ГОСТ Р ИСО/МЭК 10031-1—2000

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ТЕКСТОВЫЕ И УЧРЕЖДЕНЧЕСКИЕ СИСТЕМЫ. МОДЕЛЬ ПРИЛОЖЕНИЙ РАСПРЕДЕЛЕННОГО УЧРЕЖДЕНИЯ

Часть 1

Общая модель

Издание официальное

БЗ 8—2000/259

ГОССТАНДАРТ РОССИИ
Москва

Предисловие

1 РАЗРАБОТАН Государственным научно-исследовательским и конструкторско-технологическим институтом «ТЕСТ» Государственного комитета Российской Федерации по телекоммуникациям

ВНЕСЕН Государственным комитетом Российской Федерации по телекоммуникациям

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 28 ноября 2000 г. № 317-ст

Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 10031-1—91 «Информационная технология. Текстовые и учрежденческие системы. Модель приложений распределенного учреждения. Часть 1. Общая модель»

3 ВВЕДЕН ВПЕРВЫЕ

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ТЕКСТОВЫЕ И УЧРЕЖДЕНЧЕСКИЕ СИСТЕМЫ. МОДЕЛЬ ПРИЛОЖЕНИЙ РАСПРЕДЕЛЕННОГО УЧРЕЖДЕНИЯ

Часть 1

Общая модель

Information technology. Text and office systems. Distributed-office-applications model. Part 1. General model

Дата введения 2002—01—01

1 Область применения

Стандарты серии ГОСТ Р ИСО/МЭК 10031 предоставляют каркас для разработки стандартов протоколов распределенных учрежденческих приложений (РУП, distributed-office-application — DOA). Они применимы для приложений, распределенных на значительных физических расстояниях, а также для «тесно связанных» учрежденческих систем.

Стандарты серии ГОСТ Р ИСО/МЭК 10031 описывают модель. Для того чтобы распределенные учрежденческие приложения были стандартизованы, они должны использовать принципы, установленные в этих стандартах.

Стандарты серии ГОСТ Р ИСО/МЭК 10031 предоставляют руководства по проектированию протоколов, которые открывают доступ к различным приложениям и взаимодействию между приложениями. Протоколы для распределенных приложений находятся в пределах прикладного уровня ВОС и соответствуют прикладным операциям, определенным в стандартах серии ГОСТ Р ИСО/МЭК 9072.

В стандартах серии ГОСТ Р ИСО/МЭК 10031 подразумевается, что элементы системы, соответствующей какому-либо из них, могут быть реализованы на устройствах, поставляемых разными продавцами и разными поставщиками услуг.

В стандартах серии ГОСТ Р ИСО/МЭК 10031 не определяется какой-либо интерфейс человек—машина, используемый с распределенными приложениями. Также в них не определяется интерфейс между программным обеспечением, непосредственно взаимодействующим с пользователем, и программным обеспечением конкретных приложений.

Содержание стандартов серии ГОСТ Р ИСО/МЭК 10031 разбито на две части.

В настоящем стандарте описана общая модель распределенных учрежденческих приложений, и он подразделяется на две части:

а) модель;

б) руководства по проектированию протоколов.

В ИСО/МЭК 10031-2 [1] описана отличающая объект ссылка и соответствующие процедуры, которые могут быть использованы всеми РУП.

Требования соответствия настоящему стандарту не устанавливаются. В других стандартах серии ГОСТ Р ИСО/МЭК 10031 могут быть установлены требования соответствия для систем, реализующих процедуры этих стандартов.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ 6.20.1—90 Электронный обмен данными в управлении, торговле и на транспорте (ЭДИФАКТ). Синтаксические правила

ГОСТ 34.971—91 (ИСО 8822—88) Информационная технология. Взаимосвязь открытых систем. Определение услуг уровня представления с установлением соединения

Издание официальное

ГОСТ 34.981—91 (ИСО 8649—88) Информационная технология. Взаимосвязь открытых систем. Определение услуг сервисного элемента управления ассоциацией

ГОСТ 28906—91 (ИСО 7498—84, ИСО 7498—84, Доп. 1—84) Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель

ГОСТ Р ИСО/МЭК 7498-2—99 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура защиты информации

ГОСТ Р ИСО/МЭК 7498-3—97 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 3. Присвоение имен и адресация

ГОСТ Р ИСО/МЭК 8824—93 Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии один (АСН.1)

ГОСТ Р ИСО/МЭК 9066-1—93 Системы обработки информации. Передача текста. Надежная передача. Часть 1. Модель и определение услуг

ГОСТ Р ИСО/МЭК 9072-1—93 Системы обработки информации. Передача текста. Удаленные операции. Часть 1. Модель, нотация и определение услуг

ГОСТ Р ИСО/МЭК 9072-2—93 Системы обработки информации. Передача текста. Удаленные операции. Часть 2. Спецификация протокола

ГОСТ Р ИСО/МЭК 9594-1—98 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 1. Общее описание принципов, моделей и услуг

ГОСТ Р ИСО/МЭК 9594-3—98 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 3. Определение абстрактных услуг

ГОСТ Р ИСО/МЭК 10021-2—98 Информационная технология. Передача текста. Системы обмена текстами, ориентированные на сообщения (MOTIS). Часть 2. Общая архитектура

ГОСТ Р ИСО/МЭК 10021-3—98 Информационная технология. Передача текста. Системы обмена текстами, ориентированные на сообщения (MOTIS). Часть 3. Соглашения по определению абстрактных услуг

ГОСТ Р ИСО/МЭК 10021-5—96 Информационная технология. Передача текста. Системы обмена текстами, ориентированные на сообщения (MOTIS). Часть 5. Хранилище сообщений: определение абстрактных услуг

3 Определения

3.1 Определения базовой эталонной модели ВОС

В настоящем стандарте используют следующие термины, определенные в ГОСТ 28906:

а) **прикладной уровень;**

б) **прикладная категория;**

в) **элемент услуги прикладного уровня;**

г) **уровень представления;**

д) **соединение уровня представления;**

е) **протокол;**

ж) **определение услуги.**

3.2 Определения безопасности базовой эталонной модели ВОС

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 7498-2:

а) **аутентификация;**

б) **авторизация;**

г) **полномочия;**

д) **политика безопасности.**

3.3 Определения сервисного элемента управления ассоциацией (СЭУА, ACSE)

В настоящем стандарте используют следующие термины, определенные в ГОСТ 34.981:

а) **прикладной контекст;**

б) **сервисный элемент управления ассоциацией (СЭУА).**

3.4 Определение услуг уровня представления

В настоящем стандарте используют следующий термин, определенный в ГОСТ 34.971:

абстрактный синтаксис.

3.5 Определения абстрактной синтаксической нотации

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 8824:

а) **АСН.1**

б) **внешний тип;**
в) **обобщенное время;**
г) **макро;**
д) **идентификатор объекта;**
е) **время UTC.**

3.6 Определение сервисного элемента надежной передачи (СЭНП, RTSE)

В настоящем стандарте используют следующий термин, определенный в ГОСТ Р ИСО/МЭК 9066-1:

сервисный элемент надежной передачи (СЭНП).

3.7 Определения сервисного элемента удаленных операций (СЭУО, ROSE)

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 9072-1:

а) **аргумент;**
б) **операция связывания;**
в) **вызов;**
г) **операция;**
д) **выполнение;**
е) **удаленные операции;**
ж) **сервисный элемент удаленных операций (СЭУО);**
з) **результат;**
и) **операция развязывания.**

3.8 Определения справочника

В настоящем стандарте используют следующие термины, определенные в стандартах серии ГОСТ Р ИСО/МЭК 9594:

а) **атрибут;**
б) **макроатрибут;**
в) **тип атрибута;**
г) **значение атрибута;**
д) **фильтр.**

3.9 Определение EDIFACT

В настоящем стандарте используют следующий термин, определенный в ГОСТ 6.20.1:

EDIFACT.

3.10 Определения систем передачи текста, ориентированных на сообщения (MOTIS)

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 10021-2:

а) **часть тела (сообщения);**
б) **IP-сообщение;**
в) **сообщение.**

3.11 Определения соглашения по определению абстрактных услуг

В настоящем стандарте используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 10021-3:

а) **абстрактная модель;**
б) **абстрактные операции;**
в) **абстрактные услуги;**
г) **макро абстрактных услуг;**
д) **асимметричный;**
е) **порт;**
ж) **очистка;**
з) **симметричный.**

3.12 Определения модели распределенных учрежденческих приложений (МРУП)

В настоящем стандарте используют следующие определения:

3.12.1 **исполнитель:** *x*-сервер, который может присваивать отличающие объекты ссылки (ООС) объектам (которыми управляют по запросам от *x*-клиентов) и осуществлять операции, обращающиеся к объектам по присвоенным им ООС.

3.12.2 **соучастник:** *x*-сервер, который может осуществлять операции, объекты которых указываются ООС, обращаясь к исполнителю с этими ООС.

3.12.3 **атрибуты управления:** Атрибуты, связанные с объектом защиты, которые, при согласо-

вании с атрибутами привилегий субъекта защиты, используются для предоставления доступа или отказа в доступе к объекту защиты.

3.12.4 **пакет атрибутов управления:** Совокупность атрибутов управления.

3.12.5 **операция потребления:** Операция, вызванная *x*-клиентом у соучастника, объекты которой указаны ООС.

3.12.6 **объект данных:** Объект, который представляет данные.

3.12.7 **значение объекта данных:** Значение, полученное из объекта данных в соответствии с набором правил или, при отсутствии таких правил, значение всего объекта.

3.12.8 **прямой доступ к значению:** Доступ к объекту данных по значению, а не по ссылке.

3.12.9 **прямая передача значения:** Непосредственная передача значения объекта данных, а не передача ссылки.

3.12.10 **отличающая объект ссылка (ООС):** Однозначная ссылка на реальный объект в среде РУП.

3.12.11 **распределенное учрежденческое приложение:** Набор ресурсов обработки информации, распределенных в одной или нескольких открытых системах, который обеспечивает хорошо определенный набор функциональных возможностей для пользователя (человека) для помощи при решении данной учрежденческой задачи.

3.12.12 **документ:** Структурированная информация, прямо или косвенно предназначенная для восприятия человеком, которая может передаваться, храниться, разыскиваться и обрабатываться с помощью учрежденческих приложений.

3.12.13 **инициатор:** *x*-клиент, который вызывает операции, запрашивающие у исполнителя ООС, а не значение объекта данных, и который вызывает у соучастника операции, объекты которых указываются полученными ООС.

3.12.14 **объект учрежденческих данных:** Объект, который представляет учрежденческую информацию.

3.12.15 **учрежденческая информация:** Данные, используемые в учреждении.

3.12.16 **атрибуты привилегий:** Атрибуты, связанные с субъектом защиты, которые, при согласовании с атрибутами контроля объекта защиты, используются для предоставления доступа или отказа в доступе к этому объекту защиты.

3.12.17 **сертификат атрибутов привилегий:** Сертификат, использующий атрибуты привилегий.

3.12.18 **операция создания:** Операция, вызванная *x*-клиентом у исполнителя, в которой запрашивается ООС, а не значение объекта данных.

3.12.19 **квалифицированные атрибуты:** Атрибуты, которые имеют квалификацию использования.

3.12.20 **ссылочный доступ к объекту (СДО):** Доступ к объектам с помощью ссылок.

3.12.21 **операция СДО:** Операция, вызванная соучастником у исполнителя.

3.12.22 **атрибуты безопасности:** Общий термин, охватывающий как атрибуты привилегий, так и атрибуты контроля. Использование атрибутов безопасности определяется политикой безопасности.

3.12.23 **объект безопасности:** Играющая пассивную роль категория, доступ к которой предоставляется или в доступе отказывается в соответствии с политикой авторизации.

3.12.24 **субъект безопасности:** Играющая активную роль категория, которой предоставляет доступ или отказывает в доступе к объекту безопасности в соответствии с политикой авторизации.

3.12.25 **прикладной процесс пользователя:** Прикладной процесс, который содержит УП-пользователя и одного или нескольких клиентов распределенных (учрежденческих) приложений (например, *x*-клиент, *y*-клиент и т. п.).

3.12.26 ***x*-:** Родовое представление для конкретных имен приложений.

3.12.27 ***x*-доступ:** Определение функциональных возможностей *x*-приложения так, как они видны между *x*-клиентом и *x*-сервером.

3.12.28 **абстрактная услуга *x*-доступа:** Абстрактная услуга между *x*-клиентом и *x*-сервером.

3.12.29 **протокол *x*-доступа:** Протокол, используемый между *x*-клиентом и *x*-сервером.

3.12.30 ***x*-приложение:** Некоторого рода распределенное учрежденческое приложение, например приложение электронной почты или приложение хранения и поиска.

3.12.31 ***x*-прикладная система:** Совокупность *x*-клиентов и системы *x*-сервера, которые вместе обеспечивают *x*-пользователю функциональные возможности *x*-приложения.

3.12.32 ***x*-клиент:** Та часть *x*-приложения, которая является частью прикладного процесса, содержащей *x*-пользователь.

3.12.33 ***x*-сервер:** Та часть *x*-приложения, которая является частью прикладного процесса *x*-сервера и обеспечивает функциональные возможности, специфицированные в определении абстрактных услуг *x*-доступа.

3.12.34 **система *x*-серверов:** Совокупность одного или нескольких *x*-серверов.

3.12.35 **абстрактные услуги *x*-системы:** Абстрактные услуги между *x*-серверами.

3.12.36 **протокол *x*-системы:** Протокол, используемый между *x*-серверами.

3.12.37 ***x*-пользователь:** Часть прикладного процесса, играющая роль, принятую при использовании *x*-приложения.

4 Сокращения

В настоящем стандарте использованы следующие сокращения:

АСН.1 — абстрактная синтаксическая нотация версии 1

ВОС — взаимосвязь открытых систем

КУ — качество услуги

МРУП — модель распределенного учрежденческого приложения

ООС — отличающая объект ссылка

РУП — распределенное учрежденческое приложение

САП — сертификат атрибутов привилегий

СДО — ссылочный доступ к объекту

СЭНП — сервисный элемент надежной передачи

СЭУА — сервисный элемент управления ассоциацией

СЭУО — сервисный элемент удаленных операций

EDIFACT — электронный обмен данными в управлении, коммерции и на транспорте (Electronic Data Interchange For Administration, Commerce and Transport)

UTC — всемирное скоординированное время (Coordinated Universal Time)

5 Модель

Примечание — Информацию, служащую обоснованием концепции, использованной в данном разделе, см. в приложении D.

