFIER,
        параметры                   ANY DEFINED BY алгоритм
                                      OPTIONAL }
```

7.3 Статья Справочника каждого пользователя A, являющегося участником строгой аутентификации, содержит сертификат(ы) этого пользователя. Этот сертификат вырабатывает сертификатный руководящий орган пользователя A, являющийся элементом дерева ИДС. Сертификатный орган пользователя A, который может быть и не единственным, обозначается через CA(A) или просто CA, если A однозначно понимаем. Общедоступный ключ пользователя A может быть обнаружен любым пользователем, знающим общедоступный ключ органа CA. Таким образом, обнаружение общедоступных ключей рекурсивно.

7.4 Если пользователь A, старающийся обнаружить общедоступный ключ пользователя B, уже обнаружил общедоступный ключ органа CA(B), то процесс завершен. Чтобы дать возможность пользователю A извлечь общедоступный ключ органа CA(B), статья Справочника каждого сертификатного руководящего органа X содержит несколько сертификатов. Эти сертификаты двух типов. К первому типу относятся сертификаты для продвижения вперед; эти сертификаты выработали другие сертификатные руководящие органы. Ко второму типу относятся сертификаты для продвижения назад; эти сертификаты выработал сам орган X; они являются подтвержденными общедоступными ключами других сертификатных органов. Наличие этих сертификатов позволяет пользователям создавать ветвь сертификации от одной точки до другой.

7.5 Список сертификатов, которые необходимы для того, чтобы некоторый конкретный пользователь мог извлечь общедоступный ключ другого пользователя, называется *ветвью сертификации*. Каждый элемент этого списка является сертификатом сертификатного руководящего органа следующего элемента списка. Ветвь сертификации от A к B (обозначаемая A→B):

— начинается сертификатом, выработанным CA(A), а именно $CA(A)\langle\langle X^1\rangle\rangle$ некоторого элемента X^1;

— продолжается по следующим сертификатам $X^i\langle\langle X^{i+1}\rangle\rangle$;

— заканчивается сертификатом пользователя B.

Ветвь сертификации логически образует в информационном дереве Справочника цепь без разрывов из пользующихся доверием точек. Эта цепь соединяет двух пользователей, желающих принять участие во взаимной аутентификации. Конкретный метод, используемый пользователями A и B для получения ветвей сертификации A→B и B→A, может изменяться. Один из подходов, способствующих построению ветви сертификации, заключается в иерархическом упорядочении CA, причем эта иерархия может совпадать, а может и не совпадать с иерархией всего ИДС или его части. Преимуществом этого является то, что те пользователи, которые имеют свои CA в этой иерархии, могут установить между собой ветвь сертификации, используя сам Справочник и не нуждаясь ни в какой предварительной информации. Чтобы предусмотреть это, каждое CA должно хранить один сертификат для продвижения вперед и один сертификат для продвижения назад, причем этот последний должен быть помечен как соответствующий непосредственно предшествующему CA.

7.6 Сертификаты хранятся в статьях Справочника в качестве атрибутов типов СертификатПользователя, Сертификат CA и ПерекрестнаяПараСертификатов. Эти типы атрибутов известны Справочнику. Этими атрибутами можно оперировать с помощью тех же самых операций протокола, что и остальными атрибутами. Определение этих типов может быть найдено в §3.3 настоящей Рекомендации. Спецификация этих типов атрибутов имеет следующий вид:

```
СертификатПользователя          ::=  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Сертификат
СертификатCA                    ::=  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Сертификат
ПерекрестнаяПараСертификатов    ::=  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ПараСертификатов
ПараСертификатов                ::=
    SEQUENCE{
        вперед [0] Сертификат OPTIONAL
        назад  [1] Сертификат OPTIONAL
        -- хоть один из этих двух должен присутствовать -- }
```

Пользователь может приобрести один или более сертификатов от одного или нескольких сертификатных руководящих органов. Каждый сертификат несет в себе имя того сертификатного органа, который его выработал.

7.7 В общем случае, прежде чем пользователи могут аутентифицировать друг друга, Справочник должен обеспечить полную сертификацию и ветви сертификации для продвижения назад. Однако на практике количество информации, которое должно быть получено из Справочника, может быть сокращено для каждого конкретного случая аутентификации. Это достигается за счет того, что:

a) если пользователи, желающие аутентифицировать друг друга, обслуживаются одним и тем же сертификатным органом, то ветвь сертификации становится тривиальной, и пользователи могут непосредственно раскрыть сертификаты один другого;

b) если CA пользователей расположены в иерархическом порядке, то пользователь может запомнить общедоступные ключи, сертификаты и сертификаты для продвижения назад всех сертификатных органов между собой и корнем ИДС. На практике это потребует от пользователя знания общедоступного ключа и сертификатов всего трех или четырех сертификатных органов; после этого пользователю потребуется только извлечение ветвей сертификации от общей пользующейся доверием точки;

c) если пользователь часто взаимодействует с пользователем, сертификаты которых вырабатываются каким-то конкретным другим CA, то этому пользователю достаточно запомнить ветвь сертификации к этому CA и обратную ветвь сертификации от этого CA; поэтому ему будет достаточно извлекать из Справочника только сертификат другого пользователя;

d) сертификатные органы могут взаимно подтверждать один другого на основе двусторонних соглашений; в результате этого сокращается ветвь сертификации;

e) если два пользователя уже взаимодействовали ранее и узнали один сертификат другого, то они могут аутентифицировать друг друга без какого-либо обращения к Справочнику.

Во всех случаях, узнав из ветви сертификации сертификаты друг друга, пользователи должны проверить действительность полученных сертификатов.

7.8 (Пример). Рис. 4/X.509 иллюстрирует гипотетический пример фрагмента ИДС, в котором CA образуют иерархию. Помимо информации, изображенной при CA, мы предполагаем, что каждому пользователю известны общедоступный ключ его сертификатного органа, а также его собственные общедоступный и секретный ключи.

7.8.1 Если CA пользователей расположены иерархически, то пользователь A может сформировать ветвь сертификации к пользователю B, получив от Справочника нижеследующие сертификаты:

X«W», W«V», V«Y», Y«Z», Z«B»

Получив эти сертификаты, пользователь A может развернуть ветвь сертификации в последовательность, чтобы получить содержимое сертификата пользователя B, включая Bp:

Bp =Xp · X « W » W«V» V«Y» Y«Z» Z«B»

Чтобы сформировать обратную ветвь сертификации от B к A, пользователь A, как правило, должен также извлечь из Справочника следующие сертификаты:

Z«Y», Y«V», V«W», W«X», X«A».

Получив от пользователя A эти сертификаты, пользователь B может развернуть обратную ветвь сертификации в последовательность, чтобы получить содержимое сертификата A, включая Ap:

$$Ap = Zp \cdot Z\ll Y\gg Y\ll V\gg V\ll W\gg W\ll X\gg X\ll A\gg.$$

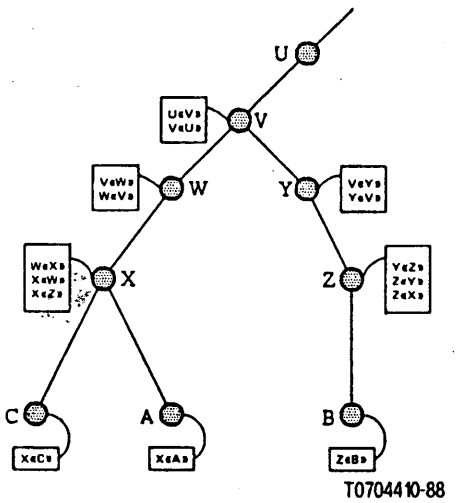


РИСУНОК 4/X.509

Иерархия CA — гипотетический пример

7.8.2 Используем оптимизацию, описанную в § 7.7:

a) возьмем, например, пользователей A и C; им обоим известен Xp и поэтому A может просто непосредственно извлечь сертификат пользователя C. Развертывание ветви сертификации сводится к

$$Cp = Xp \cdot X\ll C\gg,$$

а развертывание обратной ветви сертификации сводится к

$$Ap = Xp \cdot X\ll A\gg;$$

b) предположим, что пользователю A известны $W\ll X\gg$, Wp, $V\ll W\gg$, Vp, $U\ll V\gg$, Up, и т.д. Тогда информация, которую A должен извлечь из Справочника, чтобы сформировать ветвь сертификации, сводится к

$$V\ll Y\gg, Y\ll Z\gg, Z\ll B\gg,$$

а информация, которую A должен извлечь из Справочника, чтобы сформировать обратную ветвь сертификации, сводится к

$$Z\ll Y\gg, Y\ll V\gg;$$

c) предположим, что пользователь A часто взаимодействует с пользователями, сертификатным органом которых является Z. Тогда (помимо тех общедоступных ключей, которые он узнал в предшествующем пункте b), он может узнать $V\ll Y\gg$, $Y\ll V\gg$, $Y\ll Z\gg$ и $Z\ll Y\gg$. Поэтому, чтобы взаимодействовать с пользователем B, ему достаточно извлечь из Справочника только $Z\ll B\gg$;

d) предположим, что пользователи, сертификатными органами которых являются X и Z, часто взаимодействуют друг с другом. Тогда в статье Справочника элемента X будет храниться $X\ll Z\gg$ и наоборот (это изображено на рис. 4/X.509). Если A хочет аутентифицировать себя пользователю B, то A должен извлечь только

$$X\ll Z\gg, Z\ll B\gg,$$

чтобы сформировать ветвь сертификации, и

$$Z\ll X\gg,$$

чтобы сформировать обратную ветвь сертификации;

e) предположим, что пользователи A и C уже вступали ранее во взаимодействие и узнали один сертификат другого. Тогда они могут воспользоваться один непосредственно общедоступным ключом другого, то есть

$Cp = Xp \cdot X\langle\langle C\rangle\rangle$,

и

$Ap = Xp \cdot X\langle\langle A\rangle\rangle$.

7.8.3 В более общем случае между сертификатными руководящими органами нет иерархических отношений. Рассмотрим гипотетический пример на рис. 5/X.509 и предположим, что пользователь D, чьим сертификатным органом является U, хочет аутентифицировать себя пользователю E, чьим сертификатным органом является W. Статья Справочника пользователя D должна хранить сертификат $U\langle\langle D\rangle\rangle$, а статья пользователя E должна хранить сертификат $W\langle\langle E\rangle\rangle$.

Пусть V является CA, с которым сертификатные органы U и W обменялись когда-то ранее общедоступными ключами, причем способом, вызывающим доверие. В результате этого были выработаны и сохранены в Справочнике сертификаты $U\langle\langle V\rangle\rangle$, $V\langle\langle U\rangle\rangle$, $W\langle\langle V\rangle\rangle$, $V\langle\langle W\rangle\rangle$. Предположим, что $U\langle\langle V\rangle\rangle$ и $W\langle\langle V\rangle\rangle$ сохранены в статье V, $V\langle\langle U\rangle\rangle$ сохранен в статье U и $V\langle\langle W\rangle\rangle$ сохранен в статье W.

Пусть пользователь D должен найти ветвь сертификации к пользователю E. Могут быть применены различные стратегии. Одна такая стратегия заключается в рассмотрении пользователей и CA как вершин, а сертификатов как дуг ориентированного графа. В этих терминах пользователь D должен совершить поиск, чтобы отыскать ориентированный путь, соединяющий D с E. Одним таким путем является $U\langle\langle V\rangle\rangle$, $V\langle\langle W\rangle\rangle$, $W\langle\langle E\rangle\rangle$. После того как эта ветвь обнаружена, обратная ветвь $W\langle\langle V\rangle\rangle$, $V\langle\langle U\rangle\rangle$, $U\langle\langle D\rangle\rangle$ также может быть сформирована.

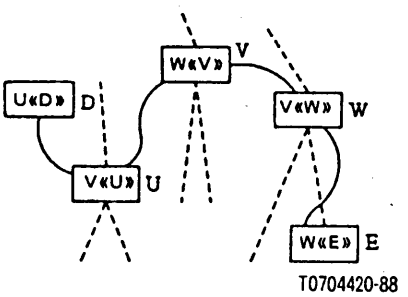


РИСУНОК 5/X.509

Неиерархический сертификатный путь — пример

8 Цифровые подписи

Настоящий раздел не направлен на спецификацию стандартов на цифровые подписи в общем случае. В нем лишь специфицируется способ, которым в Справочнике подписываются мандаты.

8.1 Информация (инфо) подписывается присоединением к ней зашифрованного резюме информации. Резюме вырабатывается с помощью однонаправленной хэш-функции, а шифровка осуществляется применением секретного ключа подписывающего (см. рис. 6/X.509). Таким образом

$$X\{\text{Инфо}\} = \text{Инфо}, Xs[h(\text{Инфо})]$$

Примечание. — Шифровка с помощью секретного ключа обеспечивает невозможность подделки подписи. Однонаправленность хэш-функции обеспечивает невозможность подстановки фальшивой информации, выработанной с тем, чтобы выдавать тот же хэш-результат (а, следовательно, и подпись).

8.2 Получатель подписанной информации проверяет подпись:

— применением к информации однонаправленной хэш-функции;

— сравнением полученного результата с результатом дешифровки подписи с помощью общедоступного ключа подписывающего.

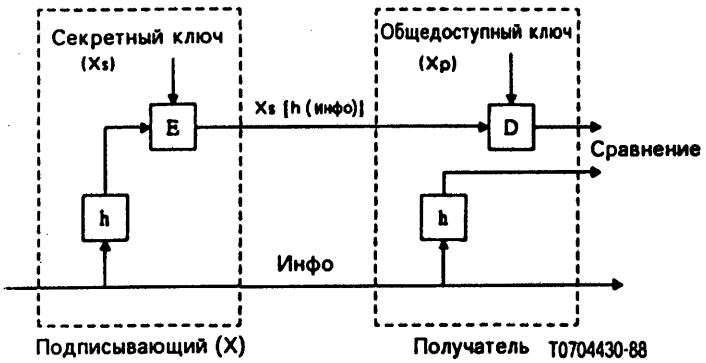


РИСУНОК 6/X.509

Цифровые подписи

8.3 Структура аутентификации не требует использования при подписывании одной единственной однонаправленной хэш-функции. Предполагается, что описываемая структура будет применима к любой подходящей хэш-функции и, следовательно, сможет поддержать те изменения используемых методов, которые окажутся результатом дальнейших исследований в криптографии, в математической технике и в вычислительных возможностях. Однако, если два пользователя хотят уметь аутентифицировать друг друга, то они должны использовать одну и ту же хэш-функцию, чтобы аутентификация выполнялась корректно. Таким образом, в контексте некоторого набора связанных приложений выбор одной какой-нибудь функции будет способствовать максимальному расширению того сообщества пользователей, которые смогут аутентифицировать друг друга и взаимодействовать безопасным образом. Один пример хэш-функции специфицируется в Приложении D.

Подписанная информация включает также индикаторы, идентифицирующие хэш-алгоритм и алгоритм шифровки, используемые при вычислении цифровой подписи.

8.4 Шифровка некоторого элемента данных может быть описана с помощью следующего НАС.1-макроса:

```
ENCRYPTED MACRO      ::=
BEGIN

TYPE NOTATION        ::=  type (БытьЗашифрованным)
VALUE NOTATION       ::=  value (VALUE BIT STRING)
END
```

Значение цепочки битов формируется следующим образом: к октетам, образующим полный код (согласно Основным правилам кодирования НАС.1) значения, принадлежащего типу БытьЗашифрованным, применяется процедура шифровки.

Примечание 1. — Процедура шифровки требует согласования используемого алгоритма, включая любые параметры алгоритма, такие как требующиеся ключи, значения при инициализации и инструкции по заполнению пустых мест. Процедуры шифровки ответственны за спецификацию средств, с помощью которых осуществляется синхронизация отправителя и получателя данных; для этого может потребоваться включение некоторой информации в пересылаемые биты.

Примечание 2. — Процедура шифровки должна получать в качестве входа цепочку октетов и вырабатывать единую цепочку битов в качестве выхода.

Примечание 3. — Механизмы безопасного согласования алгоритма шифровки и его параметров между отправителем и получателем данных выходят за рамки настоящей Рекомендации.

8.5 Для тех случаев, в которых к типу данных должна быть присоединена подпись, может быть использован следующий НАС.1-макрос, описывающий тот тип данных, который образуется от применения подписи к заданному типу данных.

```
SIGNED MACRO            ::=
BEGIN

TYPE NOTATION           ::=  type (БытьПодписанным)

VALUE NOTATION          ::=  value (VALUE

    SEQUENCE{
        БытьПодписанным,
        ИдентификаторАлгоритма,
        -- алгоритма, использованного для
        -- вычисления подписи
        ENCRYPTED OCTET STRING
        -- где цепочка октетов является
        -- результатом хэширования значения
        -- "БытьПодписанным" --}

END -- макроса SIGNED.    )
```

8.6 Для тех случаев, в которых требуется только подпись, может быть использован следующий НАС.1-макрос, определяющий тот тип данных, который образуется от применения подписи к заданному типу данных.

```
SIGNATURE MACRO         ::=
BEGIN

TYPE NOTATION           ::=  type (ТипаПодпись)

VALUE NOTATION          ::=  value (VALUE
    SEQUENCE{
        ИдентификаторАлгоритма,
        -- алгоритма, использованного для
        -- вычисления подписи
        ENCRYPTED OCTET STRING
        -- где цепочка октетов является некоторой
        -- функцией (например, сжатия
        -- или хэширования) значения
        -- типа "ТипаПодпись", которое может включать
        -- идентификатор алгоритма, использованного
        -- для вычисления подписи -- }

END -- макроса SIGNATURE.    )
```

8.7 Для того чтобы в распределенной среде можно было проверить действительность типов SIGNED или SIGNATURE, требуется специальное кодирование. Специальное кодирование значений данных типа SIGNED или SIGNATURE может быть получено применением Основных правил кодирования, определенных в Рекомендации X.209, однако со следующими ограничениями:

a) должна использоваться некоторая конкретная форма кодирования, приводящая к минимальному числу октетов;

b) для цепочечных типов не должна использоваться структурная форма кодирования;

c) если значением некоторого типа является его значение по умолчанию, то оно должно отсутствовать;

d) компоненты типа set должны кодироваться в порядке возрастания значений их меток;

e) компоненты типа set-of должны кодироваться в порядке роста значений их октетов;

f) если значением Булевского типа является true, то при кодировании содержимое октета, кодирующего это значение, должно равняться "FF_{16}";

g) значение каждого неиспользуемого бита последнего октета при кодировании значения типа ЦепочкаБитов должно равняться нулю;

h) при кодировании чисел Вещественного типа не должны использоваться основания 8, 10 и 16 и двоичный коэффициент масштабирования должен равняться нулю.

9 Процедура строгой аутентификации

9.1 *Общее описание*

9.1.1 Основной подход к аутентификации был определен выше. Он заключается в подтверждении подлинности демонстрацией обладания секретным ключом. Однако возможно много различных процедур, опирающихся на этот подход. Вообще говоря, каждое приложение должно само определить подходящие процедуры, чтобы удовлетворить свою политику безопасности. В настоящем параграфе описываются три конкретные процедуры, которые могут оказаться полезными для приложений большого диапазона.

Примечание. — В настоящей Рекомендации процедуры специфицируются не столь подробно, как это необходимо для реализации. Можно, однако, предвидеть появление дополнительных стандартов, которые сделают это, причем либо в проблемно-специфицированной, либо в общецелевой форме.

9.1.2 Три процедуры содержат различное число обменов аутентифицирующей информацией и, следовательно, обеспечивают своим участникам различные степени надежности. А именно:

a) аутентификация только в одном направлении, описываемая в § 9,2 содержит одну передачу информации от одного пользователя (A) другому пользователю (B); эта передача информации устанавливает:

   — подлинность пользователя A и тот факт, что аутентификационный мандат был фактически сгенерирован пользователем A;

   — подлинность пользователя B и тот факт, что аутентификационный мандат был действительно предназначен для посылки пользователю B;

   — целостность и "оригинальность" (это свойство означает, что пересылаемая информация не повторяется два или более раз) пересылаемого аутентификационного мандата.

   Указанные свойства могут быть установлены также для любых дополнительных данных, сопровождающих передачу;

b) аутентификация в двух направлениях, описываемая в § 9.3, содержит дополнительно ответ пользователя B пользователю A; она устанавливает дополнительно

   — тот факт, что аутентификационный мандат, выработанный в ответе, был фактически выработан пользователем B и предназначался для посылки пользователю A;

   — целостность и оригинальность аутентификационного мандата, посылаемого в ответе;

   — (возможно) взаимную секретность части мандатов;

c) аутентификация в трех направлениях, описываемая в § 9.4, дополнительно содержит последующую передачу от пользователя A к пользователю B; она устанавливает те же свойства, что и аутентификация в двух направлениях, но делает это, не требуя проверки штемпеля времени ассоциации.

Во всех случаях, в которых имеет место строгое установление подлинности, пользователь A должен извлечь общедоступный ключ пользователя B и определить ветвь сертификации для продвижения назад от B к A и сделать это до какого-либо обмена информацией. В это может входить обращение к Справочнику, описанное в § 7. Эти обращения не упоминаются в нижеследующих описаниях процедур.

Проверка штемпеля времени, упоминаемая в следующих пунктах, применяется только в тех случаях, если в локальной среде используются синхронизированные часы или если часы логически синхронизированы на основании двусторонних соглашений. При всех условиях рекомендуется использование стандартного времени от Гринвича.

Для каждой из трех описываемых ниже процедур аутентификации предполагается, что сторона A проверила действительность всех сертификатов ветви сертификации.

9.2 *Аутентификация в одном направлении*

При этой аутентификации предпринимаются шаги, изображенные на рис. 7/X.509.

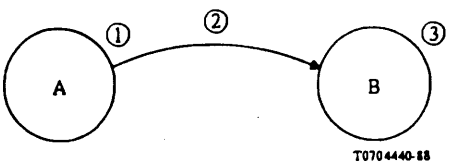


РИСУНОК 7/X.509

Аутентификация в одном направлении

1) A генерирует неповторяющееся число r^A, используемое для обнаружения попыток розыгрыша и для предотвращения подделок.

2) A посылает пользователю B следующее сообщение

$B \rightarrow A,\ A\{t^A, r^A, B\},$

где t^A — штемпель времени. Этот штемпель состоит из одной или двух дат: время генерации мандата (необязательное) и срока истечения его годности. Альтернативно, если в цифровую подпись должна быть включена аутентификация источника значения "датаПдпс", то сообщение принимает вид

$B \rightarrow A,\ A\{t^A, r^A, B, \text{датаПдпс}\}.$

В тех случаях, когда "датаШифр" будет в последующем использована в качестве секретного ключа, сообщение принимает вид:

$B \rightarrow A,\ A\{t^A, r^A, B, \text{датаПдпс}, Bp[\text{датаШифр}]\}.$

Использование "датаШифр" в качестве секретного ключа подразумевает, что он должен быть выбран очень осторожно, например, быть строгим ключом для любой криптсистемы, как это указывается в поле "датаПдпс" мандата.

3) B выполняет следующие действия:

a) извлекает Ap из $B \rightarrow A$, убеждаясь в том, что сертификат пользователя A еще не истек;

b) проверяет подпись и тем самым целостность подписанной информации;

c) убеждается в том, что именно ему, пользователю B, предназначалась посланная информация;

d) убеждается в том, что штемпель времени является "текущим";

e) убеждается в том, что число r^A не повторялось; это может быть осуществлено, например, включением в r^A части, предназначенной для хранения последовательного номера, что дает возможность локальной установке убедиться в уникальности значения r^A.

Значение r^A действительно вплоть до срока истечения, указанного с помощью t^A; r^A всегда сопровождается частью, содержащей последовательные значения, которые указывают, что A не повторяет дважды мандата в течение интервала времени t^A, а, следовательно, нет необходимости в проверке самого значения r^A.

При всех условиях стороне B имеет смысл хранить в чистом виде часть, предназначенную для последовательного значения, наряду со штемпелем времени t^A и с хэш-частью мандата в течение всего интервала времени t^A.

9.3 *Аутентификация в двух направлениях*

При этой аутентификации выполняются следующие шаги, изображенные на рис. 8/X.509.

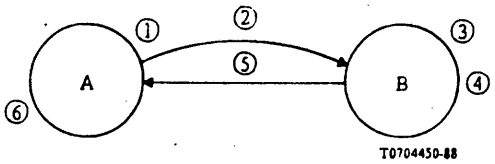


РИСУНОК 8/X.509

Аутентификация в двух направлениях

1) Такой же, как в § 9.2.

2) Такой же, как в § 9.2.

3) Такой же, как в § 9.2.

4) B генерирует неповторяющееся число r^B, используемое в тех же целях что и r^A.

5) B посылает пользователю A следующий аутентификационный мандат:

$B\{t^B, r^B, A, r^A\}$,

где t^B – штемпель времени, определенный тем же способом, что и t^A.

Альтернативно, если в цифровую подпись должна быть включена аутентификация источника значения "датаПдпс", то посылаемое сообщение принимает вид:

$B\{t^B, r^B, A, r^A,$ датаПдпс$\}$.

В тех случаях, когда "датаШифр" будет в последующем использована в качестве секретного ключа, сообщение принимает вид:

$B\{t^B, r^B, A, r^A,$ датаПдпс, Ар[датаШифр]$\}$.

Использование "датаШифр" в качестве секретного ключа подразумевает, что он должен быть выбран очень осторожно, например, быть строгим ключом для любой криптосистемы, как это указывается в поле "датаПдпс" мандата.

6) A выполняет следующие действия:

a) проверяет подпись и тем самым целостность подписанной информации;

b) проверяет, является ли A тем, кому было предназначено отправление;

c) убеждается в том, что штемпель времени t^B является "текущим";

d) убеждается (не обязательное действие) в том, что r^B не повторилось [см. § 9.2, шаг 3) e)].

9.4 *Аутентификация в трех направлениях*

При этой аутентификации предпринимаются шаги, изображенные на рис. 9/X.509.

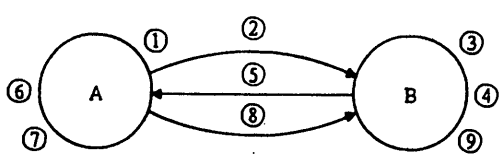


РИСУНОК 9/X.509

Аутентификация в трех направлениях

1) Такой же, как в § 9.3.

2) Такой же, как в § 9.3. Штемпель времени t^A может равняться нулю.

3) Такой же, как в § 9.3, за исключением того, что штемпель времени не должен проверяться.

4) Такой же, как в § 9.3.

5) Такой же, как в § 9.3. Штемпель времени t^B может равняться нулю.

6) Такой же, как в § 9.3, за исключением того, что штемпель времени не должен проверяться.

7) А убеждается в том, что полученное r^A совпадает с тем r^A, которое было послано.

8) А посылает пользователю В следующий аутентификационный мандат:

$A\{r^B\}$.

9) В выполняет следующие действия:

a) проверяет подпись и тем самым целостность подписанной информации;

b) убеждается в том, что полученный r^B совпадает с тем r^B, который был послан пользователем В.

10 Управление ключами и сертификатами

10.1 *Генерация парных ключей*

10.1.1 Общая политика безопасности данной реализации определяет жизненный цикл парных ключей и, следовательно, находится за пределами структуры установления подлинности. Тем не менее, для общей безопасности крайне необходимо, чтобы все секретные ключи оставались секретом тех, кому эти ключи принадлежат.

Человеку трудно помнить данные, связанные с ключами, и поэтому должен быть выработан подходящий метод их хранения в удобной и транспортабельной форме. Одним таким способом будет использование "Ловкой карты". В ней будут храниться секретные и (необязательно) общедоступные ключи пользователя, сертификат пользователя и копия общедоступного ключа сертификатного органа. Использование этой карты должно быть в свою очередь защищено. Для этого должен использоваться, например, хотя бы ПИН (персональный идентификационный номер), повышая безопасность системы требованием к пользователю иметь такую карту и знать, как осуществлять к ней доступ. Однако точный метод хранения этих данных выходит за рамки настоящей Рекомендации.

10.1.2 Существует три метода выработки парных ключей. Они описаны в § 10.1.2.1—10.1.2.3.

10.1.2.1 Пользователь сам генерирует свои собственные парные ключи. Этот метод имеет то преимущество, что секретный ключ пользователя никогда не становится известным кому бы то ни было, но требует от пользователя некоторой степени компетентности, как это описано в Приложении С.

10.1.2.2 Парные ключи генерируются третьей стороной. Третья сторона должна сообщить пользователю секретный ключ физически безопасным способом, после чего она должна полностью разрушить всю информацию, касающуюся выработки парных ключей, и плюс сами ключи. Должны использоваться подходящие физически безопасные методы, ограждающие третью сторону, а также все манипулирования данными от вмешательства.

10.1.2.3 Парные ключи генерируются сертификатным руководящим органом. Это — частный случай того случая, который описан в § 10.1.2.2, и все приведенные там рассуждения сохраняют силу в данном случае.

Примечание. — Сертификатное ведомство уже является органом, пользующимся доверием пользователя, и подчиняется необходимым физическим мерам безопасности. Этот метод имеет то преимущество, что не требует безопасной передачи данных сертификатному органу на предмет сертификации.

10.1.2.4 Используемая криптосистема накладывает конкретные (технические) ограничения на генерацию ключей.

10.2 *Управление сертификатами*

10.2.1 Сертификат связывает общедоступный ключ с уникальным выделенным именем того пользователя, которого описывает данный сертификат. Следовательно:

a) сертификатный руководящий орган должен считать приемлемым подтверждение подлинности пользователя, прежде чем он будет создавать сертификат этого пользователя;

b) сертификатное ведомство не должно создавать сертификатов для двух пользователей с одинаковыми именами.

10.2.2 Производство сертификатов должно осуществляться в офлайновом режиме и не должно использовать автоматического механизма опроса/ответа. Преимущество такой сертификации заключается в том, что (в силу того, что секретный ключ сертификатного органа никому не известен, за исключением самого изолированного и физически безопасного CA), секретный ключ CA может быть узнан только прямой атакой на само CA, что делает маловероятной компрометацию ключа.

10.2.3 Очень важно, чтобы передача информации сертификатному органу была защищена от вмешательства, и, следовательно, должны быть приняты подходящие физические меры безопасности. В силу этого:

a) серьезным нарушением безопасности будет выработка сертификатным органом сертификата для пользователя, общедоступный ключ которого подвергся вмешательству;

b) если использовался метод выработки парных ключей, описанный в § 10.1.2.3, то безопасной передачи не требуется;

c) если для выработки парных ключей применялись методы, описанные в § 10.1.2.1 или в § 10.1.2.2, то пользователь может использовать различные режимы (онлайновые или офлайновые) для безопасной передачи своего общедоступного ключа сертификатному органу. Онлайновый режим может обеспечить некоторую дополнительную гибкость в осуществлении удаленных операций между CA и пользователем.

10.2.4 Сертификат является общедоступной порцией информации и никакие специальные меры безопасности не должны применяться при передаче сертификата Справочнику. Так как сертификат был выработан офлайновым сертификатным органом в интересах пользователя, которому будет дана копия сертификата, то от пользователя требуется только сохранение этой информации в своей статье в Справочнике, что он может сделать при последующем обращении к Справочнику. Альтернативно, CA может сам вписать сертификат за пользователя, но тогда этому агенту должны быть предоставлены требующиеся права доступа.

10.2.5 Срок действия сертификата ограничен; по истечении этого срока сертификаты перестают существовать. Для обеспечения непрерывности службы сертификатный орган должен обеспечить возможность периодической замены сертификатов, чтобы предупредить истечение срока годности сертификатов. Этому вопросу присуще несколько аспектов, обсуждаемых в § 10.2.5.1 и 10.2.5.2.

10.2.5.1 Действительность сертификатов может быть организована таким образом, что каждый из них становится действительным тогда, когда истекает срок годности предшествующего сертификата. Кроме того, допустимо и перекрытие интервалов годности двух последовательных сертификатов. Последнее предохраняет сертификатный орган от внедрения и рассылки большого числа сертификатов, срок годности которых истекает одновременно.

10.2.5.2 Потерявшие силу сертификаты, вообще говоря, должны удаляться из Справочника. В зависимости от политики безопасности и компетенции CA решается вопрос о сохранении старых сертификатов в течение некоторого времени, если обеспечивается служба неаннулирования данных.

10.2.6 Сертификаты могут быть отменены до истечения срока их годности, например, в случае, если предполагается, что был скомпрометирован секретный ключ пользователя, или в случае, когда прекращается обслуживание данного пользователя данным сертификатным органом, или если предполагается, что был скомпрометирован секретный ключ самого CA. Этому вопросу присуще несколько аспектов, обсуждаемых в § 10.2.6.1—10.2.6.4.

10.2.6.1 Об отмене сертификата пользователя или сертификата сертификатного органа оповещает сам сертификатный орган. Он же делает доступным новый сертификат, если того требуют обстоятельства. После этого CA оповещает владельца сертификата относительно отмены сертификата. Это делается с помощью некоторой офлайновой процедуры.

10.6.2.2 CA должно поддерживать:

a) список (со штемпелем времени) тех сертификатов, которые оно выработало и затем отменило;

b) список (со штемпелем времени) всех отмененных сертификатов всех тех CA, известных данному CA, сертификаты которых выработаны данным CA.

Оба этих списка должны существовать, даже если они пусты.

10.2.6.3 Поддержание статей Справочника, затронутых списками сертификатного органа, лежит на ответственности Справочника и его пользователей, действующих в соответствии со стратегией безопасности. Например, пользователь может изменить свою статью объекта заменой в ней старого сертификата на новый. Последний будет затем использован для аутентификации пользователя Справочником.

10.2.6.4 Списки отмен ("черные списки") сохраняются в статьях в качестве атрибутов типов "СписокОтмененныхСертификатов" и "СписокОтмененныхРуководящихОрганов". Этими атрибутами можно манипулировать с помощью тех же операций, с помощью которых осуществляется манипулирование прочими атрибутами. Эти типы атрибутов определяются следующим образом:

```
СписокОтмененныхСертификатов:    ::=  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX СписокСертификатов
СписокОтмененныхРуководящихОрганов  ::=  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX СписокСертификатов
СписокСертификатов                  ::=  SIGNED SEQUENCE{
    подпись ИдентификаторАлгоритма,
    источник Имя,
    последнееОбновление ВремяотГринвича,
    отмененныеСертификаты
        SIGNED SEQUENCE OF SEQUENCE{
            подпись ИдентификаторАлгоритма,
            источник Имя, ПорядковыйНомерСертификата субъект,
            датаОтмены ВремяОтГринвича}
                                    OPTIONAL}
```

Примечание 1. — Проверка всего списка сертификатов является локальным вопросом.

Примечание 2. — Если служба неаннулирования данных зависит от ключей, обеспечиваемых сертификатным органом, то служба должна обеспечивать, чтобы все касающиеся органа ключи (отмененные или срок действия которых истек) и все проштемпелеванные списки отмен были помещены в архив и снабжены сертификатами текущим органом.

ПРИЛОЖЕНИЕ А

(к Рекомендации X.509)

Требования к безопасности

Настоящее Приложение не является составной частью настоящей Рекомендации.

[Дополнительный материал, относящийся к этому вопросу, может быть найден в ISO 7498 "Системы обработки информации — Эталонная модель ВОС — часть 2, Архитектура безопасности".]

Большое число ВОС-приложений, служб, определенных МККТТ, и служб, не определенных МККТТ, будут нуждаться в обеспечении безопасности. Требования к безопасности возникают из необходимости защитить передачу информации от целого спектра потенциальных опасностей.

А.1 *Опасности*

Некоторыми хорошо известными опасностями являются:

a) *вторжение в идентификацию* — идентифицирующая информация одного или нескольких пользователей, участвующих во взаимосвязи, изучается с неблаговидными целями;

b) *маскарад* — претензия пользователя изображать из себя некоторого другого пользователя с целью получения доступа к дополнительной информации или для получения дополнительных привилегий;

c) *розыгрыш* — записывание и последующее разыгрывание связи в некоторый более поздний срок;

d) *перехват данных* — обзор данных пользователя в процессе связи несанкционированным пользователем;

e) *манипулирование* — замена, вставка, удаление или нарушение порядка данных пользователя в процессе связи несанкционированным пользователем;

f) *отказ* — отрицание пользователем его участия в части или всем сеансе связи;

g) *нарушение службы* — недопущение связи, или вмешательство в связь, или задержка срочных операций;

Примечание. — Эта угроза безопасности является более общей и зависит от конкретного приложения или от намерений к несанкционированному разрушению и поэтому выходит за пределы непосредственно структуры аутентификации;

h) *искажение пути* — искажение линии связи, ведущей от одного пользователя к другому.

Примечание. — Ясно, что искажение пути будет возникать в уровнях 1—3 ВОС. Поэтому искажение пути выходит за пределы непосредственно структуры аутентификации. Однако, может быть,появится возможность исключить последствия искажения пути за счет использования соответствующих служб безопасности, как они обеспечиваются в пределах структуры аутентификации.

i) *анализ трафика* — обзор информации, касающейся связи между пользователями (например, наличие/отсутствие, частота, направление, последовательность, тип, объем и т.д.).

Примечание. — Ясно, что опасность анализа трафика не связана с каким-нибудь одним уровнем ВОС. Поэтому, вообще говоря, анализ трафика выходит за пределы структуры аутентификации. Однако анализ трафика может быть частично защищен за счет генерации дополнительного бессодержательного трафика (набивка трафика) с помощью шифрованных или случайных данных.

A.2 *Служба безопасности*

Для защиты различных осознанных угроз должны быть обеспечены различные службы безопасности. Службы безопасности, обеспечиваемые структурой аутентификации, осуществляются с помощью механизмов безопасности, описываемых в разделе A.3 настоящего Приложения.

a) *аутентификации равноуровневых элементов* — эта служба обеспечивает подтверждение того, что в данный момент связи пользователь является действительно тем пользователем, за которого он себя выдает. Могут быть запрошены две различные службы аутентификации равноуровневых элементов:

- *аутентификация одного элемента* (то есть аутентификация или *отправителя данных* или *получателя данных*);
- *обоюдная аутентификация*, где оба пользователя, поддерживающие связь, аутентифицируют друг друга.

Запрашивая службу аутентификации равноуровневых элементов, пользователи договариваются о том, будет или не будет защищена информация, идентифицирующая их.

Служба аутентификации равноуровневых элементов обеспечивается структурой аутентификации. Она может быть использована в качестве защиты от маскарада и розыгрыша, касающихся идентификации пользователей;

b) *управление доступом* — эта служба может быть использована в качестве защиты от несанкционированного использования ресурсов. Службу управления доступом обеспечивает Справочник или некоторое другое приложение, и поэтому она не входит в сферу интересов структуры аутентификации;

c) *секретность данных* — эта служба может быть использована в качестве защиты от несанкционированного раскрытия данных. Служба секретности данных обеспечивается структурой аутентификации. Она может быть использована против перехвата данных;

d) *целостность данных* — эта служба обеспечивает доказательства целостности данных в процессе связи. Служба целостности данных обеспечивается структурой аутентификации. Она может быть использована для обнаружения манипулирования и защиты от него;

e) *сохранность* — эта служба обеспечивает доказательство целостности и источника данных — во взаимоотношениях без подделки, — которые могут быть проверены в любой момент времени любой третьей стороной.

A.3 *Механизм безопасности*

Механизмы безопасности, описываемые в настоящем разделе, выполняют службы безопасности, описанные в разделе A.2.

a) *обмен аутентификацией* — существуют два уровня структуры аутентификации:

- *простая аутентификация* — опирается на предоставление источником своих имени и пароля; эти данные проверяются получателем;

- *строгая аутентификация* — опирается на использование криптографической техники для защиты обмена удостоверяющей информации; в структуре аутентификации строгая аутентификация опирается на несимметричную схему.

Механизм обмена аутентификацией используется для обеспечения службы аутентификации равноуровневых элементов;

b) *шифровка* — структура установления подлинности предусматривает шифровку данных в процессе передачи. Могут быть использованы как симметричная, так и несимметричная схемы. Необходимый для каждой из схем обмен ключами осуществляется либо в процессе предшествующего обмена аутентификацией либо в офлайновом режиме в любой момент времени, предшествующий намечаемому сеансу связи. Последний случай выходит за пределы структуры аутентификации. Механизм шифровки обеспечивает службу секретности данных;

c) *целостность данных* — этот механизм включает шифровку сжатой цепочки, соответствующей данным, подлежащим передаче. Совместно с открытыми данными, это сообщение посылается получателю. Получатель повторяет сжатие и последующую шифровку открытого текста и сравнивает полученный результат с тем, что было сформировано отправителем, проверяя тем самым целостность данных.

Механизм целостности данных может быть обеспечен шифровкой сжатого открытого текста как несимметричной, так и симметричной схемами. (При симметричной схеме сжатие и шифровка могут осуществляться одновременно.) Этот механизм не обеспечивается явным образом структурой аутентификации. Однако он полностью обеспечивается как составная часть механизма цифровой подписи (см. ниже), используя несимметричную схему.

Механизм целостности данных обеспечивает службу целостности данных. Кроме того, он частично обеспечивает службу сохранности (для полного обеспечения этой службы требуется использование еще и механизма цифровой подписи);

d) *цифровая подпись* — этот механизм содержит шифровку секретным ключом источника сжатой цепочки, соответствующей данным, подлежащим передаче. Цифровая подпись наряду с открытыми данными посылаются получателю. Аналогично механизму целостности данных это сообщение обрабатывается получателем для проверки целостности. Механизм цифровой подписи, кроме того, подтверждает аутентичность источника и безусловную зависимость между источником и посланными данными.

Структура установления подлинности обеспечивает механизм цифровой подписи, используя несимметричную схему.

Механизм цифровой подписи обеспечивает службу целостности данных и, кроме того, службу сохранности.

A.4 *Защита от опасностей службами безопасности*

В конце настоящего приложения имеется таблица, в которой указываются те из опасностей, от которых может защитить каждая из служб безопасности. Наличие звездочки (*) указывает на то, что данная служба безопасности может защитить от данной опасности.

A.5 *Согласование служб и механизмов безопасности*

Обеспечение свойств безопасности в процессе одного конкретного сеанса связи требует согласования того контекста, в котором требуются эти службы безопасности. Это влечет за собой соглашение о типах механизмов безопасности и параметрах безопасности, которые необходимы для обеспечения таких служб безопасности. Процедуры, требующиеся для согласования механизмов и параметров, могут быть осуществлены либо как составная часть обычной процедуры установления соединения, либо в качестве самостоятельного процесса. Конкретные детали этих процедур согласования в настоящем Приложении не специфицируются.

	СЛУЖБЫ			
ОПАСНОСТИ	Аутентификация элемента	Секретность данных	Целостность данных	Сохранность
Вторжение в идентификацию	* (если требуется)			
Перехват данных		*		
Маскарад	*			
Розыгрыш	* (идентифицирующей информации)		* (данные)	*
Манипуляция			*	*
Несохранность				*

ПРИЛОЖЕНИЕ B

(к Рекомендации X.509)

Введение в криптографию с общедоступными ключами

Данное Приложение не является составной частью настоящей Рекомендации.

В обычных криптографических системах ключ, используемый для шифровки информации источником секретного сообщения, совпадает с ключом, используемым законным получателем для дешифровки сообщения.

Однако в криптосистемах с общедоступными ключами (КОДК) используются парные ключи, причем один ключ из пары используется для шифровки, а другой — для дешифровки. Каждая пара ключей ассоциируется с некоторым конкретным пользователем X. Один из ключей, называемый общедоступным ключом (Xp), является общеизвестным и может быть использован любым пользователем для шифровки данных. Только X, который владеет дополнительным секретным ключом (Xs), может расшифровать данные. (Это изображают с помощью следующей нотации: D = Xs[Xp[D]].) Не существует вычислительных средств, которые могли бы вывести секретный ключ из общедоступного ключа. Таким образом, каждый пользователь может передать порцию информации, которую может раскрыть только X; для этого этот пользователь должен зашифровать информацию с помощью Xp. Расширение этого метода позволяет любым двум пользователям секретно взаимодействовать, используя каждый общедоступный ключ другого для шифровки данных, как это изображено на рис. B-1/X.509.

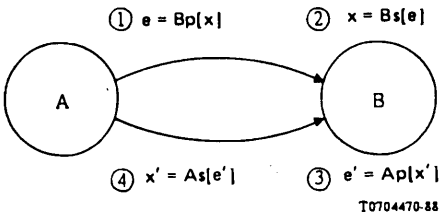


РИСУНОК B-1/X.509

Использование КОДК для обмена секретной информацией

Пользователь A владеет общедоступным ключом Ap и секретным ключом As; подобно этому пользователь B владеет другой парой ключей: Bp и Bs. И A, и B знают каждый общедоступный ключ другого, но не знают один секретный ключ другого. Поэтому A и B могут обмениваться друг с другом секретной информацией, выполняя следующие шаги (проиллюстрированные на рис. B-1/X.509):

1) Пользователь A хочет послать пользователю B некоторую секретную информацию x. Для этого A шифрует x с помощью шифрующего ключа пользователя Bp и посылает зашифрованную информацию e пользователю B. Это изображается нотацией:

   $e = Bp[x]$.

2) Пользователь B может расшифровать эту шифрограмму e и получить информацию x, используя для этого секретный ключ Bs. Обратите внимание на то, что B является единственным обладателем ключа Bs. Так как этот ключ никогда не будет ни раскрыт, ни послан куда-нибудь, никто другой не сможет извлечь информацию x. Обладание ключом Bs определяет подлинность пользователя B. Операция расшировки изображается нотацией:

   $X = Bs[e]$ или $x = Bs[Bp[x]]$.

3) В свою очередь, B может послать некоторую секретную информацию x' пользователю A, используя для этого шифровальный ключ Ap пользователя A:

   $e' = Ap[x']$.

4) Пользователь A извлекает информацию x', расшифровывая e':

   $x' = As[e']$ или $x' = As[Ap[x']]$.

С помощью этих средств A и B обменялись секретной информацией x и x'. Эта информация не может быть извлечена никем другим, помимо A и B, если их секретные ключи не раскрыты.

Такой обмен, помимо самой передачи секретной информации, может также служить средством установления подлинности пользователей. Конкретнее, A и B идентифицируются обладанием их секретных ключей расшифровки As и Bs соответственно. A может проверить, обладает ли B секретным ключом Bs, если B включит в сообщение x часть из полученной им информации x. Это укажет пользователю A, что он вступил в связь с обладателем ключа Bs. Пользователь B может аналогичным образом проверить подлинность A.

Некоторые КОДК обладают тем свойством, что в них шаги шифровки и дешифровки могут быть переставлены, как это изображается нотацией $D = Xp[Xs[D]]$. Это позволит любому пользователю (обладающему ключом Xp) прочесть порцию информации, которую мог выработать только пользователь X. Этот метод может быть использован для определения источника информации; метод лежит в основе цифровой подписи. В описываемой структуре аутентификации годятся для использования только такие КОДК, которые обладают указанным свойством (переставляемости). Один такой алгоритм описан в Приложении C.

Дальнейшая информация может быть получена из

DIFFIE, W. and HELLMAN, M. E. (November 1976) — New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22, No. 6.

ПРИЛОЖЕНИЕ C

(к Рекомендации X.509)

Криптосистема RSA с общедоступными ключами

Данное Приложение не является составной частью настоящей Рекомендации.

Примечание. — Криптосистема, описываемая в настоящем Приложении, была разработана Райвестом (Rivest), Шамиром (Shamir) и Адлеманом (Adleman) и широко известна под именем RSA.

C.1 *Предмет рассмотрения и область применения*

Полное обсуждение криптосистемы RSA выходит за пределы настоящего Приложения. Здесь приводится всего лишь краткое описание метода, опирающегося на возведение в степень по некоторому модулю.

C.2 *Ссылки*

Дальнейшая информация имеется в:

1) Общие вопросы

RIVEST, R. L., SHAMIR, A. and ADLEMAN, L. (February 1978) — A Method for Obtaining Digital Signatures and Public-key Cryptosystems, *Communications of the ACM*, 21, 2, 120—126.

2) Генерация ключей

GORDON, J. — Strong RSA Keys, *Electronics Letters*, 20, 5, 514—516.

3) Метод расшифровки

QUISQUATER, J. J. and COUVREUR, C. (October 14, 1982) — Fast Decipherment Algorithm for RSA Public-key Cryptosystems, *Electronics Letters*, 18, 21, 905—907.

C.3 *Определения*

a) *общедоступный ключ* — пара параметров, состоящая из общедоступной экспоненты и арифметического модуля.

Примечание. — НАС.1-элемент данных **ОбщедстпКлючСубъекта**, определяемый как **BIT STRING** (см. Приложение G), должен интерпретироваться для случая RSA как принадлежащий типу:

SEQUENCE {INTEGER, INTEGER},

где первое целое является арифметическим модулем, а второе целое — общедоступной экспонентой. Последовательность изображается согласно Основным правилам кодирования НАС.1.

b) *секретный ключ* — пара параметров, состоящая из секретной экспоненты и арифметического модуля.

C.4 *Обозначения и сокращения*

X, Y — блоки данных, которые арифметически меньше модуля

n — арифметический модуль

e — общедоступная экспонента

d — секретная экспонента

p, q — простые числа, произведение которых образует арифметический модуль (n).

Примечание. — Хотя предпочтительнее формировать арифметический модуль как произведение двух простых чисел, использование трех или более простых сомножителей не исключается.

mod n — арифметика по модулю n.

C.5 *Описание*

Этот асимметричный алгоритм использует степенную функцию для преобразования блоков данных, так чтобы:

$Y = X^e \bmod n$, при $0 \leqslant X < n$:

$X = Y^d \bmod n$, при $0 \leqslant Y < n$.

Для этого можно, например, положить

$ed \bmod \text{О.Н.К.}\ (p-1, q-1) = 1$,

$ed \bmod (p-1)(q-1) = 1$.

Для осуществления этого процесса блоки данных должны интерпретироваться как целые числа. Это достигается рассмотрением всего блока как упорядоченной последовательности битов (например, длины l). Далее первому из битов приписывается вес 2^{l-1}, а вес каждого последующего бита равен весу предшествующего бита, поделенному на 2 (вес последнего бита равен 1). Целое число определяется как сумма из произведений битов на приписанные им веса.

Длина блока данных должна равняться наибольшему числу октетов, содержащему меньшее число битов, чем модуль. Неполные блоки должны быть дополнены любым желаемым образом. Может быть добавлено любое число блоков дополнительной набивки.

C.6 *Требования безопасности*

C.6.1 *Длины ключей*

Надо учитывать, что допустимая длина ключа будет вероятно меняться во времени в зависимости от стоимости и доступности аппаратуры, взятого момента времени прогресса техники и требуемого уровня безопасности. Рекомендуется для начала взять значение n, равное 512 битам, но этот вопрос подлежит *дальнейшему изучению.*

C.6.2 *Генерация ключа*

Безопасность RSA опирается на трудности факторизации n. Существует много алгоритмов, реализующих эту операцию, но чтобы помешать использованию любого известного в настоящее время метода, значения p и q должны быть выбраны очень осторожно в соответствии с нижеследующими правилами [см., например, раздел C.2, ссылку 2)]:

a) они должны быть выбраны случайным образом;

b) они должны быть большими;

c) они должны быть простыми;

d) $|p-q|$ должны быть большими;

e) $(p+1)$ должно иметь большой простой делитель;

f) $(q+1)$ должно иметь большой простой делитель;

g) $(p-1)$ должно иметь большой простой делитель (обозначим его r);

h) $(q-1)$ должно иметь большой простой делитель (обозначим его s);

i) $(r-1)$ должно иметь большой простой делитель;

j) $(s-1)$ должно иметь большой простой делитель.

После генерации общедоступного и секретного ключей, например "Xp" и "Xs" (как они определены в § 3.3 и § 4.1 настоящей Рекомендации), состоящих из e, d и n, значения p и q, а также все прочие выработанные данные, такие как произведение $(p-1)\cdot(q-1)$ и большие простые делители, желательно уничтожить. Однако локальное сохранение p и q может повысить пропускную способность расшифровки от двух до четырех раз. Решение о сохранении p и q рассматривается как локальный вопрос [см. ссылку 3)].

Должно быть обеспечено неравенство $e > \log_2(n)$, чтобы исключить возможность раскрытия открытого текста взятием корня степени e по модулю n.

C.7 *Общедоступная экспонента*

Общедоступная экспонента (e) может быть одной и той же для всех участников. Это минимизирует длину той части общедоступного ключа, которая фактически должна быть распространена среди пользователей. Это, в свою очередь, уменьшит объемы передач и сложность преобразования (см. примечание 1).

Экспонента *e* должна быть достаточно велика, но такой, чтобы возведение в степень *e* могло быть осуществлено эффективно в смысле затрачиваемого времени и используемой емкости памяти. Если желательно иметь фиксированный общедоступный ключ, то использование числа Ферма F_4 (см. примечание 2) имеет ряд существенных достоинств.

$F_4 = 2^{2^4} + 1$

= 65537 (в десятичном виде)

= 1 0000 0000 0000 0001 (в двоичном виде)

Примечание 1 — Хотя и модуль *n* и экспонента *e* являются общедоступными, модуль не должен быть частью, принадлежащей группе пользователей. Знания модуля *n*, общедоступной экспоненты *e* и секретной экспоненты *d* достаточно для разложения *n* на множители. Следовательно, если бы модуль был общим, то каждый мог бы вывести его сомножители, получая, таким образом, возможность обнаружить секретный ключ каждого пользователя.

Примечание 2. — Фиксированная экспонента должна быть большой и простой, но должна также обеспечивать эффективную обработку. Число Ферма F_4 удовлетворяет этим требованиям. Например, аутентификация требует только 17-ти умножений и выполняется в среднем в 30 раз быстрее, чем дешифровка.

C.8 *Соответствие*

Хотя в настоящем Приложении специфицируется алгоритм получения общедоступного и секретного ключей, в нем не определяется метод выполнения собственно вычислений; поэтому возможны несколько продуктов, удовлетворяющих настоящему Приложению и взаимно совместимых.

ПРИЛОЖЕНИЕ D

(к Рекомендации X.509)

Хэш-функции

Данное Приложение не является составной частью настоящей Рекомендации.

D.1 *Требования к хэш-функции*

Для использования хэш-функции в качестве надежной однонаправленной функции должно выполняться требование о том, что достаточно легкое получение одного и того же хэш-результата из различных комбинаций входных сообщений недопустимо.

Строгая хэш-функция должна удовлетворять следующим требованиям:

a) хэш-функция должна быть однонаправленной; это означает, что вычислительно невозможно создать такое входное сообщение, результат хэширования которого совпал бы с некоторым заданным результатом хэширования;

b) хэш-функция должна быть свободной от коллизий, то есть вычислительно невозможно создать два таких входных сообщения, результаты хэширования которых совпадали бы между собой.

D.2 *Описание хэш-функции*

Нижеследующая хэш-функция ("квадрат по mod n") выполняет сжатие данных на блок на поблочной основе.

Хэширование осуществляется в три основных шага.

1) Цепочка данных, подлежащих хэшированию, разбивается на блоки B равной длины. Эта длина определяется, исходя из характеристик той несимметричной криптосистемы, которая используется для подписывания. Для криптосистемы RSA эта длина (в октетах) равна наибольшему целому шагу l, для которого $16l < \log_2 n$, где n — модуль RSA.

2) Из соображений неинвертируемости каждый октет разбит пополам. Каждой из половин приписывается спереди "головка", состоящая из двоичных единиц. С помощью такого зонирования вносится некоторая жесткость или избыточность, существенно повышающая свойство неинвертируемости хэш-функции. Каждый блок, построенный на шаге 1, расширяется так, чтобы его длина стала равной модулю n.

3) Каждый из блоков, полученных на шаге 2, складывается с предыдущим по модулю 2, возводится в квадрат и сокращается до значения по модулю n; это повторяется до тех пор, пока не будут обработаны все блоки.

Результатом является величина H_m, где

$H_0 = 0$

$H_1 = (H_{i-1} \oplus B_i)^2 \bmod n$, для $1 \leqslant i \leqslant m$.

Если последний подлежащий хэшированию блок укороченный, то он дополняется единицами.

ПРИЛОЖЕНИЕ E

(к Рекомендации X.509)

Опасности, от которых защищает строгая аутентификация

Данное Приложение не является составной частью настоящей Рекомендации.

Строгая аутентификация, описанная в настоящей Рекомендации, предоставляет защиту от опасностей, описанных в Приложении A.

Кроме того, существует ряд потенциальных опасностей, специфичных для самого метода строгой аутентификации. Это следующие опасности:

Компрометация секретного ключа пользователя — одним из основных принципов строгой аутентификации является безопасность секретного ключа пользователя. Существует целый ряд практических методов, позволяющих пользователю хранить свой секретный ключ так, чтобы обеспечить ему необходимую безопасность. Воздействие компрометации ограничивается разрушением связей, в которые вовлечен данный пользователь.

Компрометация секретного ключа самого CA — безопасность секретного ключа самого CA также является основным принципом строгой аутентификации. Применяются методы физической безопасности и методы, "требующие знания". Воздействие компрометации ограничивается разрушением связей всех пользователей, получивших сертификаты от данного CA.

Вовлечение CA в производство недействительных сертификатов — тот факт, что CA функционируют в офлайновом режиме, предоставляет некоторую защиту. CA ответственен за проверку действительности представленных удостоверений до выработки сертификатов. Воздействие компрометации ограничивается разрушением связей, включающих того пользователя, для которого был создан сертификат, и всех тех, которые подверглись воздействию недействительного сертификата.

Сговор между недобросовестным CA и пользователем — такой сговор поразит весь метод. Это будет означать предательство со стороны CA пользователей, доверявших ему. Воздействие недобросовестного CA ограничивается разрушением связей всех пользователей, получивших сертификаты от данного CA.

Подделка сертификата — метод строгой аутентификации защищает от подделки сертификата тем, что CA подписывает сертификат. Метод опирается на поддержание секретности секретного ключа CA.

Подделка мандата — метод строгой аутентификации защищает от подделки мандата тем, что отправитель подписывает мандат. Метод опирается на поддержание секретности секретного ключа отправителя.

Розыгрыш мандата — методы строгой аутентификации в одном и двух направлениях защищают от розыгрыша мандата тем, что включают в мандат штемпель времени. Метод трех направлений осуществляет это проверкой случайных чисел.

Атака на криптографическую систему — разумно предположить, что будет использован эффективный криптоанализ системы, опирающийся на развитие вычислительной теории чисел и ведущий к увеличению длины ключа.

ПРИЛОЖЕНИЕ F

(к Рекомендации X.509)

Секретность данных

Данное Приложение не является составной частью настоящей Рекомендации.

F.1 *Введение*

Процесс засекречивания данных может быть начат после того, как был совершен необходимый обмен шифровальными ключами. Это могло быть достигнуто или предшествующим обменом аутентификацией, как это было описано в § 9, или каким-либо иным процессом обмена ключами, описание которого выходит за пределы данной Рекомендации.

Секретность данных может быть обеспечена как симметричной, так и несимметричной шифровальными схемами.

F.2 *Засекречивание данных несимметричной шифровкой*

В этом случае засекречивание данных осуществляется средствами отправителя шифрованием подлежащих пересылке данных, причем для шифрования используется общедоступный ключ того получателя, которому данные предназначаются. Получатель расшифровывает данные, используя свой секретный ключ.

F.3 *Засекречивание данных симметричной шифровкой*

В этом случае засекречивание данных осуществляется использованием какого-нибудь симметричного алгоритма шифрования. Его выбор лежит вне рамок структуры аутентификации.

Если обмен аутентификацией (в соответствии с § 9) был осуществлен двумя вовлеченными в обмен сторонами, то ключ для использования симметричного алгоритма может быть выведен. Выбор секретного ключа зависит от того преобразования, которое будет использовано. Стороны должны быть уверены в том, что выбраны строгие ключи. В настоящей Рекомендации не специфицируется то, как делается этот выбор, хотя очевидно, что этот вопрос должен быть согласован заинтересованными сторонами или должен быть специфицирован в других стандартах.

ПРИЛОЖЕНИЕ G

(к Рекомендации X.509)

Структура аутентификации на HAC.1

Данное Приложение является составной частью настоящей Рекомендации.

Данное Приложение содержит HAC.1-модуль, СтруктураАутентификации, в который включены все HAC.1-определения типов, макросов и значений, приведенные в настоящей Рекомендации.