5.1 Абстрактная модель РУП

5.1.1 Абстрактная модель доступа

Распределенные учрежденческие приложения должны развиваться в соответствии с абстрактной моделью клиент—сервер, показанной на рисунке 1, используя соглашения по определению абстрактных услуг, приведенные в ГОСТ Р ИСО/МЭК 10021-3.

На рисунке 1 *x*-пользователь есть пользователь *x*-приложения, которое обеспечивается *x*-прикладной системой. *x*-Пользователь взаимодействует с *x*-клиентом для использования услуг, предоставляемых *x*-приложением. *x*-Клиент получает доступ к *x*-серверу через *x*-доступ. Система *x*-серверов может быть разделенной и распределенной по нескольким *x*-серверам. Детали внутренней структуры системы *x*-серверов определены в 5.1.2.

Между *x*-клиентом и *x*-сервером может быть определен один или несколько портов. Для каждого порта его тип должен быть асимметричным.

Примечание 1 — Услуги доступа, предоставляемые асимметричными портами, находятся вне области применения настоящего стандарта.

Информация, которой обмениваются *x*-клиент и система *x*-серверов, должна быть учрежденческой. Учрежденческая информация является данными, используемыми в учреждении, например:

а) документы,

б) сообщения,

в) данные EDIFACT,

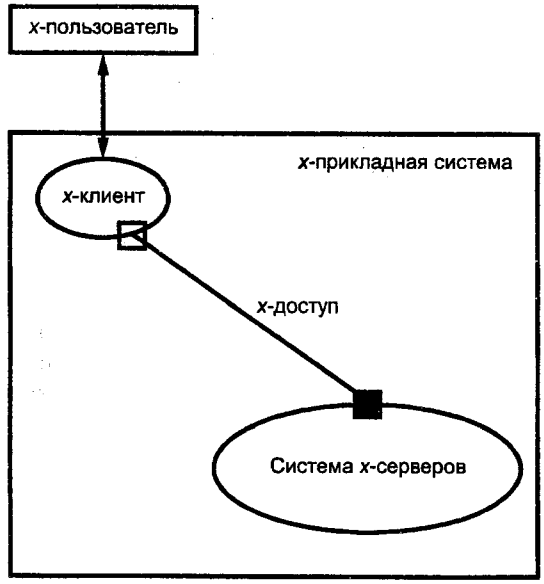


Рисунок 1 — Абстрактная модель доступа распределенного учрежденческого приложения

г) атрибуты документов,
д) время,
е) информация, относящаяся к сообщениям,
ж) информация о файлах документов,
з) информация для печати документов (включая шрифты),
и) управляющая информация для серверов.

Эта информация выглядит как совокупность учрежденческих объектов данных, к которым можно получить доступ и работать с ними по отдельности или в группе.

Примечание 2 — Обмен информацией, отличной от учрежденческой, определен или будет определен в других стандартах.

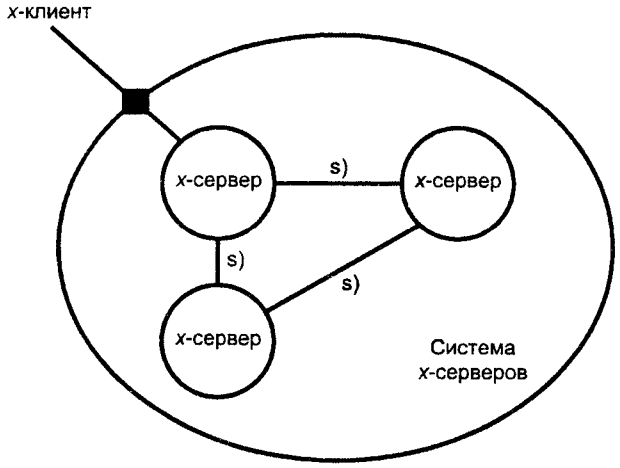


Рисунок 2 — Детализация системы *x*-серверов

5.1.2 Система *x*-серверов на рисунке 1 может быть уточнена для распределенной системы *x*-серверов путем определения абстрактных услуг между серверами, как предлагается в ГОСТ Р ИСО/МЭК 10021-3. На рисунке 2 показана детализация системы *x*-серверов.

На рисунке 2 *x*-клиент получает доступ к системе *x*-серверов через абстрактную услугу *x*-доступа (a). В системе *x*-серверов доступу соответствует *x*-сервер. Для предоставления запрошенных *x*-клиентом услуг *x*-сервер может взаимодействовать с другими *x*-серверами через абстрактные услуги *x*-системы (s).

Система *x*-серверов может охватывать различные типы *x*-серверов.

Между *x*-серверами может быть определен один или несколько портов. Может использоваться порт любого типа.

5.2 Реализация абстрактной модели РУП

5.2.1 Реализация абстрактной модели доступа

Для реализации абстрактной модели доступа должен использоваться СЭУО, определенный в стандартах серии ГОСТ Р ИСО/МЭК 9072, и его отображение в ВОС. Модель уровня приведена на рисунке 3. Дополнительную информацию об идентификации *x*-клиентов *x*-серверов см. в 6.4.4.

5.2.2 Реализация абстрактной модели для систем серверов

Нет ограничений на реализацию абстрактной модели для систем серверов. Примеры приведены в приложении D.

5.3 Ссылочный доступ к объекту

5.3.1 Классы доступа к данным

Доступ к значениям объектов данных концептуально вовлекает три стороны:

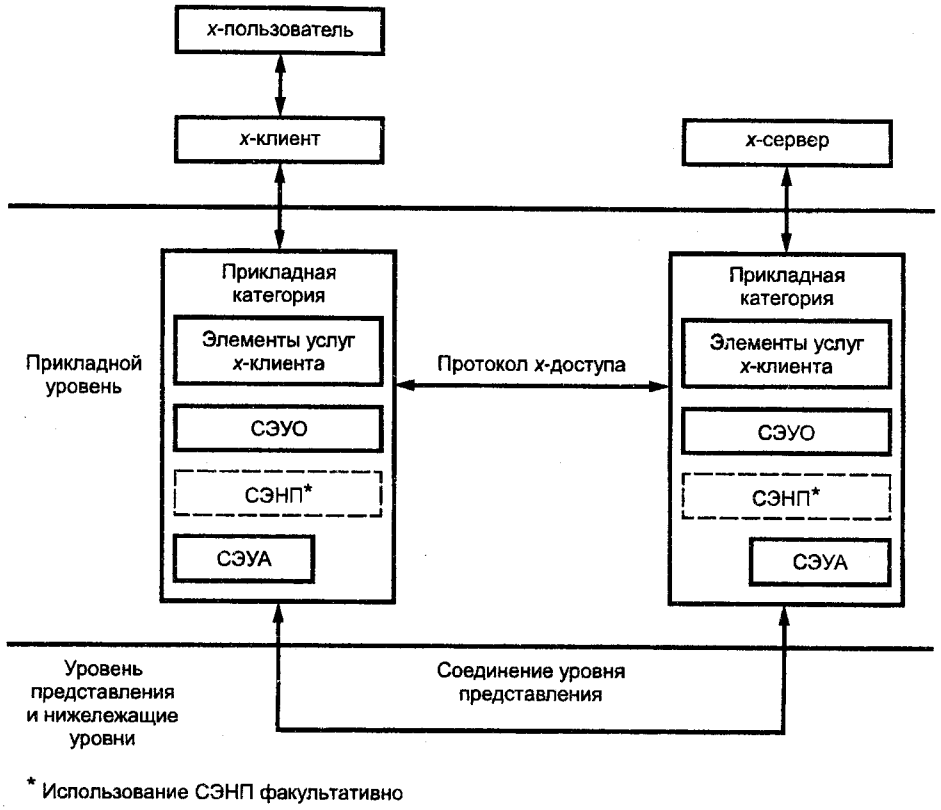
а) инициатора, который запрашивает доступ;
б) исполнителя, который хранит и создает значение объекта данных;
в) соучастника, который потребляет или модифицирует значение объекта данных.

В распределенных учрежденческих приложениях должны быть положения, действующие как исполнитель или соучастник для объектов данных, например файлов, документов или их частей.

Когда инициатор размещен вместе с исполнителем или соучастником, доступ к данным осуществляется как часть запроса доступа. Это называется прямым доступом к значению.

В случаях, когда инициатор отделен как от соучастника, так и исполнителя либо физически, либо во времени, использование прямого доступа к значению может потребовать две передачи данных (операции «чтения» и «записи»). Альтернативно, для более эффективного использования сетевых возможностей, инициатор может запросить accessee вернуть ссылку на объект данных, а не его фактическое значение. Затем эта ссылка может быть передана инициатором соучастнику, который может непосредственно взаимодействовать с исполнителем для доступа к значению объекта данных с единственной его передачей.

Аналогично, в языках программирования высокого уровня аргумент или результат передаются «по значению», когда используется прямой доступ, и «по имени», когда используется ссылочный доступ к данным.



* Использование СЭНП факультативно

Примечания

1 Данный рисунок является примером и не ограничивает реализации.

2 *x*-клиент и *x*-сервер могут иметь по несколько прикладных категорий и наоборот.

3 Прикладная категория может предоставлять услуги серверам разных типов.

Рисунок 3 — Модель уровней для реализации доступа в абстрактной модели РУП

5.3.2 Функциональная модель для ссылочного доступа к объекту

5.3.2.1 *Функциональная модель*

Функциональная модель ссылочного доступа к объекту (СДО) применяется, когда инициатор, исполнитель и соучастник разделены либо в пространстве, либо во времени. Эта модель показана на рисунке 4. Например, инициатор, исполнитель и соучастник могут выполняться в трех разных системах, или система инициатора может работать там же, где позже будет работать исполнитель или соучастник.

При СДО инициатору в результате операции создания возвращается ссылка на значение объекта данных (которая называется отличающей объект ссылкой или ООС) и передается соучастнику в аргументе операции потребления. Затем соучастник вызывает операцию СДО. В случае операции записи новое значение или инструкции по модификации значения данных могут быть переданы вместе с запросом доступа. В случае операции чтения фактическое значение указанного объекта данных возвращается в результате операции доступа.

Значение, доступное при выполнении операции СДО, относится к данным, известным исполнителю по специфическим для приложения правилам, которые связаны с ООС во время операции создания. Например, ООС может быть определена так, чтобы ссылаться на первую часть тела конкретного сообщения MOTIS.

Рисунок 4 — Модель ссылочного доступа к объектам

Операции СДО не обязательно ограничиваются доступом к фиксированным или постоянным объектам данных.

На рисунке 4 ООС создана исполнителем в ответ на операцию создания, вызванную инициатором. ООС предоставлена соучастнику инициатором в качестве параметра в операции потребления. Соучастник может использовать ООС для взаимодействия с исполнителем.

5.3.2.2 *Операции создания*

В некоторых операциях данных инициатор использует специфический для приложения протокол для выбора значения объекта данных (полного объекта данных, некоторого его подмножества или производного от объекта данных) у исполнителя. Операции этого класса называются «операциями создания».

При непосредственной передаче значения исполнитель возвращает значение объекта данных инициатору, а инициатор играет роль соучастника. С другой стороны, при непрямой передаче инициатор запрашивает ссылку на объект данных, а не его значение, и исполнитель возвращает инициатору ООС. ООС однозначно идентифицирует значение связанных с нею данных.

Когда ООС поддерживаются элементами протокола создания, в вызове требуется параметр для спецификации того, что должно быть возвращено: непосредственно значение данных или ООС. Соответственно, результат будет содержать значение данных или ООС.

5.3.2.3 *Операции потребления*

Инициатор может использовать специфический для приложения протокол также для того, чтобы значение объекта данных было доступно соучастнику. Операции этого класса называются «операциями потребления». При непосредственной передаче значения инициатор действует и как исполнитель и предоставляет значение объекта данных через протокол. При непрямой передаче инициатор предоставляет соучастнику ООС (ранее полученную от исполнителя). Соучастник использует эту ООС для осуществления операции доступа к исполнителю для чтения или записи указанного значения объекта данных.

Когда ООС поддерживаются элементами протокола потребления, предоставляемые данные могут быть либо значением объекта данных, либо ООС. В обоих случаях результат будет иметь идентичную семантику, но если используется ООС, то соучастнику может потребоваться подождать результат операции СДО до возвращения результата операции потребления.

5.3.2.4 *Операции СДО*

В модели распределенных учрежденческих приложений определен специальный класс протоколов, которые всегда используют ООС для взаимодействия с прикладными объектами исполнителя и которые обеспечивают обобщенный набор операций. Протоколы этого класса называются «протоколами ссылочного доступа к объекту (протоколы СДО)».

5.3.2.5 *Функции, подразумеваемые поддержкой СДО*

СДО требует, чтобы в исполнителя и соучастника были встроены дополнительные функциональные возможности:

а) исполнитель должен быть способен предоставлять в ответ на операцию создания ООС, а не значение объекта данных;

б) соучастник должен быть способен принимать в операции потребления ООС, а не значение объекта данных;

в) соучастник должен быть способен вызвать операцию доступа;

г) исполнитель должен быть способен осуществить операцию доступа.

В специфических для приложений протоколах стандарты могут:

а) либо допускать использование ООС в любых элементах протокола, когда предоставляется или возвращается ссылка на объект данных,

б) либо налагать дополнительные ограничения на допустимость использования ООС,

в) либо определять специальные элементы протокола для работы с операциями СДО-создания и СДО-потребления.

Первое из этих перечислений является наиболее предпочтительным и должно использоваться всюду, где это возможно.

Если исполнитель или соучастник в конкретном предлагаемом доступе не поддерживает ООС, то у инициатора нет выбора, как только выполнить две последовательных непосредственных передачи значения. В этом случае операция создания, вызванная инициатором, возвращает значение объекта данных от исполнителя к инициатору, а инициатор передает это значение объекта данных соучастнику в аргументе операции потребления. (Это описание применимо к операции чтения. В случае операции записи данные будут двигаться в противоположном направлении).