```
СтруктураАутентификации  { joint-iso-ccitt ds(5) modules(1)
                           authenticationFramework(7)}

DEFINITIONS ::=
BEGIN

EXPORTS     ИдентификаторАлгоритма, СписокОтмененныхРукОрганов, СертификатCA, Сертификат,
            Сертификаты, ВетвьСертификации, СписокОтмененныхСертификатов, СертификатПользователя,
            ПерекрестнаяПараСертификатов, ПарольПользователя, ALGORITHM,
            ENCRYPTED, PROTECTED, SIGNATURE, SIGNED;
```

```
IMPORTS
    СтруктураИнформации, избранныеТипыАтрибутов, ВерхниеГраницы
        FROM ПолезныеОпределения {joint-iso-ccitt ds(5)modules(1)
                usefulDefinitions(0)}

    Имя, ATTRIBUTE, ATTRIBUTE-SYNTAX
        FROM СтруктураИнформации структураИнформации

вг-пароля-пользователя FROM ВерхниеГраницы верхниеГраницы

-- типы

Сертификат ::= SIGNED SEQUENCE {
                   версия                      [0] Версия DEFAULT 1988,
                   порядковыйНомер             ПорядковыйНомер,
                   подпись                     ИдентификаторАлгоритма,
                   источник                    Имя,
                   годность                    Годность,
                   субъект,                    Имя,
                   инфоОбщедстКлючаСубъекта    ИнфоОбщедстКлючаСубъекта}

Версия          ::= INTEGER {1988(0)}
ПорядковыйНомер ::= INTEGER

Годность ::= SEQUENCE {
                 неРаньше    ВремяОтГринвича,
                 неПозже     ВремяОтГринвича}

ИнфоОбщедстКлючаСубъекта ::= SEQUENCE {
                                 алгоритм            ИдентификаторАлгоритма,
                                 общедстКлючСубъекта BIT STRING}

ИдентификаторАлгоритма ::= SEQUENCE {
                               алгоритм    OBJECT IDENTIFIER,
                               параметры   ANY DEFINED BY алгоритм OPTIONAL}

Сертификаты ::= SEQUENCE {
                    сертификат          Сертификат,
                    ветвь сертификации  ВетвьСертификацииВперед OPTIONAL}

ВетвьСертификацииВперед ::= SEQUENCE OF ПерекрестныеСертификаты

ВетвьСертификации ::= SEQUENCE {
                          сертификатПользователя  Сертификат,
                          сертификатыCA           SEQUENCE OF ПараСертификатов
                                                      OPTIONAL}

ПерекрестныеСертификаты ::= SET OF Сертификат
СписокСертификатов      ::= SIGNED SEQUENCE {
                                подпись                 ИдентификаторАлгоритма,
                                источник                Имя,
                                последнееОбновление     ВремяОтГринвича,
                                ОтмененныеСертификаты   SIGNEDSEQUENCE OF SEQUENCE {
                                                            подпись                 ИдентификаторАлгоритма,
                                                            источник                Имя,
                                                            сертификатПользователя  ПорядковыйНомер,
                                                            датаОтмены              ВремяОтГринвича}
                                                                OPTIONAL}

ПараСертификатов ::= SEQUENCE {
                         вперед [0]  Сертификат OPTIONAL,
                         назад [1]   Сертификат OPTIONAL
                         -- хоть один из пары должен присутствовать --}

-- типы атрибутов

СертификатПользователя ::= ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAXСертификат

СертификатCA           ::= ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAXСертификат
```

```
ПерекрестнаяПараСертификатов        ::=  ATTRIBUTE
                                           WITH ATTRIBUTE-SYNTAXПараСертификатов

СписокОтмененныхСертификатов        ::=  ATTRIBUTE
                                           WITH ATTRIBUTE-SYNTAXСписокСертификатов

СписокОтмененныхРукОрганов          ::=  ATTRIBUTE
                                           WITH ATTRIBUTE-SYNTAXСписокСертификатов

ПарольПользователя                  ::=  ATTRIBUTE
                                           WITH ATTRIBUTE-SYNTAX
                                           OCTETSTRING (SIZE(0. . . вг-пароля-пользователя))
                                           MATHES FOR EQUALITY

-- макросы
ALGORITHM MACRO                     ::=
BEGIN
TYPE NOTATION                       ::=  "PARAMETER" type
VALUE NOTATION                      ::=  value (VALUE OBJECT IDENTIFIER)
END -- макроса ALGORITHM

ENCRYPTED MACRO                     ::=
BEGIN

TYPE NOTATION                       ::=  type (БытьЗашифрованным)
VALUE NOTATION                      ::=  value (VALUE BIT STRING
    -- значение цепочки битов образуется применением
    -- процедуры шифровки к тем октетам,
    -- которые являются полным результатом
    -- кодирования (с использованием Основных
    -- правил кодирования НАС.1) значения,
    -- принадлежащего типу "БытьЗашифрованным" --

END

SIGNED MACRO                        ::=
BEGIN
TYPE NOTATION                       ::=  type (БытьПодписанным)
VALUE NOTATION                      ::=  value (VALUE

SEQUENCE{
    БытьПодписанным,
    ИдентификаторАлгоритма, -- алгоритма, использованного для выработки подписи
    ENGRYPTED OCTET STRING
    -- где цепочка октетов является
    -- результатом хэширования значения
    -- "БытьПодписанным" -- }
                                    )
END -- макроса SIGNED

SIGNATURE MACRO                     ::=
BEGIN

TYPE NOTATION                       ::=  type (ТипаПодпись)

VALUE NOTATION                      ::=  value (VALUE
    SEQUENCE{
        ИдентификаторАлгоритма,
        -- алгоритма, использованного для
        -- вычисления подписи
        ENCRYPTED OCTET STRING
        -- где цепочка октетов является некоторой
        -- функцией (например, сжатия
        -- или хэширования) значения
        -- типа "ТипаПодпись", которое может включать
        -- идентификатор алгоритма использования
        -- для вычисления подписи -- }
                                    )
END -- макроса SIGNATURE

PROTECTED MACRO                     ::=  SIGNATURE

END -- Определений Структуры Аутентификации
```

ПРИЛОЖЕНИЕ H

(к Рекомендации X.509)

Определение идентификаторов объектов алгоритмов

Данное Приложение не является составной частью настоящей Рекомендации.

В настоящем Приложении определяются идентификаторы объектов, присвоенные алгоритмам аутентификации и шифровки, если отсутствует формальный регистратор алгоритмов. Предполагается использование такого регистратора, как скоро он станет доступен. Определения имеют форму HAC.1-модуля, ИдентификаторыОбъектовАлгоритмов.

```
ИдентификаторыОбъектовАлгоритмов {joint-iso-ccitt ds(5) modules(1)
                                   algorithmObjectIdentifiers(8)}

DEFINITIONS ::=
BEGIN

EXPORTS
    алгоритмШифровки, алгоритмХэширования, алгоритмПодписи,
    rsa, квадратМод-n, квадратМод-nДляRSA;

IMPORTS
    алгоритм, структураАутентификации
        FROM ПолезныеОпределения {joint-iso-ccitt ds(5)modules(1)
                                  usefulDefinitions(0)}

    ALGORITHM FROM СтруктураАутентификации структураАутентификации;

-- категории идентификаторов объектов

алгоритмШифровки OBJECT IDENTIFIER      ::= {алгоритм 1}

алгоритмХэширования OBJECT IDENTIFIER   ::= {алгоритм 2}

алгоритмПодписи OBJECT IDENTIFIER       ::= {алгоритм 3}

-- алгоритмы

rsa ALGORITHM
    PARAMETER РазмерКлюча
    ::= {алгоритмШифровки 1}

РазмерКлюча ::= INTEGER

квадратМод-n ALGORITHM
    PARAMETER РазмерБлока
    ::= {алгоритмХэширования 1}

РазмерБлока ::= INTEGER
квадратМод-nLkzRSA ALGORITHM
    PARAMETER РазмерКлючаИБлока
    ::= {алгоритмПодписи 1}

РазмерКлючаИБлока ::= INTEGER

END -- Определений Идентификаторов Объектов Алгоритмов
```

СПРАВОЧНИК — ОПРЕДЕЛЕНИЕ АБСТРАКТНОЙ СЛУЖБЫ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

[1)] Рекомендация X.511 и ISO 9594-3 "Системы обработки информации — Взаимосвязь открытых систем — Справочник — Определение абстрактных служб" были разработаны в тесном сотрудничестве и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой в них справочной информацией может рассматриваться как единое целое, называемое *Справочником.* Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами могут служить прикладные процессы, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

- поставляемых различными производителями;
- находящихся под различным управлением;
- различной степени сложности;
- различных поколений.

0.3 В настоящей Рекомендации определяются возможности, обеспечиваемые Справочником его пользователям.

0.4 Приложение А содержит модуль на НАС.1, в котором приведены все определения, связанные с абстрактными службами.

1 Предмет рассмотрения и область применения

1.1 Настоящая Рекомендация определяет абстрактным образом службы, обеспечиваемые Справочником, как они видны извне Справочника.

1.2 Эта Рекомендация не специфицирует отдельные реализации или разработки.

РАЗДЕЛ 1 – *Общие вопросы*

2 Библиография

Рекомендация X.200 "Взаимосвязь Открытых Систем — Основная эталонная модель".

Рекомендация X.208 "Спецификация нотации абстрактного синтаксиса номер один (НАС.1)".

Рекомендация X.500 "Справочник — Обзор концепций, моделей и служб".

Рекомендация X.501 "Справочник — Модели".

Рекомендация X.518 "Справочник — Процедуры распределенной операции".

Рекомендация X.519 "Справочник — Спецификация протоколов".

Рекомендация X.520 "Справочник — Избранные типы атрибутов".

Рекомендация X.521 "Справочник — Избранные классы объектов".

Рекомендация X.509 "Справочник — Структура аутентификации".

Рекомендация X.219 "Удаленные операции — модель, нотация и определение услуг".

Рекомендация X.229 "Удаленные операции — Спецификация протоколов".

Рекомендация X.407 "Соглашения по определению абстрактных служб".

3 Определения

3.1 *Основные определения Справочника*

Настоящая Рекомендация использует следующие термины, определенные в Рекомендации X.500:

a) *Справочник;*

b) *информационная база Справочника (ИБС);*

c) *пользователь (Справочника).*

3.2 *Определения модели Справочника*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.501:

a) *системный агент Справочника;*

b) *агент пользователя Справочника.*

3.3 *Определения информационной базы Справочника*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.501:

a) *статья псевдонима;*

b) *информационное дерево Справочника;*

c) *статья (Справочника);*

d) *непосредственно предшествующий;*

e) *непосредственно предшествующая/щий статья/объект;*

f) *объект;*

g) *класс объектов;*

h) *статья объекта;*

i) *последующий;*

j) *предшествующий.*

3.4 *Определения статьи Справочника*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.501:

a) *атрибут;*

b) *тип атрибута;*

c) *значение атрибута;*

d) *проверка значения атрибута.*

3.5 *Определения имени*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.501:

a) *псевдоним; имя псевдонима;*

b) *выделенное имя;*

c) *имя (в Справочнике);*

d) *потенциальное имя;*

e) *относительно выделенное имя.*

3.6 *Определения распределенной операции*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.518:

a) *сцепление,*

b) *отсылка.*

3.7 *Определения абстрактных служб*

В настоящей Рекомендации определяются следующие термины:

a) *фильтр* — проверка наличия или значения некоторых атрибутов статьи в целях сужения области поиска;

b) *параметры службы* — параметры, передаваемые как часть абстрактной операции, которые ограничивают различные аспекты ее выполнения;

c) *пункт порождения* — пользователь, порождающий операцию.

4 Сокращения

В настоящей Рекомендации используются следующие сокращения:

ПЗА — проверка значения атрибута

ИБС — информационная база Справочника

ИДС — информационное дерево Справочника

САС — системный агент Справочника

АПС — агент пользователя Справочника

ОВИ — относительно выделенное имя

5 Соглашения

Настоящая Рекомендация использует соглашения по определению абстрактных служб, описанные в Рекомендации X.407.

РАЗДЕЛ 2 – *Абстрактные службы*

6 Обзор служб, предоставляемых Справочником

6.1 Как описано в Рекомендации X.501, службы Справочника обеспечиваются через пункты доступа АПС, каждый из которых действует от имени пользователя. Эти концепции изображены на рис. 1/X.511.

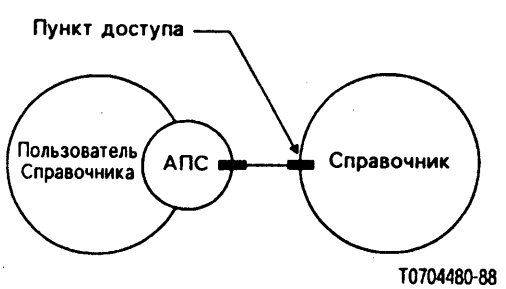


РИСУНОК 1/X.511

Доступ к Справочнику

6.2 В принципе, пункты доступа к Справочнику могут быть различных типов и, соответственно, обеспечивать различные комбинации служб. Необходимо рассматривать Справочник как *объект,* обеспечивающий *порты* нескольких типов. Каждый порт определяет конкретный вид взаимодействия, в котором может участвовать Справочник совместно с АПС. Каждый пункт доступа соответствует конкретной комбинации типов портов.

6.3 Используя обозначения, описанные в Рекомендации X.407, Справочник может быть определен следующим образом:

```
справочник
    OBJECT
        PORTS   { портЧтения [S],
                  портПоиска [S],
                  портМодификации [S]}
::=  ид-от-справочника
```

Справочник обеспечивает операции посредством Портов Чтения, которые обеспечивают чтение информации из отдельной поименованной статьи в ИБС; Портов Поиска, дающих возможность дополнительного "исследования" ИБС; и Портов Модификации, которые делают возможным модифицировать статьи в ИБС.

Примечание. — Предполагается, что в будущем у Справочника могут быть другие типы портов.

6.4 Подобным образом АПС (с точки зрения Справочника) может быть определен следующим образом:

```
апс
    OBJECT
        PORTS   { портЧтения [C],
                  портПоиска [C],
                  портМодификации [C]}
::=  ид-от-апс
```

АПС воспринимает службы, обеспечиваемые Справочником.

6.5 Порты, о которых говорилось в § 6.2—6.4, могут быть определены следующим образом:

```
портЧтения
    PORT
        CONSUMER INVOKES {
            Чтение, Сравнение, Отказ}
::=  ид-пт-чтения

портПоиска
    PORT
        CONSUMER INVOKES{
            Список, Поиск}
::=  ид-пт-поиска

портМодификации
    PORT
    CONSUMER INVOKES{
            ДобавлениеСтатьи, УдалениеСтатьи,
            МодификацияСтатьи,МодификацияОВИ}
        ::=  ид-пт-модификации
```

6.6 Операции, которые можно выполнить через портЧтения, портМодификации, портПоиска, определяются в § 9, 10 и 11 соответственно.

6.7 Эти порты используются только как метод структурированного описания служб Справочника. Согласованность с операциями Справочника специфицирована в Рекомендации X.519.

7 Типы информации

7.1 *Введение*

7.1.1 Этот параграф идентифицирует, а в некоторых случаях определяет некоторое число типов информации, которые впоследствии используются при определении операций Справочника. Типы информации, о которых идет речь, это те, которые являются общими для более чем одной операции, или же, возможно, будут такими в будущем, или же, довольно сложны или независимы, так что имеет смысл определить их отдельно от использующей их операции.

7.1.2 Многие типы информации, используемые в определениях, фактически описаны в других местах. В § 7.2 идентифицируются типы и указываются источники их определения. Каждый из оставшихся параграфов (7.3—7.10) идентифицирует и определяет тип информации.

7.2 *Типы информации, определенные в других местах*

7.2.1 В Рекомендации X.501 определены следующие типы информации:

a) **Атрибут;**

b) **ТипАтрибута;**

c) **ЗначениеАтрибута;**

d) **ПроверкаЗначенийАтрибута;**

e) **ВыделенноеИмя;**

f) **Имя;**

g) **ОтносительноВыделенноеИмя.**

7.2.2 В Рекомендации X.520 определен следующий тип информации:

a) **АдресВУровнеПредставлений.**

7.2.3 В Рекомендации X.509 определены следующие типы информации:

a) **Сертификат;**

b) **SIGNED;**

c) **ВетвьСертификации.**

7.2.4 В Рекомендации X.219 определен следующий тип информации:

a) **ИдВызова.**

7.2.5 В Рекомендации X.518 определены следующие типы информации:

a) **ПродвижениеОперации;**

b) **ССылкаНаПродолжение.**

7.3 *Общие аргументы*

7.3.1 Информация **ОбщиеАргументы** может быть представлена для квалификации вызова каждой из операций, которую может выполнять Справочник.

```
Общие Аргументы ::= SET{
    [30] ПараметрыСлужбы DEFAULT{ },
    [29] ПараметрыБезопасности DEFAULT { },
    реквестор [28] ВыделенноеИмя
                        OPTIONAL,
    [27] ПродвижениеОперации DEFAULT неНачата,
    оВИобъектаПсевдонима [26] INTEGER OPTIONAL,
    расширения [25] SET OF РАСШИРЕНИЕ OPTIONAL }

Расширение ::= SET{
    идентификатор     [0] INTEGER,
    критический       [1] BOOLEAN DEFAULT FALSE,
    элемент           [2] ANY DEFINED BY идентификатор}
```

7.3.2 В § 7.3.2.1—7.3.2.4 определяются значения различных компонентов.

7.3.2.1 Компонент **ПараметрыСлужбы** специфицирован в § 7.5. Его отсутствие рассматривается как эквивалент пустого множества параметров управления.

7.3.2.2 Компонент **ПараметрыБезопасности** специфицирован в § 7.9. Его отсутствие рассматривается как эквивалент пустого множества параметров безопасности.

7.3.2.3 Реквестор **ВыделенноеИмя** идентифицирует пункт порождения конкретной абстрактной операции. Он содержит имя пользователя, идентифицированное во время привязывания к Справочнику. Он может понадобиться, если надо будет подписать запрос (см. § 7.10), и должен содержать имя пользователя, инициировавшего запрос.

7.3.2.4 ПродвижениеОперации определяет роль, которую предстоит сыграть САС в распределенном осуществлении запроса. Более подробно этот параметр специфицирован в Рекомендации X.518.

7.3.2.5 Компонент оВИобъектаПсевдонима указывает САС, что компонент, задающий объект операции, был создан путем переименования псевдонима при более ранней попытке выполнения операции. Целое значение определяет число ОВИ в объекте, которые получаются при переименовании псевдонима. (Значение должно было быть установлено в отсылающем ответе, выработанном при предыдущей операции.)

7.3.2.6 Компонент расширения обеспечивает механизм для выражения стандартизованного расширения формы аргумента абстрактной-операции Справочника.

Примечание. — Форма результата такой расширенной абстрактной операции идентична форме результата при нерасширенном варианте. (Тем не менее, результат отдельной расширенной абстрактной операции может отличаться от формы при нерасширенном варианте.)

Подкомпоненты определяются в § 7.3.2.6.1—7.3.2.6.3.

7.3.2.6.1 Идентификатор служит для идентификации конкретного расширения. Значения этого компонента будут выработаны только в следующих вариантах настоящей серии Рекомендации.

7.3.2.6.2 Подкомпонент критический используется инициатором расширенной абстрактной-операции для указания того, что допустимо выполнение только расширенного варианта абстрактной операции (то есть нерасширенный вариант неприемлем). В этом случае расширение является критическим расширением. Если Справочник или некоторая его часть не в состоянии выполнить критическое расширение, он возвращает недоступноеКритическоеРасширение (как ОшибкаСлужбы или КвалификаторЧастичногоРезультата). Если Справочник не в состоянии выполнить расширение, не являющееся критическим, он игнорирует наличие расширения.

7.3.2.6.3 Подкомпонент элемент обеспечивает информацию, необходимую Справочнику для выполнения расширенной формы абстрактной-операции.

7.4 *Общие результаты*

7.4.1 Информация ОбщиеРезультаты должна быть представлена для квалификации результатов каждой операции чтения, которую может выполнить Справочник.

```
ОбщиеРезультаты ::=  SET {
    [30] ПараметрыБезопасности      OPTIONAL,
    исполнитель [29] ВыделенноеИмя  OPTIONAL,
    псевдонимПереименован [28]      BOOLEAN
                        DEFAULT FALSE}
```

7.4.2 В § 7.4.2—7.4.2.3 определяются значения различных компонентов.

7.4.2.1 Компонент ПараметрыБезопасности специфицируется в § 7.9. Его отсутствие рассматривается как эквивалент пустого множества параметров безопасности.

7.4.2.2 ВыделенноеИмя Исполнитель идентифицирует исполнителя конкретной операции. Оно может понадобиться, если надо будет подписать результат (см. § 7.10), и должно содержать имя пользователя, подписавшего результат.

7.4.2.3 Компонент псевдонимПереименован устанавливается в TRUE, если потенциальное имя объекта или базового объекта, в отношении которого выполняется операция, включено в псевдоним, подвергшийся переименованию.

7.5 *Параметры Службы*

7.5.1 Параметр ПараметрСлужбы содержит параметры (если таковые имеются), которые направляют или ограничивают обеспечение службы.

```
ПараметрыСлужбы ::=  SET {
    варианты [0] BIT STRING {
        предпочтительноСцепление (0),
        запрещеноСцепление (1),
        локальнаяОбластьПрименения (2),
        неИспользоватьКопий (3),
        неПереименовыватьПсевдонимов (4)}
        DEFAULT { },

приоритет [1] INTEGER {
    низкий (0),
    средний (1),
    высокий (2) } DEFAULT средний,
```

```
ограничениеВремени  [2]  INTEGER OPTIONAL,
ограничениеДлины    [3]  INTEGER OPTIONAL,
областьПримененияОтсылок [4] INTEGER {
                                 оус (0),
                                 страна (1)}
                                 OPTIONAL }
```

7.5.2 В § 7.5.2.1—7.5.2.5 определяются значения различных компонентов.

7.5.2.1 Компонент **варианты** содержит несколько признаков, каждый из которых, если он установлен, утверждает, что условие удовлетворено таким образом:

a) **предпочтительноСцепление** указывает, что при выполнении службы предпочтение должно отдаваться сцеплению, а не отсылкам. Справочник не обязан следовать этому предписанию;

b) **запрещеноСцепление** указывает, что сцепление, как и другие методы распределения запроса по Справочнику, запрещены;

c) **локальнаяОбластьПрименения** указывает, что операция должна быть ограничена локальной областью. Определение этого варианта само является локальным вопросом. Например, внутри отдельного САС или отдельного ОУС;

d) **неИспользоватьКопий** указывает, что скопированная информация (как определяется в Рекомендации X.518) не должна использоваться при выполнении услуги;

e) **неПереименовыватьПсевдонимов** указывает, что любой псевдоним, используемый для идентификации статьи и затрагиваемый операцией, не должен переименовываться.

Примечание. — Это требуется для указания того, что операция распространяется на саму статью псевдонима, а не на ту, на которую указывает статья псевдонима; может использоваться, например, для чтения статьи псевдонима.

Если этот компонент опущен, то предполагается следующее: нет предпочтения сцеплению, но сцепление и не запрещено, отсутствует ограничение на область применения операций, разрешено использование копий, и псевдонимы должны переименоваться (за исключением операций модификации, при которых псевдонимы никогда не должны переименовываться).

7.5.2.2 **Приоритет** (низкий, средний или высокий) поставки службы. Заметим, что это не гарантированная служба, поскольку целиком весь Справочник не обеспечивает создания очередей. В нижеследующих уровнях с понятием "приоритет" нет никакой подразумеваемой связи.

7.5.2.3 **ОграничениеВремени** лимитирует максимальный промежуток времени в секундах, в течение которого должна быть обеспечена услуга. Если ограничение не может быть выполнено, выдается сообщение об ошибке. Если этот компонент опущен, то ограничение времени не устанавливается. Если при выполнении операций Список или Поиск лимит времени оказывается исчерпан, то в качестве результата будет выдана совокупность накопленных к этому моменту результатов.

Примечание. — В этот компонент не входит то время, которое было затрачено на обработку запроса в течение заданного промежутка времени; любое число САС может быть вовлечено в обработку в течение этого промежутка времени.

7.5.2.4 **ОграничениеДлины** применяется только к операциям Список и Поиск. Он указывает максимальное число возвращаемых объектов. В случае превышения ограничения длины результатом операции Список или Поиск будет произвольная выборка накопленных результатов, численно равная ограничению длины. Любые прочие результаты будут аннулированы.

7.5.2.5 **ОбластьПримененияОтсылок** указывает область, к которой может относиться отсылка, возвращаемая от САС. В зависимости от того, выбрано ли значение оус или Страна, будут возвращены только отсылки к другим САС внутри выбранной области.

Это относится также к отсылкам в параметрах **ОшибкаОтсылки** и не выявлено результатов операций Список и Поиск.

7.5.3 Различные комбинации компонентов **приоритет**, **ограничениеВремени**, **ограничениеДлины** могут приводить к противоречиям. Например, малое значение ограничения времени может противоречить низкому приоритету, большое значение ограничения длины может противоречить малому значению ограничения времени и т.д.

7.6 *Избирание информации из статьи*

7.6.1 Параметр **ИзбираниеИнформацииИзСтатьи** указывает, какая информация запрашивается из статьи в службах чтений.

```
ИзбиранИнформацииИзСтатьи ::= SET{
    типыАтрибутов
        CHOICE{
                всеАтрибуты [0] NULL,
                избрать [1]}SET OF ТипыАтрибутов
                -- пустое множество означает, что никакие
                -- атрибуты не запрошены}
        DEFAULT всеАтрибуты NULL,

    типыИнфо [2] INTEGER{
        толькоТипыАтрибутов (0),
        типыИЗначенияАтрибутов (1)}
        DEFAULT типыИЗначенияАтрибутов}
```

7.6.2 В § 7.6.2.1 — 7.6.2.2 определяются значения различных компонентов.

7.6.2.1 Компонент типыАтрибутов специфицирует набор атрибутов, относительно которых запрошена информация:

a) если выбран вариант избрать, то перечисляются интересующие атрибуты. Если атрибут присутствует, то информация о выбранных атрибутах будет возвращена. Если ни один из выбранных атрибутов не присутствует, то возвращается только сообщение ОшибкаАтрибута с указанием нетТакогоАтрибута;

b) если выбран вариант всеАтрибуты, то это означает, что запрашивается информация относительно всех атрибутов статьи.

Информация об атрибутах возвращается, только если удовлетворены права доступа. Если право доступа запрещает чтение всех запрошенных атрибутов, то возвращается только сообщение ОшибкаБезопасности (с указанием недостаточноеПравоДоступа).

7.6.2.2 Компонент типыИнфо специфицирует, запрошена ли информация как о типе атрибута, так и о значении (по умолчанию) или только информация о типе атрибута. Если компонент **типыАтрибутов** (§ 7.6.2.1) таков, что не запрашивает ни одного атрибута, то этот последний компонент не имеет смысла.

7.7 *Информация из статьи*

7.7.1 Параметр ИнформацияИзСтатьи передает выбранную из статьи информацию.

```
ИнформацияИзСтатьи ::= SEQUENCE{
    ВыделенноеИмя,
    изСтатьи BOOLEAN DEFAULT TRUE,
    SET OF CHOICE{
        ТипАтрибута,
        Атрибут}OPTIONAL}
```

7.7.2 ВыделенноеИмя статьи присутствует всегда.

7.7.3 Параметр изСтатьи указывает, получена ли информация из статьи (TRUE) или из копии статьи (FALSE).

7.7.4 Смотря по обстоятельствам, параметр содержит ТипыАтрибутов или Атрибуты, каждый из которых может быть или один, или сопровождаться одним или более значениями атрибутов.

7.8 *Фильтр*

7.8.1 Параметр Фильтр представляет собой тест, которому либо удовлетворяет, либо не удовлетворяет отдельная статья. Фильтр выражается в терминах проверок на наличие или значение каких-то конкретных атрибутов статьи; тест удовлетворяется тогда и только тогда, когда в результате проверок ему присваивается значение TRUE.

Примечание. — Фильтр может быть TRUE, FALSE или неопределенным.

```
Фильтр      ::= CHOICE{
    элемент [0] ЭлементФильтра,
    and     [1] SET OF Фильтр,
    or      [2] SET OF Фильтр,
    not     [3] Фильтр}
```

```
ЭлементФильтра  ::=  CHOICE{
    равенство            [0] ПроверкаЗначенияАтрибута,
    подцепочки           [1] SEQUENCE{
        тип                  ТипАтрибута,
        цепочки SEQUENCE OF CHOICE{
                         начальный [0]  ЗначениеАтрибута,
                         любой     [1]  ЗначениеАтрибута,
                         конечный  [2]  ЗначениеАтрибута }},
    большеИлиРавно       [2] ПроверкаЗначенияАтрибута,
    меньшеИлиРавно       [3] ПроверкаЗначенияАтрибута,
    присутствует         [4] ТипАтрибута,
    приблизительноеСовпадение [5] ПроверкаЗначенияАтрибута}
```

7.8.2 Фильтр является либо ЭлементомФильтра (см. § 7.8.3), либо выражением, содержащим более простые Фильтры, соединенные между собой с помощью логических операций and, or, not. Фильтр считается неопределенным, если он является ЭлементомФильтра, который неопределен, либо если он состоит из одного или из более простых Фильтров, которые все не определены. В противном случае, если Фильтр является:

a) элементом, то он принимает значение TRUE тогда и только тогда, когда соответствующий Элемент Фильтра имеет значение TRUE;

b) and, то он принимает значение TRUE, если ни один из вложенных Фильтров не FALSE;

Примечание. — Следовательно, если нет вложенных Фильтров, то and приводит к TRUE.

c) or, то он принимает значение FALSE, если ни один из вложенных Фильтров не TRUE;

Примечание. — Следовательно, если нет вложенных Фильтров, то or равно FALSE.

d) not, то он принимает значение TRUE тогда и только тогда, когда значение вложенного Фильтра равно FALSE.

7.8.3 ЭлементФильтра является проверкой наличия или значения(ний) атрибута конкретного типа в тестируемой статье. Каждая такая проверка либо TRUE, либо FALSE, либо не определена.

7.8.3.1 Каждый ЭлементФильтра включает ТипАтрибута, который идентифицирует конкретный рассматриваемый атрибут.

7.8.3.2 Каждая проверка значения такого атрибута определена только, если известен ТипАтрибута, а предполагаемое Значение(я)Атрибута согласуется с синтаксисом атрибута, определенным для этого типа атрибута.

Примечание 1. — Если эти требования не удовлетворены, то ЭлементФильтра не определен.

Примечание 2. — Ограничения на право доступа могут потребовать, чтобы ЭлементФильта рассматривался как неопределенный.

7.8.3.3 Проверка значения атрибута осуществляется путем использования правил сопоставления, относящихся к синтаксису атрибута, определенному для данного типа атрибута. Правило сопоставления, не определенное для конкретного синтаксиса атрибута, не может быть применено для проверок этого атрибута.

Примечание. — Если это требование не удовлетворено, то ЭлементФильтра не определен.

7.8.3.4 ЭлементФильтра может быть не определен (как это описано выше в § 7.8.3.2 и 7.8.3.3). В противном случае, если в ЭлементеФильтра выполняется проверка на:

a) равенство, то результатом будет TRUE тогда и только тогда, когда существует значение атрибута, равное тому, на которое выполняется проверка;

b) подцепочки, то результатом будет TRUE тогда и только тогда, когда существует значение атрибута, в котором присутствуют специфицированные подцепочки и при том в заданном порядке. Подцепочки не должны перекрываться и могут (но не обязаны) быть отделены от концов значения атрибута и друг от друга нулем или более элементами цепочки.

Если используется вариант начальный, то подцепочка должна совпадать с первой подцепочкой в значении атрибута; если используется вариант конечный, то подцепочка должна совпадать с последней подцепочкой в значении атрибута; наконец, в случае любой, совпадение может иметь место с любой подцепочкой в значении атрибута.

c) большеИлиРавно, то результатом будет TRUE тогда и только тогда, когда при относительном упорядочении (как оно определено в соответствующем алгоритме упорядочения) помещает представленное значение перед любым значением атрибута или считает равным ему;

d) меньшеИлиРавно, то результатом будет TRUE тогда и только тогда, когда при относительном упорядочении (как оно определено в соответствующем алгоритме упорядочения) помещает представленное значение после любого значения атрибута или считает равным ему;

e) присутствует, то результатом будет TRUE тогда и только тогда, когда такой атрибут присутствует в статье;

f) приблизительноеСовпадение, то результатом будет TRUE тогда и только тогда, когда имеется значение атрибута, совпадающее с представленным в соответствии с некоторым локально определенным алгоритмом приблизительного совпадения (например, варианты написания, фонетическое соответствие и др.). В настоящей версии Рекомендации не дается никаких специфических указаний для понятия "приблизительное совпадение". Если "приблизительное совпадение" не определено, то для данного ЭлементаФильтра совпадение должно трактоваться как равенство.

7.9 *Параметры безопасности*

7.9.1 ПараметрыБезопасности определяют выполнение различных мер безопасности, связанных с функционированием Справочника.

Примечание. — Эти параметры посылаются от отправителя получателю. Если параметры появляются в аргументе абстрактной-операции, то реквестор является отправителем, а исполнитель — получателем. В результате роли меняются на противоположные.

```
ПараметрыБезопасности ::= SET{
    ветвьСертификации       [0]
    ВетвьСертификации       OPTIONAL,
    имя                   [1] ВыделенноеИмя
                              OPTIONAL,
    время                 [2] ВремяОтГринвича OPTIONAL,
    случайноеЧисло        [3] BIT STRING OPTIONAL,
    цель                  [4] ЗапросЗащиты OPTIONAL
                                        }

ЗапросЗащиты              ::= INTEGER{
                                  отсутствует (0),
                                  подписанный (1)}
```

7.9.2 В § 7.9.2.1 — 7.9.2.5 определяются значения различных компонентов.

7.9.2.1 Компонент ВетвьСертификации состоит из сертификата отправителя и необязательной последовательности сертификатных пар. Сертификат используется для ассоциирования общедоступного ключа с выделенным именем отправителя и может использоваться для проверки подписи под аргументом или результатом. Этот параметр должен присутствовать, если подписаны аргумент или результат. Последовательность сертификатных пар состоит из перекрестных сертификатов сертификатных органов. Эта последовательность сертификатных пар используется, для того чтобы было возможным проверить подлинность сертификата отправителя. Она не требуется, если получатель входит в тот же сертификатный орган, что и отправитель. Если получателю требуется набор действительных сертификатных пар, а этот параметр отсутствует, то либо получатель отвергает подпись под аргументом или результатом, либо пытается сам сгенерировать ветвь сертификации; выбор одной из этих двух альтернатив является локальным вопросом.

7.9.2.2 Имя является выделенным именем первого предполагаемого получателя аргумента или результата. Например, если АПС генерирует подписанный аргумент, то имя является выделенным именем того САС, которому подается эта операция.

7.9.2.3 Время является тем временем, в течение которого подпись имеет силу; применяется в тех случаях, когда используется подписывание аргументов. Оно используется вместе со случайным числом для обнаружения попыток "розыгрыша".

7.9.2.4 Компонент случайноеЧисло является числом, которое должно быть различным для каждого мандата, еще не утратившего своей силы. Оно используется совместно с параметром "время" для обнаружения попыток "розыгрыша" в тех случаях, когда аргумент или результат были подписаны.

7.9.2.5 Цель ЗапросЗащиты может присутствовать только в запросе на операцию, подлежащую выполнению, и характеризует степень защищенности ответа, предпочтительную для реквестора. Предусмотрено два уровня: отсутствует (нет запроса на защиту) и подписанный (от Справочника требуется подписать результат; по умолчанию предусмотрен этот вариант). Степень защиты, фактически обеспечиваемая результату, задается формой ответа и может быть равна или ниже запрашиваемой, что зависит от возможностей Справочника.

7.10 *OPTIONALLY-SIGNED*

7.10.1 Тип информации **OPTIONALLY-SIGNED** характеризует информацию, значения которой могут быть сопровождены (при желании генератора) его цифровой подписью. Эта возможность специфицируется с помощью следующего макроса:

```
OPTIONALLY-SIGNED MACRO ::=
BEGIN

TYPE NOTATION              ::=   type (Type)
VALUE NOTATION             ::=   value (VALUE
        CHOICE { Type, SIGNED Type } )

END
```

7.10.2 Макрос **SIGNED**, который описывает структуру подписанной формы информации, специфицирован в Рекомендации X.509.

8 Операции привязывания и отвязывания

Операции **ПривязыванияКСправочнику** и **ОтвязыванияОтСправочника**, определяемые в § 8.1 и § 8.2 соответственно, используются в АПС в начальный и конечный моменты каждого периода доступа к Справочнику.

8.1 *Привязывание к Справочнику*

8.1.1 Операция **ПривязыванияКСправочнику** используется в начальный момент периода доступа к Справочнику.

```
ПривязываниеКСправочнику     ::=     ABSTRACT-BIND
    TO { портЧтения, портПоиска, портМодификации }
    BIND
    ARGUMENT                         АргументПривязыванияКСправочнику
    RESULT                           РезультатПривязыванияКСправочнику
    BIND-ERROR                       ОшибкаПривязыванияКСправочнику

АргументПривязыванияКСправочнику  ::= SET {
    удостоверения            [0]     Удостоверения OPTIONAL,
    версии                   [1]     Версии DEFAULT {v1988}

Удостоверения  ::=           CHOICE {
    простое                  [0]     ПростоеУдостоверение,
    строгое                  [1]     СтрогоеУдостоверение,
    внешняяПроцедура         [2]     EXTERNAL}

ПростоеУдостоверение         ::=     SEQUENCE {
    имя                      [0]     ВыделенноеИмя,
    действительность         [1]     SET {
        время1               [0]     ВремяОтГринвича OPTIONAL,
        время2               [1]     ВремяОтГринвича OPTIONAL,
        случайноеЧисло1      [2]     BIT STRING OPTIONAL,
        случайноеЧисло2      [3]     BIT STRING OPTIONAL } OPTIONAL,
        -- в большинстве случаев аргументы для
        -- времени и случайного числа зависят
        -- от примененного в диалоге механизма
        -- в результате двусторонних соглашений

    пароль [2] OCTET STRING OPTIONAL }
        -- значением может быть не защищенный
        -- пароль или Защищенность1 или Защищенность2,
        -- как это специфицировано в Рекомендации X.509.

СтрогоеУдостоверение         ::=     SET {
    ветвь-сертификации [0]           ВетвьСертификации
                                     OPTIONAL,
    мандат-на-привязывание   [1]     Мандат }
```

```
Мандат      ::=          SIGNED SEQUENCE{
    алгоритм             [0]  ИдентификаторАлгоритма,
    имя                  [1]  ВыделенноеИмя,
    время                [2]  ВремяОтГринвича,
    случайноеЧисло       [3]  BIT STRING}

Версии      ::=          BIT STRING { v1988(0)}

РезультатПривязыванияКСправочнику  ::= АргументПривязыванияКСправочнику

ОшибкаПривязыванияКСправочнику ::= SET{
    версии [0] Версии DEFAULT  v1988,
    CHOICE{
        ошибкаСлужбы        [1]  ТрудностиСлужбы,
        ошибкаБезопасности  [2]  ТрудностиБезопасности
}}
```

8.1.2 В § 8.1.2.1 — 8.1.2.2 определяется смысл различных аргументов.

8.1.2.1 Удостоверения, входящие в АргументПривязыванияКСправочнику, позволяют Справочнику установить подлинность пользователя. Они могут быть или простыми, или строгими (как это описано в Рекомендации X.509), или определенными извне (внешнейПроцедурой).

8.1.2.1.1 ПростоеУдостоверение состоит из имени (это всегда выделенное имя объекта) и (по выбору) пароля. Это обеспечивает ограниченный уровень безопасности. Если пароль защищен так, как это описано в § 5 Рекомендации X.509, то ПростоеУдостоверение включает имя, пароль и (по выбору) время и/или случайные числа, которые используются для обнаружения розыгрыша. В некоторых случаях защищенный пароль может быть проверен объектом, которому пароль известен, только после локального регенирирования защиты его собственной копии пароля и вычисления результата на основании значения в аргументе привязывания (пароля). В других случаях может быть использовано непосредственное сравнение.

8.1.2.1.2 СтрогоеУдостоверение содержит мандат на привязывание и дополнительные сертификат и последовательности перекрестных сертификатов сертификатного органа (как это определено в Рекомендации X.509). Это дает возможность Справочнику подтвердить подлинность запроса на привязывание и наоборот.

Аргументы мандата-на-привязывание используются следующим образом: алгоритм является идентификатором алгоритма, использованного для подписания информации; имя является именем предполагаемого получателя. Параметр время содержит время, в течение которого действителен мандат; случайноеЧисло является числом, которое должно быть различным для каждого еще действительного мандата и может быть использовано получателем для обнаружения и попыток розыгрыша.

8.1.2.1.3 Если используется внешняяПроцедура, то тогда семантика используемой схемы аутентификации выходит за пределы настоящего документа.

8.1.2.2 Аргумент Версии, входящий в АргументПривязыванияКСправочнику, идентифицирует версии службы, в которых готов участвовать АПС. Для настоящей версии протокола значение должно равняться **v1988**(0).

8.1.2.3 Переход к последующим версиям Справочника должен быть облегчен за счет того, что:

a) любой элемент АргументаПривязыванияКСправочнику, отличный от определенных в данной Рекомендации, будет принят и проигнорирован;

b) дополнительные варианты именованных битов, входящих в АргументПривязыванияКСправочнику (например, Версии), которые сейчас не определены, будут приняты и проигнорированы.

8.1.3 Если запрос на привязывание будет успешным, то должен быть возвращен результат. Параметры результата имеют значения, определенные в § 8.1.3.1 и 8.1.3.2.

8.1.3.1 Удостоверения, входящие в РезультатПривязыванияКСправочнику, позволяют пользователю идентифицировать САС. Они позволяют передать АПС информацию, идентифицирующую САС (непосредственно обеспечивающего службы Справочника). Они должны иметь ту же форму (то есть CHOICE), что и представленная пользователем.

8.1.3.2 Параметр Версии, входящий в РезультатПривязыванияКСправочнику, указывает, какая из версий службы, запрошенной АПС, в действительности обеспечивается этим САС.

8.1.4 Если запрос на подключение заканчивается безуспешно, то будет возвращена ошибка привязывания, как это определено в § 8.1.4.1 и 8.1.4.2.

8.1.4.1 Параметр Версии, входящий в ОшибкаПривязыванияКСправочнику, указывает, какая версия поддерживается данным САС.

8.1.4.2 ОшибкаСлужбы или ошибкаБезопасности будут поставлены следующим образом:

— ошибкаБезопасности — неприемлемаяАутентификация
недействительноеУдостоверение

— ошибкаСлужбы — не обеспечена

8.2 *Отвязывание от Справочника*

8.2.1 Операция ОтвязыванияОтСправочника используется в заключительный момент периода доступа к Справочнику.

```
ОтвязываниеОтСправочника  ::= ABSTRACT-UNBIND
    FROM { портЧтения, портПоиска, портМодификации}
```

8.2.2 ОтвязываниеОтСправочника не имеет аргументов.

9 Операции чтения из Справочника

Имеются две операции "подобные чтению": Чтение и Сравнение, определяемые в § 9.1 и 9.2 соответственно. Операция Отказ, определяемая в § 9.3, сгруппирована вместе с операциями Чтения для удобства.

9.1 *Чтение*

9.1.1 Операция Чтение используется для извлечения информации из явно идентифицированной статьи. Кроме того, она может использоваться для проверки выделенного имени. Аргументы операции могут быть по выбору подписаны (см. § 7.10) реквестором. Справочник может подписать результат, если это было специфицировано в запросе.

```
Чтение ::= ABSTRACT-OPERATION
    ARGUMENT        АргументЧтения
    RESULT РезультатЧтения
    ERRORS{
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности}

АргументЧтения  ::= OPTIONALLY-SIGNED SET {
    объект      [0] имя,
    избирание   [1] Избирание F[13] ИзбиранииИнформацииИзСтатьи
                    DEFAULT { }
    COMPONENTS OF Общие Аргументы}

РезультатЧтения ::= OPTIONALLY-SIGNED SET {
    статья      [0] ИнформацияИзСтатьи,
    COMPONENTS OF   ОбщиеРезультаты }
```

9.1.2 В § 9.1.2.1 — 9.1.2.3 определяется смысл различных аргументов.

9.1.2.1 Аргумент объект идентифицирует статью объекта, из которой запрашивается информация. Если Имя содержит один или более псевдонимов, они переименовываются (если это не запрещено соответствующими параметрами службы).

9.1.2.2 Аргумент избирание указывает, какая информация запрашивается из статьи (см. § 7.6).

9.1.2.3 ОбщиеАргументы (см. § 7.3) включают спецификацию параметров службы, применяемых к запросу. Для целей этой операции компонент ограничениеДлины неприменим и игнорируется, если он предоставляется.

9.1.3 Если запрос завершился успешно, то будет возвращен результат. Параметры результата имеют смысл, определяемый в § 9.1.3.1 и § 7.4.

9.1.3.1 Параметр результата статья содержит запрошенную информацию (см. § 7.7).

9.1.4 Если запрос закончится безуспешно, то будет сообщена одна из перечисленных ошибок. Если не может быть возвращен ни один из явно перечисленных атрибутов, то выдается сообщение ОшибкаАтрибута с причиной нетТакогоАтрибута. Обстоятельства, при которых поступит сообщение о других ошибках, описаны в § 12.

9.2 *Сравнение*

9.2.1 Сравнение используется для сравнения значения (которое поставляется как аргумент запроса) со значением (значениями) конкретного типа атрибута в конкретной статье объекта. Аргументы операции могут быть по выбору подписаны (см. § 7.10) реквестором. Справочник может подписать результат, если это было специфицировано в запросе.

```
Сравнение ::=          ABSTRACT-OPERATION
    ARGUMENT           АргументСравнения
    RESULT             РезультатСравнения
    ERRORS{
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности }

АргументСравнения  ::= OPTIONALLY-SIGNED
SET{
    объект         [0] Имя,
    потенциальный  [1] ПроверкаЗначенияАтрибута,
    COMPONENTS OF  ОбщиеАргументы }

РезультатСравнения ::= OPTIONALLY-SIGNED
SET{
    ВыделенноеИмя      OPTIONAL,
    сопоставим     [0] BOOLEAN,
    изСтатьи       [1] BOOLEAN DEFAULT TRUE,
    COMPONENTS OF  ОбщиеРезультаты }
```

9.2.2 В § 9.2.2.1 — 9.2.2.3 определяются значения различных аргументов.

9.2.2.1 Аргумент объект является именем конкретной интересующей статьи объекта. Если Имя содержит один или более псевдонимов, они переименовываются (если это не запрещено соответствующими параметрами службы).

9.2.2.2 Аргумент потенциальный идентифицирует тип атрибута и значение, которое должно сравниваться со значением в статье.

9.2.2.3 ОбщиеАргументы (см. § 7.3) специфицируют параметры службы, применяемые к запросу. Для целей этой операции компонент ограничениеДлины неприменим и игнорируется, если он представляется.

9.2.3 Если запрос завершился успешно (то есть сравнение действительно выполнено), то будет возвращен результат. Параметры результат имеют смысл, определяемый в § 9.2.3.1, 9.2.3.2 и 7.4.

9.2.3.1 ВыделенноеИмя присутствует, если псевдоним был переименован. Оно представляет собой выделенное имя самого объекта.

9.2.3.2 Параметр сопоставим содержит результат сравнения. Параметр принимает значение TRUE, если параметры были сравнены и оказались сопоставимыми, и значение FALSE — в противном случае.

9.2.3.3 Если изСтатьи имеет значение TRUE, информация сравнивалась со статьей, если — FALSE, то часть информации сравнивалась с копией.

9.2.4 Если запрос закончится безуспешно, то будет сообщена одна из перечисленных ошибок. Обстоятельства, в зависимости от которых будут выданы сообщения о конкретных ошибках, определены в § 12.

9.3 *Отказ*

9.3.1 От операций, опрашивающих Справочник, можно отказаться, используя операцию Отказ, если пользователя больше не интересует результат.

```
Отказ ::=              ABSTRACT-OPERATION
    ARGUMENT           АргументОтказа
    RESULT             РезультатОтказа
    ERRORS { НевыполненныйОтказ }

АргументОтказа     ::= SEQUENCE{
    идВызова       [0] ИдВызова }

РезультатОтказа    ::= NULL
```

9.3.2 Имеется единственный аргумент, идВызова, идентифицирующий операцию, от которой отказываются. Значение идВызова такое же, как и идВызова в той операции, от которой отказываются.

9.3.3 Если запрос завершился успешно, то будет возвращен результат, хотя он и не содержит никакой информации. Исходная операция закончится безуспешно с ошибкой Отказано.

9.3.4 Если запрос завершится безуспешно, то сообщается ошибка НевыполненныйОтказ. Эта ошибка описана в § 12.3.

9.3.5 Отказ применим только к опрашивающим операциям, то есть к операциям чтения, сравнения, списка и поиска.

9.3.6 САС может локально прекратить выполнение операции. Если САС создаст цепочку или многоадресную рассылку к другим САС, то он может сам запросить другие САС об отказе от операции. Если САС предпочтет не отказываться от операции, то он вернет ошибку НевыполненныйОтказ.

10 Операции поиска в Справочнике

Имеются две операции "подобные поиску": Список и Поиск, определяемые в § 10.1 и § 10.2 соответственно.

10.1 *Список*

10.1.1 Операция Список используется для получения списка статей, непосредственно следующих за явно идентифицированной статьей. При некоторых обстоятельствах возвращаемый список может быть неполным. Аргументы операции могут быть по выбору подписаны (см. § 7.10) реквестором. Справочник может подписать результат, если это было специфицировано в запросе.

```
Список ::=          ABSTRACT-OPERATION
    ARGUMENT                        АргументСписка
    RESULT    РезультатСписка
    ERRORS {

ОшибкаАтрибута                      ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности }

АргументСписка    ::=               OPTIONALLY-SIGNED SET {
    объект          [0]             Имя,
    COMPONENTS OF ОбщиеАргументы }

РезультатСписка    ::=              OPTIONALLY-SIGNED
CHOICE {
    ИнфоСписка  SET {
    ВыделенноеИмя  OPTIONAL,
    подчиненные    [1]              SET OF SEQUENCE {
        ОтносительноВыделенноеИмя,
        статьяПсевдонима [0]        BOOLEAN DEFAULT FAISE,
        изСтатьи    [1]             BOOLEAN DEFAULT TRUE },
квалификаторЧастичногоРезультата [2]
                                    КвалификаторЧастичногоРезультата  OPTIONAL
    COMPONENTS OF ОбщиеРезультаты },
    некореллированныйИнфоСписка    [0] SET OF
                                    РезультатСписка}

КвалификаторЧастичногоРезультата ::= SET{
    трудностиОграничения       [0]  ТрудностиОграничения
        OPTIONAL,
    неисследовано              [1]      SET  OF
        ССылкаНаПродолжение OPTIONAL,
    недоступноеКритическоеРасширение [2]   BOOLEAN DEFAULT FAISE }

ТрудностиОграничения ::= INTEGER {
    превышеноОграничениеВремени (0),
    превышеноОграничениеДлины (1),
    превышеноАдминистративноеОграничение (2) }
```

10.1.2 В § 10.1.2.1 и § 7.3 определяются значения различных аргументов.

10.1.2.1 Аргумент объект идентифицирует статью объекта (или, возможно, корня), для которой должны быть перечислены непосредственно последующие за ней статьи. Если Имя содержит один или более псевдонимов, то они переименовываются (если это не запрещено соответствующими параметрами службы).

10.1.3 Запрос завершается успешно, если местонахождение объекта определено, вне зависимости от того, имеется ли для возврата подчиненная информация. Параметры результата имеют смысл, определенный в § 10.1.3.1—10.1.3.4 и 7.4.

10.1.3.1 ВыделенноеИмя присутствует, если псевдоним был переименован. Оно представляет выделенное имя самого объекта.

10.1.3.2 Параметр подчиненные передает информацию о непосредственно последующих статьях (если они имеются) именованной статьи. Если какие-либо из подчиненных статей являются статьями псевдонима, то они не переименовываются.

10.1.3.2.1 ОтносительноВыделенноеИмя является таковым подчиненной статьи.

10.1.3.2.2 Параметр изСтатьи указывает, получена ли информация из статьи (TRUE) или из копии статьи (FALSE).

10.1.3.2.3 Параметр статьяПсевдонима указывает, является ли подчиненная статья статьей псевдонима (TRUE) или нет (FALSE).

10.1.3.3 КвалификаторЧастичногоРезультата содержит три подкомпонента, которые определены в § 10.1.3.3.1—10.1.3.3.3. Этот параметр должен присутствовать, если результат неполный.

10.1.3.3.1 Параметр ТрудностиОграничения указывает, превышено ли ограничение времени или ограничение длины, или административное ограничение. Возвращаемый результат ограничивается ответами, полученными к моменту достижения ограничения.

10.1.3.3.2 Параметр неисследовано должен присутствовать, если некоторые области ИДС не были исследованы. Этот параметр позволяет АПС продолжить выполнение операции Список обращением к другим пунктам доступа, если он решит предпринять это. Параметр содержит набор (возможно, пустой) СсылокНаПродолжение, каждая из которых содержит имя базового объекта, с которого должна быть продолжена операция, соответствующее значение ПродвижениеОперации и набор пунктов доступа, с которых может быть продолжена операция. Возвращенные СсылкиНаПродолжение должны быть в пределах отсылок, запрошенных параметром службы операций.

10.1.3.3.3 Параметр недоступноеКритическоеРасширение указывает, если присутствует, что одно или более критических расширений были недоступны в некоторой части Справочника.

10.1.3.4 Если в запросе АПС была запрошена защита типа подписанный, то параметр некореллированныйИнфоСписка может содержать некоторое число наборов параметров результата, полученных и подписанных различными компонентами Справочника. Если никакой САС в цепочке не может скореллировать результаты, то АПС должен собрать воедино фактический результат из различных кусков.

10.1.4 Если запрос закончится безуспешно, должна быть сообщена одна из перечисленных ошибок. Обстоятельства, в зависимости от которых будет выдано сообщение о конкретных ошибках, определены в § 12.

10.2 *Поиск*

10.2.1 Операция Поиск используется для поиска в части ИДС статьи, представляющей интерес, и возвращения информации, избранной из этой статьи. Аргументы операции могут быть по выбору подписаны (см. § 7.11) реквестором. Справочник может подписать результат, если это было специфицировано в запросе.