5.3.2.6 *Качество услуги*

Значения некоторых объектов данных будут изменяться со временем или объект может быть удален. Отдельные протоколы могут устанавливать, что ООС:

а) либо указывает на значение объекта данных, которое было в момент создания ООС,

б) либо указывает на текущее значение объекта данных,

в) либо становится неопределенным, если объект был обновлен.

Для обеспечения управления ссылками на динамически изменяющиеся объекты ООС может включать в себя индикацию качества услуги (КУ). КУ описывает ожидаемую или требуемую область для допустимости как ООС, так и значения соответствующих данных. Протоколы *x*-доступа могут нуждаться в поддержке элементов протокола для обновления КУ.

5.3.2.7 *Структура ООС*

Подробности структуры ООС и связанные с ней процедуры определены в ИСО/МЭК 10031-2 [1].

6 Руководство по проектированию протоколов

6.1 Введение

В данном разделе приведено руководство по проектированию протоколов, которого должны придерживаться все стандарты распределенных учрежденческих приложений.

6.2 Учрежденческая информация

Главной задачей распределенных учрежденческих приложений является обмен, хранение, поиск и обработка учрежденческой информации.

Для обеспечения многообразия существующих и будущих концепций и типов учрежденческой информации абстрактный синтаксис и семантика объектов учрежденческих данных обычно могут быть прозрачными для протоколов распределенных учрежденческих приложений. В этом случае объект учрежденческих данных должен появляться как внешний тип АСН.1 в варианте «непосредственного указания» (т. е. без согласования правил кодирования уровня представления) в абстрактном синтаксисе протоколов РУП. Значение «непосредственного указания идентификатора объекта» внешнего типа указывает как абстрактный синтаксис, так и кодирование объекта. Это значение должно использоваться в атрибутах, идентифицирующих тип объекта.

6.3 Модель объекта и удаленные операции

6.3.1 Использование удаленных операций

Удаленные операции, определенные в ГОСТ Р ИСО/МЭК 9072-1 и ГОСТ Р ИСО/МЭК 9072-2, обеспечивают нотацию и спецификацию протокола для операций связывания, развязывания и операций, вызванных в модели объекта операциями типа. В последующих подразделах приведено руководство по наименованию и стандартным установкам для операций.

Все протоколы доступа для распределенных учрежденческих приложений должны соответствовать удаленным операциям, установленным в ГОСТ Р ИСО/МЭК 9072-1 и ГОСТ Р ИСО/МЭК 9072-2. Протоколы доступа должны использовать нотацию и концепции настоящего стандарта и должны допускать любое отображение, определенное в ГОСТ Р ИСО/МЭК 9072-1, раздел 11. В приложении J дано краткое введение в эти концепции в контексте протокола доступа с учетом правил 6.4.

Для системных протоколов также рекомендуется использовать удаленные операции всюду, где это возможно.

6.3.2 Использование метода абстрактных услуг для определения *x*-услуги

Метод абстрактных услуг основан на ряде макросов АСН.1, которые используются для описания функций и параметров услуг. Этот метод описания услуг тесно связан со способом формального описания удаленных операций. Метод гарантирует полную согласованность между определениями услуг и спецификациями протоколов. Он позволяет избежать дублирования работы и документации при импорте определений из услуг в формальные протоколы. При этом столь же легко можно импортировать определения из одного РУП в другое без их дублирования. Все последующие РУП должны использовать этот метод для документирования услуг.

Макросы абстрактных услуг определены в ГОСТ Р ИСО/МЭК 10021-3.

6.4 Прикладные правила

Следующие правила установлены для того, чтобы упростить управление совместно используемыми рядом приложений ресурсами.

6.4.1 Конкуренция и разделение ресурсов

6.4.1.1 *Конкуренция*

В централизованных системах установлены методы для управления конкурирующим доступом и сохранения целостности данных. Для распределенных систем нет общих экономичных решений для общего случая распределенных данных.

Приложения должны избегать общего случая. Пока не выработаны строгие требования и решение принимается для конкретного приложения, следует руководствоваться более мягкими требованиями согласованности, а именно:

а) допускать несогласованные данные;

б) иметь одну основную копию для каждого элемента данных и один конкретный сервер, ответственный за ее обновление;

в) иметь одну последовательность распространения изменений на копии этого элемента данных и изменений связанных с ним элементов данных;

г) минимизировать взаимосвязи между элементами данных на разных серверах;

д) обеспечивать административный контроль для того, чтобы регулировать продолжительность распространения изменения;

е) проектируемые приложения должны быть терпимыми или гибкими относительно устаревших данных.

Если руководствоваться этими положениями, управление конкуренцией может быть ограничено одним *x*-сервером или, самое большее, *x*-системой. При этом столкновение протоколов ограничивается столкновением совместного использования ресурсов.

6.4.1.2 *Разделение ресурсов*

Разделением ресурсов в пределах сервера управляет сервер (который, в свою очередь, полагается на нижележащую операционную систему узла).

Столкновение протоколов ограничивается тем, что в результате при взаимодействии сервер не будет способен какое-то время отвечать. Это может быть объявлено отказом в принятии взаимодействия, ответом, указывающим задержку, или отсроченным ответом. Категория, действующая как *x*-пользователь, может быть обязана использовать перерыв (тайм-аут).

Если потребуется, то *x*-система может использоваться для управления ресурсами, совместно используемыми ее *x*-серверами. Это должно быть отражено в протоколе *x*-системы, но не должно влиять на протокол *x*-доступа.

6.4.2 Прозрачность сети

Для того чтобы изолировать пользователей от деталей конфигурации сети, серверы и клиенты должны указываться по именам, а не по адресам уровня представления. Для обеспечения перевода между именами и адресами может использоваться справочник.

6.4.3 Общее определение времени

Все протоколы в среде распределенных учрежденческих приложений должны выражать время, используя тип данных «GeneralizedTime», определенный в ГОСТ Р ИСО/МЭК 8824.

Time:: = GeneralizedTime

Примечание — В стандартах серий ГОСТ Р ИСО/МЭК 9594 и ГОСТ Р ИСО/МЭК 10021 в настоящее время используется тип «UTCTime» вместо « GeneralizedTime». В этих стандартах планируется использование типа «GeneralizedTime», сохраняющего обратную совместимость.

6.4.4 Общее определение идентификаторов

Все объекты, определенные в стандартах РУП, должны иметь по крайней мере одно глобально однозначное имя. Общее понимание имен и общее определение идентификаторов определены в ГОСТ Р ИСО/МЭК 7498-3, ГОСТ Р ИСО/МЭК 8824 и ГОСТ Р ИСО/МЭК 9594-1. Подробнее см. приложение Е.

Данная модель придерживается кодирования АСН.1 имен, используемого приложениями, определенного в ГОСТ Р ИСО/МЭК 9594-1.

6.4.5 Использование атрибутов и фильтров

Многие объекты в контексте распределенных учрежденческих приложений (т. е. представленные в информационной базе) характеризуются атрибутами. Атрибут состоит из типа атрибута, который идентифицирует класс информации, даваемой этим атрибутом, и соответствующего(их) значения(й) атрибута.

Понятие атрибута, нотация, обеспечивающая определения атрибутов, и абстрактный синтаксис атрибутов определены в ИСО/МЭК 9594-2 [2]. Подмножество определено в ГОСТ Р ИСО/МЭК

10021-5. Стандарты распределенных учрежденческих приложений должны использовать атрибуты, когда это возможно, и должны ссылаться на ИСО/МЭК 9594-2 [2].

Если объекты, представленные категориями в информационной базе, характеризуются атрибутами, то поиск информации (т. е. выбор записи) может потребовать использования фильтров. Фильтр использует проверку, которая либо удовлетворяется, либо нет (для конкретной записи). Фильтр выражается в терминах утверждений о наличии или значениях некоторых атрибутов (в записи).

Семантика и абстрактный синтаксис фильтров определены в ГОСТ Р ИСО/МЭК 9594-3, а подмножество — в ГОСТ Р ИСО/МЭК 10021-5. Стандарты для распределенных учрежденческих приложений должны использовать фильтры, когда это возможно, и должны ссылаться на ГОСТ Р ИСО/МЭК 9594-3.

Типы атрибутов, определенные для одного приложения, могут использоваться другим приложением, если определения и семантика одинаковы. В качестве инструмента для достижения этого используется идентификатор объекта, определенный в ГОСТ Р ИСО/МЭК 8824.

Макрос для атрибутов определен в ИСО/МЭК 9594-2 [2].

Можно рассматривать применение метода «квалификации атрибутов» в распределенных учрежденческих приложениях. Он, например, позволяет клиенту сигнализировать, является ли данный атрибут существенным в том смысле, что отвечающий сервер должен понимать семантику атрибута и знать, как отвечать, или сервер может игнорировать атрибут или подставить значение по умолчанию.

6.4.6 Ссылочный доступ к объекту

Значения объектов данных и ООС в протоколах доступа обычно могут появляться как внешние типы АСН.1.

6.4.7 Элементы прикладных услуг и прикладной контекст

Элементы прикладных услуг, выполняющие функции, требуемые для нескольких протоколов доступа, могут быть объединены в одном прикладном контексте, что потребует различающиеся абстрактные синтаксисы для каждого из этих элементов прикладных услуг (см. рисунок 3).

6.5 Правила безопасности

6.5.1 Введение

Следующие правила были установлены для того, чтобы упростить выполнение и управление вопросами безопасности управления доступом и аутентификации.

Правила допускают использование без изменений широкого диапазона политики безопасности для протоколов распределенных учрежденческих приложений, включая индивидуальную ответственность лиц, использующих распределенные учрежденческие приложения.

Примечание — Подробнее о понятии безопасности см. приложение F.

6.5.2 Субъект безопасности

Субъект безопасности часто несет индивидуальную ответственность за выполняемые операции. При некоторой политике безопасности ответственность может быть возложена на группу лиц или на пользователя. Распределенные учрежденческие приложения могут управлять доступом в терминах идентичности субъекта безопасности или в терминах возможностей, требуемых субъектом безопасности (имеются методы проверки таких требований). Сертификат атрибута привилегий (САП) является структурой данных, которая гарантирует передачу атрибутов субъекта безопасности; значения, присутствующие в этой структуре, зависят от субъекта и политики безопасности.

Распределенные учрежденческие приложения должны использовать САП для выражения аутентичности и привилегий субъекта безопасности. Когда используемая политика безопасности требует индивидуальной ответственности, САП должен содержать необходимую идентификацию.

САП должен передаваться в примитиве BIND и применяться ко всем последующим операциям в ассоциации до тех пор, пока примитив UNBIND или ABORT ее не завершат. Каждая отдельная операция должна позволять клиенту передать другой САП; он будет дополнением для данной операции к САП из примитива BIND.

Когда операция вызывает последующую обработку или следует за примитивами UNBIND или ABORT, САП, использованный в операции, должен применяться и для последующей обработки.

6.5.3 Объекты безопасности

Объект безопасности является объектом, который должен быть защищен в том смысле, что доступ субъектов безопасности регулируется используемой политикой безопасности. Политика безопасности может требовать, чтобы проверки осуществлялись на основе привилегий, потребованных идентификацией субъекта безопасности, или на основе привилегий, потребованных субъ-

ектом безопасности, или на основе их комбинации. Пакет атрибутов управления (ПАУ) является структурой данных, которая гарантирует передачу информации управления доступом; присутствующие в нем значения зависят от используемой политики безопасности и от отдельного объекта.

Распределенные учрежденческие приложения должны использовать ПАУ для передачи информации управления доступом объекта учрежденческих данных. Он может встречаться при создании, модификации или передаче объекта с его информацией управления доступом.

6.5.4 Управление доступом

Предыдущие правила безопасности отделяют проектирование протоколов распределенных учрежденческих приложений от проектирования элементов безопасности, передаваемых в этих протоколах. Это позволяет использовать одни и те же протоколы в различных средах обеспечения безопасности.

Определения операций и модели структур данных распределенных учрежденческих приложений должны допускать использование с этими протоколами широкого диапазона политики безопасности.

Любые положения политики безопасности, такие как неявное копирование атрибутов управления доступом из одного объекта безопасности в другой, должны оговариваться тем, что «субъект ограничивается используемой политикой безопасности».

6.5.5 Ошибки безопасности

Почти любая операция может быть полностью отвергнута на основании безопасности.

Об отвержениях на основании безопасности следует сообщать как об ошибках, имеющих предпочтение перед всеми другими ошибками (за исключением ошибок протокола, которые предотвращают определение субъекта безопасности, объекта безопасности и операции). Ответ ошибки безопасности не должен быть сформулирован таким образом, не должен сопровождаться какой-либо информацией и за ним не должно следовать сообщения о другой ошибке, которые могли бы явно или неявно передать информацию о том, какой субъект безопасности не должен быть задан.

Операции, которые вызвали доступ к одному объекту безопасности, не должны оказывать влияния на объект, если этот доступ был отвергнут.

Операции, которые вызывают доступ к нескольким объектам безопасности, могут столкнуться с отвержениями на основании безопасности для некоторых из этих объектов. Результирующие эффекты и ответы таких операций должны быть определены.

6.6 Стандартный набор операций

В данном подразделе приведен стандартный набор абстрактных операций.

Главной целью этого подраздела является гармонизация наборов абстрактных операций и их имен для различных распределенных учрежденческих приложений. Такая гармонизация уменьшает время и усилия, необходимые для стандартизации конкретных распределенных учрежденческих приложений, и облегчает работу реализаторам стандартов РУП, использующим операции конкретного приложения.

Разработчики конкретного распределенного учрежденческого приложения должны использовать операции этого стандартного набора для определения набора операций в их конкретном распределенном учрежденческом приложении, когда это возможно.

Стандартный набор абстрактных операций РУП включает в себя следующие операции:

а) перечислить (List);
б) читать (Read);
в) изменить (Modify);
г) копировать (Copy);
д) переместить (Move);
е) искать (Search);
ж) создать (Create);
и) удалить (Delete);
к) зарезервировать (Reserve);
л) отметить (Notify);
м) отказать (Abandon).

Пример этих операций приведен в приложении К.

ПРИЛОЖЕНИЕ А
(справочное)

Ссылки, определения и сокращения для последующих справочных приложений

А.1 Введение

В данном приложении приведены нормативные ссылки, определения и сокращения, использованные в последующих приложениях к настоящему стандарту. Нормативные ссылки, определения и сокращения, указанные в основном тексте настоящего стандарта, не включены в данное приложение.

А.2 Нормативные ссылки

В последующих приложениях к настоящему стандарту использованы ссылки на следующие стандарты:

ГОСТ Р 34.980.1—92 (ИСО 8571-1—88) Информационная технология. Взаимосвязь открытых систем. Передача, доступ и управление файлом. Часть 1. Общее описание

ГОСТ Р ИСО/МЭК 7498-4—99 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 4. Основы административного управления

ГОСТ Р ИСО 8613-1—99 Информационная технология. Текстовые и учрежденческие системы. Архитектура учрежденческих документов (ODA) и формат обмена. Часть 1. Введение и общие принципы

ГОСТ Р ИСО/МЭК 8879—99 Обработка информации. Текстовые и учрежденческие системы. Стандартный обобщенный язык разметки (SGML)

ГОСТ Р ИСО/МЭК 9545—98 Информационная технология. Взаимосвязь открытых систем. Структура прикладного уровня

А.3 Определения

А.3.1 Определения базовой эталонной модели ВОС

В последующих приложениях к настоящему стандарту используют следующий термин, определенный в ГОСТ 28906: **прикладной процесс.**

А.3.2 Определения безопасности базовой эталонной модели ВОС

В последующих приложениях к настоящему стандарту используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 7498-2:

а) **управление доступом;**
б) **список управления доступом;**
в) **проверка;**
г) **прослеживание проверки;**
д) **информация аутентификации;**
е) **средство;**
ж) **конфиденциальность;**
и) **целостность данных;**
к) **аутентификация источника данных;**
л) **цифровая подпись;**
м) **шифрование;**
н) **ключ;**
о) **управление ключами;**
п) **отвержение.**

А.3.3 Определения систем передачи текста, ориентированных на сообщения (MOTIS)

В последующих приложениях к настоящему стандарту используют следующие термины, определенные в ГОСТ Р ИСО/МЭК 10021-2:

а) **передача сообщений;**
б) **система передачи сообщений;**
в) **хранилище сообщений;**
г) **P2;**
д) **агент пользователя.**

А.3.4 Определения модели распределенных учрежденческих приложений (МРУП)

В последующих приложениях к настоящему стандарту используют следующие определения:

А.3.4.1 **политика управления доступом:** Набор правил, часть политики безопасности, по которым проводится аутентификация людей-пользователей или их представителей и по которым доступ этих пользователей к услугам и объектам безопасности предоставляется или закрывается.

А.3.4.2 **контекст доступа:** Контекст в терминах таких переменных, как положение, время дня, уровень безопасности установленных ассоциаций и т. п., в котором осуществляется доступ к объектам безопасности.

А.3.4.3 **криптографический ключ:** См. ключ.

А.3.4.4 **спецификация формата объекта данных:** Тип данных в смысле АСН.1, который определен независимо от протоколов *x*-доступа.

А.3.4.5 **узел:** Средство обработки данных, которое обеспечивает, как часть сети, ресурсы обработки информации. Узел может поддерживать прикладные процессы пользователей, прикладные процессы сервера или комбинацию обоих видов процессов.

А.3.4.6 **пользователь УП:** Часть прикладного процесса, которая непосредственно взаимодействует с пользователем-человеком и использует одно или несколько учрежденческих приложений в интересах этого пользователя-человека.

А.3.4.7 **администратор безопасности:** Уполномоченный (лицо или группа лиц), ответственный за реализацию политики безопасности для области безопасности.

А.3.4.8 **область безопасности:** Ограниченная группа объектов и субъектов безопасности, к которым применяется единая политика безопасности, осуществляемая единственным администратором безопасности.

А.3.4.9 **средства безопасности:** Процедуры, процессы, методы или их наборы, которые моделируют функции, относящиеся к безопасности.

А.3.4.10 **прикладной процесс сервера:** Прикладной процесс, который выполняет все или часть функциональных возможностей, установленных определением *x*-сервера.

А.3.4.11 **пользователь:** Пользователь-человек или *x*-пользователь.

А.3.4.12 **прикладной процесс пользователя:** Прикладной процесс, который содержит пользователя УП и одного или нескольких клиентов распределенного (учрежденческого) приложения (например, *x*-клиент, *y*-клиент и т. п.).

А.3.4.13 ***x*-, *y*-, *z*-, . . . :** Родовые представления для конкретных имен приложений.

А.3.4.14 **интерфейс *x*-приложения:** Интерфейс к *x*-приложению так, как он виден между *x*-пользователем и *x*-клиентом.

А.3.4.15 **определение *x*-услуги:** Определение функциональных возможностей *x*-приложения так, как они видны между *x*-клиентами и *x*-системой.

А.4 Сокращения

В последующих приложениях к настоящему стандарту использовано следующее сокращение:

СЭП — сервисный элемент приложения.

ПРИЛОЖЕНИЕ В
(справочное)

Взаимосвязь с другими стандартами

В.1 Содержание

В настоящем стандарте рассмотрены вопросы, относящиеся к интегрированным, хотя и распределенным, учрежденческим системам, обеспечивающим профессиональных, технических и административных пользователей.

В стандарте не рассматриваются ни вопросы реального времени, ни обработка транзакций, которые, в общем случае, обеспечивают операционный персонал в таких местах, как торговые точки, системы резервирования, места наличных расчетов и т. п.

Настоящий стандарт ориентирован на использование оборудования различных поставщиков, придерживающегося концепции ВОС, и учитывает взаимодействие между организациями. В стандарте учитывается обеспечение безопасности; более строгие руководства будут даны после того, как стандарты безопасности ВОС станут более четкими и стабильными.

В.2 Использование других стандартов

Протоколы, разработанные по данной модели, передаются через СЭУО в нормальном режиме протоколов верхнего уровня ВОС. Принято, что во всей распределенной учрежденческой системе должен быть доступен справочник.

В.3 Согласованность с другими стандартами

Система обмена текстами, ориентированными на сообщения (MOTIS), и справочник ранее определили ряд вопросов, общих для распределенных учрежденческих приложений. По историческим причинам в MOTIS имеются вопросы, не полностью встраивающиеся в настоящий стандарт.

Документы как обрабатываемая, печатаемая и пересылаемая информация являются основным видом учрежденческой информации. Хотя протоколы, которые будут разработаны по настоящей модели, не зависят от использования конкретного кодирования документов, они должны быть согласованы со стандартами по архитектуре учрежденческих (открытых) документов (ODA), стандартному обобщенному языку разметки (SGML), стандартному языку описания страниц (SPDL), языку семантики и спецификации стиля документа (DSSSL).

Удаленный доступ к базе данных (RDA) может использоваться как учрежденческая функция, и может быть полезной какая-либо гармонизация с МРУП.

В.4 Сосуществование с другими стандартами

В.4.1 Протокол виртуального терминала (VTP)

Так как VTP расположен между приложениями и пользователем-человеком, то он находится вне сферы действия настоящего стандарта.

В.4.2 Протоколы передачи файлов

Передача, доступ и управление файлами (FTAM) непосредственно не касаются большинства пользователей учрежденческих систем, которые работают в терминах документов и реляционно-подобного доступа к базе данных. Настоящий стандарт устанавливает руководство, которое не выполняется в рамках FTAM. Тем не менее приложения пользователей, в которых желательно использовать FTAM-подобные модели файлохранилища и соответствующие функциональные возможности, должны быть способны использовать FTAM наряду с учрежденческими приложениями.

В.4.3 Обработка транзакций (TP), завершение, конкуренция и откатка (CCR)

В настоящее время нет требований по использованию учрежденческих приложений в среде распределенной TP и к дисциплинам CCR. Следовательно, настоящий стандарт в настоящее время не требует, чтобы протоколы учрежденческих приложений проектировались в стиле, позволяющем удовлетворять этим стандартам.

В.4.4 Открытая распределенная обработка (ODP)

В настоящем стандарте не принимался во внимание рабочий элемент ODP. Однако в последующем ожидается гармонизация этих двух стандартов.

ПРИЛОЖЕНИЕ С
(справочное)

Требования

С.1 Введение

Распределенные учрежденческие приложения используются интегрированными распределенными учрежденческими системами. Распределенная учрежденческая система состоит из узлов пользователей и узлов серверов, связанных сетью. Узлы пользователей получают доступ к узлам серверов через сеть, используя протоколы доступа. Интегрированная учрежденческая система управляет согласованностью взаимодействия различных учрежденческих приложений.

В такой среде приложения обработки данных, которые в пределах одного хоста действуют как единое целое, расщепляются среди различных интеллектуальных компонентов системы. Такое расщепление приводит к необходимости стандартизации взаимодействия между различными частями приложения.

В распределенных учрежденческих системах не может быть гарантирована одновременная доступность всех ресурсов и ни поддерживающие, ни обрабатывающие приложения не должны предполагать, что все части конкретного распределенного процесса (например, передачи сообщения) одновременно находятся в состоянии взаимодействия, отличного от требований семантики данного процесса. Это приводит к понятию запоминания и последующего взаимодействия, например для передачи сообщения, когда агенты создателя и потенциального получателя сообщения не находятся одновременно в режиме взаимодействия.

С.2 Функциональные требования

Среди услуг, которые могут потребоваться пользователю от распределенной учрежденческой системы и которые охватываются настоящей моделью, имеются следующие:

а) межперсональные сообщения — для взаимодействия с другими пользователями;

б) групповое взаимодействие — для групп пользователей;

в) преобразование — для обмена документами с разными синтаксисами или кодированиями;

г) запись и поиск документов — для упорядоченной записи и многоключевого поиска документов;

д) ввод и вывод документов — для распределенных учрежденческих систем с различными физическими устройствами, такими как сканеры, принтеры и пр.;

е) справочник — для того, чтобы знать, где и как получить доступ к удаленным элементам взаимодействия, приложениям или пользователям;

ж) аутентификация — для предотвращения несанкционированного доступа к различным приложениям;

и) локально доступное базовое время — для таких целей, как отметка времени файлов и сообщений в сети;

к) прямой доступ — для удаленных серверов (например, видеотексту) и пользователей (например, через видеотекст);

л) непрямое взаимодействие (запоминание и последующее взаимодействие) с удаленными системами — для передачи информации вне реального времени или для доступа к другим сетям, например к системе передачи сообщений;

м) передача данных между различными приложениями — для удаленных серверов.

Приведенный список приложений в будущем расширится (например, за счет доступа к базам данных). Большинство из этих приложений, вероятно, будут обрабатывающими приложениями, так как их главная цель — обеспечить специфические возможности учрежденческих служащих.

Типичное использование этих приложений требует высокой степени интеграции. Например, документ может быть вызван с сервера хранения сообщений, запомнен на сервере записи и поиска документов и распечатан на сервере печати.

Операции некоторых из перечисленных выше распределенных учрежденческих приложений (обычно, групповые взаимодействия, запись и поиск документа, печать) могут потребовать использования других приложений, которые играют роль поддерживающих приложений (например, справочник, аутентификация).

Однако приложения и их пользователи не должны быть ограничены тем, как работают поддерживающие приложения. На практике фактическое распределение должно быть как можно более прозрачным для пользователей.

Поддерживающие приложения не обязательно видны для человека-пользователя, но они обеспечивают операции безопасности, надежности и непрерывности всей системы.

С.3 Требования проектирования

Проектирование протоколов должно обеспечивать:

а) устойчивость —
рекомендуемые принципы проектирования должны быть такими, чтобы проектировщик протокола распределенного учрежденческого приложения мог специфицировать высокостабильные взаимодействия распределенных приложений при полном использовании общих поддерживающих приложений;

б) модульное проектирование:
1) минимизирующее взаимозависимости между различными учрежденческими приложениями из пока еще неполного списка обрабатывающих приложений,
2) допускающее требование высокой степени интеграции;

в) общий стиль проекта протокола, включающий единообразное использование:
1) поддерживающих приложений и возможностей,
2) элементов услуг удаленных операций;

г) безопасность — различные уровни, задаваемые пользователем;

д) простоту — это ключевой момент с человеческой точки зрения на распределенные учрежденческие приложения. Это означает, что пользователь не должен быть вовлечен в то, как приложения управляют его запросом.

В настоящем стандарте установлены принципы взаимодействия между различными приложениями.

ПРИЛОЖЕНИЕ D
(справочное)

Основные понятия

D.1 Введение

Распределенные учрежденческие приложения подразделяются на категории как ряд приложений, которые, с точки зрения человека, образуют интегрированную учрежденческую систему.

Распределенная учрежденческая система состоит из узлов, связанных сетью. Хотя эта система существует для удобства пользователей-людей, любая часть прикладного процесса может использовать учрежденческие приложения.

Узел пользователя является устройством, с которым непосредственно взаимодействует пользователь-человек. Он обеспечивает непосредственные функции взаимодействия.

Узел сервера является устройством, которое управляет ресурсами, совместно используемыми несколькими пользователями.

Взаимодействия пользователей-людей с распределенной учрежденческой системой могут вызвать ряд активностей, которые должны осуществляться в ряде узлов. Как эти отдельные активности в узле представлены процессами, задачами и пр. — находится вне рассмотрения настоящего стандарта. Взаимодействия между одной активностью в одном узле и одной активностью в другом представлены в модели ВОС взаимодействиями между парами вызовов прикладных процессов, по одному в каждом узле. Вызов прикладного процесса выполняет функции прикладного процесса ВОС.

В данном подразделе взаимодействия описаны в терминах прикладных процессов. Каждая функциональная категория, описанная здесь, относится к прикладному процессу; каждая отдельная активность, выполняющая функции, относится к отдельному вызову прикладного процесса.

D.2 Модель клиент-сервер

В данном подразделе описывается подход к объяснению распределения единственного *x*-приложения, приложения определенного вида.