```
Поиск ::= ABSTRACT-OPERATION
    ARGUMENT        АргументПоиска
    RESULT  РезультатПоиска
    ERRORS{
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности}

АргументПоиска ::= OPTIONALLY-SIGNED
SET{
    базовыйОбъект       [0] Имя,
    поднабор            [1]  INTEGER{
        базовыйОбъект (0),
        одинУровень (1),
        целоеПоддерево (2) DEFAULT базовыйОбъект,
    фильтр              [2]  Фильтр DEFAULT and{ }
```

```
        поискПсевдонимов  [3] BOOLEAN DEFAULT TRUE,
        избирание         [4] ИзбираниеИнформацииИзСтатьи DEFAULT { }
            COMPONENTS OF ОбщиеАргументы }

    РезультатПоиска ::=     OPTIONALLY-SIGNED
        CHOICE {
        инфоПоиска SET {
        ВыделенноеИмя OPTIONAL,
        статьи [0]   SET OF ИнформацияИзСтатьи,
        квалификаторЧастичногоРезультата
            [2] КвалификаторЧастичногоРезультата OPTIONAL,
    COMPONENTS OF ОбщиеРезультаты },
    некореллированныйИнфоПоиска [0] SET OF
        РезультатПоиска }
```

10.2.2 В § 10.2.2.1 — 10.2.2.3, § 10.2.2.5 и § 7.3 определяются значения различных аргументов.

10.2.2.1 Аргумент базовыйОбъект идентифицирует статью (или, возможно, корень) объекта, по отношению к которой ведется поиск.

10.2.2.2 Аргумент поднабор указывает, требуется ли применить поиск:

a) только к базовомуОбъекту;

b) только к статьям, непосредственно следующим за базовым объектом (одинУровень);

c) к базовому объекту и всем статьям, следующим за базовым объектом (целоеПоддерево).

10.2.2.3 Аргумент фильтр используется для исключения из области поиска статей, не представляющих интереса. Информация будет возвращена только о статьях, удовлетворяющих фильтру (см. § 7.8).

10.2.2.4 При определении местоположения базового объекта псевдонимы должны быть переименованы; допустимость этого зависит от значения параметраСлужбынеПереименовыватьПсевдонима. Псевдонимы среди статей, следующих за базовым объектом, могут быть переименованы во время поиска; допустимость этого зависит от значения параметра поискПсевдонимов. Если параметр поискПсевдонимов имеет значение TRUE, то псевдонимы должны переименовываться. Если же параметр равен FALSE, то псевдонимы не должны переименовываться.

10.2.2.5 Аргумент избирание идентифицирует запрашиваемую информацию (см. § 7.6).

10.2.3 Запрос завершается успешно, если определено местоположение базового объекта, вне зависимости от того, имеются ли последующие статьи, которые можно было бы возвратить.

Примечание. — Отсюда следует, что результат (нефильтрованного) поиска, примененного к одиночной статье, не идентичен операции Чтения, которая запрашивает опрос того же самого множества атрибутов статьи. Это происходит потому, что Чтение должно возвратить ошибкаАтрибута, если ни одного из избираемых атрибутов в статье не существует.

Параметры результата имеют значение, определенное в § 10.2.3.1 — 10.2.3.4 и § 7.3.

10.2.3.1 ВыделенноеИмя присутствует, если псевдоним был переименован. Оно представляет выделенное имя базового объекта.

10.2.3.2 Параметр статьи передает запрошенную информацию из каждой статьи (нуль или более), удовлетворяющей фильтру (см. § 7.5).

10.2.3.3 КвалификаторЧастичногоРезультата содержит два компонента, описанных для операции Список в §10.1.3.4.

10.2.3.4 Параметр некореллированныйИнфоПоиска совпадает с некореллированнымИнфоСписка в § 10.1.3.4.

10.2.4 Если запрос закончится безуспешно, возвращается одна из перечисленных ошибок. Обстоятельства, в зависимости от которых будут выданы сообщения о конкретных ошибках, определены в § 12.

11 Операции модификации Справочника

Имеется четыре операции модификации Справочника: ДобавлениеСтатьи, УдалениеСтатьи, МодификацияСтатьи и МодификацияОВИ, определяемые в § 11.1—11.4 соответственно.

Примечание 1. — Каждая из этих абстрактных операций определяет статью, являющуюся объектом операции, с помощью ее *выделенного* имени.

Примечание 2. — Успешность выполнения операций ДобавлениеСтатьи, УдалениеСтатьи и Модификация ОВИ будет зависеть от физического распределения ИБС по Справочнику. О безуспешном выполнении будет послано сообщение, содержащее ошибкаОбновления и причинавлияниеКратныхСАС. См. Рекомендацию X. 518.

11.1 *Добавление статьи*

11.1.1 Операция ДобавлениеСтатьи используется для добавления к ИДС статьи, являющейся листом (равно статьи объекта или статьи псевдонима). Аргументы операции могут быть по выбору подписаны (см. § 7.10) реквестором.

```
ДобавлениеСтатьи  ::=      ABSTRACT-OPERATION
    ARGUMENT                    АргументДобавленияСтатьи
    RESULT                  РезультатДобавленияСтатьи
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }

АргументДобавленияСтатьи  ::=  OPTIONALLY-SIGNED
SET {
    объект                  [0]  ВыделенноеИмя,
    статья                  [1]  SET OF Атрибут,
    COMPONENTS OF   ОбщиеАргументы }

РезультатДобавленияСтатьи  ::=  NULL
```

11.1.2 В § 11.1.2.1 — 11.1.2.3 определяются значения различных аргументов.

11.1.2.1 Аргумент объект идентифицирует статью, подлежащую добавлению. Непосредственно предшествующая ей статья, которая должна существовать, чтобы операция могла завершиться успешно, может быть определена путем удаления последнего компонента ОВИ (который принадлежит создаваемой статье).

11.1.2.2 Аргумент статья содержит информацию об атрибуте, которая совместно с такой же из ОВИ образует создаваемую статью. Справочник должен гарантировать, что статья согласуется со схемой Справочника. Если создаваемая статья является статьей псевдонима, то проверка того, что атрибут имяОбъектаПсевдонима указывает на действительную статью, не производится.

11.1.2.3 ОбщиеАргументы (см. § 7.3) включает в себя спецификацию параметров службы, применяемых к запросу. С целями данной операции не согласуются средство по выбору неПереименовыватьПсевдонима и компонент ограничениеДлины и если они присутствуют, то игнорируются. Эта операция никогда не переименовывает псевдонимов.

11.1.3 Если запрос завершится успешно, то возвращается результат, хотя в нем и не должно содержаться никакой информации.

11.1.4 Если запрос завершится безуспешно, то возвращается одна из перечисленных ошибок. Обстоятельства, в зависимости от которых будет выдано сообщение о конкретных ошибках, определяются в § 12.

11.2 *Удаление статьи*

11.2.1 Операция УдалениеСтатьи используется для удаления статьи, являющейся листом (равно статьи объекта или статьи псевдонима) из ИДС. Аргументы операции могут быть по выбору подписаны (см. § 7.10) реквестором.

```
УдалениеСтатьи  ::=  ABSTRACT-OPERATION
    ARGUMENT         АргументУдаленияСтатьи
    RESULT           РезультатУдаленияСтатьи
    ERRORS {
        ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }
```

```
АргументУдаленияСтатьи      ::=  OPTIONALLY-SIGNED SET{
    объект                  [0]      ВыделенноеИмя,
    COMPONENTS OF ОбщиеАргументы }

РезультатУдаленияСтатьи     ::=  NULL
```

11.2.2 В § 11.2.2.1 и § 11.2.2.2 определяются значения различных аргументов.

11.2.2.1 Аргумент объект идентифицирует статью, которая должна быть удалена. Псевдонимы в имени не будут переименовываться.

11.2.2.2 ОбщиеАргументы (см. § 7.3) содержит спецификацию параметров службы, применимых к запросу. С целями данной операции не согласуется вариант неПереименовыватьПсевдонима и компонент ограничение Длины и если они присутствуют, то игнорируются. Эта операция никогда не переименовывает псевдонимов.

11.2.3 Если вопрос завершится успешно, то возвращается результат, хотя он и не содержит информации.

11.2.4 Если запрос завершится безуспешно, то возвращается одна из перечисленных ошибок. Обстоятельства, в зависимости от которых будет выдано сообщение о конкретных ошибках, определяются в § 12.

11.3 *Модификация статьи*

11.3.1 Операция МодификацияСтатьи используется для выполнения серии из одной или более следующих модификаций отдельной статьи:

a) добавления нового атрибута;

b) удаления атрибута;

c) добавления значений атрибута;

d) удаления значений атрибута;

e) замены значений атрибута;

f) модификации псевдонима.

Аргументы операции могут быть по выбору подписаны (см. § 7.11) реквестором.

```
МодификацияСтатьи   ::=  ABSTRACT-OPERATION
    ARGUMENT             АргументМодификацииСтатьи
    RESULT               РезультатМодификацииСтатьи
    ERRORS{
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления}

АргументМодификацииСтатьи  ::=  OPTIONALLY-SIGNED SET{
    объект                 [0]  ВыделенноеИмя,
    изменения              [1]  SEQUENCE OF  МодифицированиеСтатьи,
    COMPONENTS OF  ОбщиеАргументы

РезультатМодификацииСтатьи ::=  NULL

МодифицированиеСтатьи      ::=  CHOICE{
    добавлениеАтрибута     [0]  Атрибут,
    удалениеАтрибута       [1]  ТипАтрибута,
    добавлениеЗначений     [2]  Атрибут,
    удалениеЗначений       [3]  Атрибут}
```

11.3.2 В § 11.3.2.1 — 11.3.2.2 определяются значения различных атрибутов.

11.3.2.1 Аргумент объект идентифицирует статью, к которой должна быть применена модификация. Ни один псевдоним в имени не будет переименован.

11.3.2.2 Аргумент изменения определяет последовательность модификаций, которые должны выполняться в специфицированном порядке. Если какая-то из индивидуальных модификаций завершается безуспешно, то генерируется ОшибкаАтрибута и статья остается в состоянии, в котором она была до операции. Таким образом, операция является атомарной. Окончательный результат последовательности модификаций не должен нарушать схемы Справочника. Однако допустимо, а в некоторых случаях и необходимо, чтобы некоторые изменения, вызванные МодифицированиемСтатьи, производили впечатления, что они делают это. Могут быть выполнены следующие типы модификации:

a) добавлениеАтрибута — здесь идентифицируется новый атрибут, который следует добавить к статье, полностью специфицируемой аргументом. Попытка добавления уже существующего атрибута приводит к ОшибкеАтрибута;

b) удалениеАтрибута — аргумент идентифицирует (своим типом) атрибут, который должен быть удален из статьи. Попытка удаления несуществующего атрибута приводит к ОшибкеАтрибута;

Примечание. — Эта операция недопустима, если тип атрибута представлен в ОВИ.

c) добавлениеЗначений — здесь идентифицируется атрибут посредством типа атрибута в аргументе и специфицируются одно или более значений атрибутов, которые должны быть добавлены к атрибуту. Попытка добавления уже существующего значения приводит к ошибке. Попытка добавления значения к несуществующему типу приводит к ошибке;

d) удалениеЗначений — здесь идентифицируется атрибут посредством типа атрибута в аргументе и специфицируются одно или более значений атрибута, которые должны быть удалены из атрибута. Если значения не представлены в атрибуте, то в результате выдается ОшибкаАтрибута. Если делается попытка произвести модификацию атрибута класса объектов, то возвращается ошибка обновления.

Примечание. — В настоящее время эта операция допустима, если одно из значений представлено в ОВИ.

Значения могут быть заменены комбинацией добавленияЗначений и удаленияЗначений в одной операции МодификацияСтатьи.

11.3.2.3 ОбщиеАргументы (см. § 7.3) содержат спецификацию параметров службы, применимых к запросу. Для целей этой операции вариант неПереименовыватьПсевдонима и компонент ограничениеДлины недопустимы и если они присутствуют, то игнорируются. Эта операция никогда не переименовывает псевдонимов.

11.3.3 Если запрос завершается успешно, то должен быть возвращен результат, хотя он не содержит в себе информации.

11.3.4 Если запрос завершается безуспешно, то передается одно из перечисленных сообщений об ошибках. Обстоятельства, в зависимости от которых будет выдано сообщение о конкретных ошибках, определяются в § 12.

11.4 *Модификация ОВИ*

11.4.1 Операция МодификацияОВИ используется для изменения относительно выделенного имени статьи, являющейся листом (равно статьи объекта или статьи псевдонима) в ИДС. Аргумент операции может быть по выбору подписан (см. § 7.11) реквестором.

```
МодификацияОВИ  ::=     ABSTRACT-OPERATION
    ARGUMENT            АргументМодификацииОВИ
    RESULT              РезультатМодификацииОВИ
    ERRORS {
        ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }

АргументМодификацииОВИ  ::=  OPTIONALLY-SIGNED SET {
    объект                  [0]  ВыделенноеИмя,
    новоеОВИ                [1]  ОтносительноВыделенноеИмя,
    удалениеСтарогоОВИ      [2]  BOOLEAN DEFAULT FALSE,
    COMPONENTS OF  ОбщиеАргументы }

РезультатМодификацииОВИ  ::=  NULL
```

11.4.2 В § 11.4.2.1 — 11.4.2.5 определяются значения различных параметров.

11.4.2.1 Аргумент объект идентифицирует статью, относительно выделенное имя которой должно быть модифицировано. Псевдонимы в имени не будут переименовываться. Непосредственно предшествующая статья не должна иметь неспецифицированных ссылок вниз (см. Рекомендацию X.518).

11.4.2.2 Аргумент новоеОВИ специфицирует новое ОВИ статьи.

11.4.2.3 Если значение атрибута в новом ОВИ не присутствует в статье (или как часть старого ОВИ, или как невыделенное значение), то оно добавляется. Если оно не может быть добавлено, возвращается ошибка.

11.4.2.4 Если флажок удалениеСтарогоОВИ установлен, то все значения атрибутов старого ОВИ, которых нет в новом ОВИ, удаляются. Если этот флажок не установлен, то старые значения должны остаться в статье (но не как часть ОВИ). Флажок следует устанавливать, если атрибут, имеющий в ОВИ единственное значение, изменяет свое значение в результате операции. Если эта операция удаляет последнее значение атрибута некоторого атрибута, то этот атрибут должен быть удален.

11.4.2.5 **ОбщиеАргументы** (см. § 7.3) содержат спецификацию параметров службы, применимых к запросу. С целями данной операции не согласуются вариант **неПереименовыватьПсевдонимов** и компонент **ограничение Длины** и если они присутствуют, то игнорируются. Эта операция не переименовывает псевдонимов.

11.4.3 Если запрос завершается успешно, то возвращается результат, хотя он и не содержит в себе никакой информации.

11.4.4 Если запрос завершается безуспешно, то возвращается одна из перечисленных ошибок. Обстоятельства, в результате которых будет выдано сообщение о конкретных ошибках, определяются в § 12.

11.4.5 Как определено в данной Рекомендации, эта операция может применяться только к статье, являющейся листом.

12 Ошибки

12.1 *Приоритет ошибок*

12.1.1 Справочник прекращает выполнение операции тогда, когда установлено, что должна быть возвращена ошибка.

Примечание 1. — Из этого правила следует, что первая обнаруживаемая ошибка при повторных выполнениях того же опроса может быть различной, поскольку не специфицирован логический порядок реализации заданного опроса. Например, обращения к САС могут выполняться в различном порядке.

Примечание 2. – Правило приоритета ошибок, специфицируемое здесь, применимо только к абстрактной службе, обеспечиваемой Справочником в целом. Если принимать в расчет внутреннюю структуру Справочника, то будут использоваться различные правила.

12.1.2 Если Справочник обнаруживает одновременно несколько ошибок, то нижеследующий список определяет, какая из ошибок будет сообщена. Ошибка, находящаяся в списке выше, имеет более высокий логический приоритет, чем находящаяся ниже, и является той ошибкой, о которой будет сообщено:

a) **ОшибкаИмени**;

b) **ОшибкаОбновления**;

c) **ОшибкаАтрибута**;

d) **ОшибкаБезопасности**;

e) **ОшибкаСлужбы**;

12.1.3 Следующие ошибки не вызывают конфликта в смысле их взаимного приоритета:

a) **НевыполненныйОтказ**, так как эта ошибка специфична для одной операции, **Отказ**, не имеющей других ошибок;

b) **Отказано**, о которой не сообщается, если операция **Отказ** поступила одновременно с обнаружением ошибки. В этом случае ошибка **НевыполненныйОтказ**, сообщающая о причине **слишкомПоздно**, присылается совместно с фактически наступившей ошибкой;

c) Отсылка, которая не является "настоящей" ошибкой, а только указывает на то, что Справочник обнаружил, что АПС должен представить свой запрос в другой точке доступа.

12.2 *Отказано*

12.2.1 Об этом результате может быть сообщено относительно некоторой незавершенной операции опроса Справочника (то есть **Чтения**, **Записи**, **Сравнения**, **Списка**), когда АПС выдает операцию **Отказ** с соответствующим **ИДВызова**.

Отказано ::= **ABSTRACT-ERROR** *– – не является фактически "ошибкой"*

12.2.2 Не существует параметров, связанных с этой ошибкой.

12.3 *Невыполненный отказ*

12.3.1 Ошибка **НевыполненныйОтказ** сообщает о трудностях, возникающих в процессе попыток отказаться от операции.

```
НевыполненныйОтказ ::= ABSTRACT-ERROR
    PARAMETER SET{
        трудность       [0] ТрудностиОтказа,
        операция        [1] ИДВызова}

ТрудностиОтказа ::= INTEGER{
                    нетТакойОперации (1),
                    слишкомПоздно (2),
                    отказНевозможен (3)}
```

12.3.2 В § 12.3.2.1 — 12.3.2.2 определяются значения различных параметров.

12.3.2.1 Специфицируется конкретная встретившаяся трудность. Может быть указана одна из следующих трудностей:

a) нетТакойОперации, когда Справочнику ничего не известно относительно операции, от которой требуется отказаться (это может быть потому, что не было такого запроса или Справочник забыл о нем);

b) слишкомПоздно, когда Справочник уже ответил на операцию;

c) отказНевозможен, когда была сделана попытка отказа от операции, для которой это запрещено (например, модификации), или отказ не может быть выполнен.

12.3.2.2 Идентификация конкретной операции (возбужденной), от которой следует отказаться.

12.4 *Ошибка Атрибута*

12.4.1 ОшибкаАтрибута сообщает о трудностях, связанных с атрибутом.

```
ОшибкаАтрибута ::= ABSTRACT-ERROR
    PARAMETER SET{
        объект [0]    Имя,
        трудности [1] SET OF SEQUENCE{
          трудность  [0]  ТрудностиАтрибута,
          тип        [1]  ТипАтрибута,
          значение   [2]  ЗначениеАтрибута
                                  OPTIONAL}}

ТрудностиАтрибута ::= INTEGER{
    нетТакогоАтрибутаИлиЗначения (1),
    недействительныйСинтаксисАтрибута (2),
    неопределенныйТипАтрибута (3),
    неприемлемоеСопоставление (4),
    нарушениеОграничений (5),
    атрибутИлиЗначениеУжеСуществуют (6)}
```

12.4.2 В § 12.4.2.1 — 12.4.2.2 определяются значения различных параметров.

12.4.2.1 Параметр объект идентифицирует статью, к которой была применена операция, когда появилась ошибка.

12.4.2.2 Могут быть специфицированы одна или более трудностей. Каждая трудность, определенная ниже, сопровождается указанием на тип атрибута, и если необходимо устранить двусмысленность, то указывается значение, которое вызвало трудность:

a) нетТакогоАтрибутаИлиЗначения — указанная статья не содержит одного из атрибутов или значений, специфицированных в качестве аргумента операции;

b) недействительныйСинтаксисАтрибута — потенциальное значение атрибута, специфицированное в качестве аргумента операции, не соответствует синтаксису атрибута данного типа;

c) неопределенныйТипАтрибута — в качестве аргумента операции был использован ранее не определенный тип атрибута. Эта ошибка может появиться только в отношении операций Добавление, Удаление, Модификация или МодификацияОВИ;

d) неприемлемоеСопоставление — была сделана попытка, например в фильтре, использовать правило сопоставления, не определенное для рассматриваемого типа атрибута;

e) нарушениеОграничения — атрибут или значение атрибута, заданные в аргументе абстрактной операции, не удовлетворяет ограничениям, приведенным в Рекомендации X.501 или в определении атрибута (например, значение превышает максимально допустимый размер);

f) **атрибутИлиЗначениеУжеСуществуют** — была сделана попытка добавить атрибут, который уже имеется в статье, или значение, которое уже существует в атрибуте.

12.5 *Ошибка Имени*

12.5.1 **ОшибкаИмени** сообщает о трудности, связанной с именем, заданным как аргумент операции.

```
ОшибкаИмени ::= ABSTRACT-ERROR
    PARAMETER SET{
        трудность [0] ТрудностиИмени,
        сопоставим [1] Имя}

ТрудностиИмени ::= INTEGER{
    нетТакогоОбъекта (1),
    трудностиПсевдонима (2),
    недействительныйСинтаксисАтрибута (3),
    трудностьПереименованияПсевдонима (4)}
```

12.5.2 В § 12.5.2.1 и § 12.5.2.2 определяются значения различных параметров.

12.5.2.1 Встретилась конкретная трудность. Может быть указана любая из нижеследующих трудностей:

a) **нетТакогоОбъекта** — представленное имя не соответствует имени какого-либо объекта;

b) **трудностиПсевдонима** — был переименован псевдоним, который не именует никакого объекта;

c) **недействительныйСинтаксисАтрибута** — тип атрибута и сопутствующие ему значения атрибута, указанные в ПЗА, в имени несовместимы;

d) **трудностьПереименованияПсевдонима** — был использован псевдоним в ситуации, когда это недопустимо.

12.5.2.2 Параметр сопоставим содержит имя низшей статьи (объекта или псевдонима) в ИДС, с которой сопоставление осуществилось. Имя этой статьи либо статьи, получившейся в результате переименования, если таковое имело место, является усеченной формой предоставленного имени.

Примечание. — Если имеются трудности с типом атрибута и/или значением в том имени, которое было представлено в аргументе операции Справочника, об этом сообщается посредством **ОшибкаИмени** (с трудностью **недействительныйСинтаксисАтрибута**), а не посредством **ОшибкаАтрибута** или **ОшибкаОбновления**.

12.6 *Отсылка*

12.6.1 Отсылка перенаправляет пользователя-службы в один или более пунктов доступа, лучше оборудованных для выполнения операции.

```
Отсылка ::= ABSTRACT-ERROR -- не является фактической "ошибкой"
    PARAMETER SET{
    кандидат          [0] ССылкаНаПродолжение }
```

12.6.2 Ошибка содержит всего один параметр, содержащий **ССылкуНаПродолжение**, которая может быть использована для продвижения операции (см. Рекомендацию X.518).

12.7 *Ошибка Безопасности*

12.7.1 **ОшибкаБезопасности** сообщает о трудностях в выполнении операции, связанных с соображениями безопасности.

```
ОшибкаБезопасности ::= ABSTRACT-ERROR
    PARAMETER SET{
        проблема [0] ТрудностиБезопасности }

ТрудностиБезопасности ::= INTEGER{
    неприемлемаяАутентификация (1),
    недействительноеУдостоверение (2),
    недостаточныеПраваДоступа (3),
    недействительнаяПодпись (4),
    требуетсяЗащита (5),
    нетИнформации (6)}
```

12.7.2 Ошибка имеет единственный параметр, который сообщает о встретившейся конкретной трудности. Могут быть указаны следующие трудности:

a) **неприемлемаяАутентификация** — уровень безопасности, связанный с удостоверением реквестора, не удовлетворяет уровню запрашиваемой защиты; например, было предоставлено простое удостоверение, хотя требовалось строгое;

b) недействительноеУдостоверение — представленное удостоверение было недействительным;

c) недостаточныеПраваДоступа — реквестор не имеет права выполнять запрошенную операцию;

d) недействительнаяПодпись — обнаружено, что подпись под запросом недействительна;

e) требуетсяЗащита — Справочник не захотел выполнять запрошенную операцию, так как аргумент не был подписан;

f) нетИнформации — запрошенная операция привела к ошибке, для которой нет доступной информации.

12.8 *Ошибка Службы*

12.8.1 ОшибкаСлужбы сообщает о трудностях, связанных с обеспечением службы.

```
ОшибкаСлужбы ::= ABSTRACT-ERROR
    PARAMETER SET {
    трудность [0] ТрудностиСлужбы }

ТрудностиСлужбы ::= INTEGER {
        занят (1),
        недоступен (2),
        нерасположенВыполнить (3),
        требуетсяСцепление (4),
        продвижениеНевозможно (5),
        недействительнаяСсылка (6),
        ограничениеВремениПревышено (7),
        административноеОграничениеПревышено (8),
        обнаруженаПетля (9),
        недоступноеКритическоеРасширение (10),
        внеОбластиПрименения (11),
        ошибкаИдс (12) }
```

12.8.2 Ошибка имеет единственный параметр, который сообщает о конкретной встретившейся трудности. Могут быть указаны следующие трудности:

a) занят — Справочник (или некоторая его часть) в настоящее время занят, чтобы выполнять запрошенную операцию, но возможно выполнит ее некоторое время спустя;

b) недоступен — Справочник (или некоторая его часть) в настоящее время недоступен;

c) нерасположенВыполнить — Справочник (или некоторая его часть) не готов к выполнению этого запроса; например потому, что это приведет к неумеренному использованию ресурсов или к нарушению порядка, установленного Административным органом;

d) требуетсяСцепление — Справочник не может удовлетворить запрос иначе, чем с помощью сцепления, однако сцепление было запрещено посредством варианта параметра службы, равного сцепление Запрещено;

e) продвижениеНевозможно — САС, возвращающий эту ошибку, не имеет административных прав над соответствующим именующим контекстом и, как следствие, не мог принять участия в разрешении имени;

f) недействительнаяСсылка — САС не мог выполнить запрос, как он предписан со стороны АПС (в ПродвижениеОперации). Это могло возникнуть из-за использования недействительной отсылки;

g) ограничениеВремениПревышено — Справочник исчерпал выделенное ему время, установленное пользователем с помощью параметра службы; нет доступного частичного результата, который мог бы быть возвращен пользователю;

h) административноеОграничениеПревышено — Справочник достиг некоторого ограничения, установленного административным органом, и нет доступного частичного результата, который мог бы быть возвращен пользователю;

i) обнаруженаПетля — Справочник не в состоянии завершить запрос из-за внутреннего цикла;

j) недоступноеКритическоеРасширение — Справочник не смог выполнить запроса, так как одно или более критических расширений оказались недоступны;

k) **внеОбластиПрименения** — никакое переименование не было доступно в пределах запрошенной области применения;

l) **ошибкаИдс** — Справочник не в состоянии завершить запрос из-за трудностей с цельностью ИДС.

12.9 *Ошибка Обновления*

12.9.1 **ОшибкаОбновления** сообщает о трудностях, связанных с попытками добавить, удалить или модифицировать информацию в ИБС.

```
ОшибкаОбновления ::= ABSTRACT-ERROR
        PARAMETER SET {
            трудность  [0]  ТрудностиОбновления }

ТрудностиОбновления  ::= INTEGER {
        нарушениеИменования (1),
        нарушениеКлассаОбъектов (2),
        недопустимДляНеЛиста (3),
        недопустимДляОВИ (4),
        статьяУжеСуществует (5),
        затрагиваетНесколькоCAC (6),
        запрещенаМодификацияКлассаОбъектов (7)}
```

12.9.2 Ошибка имеет единственный параметр **трудность**, который сообщает о конкретной встретившейся трудности. Могут быть указаны следующие трудности:

a) **нарушениеИменования** — попытка добавления или модификации приведет к нарушению структурных правил ИДС, как они определены схемой Справочника в Рекомендации X.501. Другими словами, статья будет помещена как последующая за статьей псевдонима или в область ИДС, недопустимую для членов ее класса, или будет определено такое ОВИ статьи, при котором в ОВИ надо будет включить запрещенный тип атрибута;

b) **нарушениеКлассаОбъектов** — попытка обновления приведет к созданию статьи, несовместной с определением, предусмотренным для ее класса объектов или с определениями Рекомендации X.501, касающимися свойств классов объектов;

c) **неДопустимДляНеЛиста** — была сделана попытка выполнить операцию, допустимую только в отношении статей, являющихся листьями в ИДС;

d) **неДопустимДляОВИ** — была сделана попытка выполнения операции, воздействующей на ОВИ (например, удаления атрибута, который является частью ОВИ);

e) **статьяУже Существует** — предпринята попытка выполнить операцию **ДобавитьСтатью**, именующую уже существующую статью;

f) **затрагиваетНесколькоCAC** — попытка обновления потребует работы с несколькими CAC, что запрещено;

g) **запрещенаМодификацияКлассаОбъекта** — попытка выполнить операцию модификации атрибута "класс объектов".

Примечание. — **ОшибкаОбновления** не используется для сообщения о трудностях, связанных с типами атрибутов, значениями или ограничениями, встретившихся при выполнении операции **ДобавлениеСтатьи, Удаление Статьи, МодификацияСтатьи** и **МодификацияОВИ**. Об этих трудностях сообщается посредством **ошибкиАтрибута**.

ПРИЛОЖЕНИЕ A

(к Рекомендации X.511)

Абстрактные службы на НАС.1

Данное Приложение является составной частью стандарта.

Данное Приложение содержит НАС.1-модуль **АбстрактнаяСлужбаСправочника**, в который включены все НАС.1-определения типов, значений и макросов, введенные в настоящей Рекомендации.

```
АбстрактнаяСлужбаСправочника { joint-ISO-CCITT ds(5) modules(1) directoryAbstractService(2)}
DEFINITIONS ::=
BEGIN

EXPORTS
    справочник, портЧтения, портПоиска, портМодификации,
    ПривязываниеКСправочнику, АргументПривязыванияКСправочнику,
    ОтвязываниеОтСправочника,
    Чтение, АргументЧтения, РезультатЧтения,
    Отказ, АргументОтказа, РезультатОтказа,
    Сравнение, АргументСравнения, РезультатСравнения,
    Список, АргументСписка, РезультатСписка,
    Поиск, АргументПоиска, РезультатПоиска,
    ДобавлениеСтатьи, АргументДобавленияСтатьи, РезультатДобавленияСтатьи,
    УдалениеСтатьи, АргументУдаленияСтатьи, РезультатУдаленияСтатьи,
    МодификацияСтатьи, АргументМодификацииСтатьи, РезультатМодификацииСтатьи,
    МодификацияОВИ, АргументМодификацииОВИ, РезультатМодификацииОВИ,
    Отказано, НевыполненныйОтказ, ОшибкаАтрибута, ОшибкаИмени,
    Отсылка, ОшибкаБезопасности, ОшибкаСлужбы, ОшибкаОбновления,
    ПараметрыБезопасности;

IMPORTS
    СтруктураИнформации, структураАутентификации,
        распределенныеОперации, идентификаторыОбъектовСправочника
        FROM ПолезныеОпределения { joint-iso-ccitt ds(5) modules(1)
                                   usefulDefinitions(0) }

OBJECT, PORT, ABSTRACT-BIND, ABSTRACT-UNBIND,
ABSTRACT-OPERATION, ABSTRACT-ERROR
        FROM НотацияАбстрактнойСлужбы { joint-iso-ccitt mhs-motis(6)
                                        asdc(2) modules(0) notation(1)}

Атрибут, ТипАтрибута, ЗначениеАтрибута, ПроверкаЗначенияАтрибута,
ВыделенноеИмя, Имя, ОтносительноВыделенноеИмя
        FROM СтруктураИнформации СтруктураИнформации

ид-от-справочника, ид-от-апс, ид-пт-чтения, ид-пт-поиска, ид-пт-модификации
        FROM ИдентификаторыОбъектовСправочника ИдентификаторыОбъектовСправочника

СсылкаНаПродолжения, ПродвижениеОперации
        FROM РаспределенныеОперации  РаспределенныеОперации

Сертификат, ВетвьСертификации, SIGNED,
PROTECTED, ИдентификаторАлгоритма
        FROM СтруктураАутентификации СтруктураАутентификации

ИДВызова
        FROM Нотация-Удаленных-Операций { joint-iso-ccitt
                                          remoteOperations(4) notation(0) } ;


-- макрос для представления необязательного подписывания --


OPTIONALLY-SIGNED MACRO ::=
BEGIN
    TYPE NOTATION   ::= type (Type)
    VALUE NOTATION  ::= value (VALUE CHOICE { Type, SIGNED Type } )

END


-- объекты и порты --


справочник
    OBJECT
        PORTS{портЧтения [S],
              портПоиска [S],
              портМодификации [S]}

::= ид-от-справочника
```

```
апс

    OBJECT
        PORTS { портЧтения [C],
                портПоиска [C],
                портМодификации [C]}

::= ид-от-апс

портЧтения
    PORT
        CONSUMER INVOKES {
                Чтение, Сравнение, Отказ}

::=  ид-пт-чтения

портПоиска
    PORT
        CONSUMER INVOKES{
                Список, Поиск }

::= ид-пт-поиска

портМодификации
    PORT
        CONSUMER INVOKES{
                ДобавлениеСтатьи, УдалениеСтатьи,
                МодификацияСтатьи, МодификацияОВИ }

::= ид-пт-модификации


-- привязывание и отвязывание --

ПривязываниеКСправочнику  ::=  ABSTRACT-BIND
    TO { портЧтения, портПоиска, портМодификации }
    BIND
    ARGUMENT   АргументПривязыванияКСправочнику
    RESULT     РезультатПривязыванияКСправочнику
    BIND-ERROR ОшибкаПривязыванияКСправочнику

АргументПривязыванияКСправочнику ::= SET{
    удостоверение [0] Удостоверение OPTIONAL
    версии        [1] Версии DEFAULT   v1988}

Удостоверение  ::= CHOICE{
    простое  [0] ПростоеУдостоверение,
    строгое  [1] СтрогоеУдостоверение,
    внешняяПроцедура [2] EXTERNAL}

ПростоеУдостоверение  ::= SEQUENCE{
    имя  [0] ВыделенноеИмя,
    действительность [1] SET {
       время1             [0]  ВремяОтГринвича OPTIONAL,
       время2             [1]  ВремяОт Гринвича OPTIONAL,
       случайноеЧисло1    [2]  BIT STRING OPTIONAL,
       случайноеЧисло2    [3]  BIT STRING OPTIONAL}
                                   OPTIONAL,
       пароль             [2]  OCTET STRING OPTIONAL}

СтрогоеУдостоверение  ::= SET{
    ветвьСертификации       [0]  ВетвьСертификации OPTIONAL,
    мандат-на-привязывание  [1]  Мандат}

Мандат  ::= SIGNED SEQUENCE{
    алгоритм        [0]  ИдентификаторАлгоритма,
    имя             [1]  ВыделенноеИмя,
    время           [2]  ВремяОтГринвича,
    случайноеЧисло  [3]  BIT STRING}

Версии  ::=  BIT STRING   {v1988(0)}

РезультатПривязыванияКСправочнику  ::= АргументПривязыванияКСправочнику
```

```
ОшибкаПривязыванияКСправочнику ::= SET {
    версии [0] Версии DEFAULT v1988,
    CHOICE{
        ошибкаСлужбы        [1] ТрудностиСлужбы,
        ошибкаБезопасности  [2] ТрудностиБезопасности}}

ОтвязываниеОТСправочника ::= ABSTRACT-UNBIND
    FROM { портЧтения, портПоиска, портМодификации }


-- операции, аргументы и результаты --


Чтение ::= ABSTRACT-OPERATION
    ARGUMENT АргументЧтения
    RESULT   РезультатЧтения
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности }

АргументЧтения ::= OPTIONALLY-SIGNED SET{
    объект     [0] Имя,
    избирание  [1] ИзбираниеИнформацииИзСтатьи
                       DEFAULT { },
    COMPONENTS OF ОбщиеАргументы }

РезультатЧтения ::= OPTIONALLY-SIGNED SET {
    статья [0] ИнформацияИзСтатьи
    COMPONENTS OF ОбщиеРезультаты }

Сравнение ::= ABSTRACT-OPERATION
    ARGUMENT АргументСравнения
    RESULT   РезультатСравнения
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности}

АргументСравнения ::= OPTIONALLY-SIGNED SET {
    объект          [0] Имя,
    потенциальный [1] ПроверкаЗначенияАтрибута,
    COMPONENTS OF   ОбщиеАргументы }

РезультатСравнения ::= OPTIONALLY-SIGNED SET {
    ВыделенноеИмя OPTIONAL,
    сопоставим [0] BOOLEAN,
    изСтатьи   [1] BOOLEAN DEFAULT TRUE,
    COMPONENTS OF ОбщиеРезультаты }

Отказ ::= ABSTRACT-OPERATION
    ARGUMENT АргументОтказа
    RESULT РезультатОтказа
    ERRORS {НевыполненныйОтказ }

АргументОтказа ::= SEQUENCE {
    идВызова [0] ИДВызова }

РезультатОтказа ::= NULL

Список ::= ABSTRACT-OPERATION
    ARGUMENT АргументСписка
    RESULT   РезультатСписка
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности }

АргументСписка ::= OPTIONALLY-SIGNED SET {
    объект [0] Имя,
    COMPONENTS OF ОбщиеАргументы }
```

```
РезультатСписка ::= OPTIONALLY-SIGNED CHOICE {
    инфоСписка SET {
    ВыделенноеИмя OPTIONAL
    подчиненные [1] SET OF SEQUENCE {
        ОтносительноВыделенноеИмя,
        статья Псевдонима  [0] BOOLEAN DEFAULT FALSE,
        изСтатьи           [1] BOOLEAN DEFAULT TRUE},
        квалификаторЧастичногоРезультата [2] КвалификаторЧастичногоРезультата
                                             OPTIONAL,
        COMPONENTS OF ОбщиеРезультаты },
    некореллированныйИнфоСписка [0] SET OF
        РезультатСписка }

КвалификаторЧастичногоРезультата ::= SET {
    трудностиОграничения    [0] ТрудностиОграничения OPTIONAL,
    неисследовано           [1] SET OF
        ССылкаНаПродолжение OPTIONAL,
    недоступноеКритическоеРасширение [2] BOOLEAN DEFAULT FALSE }

ТрудностиОграничения ::= INTEGER {
    превышеноОграничениеВремени (0),
    превышеноОграничениеДлины (1),
    превышеноАдминистративноеОграничение (2) }

Поиск ::= ABSTRACT-OPERATION
    ARGUMENT АргументПоиска
    RESULT   РезультатПоиска
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, Отказано,
        ОшибкаБезопасности}

АргументПоиска ::= OPTIONALLY-SIGNED SET {
    базовыйОбъект [0] Имя,
    поднадбор     [1] INTEGER {
        базовыйОбъект(0),
        одинУровень(1),
        целоеПоддерево(2)} DEFAULT базовыйОбъект,
    фильтр              [2] Фильтр DEFAULT and{},
    поискПсевдонимов    [3] BOOLEAN DEFAULT TRUE,
    избирание           [4] ИзбираниеИнформацииИзСтатьи DEFAULT{},
    COMPONENTS OF       ОбщиеАргументы }

РезультатПоиска ::= OPTIONALLY-SIGNED
    CHOICE {
    инфоПоиска SET {
    ВыделенноеИмя OPTIONAL,
    статьи [0] SET OF ИнформацияИзСтатьи,
    квалификаторЧастичногоРезультата
      [2] КвалификаторЧастичногоРезультата OPTIONAL,
    COMPONENTS OF ОбщиеРезультаты},
    некореллированныйИнфоПоиска [0] SET OF
        РезультатПоиска}

ДобавлениеСтатьи ::= ABSTRACT-OPERATION
    ARGUMENT АргументДобавленияСтатьи
    RESULT   РезультатДобавленияСтатьи
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }

АргументДобавленияСтатьи ::= OPTIONALLY-SIGNED SET {
    объект [0] ВыделенноеИмя,
    статья [1] SET OF Атрибут,
    COMPONENTS OF ОбщиеАргументы }

РезультатДобавленияСтатьи ::= NULL
```

```
УдалениеСтатьи ::= ABSTRACT-OPERATION
    ARGUMENT АргументУдаленияСтатьи
    RESULT   РезультатУдаленияСтатьи
    ERRORS {
        ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }

АргументУдаленияСтатьи ::= OPTIONALLY-SIGNED SET {
        объект [0] ВыделенноеИмя,
        COMPONENTS OF ОбщиеАргументы}

РезультатУдаленияСтатьи ::= NULL

МодификацияСтатьи ::= ABSTRACT-OPERATION
    ARGUMENT АргументМодификацииСтатьи
    RESULT   РезультатМодификацииСтатьи
    ERRORS {
        ОшибкаАтрибута, ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }

АргументМодификацииСтатьи ::= OPTIONALLY-SIGNED SET {
    объект      [0] ВыделенноеИмя,
    изменения   [1] SEQUENCE OF МодифицированиеСтатьи,
    COMPONENTS OF ОбщиеАргументы }

РезультатМодификацииСтатьи ::= NULL

МодифицированиеСтатьи ::= CHOICE {
    добавлениеАтрибута [0] Атрибут,
    удалениеАтрибута   [1] ТипАтрибута,
    добавлениеЗначений [2] Атрибут,
    удалениеЗначений   [3] Атрибут }

МодификацияОВИ ::= ABSTRACT-OPERATION
    ARGUMENT АргументМодификацииОВИ
    RESULT   РезультатМодификацииОВИ
    ERRORS {
        ОшибкаИмени,
        ОшибкаСлужбы, Отсылка, ОшибкаБезопасности,
        ОшибкаОбновления }

АргументМодификацииОВИ ::= OPTIONALLY-SIGNED SET {
    объект                  [0] Выделенное Имя,
    новоеОВИ                [1] ОтносительноВыделенноеИмя,
    удалениеСтарого ОВИ     [2] BOOLEAN DEFAULT FALSE,
    COMPONENTS OF ОбщиеАргументы }

РезультатМодификацииОВИ ::= NULL


-- ошибки и параметры --


Отказано ::= ABSTRACT-ERROR -- не является фактически "ошибкой"

НевыполненныйОтказ ::= ABSTRACT-ERROR
    PARAMETER SET {
        трудность [0] ТрудностиОтказа,
        операция  [1] ИДВызова }

ТрудностиОтказа ::= INTEGER {
        нетТакойОперации (1),
        слишкомПоздно (2),
        отказНевозможен (3) }
```

```
ОшибкаАтрибута ::= ABSTRACT-ERROR
    PARAMETER SET {
        объект      [0] Имя,
        трудности   [1] SET OF SEQUENCE {
          трудность [0] ТрудностиАтрибута,
          тип       [1] ТипАтрибута,
          значение  [2] ЗначениеАтрибута OPTIONAL}}

ТрудностиАтрибута ::=
    INTEGER {
        нетТакогоАтрибутаИлиЗначения (1),
        недействительныйСинтаксисАтрибута (2),
        неопределенныйТипАтрибута (3),
        неприемлемоеСопоставление (4),
        нарушениеОграничений (5),
        атрибутИлиЗначениеУжеСуществуют (6) }

ОшибкаИмени ::= ABSTRACT-ERROR
    PARAMETER SET{
        трудность   [0] ТрудностиИмени,
        сопоставим  [1] Имя }

ТрудностиИмени ::= INTEGER {
        нетТакогоОбъекта (1),
        трудностьПсевдонима (2),
        недействительныйСинтаксисАтрибута (3),
        трудностьПереименованияПсевдонима (4) }

Отсылка ::= ABSTRACT-ERROR -- не является фактически "ошибкой"
    PARAMETER SET {
        кандидат [0] ССылкаНаПродолжение }

ОшибкаБезопасности ::= ABSTRACT-ERROR
    PARAMETER SET {
        трудность [0] ТрудностиБезопасности }

ТрудностиБезопасности ::= INTEGER {
    неприемлемаяАутентификация (1),
    недействительноеУдостоверение (2),
    недостаточныеПраваДоступа (3),
    недействительнаяПодпись (4),
    требуетсяЗащита (5),
    нетИнформации (6)}

ОшибкаСлужбы ::= ABSTRACT-ERROR
    PARAMETER SET{
        трудность [0] ТрудностиСлужбы }

ТрудностиСлужбы ::= INTEGER {
    занят (1),
    недоступен (2),
    нерасположенВыполнить (3),
    требуетсяСцепление (4),
    продвижениеНевозможно (5),
    недействительнаяСсылка (6),
    ограничениеВременипревышено (7),
    административноеОграничениеПревышено (8),
    обнаруженаПетля (9),
    недоступноеКритическоеРасширение (10),
    внеОбластиПрименения (11),
    ошибкаИДС (12)}

ОшибкаОбновления ::= ABSTRACT-ERROR
    PARAMETER SET{
        трудность [0] ТрудностиОбновления }
```

```
ТрудностиОбновления ::= INTEGER {
    нарушениеИменования (1),
    нарушениеКлассаОбъектов (2),
    недопустимДляНеЛиста (3),
    недопустимДляОВИ (4),
    статьяУжеСуществует (5),
    затрагиваетНесколькоСАС (6),
    запрещенаМодификацияКлассаОбъектов (7)}

-- общие аргументы/результаты --

ОбщиеАргументы ::= SET{
    [30] ПараметрыСлужбы DEFAULT {}
    [29] ПараметрыБезопасности DEFAULT {},
    реквестор [28] ВыделенноеИмя OPTIONAL,
    [27] ПродвижениеОперации DEFAULT неНачата,
    оВИОбъектаПсевдонима [26] INTEGER OPTIONAL,
    расширения [25] SET OF Расширение OPTIONAL }

Расширение ::= SET{
    идентификатор  [0] INTEGER,
    критический    [1] BOOLEAN DEFAULT  FALSE,
    элемент        [2] ANY DEFINED BY идентификатор}

ОбщиеРезультаты ::= SET {
    [30] ПараметрыБезопасности OPTIONAL,
    исполнитель [29] ВыделенноеИмя OPTIONAL,
    псевдонимПереименован [28] BOOLEAN DEFAULT FALSE}

-- общие типы данных --

ПараметрыСлужбы ::= SET {
    варианты [0] BIT STRING {
    предпочтительноСцепление (0),
    запрещеноСцепление (1),
    локальнаяОбластьПрименения (2),
    неИспользоватьКопий (3),
    неПереименовыватьПсевдонимов (4)}
            DEFAULT {},

    приоритет [1] INTEGER {
        низкий (0),
        средний (1),
        высокий (2) } DEFAULT средний,

ограничениеВремени  [2] INTEGER OPTIONAL,

ограничениеДлины    [3] INTEGER OPTIONAL,

областьПримененияОтсылок   [4] INTEGER {
        оус (0),
        страна (1)}
        OPTIONAL }

ИзбираниеИнформацииИзСтатьи ::= SET{
    типыАтрибутов
        CHOICE{
           всеАтрибуты [0] NULL,
           избрать       [1] SET OF ТипАтрибута
           -- пустое множество означает, что никакие
           -- атрибуты не запрошены -- }
        DEFAULT всеАтрибуты NULL,

    инфоТипы [2] INTEGER{
        толькоТипыАтрибутов (0),
        типыИЗначенияАтрибутов (1)}DEFAULT
           типыИЗначенияАтрибутов  }
```

```
ИнформацияИзСтатьи ::= SEQUENCE {
    ВыделенноеИмя,
    изСтатьи BOOLEAN DEFAULT TRUE,
    SET OF CHOICE {
        ТипАтрибута,
        Атрибут} OPTIONAL}

Фильтр ::= CHOICE{
    элемент     [0] ЭлементФильтра,
    and         [1] SET OF Фильтр,
    or          [2] SET OF Фильтр,
    not         [3] Фильтр }
ЭлементФильтра ::= CHOICE {
    равенство     [0] ПроверкаЗначенияАтрибута,
    подцепочки    [1] SEQUENCE {
        тип ТипАтрибута,
        цепочки SEQUENCE OF CHOICE {
          начальный  [0] ЗначениеАтрибута,
          любой      [1] ЗначениеАтрибута,
          конечный   [2] ЗначениеАтрибута }},
    большеИлиРавно              [2] ПроверкаЗначенияАтрибута,
      меньшеИлиРавно            [3] ПроверкаЗначенияАтрибута,
      присутствует              [4] ТипАтрибута,
    приблизительноеСовпадение   [5] ПроверкаЗначенияАтрибута}

ПараметрыБезопасности ::= SET{
    ветвьСертификации  [0] ВетвьСертификации OPTIONAL,
    имя                [1] ВыделенноеИмя OPTIONAL,
    время              [2] ВремяОтГринвича OPTIONAL,
    случайноеЧисло     [3] BIT STRING OPTIONAL,
    цель               [4] ЗапросЗащиты OPTIONAL }
ЗапросЗащиты ::= INTEGER {
                                отсутствует (0),
                                подписанный (1) }
```

ПРИЛОЖЕНИЕ B

(к Рекомендации X.511)

Идентификаторы объектов Справочника

Данное Приложение является составной частью стандарта.

Данное Приложение содержит HAC.1-модуль ИдентификаторыОбъектовСправочника, в который включены все HAC.1-идентификаторы объектов, введенные в настоящей Рекомендации.

```
ИдентификаторыОбъектовСправочника   { joint-ISO-CCITT ds(5) modules(1)
                                      directoryObjectIdentifiers(9)}

DEFINITIONS ::=
BEGIN

EXPORTS
    ид-от-справочника, ид-от-апс, ид-пт-чтения, ид-пт-поиска, ид-пт-модификации;

IMPORTS
    ид-от, ид-пт

FROM ПолезныеОпределения   {joint-iso-ccitt ds(5) modules(1),
                            usefulDefinitions(0)} ;
```

```
-- Объекты --

ид-от-справочника       OBJECT IDENTIFIER   ::=  { ид-от 1 }

ид-от-апс               OBJECT IDENTIFIER   ::=  { ид-от 2 }

-- Типы Портов --

ид-пт-чтения            OBJECT IDENTIFIER   ::=  { ид-пт 1 }

ид-пт-поиска            OBJECT IDENTIFIER   ::=  { ид-пт 2 }

ид-пт-модификации       OBJECT IDENTIFIER   ::=  { ид-пт 3 }

END
```


СПРАВОЧНИК — ПРОЦЕДУРЫ РАСПРЕДЕЛЕННЫХ ОПЕРАЦИЙ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

[1)] Рекомендация X.518 и ISO 9594-4 "Системы обработки информации – Взаимосвязь открытых систем–Справочник – Процедуры распределенных операций" были разработаны в тесном сотрудничестве и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой ими справочной информацией может рассматриваться как объединенное целое, называемое *Справочником.* Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные процессы, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых различными производителями;

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 В настоящей Рекомендации специфицируются процедуры, с помощью которых распределенные компоненты Справочника взаимодействуют между собой в целях предоставления пользователям Справочника устойчивого обслуживания.

1 Предмет рассмотрения и область применения

1.1 В настоящей Рекомендации специфицируется поведение САС, участвующих в применениях распределенного Справочника. Допустимое поведение было спроектировано таким образом, чтобы обеспечить устойчивое обслуживание при заданном широком распределении ИБС среди большого числа САС.

1.2 Справочник не рассчитан быть общецелевой системой базы данных, хотя он и может быть создан на основе таких систем. Предполагается, в частности, что, как это обычно имеет место со справочными системами связи, обращение за справками к Справочнику будет происходить значительно чаще, чем обновление его данных.

2 Библиография

Рекомендация X.200 "Взаимосвязь открытых систем — Основная эталонная модель".

Рекомендация X.208 "Взаимосвязь открытых систем — Спецификация нотации абстрактного синтаксиса (НАС.1)".

Рекомендация X.500 "Справочник — Обзор концепций, моделей и услуг".

Рекомендации X.501 "Справочник — Модели".

Рекомендация X.511 "Справочник — Определение абстрактной службы".

Рекомендация X.519 "Справочник — Спецификация протоколов".

Рекомендация X.520 "Справочник — Избранные типы атрибутов".

Рекомендация X.521 "Справочник — Избранные классы объектов".

Рекомендация X.407 "Системы обработки сообщений: Соглашения по определению абстрактных служб".

3 Определения

Определения, содержащиеся в этом параграфе, используют сокращения, определенные в § 4.

3.1 *Определения эталонной модели ВОС*

В настоящей Рекомендации используется следующий термин, определенный в X.200:

а) *титул прикладного элемента.*

3.2 *Основные определения Справочника*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.500:

a) *Справочник;*

b) *информационная база Справочника.*

3.2 *Определения модели Справочника*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.501:

a) *точка доступа;*

b) *псевдоним;*

c) *выделенное имя;*

d) *информационное дерево Справочника;*

e) *системный агент Справочника;*

f) *агент пользователя Справочника;*

g) *относительно выделенное имя.*

3.4 *Соглашения по абстрактному определению служб*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.407:

a) *абстрактная ошибка;*

b) *обстрактная операция;*

c) *результат.*

3.5 *Определения распределенных операций*

В настоящей Рекомендации используются определяемые ниже термины:

a) *сцепление* — режим взаимодействия, используемый, но не обязательно, САС в случае, если он не может самостоятельно выполнить операцию. САС осуществляет сцепление, возбуждая операцию у другого САС, передавая в последующем ответ исходному реквестору;

b) *префиксальный контекст* — последовательность ОВИ ведущих от Корня ИДС к начальной вершине именующего контекста; соответствует выделенному имени вершины;

c) *перекрестная ссылка* — ссылочные знания, содержащие информацию о том САС, который хранит статью. Используется для оптимизации. Статья может не иметь соотношений ни с предшественниками, ни с подчиненными;

d) *фрагмент ИБС* — часть ИБС, хранящаяся одним САС и содержащая один или более именующих контекстов;

e) *распределенное разрешение имени* — процесс, с помощью которого разрешение имени осуществляется более чем в одном САС;

f) *внутренняя ссылка* — ссылочные знания, содержащие внутренний указатель на статью, хранящуюся у того же САС;

g) *информационные знания* — информация, имеющаяся у некоторого САС о хранящихся у него статьях, а также о том, как локализовать прочие статьи справочника;

h) *ссылочные знания* — знания, ассоциирующие, непосредственно или косвенно, статью ИБС с тем САС, у которого он размещен;

i) *дерево знаний* – концептуальная модель информационных знаний; эту модель хранит САС, для осуществления распределенного разрешения имени;

j) *многоадресная рассылка* — режим взаимодействия, альтернативно используемый САС, когда он не может сам выполнить операцию; САС *рассылает* операцию, то есть возбуждает ту же самую операцию у нескольких других САС (параллельно или последовательно) и передает соответствующий ответ исходному реквестору;

k) *разрешение имени* — процесс локализации статьи за счет последовательного сопоставления каждого ОВИ в потенциальном имени статьи некоторой вершине ИДС;

l) *именующий контекст* — частичное поддерево ИДС, начинающееся от некоторой вершины и простирающееся вниз к вершинам, являющимся и/или не являющимся листьями. Указанные вершины образуют границы именующего контекста. Граничные вершины, не являющиеся листьями, обозначают начало последующих именующих контекстов;

m) *неспецифицированная ссылка вниз* — ссылочное значение, содержащее информацию о САС, хранящем одну или более неспецифицированных подчиненных вершин;

n) *продвижение операции* — совокупность значений, обозначающих как далеко продвинулось разрешение имени;

o) *ветвь ссылок* — непрерывная последовательность ссылочных знаний;

p) *отсылка* — ответ, который может быть возвращен некоторым САС, когда он не может сам выполнить операцию; этот ответ идентифицирует одного или несколько других САС, более способных выполнить эту операцию;

q) *разложение запроса* — разложение запроса на подзапросы, каждый из которых осуществляет часть распределенной операции;

r) *корневой контекст* — именующий контекст вершины, имя которой состоит из пустого множества ОВИ;

s) *ссылка вниз* — ссылочные знания, содержащие информацию о том САС, который хранит специфицированную подчиненную статью;

t) *подзапрос* — запрос, возникший в результате разложения запроса;

u) *ссылка вверх* — ссылочные знания, содержащие информацию о том САС, который хранит предшествующую статью.

4 Сокращения

В настоящей Рекомендации используются следующие сокращения:

ИБС — информационная база Справочника

ИДС — информационное дерево Справочника

САС — системный агент Справочника

АПС — агент пользователя Справочника

ОВИ — относительно выделенное имя

5 Обозначения

Обозначения, используемые в настоящей Рекомендации, определяются следующим образом:

a) нотация синтаксиса данных, кодирование и нотация макросов определены в Рекомендации X.208;

b) нотация абстрактных моделей и абстрактных служб определена в Рекомендации X.407.

РАЗДЕЛ 2 — *Общее описание*

6 Общее описание

Абстрактная служба Справочника позволяет осуществлять изучение информации Справочника, образующей ИБС, чтение этой информации и ее модификацию. Эта служба описана в терминах объекта "абстрактный Справочник", специфицированного в Рекомендации X.511.

При этом спецификация объекта "абстрактный Справочник" ни в малейшей степени не касается физической реализации Справочника и, в частности, не обращается к спецификации системного агента Справочника (САС), который хранит ИБС и управляет ИБС и обеспечивает службу. Более того, спецификация не рассматривает того, централизована ли ИБС, то есть содержится ли она у одного единственного САС или распределена среди нескольких САС. Следовательно, требования к САС, чтобы они обладали знаниями о других САС, умели прокладывать к ним маршруты и кооперироваться с ними в целях обеспечения абстрактной службы в распределенной окружающей среде, также не входят в описание службы.

В настоящей Рекомендации специфируется детализация объекта "абстрактный Справочник". Это уточнение выражается в терминах совокупности из одного или более САС-объектов, которые совместно образуют распределенную справочную службу. Такому подходу присущи идентификация и спецификация портов САС, которые являются внутренними портами объекта "Справочник". В настоящей Рекомендации специфицируются абстрактные службы и протоколы, связанные с каждым таким портом.

Кроме того, в настоящей Рекомендации специфицируются допустимые способы распределения ИБС среди одного или более САС. Для предельного случая, когда вся ИБС содержится у одного САС, Справочник по существу становится централизованным. Для случая, когда ИБС распределена среди двух или более САС, специфицируются знания и навигационные механизмы, обеспечивающие потенциальную достижимость всей ИБС любым из САС, хранящих некоторые группы статей.

Кроме того, для обеспечения запросов специфицируются такие взаимодействия, которые позволяют пользователям Справочника управлять отдельными функциональными характеристиками Справочника. В частности, пользователь может управлять решением следующего вопроса: будет ли САС, к которому поступил запрос, относящийся к информации, хранящейся у другого САС, иметь возможность самому обратиться с соответствующим запросом к другому (гим) САС (сцепление/многоадресная рассылка), или он должен будет выдать информацию о другом (гих) САС, который (рые) продолжают опрос (отсылка).

Обычно вопрос о том, использовать ли сцепление/многоадресную рассылку или отсылку, решается на основании параметров службы, устанавливаемых пользователем, а также на основании различных административных, функциональных и технических обстоятельств у самого САС.

Учитывая, что в общем случае Справочник будет распределенным и что поиск в Справочнике будет осуществлен произвольным числом сотрудничающих между собой САС, которые могут на основании вышеизложенного критерия прибегать либо к сцеплению/многоадресной рассылке, либо к отсылке, в настоящей Рекомендации специфицируются подходящие процедуры, которыми будут пользоваться САС при осуществлении распределенного справочного поиска. Эти процедуры должны создать у пользователя ощущение того, что распределенная служба Справочника является равно полноценной и удобной для него.

РАЗДЕЛ 3 — *Модели распределенного Справочника*

7 Системная модель распределенного Справочника

Абстрактная служба Справочника, как она определена в Рекомендации X.511, моделируется в виде объекта, обеспечивающего пользователю набор справочных служб. Службы Справочника моделируются в терминах портов и при этом каждый порт обеспечивает определенный набор справочных служб. Пользователи справочника получают доступ к его службам через пункты доступа. Справочник может иметь один или более пунктов доступа, причем каждый пункт доступа характеризуется набором предоставляемых им служб и режимом взаимодействия, используемым для предоставления этих служб.

Настоящий параграф посвящен внутренней структуре справочного объекта (то есть идентифицируются его составные объекты и их порты) и таким образом облегчает спецификацию услуг распределенного Справочника.

На рис. 1/X.518 изображен распределенный Справочник. Это изображение используется в дальнейшем в качестве основы для спецификации аспектов Справочника, касающихся его распределенности. Этот рисунок представляет справочный объект как состоящий из одного или более САС-объектов.

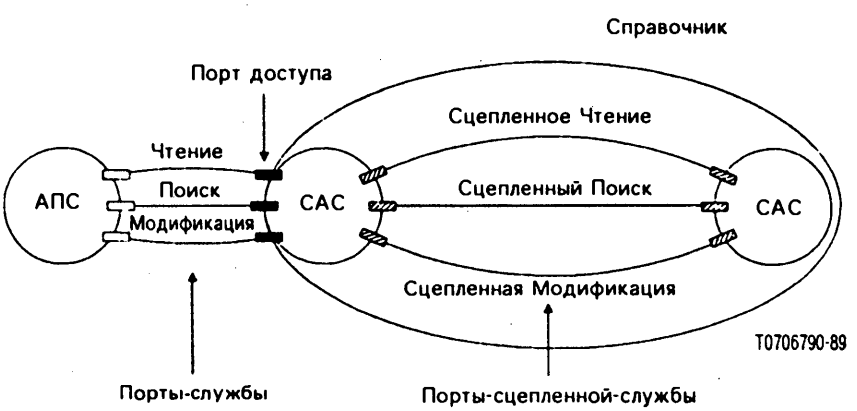


РИСУНОК 1/X.518

Объекты модели распределенного Справочника

САС-объекты детально специфицируются в последующих пунктах настоящей Рекомендации. В настоящем пункте всего лишь отмечаются несколько характеристик САС-объектов. Цель здесь двоякая: послужить введением и установить связь между этой Рекомендацией и другими Рекомендациями.

САС-объекты вводятся, чтобы разместить ИДС и чтобы физически распределенные САС могли взаимодействовать в заранее установленном, совместном режиме, обеспечивая тем самым справочные службы пользователям Справочника (АПС).

САС-объекты, так же как и объект "Справочник", характеризуются обозреваемыми снаружи портами. Порты, ассоциируемые с САС-объектами, бывают двух типов: порты-службы и порты-сцепленной-службы.

Порты-службы САС-объекта идентичны портам-службы объекта "Справочник", а именно **чтения**, поиска и модификации. На рис. 1/X.518 показано, что порты-службы, ассоциируемые с САС-объектом, образуют пункты доступа, через которые становятся доступными справочные службы.

Подробное описание портов-службы **чтения**, поиска и модификации САС-объекта, приводится в Рекомендации X.511. (Спецификация протокола соответствующих элементов прикладной службы ВОС, выводимых из этих определений портов, приводится в Рекомендации X.519.)

Помимо портов-службы САС-объекта, призванных обеспечить доступ к объекту "Справочник", определен еще и второй набор портов, названных портами-сцепленной-службы. Они обеспечивают взаимосвязь САС между собой, необходимую для того чтобы абстрактная служба Справочника могла быть осуществлена в распределенной среде.

Порты-сцепленной-службы и операции, доступные через них, находятся в прямом соответствии с аналогично именуемыми портами-службы; это соответственно **сцепленноеЧтение**, **сцепленныйПоиск** и **сцепленнаяМодификация**.

Процесс спецификации составных объектов более абстрактного объекта называется "детализацией". Спецификация детализации объекта "Справочник" выделением его составных частей (а именно САС) и спецификация абстрактных служб, предоставляемых каждой из них (абстрактная служба САС), содержится в разделе 4 настоящей Рекомендации. Спецификация протокола соответствующих элементов прикладных служб ВОС, выводимых из определения сцепленных портов, приводится в Рекомендации X.519.

8 Модель взаимодействия САС

Основной характеристикой Справочника является следующая: по имеющейся распределенной ИБС пользователь должен обладать потенциальной возможностью получить ответ на любую запрошенную им службу (конечно, если это не противоречит требованиям секретности, управления доступом и политике администрации), независимо от того, в каком пункте доступа он запросил службу. Для обеспечения этого требования необходимо, чтобы каждый САС, вовлеченный в обеспечение любого конкретного запроса на службу, обладал знаниями (как они определены в § 10 настоящей Рекомендации) о том, где расположена запрошенная информация, и либо возвратил эти знания тому, кто сделал запрос, либо попытался удовлетворить заявку от своего имени. (Реквестором может быть либо АПС, либо другой САС; в последнем случае оба САС должны иметь сцепленные порты.)

Чтобы можно было выполнить указанные выше действия, определены три режима взаимодействия САС, а именно: "сцепление", "многоадресная рассылка" и "отсылка". При этом "сцепление" и "многоадресная рассылка" предназначены для выполнения второго из двух указанных требований, а "отсылка" — первого из них.

8.1 *Сцепление*

Этот режим связи (изображенный на рис. 2/X.518) может быть использован одним САС для передачи запроса другому САС в случае, если первый из них обладает знаниями о тех именующих контекстах, которыми обладает второй. Сцепление может быть использовано для вступления в контакт с одним единственным САС, указанном в перекрестной ссылке: ссылке вверх или ссылке вниз. Многоадресная рассылка является одной из форм сцепления, описываемой в § 8.2.

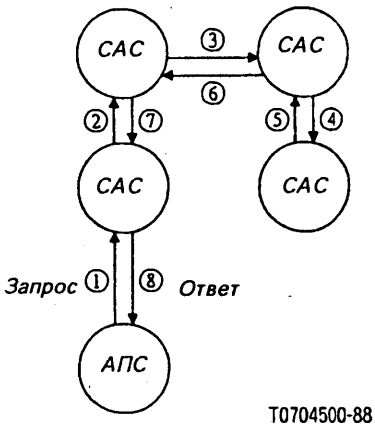


РИСУНОК 2/X.518

Режим сцепления

Примечание. — Порядок актов взаимодействия на рис. 2/X.518 определяется номерами, стоящими при стрелках.

8.2 *Многоадресная рассылка*

Этот режим связи (изображенный на рис. 3a/X.518 и 3b/X.518) используется одним САС для сцепления идентичных запросов с одним или более другими САС, причем параллельно (a) или последовательно (b). Этот режим выбирается тогда, когда первому САС неизвестны полные именующие контексты, которыми обладают другие САС. Многоадресная рассылка используется некоторым САС только для вступления в контакт с другими САС, указанными в неспецифицированной ссылке вниз. Каждому САС передается идентичный запрос. Обычно, в процессе разрешения имени только один из САС будет в состоянии продолжить обработку удаленной операции, а все остальные должны вернуть ошибкуСлужбы продвижениеНевозможно. Тем не менее, пока длится фаза распознавания при выполнении операций "поиск" и "список", все САС, указанные в неспецифицированной ссылке вниз, должны иметь возможность продолжить обработку запроса.

Примечание. — Порядок актов взаимодействия (на рис. 3a/X.518 и рис. 3b/X.518 определяется номерами, стоящими при стрелках.

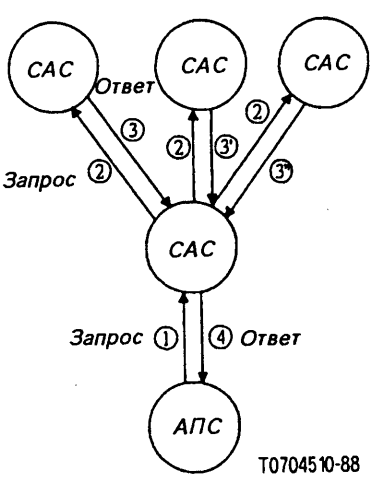


РИСУНОК 3a/X.518

Режим многоадресной рассылки

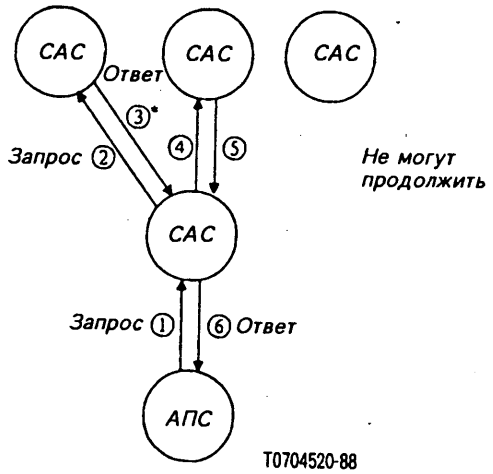


РИСУНОК 3b/X.518

Режим многоадресной рассылки

8.3 *Отсылка*

Отсылка (изображенная на рис. 4a/X.518 и 4b/X.518) возвращается САС в его ответе на запрос, о выполнении которого к нему обратился либо АПС, либо другой САС (в таком случае оба САС должны иметь сцепленные порты). Отсылка может занять либо весь ответ (в этом случае она рассматривается как ошибка), либо только часть ответа. Отсылка содержит ссылочные знания, которые могут быть либо ссылкой вверх, либо ссылкой вниз, либо перекрестной ссылкой, либо неспецифицированной ссылкой вниз.

САС (рис. 4a/X.518), получив отсылку, может использовать содержащиеся в ней ссылочные знания, чтобы затем выполнить сцепление или многоадресную рассылку (в зависимости от типа ссылки) исходной операции другим САС. Альтернативно, САС, получив отсылку, может в свою очередь возвратить отсылку назад в своем ответе. АПС (рис. 4b/X.518), получив отсылку, может использовать ее для вступления в контакт с одним или несколькими другими САС для передачи им запроса.

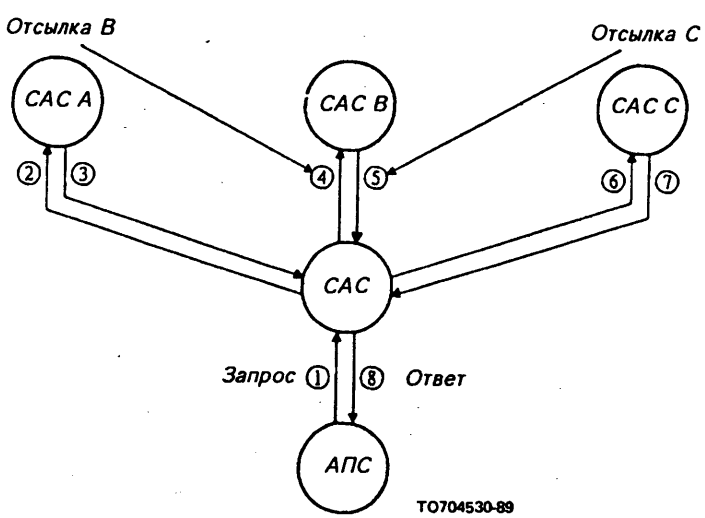


РИСУНОК 4a/X.518

Режим отсылки — САС обладает портом сцепления

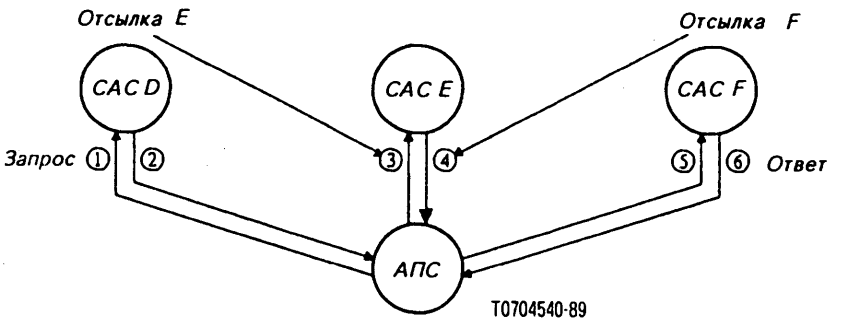


РИСУНОК 4b/X.518

Режим отсылки — АПС запрашивает САС, не обладающие портами сцепления

Примечание. — Порядок актов взаимодействия на рис. 4a/X.518 и 4b/X.518 определяется номерами, стоящими при стрелках.

8.4 *Определение режима*

Если САС не может сам полностью удовлетворить запрос, он должен использовать сцепление/многоадресную рассылку для передачи запроса (или запроса, образованного расчленением исходного) другому САС. Это недопустимо в двух случаях:

a) сцепление запрещено параметрами службы, установленными пользователем; в этом случае САС должен возвратить отсылку либо сообщение ошибкаСлужбы требуетсяСцепление (по его усмотрению); или

b) у САС имеются административные, функциональные или технические причины, в силу которых он предпочитает не осуществлять сцепления; в этом случае САС должен возвратить отсылку.

Примечание 1. — "Технической причиной", недопускающей сцепление/многоадресную рассылку, является отсутствие порта сцепления у того САС, который был указан в ссылочных знаниях.

Примечание 2. — Если параметр службы локальнаяОбластьПрименения установлен, то САС (или ОУС) должен либо сам разрешить запрос, либо возвратить сообщение об ошибке.

Примечание 3. — Если пользователь предпочитает отсылки, то он должен установить запрещено Сцепление.

9 Распределение Справочника

В настоящем параграфе устанавливаются принципы, на основании которых может быть осуществлено распределение Справочника.

Каждая статья в ИДС находится под управлением одного и только одного Администратора САС. Говорят, что такой Администратор облечен административной властью над этой статьей. Сопровождение и управление статьей может осуществлять только тот САС, который находится под одним со статьей административным управлением.

Хотя Справочник не предоставляет никаких средств для производства копий статей, тем не менее копирование можно осуществить двумя способами:

— копии статей могут быть помещены у другого САС на основании двустороннего соглашения. Средства сопровождения и управления этими копиями должны быть оговорены в двустороннем соглашении и в настоящей Рекомендации не определяются;

— копии статей могут быть получены помещением в память (локально или динамически) копии статьи, полученной в результате запроса.

Примечание. — Приобретение "закрытых" копий подчинено управлению доступом.

Пункт порождения запроса будет уведомлен (с помощью изКопии) о том, извлечена ли информация, полученная им в ответе на запрос, из копии или нет. Существует параметр службы неИспользоватьКопий, с помощью которого пользователь может запретить использование копий.

Каждый САС, входящий в Справочник, владеет *фрагментом* ИБС. Фрагмент ИБС, находящийся у некоторого САС, описан в терминах ИДС и содержит один или более именующих контекстов. *Именующий контекст* является частичным поддеревом дерева ИДС. Согласно определению он начинается с вершины и распрост-

раняется вниз к вершинам, являющимся или не являющимся листьями дерева. Эти вершины образуют границу именующего контекста. Вершины, входящие в границу, и не являющиеся листьями, обозначают начало последующего именующего контекста.

Возможен случай, при котором администратор САС будет осуществлять административную власть над несколькими раздельными именующими контекстами, то есть над контекстами, не имеющими общей высшей вершины. Если САС обладает административной властью над некоторым именующим контекстом, то он должен логически содержать последовательность ОВИ, ведущих от корня ИДС к начальной вершине того поддерева, которое образует данный именующий контекст. Эта последовательность ОВИ называется *префиксальным контекстом.*

Администратор некоторого САС может передать власть над любой статьей, непосредственно следующей за статьей, принадлежащей данному САС, другому САС. Тот САС, который передал свои полномочия, называется *предшествующим САС,* а контекст, в который входит вершина, предшествующая той вершине, власть над которой была передана, называется *предшествующим именующим контекстом.* Передача административной власти начинается от корня дерева и распространяется сверху вниз по ИДС. Иными словами, она может иметь место только от вершины к ее подчиненным.

На рис. 5/X.518 изображено гипотетическое дерево, логически разбитое на пять именующих контекстов (обозначенных буквами A, B, C, D, E), которые физически распределены между тремя САС (САС1, САС2 и САС3).

Из приведенного примера видно, что именующие контексты, принадлежащие какому-то конкретному САС, могут быть сгруппированы таким образом, чтобы быть удобными для широкого круга функциональных требований. Некоторые САС могут быть образованы таким образом, чтобы содержать статьи, изображающие более высокие уровни именующих областей в пределах некоторых логических частей ИБС, например организационные подразделения большой компании, но не обязательно все нижележащие статьи. Альтернативно, другие САС могут быть организованы так, чтобы содержать только именующие контексты, изображающие листья.

Из приведенных выше определений видно, что двумя крайними случаями будут именующие контексты, содержащие либо одну статью, либо все дерево.

Хотя отображение — логическое на физическое — ИДС на САС потенциально произвольно, задача по размещению информации и управления ею упрощаются, если САС организованы так, чтобы каждый из них содержал небольшое число именующих контекстов.

Чтобы АПС мог начать запрос, он должен обладать некоторой информацией и, в особенности, адресом в представительном уровне хотя бы одного САС, к которому он мог бы первоначально обратиться. Вопрос о том, как он приобретает и хранит эту информацию, входит в местную компетенцию.

Может случиться, что в процессе модификации статей в Справочнике обнаружатся несогласованности. Это становится особенно вероятным в случае, когда модификация затрагивает статьи псевдонимов или соответствующие им статьи объектов, которые могут быть размещены по разным САС. Несогласованность должна корректироваться специальными административными действиями, например удалением статей псевдонимов, если были удалены соответствующие им статьи объектов. Справочник должен продолжать функционировать и в периоды таких несогласованностей.

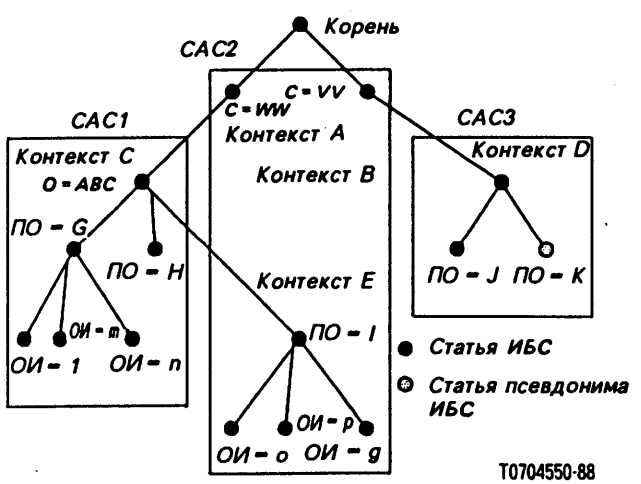


РИСУНОК 5/X.518

Гипотетическое ИДС

Примечание. — Корень не входит ни в один САС. Однако должна существовать какая-то индикация (на местном уровне) того, что такие-то вершины (например, вершины С = VV и С = WW) являются непосредственно следующими за корневой.

10 Знания

Потенциально ИБС распределена между несколькими САС так, что каждый САС владеет некоторым фрагментом ИБС. Принципы, на основании которых происходит распределение ИБС, описаны в § 9 настоящей Рекомендации.

К Справочнику предъявляется требование, согласно которому распределенность Справочника должна — по отношению к конкретным видам запросов — быть прозрачной, благодаря чему создается эффект как бы сосредоточенности всей ИБС в каждом из САС.

Чтобы обеспечить указанные функциональные требования, необходимо, чтобы каждый САС, владеющий фрагментом ИБС, умел идентифицировать и, возможно, взаимодействовать с другими фрагментами ИБС, хранящимися у других САС.

В данном параграфе определяются необходимые знания, являющиеся основой отображения имени на место его помещения во фрагменте ИДС.

Концептуально САС хранит два типа информации:

a) справочную информацию;

b) информационные знания.

Справочная информация состоит из совокупности статей, образующих именующий (щие) контекст (ы), на который (рые) распространяется административная власть того Администратора, которому подчинен данный САС.

Информационные знания включают в себя Именующий (щие) Контекст (ы), хранящийся (щиеся) у некоторого конкретного САС, и сведения о том, как этот (эти) Контекст (ы) соотносится (сятся) с иерархической структурой всего ИДС. Разрешение имени, то есть процесс соотнесения имени некоторой статьи с тем САС, который имеет административную власть над данной статьей, основывается на информационных знаниях.

Префиксальный контекст образован последовательностью ОВИ, ведущих от корня ИДС к начальной вершине именующего контекста, и соответствует выделенному имени этой вершины.

Именующий контекст состоит из набора ссылочных знаний и префиксального контекста. Именующий Контекст должен содержать именно следующие ссылочные знания:

— все внутренние ссылки, определяющие внутреннюю структуру части ИДС, входящей в Именующий Контекст;

— все ссылки вниз и все неспецифицированные ссылки вниз к другим Именующим Контекстам.

10.1 *Минимальные ссылочные знания*

Справочник должен обеспечить достижимость каждой статьи каждым АПС, независимо от того, где был выработан запрос.

Чтобы удовлетворить этому требованию, САС должен обеспечить как минимум следующие ссылочные знания:

— *ссылки вниз,* как они определены в § 10.3.2, и/или неспецифицированные ссылки вниз, как они определены в § 10.3.5; и

— *ссылки вверх,* как они определены в § 10.3.3.

Это обеспечит возможность установления *ветви ссылок* в виде непрерывной последовательности ссылочных знаний по всем именующим контекстам в Справочнике.

Возможно, что перекрестные ссылки, как они определены в § 10.3.4, смогут образовать некоторые части ветви ссылок, оптимизируя тем самым производительность.

10.2 *Корневой контекст*

В силу независимости различных стран или межнациональных компаний не существует такого "одного САС", который содержал бы корневой контекст. Функциональные обязанности "корневого САС", связанные с разрешением имени, должны быть возложены на те САС, которые обладают административной властью над

именующими контекстами, непосредственно следующими за корнем. Эти САС именуются *САС Первого уровня*. Каждый САС первого уровня должен уметь "симулировать" поведение "корневого САС". Это требует полной информации о корневом именующем контексте. Корневой контекст репродуцируется у каждого САС первого уровня, и поэтому должен управляться совместно независимыми административными руководящими органами первого уровня. Административные процедуры должны быть определены в многосторонних соглашениях за пределами предмета рассмотрения настоящей Рекомендации.

— Каждый САС первого уровня должен обладать корневым контекстом, что включает в себя ветвь ссылок к любому другому САС первого уровня.

— Каждый САС не первого уровня должен обладать ссылкой вверх, что включает в себя ветвь ссылок к любому САС первого уровня.

10.3 *Ссылочные знания*

Знания, которыми обладает некоторый САС, определяются в терминах набора одного или нескольких ссылочных знаний, причем каждая ссылка ассоциирует прямо или косвенно статьи ИБС с САС, владеющим этими статьями.

Для обеспечения достижимости каждой статьи ИБС каждым САС требуется, чтобы каждый САС обладал знаниями о статьях, содержащихся у него самого, а также о последующих и, возможно, предшествующих статьях. Так возникают следующие типы ссылочных знаний:

— внутренние ссылки;

— ссылки вниз;

— ссылки вверх;

— неспецифицированные ссылки вниз.

Кроме того, в оптимизационных целях рассматривается также следующий необязательный тип ссылок:

— перекрестные ссылки.

В случае, если знания некоторого САС ограничиваются только внутренними ссылками, то такой САС не имеет никаких сведений о других САС и таким образом ИБС становится централизованной.

10.3.1 *Внутренние ссылки*

Внутренняя ссылка содержит:

— ОВИ, соответствующее статье ИБС;

— внутренний указатель, определяющий, где в локальной ИБС размещена статья. (Спецификация этого указателя не входит в предмет рассмотрения настоящей Рекомендации.)

Все статьи, находящиеся под административным управлением некоторого конкретного САС, представлены внутренними ссылками в информационных знаниях данного САС.

10.3.2 *Ссылка вниз*

Ссылка вниз содержит:

— ОВИ, соответствующее непосредственно следующей статье ИБС;

— пункт доступа к тому САС, которому было делегировано административное управление этой статьей.

Все последующие статьи, которыми владеет другой САС и которому данный САС передал административное управление этими статьями, должны быть представлены ссылками вниз (или неспецифицированными ссылками вниз, как это описано в § 10.3.5).

10.3.3 *Ссылки вверх*

Ссылка вверх содержит:

— пункт доступа к САС.

Каждый САС не первого уровня содержит ровно одну ссылку вверх. Ссылка вверх образует часть ветви ссылок к корневому узлу. Для обеспечения этого должен существовать некоторый метод вне стандарта, например внутри ОУС, или должна существовать ссылка на тот САС, который содержит именующий контекст такого префиксального контекста, который содержит меньше ОВИ, чем префиксальный контекст, принадлежащий данному САС и содержащий минимальное число ОВИ.

Если вводится новый САС не первого уровня, то он должен обладать минимальными начальными знаниями, представленными ссылками вверх. Всякие дальнейшие знания будут добавлены ссылками вниз или

перекрестными ссылками (как это описано в § 10.3.4). Если вводится новый САС первого уровня, то он должен приобрести корневой контекст и уведомить все остальные САС первого уровня. Осуществление этого не входит в предмет рассмотрения настоящей Рекомендации.

10.3.4 *Перекрестные ссылки*

Перекрестная ссылка содержит:

— префиксальный контекст;

— пункт доступа к тому САС, который осуществляет административное управление данным именующим контекстом.

Этот тип ссылок является необязательным и служит целям оптимизации Разрешения Имени. САС может содержать любое число (включая ноль) перекрестных ссылок.

10.3.5 *Неспецифицированная ссылка вниз*

Неспецифицированная ссылка вниз содержит:

— пункт доступа к тому САС, который содержит хоть один непосредственно следующий именующий контекст.

Этот тип ссылок является необязательным. Он предназначен для случая, когда известно, что САС содержит несколько подчиненных статей, но конкретные ОВИ этих статей неизвестны.

Для каждого именующего контекста, которым владеет САС, он может содержать любое число (включая ноль) неспецифицированных ссылок вниз, которые будут выявлены, если будут продолжены все специфицированные внутренние ссылки и ссылки вниз. Те САС, доступ к которым будет осуществлен средствами неспецифицированных ссылок, должны уметь непосредственно разрешать запрос (успешно или неуспешно). В случае неуспешного результата реквестору возвращается ошибкаСлужбы ПродвижениеНевозможно.

10.4 *Управление знаниями*

Чтобы пользоваться широко распределенным Справочником, обеспечивающим приемлемые уровни согласованности и производительности, требуется наличие процедур сопровождения и развития знаний, которыми обладают САС. Те же процедуры требуются для создания начальных знаний.

Знания могут быть обеспечены средствами:

a) САС или его административным органом, сообщающим об изменениях в знаниях тем САС, которые обладают любыми видами ссылок на данный САС, если только изменения в данном САС делают недействительными эти ссылки. Это единственный путь поддержания ссылок вверх, ссылок вниз и неспецифицированных ссылок вниз.

b) САС, выдающими и получающими перекрестные ссылки для повышения производительности с помощью обычных операций Справочника.

В настоящей Рекомендации процедуры, распространяющие знания об изменениях так, как это описано в пункте a), не определяются. О таких процедурах должны быть разработаны двусторонние соглашения.

10.4.1 *Расширение перекрестных ссылок*

Для повышения производительности Справочника можно расширять набор локальных перекрестных ссылок, используя для этого обычные операции Справочника. Если САС обладает портом сцепления, то он может запросить у другого САС (который также должен иметь порт сцепления) возвратить те ссылочные знания, которые содержат данные о размещении именующих контекстов, имеющих отношение к именам объектов, являющихся конечной целью обычной операции Справочника.

Если компоненту **возвратитьПерекрССылку АргументаСцепления** дано значение **TRUE**, то, возможно, будет присутствовать компонент **перекрестнаяССылка результатаСцепления**; этот компонент будет состоять из последовательности элементов, являющихся перекрестными ссылками.

Если САС не может осуществить сцепление запроса со следующим САС, то исходному САС возвращается отсылка. Если компонент **возвратитьПерекрССылку** аргумента сцепления имел значение **TRUE**, то отсылка может дополнительно содержать префиксальный контекст именующего контекста, на который ссылается отсылка. Компонент **префиксальныйКонтекст** отсутствует, если основой отсылки является неспецифицированная ссылка вниз. Перекрестная ссылка, возвращаемая в отсылке, основывается только на знаниях, принадлежащих тому САС, который выработал отсылку.

В обоих случаях (сцепление и отсылка) административный орган через своего САС может предпочесть проигнорировать запрос на возврат перекрестных ссылок.

10.4.2 *Несогласованность знаний*

Справочник должен обеспечить механизмы проверки согласованности, чтобы гарантировать некоторый уровень согласованности знаний.

10.4.2.1 *Обнаружение несогласованности знаний*

Характер несогласованности и ее обнаружение меняются в зависимости от вида ссылочных знаний.

— *Перекрестные ссылки и ссылки вниз:*

Этот тип ссылок оказывается недействительным, если САС, на который делалась ссылка, не имеет локального именующего контекста, префиксальный контекст которого содержится в ссылке. Эта несогласованность будет обнаружена в течение определения начального именующего контекста процесса разрешения имени за счет продвижения операции и типов компонентов ссылок АргументаСцепления.

– *Неспецифицированные ссылки вниз:*

Этот тип ссылок оказывается недействительным, если САС, на который делалась ссылка, не имеет локального именующего контекста, непосредственный префиксальный контекст которого содержится в ссылке, то есть ссылка содержит локальный префиксальный контекст данного САС, но без последнего ОВИ. Проверка согласованности осуществляется, как указано выше.

— *Ссылка вверх:*

Недействительная ссылка вверх — это такая ссылка, которая не является частью ветви ссылок к корню. Поддержание ссылок вверх должно осуществляться внешними средствами и не входит в предмет рассмотрения настоящей Рекомендации.

Примечание. — Не всегда возможно обнаружить недействительную ссылку вверх.

10.4.2.2 *Сообщения о несогласованности знаний*

Если для выполнения запроса к Справочнику осуществлялось сцепление, то все несогласованности в знаниях будут обнаружены тем САС, которому принадлежит недействительная ссылочная информация; он обнаружит это благодаря тому, что получит ОшибкуСлужбы, в которой указана недействительнаяСсылка.

Если САС возвращает отсылку, основанную на недействительной ссылочной информации, то реквестору, если он использует отсылку, будет выдана ошибкаСлужбы, в которой указана недействительнаяСсылка. Вопрос о том, как обнаруженная ошибка будет продвинута к тому САС, которому принадлежит недействительная ссылка, не входит в предмет рассмотрения настоящей Рекомендации.

10.4.2.3 *Обработка недействительной ссылочной информации*

После того как САС обнаружит недействительную ссылку, он должен предпринять попытки восстановить согласованность знаний. Этого можно достигнуть, например, простым исключением недействительной перекрестной ссылки или заменой ее на действительную, которая может быть получена посредством использования механизма запроситьПерекрестнуюСсылку.

Метод, согласно которому САС фактически обрабатывает недействительные ссылки, является локальной проблемой и не входит в предмет рассмотрения настоящей Рекомендации.

РАЗДЕЛ 4 — *Абстрактная служба САС*

11 Общее описание абстрактной службы САС

11.1 Абстрактная служба Справочника полностью описана в Рекомендации X.511. Если такая служба обеспечивается в распределенной среде, как это смоделировано в § 7 настоящей Рекомендации, то можно считать, что она обеспечивается набором САС. Это изображено на рис. 1/X.518.

11.2 Для описания этой модели детализация объекта Справочник разложением на компоненты, то есть на сас-объекты, может быть представлена следующим образом:

```
ДетализацияСправочника ::= REFINE справочник AS
    cac                    RECURRING
        портЧтения                      [S] VISIBLE
        портПоиска                      [S] VISIBLE
        портМодификации                 [S] VISIBLE
        портСцепленногоЧтения           PAIRED WITH cac
        портСцепленногоПоиска           PAIRED WITH cac
        портСцепленнойМодификации       PAIRED WITH cac
```

11.3 Cac-объект может быть определен следующим образом:

```
cac OBJECT
    PORTS { портЧтения                  [S],
            портПоиска                  [S],
            портМодификации             [S],
            портСцепленногоЧтения,
            портСцепленногоПоиска,
            портСцепленнойМодификации}

::= ид-от-cac
```

CAC обеспечивает порты Чтения, Поиска и Модификации, делая таким образом обозримыми эти службы пользователям объекта "Справочник", а именно АПС. Кроме того, CAC обеспечивает "сцепленные" версии этих портов, а именно Сцепленного Чтения, Сцепленного Поиска и Сцепленной Модификации. Это позволяет CAC продвигать запросы на эти службы к другим CAC.

11.4 Порты, указанные в § 11.2 и 11.3 (исключая те, которые определены в Рекомендации X.511), определяются следующим образом:

```
портСцепленногоЧтения      PORT
    ABSTRACT OPERATIONS {
        СцепленноеЧтение, СцепленноеСравнение,
        СцепленныйОтказ }
    ::= ид-пт-сцепленного-чтения

портСцепленногоПоиска      PORT
    ABSTRACT OPERATIONS {
        СцепленныйСписок, СцепленныйПоиск }
    ::= ид-пт-сцепленного-поиска

портСцепленной Модификации  PORT
    ABSTRACT OPERATIONS {
        СцепленноеДобавлениеСтатьи,
        СцепленноеУдалениеСтатьи,
        СцепленнаяМодификацияСтатьи,
        СцепленнаяМодификацияОВИ }
    ::= ид-пт-сцепленной-модификации
```

12 Типы информации

12.1 *Введение*

12.1.1 В настоящем параграфе идентифицируются, а в некоторых случаях определяются, несколько типов информации, которые в последующем используются при определении различных операций абстрактной службы CAC. Рассматриваются такие типы информации, которые присущи более чем одной операции, или, по-видимому, станут таковыми в будущем. Кроме того, рассматриваются типы, которые либо достаточно сложны, либо настолько самостоятельно закончены, что заслуживают специального определения, отделенного от описания использующих их операций.

12.1.2 Некоторые из типов информации, используемые при описании абстрактной службы CAC, фактически определены в других местах. В § 12.2 эти типы идентифицируются и указываются источники, в которых они определяются. Во всех остальных (§ 12.3—12.9) идентифицируются и определяются типы информации.

12.2 *Типы информации, определенные вне настоящей Рекомендации*

12.2.1 Следующие типы информации определены в Рекомендации X.501:

a) **имяОбъектаПсевдонима**;

b) **ВыделенноеИмя**;

c) **Имя**;

d) **ОтносительноВыделенноеИмя**.

12.2.2 Следующие типы информации определены в Рекомендации X.511:

(Абстрактное привязывание)

a) **ПривязываниеКСправочнику**

(Абстрактные-операции)

b) **Отказ**;

(Абстрактные-ошибки)

c) **Отказано**;

d) **ОшибкаАтрибута**;

e) **ОшибкаИмени**;

f) **ОшибкаБезопасности**;

g) **ОшибкаСлужбы**;

h) **ОшибкаОбновления**;

(Макрос)

i) **OPTIONALLY-SIGNED**

(Типы данных)

j) **ПараметрыБезопасности**

12.2.3 Следующий тип информации определен в Рекомендации X.520.

a) **АдресВУровнеПредставлений**

12.3 *Аргументы сцепления*

12.3.1 Каждая Сцепленная абстрактная операция содержит АргументыСцепления, передающие САС информацию, требующуюся для успешного выполнения им его доли общей задачи:

```
АргументыСцепления ::=      SET{
    пунктПорождения          [0]  ВыделенноеИмя OPTIONAL,
    целевойОбъект            [1]  ВыделенноеИмя OPTIONAL,
    продвижениеОперации      [2]  ПродвижениеОперации DEFAULT { неНачато },
    информацияСледа          [3]  ИнформацияСледа,
    псевдонимПереименован    [4]  BOOLEAN DEFAULT FALSE,
    оВИОбъектаПсевдонима     [5]  INTEGER OPTIONAL
        -- используется только если псевдонимПереименован = TRUE
    возвратитьПерекрССылки   [6]  BOOLEAN DEFAULT FALSE,
    типССылки                [7]  ТипССылки DEFAULT вверх,
    инфо                     [8]  ОбластьИнфо OPTIONAL,
    ограничениеВремени       [9]  ВремяОтГринвича OPTIONAL,
                             [10] ПараметрыБезопасности DEFAULT {}}
```

12.3.2 В §§ 12.3.2.1 — 12.3.2.11 определяется смысл различных компонентов.

12.3.2.1 Компонент **пунктПорождения** передает имя того, кто первоначально выдал запрос, если только это имя не было уже специфицировано в параметрах безопасности. Если **ОбщиеАргументы** содержат реквестора, то этот аргумент может быть опущен.

12.3.2.2 Компонент **целевойОбъект** передает имя того объекта, к статье которого сейчас прокладывается маршрут. Роль этого объекта зависит от рассматриваемой конкретной абстрактной операции: это может быть объект, статья которого подлежит обработке, или объект, статья которого послужит основой запроса или подзапроса, охватывающего несколько объектов (например, **СцепленныйСписок** или **СцепленныйПоиск**). Этот аргумент мог быть опущен только в том случае, если бы его значение совпадало со значением параметра основного объекта в **ХАргументе** (см. § 14.3.1); в таком случае его значение по умолчанию равняется этому значению.

12.3.2.3 Компонент **продвижениеОперации** используется для того, чтобы информировать САС о том, как далеко продвинулась операция, а следовательно, и о том, какую роль должен выполнить данный САС в общем выполнении операции. Информация, передаваемая в этом компоненте, специфицируется в § 12.5.

12.3.2.4 Компонент **информацияСледа** используется для предотвращения образования замкнутых петель среди САС при выполнении сцепления. Перед тем как сцепить операцию по следующим САС, каждый САС добавляет к информации о следе новый элемент. Получив запрос на выполнение операции, САС анализирует информацию о следе и убеждается в том, что операция не образовала петлю. Информация, передаваемая в этом компоненте, специфицируется в § 12.6.

12.3.2.5 Компонент **псевдонимПереименован** является Булевским значением, используемым для фиксации того, была ли в процессе распределенного разрешения имени уже встречена и переименована хоть одна статья псевдонима. Значение по умолчанию **FALSE** свидетельствует о том, что пока не была переименована ни одна статья псевдонима.

12.3.2.6 Компонент **ОВИОбъектаПсевдонима** указывает на то, сколько ОВИ, входящих в **целевойОбъектИмя**, было сгенерировано из атрибутов **имениОбъектаПсевдонима** одной или нескольких статей псевдонима. Некоторое целое значение устанавливается каждый раз, когда обрабатывается и переименовывается статья псевдонима. Этот компонент присутствует тогда и только тогда, когда значение компонента **псевдонимПереименован** равно **TRUE**.

12.3.2.7 Компонент **возвратитьПерекрССылки** является Булевским значением, являющийся индикатором того, требуется или не требуется возвратить начальному САС ссылочные знания, использованные в процессе выполнения распределенной операции; эти ссылочные знания будут возвращены в качестве перекрестных ссылок совместно с результатом отсылки. Значение по умолчанию **FALSE** означает, что эти ссылочные знания возврату не подлежат.

12.3.2.8 Компонент **ТипССылки** указывает вызванному на выполнение абстрактной операции САС, какого типа информация была использована при маршрутизации к нему запроса. Благодаря этому САС сможет обнаружить ошибку в знаниях инициатора запроса. Если такая ошибка будет обнаружена, она будет специфицирована выдачей **ошибкаСлужбы** с указанием трудности **недействительнаяССылка**. ТипССылки описывается полностью в § 12.7.

Примечание. — Если **типССылки** отсутствует, то предполагается значение, равное **вверх**.

12.3.2.9 Компонент **инфо** используется для распространения среди САС, вовлеченных в процесс общего запроса, информации, специфичной для ОУС. Тип этого компонента, названный **ОбластьИнфо**, является неопределенным:

```
ОбластьИнфо ::= ANY
```

12.3.2.10 Компонент **ограничениеВремени**, если он присутствует, устанавливает время, до которого операция должна быть завершена.

12.3.2.11 Компонент **ПараметрыБезопасности** специфицирован в Рекомендации X.511. Его отсутствие считается эквивалентным наличию пустого множества параметров безопасности.

12.4 *Результаты сцепления*

12.4.1 **РезультатСцепления** присутствует в результате любой абстрактной операции и обеспечивает обратную связь с тем САС, который возбудил абстрактную операцию.

```
РезультатСцепления ::=      SET{
    инфо                    [0] ОбластьИнфо OPTIONAL,
    перекрестныеССылки      [1] SEQUENCE OF ПерекрестнаяССылка OPTIONAL,
                            [2] ПараметрыБезопасности DEFAULT {}}
```

12.4.2 В §§ 12.4.2.1—12.4.2.3 определяется смысл различных компонентов.

12.4.2.1 Компонент **инфо** используется для распространения среди САС, вовлеченных в процесс общего запроса, информации, специфичной для ОУС. Тип этого компонента, названный **ОбластьИнфо**, является неопределенным.

12.4.2.2 Компонент **перекрестныеССылки** присутствует в **РезультатеСцепления** только тогда, когда компонент **возвратитьПерекрССылки** соответствующего запроса имеет значение **TRUE**. Этот компонент состоит из цепочки элементов **ПерекрестнаяССылка**, каждый из которых содержит **префиксальныйКонтекст** и дескриптор **пунктДоступа** (см. § 12.8).

```
ПерекрестнаяССылка ::=      SET{
    префиксальныйКонтекст   [0] ВыделенноеИмя,
    пунктДоступа            [1] ПунктДоступа}
```

САС может добавить **перекрестнуюССылку**, если один из его префиксальных контекстов совпадает с частью аргумента **целевойОбъект** абстрактной-операции. Административный орган, осуществляющий управление САС, может придерживаться политики невозвращения таких знаний. В этом случае он не станет добавлять элемента к последовательности.

12.4.2.3 Компонент параметрыБезопасности специфицирован в Рекомендации X.511. Его отсутствие считается эквивалентным наличию пустого множества параметров безопасности.

12.5 *Продвижение операции*

12.5.1 Значение продвиженияОперации описывает состояние продвижения при выполнении абстрактной-операции, в котором принимают участие несколько САС.

```
ПродвижениеОперации ::= SET {
    фазаРазрешенияИмени [0]
        ENUMERATED {
                неНачата        (1),
                продолжается    (2),
                завершена       (3) },
    следующееРазрешаемоеОВИ[1]
        INTEGER OPTIONAL }
```

12.5.2 В §§ 12.5.2.1 и 12.5.2.2 определяется смысл различных компонентов.

12.5.2.1 Компонент фазаРазрешенияИмени указывает на фазу, достигнутую при обработке имени целевого-Объекта некоторой операции. Если этот компонент указывает, что разрешение имени неНачато, то это значит, что данный САС еще не достигнут именующим контекстом, содержащим начальный (ные) ОВИ имени. Если разрешение имени продолжается, то это означает, что начальная часть имени уже была распознана, но еще не достигнут тот САС, у которого хранится целевой объект. Компонент следующееРазрешаемоеОВИ указывает на то, какая часть имени уже была раскрыта (§ 12.5.2.2). Если разрешение имени завершено, то это означает, что уже достигнут САС, хранящий целевой объект, и что продолжается выполнение собственно операции.

12.5.2.2 С помощью компонента следующееРазрешаемоеОВИ САС узнает, которое из ОВИ, входящих в имя целевогоОбъекта, сейчас подлежит разрешению. Этот компонент имеет целочисленное значение в диапазоне от единицы до числа ОВИ в имени. Наличие этого компонента возможно только в том случае, если компонент фазаРазрешенияИмени имеет значение продолжается.

12.6 *Информация о следе*

12.6.1 Значение информацияСледа несет с собой опись тех САС, которые были вовлечены в выполнение операции. Он используется для обнаружения, а следовательно, и избежания замкнутых петель, которые могут возникнуть в результате несогласованности знаний или наличия петель псевдонимов в ИДС.

```
ИнформацияСледа    ::=    SEQUENCE OF ЭлементСледа
ЭлементСледа       ::=    SET {
    сас                       [0]  Имя,
    целевойОбъект             [1]  Имя OPTIONAL,
    продвижениеОперации       [2]  ПродвижениеОперации }
```

12.6.2 Каждый САС, который продвигает выполнение операции к следующему САС, добавляет к информации о следе новый элемент. Каждый такой ЭлементСледа содержит:

a) Имя того САС, который добавляет элемент;

b) целевойОбъект Имя, полученное с входящим запросом тем САС, который сейчас добавляет элемент. Этот параметр опускается, если сцепляемый опрос поступил от АПС (в каковом случае его неявным значением является объект или базовыйОбъект в "ХОперации"), или если его значение совпадает со значением (фактическим или неявным) целевогоОбъекта в АргументеСцепления исходящего запроса;

c) продвижениеОперации, которое получил во входящем запросе тот САС, который сейчас добавляет элемент.

12.7 *Типы ссылок*

12.7.1 Значение ТипСсылки указывает на один из видов ссылки, определенных в § 10.

```
ТипСсылки ::=

    ENUMERATED {
        вверх                       (1),
        вниз                        (2),
        перекрестная                (3),
        НеСпецифицированнаяВниз     (4)}
```

12.8 *Пункт доступа*

12.8.1 Значение ПунктаДоступа специфицирует то конкретное место, через которое можно обратиться к Справочнику, а точнее, к САС. Пункт доступа имеет Имя, совпадающее с именем рассматриваемого САС, и АдресВУровнеПредставлений, который должен использоваться при обращениях к этому САС в среде ВОС.

```
ПунктДоступа  ::=      SET {
    титул-пэ           [0] Имя,
    адрес              [1] АдресВУровнеПредставлений }
```

12.9 *Ссылка на продолжение*

12.9.1 СсылкаНаПродолжение описывает, как может быть продолжено выполнение всей абстрактной операции (или ее части) обращением к другому (или другим) САС. Обычно этот параметр возвращается в качестве отсылки, когда САС, вовлеченный в выполнение операции, не может или не хочет сам распространить операцию на другие САС.

```
СсылкаНаПродолжение          ::=  SET {
    целевойОбъект            [0]  Имя,
    оВИОбъектаПсевдонима     [1]  INTEGER OPTIONAL,
    продвижениеОперации      [2]  ПродвижениеОперации,
    разрешенныеОви           [3]  INTEGER OPTIONAL,
    типСсылки                [4]  ТипСсылки

    -- -- используется только в САС
    пунктДоступа             [5]  SET OF ПунктДоступа }
```

12.9.2 В §§ 12.9.2.1—12.9.2.6 определяется смысл различных компонентов.

12.9.2.1 ЦелевойОбъект Имя используется для продолжения операции. Он может быть отличен от ЦелевойОбъект Имя, полученного в поступившем запросе; это может произойти в случае, когда, например, был переименован псевдоним или когда в поиске был локализован базовый объект.

12.9.2.2 Компонент оВИОбъектаПсевдонима указывает, сколько ОВИ (если таковые вообще были) в имени целевого объекта было выработано в результате переименований псевдонимов. Этот аргумент присутствует только, если переименование имело место.

12.9.2.3 Достигнутое продвижениеОперации; оно будет управлять дальнейшим выполнением операции названными САС, при условии, если САС или АПС, получивший ССылкуНаПродолжение, будет ей следовать.

12.9.2.4 Значение компонента разрешенныеОви (который должен использоваться только в том случае, если некоторые ОВИ не были вовлечены в полное разрешение имени, но которые считаются корректными с точки зрения перекрестных ссылок) указывает, как много ОВИ было фактически разрешено, используя только внутренние ссылки.

12.9.2.5 Компонент типСсылки, который присутствует только в абстрактной службе САС, указывает на то, какой тип знания был использован при выработке настоящего продолжения.

12.9.2.6 Компонент пунктДоступа указывает на те точки доступа, которые должны быть пройдены для осуществления настоящего продолжения. Если используется неспецифицированная ссылка вниз, то пунктовДоступа может быть не один, а несколько, и каждый из них должен быть пройден, например, с помощью многоадресной рассылки.

13 Абстрактное-привязывание и абстрактное-отвязывание

САСПривязывание и САСОтвязывание используются САС соответственно в начальный и в конечный моменты периода установления связи с другим САС.

13.1 *Привязывание САС*

13.1.1 САС использует операцию-абстрактного-привязывания САСПривязывание для привязки его портов сцепленногоЧтения, сцепленногоПоиска и сцепленнойМодификации к соответствующим портам другого САС.

```
САСПривязывание  ::=      ABSTRACT-BIND
    ТО                    {сцепленноеЧтение,
                           сцепленныйПоиск,
                           сцепленнаяМодификация }
    ПривязываниеКСправочнику
```

13.1.2 Компоненты **САСПривязывания** идентичны их двойникам в **ПривязыванииКСправочнику** (см. Рекомендацию X.511), но со следующими отличиями.

13.1.2.1 **Удостоверения АргументаПривязыванияКСправочнику** позволяют передавать информацию, идентифицирующую Титул-ПЭ того САС, который возбуждает операцию, к САС-респондеру. Титул-ПЭ должен иметь форму Выделенного Имени.

13.1.2.2 **Удостоверения РезультатаПривязыванияКСправочнику** позволяют передавать информацию, идентифицирующую Титул-ПЭ САС-респондера, к САС, возбуждающему операцию. Титул-ПЭ должен иметь форму Выделенного Имени.

13.2 *Отвязывание САС*

13.2.1 Операция **САСОтвязывания** используется для отвязывания друг от друга портов Сцепленного Чтения, Сцепленного Поиска и Сцепленной Модификации двух САС.

```
САСОтвязывание      ::=     ABSTRACT-UNBIND
    FROM                    { сцепленноеЧтение,
                              сцепленныйПоиск,
                              сцепленнаяМодификация }
```

13.2.2 Нет никаких аргументов, результатов или ошибок.

14 Сцепленные абстрактные-операции

14.1 Каждому порту абстрактной службы Справочника соответствует порт у любого САС; эти порты обеспечивают возможность совместно работающим нескольким САС поставлять абстрактную службу. В таком же взаимооднозначном соответствии находятся и абстрактные операции соответствующих портов. Имена портов и абстрактных служб выбраны таким образом, чтобы отражать это соответствие. Для этого имена портов и абстрактных операций абстрактной службы САС образованы из имен соответствующих понятий абстрактной службы Справочника добавлением к ним префикса "Сцепленный". В результате порты и операции получили названия:

```
ПортСцепленногоЧтения:          СцепленноеЧтение,
                                СцепленноеСравнение,
                                СцепленныйОтказ

ПортСцепленногоПоиска:          СцепленныйСписок,
                                СцепленныйПоиск,

ПортСцепленнойМодификации:      СцепленноеДобавлениеСтатьи,
                                СцепленноеУдалениеСтатьи,
                                СцепленнаяМодификацияСтатьи,
                                СцепленнаяМодификацияОВИ
```

14.2 Аргументы, результаты и ошибки сцепленных абстрактных операций единообразным способом (за одним исключением) образованы из аргументов, результатов и ошибок соответствующих абстрактных операций абстрактной службы Справочника (как это описано в § 14.3). Единственное исключение составляет абстрактная операция СцепленногоОтказа, которая синтаксически эквивалентна своему двойнику в абстрактной службе Справочника (описана в § 14.4).

14.3 Абстрактная-операция **СцепленноеХ** используется для распространения между несколькими САС запроса, инициированного (в нормальном случае) некоторым АПС, выдавшим запрос на абстрактную-операцию Х. Запрос был выдан тому САС, который был выбран для сцепления запроса. Аргументы абстрактной операции могут быть при желании подписаны инициатором запроса и, если это будет затребовано, результат может быть подписан тем САС, который выполнил запрос.

14.3.1 Единообразный вывод Сцепленной абстрактной операции **СцепленноеХ** из его двойника Х имеет следующий вид:

если

```
X  ::=
    ABSTRACT-OPERATION
        ARGUMENT        XАргумент
        RESULT          XРезультат
        ERRORS      { ..., Отсылка, ... },
```

то сцепленная абстрактная операция имеет вид:

```
СцепленноеX ::=
    ABSTRACT-OPERATION
        ARGUMENT OPTIONALLY-SIGNED SET {
                АргументСцепления,
                [0] АргументX }
        RESULT OPTIONALLY-SIGNED SET {
                РезультатСцепления
                [0] РезультатX }
        ERRORS {..., СасОтсылка, ...}
```

Примечание. — Окончательная спецификация абстрактной службы САС, приведенная в Приложении А, полностью применяет этот метод вывода к Сцепленным абстрактным-операциям.

14.3.2 В § 14.3.2.1 и § 14.3.2.2 определяется смысл аргументов введенных абстрактных-операций.

14.3.2.1 **АргументСцепления** содержит информацию сверх и помимо той, которая была выдана АПС-инициатором и которая нужна текущему САС для выполнения операции. Этот тип информации определен в § 12.3.

14.3.2.2 **XАргумент** содержит исходные аргументы, выданные АПС-инициатором; эти аргументы специфицируются в соответствующих пунктах Рекомендации X.511.

14.3.3 Если запрос завершается успешно, то возвращается результат. Смысл параметров результата описан в § 14.3.3.1 и § 14.3.3.2.

14.3.3.1 **РезультатСцепления** содержит информацию сверх и помимо той, которая должна быть доставлена АПС-инициатору и которая может понадобиться предшествующим САС в цепочке. Этот тип информации определен в § 12.4.

14.3.3.2 **XРезультат** содержит результат, возвращенный тем САС, который выполнил запрос; это та информация, которая должна быть передана в обратном направлении в результате, выдаваемом АПС-инициатору. Эта информация специфицируется в соответствующих пунктах Рекомендации X.511.

14.3.4 Если запрос заканчивается безуспешно, то возвращается одна из ошибок, приводимых в списке. Составы ошибок, которые могут быть сообщены, те же, что и описанные для соответствующих абстрактных операций в Рекомендации X.511, за исключением того, что вместо Отсылки будет выдаваться САС-отсылка. Различные ошибки определены в § 15 или в § 15 указаны ссылки на них.

14.4 Абстрактная-операция **СцепленныйОтказ** используется одним САС для того, чтобы оповестить другой САС о том, что он более не заинтересован в продолжении возбужденной им ранее операции. Это может иметь множество разных оснований, примерами чего могут служить следующие:

a) в отношении исходной операции, в результате которой запрос был передан другому САС, был выполнен отказ или исходная операция была неявно прекращена из-за разрыва ассоциации;

b) САС получил требующуюся ему информацию другим способом, например от более быстро реагирующего другого САС, вовлеченного в многоадресную рассылку.

Однако САС не обязан выдать СцепленныйОтказ или прекратить операцию, если даже к нему поступил запрос на это.

Если СцепленныйОтказ завершается успешно, то есть если ему удается прекратить выполнение операции, то будет возвращен результат, в котором прекращенная операция указывает абстрактную-ошибку Отказано. Если же операции СцепленныйОтказ не удается прервать выполнение другой операции, то она сама возвратит ошибку НевыполненныйОтказ.

15 Сцепленные абстрактные-ошибки

15.1 *Введение*

15.1.1 В основном абстрактная служба САС может возвратить те же абстрактные-ошибки, что и абстрактная-служба Справочника. Исключением является возврат "ошибки" САСОтсылка (см. § 15.2) вместо Отсылка. Кроме того, нижеперечисленные трудности службы имеют тот же синтаксис, но отличную семантику.

a) недействительнаяСсылка;

b) обнаруженаПетля.

15.1.2 Приоритеты абстрактных-ошибок совпадают с их приоритетом для абстрактной службы Справочника, как это специфицировано в Рекомендации X.511.

15.2 *CAC-отсылка*

15.2.1 Абстрактная-ошибка CACОтсылка вырабатывается CAC в том случае, если независимо от побуждающих причин, он не желает продолжать выполнение абстрактной операции методом сцепления или многоадресной рассылки, направленным к одному или нескольким CAC. Обстоятельства, при которых CAC может возвратить отсылку, описаны в § 8.4.