D.2.1 Единственное нераспределенное приложение

В единственном нераспределенном *x*-приложении *x*-пользователь и *x*-приложение находятся в одном месте. В специальном случае, когда *x*-пользователь находится во взаимодействии с пользователем-человеком и с *x*-приложением в интересах этого пользователя-человека, *x*-пользователь называется УП-пользователь, а прикладной процесс — прикладным процессом пользователя (см. D.3.1). *X*-пользователь взаимодействует с *x*-приложением через интерфейс *x*-приложения, который, обычно, является чей-либо собственностью и не является целью стандартизации (см. рисунок D.1).

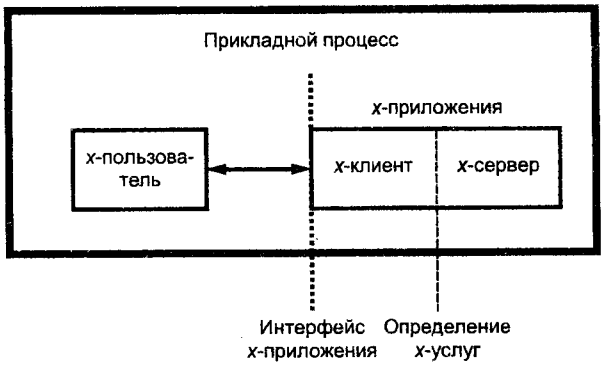


Рисунок D.1 — Нераспределенное учрежденческое приложение

Если *x*-приложение является кандидатом для распределения, то требуется определение *x*-услуг как разграничительной линии для последующего потенциального распределения.

D.2.2 Единственное распределенное приложение

Распределение *x*-приложения должно быть прозрачно для *x*-пользователя, так что интерфейс *x*-приложения меняться не будет. *X*-клиент расположен в том же самом месте, что и *x*-пользователь. *X*-пользователь и *x*-клиент вместе находятся в одном и том же прикладном процессе.

X-сервер, вообще говоря, удален от пользователя. *X*-сервер является частью прикладного процесса, который называется прикладным процессом сервера.

X-клиент и *x*-сервер взаимодействуют через сеть с помощью протокола *x*-доступа. Между *x*-клиентом и *x*-сервером может быть несколько независимых взаимодействий.

Новая ситуация отображена на рисунке D.2, где показано определение *x*-услуг рисунка D.1, расширенное до квадрата, содержащего протокол *x*-доступа.

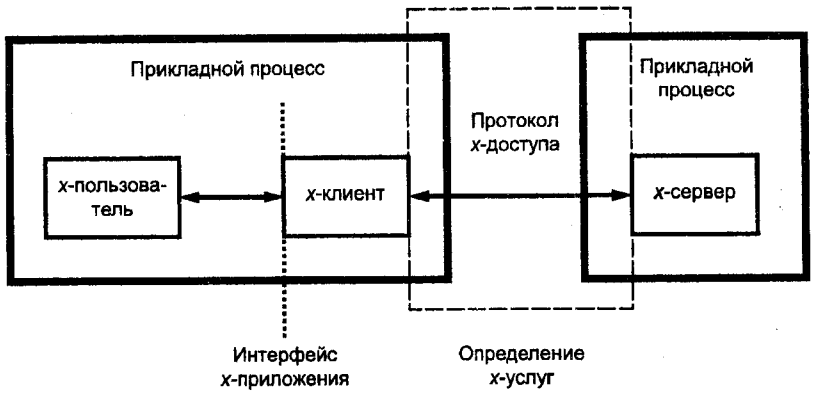


Рисунок D.2 — Распределенное учрежденческое приложение

В случае распределения должны быть стандартизованы определение *x*-услуг и протокол *x*-доступа.

D.2.3 Взаимодействие ВОС клиент — сервер

Протокол *x*-доступа является стандартным для *x*-клиента способом получить доступ к удаленному *x*-серверу. Следующая модель показывает, как принципы ВОС используются для спецификации протокола *x*-доступа.

Согласно базовой модели ВОС, протокол используется между парой категорий. На рисунке D.2 взаимосвязь ВОС осуществляется только в пределах прямоугольника, изображенного пунктиром. Таким образом, взаимосвязь ВОС пары категорий имеется только внутри, но не вне этого прямоугольника.

В соответствии с моделью ВОС, *x*-клиент и *x*-сервер рассматриваются как часть прикладных процессов и имеют связанные с ними прикладные категории. Прикладные категории являются частью прикладного уровня эталонной модели ВОС и содержат множество элементов прикладных услуг. Элементы прикладных услуг обеспечивают функции взаимосвязи, в соответствии с определением услуг, для *x*-клиента и *x*-сервера и реализуют протокол *x*-доступа. При этом элемент прикладной услуги может использовать услуги, предоставляемые другими элементами прикладных услуг в той же самой прикладной категории, и услуги, предоставляемые уровнем представления эталонной модели ВОС.

На рисунке D.3 содержимое изображенного пунктиром прямоугольника показывает более детальную модель взаимосвязи ВОС между *x*-клиентом и *x*-сервером.

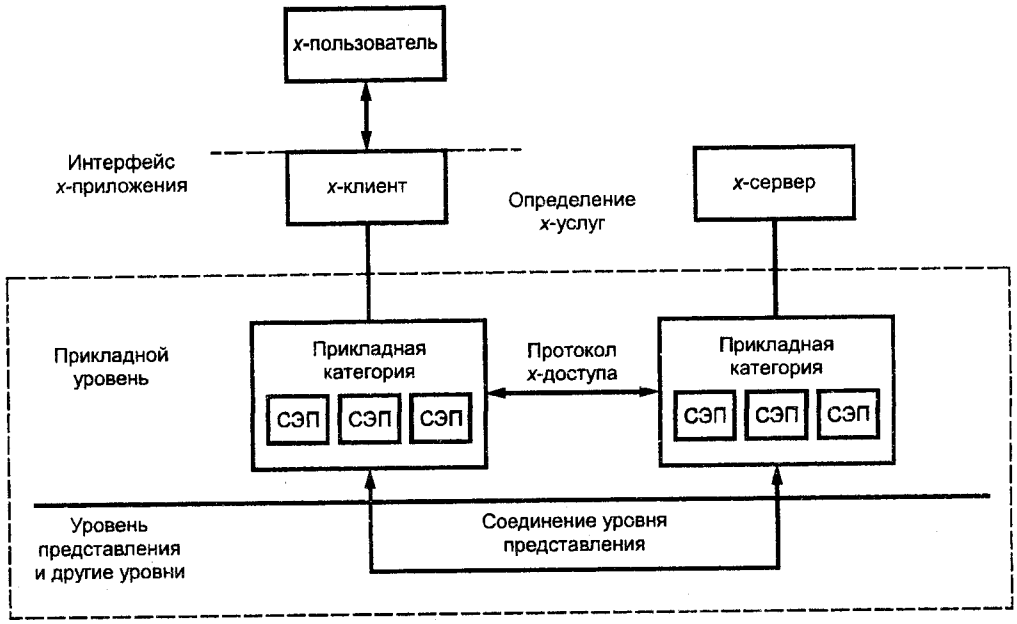


Рисунок D.3 — Распределенное учрежденческое приложение с взаимосвязью ВОС

Взаимодействие происходит между вызовами прикладных категорий. Дискретный набор взаимодействий между одной и той же парой вызовов прикладных процессов, если требуется, осуществляется между дискретными парами вызовов прикладных категорий.

Однако для большинства практических целей нет необходимости ссылаться на приведенную выше структуру. Вместо этого модель взаимосвязи *x*-клиент/*x*-сервер с помощью протокола *x*-доступа достаточна для обсуждения структуры распределенных учрежденческих приложений.

Дополнительные подробности взаимосвязи клиент/сервер описаны в D.4.

D.2.4 Объектная модель распределенного учрежденческого приложения

Взаимодействие прикладных процессов распределенного учрежденческого приложения требует совместно используемой концептуальной схемы, описывающей совместно используемый универсум.

Типичный рассматриваемый универсум состоит из объектов и взаимосвязей между ними и может предоставлять классификацию объектов.

Модель клиент/сервер рассматривается как ориентированный на приложения инструмент для разработки концептуальной схемы. Объектная модель рассматривает большее и является более абстрактной. Все компоненты распределенного учрежденческого приложения (например, почтовый ящик, место хранения файлов и т. п.) рассматриваются как объекты.

Приемы, разработанные в объектной модели, используются здесь для спецификации концептуальной схемы. Концептуальная схема является основой для определения услуг.

Ничего не значит то, что объекты другого типа используются в контексте распределенных учрежденческих приложений. Последние являются объектами данных (например, типы данных АСН.1, содержание записей, форматы личных документов и пр.). Абстрактный синтаксис этих объектов данных независимо определяется протоколами *x*-доступа и протоколами *x*-системы.

D.3 Функциональная модель

D.3.1 Несколько приложений в интегрированной системе

Интегрированная учрежденческая система образована набором учрежденческих приложений (например, передачи сообщений, записи и поиска документа, печати). Интеграция учрежденческих приложений осуществляется УП-пользователем. УП-пользователь взаимодействует с пользователем-человеком и, в интересах этого пользователя-человека, — с набором учрежденческих приложений. Взаимодействия между УП-пользователем и *x*-сервером осуществляются через *x*-клиента. Взаимодействие между клиентами осуществляется через УП-пользователя. Это показано на рисунке D.4.

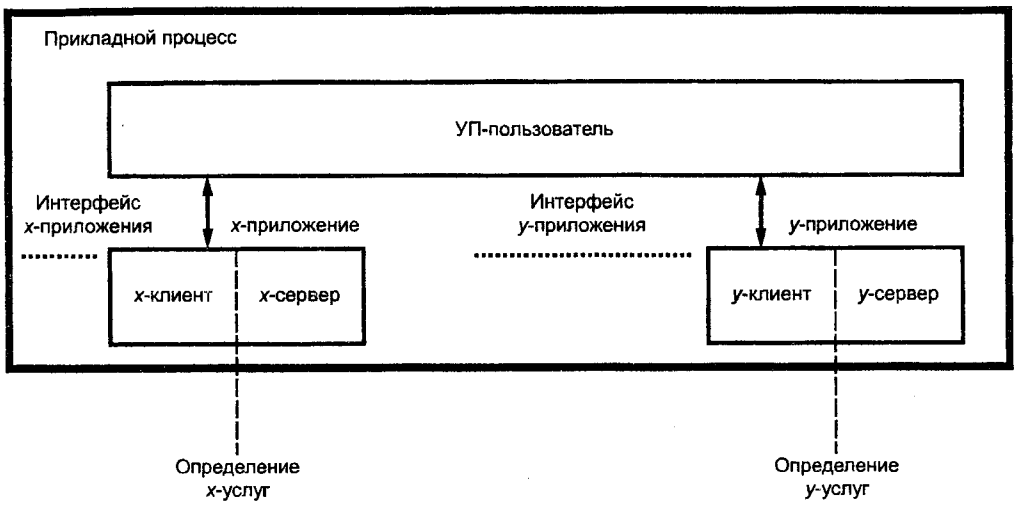


Рисунок D.4 — Несколько нераспределенных учрежденческих приложений

D.3.2 Несколько распределенных учрежденческих приложений

УП-пользователь и клиенты находятся в одном прикладном процессе, который называется прикладным процессом пользователя (см. рисунок D.5). Несколько прикладных процессов пользователей могут сосуществовать в одном узле пользователя, но в настоящем стандарте они рассматриваются по отдельности.

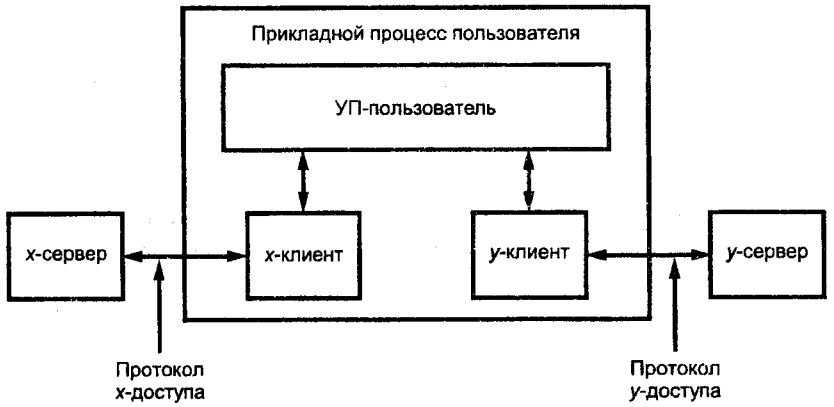


Рисунок D.5 — Несколько распределенных учрежденческих приложений

Для некоторых приложений (например, справочник, хранилище сообщений) представление каждого УП-пользователя, получившего доступ к серверу, называется агентом пользователя. Этот агент пользователя включает в себя клиентов и УП-пользователя.

D.3.3 Организация серверов

Может быть сделан второй шаг по распределению приложения — распределение функциональных возможностей серверной части *x*-приложения по нескольким *x*-серверам в нескольких узлах. Набор *x*-серверов называется *x*-системой. Каждый *x*-сервер функционально эквивалентен в том отношении, что поддерживает один и тот же протокол *x*-доступа. Каждый *x*-сервер находится в одном узле.

X-система может состоять из:

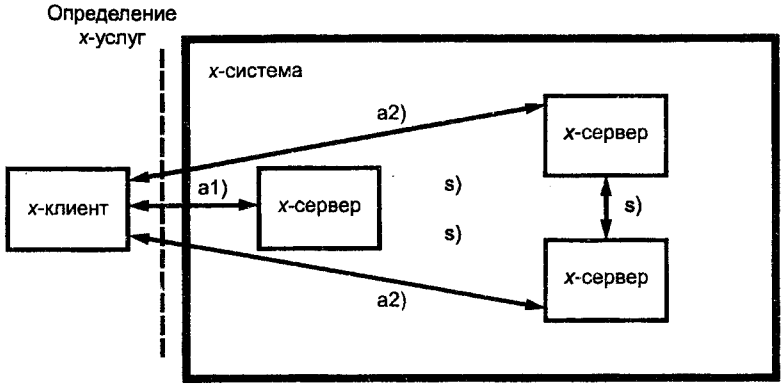
а) единственного *x*-сервера;

б) нескольких не взаимодействующих между собой *x*-серверов;

в) нескольких взаимодействующих между собой *x*-серверов.