```
CACОтсылка ::=
    ABSTRACT-ERROR
        PARAMETER SET {
            [0] ССылкаНаПродолжение,
            префиксальныйКонтекст [1] ВыделенноеИмя OPTIONAL }
```

15.2.2 В §§ 15.2.2.1 и 15.2.2.2 определяется смысл различных параметров.

15.2.2.1 Компонент ССылкаНаПродолжение содержит информацию, требующуюся возбудителю для продвижения дальнейшего соответствующего запроса, возможно к другому CAC. Этот тип информации определен в § 12.9.

15.2.2.2 Если компонент возвратитьПерекрССылки АргументовСцепления данной абстрактной-операции имеет значение TRUE и отсылка основывалась на ссылке вниз или на перекрестной ссылке, то может быть (но не обязательно) включен параметр префиксальныйКонтекст. Административный орган любого CAC решит, которые из ссылочных знаний (если таковые имеются) могут быть возвращены таким способом (остальные могут быть, например, засекречены в данном CAC).

РАЗДЕЛ 5 — *Процедуры распределенных операций*

16 Введение

16.1 *Предмет рассмотрения и ограничения*

В настоящем параграфе специфицируются процедуры распределенных операций Справочника, выполняемых CAC. Каждый CAC индивидуально выполняет описываемые ниже процедуры: совокупное действие всех CAC образует полный состав служб, обеспечиваемых Справочником его пользователям.

Описание процедур, выполняемых одним отдельным CAC, опирается на модели § 7—10 настоящей Рекомендации.

Следует отметить, что модели и процедуры включены в настоящую Рекомендацию исключительно с иллюстративной целью и не призваны ограничить или предписать фактическую реализацию CAC.

Настоящий параграф разбит на три подпараграфа: данное введение, концептуальную модель, описывающую поведение Справочника, и введение в равно CAC-фокусированные и функционально-фокусированные модели операций, выполняемых CAC.

16.2 *Концептуальная модель*

Сложность распределенных операций Справочника порождает необходимость применения как описательной, так и графической техники при разработке концептуальной модели. Но при этом ни описательный подход, ни графические диаграммы не должны рассматриваться в качестве формальных определений распределенных операций Справочника.

16.3 *Индивидуальное и совместное функционирование CAC*

Модель рассматривает работу CAC с двух различных точек зрения, которые, вместе взятые, образуют полную картину функционирования Справочника.

a) САС-фокусированная точка зрения. При этой точке зрения совокупность процедур, обеспечивающих Справочник, описывается с позиций отдельного САС; это обеспечивает возможность дать четкую спецификацию каждой процедуры и при этом полностью учесть их внутреннее взаимодействие и общую управленческую структуру. В § 18 САС описывается с САС-фокусированной точки зрения;

b) функционально-фокусированная точка зрения. САС-фокусированная точка зрения обеспечивает подробное описание, но затрудняет понимание структуры отдельных операций, которые осуществляются несколькими САС; поэтому в § 17 используется функционально-фокусированная точка зрения, с помощью которой вводятся фазы выполнения, приложимые к каждой операции.

Для обеспечения распределенных операций Справочника каждый САС должен выполнять как действия, требующиеся для реализации самих операций, так и действия, необходимые для распределения этой реализации по нескольким САС. В § 17 выявляется разница между действиями этих двух типов. В § 18 подробно изучаются оба типа этих действий.

17 Функционирование распределенного Справочника

17.1 *Совместное обеспечение операций*

Каждый САС снабжен всеми процедурами, необходимыми для полного выполнения всех операций Справочника. В случае, если все ИДС сосредоточено в одном САС, то все операции полностью выполняются этим одним САС. В случае, если ИДС распределено между несколькими САС, выполнение типичной операции разделено таким образом, что только какая-то часть этой операции выполняется каждым из потенциально многих кооперирующихся САС.

В распределенной среде типичный САС смотрит на каждую из операций как на некоторое транзитное событие; операция возбуждается или со стороны АПС или со стороны другого САС; рассматриваемый САС осуществляет обработку объекта, а затем передает его другому САС на дальнейшую обработку.

Альтернативная точка зрения изучает тотальную обработку, которой подвергается операция в процессе ее выполнения совокупностью кооперирующихся САС. Эта точка зрения раскрывает общие фазы обработки, которые приложимы ко всем операциям.

17.2 *Фазы выполнения операций*

Каждую операцию Справочника можно рассматривать как состоящую из трех отдельных фаз:

a) фазы Разрешения имени, в течение которой имя того конкретного объекта, над которым должна быть выполнена операция, используется для выявления САС, хранящего соответствующую статью;

b) фазы Осуществления, в течение которой операция, специфицированная в конкретном запросе к Справочнику (например, чтения), фактически выполняется;

c) фазы Объединения ответов, в течение которой результаты специфицированной операции возвращаются исходному АПС. Если для взаимодействия был выбран режим сцепления, то фаза объединения ответов может вовлечь несколько САС, каждый из которых сцеплял исходный запрос или подзапрос (как это определено в § 17.3.1 "Разложение запроса") с другим САС в процессе выполнения одной или обеих из двух предшествующих фаз.

При выполнении операций **Чтение**, **Сравнение**, **Список**, **Поиск** и **МодификацияСтатьи** разрешение имени осуществляется относительно имени объекта, указанного в аргументе операции. Для случая операций **ДобавлениеСтатьи**, **УдалениеСтатьи** и **МодификацияОВИ** разрешение имени осуществляется относительно имени непосредственно предшествующего объекта (получающегося извлечением последнего ОВИ из имени, заданного в аргументе операции).

Выполнение некоторой конкретной операции может быть первоначально направлено любому из САС Справочника. Этот САС использует свои знания, возможно в сочетании с другими САС, для выполнения операции по этим трем фазам.

17.2.1 *Фаза разрешения имени*

Разрешение Имени заключается в последовательном сопоставлении каждого ОВИ в заданном имени с дугами (или вершинами) ИДС. Этот процесс начинается логически от корня дерева и распространяется вниз по ИДС. При этом, поскольку ИДС распределено между произвольным числом САС, каждый из САС может выполнить лишь некоторую долю процесса разрешения имени. Некоторый САС выполняет свою долю Разрешения имени просмотром своих локальных знаний. Этот процесс описан в § 18.6 и в сопутствующих диаграммах

(рис. 7/X.518—9/X.518). Когда САС достигает границы его именующего контекста, он узнает из хранящихся там информационных знаний, может ли быть процесс разрешения имени продолжен другим САС, или имя содержит какую-то ошибку.

17.2.2 *Фаза осуществления*

После окончания фазы разрешения имени осуществляется фактически затребованная операция (например, чтение или поиск).

Операции, осуществляемые над одной статьей — **Чтение**, **Сравнение**, **ДобавлениеСтатьи**, **УдалениеСтатьи**, **МодификацияСтатьи** и **МодификацияОВИ** — могут быть выполнены полностью тем САС, в котором была локализована статья. **ДобавлениеСтатьи**, **УдалениеСтатьи** и **МодификацияОВИ** могут повлиять на знания более чем одного САС. См. § 18.7.1.

Операции, осуществляемые над несколькими статьями — **Список** и **Поиск** — требуют локализации статей, расположенных ниже целевого объекта, которые могут или не могут находиться в том же САС. Если не все они содержатся в том же САС, то операции должны быть направлены к тем САС, которые специфицированы в ссылках вниз; эти САС должны завершить процесс выполнения.

17.2.3 *Фаза объединения ответов*

Фаза объединения ответов начинается с момента, когда некоторые результаты фазы осуществления операции становятся доступны.

В тех случаях, когда операция затронула только одну статью, результат операции может быть просто возвращен запрашивавшему АПС. В тех же случаях, когда операция затронула несколько статей в нескольких САС, результаты должны быть объединены.

Допустимыми ответами, возвращаемыми реквестору после их объединения, являются:

a) полный результат операции;

b) результат, который не является полным, потому что остаются не обследованными некоторые части ИДС (относится только к **Список** и **Поиск**). Такой *частичный результат* может содержать ссылки на продолжения, связанные с теми частями ИДС, которые не были обследованы;

c) ошибка (частным случаем которой является отсылка);

d) если реквестором был САС, то **РезультатСцепления**.

17.3 *Управление распределенными операциями*

Аргумент каждой из абстрактных операций, о выполнении которой был запрошен САС, содержит информацию, характеризующую степень продвижения операции по мере того, как она проходит через отдельные САС Справочника. Это обеспечивает каждому САС возможность выполнения соответствующего ему аспекта запрошенной обработки и регистрации завершения этого аспекта, прежде чем он направит выполнение операции к другому САС.

САС содержит дополнительные процедуры, обеспечивающие физическое распределение операций и возможные другие потребности, возникающие из распределенности операций.

17.3.1 *Разложение запроса*

Разложение запроса является процессом, осуществляемым некоторым САС внутри себя, прежде чем он вступит в связь с другим или другими САС. Запрос разлагается на несколько подзапросов таким образом, что каждый из них должен осуществить какую-то часть исходной работы. Разложение запроса может быть осуществлено, например, в операции поиска после того, как будет найден базовый объект. После разложения каждый из подзапросов может затем быть подвергнут сцеплению или многоадресной рассылке с другими САС с целью продолжения обработки.

17.3.2 *САС как респондер на запрос*

САС, получивший запрос, может проверить продвижение этого запроса, используя для этого параметр Продвижение Операции. Этот параметр определяет, находится ли запрос все еще в фазе разрешения имени или же уже достиг фазы осуществления. Этот же параметр определяет, какую долю операции должен попытаться осуществить данный САС. Если САС не может полностью удовлетворить запрос, то он должен либо передать запрос одному или нескольким САС (используя сцепление или многоадресную рассылку), которые помогут ему выполнить запрос возвратить отсылку к другому САС или прервать запрос сообщением об ошибке.

17.3.3 *Завершение операций*

Каждый САС, инициировавший операцию или продвинувший выполнение операции другому или другим САС, должен следить за прохождением этой операции до тех пор, пока каждый из других САС возвратит результат или ошибку или пока не истечет предельный срок выполнения операции. Это требование касается всех операций, режимов их продвижениея и фаз обработки. Это обеспечивает упорядоченное закрытие распределенной операции, которая была распространена по Справочнику.

17.4 *Прочие аспекты распределенной операции*

17.4.1 *Проверка корректности запроса*

Получив запрос, САС должен прежде всего проверить его корректность, чтобы убедиться в возможности его выполнения. Обстоятельства, например образование петель в ИДС, связанное с необоснованным использованием псевдонимов или с использованием неверных знаний, могут привести к тому, что операции будут направлены к САС, неспособным выполнить их.

В простых случаях эти ошибочные обстоятельства могут быть адекватно обработаны процедурами разрешения имени, как это описано в § 18. Однако в тех случаях, когда обстоятельства приводят к образованию петель (как это описано в § 17.4.3), одного лишь разрешения имени оказывается недостаточно.

Действия, проверяющие корректность запроса, обеспечивают обнаружение петель до того, как делается какая бы то ни была попытка продвижения запроса по ошибочным данным, связанным с образованием петель. Процесс обнаружения выполняется процедурой обнаружения петли, специфицированной в § 18.5.1.

Там, где применяются меры безопасности, проверка запроса устанавливает также подлинность запрашивающего САС или АПС и допустимость запроса.

17.4.2 *Информация о следе и состоянии*

Продвижение операции по Справочнику и наличие условий образования петель определяется так называемым "состоянием" операции, где состояние определяется как состоящее из:

— имени САС, выполняющего операцию в данный момент;

— имени целевогоОбъекта, содержащегося в аргументе операции;

— продвиженияОперации, содержащегося в аргументе операции, как это определено в § 12.5.

Помимо текущего состояния операции, САС должен знать все предыдущие состояния данной операции. Они регистрируются в **информацииСледа** и передаются вместе с операцией.

Аргумент **информацииСледа** является основой стратегии обнаружения/исключения петель, как это специфицировано в § 17.4.3.

17.4.3 *Образование петель*

В рамках контекста некоторой конкретной операции Справочника образование петли возникает каждый раз, когда операция возвращается в одно из предшествующих состояний (определенных выше). Образование петель контролируется с помощью аргумента **информацияСледа**. Для обработки петель применяются две стратегии. При *обнаружении петли* САС проверяет, возникла ли петля во входящей операции, и, если это так, возвращает ошибку. Для *предотвращения петель* САС определяет, вызовет ли продвижение операции возникновение петли.

17.4.4 *Параметры управления службой*

Некоторые параметры управления службой требуют специального рассмотрения для случая распределенной среды, для того чтобы обеспечить выполнение операции так, как это было специфицировано в запросе.

a) **запрещеноСцепление.** САС обращается к этому параметру управления службой для определения режима продвижения операции. Если он установлен, то САС всегда использует метод отсылки. Если же он не установлен, решение того, использовать ли отсылку или сцепление, принимает сам САС, исходя из своих возможностей.

b) **ограничениеВремени.** САС должен учитывать значение этого параметра управления службой, чтобы быть уверенным, что им не нарушен лимит отведенного времени. Тот САС, к которому к первому обращается АПС с запросом на выполнение операции, выясняет у АПС и выражает в секундах параметр **ограничениеВремени**; значение этого параметра задает тот промежуток времени, в течение которого должна быть завершена операция. Если используется сцепление, то **ограничениеВремени** включается в аргумент сцепления для передачи следующему (щим) САС. В этом случае одно и то же значение используется в качестве лимита времени для каждого сцепленного запроса. Этому параметру придается значение того времени от Гринвича, к которому операция должна быть завершена,

чтобы было соблюдено первоначально установленное требование. САС, получивший аргумент сцепления, в котором специфицировано ограничениеВремени, должен выполнить это ограничение.

c) ограничениеДлины. САС должен учитывать значение этого параметра управления службой, чтобы быть уверенным, что им не будет нарушен лимит на длину списка, содержащегося в возвращаемом результате. Этот лимит, включенный в общий аргумент начального запроса, передается без изменения по мере того, как осуществляется сцепление или многоадресная рассылка запроса. Если требуется совершить разложение запроса, то то же значение включается в аргумент, передаваемый следующему САС. Иными словами, общий лимит используется для каждого подзапроса. После возвращения ответов САС-инициатор запроса разбирает поступившее множество ответов и прилагает к ним заданный ограничитель, чтобы убедиться, что возвращаемое число ответов не превосходит заданного. Если лимит был превышен, то в ответе это отмечается.

d) приоритет. При всех режимах продвижения запроса каждый САС ответственен за такое упорядочение поступивших к нему разных запросов, при котором выполняется требование этого параметра управления службой, в случае его наличия.

e) локальнаяОбластьПрименения. Операция должна выполняться в пределах локально определенной области применения и не может быть распространена за ее пределы никаким из режимов продвижения запросов.

f) областьПримененияОтсылок. Если в ответ на операции Список или Поиск САС возвращает отсылки или частичный результат, то вложенная СсылкаНаПродолжение должна быть в пределах запрошенной области применения.

Должны быть соблюдены требования, содержащиеся во всех остальных параметрах управления службой, но их использование не требует никакого дополнительного рассмотрения, связанного со спецификой распределенной среды.

17.4.5 *Расширения*

17.4.5.1 Если в течение фазы разрешения имени САС обнаруживает расширенную абстрактную-операцию и устанавливает, что абстрактная-операция подлежит сцеплению с другими САС, то он должен включить без всяких изменений эти расширения в сцепление абстрактной-операции.

Примечание. — Административный орган может принять решение, согласно которому будет возвращаться ошибкаСлужбы с установленным значением нерасположенВыполнить, если он считает нецелесообразным продвигать эти расширения.

17.4.5.2 Если в течение фазы осуществления операции САС обнаруживает расширение, то возможны два решения. Если расширение не является критическим, САС проигнорирует расширение. Но, если расширение критическое, САС возвратит ошибкуСлужбы с указанием недоступноеКритическоеРасширение.

Критическое расширение операции, связанной с многими объектами, может привести как к выработке результата, так и к выработке ошибки с указанной спецификацией. САС, осуществляющий объединение полученных результатов и ошибок, аннулирует эти ошибки и установит компоненту недоступноеКритическоеРасширение в квалификатореЧастичногоРезультата, как это описано в § 10.1.1 Рекомендации X.511.

17.4.6 *Переименование псевдонима*

Переименование псевдонима является процессом создания нового целевого объекта. Делается это заменой части выделенного имени статьи псевдонима, входящей в имя исходного целевого объекта, на значение атрибута "Имя объекта Псевдонима" статьи псевдонима. Переименование псевдонима не влияет на имя объекта операции.

17.5 *Аутентификация распределенных операций*

Пользователи Справочником и Административный орган, осуществляющий управление Справочником, могут договориться о необходимости аутентификации операций Справочника. Характер аутентификации для каждой конкретной операции будет зависеть от текущей стратегии безопасности.

Возможны два набора процедур аутентификации, в своей совокупности обеспечивающих широкий диапазон требований к аутентификации. Один набор процедур обеспечивается Привязыванием; эти процедуры способствуют аутентификации между двумя прикладными элементами Справочника в целях установления ассоциации. Процедуры Привязывания содержат спектр средств аутентификации от простого обмена идентификаторами до строгой аутентификации.

Помимо средств аутентификации между двумя равноуровневыми элементами, обеспечиваемых Привязыванием, Справочник обладает еще и дополнительными процедурами аутентификации отдельных операций. Определены два различных набора процедур аутентификации Справочника. Один набор обеспечивает службы аутентификации инициатора запроса; при этом службы предоставляются тому САС, к которому был обращен первоначальный запрос на службу. Второй набор обеспечивает службы аутентификации любых возвращаемых результатов; при этом службы предоставляются инициатору запроса.

Для аутентификации источника запроса определены две процедуры: одна, называемая аутентификацией на основе проверки подлинности, опирается на простой обмен свидетельством о подлинности, а другая, называемая аутентификацией на основе подписи, опирается на технику цифрового подписывания. Первый из этих методов имеет чисто рудиментарный характер, поскольку обмен свидетельствами о подлинности опирается на обмен выделенными именами, передающимися открыто.

Для аутентификации результатов предусмотрена процедура аутентификации одного результата, опирающаяся на технику цифрового подписывания. В силу того что в общем случае объединение результатов достаточно сложно, более простая аутентификация, опирающаяся на проверку подлинности, для этого случая не предусмотрена.

Аутентификация ответов-ошибок этими процедурами не обеспечивается.

Службы, описанные выше, должны рассматриваться как добавка к службам, обеспечиваемым Привязыванием. Предполагается, что процедуры Привязывания были успешно использованы до аутентификации операций Справочника.

Процедуры, используемые САС для аутентификации источника и результатов, описаны в § 18.9.

18 Функционирование САС

18.1 *Введение*

В ответ на каждую операцию, запрошенную реквестором (например, САС или АПС), тот САС, который выполняет этот запрос, должен функционировать в соответствии со строго определенными процедурами таким образом, чтобы соответствующие детерминированные результаты были возвращены. В настоящем параграфе специфицируется допустимое функционирование САС, что достигается моделированием САС в терминах процесса, реализующего специальный набор процедур. Важно понять, что САС должен соответствовать только тому, как его функционирование, порожденное этими процедурами, обозревается извне САС, но не самим процедурам.

18.2 *Общее описание функционирования САС*

Функционирование распределенного Справочника в целом слагается из функционирования кооперирующихся САС, образующих Справочник.

На рис. 6/X.518 показана внутренняя картина функционирования САС.

Основной управляющей процедурой САС является Диспетчер операций. Он проводит операцию через три фазы обработки, описанные в § 17.2.

Нижеследующие процедуры поддерживают работу Диспетчера операций: Разрешение имени, Отыскание именующего контекста, Локальное разрешение имени, Осуществление, Одноэлементное осуществление, Многоэлементное осуществление и Объединение результатов. Взаимосвязь между этими процедурами отображена на рис. 6/X.518.

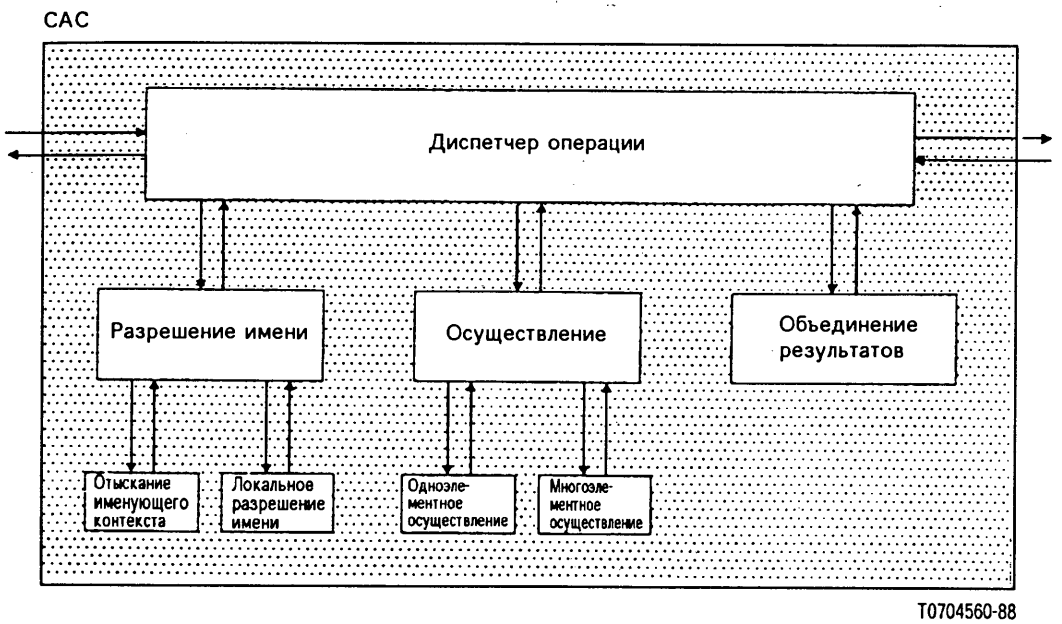


РИСУНОК 6/X.518

Функционирование CAC — вид изнутри

18.2.1 *Диспетчер операций*

Впервые получив операцию, Диспетчер операций удостоверяется в ее действительности, проверяя на наличие петель или нарушения аутентификации. Если ни того, ни другого не обнаружено, то Диспетчер операций вызывает процедуру Разрешение имени, которая возвращает либо индикатор Найден, либо Ссылку, либо индикатор ошибки. Ссылки обрабатываются либо Отсылкой, либо Сцеплением/Многоадресной рассылкой; индикатор Найден — вызовом процедуры Осуществление, которая фактически осуществляет запрошенную операцию. Возвращенные результаты, как внутренние, так и внешние, сливаются процедурой Объединения результатов и, если ошибок нет, возвращаются запросившему АПС или CAC.

18.2.2 *Разрешение имени*

Разрешение имени вызывает процедуру Отыскания именующего контекста. Если возвращенный контекст является локальным, то вызывается Локальное разрешение имени. В противном случае Разрешение возвращает "ссылку" или "ошибку" и прекращает свою работу. Если Локальное разрешение имени обнаруживает псевдоним, то выполняется переименование (если оно допустимо) и Разрешение имени повторяет весь анализ с начала. В противном случае Локальное разрешение имени возвращает либо индикатор Найден, либо ошибку, либо Отсылку; этот результат возвращается назад к Диспетчеру операций.

18.2.3 *Отыскание именующего контекста*

Процедура Отыскания именующего контекста пытается сопоставить потенциальное имя с префиксальными контекстами. Если ни для одного из них сопоставление не дает положительного результата, то Отыскание именующего контекста пытается выявить перекрестные ссылки или ссылки вверх. Если же сопоставление дало положительный результат, то Отыскание именующего контекста возвращает перекрестную ссылку, ведущую вниз по ИДС, или индикацию того, что подходящий именующий контекст найден локально, и устанавливает значение параметра **ФазаРазрешенияИмени**, равное "продолжается".

18.2.4 *Локальное разрешение имени*

Процедура Локального разрешения имени пытается внутренне сопоставить ОВИ, содержащиеся в потенциальном имени, пока она сможет вернуть индикатор Найден. Если внутреннее сопоставление не дает положительного результата для всех ОВИ, то процедура пытается идентифицировать сначала специфицированную, а потом уже неспецифицированную ссылку вниз и возвращает эти ссылки процедуре Разрешения имени. Если обнаруживается псевдоним и если параметры управления службой допускают переименование, то возвращается индикатор переименования псевдонима. Если переименование не разрешено, то индикатор Найден возвращается в том и только в том случае, если все ОВИ совпали к моменту, как был обнаружен псевдоним; в противном случае возвращается **ОшибкаИмени**.

18.2.5 *Осуществление*

Процедура Осуществления выполняет фактически запрошенную операцию Справочника над целевым объектом. В зависимости от типа операций вызываются либо процедура Одноэлементного осуществления, либо процедура Многоэлементного осуществления.

18.2.6 *Одноэлементное осуществление*

Процедура Одноэлементного осуществления вызывается для операций **Чтение**, **Сравнение**, **ДобавлениеСтатьи**, **УдалениеСтатьи**, **МодификацияСтатьи** и **МодификацияОВИ**. Фактическая выборка, проверка или изменение атрибутов осуществляются именно этими процедурами.

18.2 7 *Многоэлементное осуществление*

Процедура Многоэлементного осуществления вызывается для операций **Поиск** и **Список**, чтобы осуществить проверку фильтров, выборку результатов и при необходимости диспетчеризацию подзапросов.

18.2.8 *Объединение результатов*

Процедура Объединения результатов сливает результаты и ошибки, полученные от других САС, с локально выбранными результатами.

18.3 *Специфичные операции*

Каждая операция попадает в один из трех классов операций (в каждом случае и операция и ее Сцепленный двойник оказываются в одном классе).

a) Одноэлементные операции: **Чтение**, **Сравнение**, **ДобавлениеСтатьи**, **МодификацияСтатьи**, **МодификацияОВИ**, **УдалениеСтатьи**;

b) Многоэлементные операции: **Список** и **Поиск**;

c) Операция Отказа, то есть **Отказ**.

Обработка этих классов описывается в § 18.3.1—18.3.3 соответственно. В силу большего подобия в том, как функционирует САС при выполнении операций портов служб и как он функционирует при выполнении сцепленного двойника операций портов сцепленных служб, приводится одно описание для обоих случаев; исключения из этого правила оговариваются.

18.3.1 *Одноэлементные операции*

Одноэлементными операциями являются те операции, которые воздействуют на одну статью и которые, следовательно, могут быть полностью выполнены внутри САС, содержащего тот элемент, над которым должна быть выполнена операция. Такие операции могут быть единообразно описаны в виде приведенной ниже цепочки событий:

1) Активизировать Диспетчера операций.

2) Выполнить процедуру Разрешения имени для локализации того объекта, имя которого было специфицировано в качестве аргумента операции.

3) Выполнить процедуру Одноэлементного осуществления.

4) Параметры управления службой, такие как Ограничение времени, должны проверяться в процессе выполнения операции, чтобы обеспечить соблюдение ограничений, специфицированных пользователями.

5) Возвратить результаты тому АПС или тому САС, которые продвинули запрос.

18.3.2 *Многоэлементные операции*

Многоэлементными операциями являются операции, воздействующие на несколько статей, которые могут быть, а могут и не быть расположенными совместно в пределах одного САС. Таким образом, такие операции могут потребовать совместных усилий нескольких САС для локализации и обработки всех статей, затрагиваемых запрошенной операцией. Общее функционирование таких операций может быть суммировано следующим образом:

1) Активизировать Диспетчера операций.

2) Выполнить процедуру Разрешения имени для локализации того элемента, имя которого было специфицировано в качестве аргумента операции.

3) После того как целевой объект операции был локализован, выполнить процедуры Многоэлементного осуществления.

4) Если в одной из процедур Многоэлементного осуществления имело место разложение запроса, а подзапросы подверглись сцеплению/многоадресной рассылке, то Диспетчер операции сохраняет локальные результаты, ждет результатов сцепления и активизирует процедуру Объединения результатов.

5) Параметры управления службой, такие как Ограничение времени и Ограничение длины, должны проверяться в процессе выполнения операции, чтобы обеспечить соблюдение ограничений, специфицированных в общем аргументе.

6) Возвратить результаты или ошибки тому САС или тому АПС, которые продвинули запрос.

18.3.3 *Операция отказа*

Получив операцию отказа, САС определяет, может ли он прекратить специфицированную операцию и, если да, то прекращает ее и возвращает результат (операция, на которую поступил Отказ, возвращает ошибку **Отказано**). Если же САС не может прекратить операцию, то он возвращает ошибку **НевыполненныйОтказ**.

Ниже специфицируется процедура, свойственная операции Отказа.

1) Локализовать операцию, идентификатор возбуждения которой специфицирован в качестве аргумента операции Отказа.

2) Если это необходимо, то составить запрос(ы) другим САС с требующимся ид-вызова на отказ от еще невыполненных операций, выполняемых в результате сцепления/многоадресной рассылки.

3) Возможен вариант, когда операция Отказа выполняется локально таким образом, как это определено в Рекомендации X.511.

4) Возвратить результаты или ошибки тому САС или тому АПС, который продвинул запрос.

18.4 *Диспетчер операций*

18.4.1 *Введение*

Диспетчер операций использует процедуру Разрешения имени, описываемую в § 18.6 настоящей Рекомендации, и все взаимодействия (между двумя САС или между САС и АПС), необходимые для локализации целевых статей в распределенном Справочнике. На рис. 7/X.518 приводится детальная диаграмма, описывающая Диспетчера операций. Алгоритм работы подытожен ниже.

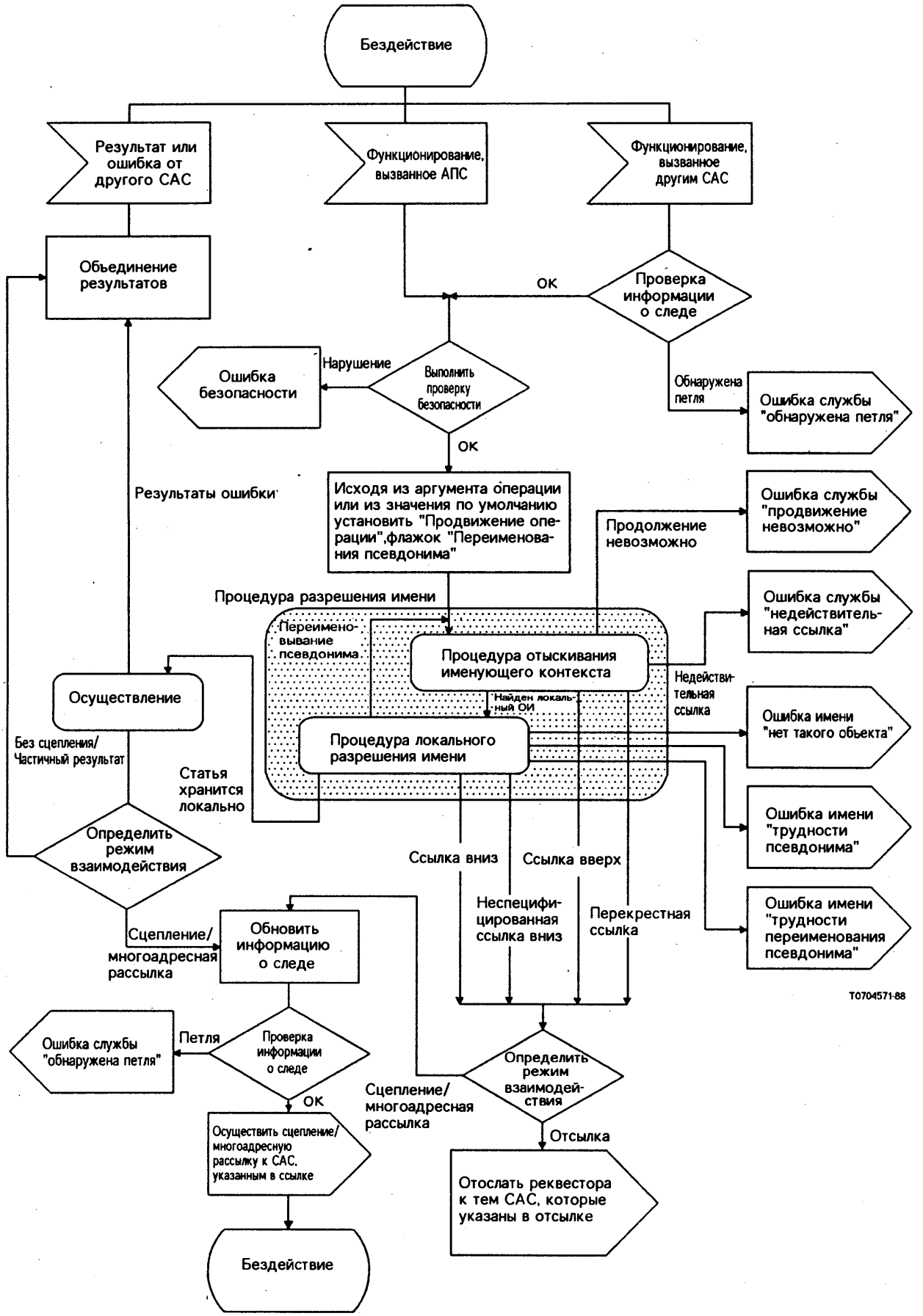


РИСУНОК 7/X.518

Диспетчер операций

18.4.2 *Неявные действия*

18.4.2.1 *Безопасность*

Следует отметить, что хотя проверка подписей не включена явно в этот алгоритм, тем не менее это действие всегда осуществляется в качестве первого шага, когда подписанные операции, результат или ошибка прибывают к некоторому САС.

Примечание. — Это не касается встроенных подписей.

Если подпись недействительна или она отсутствует в том случае, когда требуется ее наличие, то возвращается **ошибкаБезопасности**. Любая деятельность по выполнению операции прекращается, и Диспетчер операции переходит в состояние "бездействие".

В равной мере подписывание результата операции, если таковое требуется, является неявным заключительным шагом перед отсылкой результата.

18.4.2.2 *ПараметрыСлужбы*

ПараметрыСлужбы явно не упоминаются, тем не менее налагаемые ими требования должны выполняться. Например, проверка **ограниченияВремени** в поступившей операции и проверка **длиныСписка** перед отсылкой результата являются обязательными. Они описаны в § 17.4.4.

18.4.2.3 *Информация о следе*

ИнформацияСледа обновляется всегда, когда состояние прибывает к САС, и это обновление должно выполняться перед включением ее в **аргументыСцепления**. Однако это явно не отмечается в нижеследующем тексте.

18.4.3 *Аргументы*

Аргументы Сцепления конкретной операции.

18.4.4 *Результаты*

Результат Сцепления конкретной операции.

18.4.5 *Ошибки*

Любая ошибка, определенная в настоящей Рекомендации.

18.4.6 *Алгоритм*

1) Прибытие операции.
Если операция исходит от другого САС, то она содержит аргументы сцепления, включая **продвижениеОперации**, **псевдонимПереименован**, **оВИОбъектаПсевдонима**, **целевойОбъект** Имя и **информацииСледа**, равно как и аргументы исходной операции.

Если операция исходит от АПС, то она не содержит признака **псевдонимПереименован**, которому тем самым присваивается значение **FALSE**. Аргумент не содержит **информацииСледа**, что исключает проверку на образование петли. Требуется установить имя **целевойОбъект** Имя равным имени целевого объекта операции (см. § 17.2). Остальные аргументы сцепления устанавливаются в соответствии с параметрами функционирования ПДС. **ПунктуПорождения** присваивается имя пользователя.

2) Если операция поступила от САС, надо проверить информацию о следе на наличие петли (активизировать Обнаружение петли). Если обнаруживается наличие петли, то надо возвратить **ОшибкуСлужбы**, в которой указать причину **обнаруженаПетля**. После этого операция прекращается.

3) Осуществить проверку безопасности по отношению к операции (поступившей равно от АПС или САС). Если имеются нарушения, то возвращается **ошибкаБезопасности**. В противном случае надо установить **продвижениеОперации** и **псевдонимПереименован** в соответствии с аргументом операции или по умолчанию.

4) Выполнить процедуру Разрешение имени.
Процедура Разрешения имени возвратит либо индикатор "найдено", либо удаленную ссылку, либо индикатор ошибки.

5) Может возникнуть одна из следующих ошибок:
ОшибкаСлужбы (**ПродвижениеНевозможно**) — в случае, если САС устанавливает, что к нему была направлена операция, требующая информации, которой у него нет.

ОшибкаСлужбы (**НедействительнаяСсылка**) — в случае, если САС устанавливает, что были использованы недействительные знания о ссылке;

ОшибкаИмени (**нетТакогоОбъекта**) — если потенциальное имя, специфицированное в запросе на операцию, является недействительным;

ОшибкаИмени (**трудностьПсевдонима**) — если был переименован псевдоним, которому не соответствовал никакой объект;

ОшибкаИмени (**трудностьПереименованияПсевдонима**) — если был обнаружен псевдоним в ситуации, в которой он недопустим.

Получив любую из этих ошибок, Диспетчер операций прекращает свою работу и возвращает сообщение об ошибке тому САС или АПС, которые выдали запрос на распределенную операцию.

6) Если возвращено "Найдено", то надо активизировать процедуру Осуществления.

7) Если возвращена удаленная ссылка (как от процедуры Разрешения имени, так и от процедуры Осуществления), то этой ссылкой может быть любая из нижеследующих: перекрестная ссылка, ссылка вверх, ссылка вниз и неспецифицированная ссылка вниз.

Поступление любой из этих ссылок означает, что Разрешение имени или Осуществление не может быть завершено в пределах данного САС, но должно вовлечь того САС, который указан в ссылке.

Диспетчер операции должен определить режим: сцепление или отсылка.

8) Если для взаимодействия был выбран режим отсылки, то, в зависимости от **областьПримененияОтсылок**, исходному САС или АПС будет возвращена либо информация, содержащаяся в возвращенной ссылке, либо **ОшибкаСлужбы** **внеОбластиПрименения**.

Примечание. — Если **возвратитьПерекрССылки** имеет значение TRUE и ссылка не является неспецифицированной ссылкой вниз или ссылкой вверх и, кроме того, Административный орган согласен обеспечить знаниями, то в отсылке может быть установлен префиксальный контекст.

9) Если для взаимодействия был выбран режим сцепления, то операция продвигается к САС, специфицированному в ссылке. Для случая неспецифицированной ссылки вниз операция должна быть продвинута к каждому САС, имя которого было достигнуто как часть неспецифицированной ссылки вниз. Это продвижение может быть осуществлено либо многоадресной рассылкой, либо последовательным сцеплением операции.

10) Надо выполнить Исключение петель для каждой операции, подлежащей передаче. Если обнаруживается, что исключение петель является не применимым или что петель нет, то надо присвоить значения аргументам сцепления, включая обновление значения **информацияСледа** и переслать операции.

Если никакая операция не была передана (в силу трудностей с петлями), то надо возвратить **ошибкуСлужбы** с указанием **обнаруженаПетля** и прекратить обработку этой операции.

Примечание. — Если одну из подопераций пришлось прервать из-за исключения петли на этом шаге, то на местном уровне должен решиться вопрос: прерывать ли выполнение всей операции и, если да, то возвратить ошибку. Если выбран этот случай, то надо возвратить **ошибкаСлужбы** (с указанием **обнаруженаПетля**) и прекратить процесс.

11) Надо дождаться ответов, после чего выполнить процедуру объединения ответов.

18.5 *Образование петель*

В контексте какой-либо конкретной операции Справочника петля возникает в любой момент, в который операция возвращается в предшествующее состояние (как определено в § 17.4.2). Это не означает, что операция не может быть обработана некоторым конкретным САС несколько раз. Но это означает, что САС не будет обрабатывать несколько раз ту же операцию при том же состоянии.

Образование петель управляется с помощью аргумента **информацияСледа**, как это описано в § 12.6. Определены две стратегии определения петель: обнаружения петель и исключения петель, описываемые в § 18.5.1 и § 18.5.2 соответственно.

18.5.1 *Обнаружение петель*

При стратегии обнаружения петель САС, получая входящую операцию, проверяет, встречается ли текущее состояние в последовательности предшествующих состояний, зарегистрированных в аргументе информация-Следа данной операции. Если да, то это означает, что в выполнении операции образуется петля. Должна быть возвращена ошибкаСлужбы (с указанием обнаруженаПетля). В противном случае САС продолжает обработку операции в соответствии с процедурами, специфицированными в § 18.4.

18.5.2 *Исключение петель*

При стратегии исключения петель САС непосредственно перед продвижением операции другому САС (как часть сцепления, многоадресной рассылки или процедуры разложения запроса) выясняет, совпадет ли последующее состояние операции (если оно может быть установлено) с одним из предшествующих состояний, зарегистрированных в аргументе информация-следа исходной-входящей операции. Последующее состояние является тем значением элементаСледа, который будет добавлен к ИнформацииСледа получающим САС.

В случае, если исходная входная операция направлялась в порт службы (а не в порт сцепления службы), не будет никакой информации о следе и процедура исключения петель окажется неприменимой.

Если последующее состояние операции известно и обнаруживается в информацииСледа, то операция, если она будет возбуждена, образует петлю. В этих условиях ответом на исходную операцию будет ошибкаСлужбы (с указанием обнаруженаПетля).

18.6 *Процедура разрешения имени*

В настоящем пункте приводится детальное описание процедуры Разрешения имени, ее входных и выходных параметров и условия возможных ошибок. На рис. 7/X.518 эта процедура изображена укрупненно в виде диаграммы. Процедура Разрешения имени вызывает две процедуры, являющиеся ее компонентами.

1) Отыскание Именующего Контекста (рис. 8/X.518).

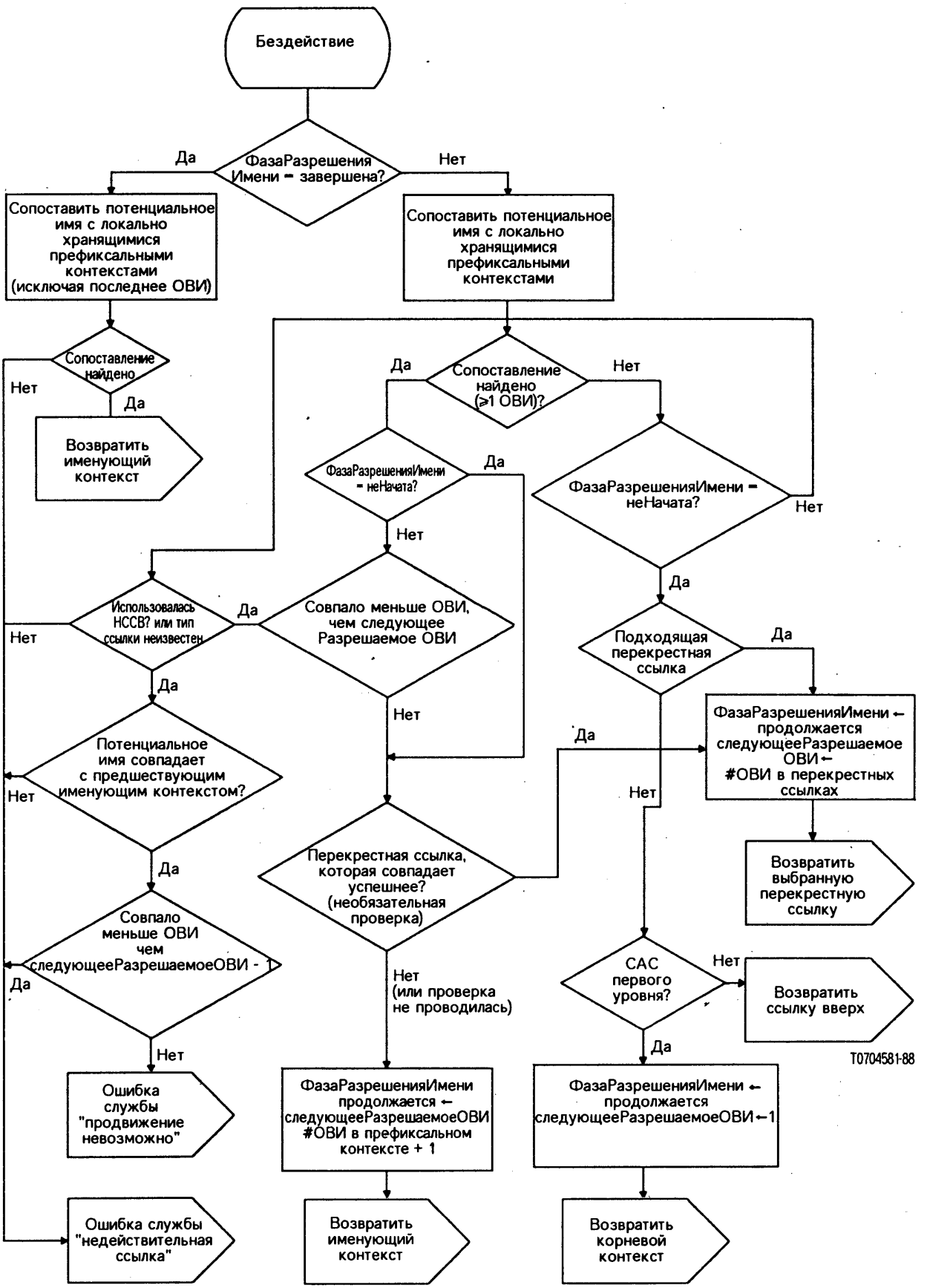


РИСУНОК 8/X.518

Отыскание Именующего Контекста

2) Локальное Разрешение Имени (рис. 9/X.518).

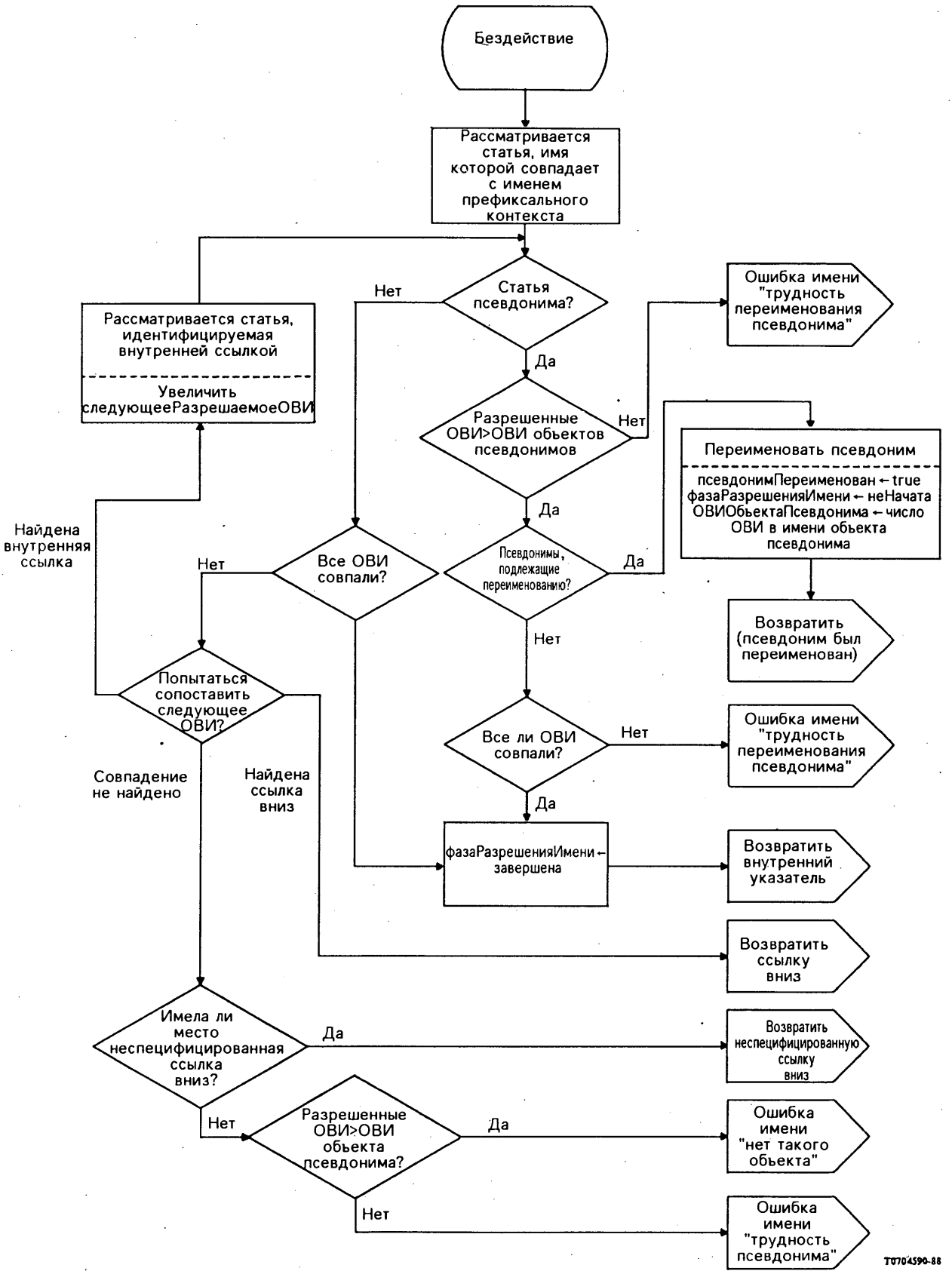


РИСУНОК 9/X.518

Локальное Разрешение Имени

Процедура Разрешения имени направляет обратно Диспетчеру операции результаты двух указанных компонентных процедур за исключением двух нижеследующих случаев. Первым случаем является тот, при котором процедура Отыскания Именующего контекста выделяет подходящий контекст, который подлежит дальнейшему анализу; в этом случае процедура возвращает локальный именующий контекст. Вторым случаем является тот, при котором процедура Локального Разрешения имени вырабатывает индикатор, гласящий, что ею был переименован псевдоним. В первом случае процедура Разрешения имени вызывает процедуру Локального Разрешения имени. Во втором случае вновь активизируется процедура Разрешения имени, но уже с новым именем целевого объекта.

18.6.1 *Аргументы*

Процедура использует следующие аргументы:

— имя целевого объекта (потенциальное имя);

— продвижение операции;

— значение параметра управления службой неПереименовыватьПсевдонимов;

— значение параметра ОВИОбъектаПсевдонима;

— значение параметра псевдонимПереименован.

18.6.2 *Результаты*

Существует два случая успешных результатов.

В первом случае будут возвращены:

— ссылки;

— продвижение операции (обновленное соответствующим образом);

— индикатор псевдонимПереименован и, возможно, ОВИОбъектаПсевдонима.

Во втором случае будут возвращены:

— индикатор того, что именующий контекст найден (совместно с локальным указателем на статью);

— продвижение операции (обновленное соответствующим образом);

— индикатор псевдонимПереименован и, возможно, ОВИОбъектаПсевдонима.

18.6.3 *Ошибки*

Может быть возвращена одна из следующих ошибок:

— ОшибкаСлужбы (продвижениеНевозможно);

— ОшибкаСлужбы (недействительнаяСсылка);

— ОшибкаИмени (трудностьПсевдонима, нетТакогоОбъекта или трудностьПереименованияПсевдонима).

18.6.4 *Процедура*

1) Активизировать процедуру Отыскания именующего контекста.

2) Ждать ответа от процедуры Отыскания именующего контекста.

3) Получить возвращаемые ответы или ошибки, то есть Найден Локальный именующий контекст, Удаленная ссылка, Ошибка Продвижение невозможно, Ошибка имени или НедействительнаяСсылка.

4) Выполнить действия, основывающиеся на возвращаемых результатах или ошибках.

   a) Если был найден локальный именующий контекст, то активизировать процедуру Локального разрешения имени. Эта процедура может возвратить одно из следующих: Найдена внутренняя ссылка, Удаленная ссылка, Псевдоним Переименован, ОшибкаИмени. Каждый из этих ответов вызывает прекращение работы процедуры Разрешения имени с выдачей соответствующего сообщения. Исключение составляет случай переименования псевдонима; действия должны быть возобновлены начиная с шага 1).

   b) Любой другой ответ должен быть возвращен назад Диспетчеру операции.

18.6.5 *Процедура Отыскания именующего контекста*

18.6.5.1 *Введение*

На рис. 8/X.518 эта процедура изображена в виде диаграммы. Ниже следует ее словесное описание. При этом предполагается, что по выходе из процедуры обязательно возвращается текущее значение Продвижения Операции.

18.6.5.2 *Аргументы*

Процедура использует нижеследующие аргументы:

— имя целевого объекта (потенциальное имя);

— продвижение операции.

18.6.5.3 *Результаты*

Существует два случая успешных результатов.

В первом случае будут возвращены:

— ссылка;

— продвижение операции (обновленное соответствующим образом).

Во втором случае будут возвращены:

— индикатор того, что подходящий именующий контекст был найден локально;

— продвижение операции (обновленное соответствующим образом).

18.6.5.4 *Ошибки*

Может быть возвращена одна из следующих ошибок:

— **ОшибкаСлужбы** (**продвижениеНевозможно**);

— **ОшибкаСлужбы** (**недействительнаяСсылка**).

18.6.5.5 *Процедура*

1) Если значением **фазыРазрешенияИмени** является **завершена**, то попытаться сопоставить потенциальное имя предшествующим префиксальным контекстам всех локально хранящихся именующих контекстов. Если совпадение не найдено, то возвратить **ошибкуСлужбы недействительнаяСсылка**.

2) Если значением **фазыРазрешенияИмени** не является **завершена**, то попытаться сопоставить префиксальные контексты последовательности из одного или более ОВИ, входящих в начальную часть потенциального имени. Чтобы было установлено совпадение, оно должно иметь место для всех ОВИ. При этом нужно использовать те префиксальные контексты, которые входят в Именующие Контексты, находящиеся под административным управлением данного САС. Если имело место несколько совпадений, то выбрать то из них, для которого число совпадений максимальное.

   Если совпадение найдено, выполнить (3).

   Если совпадение не найдено, выполнить (5).

3) Если значением **фазыРазрешенияИмени** является **неНачата**, выполнить (4). Если число тех ОВИ в начальной части заданного имени, для которых имело место совпадение, описанное в приведенном выше шаге (2), больше или равно компоненте **следующееРазрешаемоеОВИ** в **продвижениеОперации**, то перейти к шагу (4); в противном случае перейти к шагу (9).

4) Значение **следующегоРазрешаемогоОВИ** установить равным числу совпавших ОВИ плюс 1, а значение **фазыРазрешенияИмени** установить равным **продолжается**. Возвратить контекст. Работа этой процедуры прекращается.

   Для повышения производительности можно добавить сопоставление потенциального имени перекрестным ссылкам, имеющимся у САС. Если при сопоставлении с перекрестными ссылками число совпадений окажется большим, чем при сопоставлении с локальными префиксальными контекстами, то перейти к шагу (7).

Примечание. — В этом случае процедура Разрешения имени вызовет процедуру Локального разрешения имени.

5) Если сопоставление окончилось безуспешно, то надо проверить значение параметра фазаРазрешения Имени. Если оно равно неНачато, то перейти к шагу (6).

Если же значение равно продолжается или завершена, то перейти к шагу (9).

6) Попытаться сопоставить одно или более ОВИ в начальной части потенциального имени с префиксальными контекстами перекрестных ссылок. Если имело место несколько совпадений, то выбрать то из них, для которого число совпадений ОВИ максимальное.

7) Если совпадение с перекрестными ссылками было найдено, то установить следующееРазрешаемоеОВИ равным числу ОВИ в выбранной перекрестной ссылке. Возвратить перекрестную ссылку. Работа этой процедуры прекращается.

8) Если сопоставление с перекрестными ссылками не найдено, то проверить, являются ли данные САС системным агентом Справочника первого уровня. Если нет, то САС должен содержать ссылку вверх. Возвратить эту ссылку. Работа этой процедуры прекращается.

Если САС является системным агентом Справочника первого уровня, то установить следующееРазрешаемоеОВИ равным единице, а фазуРазрешенияИмени равной продолжается. Возвратить корневой именующий контекст и прекратить работу этой процедуры.

9) Проверить значение компонента типСсылки в АргументахСцепления. Если была использована неспецифицированная ссылка вниз или же если запрос поступил от АПС, то выполнить (10); в противном случае возвратить ОшибкуСлужбы с указанием недействительнаяСсылка и прекратить работу процедуры.

10) Сравнить начальную часть потенциального имени с префиксными контекстами (минус их последнее ОВИ) локальных именующих контекстов. Фактически это означает сравнение с именующими контекстами САС, непосредственно предшествующими данному САС.

Если совпадений нет, то возвратить ОшибкуСлужбы с указанием недействительнаяСсылка и прекратить работу процедуры.

Если совпадение найдено, но число совпавших ОВИ на единицу меньше, чем следующееРазрешаемоеОВИ, то возвратить ОшибкуСлужбы с указанием недействительнаяСсылка; в противном случае возвратить ОшибкуСлужбы с указанием продвижениеНевозможно. Работу процедуры прекратить.

18.6.6 *Локальное разрешение имени*

18.6.6.1 *Введение*

Процедура Локального разрешения имени сопоставляет ОВИ, входящие в потенциальное имя внутренним ссылочным знаниям. Она возвращает "Найдено", "Удаленная ссылка", "Переименование Псевдонима" или "Индикатор ошибки".

На рис. 9/X.518 эта процедура изображена в виде диаграммы. Ниже следует ее словесное описание.

18.6.6.2 *Аргументы*

Процедура использует следующие аргументы:

— внутренние ссылки на именующий контекст (при этом указатель задает статью, имя которой совпадает с префиксальным контекстом);

— имя целевого объекта (потенциальное имя);

— продвижение операции;

— значение параметра управления службой неПереименовыватьПсевдонимов;

— значение параметра ОВИОбъектаПсевдонима;

— значение параметра псевдонимПереименован.

18.6.6.3 *Результаты*

Существует три случая успешных результатов.

В первом случае будут возвращены:

— ссылка;

— продвижение операции (обновленное соответствующим образом).

Во втором случае будут возвращены:

— индикатор того, что статья была найдена локально;

— продвижение операции (обновленное соответствующим образом).

В третьем случае будут возвращены:

— индикатор переименования псевдонима;

— продвижение операции (установленное вновь на "не начата").

18.6.6.4 *Ошибки*

Может быть возвращена одна из следующих ошибок:

— ошибка имени.

18.6.6.5 *Процедура*

Именующий контекст, возвращенный ОтысканиемИменующего Контекста, указывает на статью корня поддерева. В случае корневого контекста статья является только нулевой статьей.

1) Если врутренняя ссылка делается на псевдоним, то выполнить шаг (7), в противном случае — шаг (2).

2) Если сопоставление со всеми ОВИ, входящими в потенциальное имя, оказалось успешным, то это означает, что найдена целевая статья. Установить значение **фазыРазрешенияИмени** равным **завершена**. Возвратить внутренний указатель и прекратить работу процедуры.

   В противном случае нужно выполнить шаг (3).

   Примечание. — Сопоставление может осуществляться либо с префиксальным контекстом как таковым, либо с префиксальным контекстом плюс последовательные ОВИ, содержащиеся во внутренних ссылках дерева знаний.

3) Если внутренняя ссылка делается на статью, подчиненную в дереве знаний текущей статье, причем эта ссылка совпадает со следующим ОВИ в заданном имени, то надо увеличить значение параметра **следующееРазрешаемоеИмя**, назвать указанную подчиненную статью текущей и вновь выполнить шаг (1) настоящей процедуры.

4) Если текущая статья имеет ссылку вниз, ОВИ которой совпадает со следующим в потенциальном имени, то возвратить эту ссылку и прекратить работу процедуры.

5) Если существуют какие-нибудь неспецифицированные ссылки вниз, подчиненные в дереве знаний текущей статье, то возвратить их в качестве ссылок и прекратить работу процедуры.

6) Если на найдены ни внутренняя ссылка, ни ссылка вниз, ни неспецифицированная ссылка вниз, то надо проверить число тех ОВИ в потенциальном имени, для которых сопоставление имело место. Если это число превосходит компоненту **ОВИОбъектаПсевдонима** в **АргументахСцепления**, то надо возвратить **ОшибкуИмени** с указанием **нетТакогоОбъекта**. Если же это число меньше указанной компоненты, то надо возвратить **ОшибкуИмени** с указанием **трудностиПсевдонима**.

7) Если число тех ОВИ в потенциальном имени, для которых сопоставление имело место, меньше или равно компоненте **ОВИОбъектаПсевдонима** в **АргументахСцепления** (если таковая вообще существует), то предшествующий переименованный псевдоним (если таковой вообще существует) в свою очередь ссылается на другой псевдоним. В этом случае надо возвратить **ОшибкуИмени** с указанием **трудностьПереименованияПсевдонима**.

8) Если компонент **ОВИОбъектаПсевдонима** отсутствует или если число совпавших ОВИ больше, чем компонент **ОВИОбъектаПсевдонима** в **АргументахСцепления**, то надо проверить параметр управления службой **неПереименовыватьПсевдонимов**. Если псевдонимы могут быть переименованы, то выполнить шаг (9), в противном случае — шаг (10).

9) Переименовать псевдонимы. Установить значение **фазыРазрешенияИмени ПродвиженияОперации** равным **неНачата**. Установить компонент **псевдонимПереименован** в **АргументахСцепления** равным **TRUE**, а **ОВИОбъектаПсевдонима** — равным числу ОВИ в атрибуте **имяОбъектаПсевдонима** статьи псевдонима. Установить значение **целевогоОбъекта** равным новому имени. Прекратить работу процедуры. (Процесс Разрешения Имени будет возобновлен.)

10) Если все ОВИ в потенциальном имени совпали, выполнить шаг (2). В противном случае возвратить **ОшибкуИмени** и **трудностьПереименованияПсевдонима**.

18.7 *Процедуры осуществления*

Описываемые процедуры осуществления содержат две категории процедур:

a) процедуры Одноэлементного осуществления;

b) процедуры Многоэлементного осуществления.

На рис. 10/X.518 изображена процедура осуществления.

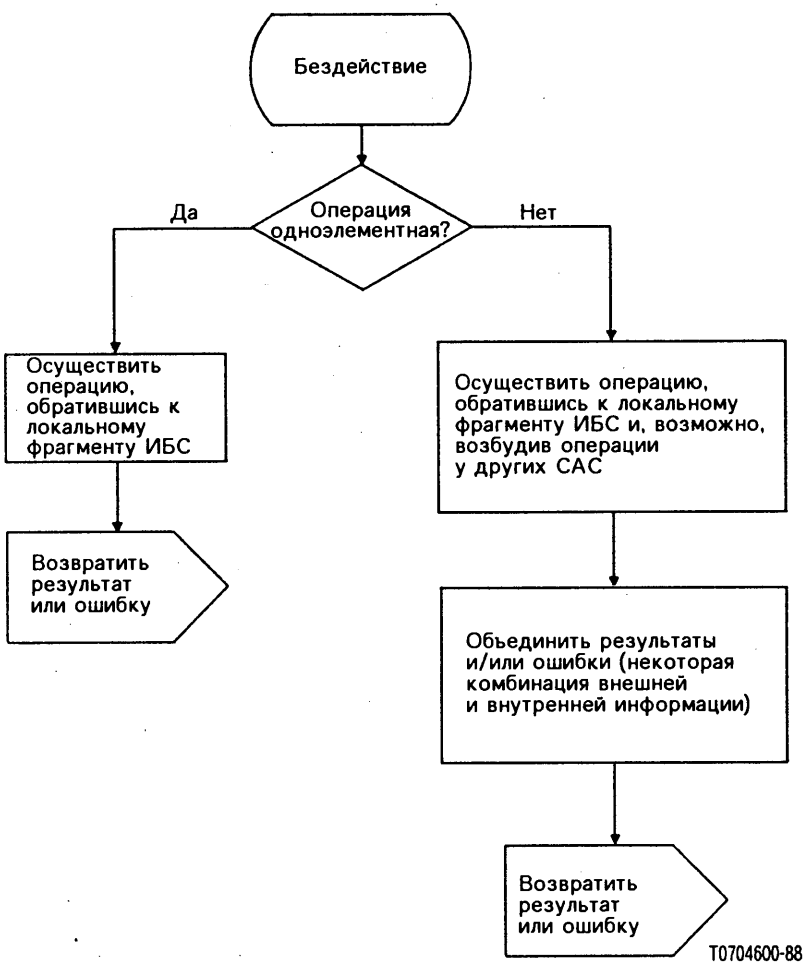


РИСУНОК 10/X.518

Осуществление и объединение результатов

18.7.1 *Процедуры Одноэлементного осуществления*

Процедуры Одноэлементного осуществления, являющиеся общими для класса операций, связанных с доступом к одному элементу, выполняются непосредственно над этим элементом и при этом инициатору возвращается либо ответ, либо ошибка.

Этими операциями являются: **Чтение**, **Сравнение**, **ДобавлениеСтатьи**, **УдалениеСтатьи**, **МодификацияСтатьи** и **МодификацияОВИ**.

Действия, осуществляемые со статьей, описаны в соответствующих параграфах Рекомендации X.511.

Операции **ДобавленияСтатьи**, **УдаленияСтатьи** и **МодификацииОВИ** воздействуют на знания. Если статья, непосредственно предшествующая данной находится у другого САС, то должны быть скорректированы внешние ссылочные знания. Метод осуществления этого выходит за пределы настоящей Рекомендации.

Вопрос о методе выбора того САС, которому придается статья, созданная операцией **ДобавленияСтатьи**, выходит за пределы настоящей Рекомендации.

Если статья, непосредственно предшествующая статье, созданной операцией **ДобавленияСтатьи** или модифицированной операцией **МодификацияОВИ**, содержит неспецифицированные ссылки вниз, то должны быть осуществлены процедуры, выходящие за пределы настоящей Рекомендации, которые препятствовали бы появлению двух статей с совпадающими выделенными именами.

Запросы, которые не могут быть удовлетворены с соблюдением этих условий, заканчиваются безуспешно с выдачей **ОшибкаОбновления** и указанием **затрагиваетНесколькоСАС**.

18.7.2 *Процедуры многоэлементного осуществления*

Процедуры многоэлементного осуществления, являющиеся общими для класса операций, связанных с доступом к нескольким объектам, специфицируются в нижеследующих подпараграфах.

Этими операциями являются **Список** и **Поиск**, а также их двойники сцепления.

18.7.2.1 *Список*

Этот параграф специфицирует процедуру, касающуюся специально операций **Список** и **СцепленныйСписок**. (Ниже под термином "Список" подразумеваются обе эти операции.)

18.7.2.1.1 *Процедура Список* (I)

Эта процедура касается запроса на операцию **Список** в тех случаях, когда компонент **фазаРазрешенияИмени** в **продвиженииОперации** имеет значение, равное **неНачата** или **продолжается**, и когда САС, выполнив Разрешение Имени, выясняет, что он владеет базовым объектом.

Базовый объект будет обозначаться буквой "e".

1) Выбрать все локально хранящиеся элементы, непосредственно следующие за *e* для сформирования локального набора результатов. Присвоить параметрам **статьяПседонима** и **изСтатьи** в **РезультатеСписка** соответствующие значения.

2) Выбрать набор ссылок вниз и неспецифицированных ссылок вниз от всех САС, у которых имеются элементы, непосредственно следующие за *e*.

3) Передать подзапрос с базовым элементом = *e* и **ПродвижениемОперации**, равным **завершено**, Диспетчеру операций, который продвинет их далее ко всем тем САС, у которых находятся элементы, непосредственно следующие за *e*.

Примечание. — Если САС содержит ссылки вниз в которых имеется индикатор того, являются ли или нет элементы, указанные в ссылках, псевдонимами, и если значение **неПользоватьсяКопией** равно **FALSE**, то в отношение этих статей указанный шаг может быть опущен. Информация о последующих элементах доступна непосредственно.

18.7.2.1.2 *Процедура Список* (II)

Эта процедура касается запроса на операцию **Список** в тех случаях, когда компонент **фазаРазрешенияИмени** в **продвиженииОперации** имеет значение, равное **завершено**.

Базовый объект будет обозначаться буквой "e".

1) Выбрать все локально хранящиеся элементы, непосредственно следующие за *e*, для сформирования локального набора результатов. Присвоить **статьеПсевдонима** и **изСтатьи** в **РезультатахСписка** соответствующие значения.

2) Передать ответы Диспетчеру операций, который продвинет их к запрашивающему САС или АПС.

18.7.2.2 *Поиск*

Этот параграф специфицирует процедуру, касающуюся специально операций **Поиска** и **Сцепленного-Поиска**. (Ниже под термином "Поиск" подразумеваются обе эти операции.)

Отметим, что в силу существующих двух обстоятельств требуются две различные процедуры. Первая процедура (§ 18.7.2.2.1) применима к случаю, когда САС, выполняющий Поиск, содержит **целевойОбъект** в качестве внутренней статьи. Вторая процедура (§ 18.7.2.2.2) применима к случаю, когда САС, выполняющий Поиск, не содержит **целевогоОбъекта**, а только объекты, подчиненные ему.

18.7.2.2.1 *Процедура Поиск* (I)

Эта процедура касается запроса на операцию Поиска в тех случаях, когда компонент **фазаРазрешенияИмени** в **продвиженииОперации** имеет значение, равное **неНачато** или **продолжается** и когда САС, выполнив Разрешение имени, выясняет, что он владеет целевым объектом.

Базовый объект будет обозначаться буквой "e".

1) если аргумент поднабор равняется базовомуОбъекту или целомуПоддереву, то применить к статье *e* аргумент фильтра, специфицированного в запросе на Поиск, для сформирования набора локальных результатов. Возвратить результаты для Объединения результатов. Если аргумент поднабор равняется базовомуОбъекту, то прекратить работу процедуры; в противном случае перейти к шагу (2).

2) Если аргумент поднабор равняется одномуУровню или целомуПоддереву, то сформировать набор E из локальнохранящихся статей, непосредственно следующих за *e*, но со следующим исключением:

   Если статьи псевдонимов подлежат переименованию, то есть параметр поискПсевдонимов равен TRUE, то все найденные статьи псевдонимов обрабатываются как в шаге (5), ниже, и не входят в эти результаты.

   Применить к набору E фильтр для получения отфильтрованного набора E′gE; возвратить этот набор E локальных результатов для Объединения результатов.

3) Остальные элементы, следующие за *e*, могут входить в другие CAC; если это так, то они будут указаны в ссылках вниз или в неспецифицированных ссылках вниз. Для каждого CAC, на которого имеется такая ссылка, надо подготовить новую операцию Поиска, в которой целевойОбъект = *e* и в которой фазаРазрешенияИмени в продвиженииОперации имеет значение завершена. Возвратить каждый подзапрос Поиск Диспетчеру операций для дальнейшего продвижения. Если в результате подзапроса в качестве ответа получена ошибка, то такой ответ должен быть проигнорирован, как если бы подзапроса вообще не было.

4) Если аргумент поднабор равняется одномуУровню, то Поиск закончен и работа процедуры прекращается.

   Если аргумент поднабор равняется целоеПоддерево, то:

   если набор E из шага (2) пуст, то это означает, что поиском было охвачено все поддерево, содержащееся в этом CAC, и, следовательно, работа процедуры прекращается;

   в противном случае процесс должен быть продолжен следующим образом:

   обозначим через *e* любую статью, входящую в E; тогда для каждого такого *e* должна быть повторена процедура Поиска но начиная с шага (2).

5) Если статьи псевдонимов подлежат переименованию, то сформировать множество D, включив в него каждую статью псевдонима, найденную в шаге (2). Для каждого d, входящего в D, надо переименовать псевдоним и сформулировать новую операцию Поиска, для которой значение фазыРазрешенияИмени равно неНачато, а целевойОбъект образован из атрибута имяОбъектаПсевдонима и старого имени целевогоОбъекта.

   Если аргумент поднабор равнялся одномуУровню, то в новом подзапросе этому аргументу надо придать значение базовыйОбъект; в противном случае надо установить его значение равным целоеПоддерево.

   Если в результате подзапроса в качестве ответа будет получена ошибка, то такой ответ должен быть проигнорирован, как если бы подзапроса вообще не было.

18.7.2.2.2 *Процедура Поиск* (II)

Эта процедура касается запроса на операцию Поиска в тех случаях, когда компонент фазаРазрешенияИмени в продвиженииОперации имеет значение, равное завершено.

Целевой объект будет обозначаться буквой "e".

Для каждого локально хранящегося e', непосредственно следующего за *e*, надо сформулировать новый запрос, для которого целевойОбъект = e'. Если аргумент поднабор равнялся одномуУровню, то надо установить его значение равным базовыйОбъект; в противном случае необходимо оставить ему его значение целоеПоддерево. Далее надо выполнить шаги от (1) до (5), описанные в § 18.7.2.2.1. Если таких последующих элементов нет, то надо возвратить ошибкуСлужбы с указанием продвижениеНевозможно.

18.8 *Процедура объединения результатов*

Обращение к этой процедуре имеет место тогда, когда в наличии имеются поступившие извне ответы и/или ошибки. Кроме того, может быть и один внутренний результат. Предполагается, что все ответы и ошибки хранятся в пределах одного CAC, пока процедура не будет закончена.

Информация извне может возникнуть в результате осуществления сцепления, многоадресной рассылки или разложения запроса.

В случае сцепления будет всего один результат или одна ошибка. В случае многоадресной рассылки может не быть ни одного результата, один результат или несколько идентичных результатов. Кроме того, могут быть и некоторые ошибки. Если результатов больше чем один, то все, кроме одного, наудачу взятого, должны быть проигнорированы. Всегда предпочтительнее возвратить результат, а не ошибку. Если результатов нет, то возвращается ошибка, но со следующими исключениями:

i) Если была возвращена недействительнаяСсылка, то ссылка получает такую пометку, и САС может либо использовать соответствующую альтернативную внешнюю ссылку для продолжения запроса, либо вернуть реквестору ошибкуИдс. (Обработка недействительных внешних ссылок выходит за пределы настоящей Рекомендации.)

ii) Для случая многоадресной рассылки ошибка продвижениеНевозможно должна быть проигнорирована; если же все ответы этого же типа, то респондеру должна быть возвращена **ОшибкаИмени нетТакогоОбъекта**. Если же возвращен хоть один ответ, то все ошибки можно проигнорировать.

iii) Для случая отсылок они не должны рассматриваться как ошибки и могут быть подвергнуты обработке.

Если объединение связано с разложением запроса, то объединение заключается в формировании одного ответа из всех поступивших.

Если в результате разложения запроса требуется объединить и ответы, и ошибки, то реквестору возвращается неполный результат.

На этом этапе САС может принять решение выделить отсылки из поступающих ответов и ошибок, подлежащих объединению. Затем он может принять решение самому проанализировать все или часть этих отсылок, в каковом случае осуществляется сцепление операций. Старый результат должен быть сохранен и впоследствии объединен с ответами и ошибками, которые возникнут в результате сцепления.

Обработка подписей, которые могут присутствовать в поступающих ответах, специфицирована в § 18.9.2, ниже.

18.9 *Процедуры распределенной аутентификации*

В настоящем пункте специфицируются процедуры, необходимые для обеспечения распределенной службы справочника по аутентификации. Эти службы, а следовательно, и соответствующие процедуры классифицируются на:

— аутентификацию пункта порождения, обеспечиваемую либо незащищенным способом (простой способ, основанный на проверке подлинности), либо безопасным способом (основанным на цифровом подписывании);

— аутентификацию результатов, защищенных аналогичным способом (также основанном на цифровом подписывании).

18.9.1 *Аутентификация пункта порождения*

18.9.1.1 *Аутентификация, основанная на проверке подлинности*

Служба аутентификации, основанная на проверке подлинности, дает возможность САСам аутентифицировать исходного реквестора информации. Это делается для активизации локальных параметров управления доступом. САС, желающие использовать эту службу, должны применить следующую процедуру:

— САС, желающий аутентифицировать запрос на ПДС, получает выделенное имя запрашивающего с помощью процедур Привязывания; делается это в процессе установления ассоциации с АПС (с АПС или с САС). Успешное завершение этих процедур ни в коей мере не предопределяет того уровня аутентификации, который в дальнейшем может понадобиться при выполнении операций, использующих эту ассоциацию;

— САС, для которого установлена ассоциация с АПС, должен ввести выделенное имя реквестора в поле "инициатор" АргументовСцепления для всех последующих сцепленных операций с другими САС;

— САС, получив сцепленную операцию, может удовлетворить, а может и не удовлетворить запроса в зависимости от определения прав доступа (локально устанавливаемый механизм). Если результат проверки оказывается неудовлетворительным, то может быть возвращена **ОшибкаБезопасности**, в которой **ТрудностиБезопасности** имеют значение **недостаточныеПраваДоступа**.

18.9.1.2 *Аутентификация, основанная на подписях*

Служба аутентификации пункта порождения запроса, основанная на подписи, позволяет САС аутентифицировать (безопасным способом) инициатора некоторой конкретной службы. В настоящем параграфе описываются процедуры, которые должны быть использованы САС для реализации этой службы.

Служба аутентификации, основанная на подписи, возбуждается АПС, используя вариант **SIGNED** запроса на службу, подписывание которой не является обязательным.

САС, получив от другого САС подписанный запрос, должен удалить эту подпись, прежде чем приступить к выполнению операции. Если процесс верификации подписи дал положительный результат, то САС продолжит выполнение операции. Если в процессе этого САС должен будет выполнить сцепление, многоадресную рассылку или разложение запроса, то набор аргументов каждой связанной с этим сцепленной операции должен быть сформирован следующим образом:

— САС формирует набор аргументов, которые при желании могут быть подписаны; набор аргументов состоит из входящего подписанного набора аргументов совместно с модифицированным **АргументомСцепления**.

В случае, если САС может добавить какую-нибудь информацию к ответу, то аутентификация пункта порождения, основанная на подписанных запросах на службу, может быть использована для проверки прав доступа к этой информации.

Если САС получает неподписанный запрос на информацию, доступ к которой требует аутентификации пункта порождения, то он возвратит **ОшибкуБезопасности**, в которой **ТрудностиБезопасности** имеют значение **требуетсяЗащита**.

18.9.2 *Аутентификация результатов*

Назначением этой службы является предоставление реквесторам (равно АПС или САС) операций Справочника возможности проверки (безопасным, основанным на технике цифрового подписывания, способом) источника ответа. Служба аутентификации результатов может быть запрошена независимо от того, будет ли использоваться аутентификация пункта порождения.

Служба аутентификации результатов инициируется использованием значения подписанный компонента **запросЗащиты**, содержащегося в наборе аргументов операций справочника; САС, получивший операцию с этим вариантом, может затем по желанию подписывать любой из последующих результатов. Вариант подписан в **требуетсяЗащита** служит для САС индикатором того, что предпочитает реквестор; САС может фактически подписывать, а может и не подписывать последующие результаты.

В случае, когда САС осуществляет сцепление, многоадресную рассылку или разложение таких запросов, у САС имеется несколько альтернативных форм результатов, посылаемых обратно реквестору, а именно:

a) возвратить реквестору составной ответ (подписанный или неподписанный);

b) возвратить реквестору набор из двух или более отдельных не объединенных ответов (подписанных или неподписанных); в этом наборе ноль или более членов набора могут быть подписаны и ноль или один — не подписаны. Если в этом наборе окажется неподписанный частичный результат, то этот результат может оказаться объединением одного или более неподписанных частичных ответов, которые были либо получены от других САС, либо выработаны данным САС, либо и этим, и другими.

ПРИЛОЖЕНИЕ А

(к Рекомендации X.518)

Описание распределенных операций на НАС.1

Данное Приложение является составной частью настоящей Рекомендации.

Данное Приложение содержит НАС.1-модуль **РаспределенныеОперации**, в который включены все НАС-1-определения типов, значений и макросов, введенные в настоящей Рекомендации.