Взаимодействие между *x*-клиентом и *x*-сервером управляется протоколом *x*-доступа (см. рисунок D.6). Эти протоколы являются предметом стандартизации конкретного *x*-приложения и не рассматриваются подробно в настоящем стандарте.

Несколько *x*-серверов, связанные сетью, могут взаимодействовать между собой для того, чтобы образовать полную *x*-систему. В этом случае они кооперируются с помощью протокола *x*-системы. Эти протоколы являются предметом стандартизации конкретного *x*-приложения и не рассматриваются подробно в настоящем стандарте.



a1) — протокол *x*-доступа;
a2) — протокол *x*-доступа (факультативен);
s) — протокол *x*-системы (если требуется).

Рисунок D.6 — *X*-система

Подмножество протоколов *x*-доступа, доступное между *x*-клиентом и конкретным *x*-сервером, зависит от определения *x*-услуг и выделенной *x*-серверу части. Например, для *x*-серверов, которые управляются распределенной информационной базой, протокол *x*-доступа может содержать механизм переадресации (это означает, что *x*-сервер возвращает *x*-клиенту сведения о том, как и с каким(и) *x*-сервером(ами) нужно контактировать для получения конкретной части информации) в случаях, когда протокол *x*-системы не существует, или *x*-сервер не может или не хочет получить эту информацию прозрачным для *x*-клиента способом.

Отметим, что *x*-клиент не знаком с протоколом *x*-системы, хотя используются такие понятия ВОС как прикладной процесс и прикладная категория. Наследуемая асимметрия модели клиент — сервер может оказаться не очень полезной при проектировании некоторых протоколов систем.

D.3.4 Кооперация между серверами

D.3.4.1 *Сервер как пользователь другого сервера*

В некоторых случаях одна система (*x*-система) будет пользоваться другой системой (*y*-системой). Такая ситуация моделируется описанием *x*-сервера, нуждающегося в использовании *y*-системы, в роли *y*-пользователя *y*-приложения. Тогда *y*-клиент существует с прикладным процессом, содержащим *x*-сервер (см. рисунок D.7).

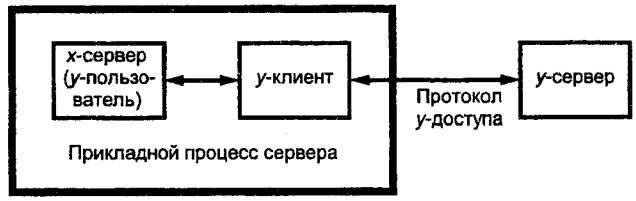


Рисунок D.7 — Сервер как пользователь другого сервера

Эта модель может быть использована и тогда, когда два сервера имеют один и тот же тип, т. е. когда *x*-сервер является пользователем другого *x*-сервера.

D.3.4.2 *Ссылочный доступ к объекту*

Для некоторых типов объектов данных (см. D.2.4) *x*-сервер действует как источник, а *y*-сервер — как сток значений данных (например, сервер хранилища сообщений может быть источником, а сервер печати — стоком печатных документов). Вообще говоря, один и тот же сервер может действовать и как источник, и как сток данных (например, сервер записи и поиска документов, сервер хранилища сообщений).

Если *x*- и *y*-клиент расположены в одном и том же месте в пределах одного прикладного процесса, а значение объекта должно быть передано от *x*-сервера к *y*-серверу, то может оказаться неэффективным передавать это значение объекта через протокол *x*-доступа от *x*-сервера к *x*-клиенту, а затем — через протокол *y*-доступа от *y*-клиента к *y*-серверу (см. рисунок D.8). В этом случае более эффективно передавать в протоколах доступа только ссылку на значение объекта. Само указанное значение объекта передается непосредственно от источника (*x*-сервера) к стоку (*y*-серверу).

Когда значение объекта передается с протоколом *x*-доступа между *x*-клиентом и *x*-сервером, для прикладного процесса пользователя может оказаться полезным управлять передачей данных, используя отличающую объект ссылку (ООС).

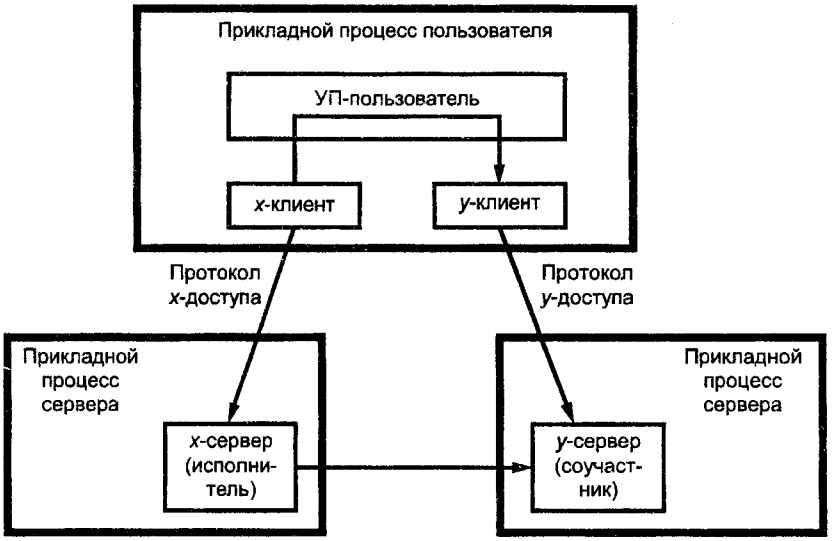


Рисунок D.8 — Ссылочный доступ к объектам

D.4 Коммуникационная модель клиент — сервер

В данном подразделе специфицированы некоторые дополнительные подробности коммуникации между *x*-клиентом и *x*-сервером на основе модели, введенной в D.2.3 и на рисунке D.3.

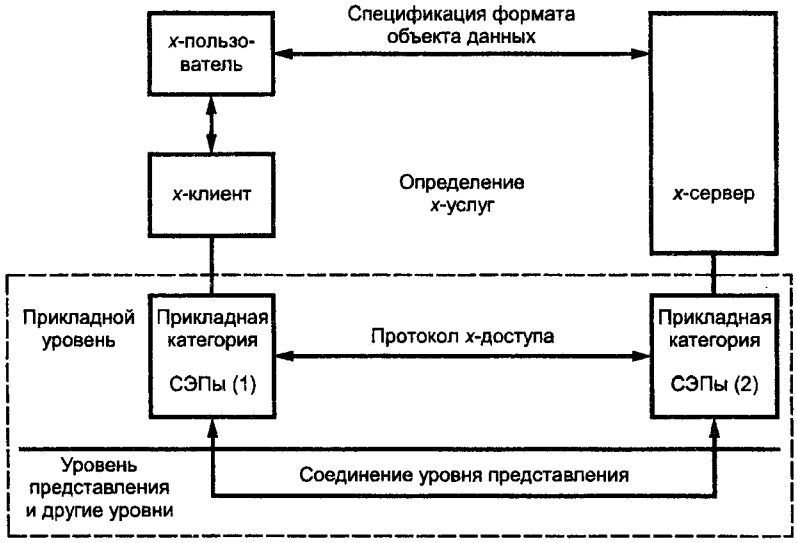
На рисунках D.9—D.12 показаны различные примеры конфигурации, которые требуют различных наборов сервисных элементов прикладного уровня (СЭП). Сервисный элемент управления ассоциацией (СЭУА) требуется в каждом наборе СЭП. Более того, в каждом наборе требуется сервисный элемент удаленных операций (СЭУО). Сервисный элемент надежной передачи (СЭНП) является факультативным. Какие дополнительные СЭП обязательны для данного набора, зависит:

а) от природы рассматриваемого распределенного учрежденческого приложения;

б) от того, связан ли набор с клиентом или с сервером;

в) от того, реализует ли набор протокол доступа или протокол системы.

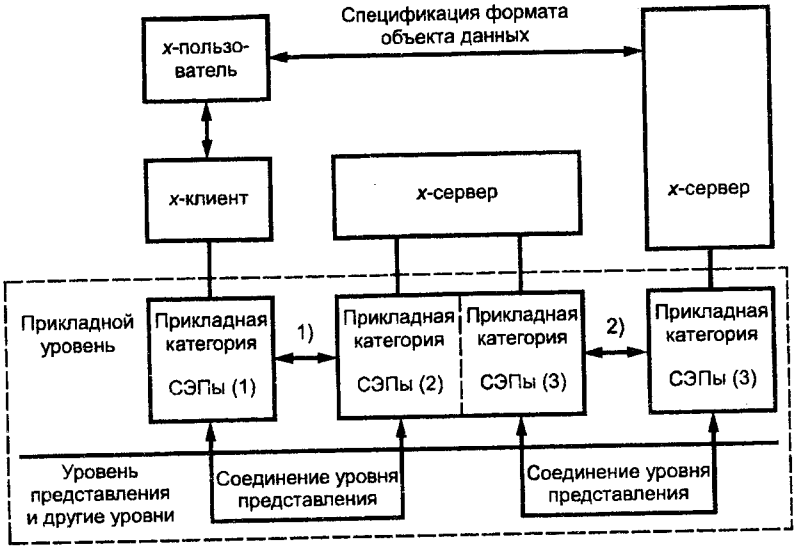
Спецификация формата объекта данных представляет собой основу кооперации между *x*-пользователем и *x*-сервером или между УП-пользователями.



СЭПы(1) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-клиентом для связи с *x*-сервером с использованием протокола *x*-доступа.

СЭПы(2) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-сервером для связи с *x*-клиентом с использованием протокола *x*-доступа.

Рисунок D.9 — Взаимодействие ВОС между *x*-клиентом и *x*-сервером



СЭПы(1) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-клиентом для связи с *x*-сервером с использованием протокола *x*-доступа.

СЭПы(2) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-сервером для связи с *x*-клиентом с использованием протокола *x*-доступа.

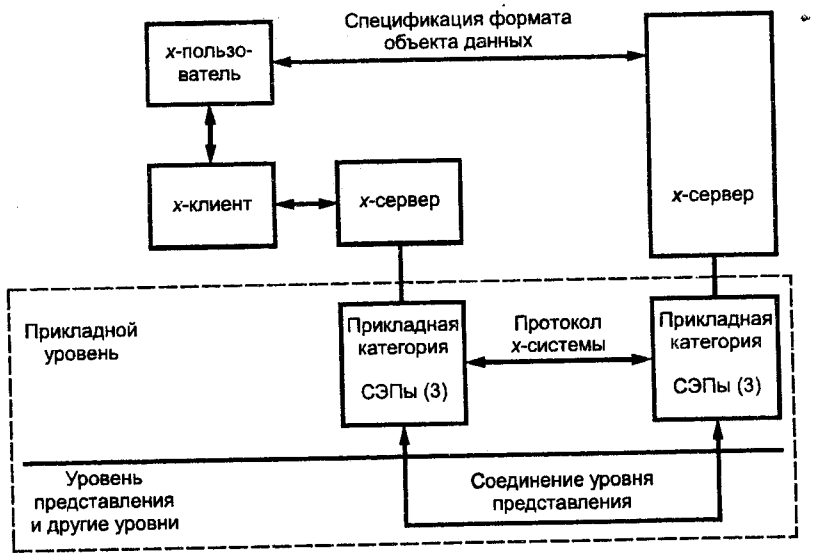
СЭПы(3) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-сервером для связи с другим *x*-сервером с использованием протокола *x*-системы.

1) Протокол *x*-доступа.

2) Протокол *x*-системы.

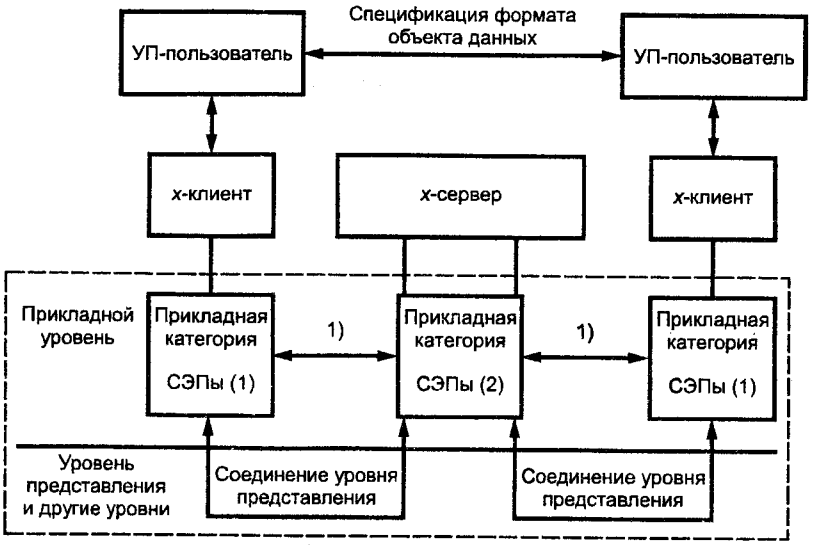
Примечание — Прикладные категории, содержащие СЭПы(2) и СЭПы(3), могут быть объединены в одну прикладную категорию, поддерживающую два прикладных контекста.

Рисунок D.10 — Взаимодействие ВОС между *x*-клиентом и *x*-сервером и между двумя *x*-серверами



СЭПы(3) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-сервером для связи с другим *x*-сервером с использованием протокола *x*-системы.

Рисунок D.11 — Взаимодействие ВОС между *x*-клиентом и совместно с ним расположенным *x*-сервером и другим *x*-сервером



СЭПы(1) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-клиентом для связи с *x*-сервером с использованием протокола *x*-доступа.

СЭПы(2) — этот набор сервисных элементов приложения выполняет функции, требуемые *x*-сервером для связи с *x*-клиентом с использованием протокола *x*-доступа.

1) Протокол *x*-доступа.

Примечание — На этом рисунке *x*-сервер использует систему сохранения и последующей передачи между двумя УП-пользователями (например, при записи и поиске документов или при передаче сообщений). В этом случае спецификация формата объекта данных предоставляет основу для кооперации между двумя УП-пользователями.

В одном прикладном контексте может быть определено несколько протоколов доступа.

Рисунок D.12 — Взаимодействие ВОС между двумя *x*-клиентами и *x*-сервером и между двумя пользователями

D.5 Функциональные категории

D.5.1 Производящие и поддерживающие приложения и возможности

В настоящем стандарте проводится различие между производящими и поддерживающими приложениями и между производящей и поддерживающей ролями приложения.

Деление приложений на «производящие» и «поддерживающие» несколько произвольно, но тем не менее полезно. Например, приложение передачи сообщений потенциально может использоваться как поддерживающее для других приложений, в частности — для обновления серверов распределенного справочника. В свою очередь, хотя приложение справочника в общем случае рассматривается как поддерживающее приложение, оно может рассматриваться и как производящее приложение, когда используется для ответа пользователю-человеку, запросившему информацию. Различие основано не столько на внутренних свойствах приложения, сколько на том, обеспечивают ли возможности приложения интересы пользователя-человека непосредственно.

Приложения, играющие поддерживающую роль, взаимодействуют для того, чтобы обеспечить пользователям стабильную, «высокоуровневую» среду. Так же, как среда программирования состоит из ряда программ (многие из которых являются общими утилитами), так и операционная сетевая среда для производящих распределенных приложений состоит из нескольких поддерживающих приложений. Эти поддерживающие приложения, построенные по тем же самым концептуальным моделям, что и производящие приложения, составляют общую операционную поддержку, которая может быть использована производящими приложениями, и предоставляют пользователям ВОС распределенные учрежденческие приложения с «высокоуровневой» средой, что позволяет пользователям не зависеть от расположения и физической адресации различных устройств и ресурсов.

Другими словами, поддерживающие приложения образуют среду поддержки высокого уровня для производящих приложений и для пользователей этих приложений.

Продуктивные приложения видны пользователю-человеку, видны как полезные, и специфически им используются. Для учрежденческого служащего производящие приложения включают в себя удаленную печать, запись и поиск документа, почту, использование справочника и пр.

D.5.2 Операционная поддержка

Общая операционная поддержка, которую могут получать производящие приложения, включает в себя, например:

а) базовое время,

б) аутентификацию и атрибуты возможностей,

в) некоторые функции справочника (например, отображение имен в адреса).

Этот список неполон. В приложении Н приведено более подробное рассмотрение данного вопроса.

D.5.3 Административное управление

Некоторые функции административного управления особенно важны, например функции, записывающие операционное поведение среды распределенных учрежденческих приложений.

Прочие функции административного управления выглядят как отдельные приложения, а для пользователя-человека — как производящие приложения. Это, в частности, справедливо для анализа и представления информации административного управления.

Административное управление может стать предметом последующей стандартизации.

D.5.4 Руководства для приложений

Некоторые поддерживающие приложения, например аутентификация, имеют существенное влияние на другие протоколы доступа. Это влияние имеет место как для информации, передаваемой протоколами, так и для последовательности, в которой осуществляются операции. Каркасная модель будет специфицировать направление, например, допустимых последовательностей операций, требуемых для аутентификации, запроса и получения доступа к серверу. Вопросы аутентификации могут быть предметом последующей стандартизации.

Другие функциональные возможности, например регистрация и подсчет, имеют влияние на прикладные проектирование и спецификацию. Вопросы безопасности регистрации и подсчета могут быть предметом последующей стандартизации.

Политику в некоторых из этих вопросов будут выбирать администраторы распределенных учрежденческих систем; таким образом, производящие приложения должны быть адаптируемыми к изменению этой политики.

В разделе 6 настоящего стандарта дано начальное руководство, которое может быть использовано в настоящее время, пока ряд вопросов, например вопросы безопасности, не определены более глубоко.

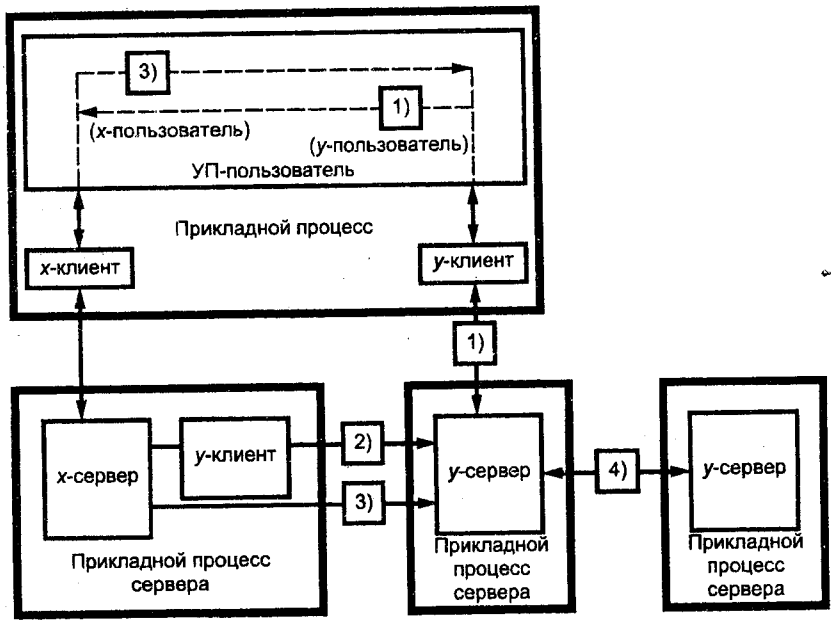
D.6 Типы взаимодействий между приложениями

В настоящем подразделе описаны и классифицированы различные типы взаимодействий в отношении к категориям поддерживающих и производящих приложений, описанных в разделе D.5.

Эти типы взаимодействий будут использоваться в последующих подразделах настоящего приложения.

Взаимодействия между *x*-пользователем и производящим *x*-приложением описаны выше и здесь не повторяются.

Взаимодействия между, например, *x*-пользователем и поддерживающим приложением могут происходить в связи с взаимодействием между УП-пользователем и производящим приложением. Это показано на рисунке D.13 как взаимодействие типа 1 между УП-пользователем и (поддерживающим) *y*-приложением. Информация,



1) Взаимодействие типа 1 между УП-пользователем или сервером, действующим в роли *x*-пользователя, и *y*-сервером, которое приводит к тому, что информация, полученная от *y*-сервера, используется во взаимодействии с *x*-приложением.

2) Взаимодействие типа 2 между *x*- и *y*-сервером, использующее *y*-клиента и протокол *y*-доступа.

3) Взаимодействие типа 3 между двумя серверами, действующими по инструкциям, переданным УП-пользователем или категорией, играющей роль *x*- и *y*-пользователя (с использованием протоколов *x*- и *y*-доступа). При передаче информации от *x*-сервера к *y*-серверу используется СДО-протокол.

4) Взаимодействие типа 4 между двумя серверами одного и того же типа, использующее системный протокол, определенный для этой цели. На рисунке показано два *y*-сервера, использующих протокол *y*-системы во взаимодействии этого типа.

Рисунок D.13 — Типы взаимодействий

полученная в ходе этого взаимодействия, используется УП-пользователем при взаимодействии с (производящим) *x*-приложением. Эти взаимодействия используют протоколы *x*- и *y*-доступа соответственно.

Взаимодействие между двумя серверами (любой из которых может быть производящим или поддерживающим) показано на рисунке D.13 как взаимодействие типа 2. *X*-сервер использует совместно с *y*-клиентом протокол *y*-доступа для доступа к *y*-серверу.

Ряд скоординированных взаимодействий существует при ссылочном доступе к объекту. При этом пользователь или категория, действующая как *x*- и *y*-пользователь, используя протоколы *x*- и *y*-доступа, дает указания *x*- и *y*-серверу об осуществлении передачи информации. На рисунке D.13 это показано как взаимодействие типа 3. Таким образом, это взаимодействие содержит два скоординированных шага. Первое действие, внутреннее для пользователя, — скоординировать установку для ссылочного доступа к объекту с *x*- и *y*-сервером, второе — запрос самого доступа.

D.7 Пример взаимодействий приложений

На рисунке D.14 приведен пример взаимодействий между УП-пользователем и приложениями, выполняющими производящие и поддерживающие функции.

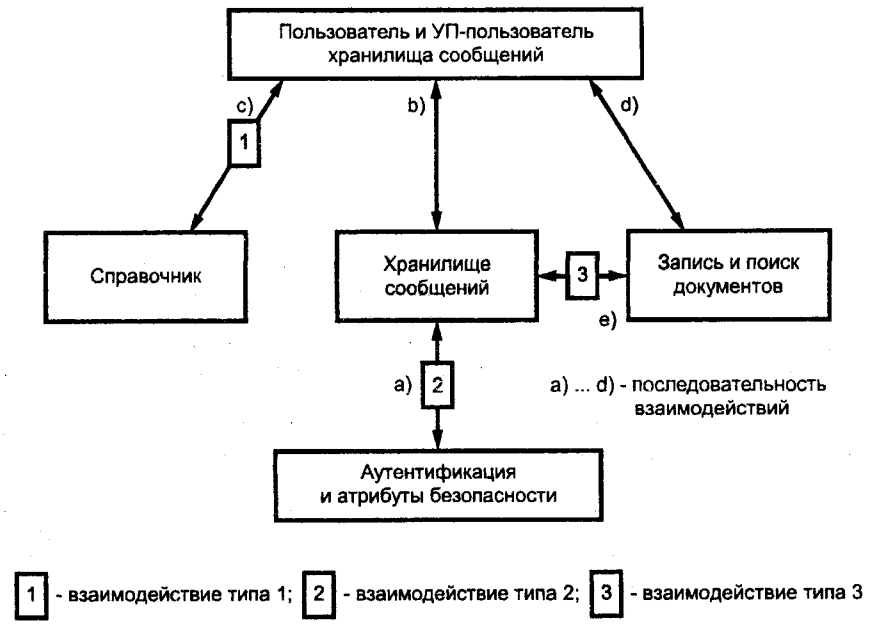


Рисунок D.14 — Взаимодействия между УП-пользователем хранилища сообщений и другими приложениями

Простейшее действие по записи в файл сообщения, полученного в хранилище сообщений, будет фактически требовать следующих операций после того, как УП-пользователь начнет работу с сервером хранилища сообщений с помощью операции связывания.

a) Сервер хранилища сообщений обращается к серверу аутентификации и атрибутов безопасности (взаимодействие типа 2).

b) УП-пользователь обращается к серверу хранилища сообщений для идентификации сообщения в качестве объекта, который должен быть предоставлен позже.

c) УП-пользователь обращается к справочнику для получения адреса сервера записи и поиска документов, способного записать сообщение (взаимодействие типа 1).

d) УП-пользователь обращается к серверу записи и поиска документов для выбора структуры, в которой сообщение должно быть записано в файл, и идентифицирует это сообщение ссылкой, полученной в результате взаимодействия b).

e) Сервер записи и поиска документов получает сообщение от сервера хранилища сообщений через СДО-протокол (взаимодействие типа 3).

ПРИЛОЖЕНИЕ Е
(справочное)

Рассмотрение идентификации

Е.1 Общие требования

Среда распределенных учрежденческих приложений характеризуется географической и логической разбросанностью таких категорий, как приложения, узлы, объекты, клиенты и серверы. Эти категории должны быть созданы для совместной работы при когерентном взаимодействии с целью эффективного решения задач распределенного учрежденского приложения.

Важным инструментом при разработке такого когерентного взаимодействия является использование такого понятия, как «наименование». «Имя» является лингвистическим понятием, идентифицирующим конкретную категорию среди множества всех категорий. Следовательно, категория, например сервер, должна иметь свое собственное имя, отличающее его от других серверов. Это отличающее имя может быть найдено в приложении справочника.

Некоторые системные категории, которые должны иметь отличающие имена, перечислены ниже:

- пользователь-человек;
- группа пользователей-людей;
- узел (пользователя и сервера);
- сервер;
- объект данных.

В дополнение к идентификации отдельных категорий понятие наименования помогает при обеспечении других функциональных возможностей, необходимых для распределенных учрежденческих приложений. Узлы пользователей и серверов, например, должны быть способны найти имена:

- сервера записи и поиска, который содержит конкретную группу файлов;
- сервера, содержащего хранилище сообщений для конкретного пользователя;
- сервера, хранящего главную копию данных (например, такого, которому позволено обновлять часть справочника);
- серверов печати, которые могут предоставить высококачественную печать и широкую каретку.

Пользователи могут выполнять свои действия потому, что справочник предоставляет следующие функциональные возможности:

- связывание имя—атрибуты — эта возможность связывает имя с порцией информации, относящейся к категории (объекту), на которую указывает имя. Связывание имя—адрес является частным случаем связывания имя—атрибуты. Большое количество категорий, которые могут находиться в системе распределенного учрежденческого приложения, делает эту возможность существенной. Эта функция аналогична «белым страницам» справочника;
- связывание атрибут—множество имен — эта возможность дает список имен категорий (объектов), имеющих данный атрибут или атрибуты. Одним из примеров являются «желтые страницы» справочника, которые могут использоваться, например, для нахождения имен всех удаленных серверов печати, имеющих высококачественную печать и широкую каретку;
- альтернативные имена — различные имена для одной и той же категории (объекта) полезны тем, что предоставляют пользователям больше гибкости при доступе к различным категориям среды распределенного учрежденческого приложения. Для того чтобы обеспечить различное написание для одного и того же объекта, может оказаться удобным использование альтернативных имен (например, MUNICH/MUNCHEN). Аналогично, в межорганизационном окружении внешне сервер может указываться по имени, отличающемся от его имени, используемом внутри организации, для того чтобы скрыть какие-либо организационные детали.

Е.2 Типы имен

Е.2.1 Обзор

Стандарты для распределенных учрежденческих приложений требуют наименования и адресации для следующей информации:

а) имена и адреса, требуемые для управления прикладными ассоциациями;

б) имена, требуемые для идентификации пользователей-людей и *х*-пользователей;

в) имена, требуемые для объектов данных.

Е.2.2 Имена и адреса для управления прикладными ассоциациями

В ГОСТ Р ИСО/МЭК 7498-3 установлены принципы для наименования и адресации. Имена и адреса, требуемые для управления прикладными ассоциациями, определены в ГОСТ 34.981.

Е.2.3 Имена для пользователей

В контексте MOTIS пользователи идентифицируются O/R-именами. O/R-имя факультативно включает в себя отличающее имя.

В контексте других распределенных учрежденческих приложений пользователи идентифицируются отличающими именами.