```
РаспределенныеОперации { joint-iso-ccitt ds(5) modules(1) distributedOperations(3) }
DEFINITIONS ::=
BEGIN
```

```
EXPORTS
        ДетализацияСправочника, портСцепленногоЧтения, портСцепленногоПоиска, портСцепленнойМодификации,
        САСПривязывание, АргументСАСПривязывания,
        САСОтвязывание,
        СцепленноеЧтение, СцепленноеСравнение, СцепленныйОтказ,
        СцепленныйСписок, СцепленныйПоиск,
        СцепленноеДобавлениеСтатьи, СцепленноеУдалениеСтатьи,
        СцепленнаяМодификацияСтатьи, СцепленнаяМодификацияОВИ,
        САСОтсылка, ССылкаНаПродолжение;

IMPORTS
        СтруктураИнформации, абстрактнаяСлужба, распределенныеОперации,
        ИдентификаторыОбъектовСправочника, отдельныеТипыАтрибутов
            FROM  ПолезныеОпределения {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

        ВыделенноеИмя, Имя, ОтносительноВыделенноеИмя,
            FROM  СтруктураИнформации структураИнформации

        ид-от-сас, ид-пт-сцепленное-чтение, ид-пт-сцепленный-поиск, ид-пт-сцепленная-модификация
            FROM  ИдентификаторыОбъектовРаспределенногоСправочника, идентификаторыОбъектовРаспределенногоСправочника

        АдресВУровнеПредставлений
            FROM  ИзбранныеТипыАтрибутов избранныеТипыАтрибутов

        справочник, портЧтения, портПоиска, портМодификации,
        ПривязываниеКСправочнику,
        АргументЧтения, РезультатЧтения,
        АргументСравнения, РезультатСравнения,
        Отказ,
        АргументСписка, РезультатСписка,
        АргументПоиска, РезультатПоиска,
        АргументДобавленияСтатьи, РезультатДобавленияСтатьи,
        АргументУдаленияСтатьи, РезультатУдаленияСтатьи,
        АргументМодификацииСтатьи, РезультатМодификацииСтатьи,
        АргументМодификацииОВИ, РезультатМодификацииОВИ,
        Отказано, ОшибкаАтрибута, ОшибкаИмени, ОшибкаСлужбы, ОшибкаБезопасности, ОшибкаОбновления,
        OPTIONALLY-SIGNED, ПараметрыБезопасности
            FROM  АбстрактнаяСлужбаСправочника абстрактнаяСлужбаСправочника

-- объекты и порты --

ДетализацияСправочника  ::= REFINE справочник AS
        сас     RECURRING
            портЧтения                  [S]     VISIBLE
            портПоиска                  [S]     VISIBLE
            портМодификации             [S]     VISIBLE
            портСцепленногоЧтения               PAIRED WITH сас
            портСцепленногоПоиска               PAIRED WITH сас
            портСцепленнойМодификации           PAIRED WITH сас

сас OBJECT
            PORTS   { портЧтения            [S],
                      портПоиска            [S],
                      портМодификации       [S],
                      портСцепленногоЧтения,
                      портСцепленногоПоиска,
                      портСцепленнойМодификации}

        ::= ид-от-сас

портСцепленногоЧтения PORT
        ABSTRACT OPERATIONS {
            СцепленноеЧтение, СцепленноеСравнение,
            СцепленныйОтказ }
        ::= ид-пт-сцепленного-чтения

портСцепленногоПоиска PORT
        ABSTRACT OPERATIONS {
            СцепленныйСписок, СцепленныйПоиск }
        ::= ид-пт-сцепленного-поиска
```

```
портСцепленнойМодификацииPORT
        ABSTRACT-OPERATIONS {
            СцепленноеДобавлениеСтатьи, СцепленноеУдалениеСтатьи,
            СцепленнаяМодификацияСтатьи, СцепленнаяМодификацияОВИ }
        ::= ид-пт-сцепленной-модификации

САСПривязывание ::= ABSTRACT-BIND
        TO { сцепленноеЧтение,
            сцепленныйПоиск,
            сцепленнаяМодификация }
        привязываниеКСправочнику

САСОтвязывание ::= ABSTRACT-UNBIND
        FROM { сцепленноеЧтение,
               сцепленныйПоиск,
               сцепленнаяМодификация }

-- операции, аргументы и результаты --

СцепленноеЧтение ::=
        ABSTRACT-OPERATION
            ARGUMENT        OPTIONALLY-SIGNED   SET {
                                                АргументСцепления,
                                                [0] АргументЧтения }
            RESULT          OPTIONALLY-SIGNED   SET {
                                                РезультатСцепления,
                                                [0] РезультатЧтения }
            ERRORS {
                СасОтсылка, Отказано, ОшибкаАтрибута, ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности }

СцепленноеСравнение ::=
        ABSTRACT-OPERATION
            ARGUMENT        OPTIONALLY-SIGNED   SET {
                                                АргументСцепления,
                                                [0] АргументСравнения }
            RESULT          OPTIONALLY-SIGNED   SET {
                                                РезультатСцепления,
                                                [0] РезультатСравнения }
            ERRORS {
                СасОтсылка, Отказано, ОшибкаАтрибута, ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности }

СцепленныйОтказ ::= Отказ
СцепленныйСписок ::=
        ABSTRACT-OPERATION
            ARGUMENT        OPTIONALLY-SIGNED   SET {
                                                АргументСцепления,
                                                [0] АргументСписка }
            RESULT          OPTIONALLY-SIGNED   SET {
                                                РезультатСцепления,
                                                [0] РезультатСписка }
            ERRORS {
                СасОтсылка, Отказано, ОшибкаАтрибута, ошибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности

СцепленныйПоиск ::=
        ABSTRACT-OPERATION
            ARGUMENT        OPTIONALLY-SIGNED   SET {
                                                АргументСцепления,
                                                [0] АргументПоиска }
            RESULT          OPTIONALLY-SIGNED   SET {
                                                РезультатСцепления,
                                                [0] РезультатПоиска }
            ERRORS {
                СасОтсылка, Отказано, ОшибкаАтрибута, ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности }
```

```
СцепленноеДобавлениеСтатьи ::=
        ABSTRACT-OPERATION
            ARGUMENT            OPTIONALLY-SIGNED       SET{
                                                        АргументСцепления,
                                                        [0]АргументДобавленияСтатьи }
            RESULT              OPTIONALLY-SIGNED       SET{
                                                        РезультатСцепления,
                                                        [0]РезультатДобавленияСтатьи}
            ERRORS{
                CacOтсылка, Отказано, ОшибкаАтрибута, ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности, ОшибкаОбновления}

СцепленноеУдалениеСтатьи ::=
        ABSTRACT-OPERATION
            ARGUMENT            OPTIONALLY-SIGNED       SET{
                                                        АргументСцепления,
                                                        [0]АргументУдаленияСтатьи}
            RESULT              OPTIONALLY-SIGNED       SET{
                                                        РезультатСцепления,
                                                        [0]РезультатУдаленияСтатьи}
            ERRORS{
                CacOтсылка, Отказано,ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности, ОшибкаОбновления}
СцепленнаяМодификацияСтатьи ::=
        ABSTRACT-OPERATION
            ARGUMENT            OPTIONALLY-SIGNED       SET{
                                                        АргументСцепления,
                                                        [0]АргументМодификацииСтатьи}
        RESULT                  OPTIONALLY-SIGNED       SET{
                                                        РезультатСцепления,
                                                        [0]РезультатМодификацииСтатьи}
            ERRORS{
                CacOтсылка, Отказано, ОшибкаАтрибута, ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности, ОшибкаОбновления}

СцепленнаяМодификацияОВИ ::=
        ABSTRACT-OPERATION
            ARGUMENT            OPTIONALLY-SIGNED       SET{
                                                        АргументСцепления,
                                                        [0]АргументМодификацииОВИ}
            RESULT              OPTIONALLY-SIGNED       SET{
                                                        РезультатСцепления,
                                                        [0]РезультатМодификацииОВИ}
            ERRORS{
                CacOтсылка, Отказано, ОшибкаИмени,
                ОшибкаСлужбы, ОшибкаБезопасности, ОшибкаОбновления}

-- ошибки и параметры --

CACOтсылка ::=
        ABSTRACT-ERROR
            PARAMETER SET{
                [0]ССылкаНаПродолжение,
                префиксальныйКонтекст [1] ВыделенноеИмя OPTIONAL}

-- общие аргументы/результаты --

АргументыСцепления ::=          SET{
        пунктПорождения         [0]  ВыделенноеИмя OPTIONAL,
        целевой Объект          [1]  ВыделенноеИмя OPTIONAL,
        продвижениеОперации     [2]  ПродвижениеОперации DEFAULT{неНачата},
        информацияСледа         [3]  ИнформацияСледа,
        псевдонимПереименован   [4]  BOOLEAN DEFAULT FALSE,
        ОВИОбъектаПсевдонима    [5]  INTEGER OPTIONAL,
            -- используется только в том случае, если псевдонимПереименован = TRUE
```

```
        возвратитьПерекрСсылку      [6]  BOOLEAN DEFAULT FALSE,
        типСсылки                   [7]  Тип Ссылки DEFAULT вверх,
        инфо                        [8]  ОбластьИнфо OPTIONAL,
        ограничениеВремени          [9]  ВремяОтГринвича OPTIONAL,
                                    [10] ПараметрыБезопасности DEFAULT {}}

РезультатСцепления      ::=         SET{
        инфо                        [0]  ОбластьИнфо OPTIONAL,
        перекрестныеСсылки          [1]  SEQUENCE OF ПерекрестнаяСсылка OPTIONAL,
                                    [2]  ПараметрыБезопасности DEFAULT{}}

ПерекрестнаяСсылка      ::=         SET{
        префиксальныйКонтекст       [0]  ВыделенноеИмя,
        пунктДоступа                [1]  ПунктДоступа}

ТипСсылки               ::=         ENUMERATED{
            вверх                                (1),
            вниз                                 (2),
            перекрестная                         (3),
            неСпецифицированнаяВниз              (4)}

ИнформацияСледа         ::=         SEQUENCE OF ЭлементСледа
        ЭлементСледа    ::=         SET{
            сас                      [0]  Имя,
            целевойОбъект            [1]  Имя,
            продвижениеОперации      [2]  ПродвижениеОперации }

ПродвижениеОперации     ::=         SET{
        фазаРазрешенияИмени          [0]  ENUMERATED{
                                              неНачата      (1),
                                              продолжается  (2),
                                              завершена     (3)}
        следующееРазрешаемоеОВИ      [1]  INTEGER OPTIONAL }

ОбластьИнфо  ::=  ANY

ССылкаНаПродолжения                  ::=  SET{
        целевойОбъект                [0]  Имя,
        оВИОбъектаПсевдонима         [1]  INTEGER OPTIONAL,
        продвижениеОперации          [2]  ПродвижениеОперации,
        разрешенныеОВИ               [3]  INTEGER OPTIONAL,
        типСсылки                    [4]  ТипСсылкн OPTIONAL,
                                     -- используется только в САС --
        пунктДоступа                 [5]  SET OF ПунктДоступа }

ПунктДоступа  ::=  SET{
        титул          [0]   Имя,
        адрес          [1]   АдресВУровнеПредставлений}
```

ПРИЛОЖЕНИЕ B

(к Рекомендации X.518)

Моделнрование знаний

Данное Приложение не является составной частью настоящей Рекомендации.

B.1 *Пример моделирования знаний*

Пример, приведенный ниже, иллюстрирует информационные знания, которые должны были бы поддерживать САС, изображенные на рис. 5/X.518 (§ 9). На рис. 5/X.518 изображено гипотетическое ИДС, логически разбитое на пять именующих контекстов (A, B, C, D и E) и распределенное физически по трем САС (CAC1, CAC2, CAC3). В этом примере CAC1 хранит контекст C, CAC2 хранит контексты A, B и E, а CAC3 хранит контекст D.

На рис. B-1/X.518-B-3/X.518 используются следующие сокращения:

ВВЕРХ — ссылка вверх

ВНИЗ — ссылка вниз

ВНТР — внутренняя ссылка

НСПСВ — неспецифицированная ссылка вниз

ПРКР — перекрестная ссылка

САСп — выделенное имя САСп

АУП — адрес в уровне представлений

ПК — префиксальный контекст

ОВИ — относительно выделенное имя

САС — выделенное имя системного агента Справочника

УКЗ — указатель

ИОП — имя объекта псевдонима

Примечание. — Рисунки, приведенные ниже, предназначены только для наглядной иллюстрации концепций, рассматриваемых в этом параграфе. Вопрос о том, как информационные знания фактически размещены и обслуживаются конкретной реализацией САС, является его *внутренним вопросом* и выходит за пределы настоящей Рекомендации.

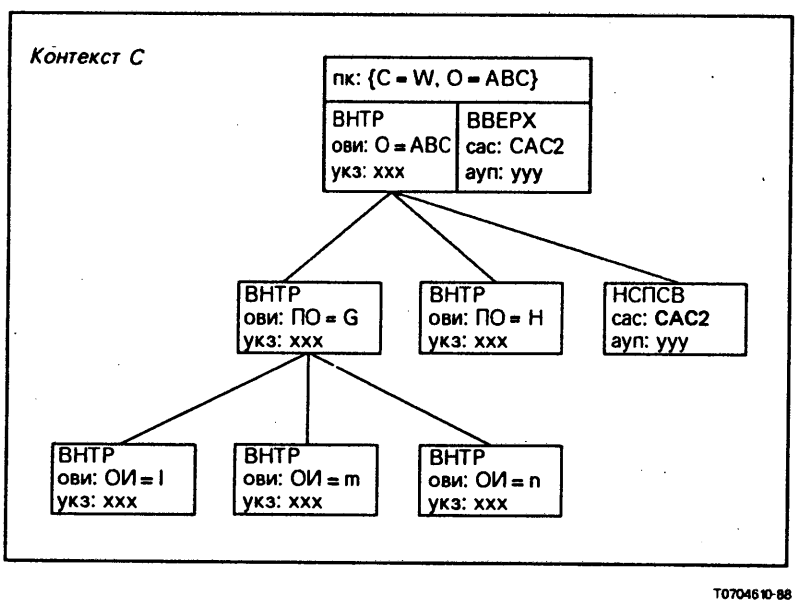


РИСУНОК B-1/X.518

Информационные знания САС1

На рис. B.1/X.518 изображены те информационные знания, которые должны храниться у САС1. Они должны содержать следующие префиксальные контексты и наборы ссылок.

Префиксальные контексты: {C =WW, O =ABC }, контекст С.

Перекрестные ссылки: { }

Ссылки вверх: {CAC2, адрес в Уровне Представлений для CAC2 }

Внутренние ссылки для контекста С: {C = WW, O =ABC },
{ПО = G } , {ПО =H }
{ПО = G, ОИ =L},
{ПО = G, ОИ =m},
{ПО = G, ОИ =n}.

Ссылки вниз: { }

Неспецифицированные ссылки вниз: {CAC2, адрес в Уровне Представлений для CAC2}.

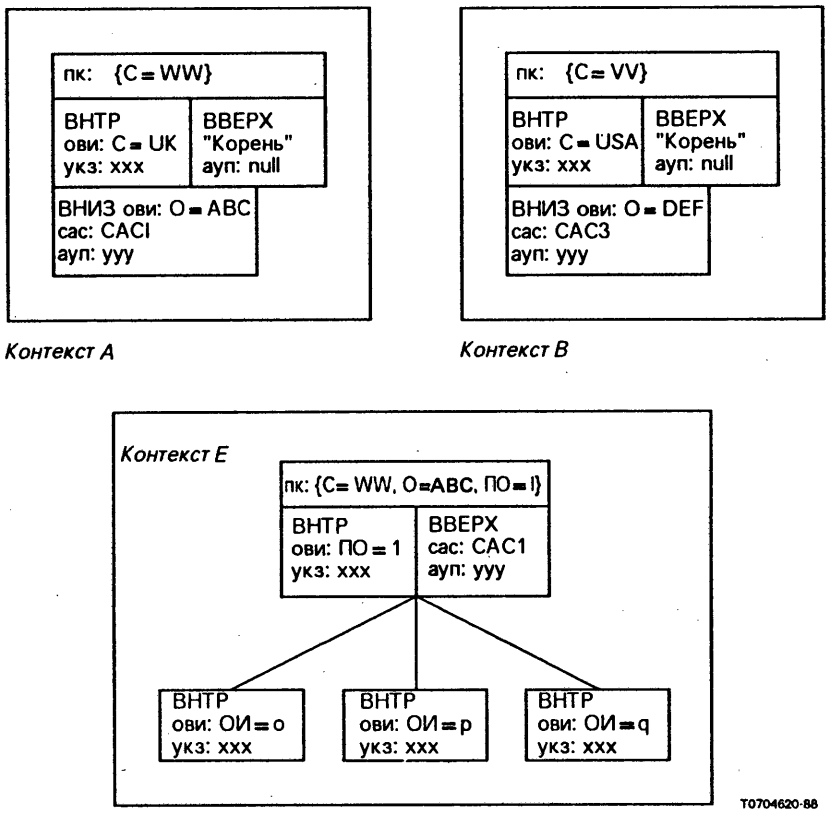


РИСУНОК B-2/X.518

Информационные знания CAC2

На рис. B-2/X.518 изображены те информационные знания, которые должны хранить CAC2. Они должны содержать следующие префиксальные контексты и наборы ссылок.

Префиксальные контексты: {C = WW}, контекст A
{C = VV }, контекст B
{C = WW, O — ABC, ПО — 1}, контекст E.

Перекрестные ссылки: { }

Ссылки вверх: { }

Внутренние ссылки для контекста A: {C = WW}

Внутренние ссылки для контекста B: {C = VV}

Внутренние ссылки для контекста E: {C = WW, O = ABC, ПО = 1},
{ОИ = o},
{ОИ = p},
{ОИ = q}.

Ссылки вниз для контекста A: {C = WW, O = ABC}

Ссылки вниз для контекста B: {C = VV, O =DEF}

Неспецифицированные ссылки вниз: { }

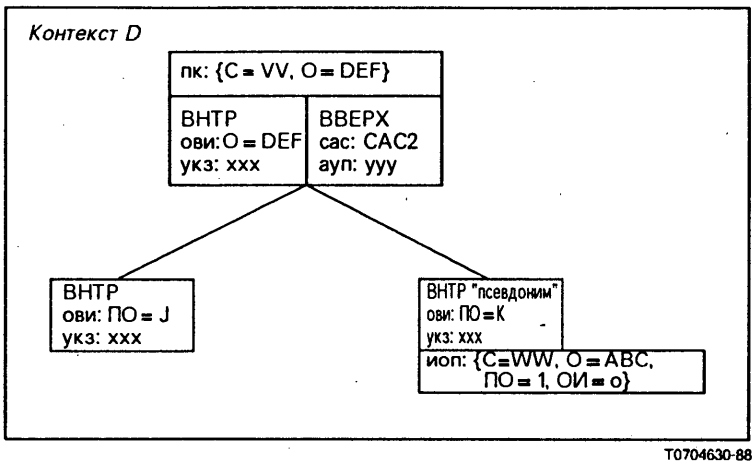


РИСУНОК B-3/X.518

Информационные знания САСЗ

На рис. B-3/X.518 изображены те информационные знания, которые должны храниться у САСЗ. Они должны содержать следующие префиксальные контексты и наборы ссылок.

Префиксальные контексты: {C = VV, O = DEF}, контекст D

Перекрестные ссылки: {{C = WW, O = ABC, ПО = H},
адрес в Уровне представлений для CAC1} (не изображен на рисунке, выше)

Ссылки вверх: {CAC2, адрес в уровне представлений для CAC2}

Внутренние ссылки для контекста D: {CAC1, адрес в уровне представлений для CAC1}
{C = VV, O = DEF}
{ПО = J }
{ПО = K }, псевдоним для {C = WW, O = ABC, ПО = I, ОИ = o}(информация о псевдонимах не является частью информационного знания).

Ссылки вниз: { }

Неспецифицированные ссылки вниз: { }

B.2 *Пример распределенного разрешения имени*

Ниже приводится пример того, как Распределенное Разрешение Имени используется для выполнения различных запросов к Справочнику. Пример основан на гипотетическом ИДС, изображенном на рис. 5/X.518 (§ 9) и соответствующих конфигурациях САС, изображенных на рис. B-1/X.518 — B-3/X.518 (Приложение B).

Предположим, что мы используем метод сцепления для продвижения запроса. Тогда приводимые ниже запросы, адресованные CAC1, будут обработаны следующим образом:

1) Запрос с выделенным именем {C = WW, O = ABC, ПО = C, ОИ = 1}

— Совпадает с префиксальным контекстом {C = WW, O = ABC} контекста C, административным органом которого является CAC1. Поэтому разрешение имени начнется в CAC1 с контекстом C.

— Разрешение имени будет продвигаться вниз по контексту C; при этом будут успешно сопоставляться все ОВИ, пока не будет обнаружено местоположение ОИ = 1.

2) Запрос с выделенным именем { C = WW, O = JPR }

— Такой запрос не может быть сопоставлен ни с одним префиксальным контекстом из тех, которые хранятся у CAC1. Поэтому CAC1, используя свою ссылку вверх, продвинет запрос к предшествующему CAC, а именно к CAC2.

— В CAC2 запрос совпадает с префиксальным контекстом {C = WW} и в CAC2 будет начато разрешение имени с контекстом A.

— Разрешение имени не найдет вершины, подчиненной вершине C = WW, которая могла бы быть сопоставлена с ОВИ, равным O = JPR, а имя будет признано недействительным (то есть ссылка на несуществующий объект).

3) Опрос с выделенным именем { C = WW, O = DEF, ПО = K }

— Не совпадает ни с одним префиксальным контекстом, хранящимся у CAC1.

— Поэтому CAC1 продвинет запрос к предшествующему CAC, а именно к CAC2.

— Запрос совпадает с префиксальным контекстом {C = VV} контекста B, хранящимся у CAC2. Поэтому разрешение имени начнется в CAC2 с контекстом B.

— Когда разрешение имени сделает попытку сопоставить O = DEF, будет обнаружена ссылка вниз, указывающая, что {C = VV, O = DEF} образует начало нового контекста, хранящегося у CAC3.

— Разрешение имени будет продолжено в CAC3, пока не будет локализовано расположение { C = VV, O = DEF, ОИ = K }.

— Считая, что псевдонимы должны быть переименованы, будет выработано новое имя, используя для этого имя объекта псевдонима, хранящееся в статье { C = VV, O = DEF, ПО = K }. В результате будет выработано новое имя { C = WW, O = ABC, ПО = I, ОИ =o } .

— CAC3 продолжит обработку запроса, используя новое имя, полученное в результате переименования.

ПРИЛОЖЕНИЕ C

(к Рекомендации X.518)

Распределенное использование аутентификации

Данное Приложение не является составной частью настоящей Рекомендации.

C.1 *Сводное описание*

Модель безопасности определена в § 10 Рекомендации X.501. Ниже приводится сводное описание основных пунктов модели.

a) Простая Аутентификация инициатора операции в СПС не обеспечивается.

b) Строгая Аутентификация, основанная на подписывании запроса и результата, в СПС обеспечивается.

c) Шифровка запроса или ответа в СПС не обеспечивается.

d) Аутентификация ошибок, включая отсылки, в СПС не обеспечивается.

В настоящем Приложении описывается реализация вышеуказанного пункта b) в распределенном Справочнике. В описании используется терминология и нотация, определенные в Рекомендации X.509.

C.2 *Простая аутентификация*

Аутентификация АПС осуществляется как часть операции Привязывания в СПС. Поэтому в CAC будет содержаться только имя АПС; оно хранится в поле "инициатор" Аргументов Сцепления.

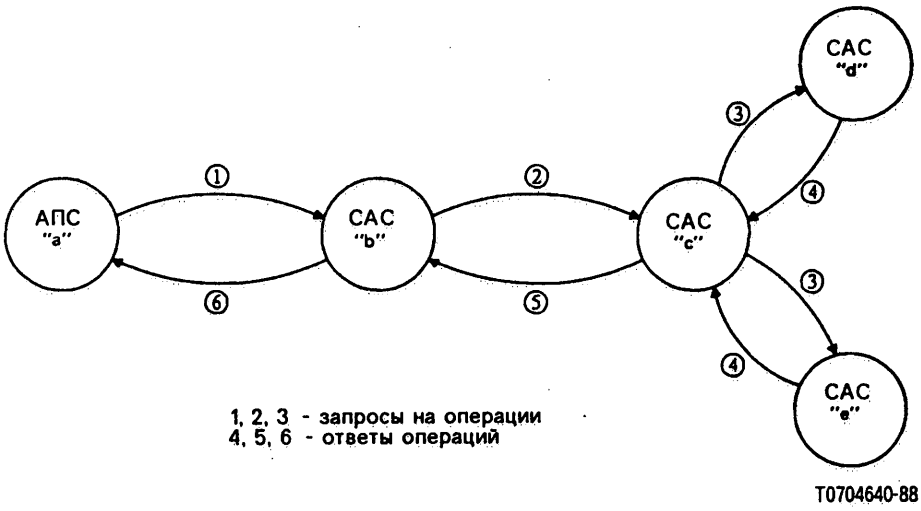


РИСУНОК C-1/X.518

Модель распределенной аутентификации

На рис. C-1/X.518 иллюстрируется модель, которая используется для спецификации процедур распределенной аутентификации. В модели идентифицируется последовательность потоков аутентификации для общего случая операций "список" или "поиск". Предполагается, что операцию инициирует АПС "a", указывая целевой объект, который хранится у САС "c"; в выполнение операции будут вовлечены следующие САС: "b", "c", "d" и "e".

Сначала САС "a" обращаются к любому САС (САС "b"), который, хотя и не хранит у себя целевого объекта, но может с помощью сцепления приложить курс к тому САС (САС "c"), который хранит целевой объект. Если бы все САС функционировали в режиме отсылки, то модель была бы существенно упрощена и каждый обмен АПС/САС сводился бы в терминах аутентификации к взаимодействию между АПС "a" и САС "b".

C.4 *От АПС к САС*

Аутентификация инициатора осуществляется как результат обмена (1). На рис. C-1/X.518 процедура аутентификации осуществляется следующим образом:

Пусть
OA = Аргументу Операции, то есть Поиска, Чтения, Сравнения и т.д. Под Аргументом понимается то, как он определен в части 3.

и

a {OA} = Аргументу Операции, подписанному АПС "a".

Аутентификация будет определена верификацией подписи.

C.5 *Перенос от ПДС к СПС*

На рис. C-1/X.518 эту процедуру осуществляет САС "b"; она представляет перенос подписанного свидетельства подлинности инициатора от ПДС к СПС.

САС "b" формирует требующийся Аргумент сцепления, как это описано в § 12.3 настоящей Рекомендации, и комбинирует его с Аргументом Операции из ПДС, образуя тем самым Сцепленную операцию СПС, то есть Сцепленное Чтение, Поиск, Список и т.д. Так сформированная Сцепленная Операция будет подписана до передачи ее другим САС (САС "c" на рис. C-1/X.518). Структура данных может быть представлена как

в {ChA, a {OA}} = Сцепленной операции, подписанной САС "b",

где

ChA = Аргументу Сцепления.

Таким образом, аутентифицирующая информация, передаваемая по СПС между двумя САС [на рис. С-1/Х.518 она помечена через (2)], состоит из двух частей:

– Аргумента Операции, подписанного инициатором, что делает возможным его аутентификацию;

– Сцепленной Операции, посылаемой САС-отправителем, что делает возможным его аутентификацию.

С.6 *Сцепление через промежуточные САС*

В модели, изображенной на рис. С-1/Х.518, процедуру осуществляет САС "с". САС "с" аннулирует подпись, представленную САС-отправителем (САС "b" на рис. С-1/Х.518), и модифицирует Аргумент Сцепления, как это описано в § 12.3 настоящей Рекомендации. САС "с" затем объединит модифицированный Аргумент Сцепления с подписанным Аргументом Операции и подпишет полученный результат для формирования подписанной Сцепленной Операции. Это можно представить в виде:

$c\{ChA', a\{OA\}\}$ = Сцепленной Операции, подписанной САС "с",

где

ChA' = модифицированному Аргументу Сцепления.

Модифицированная Сцепленная Операция на рис. С-1/Х.518 изображена в виде обмена (3). В зависимости от природы операции и от типа хранимых знаний САС "с" может выполнить разложение запроса до сцепления или многоадресной рассылки любой получившейся в результате операции. На рис. С-1/Х.518 это было изображено в виде САС "с", посылающего операции САС "d" и САС "e"; в обоих случаях процедуры аутентификации идентичны.

С.7 *Аутентификация результатов*

Службу аутентификации результатов запрашивает инициатор операции Справочника, используя для этого вариант **подписано** параметра **ЗапросЗащиты** из **ПараметраБезопасности**. Выдавая ответ на такой запрос САС может, но не обязательно, решить, подписывать ли некоторые или все результаты; служба аутентификации ответов не обеспечивает аутентификации служебных ответов.

В контексте некоторого конкретного САС, обрабатывающего ответы от любого числа САС (каждый из которых связан с отдельным запросом на службу), возможны следующие различные случаи:

– САС обеспечивает полный набор результатов операции, не нуждаясь в выполнении какой бы то ни было функции объединения (что на рис. С-1/Х.518 изображено САС "d" и "e");

– САС объединяет локальные ответы (источником которых был данный САС) с ответами от одного или нескольких других САС (что на рис. С-1/Х.518 изображено САС "c");

– САС осуществляет сцепление результата от САС или с другим САС, или с АПС и в силу этого сам ничего к результату не добавляет (что на рис. С-1/Х.518 изображено САС "b").

С.7.1 *Результат от САС – без объединения*

В настоящем параграфе рассматривается роль САС, являющегося единственным источником результатов на некоторый конкретный запрос, то есть САС не должен выполнять функций объединения. В параграфе этот случай рассматривается как для СПС, так и для ПДС.

С.7.1.1 *СПС*

САС может решить выполнить любую из следующих процедур:

– вернуть результаты неподписанными, что может быть представлено в виде:

  ChR, OR = Результату Сцепленной Операции (неподписанному),

  где

  ChR = Сцеплению Результатов,

  OR = Результату Операции;

– подписать только Результат Операции, что может быть представлено в виде:

  ChR, d {OR} = Результату Операции, подписанному САС "d";

– подписать только Результат Сцепленной Операции, что может быть представлено в виде:

d { ChR, OR } = Результату Сцепленной Операции, подписанному CAC."d";

— подписать как Результат операции, так и Результат Сцепленной Операции, что может быть представлено в виде:

d {ChR, D { OR }} = Результату Операции и Результату Сцепленной Операции, подписанным CAC "d".

Примечание. — В том случае, если Результат Операции подписан, он будет передан обратно инициатору; в том случае, если результат Сцепленной Операции подписан, CAC-приемник должен аннулировать подпись, для того чтобы модифицировать аргумент Результатом Сцепления до дальнейшей передачи Результата Сцепленной Операции.

C.7.1.2 *ПДС*

Этот случай полностью описан в Рекомендации X.511; для полноты здесь приводится итоговое описание.

CAC может предпочесть возвратить ответ либо неподписанным, что может быть представлено в виде:

OR = Результату Операции,

либо — подписанным, что может быть представлено в виде:

d {OR} = Результату Операции, подписанному CAC "d".

C.7.2 *Результаты от CAC – с объединением*

В данном параграфе рассматривается роль CAC в возвращении ответа на некоторый конкретный запрос на службу в случае, когда обязательным требованием является объединение и интеграция результатов от других CAC. В параграфе этот случай рассматривается как для СПС, так и для ПДС.

C.7.2.1 *СПС*

Учитывая, что ноль или более ответов, полученных от других CAC, могут быть подписаны, данная процедура позволяет CAC объединить и интегрировать результаты, подписать ноль или более составных частей сводного результата, а также, как необязательную возможность, подписать и весь сводный результат.

C.7.2.1.1 *Выработка аргумента результатов сцепления*

Эта процедура требует от CAC (изображенном на рис. C-1/X.518 CAC "c") удаления подписей Результатов Сцепленных Операций из всех результатов, полученных от внешних CAC (CAC "d" и "e" на рис. C-1/X.518). В итоге CAC "c" владеет набором неподписанных результатов Сцепления, набором подписанных Результатов Операций и набором неподписанных Результатов Операций.

Все Результаты Сцепления обрабатываются, как это описано в § 12.4 настоящей Рекомендации, в результате чего получается один модифицированный Результат Сцепления, представляемый в виде:

(i) ChR′ = модифицированномуРезультату Сцепления.

C.7.2.1.2 *Локально полученные не подписанные результаты*

Если CAC не желает подписать локально полученные результаты, то набор неподписанных Результатов Операций сливается с локальным результатом, образуя тем самым модифицированный набор Результатов Операций, обозначаемый в виде:

OR′ = Объединенному Результату Операции.

Таким образом, полный набор Результатов Операции является объединением набора внешне полученных подписанных результатов Операций, обозначаемых через:

d {OR} , e {OR} ,...

и слитым Результатом Операции, совокупно обозначаемым через:

(ii) OR′, d {OR}, e {OR} ... = Результату Операции

C.7.2.1.3 *Локально полученный подписанный результат*

Если CAC желает подписать локально полученные результаты, то тогда сначала сливается набор внешне полученных неподписанных Результатов Операций. Таким образом, полный набор Результатов Операций является

объединением локальных подписанных Результатов Операций, обозначенных через C {OR} , слитым набором внешних неподписанных Результатов Операций, обозначенных через OR'' и набором внешних подписанных Результатов Операций, обозначаемых через:

d {OR} , e {OR} , ..., которые совокупно обозначаются через:

(iii) c {OR} , OR'' , d {OR} , e {OR}, ... = РезультатуОперации

C.7.2.1.4 *Неподписанный результат сцепленной операции*

Если CAC не желает подписать Результат Сцепленной Операции, то этот последний будет состоять из Результатов Сцепления (обозначенных выше через (i)), добавленных к Результату Операции, идентифицированной выше либо в пункте (ii), либо в пункте (iii); совокупно они обозначаются через:

либо

ChR' , OR' , d {OR} , e {OR} , ... = Результату Сцепленной Операции (неподписанному),

либо

ChR' , c {OR} , OR'' , d {OR} , e {OR} , ... = Результату Сцепленной Операции (неподписанному) и Результату Операции, подписанному CAC "c"

C.7.2.1.5 *Подписанный результат сцепленной операции*

Если CAC желает подписать Результат Сцепленной Операции, то результат будет состоять из Результатов Сцепления (идентифицированных выше в пункте (i)), добавленных к Результату Операции (идентифицированному выше либо в пункте (ii), либо в пункте (iii); совокупно они обозначаются через:

либо

c {ChR' , OR' , d {OR} , e {OR} ,...}= Результату Сцепленной Операции, подписанному CAC "c",

либо

c {ChR' , c {OR} , OR'' , d {OR} , e {OR} , ...}= Результату Сцепленной Операции и Результату Операции, подписанным CAC "c",

C.7.2.2 *ПДС*

Процедура весьма близка описанной в § C.7.2.1 с тем лишь исключением, что ПДС не передает аргумента Результатов Сцепления.

C.7.3 *Сцепленные результаты CAC*

В данном параграфе рассматриваются процедуры, которые использует CAC при сцеплении результата операции назад к реквестору, CAC или АПС, соответственно в СПС или ПДС.

C.7.3.1 *СПС*

Сначала CAC удаляет подпись (если таковая существует) из Результата Сцепленной Операции. Далее он обрабатывает аргумент "Результатов Сцепления", как это описано в настоящей Рекомендации, в результате чего получается модифицированный аргумент Результатов Сцепления. Затем этот последний сливается назад с аргументом "Результат Операции"; в итоге получается модифицированный Результат Сцепленной Операции. В результате CAC может при желании подписать Результат Сцепленной Операции перед передачей его следующему CAC в цепочке.

C.7.3.2 *ПДС*

CAC (изображенный на рис. C-1/X.518 CAC "b") сначала удаляет подпись (если таковая существует) из Результата Сцепленной Операции. Далее он анализирует и аннулирует аргумент "Результат Сцепления" и в результате при желании подписывает остающийся аргумент "Результат Операции" перед передачей результата АПС.

ПРИЛОЖЕНИЕ D

(к Рекомендации X.518)

Идентификаторы объектов распределенного Справочника

Данное Приложение является составной частью настоящей Рекомендации.

Данное приложение содержит HAC.1-модуль **ИдентификаторыОбъектовРаспределенногоСправочника**, в который включены все HAC.1-идентификаторы объектов, введенные в настоящей Рекомендации.

```
ИдентификаторыОбъектовРаспределенногоСправочника    { joint-iso-ccitt ds(5) modules(1)
                                                      distributedDirectoryObjectIdentifiers(13)}

DEFINITIONS ::=
BEGIN

EXPORTS
        ид-от-сас, ид-пт-сцепленноеЧтение, ид-пт-сцепленныйПоиск, ид-пт-сцепленнаяМодификация;

IMPORTS
        ид-от, ид-пт
            FROM ПолезныеОпределения {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)};

-- объекты --

ид-от-сас OBJECT IDENTIFIER    ::= {ид-от 3}

-- типы частей --

ид-пт-сцепленноеЧтение          OBJECT IDENTIFIER ::= {ид-пт 4}
ид-пт-сцепленныйПоиск           OBJECT IDENTIFIER ::= {ид-пт 5}
ид-пт-сцепленнаяМодификация     OBJECT IDENTIFIER ::= {ид-пт 6}

END
```

Рекомендация X.519

СПРАВОЧНИК — СПЕЦИФИКАЦИЯ ПРОТОКОЛОВ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

---

1) Рекомендация X.519 и ISO 9594-5 "Справочник — Спецификация протоколов" были разработаны в тесном взаимодействии и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой ими справочной информацией может рассматриваться как объединенное целое, называемое *Справочником.* Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные процессы, люди, терминалы или распределенные списки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых разными производителями;

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 В настоящей Рекомендации специфицируются элементы прикладной службы и прикладной контекст двух протоколов: Протокола доступа к Справочнику (ПДС) и Системного протокола Справочника (СПС). ПДС обеспечивает доступ к Справочнику для чтения или модификации информации, содержащейся в Справочнике. СПС обеспечивает возможность сцепления запросов на выборку или модификацию информации, содержащейся в Справочнике, с теми частями распределенного Справочника, в которых может храниться искомая информация.

1 Предмет рассмотрения

В настоящей Рекомендации специфицируются Протокол доступа к Справочнику и Системный протокол Справочника, реализующие абстрактные службы, описанные в Рекомендациях X.511 и X.518.

2 Библиография

Рекомендация X.200 "Взаимосвязь Открытых Систем — Основная эталонная модель".

Рекомендация X.208 "Взаимосвязь Открытых Систем — Спецификация нотации абстрактного синтаксиса номер один (НАС.1)".

Рекомендация X.209 "Взаимосвязь Открытых Систем — Спецификация основных правил кодирования нотации абстрактного синтаксиса номер один (НАС.1)".

Рекомендация X.500 "Справочник — Обзор концепций, моделей и служб".

Рекомендация X.501 "Справочник — Модели".

Рекомендация X.511 "Справочник — Определение абстрактных служб".

Рекомендация X.518 "Справочник — Процедуры распределенных операций"

Рекомендация X.520 "Справочник — Избранные типы атрибутов".

Рекомендация X.521 "Справочник — Избранные классы объектов".

Рекомендация X.219 "Удаленные операции — Модель, нотация и определение служб".

Рекомендация X.229 "Удаленные операции — Спецификация протокола".

Рекомендация X.217 "Взаимосвязь открытых систем — Управление ассоциацией: определение служб".

Рекомендация X.216 "Взаимосвязь открытых систем — Определение служб уровня представлений".

3 Определения

Определения, содержащиеся в настоящем пункте, используют сокращения, определенные в § 4.

3.1 *Определения эталонной модели ВОС*

Настоящая Рекомендация опирается на концепции, развитые в Рекомендации X.200, и использует следующие термины, определенные в ней:

a) *элемент-прикладной-службы;*

b) *управляющая-информация-прикладного-протокола;*

c) *блок-прикладных-управляющих-данных;*

d) *прикладной-контекст;*

e) *прикладной-элемент;*

f) *абстрактный-синтаксис.*

3.2 *Основные определения Справочника*

В настоящей Рекомендации используются следующие термины, определенные в Рекомендации X.501:

a) *Справочник;*

b) *пользователь (Справочника);*

c) *Системный агент Справочника (САС);*

d) *Агент пользователя Справочника (АПС).*

3.3 *Определения распределенной операции*

В настоящей Рекомендации используются нижеследующие термины, определенные в Рекомендации X.518:

a) *сцепление;*

b) *отсылка.*

4 Сокращения

В настоящей Рекомендации используются следующие сокращения:

ПК — прикладной контекст

ЭСУА — элемент службы управления ассоциацией

ПЭ — прикладной элемент

ПкУИП — управляющая информация прикладного протокола

ПкБДП — блок данных прикладного протокола

ЭПС — элемент прикладной службы

ПДС — протокол доступа к Справочнику

САС — системный агент Справочника

СПС — системный протокол Справочника

АПС — агент пользователя Справочника

ЭСУО — Элемент службы удаленных операций

3 Соглашения

В настоящей Рекомендации используются следующие соглашения:

a) определения абстрактного синтаксиса в § 7 используют нотацию абстрактного синтаксиса, определенную в Рекомендации X.208;

b) макрос удаленных операций (нотация-УО), макрос элемента-прикладной-службы и макрос прикладного-контекста определены в Рекомендации X.219;

c) слова, входящие в определения терминов, а также имена и значения параметров службы и полей протоколов, не являющиеся именами собственными, начинаются со строчной буквы и соединяются друг с другом дефисом, например "определенный-термин". Имена собственные начинаются с прописной буквы и не соединяются дефисом, например "Имя Собственное".

6 Общее описание Протокола

6.1 *Модель Протокола Справочника*

В Рекомендации X.511 описывается абстрактная служба между АПС и Справочником, обеспечивающая доступ пользователя к службам, предоставляемым Справочником. Справочник моделируется с помощью САС, обеспечивающего конкретный рассматриваемый пункт доступа. В Рекомендации X.518 описывается взаимодействие между двумя САС Справочника, обеспечивающее пользовательские запросы, подвергающиеся сцеплению. Эти концепции иллюстрируются на рис. 1/X.519.

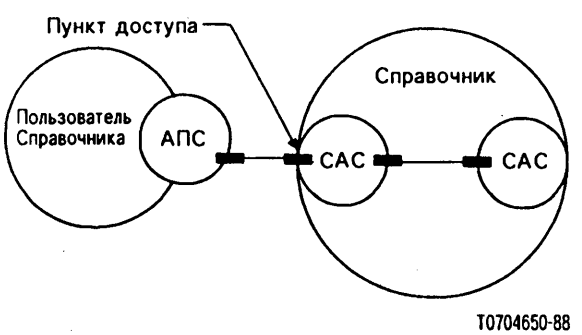


РИСУНОК 1/X.519

Взаимодействия Справочника

Если АПС находится в открытой системе, отличной от той, в которой находится САС, с которым АПС взаимодействует, то эти взаимодействия обеспечиваются Протоколом Доступа к Справочнику (ПДС), являющимся протоколом прикладного уровня ВОС. Аналогично, если два взаимодействующих САС находятся в различных открытых системах, то эти взаимодействия обеспечиваются Системным протоколом Справочника (СПС), также находящимся в прикладном уровне.

И ПДС, и СПС являются протоколами, обеспечивающими связь между двумя прикладными процессами. В среде ВОС такое взаимодействие рассматривается как связь между двумя прикладными-элементами (ПЭ), использующими службы уровня представлений. Функцию, реализуемую ПЭ, обеспечивает набор элементов-прикладной-службы (ЭПС). Взаимодействие между двумя ПЭ описывается в терминах использования ими служб, которые обеспечивают ЭПС. Два ЭПС, используемые в обоих протоколах справочника, кратко описываются в настоящем параграфе.

Элемент службы Удаленных операций (ЭСУО) обеспечивает парадигму запросов/ответов абстрактной операции, реализуемой в портах абстрактной модели. ЭПС Справочника обеспечивают отображение нотации абстрактного синтаксиса абстрактной-службы Справочника на службы, обеспечиваемые ЭСУО.

Элемент службы управления ассоциацией (ЭСУА) обеспечивает установление и освобождение прикладной-ассоциации между двумя ПЭ. Ассоциация между АПС и САС может быть установлена только по инициативе АПС. Только инициатор ассоциации может освободить ее.

6.2 *Протокол доступа к Справочнику*

Протокол доступа к Справочнику (ПДС) используется для реализации Абстрактной службы Справочника. Этот протокол содержит, помимо ЭСУА и ЭСУО, еще три специфичных для Справочника ЭПС. Ими являются: **ЭПСЧтения, эПСПоиска, эПСМодификации.** Они соответствуют **портуЧтения, портуПоиска** и **портуМодификации** абстрактной службы. **пКДоступаКСправочнику** идентифицирует комбинацию, состоящую из **эПСЧтения, эПСПоиска, эПСМодификации, эСУА** и **эСУО.**

6.3 *Системный протокол Справочника*

Системный протокол Справочника (СПС) используется для реализации функционирования распределенной операции, описанной в Рекомендации X.518. Этот протокол содержит, помимо ЭСУА и ЭСУО, еще три специфичных для Справочника ЭПС. Ими являются: **эПСсцепленногоЧтения, эПСсцепленногоПоиска** и **эПСсцепленнойМодификации.** Они соответствуют **портуСцепленногоЧтения, портуСцепленногоПоиска** и **портуСцепленнойМодификации** абстрактной службы. **пКСистемногоСправочника** идентифицирует комбинацию, состоящую из **эПССцепленногоЧтения, эПСсцепленногоПоиска, эПСсцепленнойМодификации, эСУА** и **эСУО.**

6.4 *Использование нижележащих служб*

Протоколы ПДС и СПС используют нижележащие службы, как это описывается ниже.

6.4.1 *Использование служб ЭСУО*

Элемент службы Удаленных операций (ЭСУО) описывается в Рекомендации X.219.

ЭСУО обеспечивает выполнение всей парадигмы запросов/ответов удаленных операций.

Пользователями служб ЭСУО УО-ПОБУДИТЬ, УО-РЕЗУЛЬТАТ, УО-ОШИБКА, УО-ОТКАЗАТЬ-Пл и УО-ОТКАЗАТЬ-ПС являются ЭПС Справочника.

Удаленные операции протоколов ПДС и СПС являются операциями класса 2 (асинхронными). Обратите внимание на следующее обстоятельство: так как АПС является потребителем протокола ПДС, то он может предпочесть синхронный режим работы.

Протокол ПДС использует Ассоциации класса 1. Это означает, что САС не может побудить АПС начать выполнение какой-либо операции. СПС использует Ассоциации класса 3. Это означает, САС-респондер может побудить САС-реквестора начать выполнение некоторой операции и наоборот.

6.4.2 *Использование служб ЭСУА*

Элемент службы Управления ассоциаций (ЭСУА) описывается в Рекомендации X.217.

ЭСУА обеспечивает управление (установление, освобождение, прекращение) прикладной-ассоциации между двумя ПЭ.

Привязывание к Справочнику и отвязывание от Справочника являются единственными пользователями служб Пк-АССОЦИИРОВАНИЕ и Пк-ОСВОБОЖДЕНИЕ, которые обеспечивает ЭСУА в нормальном режиме. Прикладные-процессы являются пользователями служб Пк-ПРЕКРАЩЕНИЕ и Пк-Пс-ПРЕКРАЩЕНИЕ, которые поставляет ЭСУА.

6.4.3 *Использование служб уровня представлений*

Службы уровня представлений описаны в Рекомендации X.216.

Уровень представлений координирует представление (синтаксис) семантики подлежащих обмену сообщений прикладного уровня.

В нормальном режиме для различных абстрактных-синтаксисов, включаемых в прикладные-контексты, используются различные контексты-уровня-представлений.

ЭСУА является единственным пользователем служб-уровня-представлений Пд-СОЕДИНЕНИЕ, Пд-ОСВОБОЖДЕНИЕ, Пд-Пл-ПРЕКРАЩЕНИЕ и Пд-Пс-ПРЕКРАЩЕНИЕ.

ЭСУО является пользователем службы-уровня-представлений Пд-ДАННЫЕ.

6.4.4 *Использование служб нижележащих уровней*

Сеансовые-службы описаны в Рекомендации X.215. Сеансовый уровень структурирует поток диалоговой информации между оконечными-системами.

Уровень представлений использует службы функционального блока Ядра и Дуплексного функционированного блока сеансовых-служб.

Транспортные-службы описаны в Рекомендации X.214. Транспортный уровень обеспечивает прозрачную передачу данных от одного конца связи до другого по нижележащему сетевому соединению.

Выбор класса транспортных служб, используемых Сеансовым уровнем, зависит от требований к мультиплексированию и восстановлению ошибок. Обязательным является обеспечение Транспортного класса 0 (без мультиплексирования). Транспортная служба Срочных данных не используется.

Использование остальных классов является не обязательным. Мультиплексный класс может быть использован для мультиплексирования протоколов ПДС и СПС и других протоколов по одному и тому же сетевому соединению. Класс восстановления ошибок может быть выбран для случая, когда сетевое соединение обеспечивает недопустимый уровень невыявленных ошибок.

Подразумевается использование сетевого уровня, обеспечивающего сетевые службы ВОС. Эти службы описаны в Рекомендации X.213.

Сетевой-адрес определен в Рекомендации X.121, Рекомендациях E.163/E.164 или Рекомендации X.200 (Адрес-ТДСС-ВОС).

7 Абстрактный синтаксис Протокола Справочника

7.1 *Абстрактные синтаксисы*

ЭПС Справочника, специфицируемые в § 7.2.1, 7.2.3, 7.2.5, используют один и тот же абстрактный-синтаксис, обозначаемый через **ид-ас-АСДоступаКСправочнику**. Те ЭПС, которые специфицируются в § 7.2.2, 7.2.4 и 7.2.6 используют один и тот же абстрактный синтаксис, обозначаемый через **ид-ас-системнаяАССправочника**. В обоих случаях в эти определения входит управляющая-информация-прикладного-протокола (ПкУИП). Использование ПкУИП совместно с ЭСУО определяет набор ПкБДП. ПкБДП Справочника определяются с помощью абстрактного-синтаксиса ЭСУО и ЭПС Справочника. Эти блоки совместно с абстрактным-синтаксисом ЭСУА образуют полное определение всех ПкБДП, используемых в период ассоциации со Справочником.

Для установления ассоциаций требуется абстрактный-синтаксис ЭСУА, обозначаемый через **ид-ас-эсуа**.

Эти абстрактные синтаксисы должны быть кодированы (как минимум) в соответствии с основными правилами кодирования НАС.1.

7.2 *Элементы прикладной службы Справочника*

В настоящем параграфе специфицируются те ЭПС, которые используются в § 7.3 в качестве "строительных кирпичей" при конструировании различных прикладных контекстов Справочника.

Примечание. — Эти ЭПС используются при конструировании прикладных контекстов, определяемых в настоящей Рекомендации. Однако они не претендуют на то, чтобы быть подходящими для индивидуальных ЭПС или каких-то других комбинаций ЭПС.

7.2.1 *ЭПС чтения*

Элемент **эПСЧтения** обеспечивает выполнение абстрактных операций **портаЧтения**, а именно операций **Чтения**, **Сравнения** и **Отказа**, как они определены в Рекомендации X.511.

```
эПСЧтения
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            {чтение, сравнение, отказ}
    :: = ид-эпс-ЭПСЧтения

чтение      Чтение      :: = 1

сравнение   Сравнение   :: = 2

отказ       Отказ       :: = 3
```

7.2.2 *ЭПС сцепленного чтения*

Элемент **эПССцепленногоЧтения** обеспечивает выполнение абстрактных операций **ПортаСцепленного Чтения**, а именно операций **СцепленногоЧтения**, **СцепленногоСравнения** и **СцепленногоОтказа**, как они определены в Рекомендации X.518.

```
эПССцепленногоЧтения
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {
            сцепленноеЧтение,
            сцепленноеСравнение,
            сцепленныйОтказ}
    :: = ид-эпс-ЭПССцепленногоЧтения

сцепленноеЧтение        СцепленноеЧтение        :: = 1

сцепленноеСравнение     СцепленноеСравнение     :: = 2

сцепленныйОтказ         СцепленныйОтказ         :: = 3
```

7.2.3 *ЭПС поиска*

Элемент **эПСПоиска** обеспечивает выполнение абстрактных операций **ПортПоиска**, а именно операций **Список** и **Поиск**, как они определены в Рекомендации X.511.

```
эПСПоиска
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {список, поиск}
    :: = ид-эпс-ЭПСПоиска

список  Список :: = 4

поиск   Поиск  :: = 5
```

7.2.4 *ЭПС сцепленного поиска*

Элемент **эПССцепленногоПоиска** обеспечивает выполнение абстрактных операций **ПортаСцепленного Поиска**, а именно операций **СцепленногоСписка** и **СцепленногоПоиска**, как они определены в Рекомендации X.518.

```
эПССцепленногоПоиска
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
            {сцепленныйСписок, сцепленныйПоиск}
    :: = ид-эпс-ЭПССцепленногоПоиска

сцепленныйСписок        СцепленныйСписок        :: = 4

сцепленныйПоиск         СцепленныйПоиск         :: = 5
```

7.2.5 *ЭПС Модификации*

Элемент **эПСМодификации** обеспечивает выполнение абстрактных операций **ПортаМодификации**, а именно операций **ДобавленияСтатьи**, **УдалениеСтатьи**, **МодификацияСтатьи** и **МодификацияОВИ**, как они описаны в Рекомендации X.511.

```
эПСМодификации
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            { добавлениеСтатьи, удалениеСтатьи,
            модификацияСтатьи, модификацияОВИ}
    :: = ид-эпс-ЭПСМодификации

добавлениеСтатьи      ДобавлениеСтатьи      :: = 6
удалениеСтатьи        УдалениеСтатьи        :: = 7
модификацияСтатьи     МодификацияСтатьи     :: = 8
модификацияОВИ        МодификацияОВИ        :: = 9
```

7.2.6 *ЭПС сцепленной модификации*

Элемент эПССцепленнойМодификации обеспечивает выполнение абстрактных-операций ПортаСцепленнойМодификации, а именно операций СцепленногоДобавленияСтатьи, СцепленногоУдаленияСтатьи, СцепленнойМодификацииСтатьи и СцепленнойМодификацииОВИ, как они описаны в Рекомендации X.518.

```
эПССцепленнойМодификации
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
            { сцепленноеДобавлениеСтатьи,
            сцепленноеУдалениеСтатьи,
            сцепленнаяМодификацияСтатьи,
            сцепленнаяМодификацияОВИ}
    :: = ид-эпс-ЭПССцепленнойМодификации
сцепленноеДобавлениеСтатьи   СцепленноеДобавлениеСтатьи   :: = 6
СцепленноеУдалениеСтатьи     СцепленноеУдалениеСтатьи     :: = 7
сцепленнаяМодификацияСтатьи  СцепленнаяМодификацияСтатьи  :: = 8
сцепленнаяМодификацияОВИ     СцепленнаяМодификацияОВИ     :: = 9
```

7.3 *Прикладные контексты Справочника*

7.3.1 *Прикладной контекст доступа к Справочнику*

Контекст пКДоступаКСправочнику обеспечивает АПС возможность доступа к операциям следующих ЭПС: эПСЧтения, эПСПоиска и эПСМодификации.

```
пКДоступаКСправочнику
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {эСУА}
        BIND ПривязываниеКСправочнику
        UNBIND ОтвязываниеОтСправочника
        REMOTE OPERATIONS {эСУО}
        INITIATOR CONSUMER OF{
            эПСЧтения,
            эПСПоиска,
            эПСМодификации}
        ABSTRACT SYNTAXES{
            ид-ас-эсуа,
            ид-ас-АСДоступаКСправочнику}
    :: = ид-пк-пКДоступаКСправочнику
```

7.3.2 *Системный прикладной контекст Справочника*

СистемныйПКСправочника обеспечивает САС возможность связи в целях сцепления операций.

```
системныйПССправочника
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {эСУА}
        BIND САСПривязывание
        UNBIND САСОтвязывание
```

```
            REMOTE OPERATIONS {эСУО}
            OPERATIONS OF
                { эПССцепленногоЧтения,
                эПССцепленногоПоиска,
                эПССцепленнойМодификации}
            ABSTRACT SYNTAXES {
                ид-ас-эсуа,
                ид-ас-системнаяАССправочника}
::=         ид-пк-системныйПКСправочника
```

7.4 *Ошибки*

Протокол может передавать значение ошибки, соответствующей каждой абстрактной-ошибке, описанной в абстрактной службе. Используются нижеследующие присвоения:

отказано	**Отказано**	**::= 5**
ошибкаАтрибута	**ОшибкаАтрибута**	**::= 1**
ошибкаИмени	**ОшибкаИмени**	**::= 2**
отсылка	**Отсылка**	**::= 4**
ошибкаБезопасности	**ОшибкаБезопасности**	**::= 6**
ошибкаСлужбы	**ОшибкаСлужбы**	**::= 3**
ошибкаОбновления	**ОшибкаОбновления**	**::= 8**
сАСОтсылка	**сАСОтсылка**	**::= 9**
невыполненныйОтказ	**НевыполненныйОтказ**	**::= 7**

8 Отображение на используемые службы

В настоящем параграфе описывается отображение ПДС и СПС на используемые службы.

8.1 *Отображение на службы ЭСУА*

В настоящем параграфе описывается отображение служб абстрактного-привязывания (Привязывание КСправочнику или САСПривязывание) и абстрактного-отвязывания (ОтвязываниеОтСправочника или САСОтвязывание) на службы ЭСУА. ЭСУА описан в Рекомендации X. 217.

8.1.1 *Абстрактное-привязывание на Пк-АССОЦИИРОВАНИЕ*

Служба абстрактного-привязывания отображается на службу Пк-АССОЦИИРОВАНИЯ ЭСУА. Использование параметров службы Пк-АССОЦИИРОВАНИЕ задается в следующих подпараграфах.

8.1.1.1 *Режим*

Этот параметр должен поставляться инициатором ассоциации в примитиве Пк-АССОЦИИРОВАНИЕ запрос и должен иметь значение "нормальный режим".

8.1.1.2 *Имя прикладного контекста*

Инициатор связи должен предложить или системныйПКСправочника или пКДоступаКСправочнику.

8.1.1.3 *Информация пользователя*

Отображение операций-привязывания службы абстрактного-привязывания на параметры информации пользователя примитива Пк-АССОЦИИРОВАНИЕ запрос определяется в Рекомендации X.219.

8.1.1.4 *Список определений прикладных контекстов*

Инициатор связи должен представить Список определений прикладных контекстов в примитиве Пк-АССОЦИИРОВАНИЕ запрос. Этот список должен содержать абстрактный-синтаксис ЭСУА (ид-ас-эсуа) и либо абстрактный-синтаксис ПДС (ид-ас-АСДоступаКСправочнику), либо абстрактный-синтаксис СПС (ид-ас-СистемнаяАССправочника).

8.1.1.5 *Качество обслуживания*

Этот параметр должен поставляться инициатором ассоциации в примитиве Пк-АССОЦИИРОВАНИЕ запрос и респондером ассоциации в примитиве Пс-АССОЦИИРОВАНИЕ ответ. Значение параметров "Расширенное управление" и "Оптимизированная передача диалога" должно быть установлено равным "не требуется". Для остальных параметров должно использоваться значение по умолчанию.

8.1.1.6 *Сеансовые требования*

Этот параметр должен поставляться инициатором ассоциации в примитиве Пк-АССОЦИИРОВАНИЕ запрос и респондентом в примитиве Пк-АССОЦИИРОВАНИЕ ответ. Параметр должен специфицировать следующие функциональные блоки:

a) Ядра;

b) Дуплексный.

8.1.1.7 *Титул прикладного элемента и адрес в уровне представлений*

Эти параметры должны поставляться инициатором ассоциации и респондером ассоциации (поставка Титула прикладного элемента не является обязательной). Когда АПС устанавливает ассоциацию для начального запроса, он получает эти параметры из локально хранящейся информации.

Когда АПС (или САС) устанавливает ассоциацию с САС, к которому он был отослан, он получает эти параметры из ПунктаДоступа ССылкиНаПродолжение. Если ассоциацию устанавливает САС, то он получает этот параметр из Информационных знаний, то есть из внешней ссылки.

8.1.2 *Абстрактное-отвязывание на Пк-ОСВОБОЖДЕНИЕ*

Служба абстрактного-отвязывания отображается на службу Пк-ОСВОБОЖДЕНИЕ элемента ЭСУА. Использование параметров службы Пк-ОСВОБОЖДЕНИЕ задается в следующем подпараграфе.

8.1.2.1 *Результат*

Этот параметр должен иметь значение "положительный".

8.1.3 *Использование служб Пк-ПРЕКРАЩЕНИЕ и Пк-Пс-ПРЕКРАЩЕНИЕ*

Пользователем служб Пк-ПРЕКРАЩЕНИЕ и Пк-Пс-ПРЕКРАЩЕНИЕ является прикладной процесс.

8.2 *Отображение на службы ЭСУО*

Службы ЭПС Справочника отображаются на службы УО-ПОБУДИТЬ, УО-РЕЗУЛЬТАТ, УО-ОШИБКА, УО-ОТКАЗАТЬ-Пл, УО-ОТКАЗАТЬ-Пс, поставляемые элементом ЭСУО. Отображение нотации абстрактного-синтаксиса ЭПС Справочника на службы ЭСУО осуществляется так, как это описано в Рекомендации X.219.

9 Согласованность

В настоящем параграфе описываются требования на согласованность с настоящей Рекомендацией.

9.1 *Согласованность АПС*

Реализация АПС, претендующая на согласованность с настоящей Рекомендацией, должна удовлетворять требованиям, специфицируемым в § 9.1.1 — 9.1.3.

9.1.1 *Требования к описанию*

Должны быть указаны:

a) те операции прикладного-контекста из пКДоступаКСправочнику, которые АПС может побуждать и в отношении которых заявляется их согласованность;

b) уровень (вни)-безопасности (отсутствие, простая, строгая), в отношении которой (ых) заявляется их согласованность.

9.1.2 *Статические требования*

АПС должен:

a) обеспечивать прикладной-контекст из пКдоступаКСправочнику, как он определен его абстрактным синтаксисом в § 7.

9.1.3 *Динамические требования*

АПС должен:

a) согласовываться с отображениями на используемые службы, описанными в § 8.

9.2 *Согласованность САС*

Реализация САС, претендующая на согласованность с настоящей Рекомендацией, должна удовлетворять требованиям, специфицируемым в § 9.2.1 — 9.2.3.

9.2.1 *Требования к описанию*

Должны быть указаны:

a) прикладные-контексты, в отношении которых заявляется их согласованность: либо пКдоступаК Справочнику, либо системныйПКСправочника, либо они оба. При этом, если знания САС рассредоточены, в силу чего окажется, что ссылочные знания к данному САС хранятся у другого (их) САС, находящемуся (щимся) вне его ОУС, должна быть заявлена согласованность с системнымПКСправочника;

Примечание. — Прикладной контекст не должен быть разделен на части никаким другим образом; в частности, не надо требовать согласованности с конкретными портами или операциями.

b) может или нет САС функционировать в качестве САС первого уровня, как это описано в Рекомендации X.518;

c) если заявляется согласованность с прикладным-контекстом системногоПКСправочника, то обеспечивается или нет режим сцепленного выполнения операций, как это описано в Рекомендации X.518;

d) уровень (ни) безопасности (отсутствие, простая, строгая), в отношении которого (ых) заявляется их согласованность;

e) избранные типы атрибутов, описанные в Рекомендации X.520, и остальные типы атрибутов, в отношении которых заявляется их согласованность;

f) избранные классы объектов, описанные в Рекомендации X.521, и остальные классы, в отношении которых заявляется их согласованность.

9.2.2 *Статистические требования*

САС должен:

a) обеспечивать те прикладные-контексты, как они определены их абстрактным синтаксисом в § 7, в отношении которых заявляется их согласованность;

b) обеспечивать структуру информации, определенную в Рекомендации X.501 ее абстрактным синтаксисом;

c) согласовываться с требованиями к минимальным знаниям, описанными в Рекомендации X.518;

d) если для данного САС заявляется его согласованность с требованиями к САС первого уровня, то данный САС должен согласовываться с требованиями к обеспечению корневого сегмента, как это описано в Рекомендации X.518;

e) обеспечивать те типы атрибутов, для которых заявляется их согласованность, как это определяется их абстрактными синтаксисами;

f) обеспечивать те классы объектов, для которых заявляется их согласованность с настоящей Рекомендацией, как это определяется их абстрактными синтаксисами.

9.2.3 *Динамические требования*

САС должен:

a) согласовываться с правилами отображения на используемые службы, описанными в § 8;

b) согласовываться с теми процедурами распределенных операций Справочника, которые касаются отсылок и которые описаны в Рекомендации X.518;

c) если заявляется согласованность с прикладным-контекстом из пКДоступаКСправочнику, то должна обеспечиваться согласованность с процедурами Рекомендации X.518 в той их части, в которой они касаются режима отсылки ПДС;

d) если заявляется согласованность с прикладным-контекстом из системного ПКСправочника, то должна обеспечиваться согласованность с режимом отсылки взаимодействия, как он описан в Рекомендации X.518;

e) если заявляется согласованность с режимом сцепления взаимодействия, то должна обеспечиваться согласованность с режимом сцепления взаимодействия, как он описан в Рекомендации X.518.

Примечание. — Только в этом последнем случае от САС требуется способность побуждать операции, используя системныйПКСправочника.

ПРИЛОЖЕНИЕ A

(к Рекомендации X.519)

Описание ПДС на НАС.1

Данное Приложение является составной частью Рекомендации.

Данное Приложение содержит НАС.1-модуль ПротоколДоступаКСправочнику, в который включены все НАС.1-определения типов и значений, введенные в настоящем Справочнике.

```
ПротоколДоступаКСправочнику { joint-iso-ccitt ds(5) modules(1) dap(11)}
DEFINITIONS ::=
BEGIN

EXPORTS
    пКДОступаКСправочнику, эПСЧтения, эпсПоиска, ЭПСМодификации;

IMPORTS
абстрактнуюСлужбу
    FROM    ПолезныеОпределения
            { joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, эСУА
    FROM    Расширение-Нотации-удаленных-Операций
            { joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    ид-пк-пКДоступаКСправочнику, ид-эпс-эПСЧтения, ид-эпс-эПСПоиска,
    ид-эпс-эПСМодификации, ид-ас-эсуа, ид-ас-АСДоступаКСправочнику ид-ас-эсуа
    FROM    ИдентификаторыОбъектовПротокола
            { joint-iso-ccitt ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    ПривязываниеКСправочнику, ОтвязываниеОтСправочника, Чтение, Сравнение, Отказ, Список,
    Поиск, ДобавлениеСтатьи, УдалениеСтатьи, МодификацияСтатьи, МодификацияОВИ, Отказано,
    НевыполненныйОтказ,
    ОшибкаАтрибута, ОшибкаИмени, Отсылка, ОшибкаБезопасности, ОшибкаСлужбы,
    ОшибкаОбновления
    FROM    АбстрактнаяСлужбаСправочника
            абстрактнаяСлужбаСправочника;

-- прикладные контексты --

пКДоступаКСправочнику
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS { эСУА}
        BIND ПривязываниеКСправочнику
```

```
    UNBIND ОтвязываниеОтСправочника
        REMOTE OPERATIONS {эСУО}
            INITIATOR CONSUMER OF {эПСЧтения, эПСПоиска, эПСМодификации}
        ABSTRACT SYNTAXES{
            ид-ас-эсуа, ид-ас-АСДоступаКСправочнику}
    ::= ид-пк-пКДоступаКСправочнику


-- ЭПС Чтения --

эПСЧтения
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {чтение, сравнение, отказ}
    ::= ид-эпс-ЭПСЧтения


-- ЭПС Поиска --

эПСПоиска
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {список, поиск}
    ::= ид-эпс-ЭПСПоиска


-- ЭПС Модификации --

эПСМодификации
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            { добавлениеСтатьи, удалениеСтатьи,
            модификацияСтатьи, МодификацияОВИ}
    ::= ид-эпс-ЭПСМодификации


-- Удаленные операции --

чтение              Чтение              ::= 1

сравнение           Сравнение           ::= 2

отказ               Отказ               ::= 3

список              Список              ::= 4

поиск               Поиск               ::= 5

добавлениеСтатьи    ДобавлениеСтатьи    ::= 6

удалениеСтатьи      УдалениеСтатьи      ::= 7

модификацияСтатьи   МодификацияСтатьи   ::= 8

модификацияОВИ      МодификацияОВИ      ::= 9

-- УдаленныеОшибки --

ошибкаАтрибута      ОшибкаАтрибута      ::= 1

ошибкаИмени         ОшибкаИмени         ::= 2

ошибкаСлужбы        ОшибкаСлужбы        ::= 3

отсылка             Отсылка             ::= 4

отказано            Отказано            ::= 5

ошибкаБезопасности  ОшибкаБезопасности  ::= 6

невыполненныйОтказ  НевыполненныйОтказ  ::= 7

ошибкаОбновления    ОшибкаОбновления    ::= 8

END
```

ПРИЛОЖЕНИЕ B

(к Рекомендации X.519)

Описание СПС на НАС.1

Данное Приложение является составной частью Рекомендации.

Данное Приложение содержит НАС.1-модуль СистемныйПротоколСправочника, в который включены все НАС.1-определения типов и значений, введенные в настоящей Рекомендации.

```
СистемныйПротоколСправочника { joint-iso-ccitt ds(5) modules(1) dsp(12)}
DEFINITIONS ::=
BEGIN

EXPORTS
    системныйПКСправочника, эПССцепленногоЧтения, эПССцепленногоПоиска, эПССцепленнойМоди-
    фикации;

IMPORTS
    распределенныеОперации, абстрактныеСлужбыСправочника
    FROM    ПолезныеОпределения
            { joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, эСУА
    FROM    Расширение-Нотации-Удаленных-Операций
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    ид-пк-системныйПКСправочника, ид-эпс-ЭПССцепленногоЧтения,
    ид-эпс-ЭПССцепленногоПоиска, ид-эпс-ЭПССцепленнойМодификации,
    ид-ас-системнаяАССправочника, ид-ас-эсуа;
    FROM    ИдентификаторыОбъектовПротокола
            { joint-iso-ccitt ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    Отказано, ОшибкаАтрибута, НевыполненныйОтказ,
    ОшибкаИмени, САСОтсылка, ОшибкаБезопасности, ОшибкаСлужбы, ОшибкаОбновления
    FROM    АбстрактнаяСлужбаСправочникаабстрактнаяСлужбаСправочника

    САСПривязывание, САСОтвязывание,
    СцепленноеЧтение, СцепленноеСравнение, СцепленныйОтказ,
    СцепленныйСписок, СцепленныйПоиск,
    СцепленноеДобавлениеСтатьи, СцепленноеУдалениеСтатьи, СцепленнаяМодификацияСтатьи,
    СцепленнаяМодификацияСтатьи, САСОтсылка
    FROM    РаспределенныеОперации
            распределенныеОперации;


-- Прикладные контексты --

системныйПКСправочника
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS { эСУА }
        BIND САСПривязывание
        UNBIND САСОтвязывание
        REMOTE OPERATIONS { эСУО}
            OPERATIONS OF {
                эПССцепленногоЧтения, эПССцепленногоПоиска, эПССцепленнойМодификации}
        ABSTRACT SYNTAXES{
            ид-ас-эсуа, ид-ас-системнаяАССправочника}
    ::={ид-пк-системныйПКСправочника}


-- ЭПС Сцепленного Чтения --

эПССцепленногоЧтения
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS    {сцепленноеЧтение, сцепленноеСравнение, сцепленныйОтказ}
    ::= ид-эпс-ЭПССцепленногоЧтения
```

```
-- ЭПС СцепленногоПоиска --
эПССцепленногоПоиска
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS { сцепленныйСписок, сцепленныйПоиск}
    :: = ид-эпс-ЭПССцепленногоПоиска

-- ЭПС Сцепленной Модификации --

эПССцепленнойМодификации
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {сцепленноеДобавлениеСтатьи, сцепленноеУдалениеСтатьи,
                    сцепленнаяМодификацияСтатьи, сцепленнаяМодификацияОВИ}

    :: = ид-эпс-ЭПССцепленнойМодификации

-- Удаленные операции --

сцепленноеЧтение                СцепленноеЧтение                :: = 1
сцепленноеСравнение             СцепленноеСравнение             :: = 2
сцепленныйОтказ                 СцепленныйОтказ                 :: = 3
сцепленныйСписок                СцепленныйСписок                :: = 4
сцепленныйПоиск                 СцепленныйПоиск                 :: = 5
сцепленноеДобавлениеСтатьи      СцепленноеДобавлениеСтатьи      :: = 6
сцепленноеУдалениеСтатьи        СцепленноеУдалениеСтатьи        :: = 7
сцепленнаяМодификацияСтатьи     СцепленнаяМодификацияСтатьи     :: = 8
сцепленнаяМодификацияОВИ        СцепленнаяМодификацияОВИ        :: = 9

-- Удаленные Ошибки --

ошибкаАтрибута      ОшибкаАтрибута      :: = 1
ошибкаИмени         ОшибкаИмени         :: = 2
ошибкаСлужбы        ОшибкаСлужбы        :: = 3
отказано            Отказано            :: = 5
ошибкаБезопасности  ошибкаБезопасности  :: = 6
невыполненныйОтказ  НевыполненныйОтказ  :: = 7
ошибкаОбновления    ОшибкаОбновления    :: = 8
сасОтсылка          САСОтсылка          :: = 9

END
```

ПРИЛОЖЕНИЕ С

(к Рекомендации X.519)

Ссылочные определения идентификаторов объектов Протокола

Данное Приложение является составной частью Рекомендации.

Данное Приложение содержит все НАС.1-определения Идентификаторов Объектов, присвоенные в настоящей Рекомендации, в форме НАС.1-модуля ИдентификаторыОбъектовПротоколов.

```
ИдентификаторыОбъектовПротоколов { joint-iso-ccitt ds(5) modules(1) protocolObjectIdentifiers(4)}
DEFINITIONS :: =
BEGIN
```

```
EXPORTS
    ид-пк-ПКдоступаКСправочнику, ид-пк-СистемныйПКСправочника, ид-эпс-ЭПСЧтения,
    ид-эпс-ЭПСПоиска,
    ид-эпс-ЭПСМодификации, ид-эпс-ЭПССцепленногоЧтения,
    ид-эпс-ЭПССцепленногоПоиска, ид-эпс-ЭПССцепленнойМодификации, ид-ас-эсуа,
    ид-ас-АСДоступаКСправочнику, ид-ас-СистемнаяАССправочника;

IMPORTS
    ид-пк, ид-эпс, ид-ас
    FROM    ПолезныеОпределения
            { joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0) } ;

-- Прикладные контексты --

ид-пк-ПКДоступаКСправочнику        OBJECT IDENTIFIER ::= { ид-пк 1}

ид-пк-системныйПКСправочника       OBJECT IDENTIFIER ::= { ид-пк 2}

-- ЭПС --

ид-эпс-ЭПСЧтения                   OBJECT IDENTIFIER ::= { ид-эпс 1}

ид-эпс-ЭПСПоиска                   OBJECT IDENTIFIER ::= { ид-эпс 2}

ид-эпс-ЭПСМодификации              OBJECT IDENTIFIER ::= { ид-эпс 3}

ид-эпс-ЭПССцепленногоЧтения        OBJECT IDENTIFIER ::= { ид-эпс 4}

ид-эпс-ЭПССцепленногоПоиска        OBJECT IDENTIFIER ::= { ид-эпс 5}

ид-эпс-ЭПССцепленнойМодификации    OBJECT IDENTIFIER ::= { ид-эпс 6}

-- АС --

ид-ас-АСДоступаКСправочнику        OBJECT IDENTIFIER ::= { ид-ас 1}

ид-ас-СистемнаяАССправочника       OBJECT IDENTIFIER ::= { ид-ас 2}

ид-ас-эсуа                         OBJECT IDENTIFIER ::=
                                   { joint-iso-ccitt association-control(2) abstract-syntax(1) apdus(0)
                                   version1(1)}

END
```

Рекомендация X.520

СПРАВОЧНИК — ИЗБРАННЫЕ ТИПЫ АТРИБУТОВ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

1) Рекомендация X.520 и ISO 9594-6 "Системы обработки информации — Взаимосвязь открытых систем — Справочник — Избранные типы атрибутов" были разработаны в тесном сотрудничестве и технически совместимы.

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем, совместно с хранимой ими справочной информацией, может рассматриваться как объединенное целое, называемое *Справочником*. Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные процессы, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль по взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых разными производителями;

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 В настоящей Рекомендации определяется несколько типов атрибутов, которые могут оказаться полезными для целого ряда приложений Справочника. Одним конкретным случаем применения большого числа определенных здесь типов является формирование имен и, в частности, имен классов объектов, определенных в Рекомендации X.521. Кроме того, в настоящей Рекомендации определяются также несколько синтаксисов атрибутов.

0.4 В Приложении А, которое является составной частью настоящей Рекомендации, приводится полный модуль нотации НАС.1, содержащий все определения атрибутов и синтаксисов атрибутов.

0.5 В Приложении В, которое не является составной частью настоящей Рекомендации, приведен алфавитный указатель типов атрибутов для облегчения ссылок на них.

1 Предмет рассмотрения и область применения

1.1 В настоящей Рекомендации определяется несколько типов атрибутов, которые могут оказаться полезными для целого ряда приложений Справочника.

1.2 Типы атрибутов (и синтаксис атрибутов) распадаются на три категории, описываемые в § 1.2.1 — 1.2.3.

1.2.1 Некоторые типы (синтаксисы) атрибутов используются широким кругом приложений, а также понимаются и/или используются самим Справочником.

Примечание. — Рекомендуется использовать определяемые в настоящей Рекомендации типы (синтаксисы) атрибутов, а не разрабатывать новые, когда это оказывается подходящим для приложения.

1.2.2 Некоторые типы (синтаксисы) атрибутов утверждены в качестве международных стандартов, но применяются только в специфических приложениях. Они определяются в стандартах, связанных с соответствующими приложениями.

1.2.3 Любой Административный орган вправе определить свои собственные типы (синтаксисы) атрибутов. Они не входят в международные стандарты и становятся доступными другим органам, не подпадающим под юрисдикцию того органа, который их разработал, только на основе двусторонних соглашений.

2 Библиография

ISO 3166 "Коды представления названий стран".

Рекомендация X.121 "Международный план нумерации для сетей данных общего пользования".

Рекомендация X.208 "Взаимосвязь открытых систем — Спецификация нотации абстрактного синтаксиса (НАС.1)" (см. также ISO 8824).

Рекомендация X.501 "Справочник — Модели" (см. также ISO 9594-2).

Рекомендация X.521 "Справочник — Избранные классы объектов" (см. также ISO 9594-7).

Рекомендация E.123 "Нотация национальных и международных телефонных номеров".

3 Определения

В настоящей Рекомендации используются следующие определения, входящие в Рекомендацию X.501:

a) *тип атрибута;*

b) *синтаксис атрибута;*

c) *класс объектов.*

4 Обозначения

В настоящем документе типы атрибутов и синтаксисы атрибутов определяются с помощью специальной нотации, а именно НАС.1-макросов, введенных в Рекомендации X.501. Используются два типа макроса: **ATTRIBUTE** и **ATTRIBUTE-SYNTAX**.

Для спецификации идентификаторов объектов, присваиваемых типам атрибутов и синтаксисам атрибутов, используются два "родовых" идентификатора объектов: **типАтрибута** и **синтаксисАтрибута**. Их определение может быть найдено в Приложении В к Рекомендации X.501.

Примеры использования типов атрибутов описываются с помощью следующей неформальной нотации: приводится акроним типа атрибута, за которым следует знак равенства ("="), за которым следует сам пример значения атрибута.

РАЗДЕЛ 1 — *Избранные типы атрибутов*

5 Определения избранных типов атрибутов

В настоящей Рекомендации определяется несколько типов атрибутов, которые могут оказаться полезными для целого ряда приложений Справочника.

5.1 *Системные типы атрибутов*

Эти типы атрибутов связаны с информацией об объектах, известных Справочнику.

5.1.1 *Класс объектов*

Известный Справочнику тип атрибута *Класс объектов* специфицирован в Рекомендации X.501, за исключением присвоения ему идентификатора объекта.

классОбъектов КлассОбъектов ::= {типАтрибута 0}

5.1.2 *Имя объекта псевдонима*

Этот тип атрибута специфицирован в Рекомендации X.501, за исключением присвоения ему идентификатора объекта.

имяОбъектаПсевдонима ИмяОбъектаПсевдонима ::= {типАтрибута 1}

5.1.3 *Информационные знания*

Тип атрибута *Информационные знания* специфицирует описание тех значений, накопленных и доступных для чтения человеком, которые находятся в подчинении некоторого конкретного САС.

```
информационныеЗнания ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX синтаксисКареткоНезависимыхЦепочек
    ::= {типАтрибута 2}
```

5.2 *Типы атрибутов меток*

Эти типы атрибутов касаются той информации об объектах, которая непосредственно связана с объектами с помощью присвоения им меток.

5.2.1 *Обычное имя*

Тип атрибута *Обычное имя* специфицирует идентификатор объекта. "Обычное имя" не совпадает с именем объекта в Справочнике. Это то (возможно не однозначное) имя, под которым этот объект известен в некоторой ограниченной среде (такой, как организация); оно соответствует традициям некоторой страны или культуры, с которой ассоциируется имя.

Значением атрибута "Обычное имя" является цепочка, выбранная человеком или организацией, которых это имя описывает, или организацией, ответственной за объект (устройство или прикладной элемент), описываемый этим именем. Например, в англоговорящих странах имя человека состоит из его личного титула (Гн., Гжа., Др., Проф., Сэр, Лорд), личного имени, вторых личных имен, последнего имени, указателя поколения (если таковой требуется, напр. "мл.") и наград и орденов (если таковые имеются, например, "кавалер").

Примеры:

ОИ ="Гн. Робин Лакдан МакЛеод, доктор наук"

ОИ = "Координационный Комитет"

ОИ = "Высокоскоростной модем"

Любые варианты имен объектов должны также ассоциироваться с поименованными объектами в качестве отдельных альтернативных значений данного атрибута.

Должны допускаться и другие обычные варианты, например использование второго имени вместо первого, использование деривата "Бил" вместо "Вильям" и т.д.

```
обычноеИмя ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1..вг-обычного-имени))
    ::= {типАтрибута 3}
```

5.2.2 *Фамилия*

Тип атрибута *Фамилия* специфицирует лингвистическую конструкцию, которая обычно либо наследуется человеком от его родителей, либо воспринимается в супружестве; обычно по этой "Фамилии" человек известен другим людям.

Значением атрибута "Фамилия" является цепочка, например, "МакЛеод".

```
фамилия ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-фамилии))
    ::= {типАтрибута 4}
```

5.2.3 *Порядковый номер*

Тип атрибута *Порядковый номер* специфицирует идентификатор, являющийся порядковым номером устройства.

Значением атрибута "Порядковый номер" является печатаемая цепочка.

```
порядковыйНомер ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисПечатаемойЦепочки
            (SIZE (1.. вг-порядкового-номера))
    ::= {типАтрибута 5}
```

5.3 *Географические типы атрибутов*

Эти типы атрибутов касаются географического расположения или регионов, с которыми связаны объекты.

5.3.1 *Имя страны*

Тип атрибута *Имя страны* специфицирует страну. Если название страны используется в качестве компонента имени объекта в Справочнике, то оно идентифицирует ту страну, в которой объект физически находится или с которой он ассоциируется каким-то другим важным образом.

Значением атрибута "Имя страны" является цепочка, содержащая знаки, входящие в стандарт ISO 3166.

```
имяСтраны ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ПечатаемаяЦепочка (SIZE (2)) -- только коды ISO 3166
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {типАтрибута 6}
```

Правила сопоставления значений этого типа совпадают с таковыми для синтаксисаКареткоНезависимой Цепочки.

5.3.2 *Имя местности*

Тип атрибута *Имя местности* специфицирует местность. Если "Имя местности" используется в качестве компонента имени объекта в Справочнике, то оно идентифицирует ту географическую область или местность, в которой объект либо физически находится, либо с которой ассоциируется каким-то другим важным образом.

Значением атрибута "Имя местности" является цепочка, например, ИМ — "Эдинбург".

```
имяМестности ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-имени-местности))
    ::= {типАтрибута 7}
```

5.3.3 *Имя штата или области*

Тип атрибута *Имя штата или области* специфицирует штат или область. Если имя штата или области используется в качестве компонента имени объекта в Справочнике, то оно идентифицирует тот географический подрайон, в котором объект либо физически находится, либо с которым ассоциируется каким-то важным образом.

Значением атрибута "Имя штата или области" является цепочка, например ИШ — "Огайо".

```
имяШтатаИлиОбласти ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-имени-штата))
    ::= {типАтрибута 8}
```

5.3.4 *Адрес типа "Улица"*

Тип атрибута *Адрес типа "Улица"* специфицирует в почтовом адресе пункт для локального распределения и физической доставки, то есть название улицы, площади, авеню и номер дома. Если Адрес типа "Улица" используется в качестве компонента имени объекта в Справочнике, то он идентифицирует тот пункт, в котором либо объект физически находится, либо с которым ассоциируется каким-то другим важным образом.

Значением атрибута Адрес типа "Улица" является цепочка, например "Арнулфштрассе 60".

```
адресТипаУлица ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-адреса-улицы))
    ::= {типАтрибута 9}
```

5.4 *Типы атрибутов организаций*

Эти типы атрибутов касаются организаций и могут использоваться для описания объектов в терминах тех организаций, с которыми эти объекты ассоциируются.

5.4.1 *Имя Организации*

Тип атрибута *Имя Организации* специфицирует организацию. Если "Имя Организации" используется в качестве компонента имени объекта в Справочнике, то оно идентифицирует ту организацию, с которой этот объект связан.

Значением атрибута "Имя Организации" является цепочка, выбранная самой организацией (например, ИО — "Шотландская Телекоммуникационная корпорация"). Любые варианты имени должны также ассоциироваться с поименованной организацией в качестве отдельных альтернативных значений данного атрибута.

```
имяОрганизации ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1,, вг-имени-организации))
    ::= {типАтрибута 10}
```

5.4.2 *Имя подразделения организации*

Тип атрибута *Имя подразделения организации* специфицирует подразделение организации. Если "Имя подразделения организации" используется в качестве компонента имени объекта в Справочнике, то оно идентифицирует то подразделение организации, с которым этот объект связан.

Предполагается, что подразделение организации является частью той организации, которую обозначает атрибут "имяОрганизации".

Отсюда следует, что если атрибут "Имя подразделения организации" используется в имени объекта в Справочнике, то оно должно быть связано с атрибутом "ИмяОрганизации".

Значением атрибута "Имя подразделения организации" является цепочка, выбранная той организацией, в которую входит данное подразделение (например, ПО — "Технологический отдел"). Обратите внимание на то, что обычно используемая аббревиатура "ТО" будет отдельным и альтернативным значением атрибута.

Примеры:
О — "Скоттел", ПО — "ТО"

```
имяПодразделенияОрганизации ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-имени-подразделения-организации))
    ::= {типАтрибута 11}
```

5.4.3 *Титул*

Тип атрибута *Титул* специфицирует положение или функцию объекта в организации.

Значением атрибута "Титул" является цепочка.

Пример:

Т — "Руководитель, Распределенные приложения"

```
титул ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-титула) )
    ::= {типАтрибута 12}
```

5.5 *Типы разъяснительных атрибутов*

Эти типы атрибутов касаются разъяснений (например, на естественном языке) чего-то, связанного с объектом.

5.5.1 *Описание*

Тип атрибута *Описание* специфицирует текст, описывающий связанный с ним объект.

Например, объект "Пользователи стандарта" может иметь связанное с ним описание "список рассылки информации, касающейся разработки внутрифирменных стандартов".

Значением атрибута "Описание" является цепочка.

```
описание ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1.. вг-описания) )
    ::= {типАтрибута 13}
```

5.5.2 *Указатель поиска*

Тип атрибута *Указатель Поиска* специфицирует информацию о принятом критерии поиска; эта информация может быть включена в отдельные статьи, в отношении которых можно допустить, что они окажутся подходящими базовыми-объектами для выполнения операций поиска. Такими статьями могут быть, например, статьи о странах или организациях.

Критерий поиска состоит из необязательного идентификатора класса искомых объектов и из комбинации типов атрибутов и логических операторов, используемых при конструировании фильтра. Кроме того, для каждого элемента критерия возможно установление уровня сопоставимости, например приближенной сопоставимости.

Атрибут "Указатель Поиска" может рекурентно повторяться, чтобы отразить различные виды запросов (например, поиск жителя или сотрудника), которые могут быть удовлетворены исходя из базового-объекта, из которого считывается указатель поиска.

```
указательПоиска ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        Указатель
    ::= {типАтрибута 14}

Указатель ::= SET {
    классОбъектов     [0]  OBJECT-CLASS OPTIONAL,
    критерий          [1]  Критерий}

Критерий ::=
    CHOICE {
    Тип               [0]  ЭлементКритерия,
    and               [1]  SET OF Критерий,
    or                [2]  SET OF Критерий,
    not               [3]  Критерий}

ЭлементКритерия ::=
    CHOICE {
        равенство                 [0]  типАтрибута,
        подцепочки                [1]  типАтрибута,
        большеИлиРавно            [2]  ТипАтрибута,
        меньшеИлиРавно            [3]  типАтрибута,
        приближенноеСовпадение    [4]  типАтрибута }
```

Пример: ниже приводится потенциальное значение атрибута "Указатель поиска", который может быть помещен в статьи класса-объектов «Местность для указания того, как могут быть найдены статьи класса-объектов "Житель"».

```
Указатель-потенциального-жителя Указатель  ::= {
    классОбъектов житель,
    критерий and {
        тип подцепочекобычноеИмя,
        тип подцепочекадресТипаУлица}}
```

Из этого значения "Указателя" можно непосредственно сконструировать "Фильтр".

На первом шаге строится промежуточный "Фильтр":

```
промежуточный-фильтр Фильтр  :: = and {
   элемент подцепочек {
     тип обычноеИмя,
     цепочки  { любая ЦепочкаТ61 "Дюбуа"}}  ,  -- значение, предоставляемое в качестве "Обычного
     Имени"

      элемент подцепочек {
        тип адресТипа Улица,
        цепочки { любая ЦепочкаТ61 "Хуго"}}}  -- значение, предоставляемое в качестве Адреса типа
        "Улица"
```

На втором шаге строится "Фильтр" для сопоставления со статьями "Житель" поддерева:

```
фильтр-жителя Фильтр  :: = {
   and {
    элемент равенство {
      классОбъектов,
      OBJECT-CLASS Житель } ,
    промежуточный-фильтр }}
```

5.5.3 *Категория бизнеса*

Тип атрибута *Категории бизнеса* специфицирует информацию, связанную с занятиями некоторой группы объектов, например людей. Этот атрибут предоставляет, например, возможность обратиться к Справочнику по поводу всех людей, занимающих одну и ту же должность.

```
категорияБизнеса ATTRIBUTE
     WITH ATTRIBUTE-SINTAX
          синтаксисКареткоНезависимойЦепочки
            SIZE  (1. . .вг-категория-бизнеса))
      :: = {типАтрибута 15}
```

5.6 *Типы атрибутов почтовой адресации*

Эти типы атрибутов касаются информации, требующейся для обеспечения физической доставки объекту.

5.6.1 *Почтовый адрес*

Тип атрибута *Почтовый адрес* специфицирует адресную информацию, необходимую почтовому ведомству для физической доставки почтовых сообщений указанным объектам.

Значение атрибута "Почтовый адрес" обычно состоит из избранных атрибутов, взятых из Неформатизированного Почтового адреса СОС (версии 1) в соответствии с Рекомендацией F.401. Оно не должно превышать 6 строк по 30 знаков в каждой, включая сюда и Почтовое имя страны. Обычно, почтовый адрес содержит такие данные, как имя адресата, адрес типа "улица", город, штат или область, почтовый код и, возможно, номер почтового ящика; все это зависит от специфических требований адресуемого объекта.

```
почтовыйАдрес ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX ПочтовыйАдрес
     MATCHES FOR EQUALITY
     :: =  {типАтрибута 16}

ПочтовыйАдрес  :: = SEQUENCE SIZE (1...вг-почтовой-линии)) OF CHOICE{
     ЦепочкаТ61  (SIZE (1...вг-почтовой-цепочки)),
     ПечатаемаяЦепочка  (SIZE (1...вг-почтовой-цепочки))}
```

Правила сопоставления значений этого типа совпадают с таковыми для синтаксисаКареткоНезависимойЦепочки.

5.6.2 *Почтовый код*

Тип атрибута *Почтовый код* специфицирует почтовый код адресуемого объекта. Если используется значение этого атрибута, то оно будет составной частью почтового адреса.

Значением атрибута "Почтовый код" является цепочка.

```
почтовыйКод ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE)(1...вг-почтового-кода))
    ::= { типАтрибута 17}
```

5.6.3 *Почтовый ящик*

Тип атрибута *Почтовый ящик* специфицирует почтовый ящик, по которому объект получает физическую почтовую доставку. Если используется значение этого атрибута, то оно будет составной частью почтового адреса.

```
почтовыйЯщик  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1...вг-почтового-ящика))
    ::= { типАтрибута 18}
```

5.6.4 *Наименование офиса физической доставки*

Тип атрибута *Наименование офиса физической доставки* специфицирует название города, села и т.д., в котором расположен офис физической доставки.

Значением атрибута "Наименование офиса физической доставки" является цепочка.

```
ИмяОфисаФизическойДоставки  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE(1...вг-имени-физического-офиса))
    ::= { типАтрибута  19 }
```

5.7 *Типы атрибутов телекоммуникационной адресации*

Эти типы атрибутов касаются адресной информации, требующейся для связи с объектом, использующей телекоммуникационные средства.

5.7.1 *Телефонный номер*

Тип атрибута *Телефонный номер* специфицирует номер телефона, ассоциируемого с объектом.

Значением атрибута "Телефонный номер" является цепочка, удовлетворяющая международно согласованному формату изображения международных телефонных номеров. Рекомендация E.123 (например, "+44 582 10101")

```
ТелефонныйНомер ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисТелефонногоНомера
    ::= { типАтрибута 20}
```

5.7.2 *Телексный номер*

Тип атрибута *Телексный номер* специфицирует номер телекса, код страны и код телексного терминала для обратной связи, ассоциируемых с объектом.

```
ТелексныйНомер ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ТелексныйНомер
    ::= { типАтрибута 21}
```

```
ТелексныйНомер :: = SEQUENCE {
    ТелексныйНомер . Печатаемая Цепочка
        (SIZE (1..вг-телексного-номера)),
    КодСтраны          Печатаемая Цепочка
        (SIZE (1..вг-кода-страны)),
    ОбратнаяСвязь      ПечатаемаяЦепочка
        (SIZE (1..вг-обратной-связи))}
```

5.7.3 *Идентификатор телетексного терминала*

Тип атрибута *Идентификатор телетексного терминала* специфицирует Идентификатор телетексного терминала (и, возможно, параметры), того телетексного терминала, который ассоциируется с объектом.

Значением атрибута "Идентификатор телетексного терминала" является цепочка, удовлетворяющая Рекомендации F.200 МККТТ и, возможно, набор, компоненты которого соответствуют Рекомендации Т.62.

```
идентификаторТелетексногоТерминала ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ИдентификаторТелетексногоТерминала
    :: = {типАтрибута 22}

ИдентификаторТелетексногоТерминала :: = SEQUENCE {
    телетексныйТерминал  ПечатаемаяЦепочка
        (SIZE (1..вг-ид-телетексного терминала)),
    параметры ТелетексныеНеОсновныеПараметры
        OPTIONAL}
```

5.7.4 *Номер факсимильного телефона*

Тип атрибута *Номер факсимильного телефона* специфицирует телефонный номер факсимильного терминала (и, возможно, его параметры), который ассоциируется с объектом.

Значением атрибута "Номер факсимильного телефона" является цепочка, удовлетворяющая международно согласованному формату изображения международных телефонных номеров (например, "+81 3 347 7418") и необязательная цепочка битов (формат которой соответствует Рекомендации Т.30).

```
номерФаксимильногоТелефона ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        НомерФаксимильногоТелефона
    ::= {типАтрибута 23}

НомерФаксимильногоТелефона :: = SEQUENCE {
    телефонныйНомер   ПечатаемаяЦепочка
        (SIZE (1..вг-телефонного-номера)),
    параметры ФаксимильныеНеОсновныеПараметрыГруппы3
        OPTIONAL}
```

5.7.5 *Адрес X.121*

Тип атрибута *Адрес X.121* специфицирует ассоциируемый с объектом адрес, как он определен в Рекомендации МККТТ X.121.

```
x121Адрес  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ЦифроваяЦепочка
            (SIZE (1..вг-x121-адреса)
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= { типАтрибута 24}
```

Правила сопоставления значений этого типа совпадают с таковыми для синтаксисаЦифровойЦепочки.

5.7.6 *Номер в международной ЦСИС*

Тип атрибута *Номер в международной ЦСИС* специфицирует ассоциируемый с объектом номер в международной ЦСИС.

Значением атрибута "Номер в международной ЦСИС" является цепочка, удовлетворяющая международно согласованному формату адресов в ЦСИС, описанному в Рекомендации E.164.

```
номерВМеждународнойЦСИС ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
      ЦифроваяЦепочка
        (SIZE (1..вг-цсис-адреса) )
    ::= {типАтрибута 25}
```

Правила сопоставления значений этого типа совпадают с таковыми для синтаксисаЦифровыхЦепочек.

5.7.7 *Зарегистрированный адрес*

Тип атрибута *Зарегистрированный адрес* специфицирует мнемоническое представление адреса, ассоциируемого с объектом, в некотором конкретном месте города. Это мнемоническое представление зарегистрировано в той стране, в которой находится город; оно используется в обеспечении телеграммной службы общего пользования (согласно Рекомендации F.1).

```
зарегистрированныйАдрес ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ПочтовыйАдрес
    ::= {типАтрибута 26}
```

5.7.8 *Индикатор пункта назначения*

Тип атрибута *Индикатор пункта назначения* специфицирует (в соответствии с Рекомендациями F.1 и F.3) страну и город, ассоциируемые с объектом (адресом), необходимые для обеспечения телеграммной службы общего пользования.

Значением атрибута "Индикатор пункта назначения" является цепочка.

```
индикаторПунктаНазначения ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
      ПечатаемаяЦепочка
        (SIZE (1..вг-индикатора-пункта-назначения) )
    -- только буквенные знаки
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {типАтрибута 27}
```

Правила сопоставления значений этого типа совпадают с таковыми для синтаксисаКаретконезависимой Цепочки.

5.8 *Типы атрибутов предпочтения*

Эти типы атрибутов касаются предпочтений, проявляемых объектами.

5.8.1 *Предпочитаемый метод доставки*

Тип атрибута *Предпочитаемый метод доставки* специфицирует в приоритетном порядке методы доставки, которые должны быть использованы при связи с объектом.

```
предпочитаемыйМетодДоставки ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
      SEQUENCE OF INTEGER{
            любой-метод-доставки          (0),
            сос-доставка                  (1),
            физическая-доставка           (2),
            телексная-доставка            (3),
            телетексная-доставка          (4),
            факсимильная-доставка-группы3 (5),
            факсимильная-доставка-группы4 (6),
            ма 5-терминальная-доставка    (7),
            видеотексная-доставка         (8),
            телефонная-доставка           (9)
    SINGLE VALUE

    ::= {типАтрибута 28}
```

5.9 *Типы атрибутов Приложений ВОС*

Эти типы атрибутов связаны с информацией, касающейся объектов Прикладного уровня ВОС.

5.9.1 *Адрес в уровне представлений*

Тип атрибута *Адрес в уровне представлений* специфицирует адрес в Уровне представлений тех объектов, которые являются элементами Прикладного уровня ВОС.

Значением атрибута "Адрес в уровне представлений" является тот адрес в уровне представлений, который определен в Рекомендации X.200.

```
адресВУровнеПредставлений ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        АдресВУровнеПредставлений
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::={типАтрибута 29}

АдресВУровнеПредставлений   ::= SEQUENCE{
    пклСелектор             [0] OCTET STRING OPTIONAL,
    снСелектор              [1] OCTET STRING OPTIONAL,
    тСелектор               [2] OCTET STRING OPTIONAL,
    стАдреса                [3] SET SIZE(1..MAX) OF OCTET STRING}
```

Правило сопоставления значений этого типа формулируется следующим образом: предъявленный адрес в уровне представлений совпадает с хранящимся адресом в том и только в том случае, если селекторы совпадают, а предъявленные стАдреса являются подмножествами хранящихся.

5.9.2 *Обеспечиваемый прикладной контекст*

Тип атрибута *Обеспечиваемый прикладной контекст* специфицирует идентификатор объекта того прикладного контекста, который обеспечивает объект (являющийся элементом Прикладного уровня ВОС).

```
обеспечиваемыйПрикладнойКонтекст ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисИдентификатораОбъекта
    ::= { типАтрибута 30}
```

5.10 *Типы атрибутов отношений*

Эти типы атрибутов касаются информации об объектах, которые каким-то образом связаны с некоторым конкретным объектом.

5.10.1 *Член*

Тип атрибута *Член* специфицирует группу имен, ассоциируемую с объектом.

Значением атрибута "Член" является выделенное имя.

```
член ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= { типАтрибута 31}
```

5.10.2 *Владелец*

Тип атрибута *Владелец* специфицирует имя того объекта, который в каком-то смысле ответственен за ассоциируемый с ним объект.

Значением атрибута "Владелец" является выделенное имя, которое может обозначать группу имен и которое может рекуррентно повторяться.

```
владелец ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= {типАтрибута 32}
```

5.10.3 *Сотрудник штатной должности*

Тип атрибута *Сотрудник штатной должности* специфицирует имя того объекта, который занимает некоторую штатную должность в организации.

Значением атрибута "Сотрудник штатной должности" является выделенное имя.

```
сотрудникШтатнойДолжности ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::={типАтрибута 33}
```

5.10.4 *Смотри также*

Тип атрибута *Смотри также* специфицирует имена других объектов Справочника, которые могут представлять другие аспекты (в некотором смысле) того же самого объекта реального мира.

Значением атрибута "Смотри также" является выделенное имя.

```
смотриТакже ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= {типАтрибута 34}
```

5.11 *Типы атрибутов безопасности*

Эти типы атрибутов касаются безопасности объектов или тех привилегий объектов, которые связаны с их безопасностью. Эти типы атрибутов специфицированы, за исключением присвоения им идентификаторов объектов, в Рекомендации X.509.

5.11.1 *Пароль пользователя*

```
парольПользователя ПарольПользователя
    ::= {типАтрибута 35}
```

5.11.2 *Сертификат пользователя*

```
сертификатПользователя СертификатПользователя
    ::= {типАтрибута 36}
```

5.11.3 *Сертификат CA*

```
сертификатCA СертификатCA
    ::={типАтрибута 37}
```

5.11.4 *Список отмененных руководящих органов*

```
списокОтмененныхРуководящихОрганов СписокОтмененныхРуководящихОрганов
    ::= {типАтрибута 38}
```

5.11.5 *Список отмененных сертификатов*

```
списокОтмененныхСертификатов СписокОтмененныхСертификатов
    ::={типАтрибута 39}
```

5.11.6 *Перекрестная пара сертификатов*

```
перекрестнаяПараСертификатов ПерекрестнаяПараСертификатов
    ::={типАтрибута 40}
```

РАЗДЕЛ 2 — *Синтаксис атрибутов*

6 Определения синтаксисов атрибутов

6.1 *Синтаксисы атрибутов, используемых Справочником*

6.1.1 *Неопределенный*

Синтаксис атрибута *Неопределенный* предназначен для атрибутов, значения которых Справочник предположительно сравнивать не будет.

Спецификация этого синтаксиса атрибутов для некоторого атрибута эквивалентна спецификации типа данных ANY и отсутствию правил сопоставления в макросе ATTRIBUTE для данного атрибута.

```
неопределенный ATTRIBUTE-SYNTAX
    ANY
    ::= {синтаксисАтрибутов 0}
```

6.1.2 *Выделенное имя*

Синтаксис атрибута *Выделенное имя* предназначен для атрибутов, значениями которых являются выделенные имена. Он определен, за исключением присвоения ему идентификатора объекта, в Рекомендации X.501.

```
синтаксисВыделенногоИмени СинтаксисВыделенногоИмени
    ::= {синтаксисАтрибутов 1}
```

6.1.3 *Идентификатор объекта*

Синтаксис атрибута *Идентификатор объекта* предназначен для атрибутов, значениями которых являются идентификаторы объектов. Он определен, за исключением присвоения ему идентификатора объекта, в Рекомендации X.501.

```
синтаксисИдентификаторОбъекта СинтаксисИдентификатораОбъекта
    :: = {синтаксисАтрибутов 2}
```

6.2 *Синтаксис цепочечных атрибутов*

В синтаксисах, определяемых в § 6.2.1 — 6.2.4 приведенные ниже знаки пробелов игнорируются:

— начальные знаки пробелов (то есть те знаки пробелов, которые предшествуют первому напечатанному знаку);

— замыкающие знаки пробелов (то есть те знаки пробелов, которые следуют за последним напечатанным знаком);

— несколько последовательных внутренних знаков пробелов (эти знаки пробелов рассматриваются как один знак).

Атрибуты, соответствующие этим синтаксисам, подвергаются сопоставлению таким образом, что знаки, описанные в этих правилах как несущественные, опускаются.

6.2.1 *Кареткозависимая цепочка*

Синтаксис атрибута *Кареткозависимая цепочка* предназначен для атрибутов, значениями которых являются цепочки (либо цепочки T.61, либо Печатаемые цепочки), для которых существенно положение (верхнее или нижнее) каретки при сравнении их друг с другом (например, "Данди" и "ДАНДИ" не совпадают).

```
синтаксисКареткоЗависимойЦепочки  ATTRIBUTE-SYNTAX
    CHOICE  {ЦепочкаT61, ПечатаемаяЦепочка}
    MATCHES FOR EQUALITY SUBSTRINGS
    :: = {синтаксисАтрибутов 3}
```

Две цепочки, соответствующие этому синтаксису, считаются сопоставимыми на равенство, если они имеют одинаковую длину и соответствующие знаки совпадают. Печатаемую цепочку можно сравнить с цепочкой T.61: если оба соответствующих друг другу знака входят в набор знаков Печатаемых цепочек, то сравнение выполняется обычным образом; если же знак в Цепочке T.61 не входит в набор знаков Печатаемых Цепочек, то сравнение дает отрицательный результат.

6.2.2 *Каретконезависимая цепочка*

Синтаксис атрибута *Каретконезависимых цепочек* предназначен для атрибутов, значениями которых являются цепочки (либо цепочки T.61, либо печатаемые цепочки), для которых несущественно положение (верхнее или нижнее) каретки при сравнении их друг с другом (например, "Данди" и "ДАНДИ" совпадают).

синтаксисКареткоНезависимойЦепочки ATTRIBUTE-SYNTAX
CHOICE {ЦепочкаТ61, ПечатаемаяЦепочка}
MATCHES FOR EQUALITY SUBSTRINGS
::= {синтаксисАтрибутов 4}

Правила сопоставления те же, что и для кареткозависимых цепочек, с той лишь разницей, что знаки, отличающиеся положением каретки, считаются совпадающими.

6.2.3 *Печатаемая цепочка*

Синтаксис атрибута *Печатаемых цепочек* предназначен для тех атрибутов, значениями которых являются печатаемые цепочки.

синтаксисПечатаемойЦепочки ATTRIBUTE-SYNTAX
ПечатаемаяЦепочка
MATCHES FOR EQUALITY SUBSTRINGS
::= {синтаксисАтрибутов 5}

Правила сопоставления совпадают с таковыми для синтаксиса атрибута "Кареткозависимых цепочек".

6.2.4 *Цифровая цепочка*

Синтаксис атрибута *Цифровая цепочка* предназначен для атрибутов, значениями которых являются цифровые цепочки.

синтаксисЦифровойЦепочки ATTRIBUTE-SYNTAX
ЦифроваяЦепочка
MATCHES FOR EQUALITY SUBSTRINGS
::= {синтаксисАтрибутов 6}

Правила сопоставления совпадают с таковыми для синтаксиса атрибута "Кареткозависимых цепочек" с тем исключением, что все знаки пробелов пропускаются.

6.2.5 *Каретконезависимый список*

Синтаксис атрибута *Каретконезависимого списка* предназначен для атрибутов, значениями которых являются последовательности цепочек (либо цепочек Т61, либо печатаемых цепочек), для которых несущественно положение (верхнее или нижнее) каретки при сравнении их друг с другом.

синтаксисКареткоНезависимогоСписка ATTRIBUTE-SYNTAX
SEQUENCE OF
CHOICE {ЦепочкаТ61, ПечатаемаяЦепочка}
MATCHES FOR EQUALITY SUBSTRINGS
::= {синтаксисАтрибутов 7}

Два "Каретконезависимого списка" сопоставимы на равенство тогда и только тогда, когда совпадает число цепочек в списках и соответствующие друг другу цепочки совпадают. Сопоставление последних такое же, как для синтаксиса атрибута "Каретконезависимых цепочек" (§ 6.1.3).

6.3 *Синтаксис дополнительных атрибутов*

6.3.1 *Булевский*

Синтаксис атрибута *Булевский* предназначен для атрибутов, имеющих Булевские значения (то есть представляющих "истину" или "ложь").

булевскийСинтаксис ATTRIBUTE-SYNTAX
BOOLEAN
MATCHES FOR EQUALITY
::= {синтаксисАтрибутов 8}

Два значения атрибута этого синтаксиса сопоставимы на равенство, если оба они "истина" или оба "ложь".

6.3.2 *Целочисленный*

Синтаксис атрибута *Целочисленный* предназначен для атрибутов, значениями которых являются целые числа.

```
целочисленныйСинтаксис ATTRIBUTE-SYNTAX
    INTEGER
    MATCHES FOR EQUALITY ORDERING
    ::=  {синтаксисАтрибутов 9}
```

Два значения атрибута этого синтаксиса сопоставимы на равенство, если совпадают целые числа. К этим значениям приложимы правила упорядочения целых чисел.

6.3.3 *Цепочка октетов*

Синтаксис атрибута *Цепочка октетов* предназначен для атрибутов, значениями которых являются цепочки октетов.

```
синтаксисЦепочекОктетов ATTRIBUTE-SYNTAX
    OCTET STRING
    MATCHES FOR EQUALITY SUBSTRINGS ORDERING
    ::=  {синтаксисАтрибутов 10}
```

Две цепочки этого синтаксиса совпадают, если совпадают их длины и совпадают соответствующие друг другу октеты. Порядок для этих цепочек устанавливается на основании отношения упорядоченности первых несовпавших октетов начиная от начала цепочек.

6.3.4 *Время от Гринвича*

Синтаксис атрибута *Время от Гринвича* предназначен для тех атрибутов, значения которых представляют абсолютное время.

```
синтаксисВременииОтГринвича ATTRIBUTE-SYNTAX
    ВремяОтГринвича
    MATCHES FOR EQUALITY ORDERING
    ::=  {синтаксисАтрибутов 11}
```

Два значения атрибута этого синтаксиса сопоставимы на равенство, если они изображают одно и то же время. Более раннее время считается "меньшим", чем более позднее время.

6.3.5 *Телефонные номера*

Синтаксис атрибута *Телефонный номер* предназначен для тех атрибутов, значениями которых являются номера телефонов.

```
синтаксисТелефонныхНомеров ATTRIBUTE-SYNTAX
    ПечатаемаяЦепочка
        (SIZE (1..вг-телефонного-номера))
    MATCHES FOR EQUALITY SUBSTRINGS
    :: = {синтаксисАтрибутов 12}
```

Правила сопоставления совпадают с таковыми для синтаксиса атрибута "Кареткозависимых цепочек" с тем исключением, что при сравнении все знаки пробелов и знаки "—" пропускаются.

ПРИЛОЖЕНИЕ А

(к Рекомендации X.520)

Избранные типы атрибутов на НАС.1

Данное Приложение является составной частью настоящей Рекомендации.

Данное Приложение содержит НАС.1-модуль ИзбранныетипыатрибутОв, в который включены все НАС.1-определения типов и значений, введенные в настоящей Рекомендации.

```
ИзбранныеТипыАтрибутов  {joint-iso-ccitt ds(5) modules(1)
                          selectedAttributeTypes(5)}
DEFINITIONS  ::=
BEGIN

-- экспортирует все возможное --

IMPORTS
    структуруИнформации, структуруАутентификации, типАтрибута,
    верхниеГраницы
    FROM ПолезныеОпределения          {joint-iso-ccitt ds(5) modules(1)
                                        usefulDefinitions(0)}
    ATTRIBUTE, ATTRIBUTE-SYNTAX, ТипАтрибута, OBJECT-CLASS,
    КлассОбъектов, ИмяОбъектаПсевдонима,
    СинтаксисВыделенногоИмени, СинтаксисИдентификатораОбъекта
        FROM СтруктураИнформации структураИнформации
    ФаксимильныеНеОсновныеПараметрыГруппы3,
    телетексныеНеОсновныеПараметры
        FROM СПСАбстрактныеСлужбы   {joint-ISO-CCITT mhs-motis(6)
                                      mts(3) modules(0) mts-abstract-service(1)}
    СертификатПользователя, Сертификат СА, ПерекрестнаяПараСертификатов,
    СписокОтмененныхСертификатов, СписокОтмененныхРуководящихОрганов, ПарольПользователя
        FROM СтруктураАутентификации, структураАутентификации
        вг-обратной-связи,
        вг-обычного-имени, вг-фамилии, вг-порядкового-номера,
        вг-имени-местности, вг-имени-штата,
        вг-адреса-типа-улица, вг-имени-организации,
        вг-имени-подразделения-организации, вг-титула,
        вг-описания, вг-категории-бизнеса, вг-почтовой-линии,
        вг-почтовой-цепочки, вг-почтового-кода, кг-почтового-ящика,
        вг-имени-физического-офиса, вг-телексного-номера,
        вг-кода-страны, вг-ид-телетексного-терминала,
        вг-телефонного-номера, вг-x121-адреса,
        вг-номера-в-международной-цсис, вг-индикатора-пункта-назначения,
        вг-пароля-пользователя
          FROM ВерхниеГраницы верхниеГраницы;

-- типы атрибутов --

классОбъектов КлассОбъектов  ::= {типАтрибута 0}

имяОбъектаПсевдонима  ИмяОбъектаПсевдонима  :: = {типАтрибута 1}

информационныеЗнания ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX синтаксисКареткоНезависимойЦепочки
    ::= {типАтрибута 2}

обычноеИмя  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1..вг-обычного-имени))
    ::= {типАтрибута 3}

фамилия  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
            (SIZE (1..вг-фамилии))
    ::=  {типАтрибута 4}

порядковыйНомер  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисПечатаемойЦепочки
            (SIZE (1..вг-порядкового-номера))
    ::= {типАтрибута 5}
```

```
имяСтраны ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ПечатаемаяЦепочка (SIZE (2)) -- только коды ISO 3166
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {типАтрибута 6}

имяМестности ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-имени-местности))
    ::= {типАтрибута 7}

имяШтатаИлиОбласти ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-имени-штата))
    ::= {типАтрибута 8}

адресТипаУлица ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-адреса-типа-улица))
    ::= {типАтрибута 9}

имяОрганизации ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-имени-организации))
    ::= {типАтрибута 10}

имяПодразделенияОрганизации ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-имени-подразделения-организации))
    ::= {типАтрибута 11}

титул ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-титула))
    ::= {типАтрибута 12}

описание ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-описания))
    ::={типАтрибута 13}

указательПоиска ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        Критерий
    ::= {типАтрибута 14}

Указатель ::= SET{
    классОбъектов [0] OBJECT-CLASS OPTIONAL,
        критерий [1] Критерий}

Критерий ::=
    CHOICE {
        тип [0] ЭлементКритерия,
        and [1] SET OF Критерий,
        or  [2] SET OF Критерий,
        not [3] Критерий}

ЭлементКритерия ::=
    CHOICE {
        равенство                [0] ТипАтрибута,
        подцепочки               [1] ТипАтрибута,
        большеИлиРавно           [2] ТипАтрибута,
        меньшеИлиРавно           [3] ТипАтрибута,
        приближенноеСовпадение   [4] ТипАтрибута}
```

```
категорияБизнеса ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-категории-бизнеса))
    ::= { типАтрибута 15 }

почтовыйАдрес  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ПочтовыйАдрес
    MATCHES FOR EQUALITY
    ::= { типАтрибута 16 }

ПочтовыйАдрес::= SEQUENCE SIZE (1..вг-почтовой-линии)  OF CHOICE {
        ЦепочкаТ61  (SIZE (1..вг-почтовой-цепочки)),
        ПечатаемаяЦепочка  (SIZE (1..вг-почтовой-цепочки))}

почтовыйКод ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-почтового-кода))
    ::= { типАтрибута 17 }

почтовыйЯщик ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-почтового-ящика))
    ::= { типАтрибута 18 }

имяОфисаФизическойДоставки ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисКареткоНезависимойЦепочки
          (SIZE (1..вг-имени-физического-офиса))
    ::= { типАтрибута 19 }

телефонныйНомер ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        СинтаксисТелефонныхНомеров
    ::= { типАтрибута 20 }

телексныйНомер ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ТелексныйНомер
    ::= { типАтрибута 21 }

ТелексныйНомер  ::=  SEQUENCE {
    ТелексныйНомер  ПечатаемаяЦепочка
        (SIZE (1..вг-телексного-номера)),
    кодСтраны  ПечатаемаяЦепочка
        (SIZE (1..вг-кода-страны)),
    обратнаяСвязь  ПечатаемаяЦепочка
        (SIZE (1..вг-обратной-связи))}

идентификаторТелетексногоТерминала  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ИдентификаторТелетексногоТерминала
    ::= { типАтрибута 22 }

ИдентификаторТелетексногоТерминала ::= SEQUENCE {
    телетексныйТерминал  ПечатаемаяЦепочка
        (SIZE (1..вг-ид-телетексного-терминала)),
    параметры  ТелетексныеНеОсновныеПараметры
        OPTIONAL}
номерФаксимильногоТелефона  ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        НомерФаксимильногоТелефона
    ::= { типАтрибута 23 }

НомерФаксимильногоТелефона  :: = SEQUENCE {
    телефонныйНомер  ПечатаемаяЦепочка
        (SIZE (1..вг-телефонного-номера)),
    параметры ФаксимильныеНеОсновныеПараметрыГруппы3 OPTIONAL}
```

```
x121Адрес ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ЦифроваяЦепочка
          (SIZE (1..вг-x121-адреса))
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {типАтрибута 24}

НомерВМеждународнойЦСИС ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ЦифроваяЦепочка
          (SIZE (1..вг-цсис-адреса))
    ::= {типАтрибута 25}

зарегистрированныйАдрес ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ПочтовыйАдрес
    ::= {типАтрибута 26}

индикаторПунктаНазначения ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        ПечатаемаяЦепочка
          (SIZE (1..вг-индикатора-пункта-назначения))
    -- только буквенные знаки
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {типАтрибута 27}

предпочитаемыйМетодДоставки ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        SEQUENCE OF INTEGER {
            любой-метод-доставки          (0),
            сос-доставка                  (1),
            физическая-доставка           (2),
            телексная-доставка            (3),
            телетексная-доставка          (4),
            факсимильная-доставка-группы3 (5),
            факсимильная-доставка-группы4 (6),
            ма5-терминальная-доставка     (7),
            видеотексная-доставка         (8),
            телефонная-доставка           (9)}
    SINGLE VALUE
    ::= {типАтрибута 28}

адресВУровнеПредставлений ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        АдресВУровнеПредставлений
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {типАтрибута 29}

АдресВУровнеПредставлений ::= SEQUENCE {
    пклСелектор [0] OCTET STRING OPTIONAL,
    снСелектор  [1] OCTET STRING OPTIONAL,
    тСелектор   [2] OCTET STRING OPTIONAL,
    стАдреса    [3] SET SIZE (1..MAX) OF OCTET STRING}

обеспечиваемыйПрикладнойКонтекст ATTRIBUTE
    WITH ATTRIBUTE-SUNTAX
        синтаксисИдентификатораОбъекта
    ::= {типАтрибута 30}

член ATTRIBUTE
    WITH ATRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= {типАтрибута 31}

владелец ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= {типАтрибута 32}
```

```
сотрудникШтатнойДолжности ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= {типАтрибута 33}

смотриТакже ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        синтаксисВыделенногоИмени
    ::= { типАтрибута 34 }

парольПользователя ПарольПользователя
    ::= { типАтрибута 35 }

сертификатПользователя СертификатПользователя
    ::= { типАтрибута 36 }

сертификатCA СертификатCA
    ::= { типАтрибута 37 }

списокОтмененныхРуководящихОрганов СписокОтмененныхРуководящихОрганов
    ::= {типАтрибута 38}

списокОтмененныхСертификатов СписокОтмененныхСертификатов
    ::= {типАтрибута 39}

перекрестнаяПараСертификатов ПерекрестнаяПараСертификатов
    ::= { типАтрибута 40 }


-- синтаксис атрибутов --


неопределенный ATTRIBUTE-SYNTAX
    ANY
    ::= { синтаксисАтрибутов 0}

синтаксисВыделенногоИмени СинтаксисВыделенногоИмени
    ::= {синтаксисАтрибутов 1}

синтаксисИдентификатораОбъекта СинтаксисИдентификатораОбъекта
    ::= { синтаксисАтрибутов 2}

кареткоЗависимаяЦепочка ATTRIBUTE-SYNTAX
    CHOICE { ЦепочкаT61, ПечатаемаяЦепочка}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {синтаксисАтрибутов 3}

синтаксисКареткоНезависимойЦепочки ATTRIBUTE-SYNTAX
    CHOICE { ЦепочкаT61, ПечатаемаяЦепочка }
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= { синтаксисАтрибутов 4 }

синтаксисПечатаемойЦепочки ATTRIBUTE-SYNTAX
    ПечатаемаяЦепочка
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= { синтаксисАтрибутов 5 }

синтаксисЦифровойЦепочки ATTRIBUTE-SYNTAX
    ЦифроваяЦепочка
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= { синтаксисАтрибутов 6 }

синтаксисКареткоНезависимогоСписка ATTRIBUTE-SYNTAX
    SEQUENCE OF
        CHOICE { ЦепочкаT61, ПечатаемаяЦепочка}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= { синтаксисАтрибутов 7 }

булевскийСинтаксис ATTRIBUTE-SYNTAX
    BOOLEAN
    MATCHES FOR EQUALITY
    ::= { синтаксисАтрибутов 8 }
```

```
целочисленныйСинтаксис ATTRIBUTE-SYNTAX
    INTEGER
    MATCHES FOR EQUALITY ORDERING
    ::= {синтаксисАтрибутов 9}

синтаксисЦепочекОктетов ATTRIBUTE-SYNTAX
    OCTET STRING
    MATCHES FOR EQUALITY SUBSTRINGS ORDERING
    ::= {синтаксисАтрибутов 10}

синтаксисВремениОтГринвича ATTRIBUTE-SYNTAX
    ВремяОтГринвича
    MATCHES FOR EQUALITY ORDERING
    ::= {синтаксисАтрибутов 11}

синтаксисТелефонныхНомеров ATTRIBUTE-SYNTAX
    ПечатаемаяЦепочка
        (SIZE(1.. вг-телефонного-номера))
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {синтаксисАтрибутов 12}
```

ПРИЛОЖЕНИЕ B

(к Рекомендации X.520)

Указатель типов атрибутов и синтаксисов

* Известен самому Справочнику и используется им.

ТИПЫ АТРИБУТОВ

О	Обеспечиваемый прикладной контекст	§ 5.9.2
	Обычное имя	§ 5.2.1
	Описание	§ 5.5.1
П	Пароль использователя	§ 5.11.1
	Перекрестная пара сертификатов	§ 5.11.6
	Порядковый номер	§ 5.2.3
	Почтовый адрес	§ 5.6.1
	Почтовый код	§ 6.6.2
	Почтовый ящик	§ 5.6.3
	Предпочитаемый метод доставки	§ 5.8.1
С	Сертификат пользователя	§ 5.11.2
	Сертификат СА	§ 5.11.3
	Смотри также	§ 5.10.4
	Сотрудник штатной должности	§ 5.10.3
	Список отмененных руководящих органов	§ 5.11.4
	Список отмененных сертификатов	§ 5.11.5
Т	Телексный номер	§ 5.7.2
	Телефонный номер	§ 5.7.1
	Титул	§ 5.4.3
У	Указатель поиска	§ 5.5.2
Ф	Фамилия	§ 5.2.2
Х	X.121 Адрес	§ 5.7.5
Ч	Член	§ 5.10.1

ПРИЛОЖЕНИЕ С

(к Рекомендации X.520)

Верхние границы

Данное Приложение является составной частью Рекомендации.

```
ВерхниеГраницы    { joint-ISO-CCITT ds(5) modules(1)
                    upperBounds(10)}

DEFINITIONS ::=
BEGIN

-- Экспортирует все возможное --

вг-обратной-связи                    INTEGER ::= 8

вг-обычного-имени                    INTEGER ::= 64

вг-фамилии                           INTEGER ::= 64

вг-порядкового-номера                INTEGER ::= 64

вг-имени-местности                   INTEGER ::= 128

вг-имени-штата                       INTEGER ::= 128

вг-адреса-типа-улица                 INTEGER ::= 128

вг-имени-организации                 INTEGER ::= 64

вг-имени-подразделения-организации   INTEGER ::= 64

вг-титула                            INTEGER ::= 64
```

вг-описания	**INTEGER ::= 1024**
вг-категории-бизнеса	**INTEGER ::= 128**
вг-почтовой-линии	**INTEGER ::= 6**
вг-почтовой-цепочки	**INTEGER ::= 30**
вг-почтового-кода	**INTEGER ::= 40**
вг-почтового-ящика	**INTEGER ::= 40**
вг-имени-физического-офиса	**INTEGER ::= 128**
вг-телексного-номера	**INTEGER ::= 14**
вг-кода-страны	**INTEGER ::= 4**
вг-ид-телетексного-терминала	**INTEGER ::= 24**
вг-телефонного-номера	**INTEGER ::= 32**
вг-x121-адреса	**INTEGER ::= 15**
вг-номера-в-международной-цсис	**INTEGER ::= 16**
вг-индикатора-пункта-назначения	**INTEGER ::= 128**
вг-пароля-пользователя	**INTEGER ::= 128**

END

Рекомендация X.521

СПРАВОЧНИК — ИЗБРАННЫЕ КЛАССЫ ОБЪЕКТОВ[1)]

(Мельбурн, 1988 г.)

СОДЕРЖАНИЕ

---

1) Рекомендация X.521 и ISO 9594-7 "Системы обработки информации — Взаимосвязь открытых систем — Справочник — Выделенные классы объектов" были разработаны в тесном сотрудничестве и технически совместимы.

РАЗДЕЛ 1 – *Избранные классы объектов*

0 Введение

0.1 Настоящий документ наряду с другими документами этой серии был разработан, чтобы облегчить взаимосвязь систем обработки информации с целью обеспечения справочных служб. Совокупность всех таких систем совместно с хранимой ими справочной информацией может рассматриваться как объединенное целое, называемое *Справочником*. Информация, хранимая в Справочнике, совокупно называемая Информационной базой Справочника (ИБС), обычно используется для облегчения связи между объектами, с объектами или относительно объектов; примерами объектов могут служить прикладные процессы, люди, терминалы или списки рассылки.

0.2 Справочник играет существенную роль во взаимосвязи открытых систем; его назначение заключается в обеспечении (при минимальных технических соглашениях вне самих стандартов взаимосвязи) взаимосвязи систем обработки информации:

— поставляемых разными производителями;

— находящихся под различным управлением;

— различной степени сложности;

— различных поколений.

0.3 В настоящей Рекомендации определяются (в разделе 1) несколько наборов атрибутов и классов объектов, которые могут оказаться полезными для целого ряда приложений Справочника.

0.4 В Приложении А, которое является составной частью настоящего стандарта, приводится НАС.1-модуль, содержащий все имеющиеся в настоящем документе определения типов и значений.

0.5 Приложение В, не являющееся составной частью настоящей Рекомендации, содержит некоторые правила обычных способов именования и структурирования, которые могут быть использованы, а могут и не быть использованы Административным руководящим органом.

1 Предмет рассмотрения и область применения

1.1 В настоящей Рекомендации определяются несколько избранных наборов атрибутов и классов объектов, которые могут оказаться полезными для целого ряда приложений Справочника. Определение набора атрибутов содержит идентификацию включенных в него атрибутов; эти определения облегчают определения классов объектов. Определение класса объектов содержит список некоторого числа типов атрибутов, относящихся к объектам данного класса, и может содержать присваиваемый классу идентификатор объекта. Эти определения используются Административным органом, ответственным за обеспечение Справочника.

1.2 Любой Административный орган может определять свои классы и подклассы объектов, требующиеся ему для любых его целей.

Примечание 1. — Эти определения могут использовать, а могут и не использовать нотацию, специфицированную в Рекомендации X.501.

Примечание 2. — Рекомендуется использовать определяемые в настоящей Рекомендации классы объектов или выделяемые из них подклассы, а не разрабатывать новые классы, если семантика определяемых здесь классов оказывается подходящий для конкретного Приложения.

1.3 Административный орган может обеспечивать часть или все выделенные классы объектов и, кроме того, может сам добавлять классы объектов.

Каждый Административный орган обязан обеспечивать те классы объектов, которые Справочник использует для своих собственных нужд (классы объектов "вершина", "псевдоним" и CAC).

2 Библиография

Рекомендация X.200 "Взаимосвязь открытых систем — Основная эталонная модель" (см. также ISO 7498).

Рекомендация X.500 "Справочник — Обзор концепций, моделей и служб" (см. также ISO 9594-1).

Рекомендация X.501 "Справочник — Модели" (см. также ISO 9594-2).

3 Определения и сокращения

3.1 *Определения эталонной модели ВОС*

В настоящей Рекомендации используются следующие определения из Рекомендации X.200:

a) *прикладной-элемент;*
b) *прикладной-процесс.*

3.2 *Определения модели Справочника*

В настоящей Рекомендации используются следующие определения из Рекомендации X.501:

a) *атрибут;*
b) *тип атрибута;*
c) *информационное дерево Справочника (ИДС);*
d) *системный агент Справочника (САС);*
e) *набор атрибутов;*
f) *статья;*
g) *имя;*
h) *класс объектов;*
i) *подкласс.*

4 Обозначения

В настоящей Рекомендации классы объектов определяются с помощью введенной в Рекомендации X.501 специальной нотации, а именно НАС.1-макроса **OBJECT-CLASS**. Для спецификации идентификаторов объектов, присваиваемых классам объектов, используется "родовой" идентификатор объектов классОбъектов. Его определение приведено в Приложении В к указанной Рекомендации.

В настоящей Рекомендации наборы атрибутов определяются с помощью введенной в Рекомендации X.501 специальной нотации, а именно НАС.1-макроса ATTRIBUTE-SET. Для спецификации идентификаторов объектов, присваиваемых определениям наборов атрибутов, используется "родовой" идентификатор объектов наборАтрибутов. Его определение приведено в Приложении В к указанной Рекомендации.

РАЗДЕЛ 1 – *Избранные классы объектов*

5 Определения полезных наборов атрибутов

5.1 *Телекоммуникационный набор атрибутов*

Этот набор атрибутов используется для определения тех атрибутов, которые обычно используются в коммерческой связи.

```
телекоммуникационныйНаборАтрибутов ATTRIBUTE-SET
    CONTAINS {
        номерФаксимильногоТелефона,
        ЦСИСАдрес,
        телефонныйНомер,
        идентификаторТелетексногоТерминала,
        телексныйНомер, X121Адрес,
        предпочитаемыйМетодДоставки,
        индикаторПунктаНазначения,
        зарегистрированныйАдрес }
        ::= { наборАтрибутов 0 }
```

5.2 *Почтовый набор атрибутов*

Этот набор атрибутов используется для определения тех атрибутов, которые непосредственно связаны с доставкой почты.

```
почтовыйНаборАтрибутов ATTRIBUTE-SET
    CONTAINS {
        имяОфисаФизическойДоставки,
        почтовыйАдрес,
        почтовыйКод,
        почтовыйЯщик,
        адресТипаУлица }
        ::= { наборАтрибутов 1 }
```

5.3 *Набор атрибутов местности*

Этот набор атрибутов используется для определения тех атрибутов, которые обычно используются для индикации местоположения объекта при его поиске.

```
наборАтрибутовМестности ATTRIBUTE-SET
    CONTAINS {
        имяМестности,
        имяШтатаИлиОбласти,
        адресТипаУлица }
        ::= { наборАтрибутов 2 }
```

5.4 *Набор атрибутов организаций*

Этот набор атрибутов используется для определения тех атрибутов, которыми типично владеют организации или подразделения организации.

```
наборАтрибутовОрганизации ATTRIBUTE-SET
    CONTAINS {
        описание,
        наборАтрибутовМестности,
        почтовыйНаборАтрибутов,
        телекоммуникационныйНаборАтрибутов,
        категорияБизнеса,
        смотриТакже,
        указательПоиска
        парольПользователя }
        ::= { наборАтрибутов 3 }
```

6 Определения избранных классов объектов

6.1 *Вершина*

Класс объектов *Вершина* определен в Рекомендации X.501, за исключением присвоения ему идентификатора объекта. Любой класс объектов является подклассом этого класса.

```
вершина Вершина ::= {классОбъектов 0}
```

6.2 *Псевдоним*

Класс объектов *Псевдоним* определен в Рекомендации X.501, за исключением присвоения ему идентификатора объекта. Классы статей псевдонимов могут быть выведены из класса "Псевдоним".

```
псевдоним Псевдоним ::= {классОбъектов 1}
```

6.3 *Страна*

Класс объектов *Страна* используется для определения статей стран в ИДС.

```
страна OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        имяСтраны   }
    MAY CONTAIN {
        описание,
        указательПоиска }
    ::= {классОбъектов 2}
```

6.4 *Местоположение*

Класс объектов *Местоположение* используется для определения местоположения в ИДС.

```
местоположение OBJECT-CLASS
    SUBCLASS OF вершина
    MAY CONTAIN{
        описание,
        имяМестности,
        имяШтатаИлиОбласти,
        указательПоиска,
        смотриТакже,
        адресТипаУлица}
    ::= {классОбъектов 3}
```

Должно присутствовать по меньшей мере либо Имя местности, либо Имя штата или области.

6.5 *Организация*

Класс объектов *Организация* используется для определения статей организаций в ИДС.

```
организация OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN  {
        имяОрганизации}
    MAY CONTAIN {
        наборАтрибутовОрганизации}
    ::= {классОбъектов 4}
```

6.6 *Подразделение организации*

Класс объектов *Подразделение организации* используется для определения статей ИДС, представляющих подразделения организаций.

```
подразделениеОрганизации OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        имяПодразделенияОрганизации}
    MAY CONTAIN {
        наборАтрибутовОрганизации}
    ::= {классОбъектов 5}
```

6.7 *Человек*

Класс объектов *Человек* используется для определения статей ИДС, дающих общее описание людей.

```
человек OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя,
        фамилия}
    MAY CONTAIN {
        описание,
        смотриТакже,
        телефонныйНомер,
        парольПользователя }
    ::= { классОбъектов 6}
```

6.8 *Сотрудник организации*

Класс объектов *Сотрудник организации* используется для определения статей ИДС, представляющих людей либо взятых на работу некоторой Организацией, либо каким-то иным существенным образом связанных с нею.

```
сотрудникОрганизации OBJECT-CLASS
    SUBCLASS OF человек
    MAY CONTAIN {
        наборАтрибутовМестности,
        имяПодразделенияОрганизации,
        почтовыйНаборАтрибутов,
        телекоммуникационныйНаборАтрибутов,
        титул}
    ::= { классОбъектов 7}
```

6.9 *Штатная должность организации*

Класс объектов *Штатная должность организации* используется для определения штатных должностей организаций, то есть положения или роли внутри организации. Считается, что обычно штатная должность организации занята некоторым конкретным сотрудником организации. Однако за время своего существования штатная должность организации может быть последовательно занята различными сотрудниками организации. Как правило, штатная должность организации может быть занята либо человеком, либо каким-нибудь неодушевленным элементом.

```
штатнаяДолжностьОрганизации OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя }
    MAY CONTAIN {
        описание,
        наборАтрибутовМестности,
        имяПодразделенияОрганизации,
        почтовыйНаборАтрибутов,
        предпочитаемыйМетодДоставки,
        сотрудникШтатнойДолжности,
        смотриТакже,
        телекоммуникационныйНаборАтрибутов }
    ::= { классОбъектов 8}
```

6.10 *Группа имен*

Класс объектов *Группа имен* используется для определения статей, представляющих неупорядоченный набор имен, изображающих индивидуальные объекты или другие группы имен. Членство в группе статичное, то есть оно может быть изменено только явным действием Административного органа, а не динамически определяться при каждом обращении к группе.

Состав членов группы может быть сведен к набору имен индивидуальных объектов; для этого достаточно заменить каждую группу ее членами. Этот процесс должен быть рекуррентно продолжен до тех пор, пока все групповые члены не будут заменены на индивидуальные объекты.

```
группаИмен OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя,
        член }
    MAY CONTAIN{
        описание,
        имяОрганизации,
        имяПодразделенияОрганизации,
        владелец,
        смотриТакже,
        категорияБизнеса}
    ::={классОбъектов 9}
```

6.11 *Житель*

Класс объектов *Житель* используется для определения статей, представляющих человека в среде его проживания.

```
житель OBJECT-CLASS
    SUBCLASS OF человек
    MUST CONTAIN{
        имяМестности }
    , MAY CONTAIN {
        наборАтрибутовМестности,
        почтовыйНаборАтрибутов,
        предпочитаемыйМетодДоставки,
        телекоммуникационныйНаборАтрибутов,
        категорияБизнеса  }
    ::={классОбъектов 10}
```

6.12 *Прикладной процесс*

Класс объектов *Прикладной процесс* используется для определения статей, представляющих прикладные процессы. Прикладной процесс является элементом реальной открытой системы, выполняющим обработку информации для конкретного приложения (см. Рекомендацию X.200).

```
прикладнойПроцесс OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя}
    MAY CONTAIN {
        описание,
        имяМестности,
        имяПодразделенияОрганизации,
        смотриТакже }
    ::={классОбъектов 11}
```

6.13 *Прикладной элемент*

Класс объектов *Прикладной элемент* используется для определения статей, представляющих прикладные элементы. Прикладной элемент содержит причастные к ВОС аспекты прикладного процесса.

```
прикладнойЭлемент OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя,
        адресВУровнеПредставлений}
    MAY CONTAIN{
        описание,
        названиеМестности,
```

имяОрганизации,
имяПодразделенияОрганизации,
смотриТакже,
обеспечиваемыйПрикладнойКонтекст }
::= { классОбъектов 12 }

Примечание. — Если Прикладной элемент представлен как объект Справочника, отличный от Прикладного процесса, то для хранения значения квалификатора прикладного элемента используется атрибут обычноеИмя.

6.14 *CAC*

Класс объектов *CAC* используется для определения статей, представляющих CAC. CAC определен в Рекомендации X.501.

```
cAC OBJECT-CLASS
    SUBCLASS OF прикладнойЭлемент
    MAY CONTAIN{
        информационныеЗнания }
    ::= { классОбъектов 13 }
```

6.15 *Устройство*

Класс объектов *Устройство* используется для определения статей, представляющих устройства. Устройства являются физическими узлами, способными к связи, как например, модемы, дисководы и т.д.

```
устройство OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя}
    MAY CONTAIN{
        описание,
        имяМестности,
        имяОрганизации,
        имяПодразделенияОрганизации,
        владелец,
        смотриТакже,
        порядковыйНомер }
    ::= { классОбъектов 14 }
```

Примечание. — Должен присутствовать по крайней мере один из атрибутов: имяМестности, порядковыйНомер, владелец. Выбор зависит от типа устройства.

6.16 *Пользователь строгой аутентификации*

Класс объектов *Пользователь строгой аутентификации* используется для определения статей тех объектов, которые являются участниками строгой аутентификации, как это определено в Рекомендации X.509.

```
пользовательСтрогойАутентификации OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{ сертификатПользователя }
    ::= { классОбъектов 15 }
```

6.17 *Сертификатный орган*

Класс объектов *Сертификатный орган* используется для определения статей объектов, выступающих в роли сертификатного руководящего органа, как это определено в Рекомендации X.509.

```
сертификатныйОрган OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        сертификатCA,
        списокОтмененныхСертификатов,
        списокОтмененныхОрганов }
    MAY CONTAIN { перекрестнуюПаруСертификатов }
    ::= { классОбъектов 16}
```

ПРИЛОЖЕНИЕ А

(к Рекомендации X.521)

Избранные классы объектов на НАС.1

Данное Приложение содержит в форме НАС.1-модуль ИзбранныеКлассыОбъектов, в который включены все НАС 1-определения типов и значений, введенных в настоящей Рекомендации.

```
ИзбранныеКлассыОбъектов   { joint-ISO-CCITT ds(5) modules(1)
                            selectedObjectClasses(6)}

DEFINITIONS ::=

BEGIN

-- экспортирует все возможное

IMPORTS
    классОбъектов, наборАтрибутов, структураИнформации, избранныеТипыАтрибутов
        FROM полезныеОпределения { joint-iso-ccitt ds(5) mudules(1) usefulDefinitions(0)}
    OBJECT-CLASS, ATTRIBUTE-SET, Вершина, Псевдоним
        FROM СтруктураИнформации структураИнформации
    списокОтмененныхОрганов, категорияБизнеса, СертификатСА, списокОтмененныхСертификатов,
    обычноеИмя, имяСтраны, описание индикаторПунктаНазначения, номерФаксимильногоТелефона,
    номерВМеждународнойЦСИС, информационныеЗнания, имяМестности, член, имяОрганизации,
    имяПодразделенияОрганизации, владелец, имяОфисаФизическойДоставки, почтовыйЯщик, почтовыйАдрес,
    почтовыйКод, предпочитаемыйМетодДоставки, адресВУровнеПредставлений, зарегистрированныйАдрес,
    сотрудникШтатнойДолжности, указательПоиска, смотриТакже, порядковыйНомер, имяШтатаИлиОбласти,адресТипаУлица,
    обеспечиваемыйПрикладнойКонтекст, фамилия, телефонныйНомер, идентификаторТелетексногоТерминала,
    телексныйНомер, титул, сертификатПользователя, парольПользователя, x121Адрес
        FROM ИзбранныеТипыАтрибутов избранныеТипыАтрибутов;

телекоммуникационныйНаборАтрибутов ATTRIBUTE-SET
    CONTAINS {
        номерФаксимильногоТелефона,
        цСИСАдрес,
        телефонныйНомер,
        идентификаторТелетексногоТерминала,
        телексныйНомер,
        x121Адрес, предпочитаемыйМетодДоставки, индикаторПунктаНазначения,
        зарегистрированныйАдрес }
        ::= { наборАтрибутов 0 }

почтовыйНаборАтрибутов ATTRIBUTE-SET
    CONTAINS {
        имяОфисаФизическойДоставки,
        почтовыйАдрес,
        почтовыйКод,
        почтовыйЯщик,
        адресТипаУлица }
        ::= { наборАтрибутов 1 }

наборАтрибутовМестности ATTRIBUTE-SET
    CONTAINS {
        имяМестности,
        имяШтатаИлиОбласти,
        адресТипаУлица }
        ::= { наборАтрибутов 2 }

наборАтрибутовОрганизации ATTRIBUTE-SET
    CONTAINS {
        описание,
        наборАтрибутовМестности,
        почтовыйНаборАтрибутов,
        телекоммуникационныйНаборАтрибутов,
        категорияБизнеса,
```

```
        смотриТакже,
        указательПоиска,
        парольПользователя }
        ::= { наборАтрибутов 3 }

вершина     Вершина  ::=  { классОбъектов 0 }

псевдоним   Псевдоним  ::=  { классОбъектов 1 }

страна OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        имяСтраны }
    MAY CONTAIN {
        описание,
        указательПоиска }
    ::= { классОбъектов 2 }

местность OBJECT-CLASS
    SUBCLASS OF вершина
    MAY CONTAIN {
        описание,
        имяМестности,
        имяШтатаИлиОбласти,
        указательПоиска,
        смотриТакже,
        адресТипаУлица }
    ::= { классОбъектов 3 }

организация OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        имяОрганизации }
    MAY CONTAIN {
        наборАтрибутовОрганизации }
    ::= { классОбъектов 4 }

подразделениеОрганизации OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        имяПодразделенияОрганизации }
    MAY CONTAIN {
        наборАтрибутовОрганизации }
    ::= { классОбъектов 5 }

человек OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        обычноеИмя,
        фамилия }
    MAY CONTAIN {
        описание,
        смотриТакже,
        телефонныйНомер,
        парольПользователя }
    ::= { классОбъектов 6 }

сотрудникОрганизации OBJECT-CLASS
    SUBCLASS OF человек
    MAY CONTAIN {
        наборАтрибутовМестности,
        имяПодразделенияОрганизации,
        почтовыйНаборАтрибутов,
        телекоммуникационныйНаборАтрибутов,
        титул }
    ::= { классОбъектов 7 }
```

```
штатнаяДолжностьОрганизации OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя }
    MAY CONTAIN {
        описание,
        наборАтрибутовМестности,
        имяПодразделенияОрганизации,
        почтовыйНаборАтрибутов,
        предпочитаемыйМетодДоставки,
        сотрудникШтатнойДолжности,
        смотриТакже,
        телекоммуникационныйНаборАтрибутов }
    ::= { классОбъектов 8 }

группаИмен OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        обычноеИмя,
        член }
    MAY CONTAIN {
        описание,
        имяОрганизации,
        имяПодразделенияОрганизации,
        владелец,
        смотриТакже,
        категорияБизнеса }
    ::= { классОбъектов 9 }

житель OBJECT-CLASS
    SUBCLASS OF человек
    MUST CONTAIN {
        имяМестности }
    MAY CONTAIN {
        наборАтрибутовМестности,
        почтовыйНаборАтрибутов,
        предпочитаемыйМетодДоставки,
        телекоммуникационныйНаборАтрибутов,
        категорияБизнеса }
    ::= { классОбъектов 10 }

прикладнойПроцесс OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        обычноеИмя }
    MAY CONTAIN {
        описание,
        имяМестности,
        имяПодразделенияОрганизации,
        смотриТакже }
    ::= {классОбъектов 11}

прикладнойЭлемент OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN {
        обычноеИмя,
        адресВУровнеПредставлений }
    MAY CONTAIN {
        описание,
        имяМестности,
        имяОрганизации,
        имяПодразделенияОрганизации,
        смотриТакже,
        обеспечиваемыйПрикладнойКонтекст }
    ::= { классОбъектов 12 }
```

```
cAC OBJECT-CLASS
    SUBCLASS OF прикладнойЭлемент
    MAY CONTAIN{
        информационныеЗнания}
    ::= {классОбъектов 13}

устройство OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        обычноеИмя }
    MAY CONTAIN{
        описание,
        имяМестности,
        имяОрганизации,
        имяПодразделенияОрганизации,
        владелец,
        смотриТакже,
        порядковыйНомер }
    ::= {классОбъектов 14}

пользовательСтрогойАутентификации OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        сертификатПользователя}
    ::= {классОбъектов 15}

сертификатныйОрган OBJECT-CLASS
    SUBCLASS OF вершина
    MUST CONTAIN{
        сертификатCA,
        списокОтмененныхСертификатов,
        списокОтмененныхОрганов}
    MAY CONTAIN{
        перекрестнуюПаруСертификатов}
    ::= {классОбъектов 16}

END
```

ПРИЛОЖЕНИЕ B

(к Рекомендации X.521)

Рекомендуемые форматы имен и структуры ИДС

Данное приложение не является составной частью настоящей Рекомендации.

В настоящем Приложении даются предложения о некоторых общих методах именования объектов и о структуре ИДС. Административный орган может воспользоваться, а может и не воспользоваться этими предложениями. Методы именования и определения структуры дерева для некоторого класса объектов включают в себя спецификацию атрибутов, используемых для присвоения имен, и указания для каждого класса объектов тех классов объектов, которые могут быть непосредственно предшествующими и непосредственно последующими по отношению к данному классу. Все статьи некоторого класса объектов должны содержать по меньшей мере те атрибуты, которые используются для именования объектов. Пользователи Справочника должны быть поставлены в известность о рекомендуемом формате имен, чтобы они могли предугадывать имена тех объектов, с которыми собираются связываться. В нижеследующих пунктах приводятся рекомендуемые правила формирования имен и структурирования для некоторых классов объектов.

Правила структурирования изображены на рис. B-1/X.521.

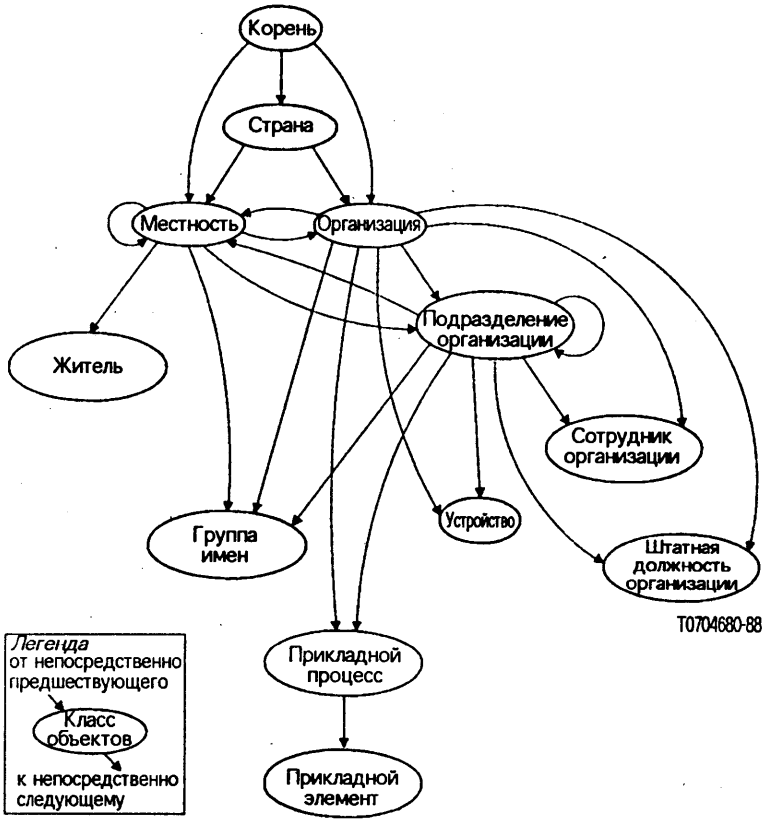


РИСУНОК B-1/X.521

Рекомендуемая структура ИДС

B.1 *Страна*

Для именования используется атрибут **имяСтраны**.

Непосредственно предшествующими статьями класса объектов **Страна** является **Корень**.

B.2 *Организация*

Для наименования используется атрибут **имяОрганизации**.

Непосредственно предшествующими статьями класса объектов **Организация** могут быть объекты **Корень**, **страна** или **местность**.

Примечание. — Если организация является непосредственно следующей за корнем, то это означает, что организация является международной. Для всех международных организаций значения атрибута **имяОрганизации** должны быть различными.

B.3 *Местность*

Для наименования используется один из атрибутов **имяМестности** или **имяШтатаИлиОбласти**.

Непосредственно предшествующими статьями класса объектов **местность** могут быть объекты **Корень**, **страна**, **местность**, **организация** или **подразделениеОрганизации**.

B.4 *Подразделение организации*

Для наименования используется атрибут **имяПодразделенияОрганизации**.

Непосредственно предшествующими статьями класса объектов **подразделениеОрганизации** могут быть объекты **организация**, **подразделениеОрганизации** или **местность**.

B.5 *Сотрудник организации*

Для наименования используется атрибут **обычноеИмя** и, возможно, атрибут **имяПодразделенияОрганизации**.

Непосредственно предшествующими статьям класса объектов **сотрудникОрганизации** могут быть объекты **организация** или **подразделениеОрганизации**.

Примечание. — Имеется два способа, с помощью которых атрибут **имяПодразделенияОрганизации** может быть включен в состав имени: либо если объект **подразделениеОрганизации** является непосредственно предшествующим, либо непосредственным включением этого атрибута.

B.6 *Штатная должность организации*

Для наименования используется атрибут **обычноеИмя**.

Непосредственно предшествующими статьям класса объектов **штатнаяДолжностьОрганизации** могут быть объекты **организация** или **подразделениеОрганизации**.

Примечание. — Имеется два способа, с помощью которых атрибут **имяПодразделенияОрганизации** может быть включен в состав имени: либо если объект **подразделениеОрганизации** является непосредственно предшествующим, либо непосредственным включением этого атрибута.

B.7 *Группа имен*

Для наименования используется атрибут **обычноеИмя**.

Непосредственно предшествующими статьям класса объектов **группаИмен** могут быть объекты **местность**, **организация** или **подразделениеОрганизации**.

Примечание. — Имеется два способа, с помощью которых атрибут **имяПодразделенияОрганизации** может быть включен в состав имени: либо если объект **подразделениеОрганизации** является непосредственно предшествующим, либо непосредственным включением этого атрибута.

B.8 *Житель*

Для наименования используется атрибут **обычноеИмя** и, возможно, атрибут **адресТипаУлицы**.

Непосредственно предшествующими статьям класса объектов **житель** могут быть объекты **местность**.

B.9 *Прикладной элемент*

Для наименования используется атрибут **обычноеИмя**. Однако **обычноеИмя** должно содержать квалификатор прикладного элемента (см. Рекомендацию X.200).

Непосредственно предшествующими статьям класса объектов **прикладнойЭлемент** могут быть объекты **прикладнойПроцесс**.

B.10 *Устройство*

Для наименования используется атрибут **обычноеИмя**.

Непосредственно предшествующими статьям класса объектов **устройство** могут быть объекты **организация** или **подразделениеОрганизации**.

Примечание. — Имеется два способа, с помощью которых атрибут **имяПодразделенияОрганизации** может быть включен в состав имени: либо если объект **подразделениеОрганизации** является непосредственно предшествующим, либо непосредственным включением этого атрибута.

B.11 *Прикладной процесс*

Для наименования используется атрибут **обычноеИмя**.

Непосредственно предшествующими статьям класса объектов **прикладнойПроцесс** могут быть объекты **организация** или **подразделениеОрганизации**.

Примечание 1. — Способ выборки **обычногоИмени** для наименования Прикладного элемента описан в Рекомендации X.200.

Примечание 2. — Имеется два способа, с помощью которых атрибут **имяПодразделенияОрганизации** может быть включен в состав имени: либо если объект **подразделениеОрганизации** является непосредственно предшествующим, либо непосредственным включением этого атрибута.

Printed in USSR • 1992 – ISBN 92-61-03734-8