O/R-имена определены в ГОСТ Р ИСО/МЭК 10021-2, отличающие имена — в ИСО/МЭК 9594-2 [2].

E.2.4 Имена объектов

Если информация об объекте хранится в справочнике (например, хранилище документов), то этот объект идентифицирован отличающим именем (см. ИСО/МЭК 9594-2 [2]).

Могут существовать правила для идентификации объектов в совокупности объектов, например для идентификации документа в хранилище документов.

Для других объектов, требующих однозначной идентификации (см. E.3), используются идентификаторы объектов АСН.1.

E.3 Регистрация идентификаторов

Ряду типов объектов должны быть присвоены стандартные идентификаторы либо в соответствующем стандарте, либо уполномоченным по регистрации. Несколько примеров приведено ниже. Идентификаторы являются идентификаторами объектов, определенными в ГОСТ Р ИСО/МЭК 8824.

E.3.1 Прикладной контекст

Сервисный элемент управления ассоциацией (ГОСТ 34.981) требует идентификации прикладных контекстов.

E.3.2 Типы содержимого сообщения

Типы содержимого сообщения, определенные в MOTIS, идентифицируют различные типы содержимого, которое может передаваться по протоколу P1. Тип содержимого является примером спецификации формата объекта данных (см. рисунок D.9). Подробнее см. ГОСТ Р ИСО/МЭК 10021-2.

E.3.3 Типы частей тела сообщения

Типы частей тела сообщения идентифицируют различные типы формата/кодирования, которые могут быть найдены как часть сообщения пользователя (см. ГОСТ Р ИСО/МЭК 10021-1). Эти типы частей тела сообщения также являются спецификациями формата объекта данных.

Некоторые типы частей тела сообщения определены в ГОСТ Р ИСО/МЭК 10021-7. Другие типы могут быть определены в соответствующих стандартах или уполномоченными по регистрации.

E.3.4 Типы объектов ООС

Типы объектов ООС определены в ИСО/МЭК 10031-2 [1].

E.3.5 Типы атрибутов

Типы атрибутов должны быть определены в стандартах для конкретных распределенных учрежденческих приложений, используя понятие «атрибуты» (см. 6.4.5). Если идентификатор типа атрибута назначен для конкретного типа информации в одном приложении, то этот тип может (и должен) использоваться в других приложениях, нуждающихся в том же самом типе атрибута.

ПРИЛОЖЕНИЕ F
(справочное)

Концепции безопасности

F.1 Введение

Настоящее приложение является общим руководством.

F.1.1 Определение «безопасности»

В настоящем стандарте «безопасность» относится к характеристикам учрежденческих систем, которые дают устойчивость относительно случайностей, сбоев и злоупотреблений, намеренных или нет. Таким образом, безопасность относится к комплексу процедурных, логических и физических мер, направленных на предотвращение, выявление и исправление определенных видов случайностей, сбоев и злоупотреблений, а также к инструментам для администрирования и управления этими мерами.

В данном определении безопасность относится не только к намеренным злоупотреблениям, например угрозам системе, но и к случайностям, таким как ошибочное уничтожение сообщения и точное определение причины этого ошибочного уничтожения, чтобы ответственная сторона могла быть идентифицирована.

Таким образом, безопасность относится не только к угрозам организации, но и обеспечивает целостность работы.

F.1.2 Область действия безопасности

Многие общие потребности безопасности подразумевают обеспечение общего набора функций безопасности, не зависящего от учрежденческих приложений. Эти общие функции безопасности будут проявляться при взаимодействиях между пользователями и производящими приложениями, между производящими и поддерживающими приложениями, а также при установке, сопровождении и управлении приложениями и базовой системы. Эти функции, их взаимодействие и управление ими образуют область действия понятия безопасность в настоящем стандарте.

F.1.3 Политика безопасности

Для того чтобы быть эффективными, меры безопасности должны быть согласованными. Следовательно, организация должна определить свои меры безопасности и методы администрирования и управления этими

мерами в политике безопасности. Ответственность за реализацию политики безопасности и за управление ее эффективностью возлагается на администратора по безопасности.

Ниже приводятся примеры вопросов безопасности, которые относятся к политике безопасности. Какие именно меры применяются как часть данной политики безопасности, зависит от среды организации:

а) целостность информации, содержащейся и/или обрабатывающейся в системе;

б) конфиденциальность (избранной) информации, содержащейся и/или обрабатывающейся в системе;

в) целостность услуг и функций, обеспечиваемых системой;

г) конфиденциальность услуг и функций, обеспечиваемых системой;

д) способы получения третьей стороной гарантий для определенных операций; другими словами, необходима верификация третьей стороной целостности процессов и информации;

е) способы аутентификации отдельных пользователей и групп пользователей в соответствии с определенными правилами;

ж) управление доступом к серверам, функциям и информации, доступное в системе;

и) управление потоком информации внутри и между системами.

В общем случае организации требуется взаимодействовать с другими организациями. Разные организации будут выбирать свою собственную политику; можно сказать, что каждая политика безопасности применяется для данной области безопасности, которая находится под управлением единственного администратора безопасности. Деятельность организаций требует взаимодействия между администраторами безопасности. Эти вопросы также относятся к политике безопасности.

В некоторых случаях две области безопасности могут взаимодействовать непосредственно, в других случаях — через третью сторону. Степень доверия между областями безопасности также может варьироваться.

F.2 Требования безопасности для распределенных учрежденческих приложений

F.2.1 Общие требования безопасности

В настоящем подразделе вводятся общие требования безопасности применительно к среде распределенного учрежденческого приложения. Эти требования отражают как подразумеваемые требования — например никакая система не может быть безопасной без некоторого вида управления доступом, так и специфические требования к функциям безопасности с точки зрения пользователя — например, аутентификация источника данных.

F.2.1.1 *Защита доступа*

F.2.1.1.1 Общие положения

Управление доступом обеспечивает способы для предоставления доступа некоторым известным пользователям, а также для управления доступом этих пользователей к конкретным ресурсам для конкретных операций. Таким образом, управление доступом имеет две главных составляющих: аутентификацию пользователей и авторизацию доступа аутентифицированных пользователей. Управление доступом должно осуществляться в соответствии с политикой управления доступом, которая применяется в данной области безопасности.

F.2.1.1.2 Аутентификация

Пользователи, получающие доступ к системе распределенных учрежденческих приложений, должны быть аутентифицированы до того, как им будет предоставлен этот доступ к любому конкретному приложению. Пользователи могут также потребовать, чтобы были аутентифицированы серверы, к которым получен доступ. Пользователи могут обращаться к *x*-серверу из узла, относящегося к той же самой области безопасности, что и этот *x*-сервер, или к другой области безопасности. В любом случае должны использоваться согласованные процедуры обмена аутентифицирующей информацией.

Аутентификация может осуществляться в момент установления связи; повторная аутентификация в ходе взаимосвязи может быть затребована политикой безопасности.

F.2.1.1.3 Авторизация доступа

Узлы в среде распределенного учрежденческого приложения могут требовать использования авторизации доступа для защиты конфиденциальности и целостности объектов безопасности и целостности узла сервера. Методы авторизации могут использовать различные способы, такие как списки управления доступом, возможности и другие атрибуты безопасности, по отдельности или в комбинации.

Пользователям должен предоставляться доступ к *x*-серверам и к объектам безопасности на *x*-серверах на основе атрибутов привилегий в соответствии с политикой безопасности вовлеченной(ых) области(ей) безопасности.

Если пользователи обращаются к серверам или объектам безопасности, не относящимся к их области безопасности, то информация авторизации, так как она требуется областью безопасности сервера, должна быть передана безопасным образом.

F.2.1.2 *Защита информации данных*

Политика безопасности может потребовать, чтобы обмен или хранение данных в распределенной учрежденческой системе были защищены от внешних атак. Термин «внешняя» в данном контексте используется для указания чего-либо, отличного от обычной процедуры доступа к системе (например, прослушивание линий, кража носителей).

Защита данных охватывает как конфиденциальность (охрану секретов), так и целостность (защиту от изменений).

В среде распределенных учрежденческих приложений применяются следующие требования по защите данных:

а) защита данных при хранении (даже на переносимых носителях);

б) защита при обмене, например, управление доступом к информации, сообщениям, электронным документам и файлам, передаваемым между системами.

Защита относится к предотвращению утечки существенной информации, а также к предотвращению «загрязнения» достоверной информации недостоверной.

Независимо от физической защиты конфиденциальность может потребовать использования шифрования, а целостность — использования цифровой подписи.

Методы шифрования требуют использования шифровальных ключей. Системы, поддерживающие шифрование, должны обеспечивать безопасный метод управления ключами как в пределах области безопасности, так и между областями безопасности.

F.2.1.3 *Защита использования ресурсов*

Политика безопасности может потребовать защиты использования ресурсов. Эта защита имеет два вида: сохранение секретности использования (конфиденциальность использования) и предотвращение отказа услуги.

F.2.1.4 *Подсчет использования ресурсов*

Политика безопасности может потребовать способов, обеспечивающих подсчет использования ресурсов. Подсчет включает выборочную регистрацию операций, прослеживаемых для проверки (как затребованных, так и завершенных), а также неотвергнутые получения и отправления данных.

Неотвержение доказывается третьей стороне путем идентификации категории, полученной или отправленной в качестве примера сообщения. Оно тесно связано с целостностью данных и обычно используется в комбинации с ней.

F.2.2 Требования управления безопасностью

F.2.2.1 *Общие положения*

Системы, поддерживающие безопасность распределенных учрежденческих приложений, должны обеспечивать эксплуатирующие организации инструментами для управления средствами безопасности этих систем. Примерами таких инструментов являются безопасность установки программного обеспечения и проверка средств проверки работы средств безопасности.

Пользователи распределенного учрежденческого приложения доверяют компонентам системы осуществлять ожидаемые функции и никакие другие.

F.2.2.2 *Вопросы управления безопасностью*

Для каждого вида функций безопасности, определенного для среды распределенных учрежденческих приложений, должны рассматриваться четыре вопроса злоупотреблений или нарушений безопасности, которые вместе определяют требования управления этими функциями. Этими вопросами являются: предотвращение, выявление, восстановление и администрирование. В зависимости от желательного уровня безопасности, некоторые или все из этих вопросов проявляются в фактических реализациях.

Предотвращение основывается на правилах для управления функцией безопасности или приложением. Примером такого правила является изменение паролей каждые 3 мес.

Выявление основывается на проверке безопасности операций системы.

Проверка безопасности, относящаяся к операциям, обеспечивает администраторов безопасности обратной связью в отношении использования и эффективности функций безопасности системы.

Проверка имеет три составляющих:

а) проверка отслеживания создания и сбора;

б) проверка отслеживания анализа;

в) проверка отслеживания архивирования.

Последнее относится и к вопросам администрирования.

В среде распределенных учрежденческих приложений одно приложение может быть распределено по нескольким областям безопасности. В таком случае могут потребоваться согласованные методы проверки средств кооперации между областями.

Восстановление нарушения безопасности — реального или предполагаемого — может потребовать изменений в процедурах безопасности и информации, доступной в различных узлах распределенной системы. Следовательно, протоколы и процедуры должны поддерживать реализацию мер по восстановлению.

Администрирование имеет два аспекта, относящиеся к жизни системы:

а) сбор информации от системы;

б) создание информации для системы.

Первый аспект относится к отчетам на основе информации, зарегистрированной в защищенных базах данных. Должны быть обеспечены специальные фильтры, чтобы администратор безопасности мог получать отчет только о том виде информации, которая ему необходима. Второй аспект относится к созданию или удалению субъектов и объектов безопасности и к определению ключей, прав и паролей (по крайней мере — начального пароля).

F.3 Модель безопасности систем

F.3.1 Обзор

В безопасности распределенной системы может участвовать ряд активностей для обеспечения этой безопасности.

Модель безопасности систем подразделяет эти активности на элементы, каждый из которых играет единственную, согласованную роль в общей картине обеспечения безопасности. Эти абстрактные элементы введены как подходящие инструменты, а не как фактические реализации функций безопасности. Эти элементы называются средства безопасности.

Имея идентифицированные средства безопасности и взаимосвязь между ними, можно показать, как они могут, будучи скомбинированными, образовать поддерживающие безопасность приложения или как они становятся достоверными компонентами процессов приложений пользователя или сервера, и можно определить стандартные протоколы, пригодные для взаимодействия этих средств друг с другом и с элементами среды распределенных учрежденческих приложений.

В терминах модели ВОС рассматриваемый здесь уровень есть прикладной уровень. Рассматриваемые поддерживающие безопасность приложения связываются, используя услуги достаточной для удовлетворения своих потребностей степени безопасности. Эти потребности имеют вид гарантий (на некотором приемлемом уровне) того, что связь между ними и их недостоверными партнерами является конфиденциальной и неизменяемой и что каждая связь осуществляется с известной и идентифицированной категорией.

Модель обеспечения этих гарантий на нижних уровнях модели ВОС может стать предметом последующей стандартизации.

Имеются два фундаментально различных уровня, к которым должны относится требования безопасности распределенной системы:

а) не зависящий от приложения уровень — для управления доступом к объектам безопасности распределенной системы, таким как прикладные процессы пользователя и сервера, рабочие станции, коммуникационные ресурсы и пр.;

б) зависящий от приложения уровень — для управления доступом к конкретным объектам безопасности в приложении (таким, как документы).

Эти два уровня имеют весьма различные требования, отражающиеся в различных подмножествах политики безопасности и приводящие к привлечению разных видов защищаемых объектов безопасности и разных компонентов, ответственных за обеспечение безопасности. Иногда это рассматривается так: защищаемый объект безопасности на первом уровне может стать субъектом безопасности на другом уровне.

F.3.2 Средства безопасности

На данной стадии описания не делается никаких предположений о степени распределенности средств; они могут находиться на единственном сервере безопасности или являться частью каждого распределенного производящего или поддерживающего приложения. Не делается предположения о том, что все эти средства обязательно должны быть доступны в каждом узле распределенной системы. Они могут рассматриваться как перечень строительных блоков, из которых по выбору может быть построено то, что подходит для проводимой политики безопасности и уровня безопасности, требуемого для распределенной системы. Однако идентифицируя полный перечень, модель позволяет сделать очевидными упущения и получающиеся в результате слабости в обеспечении безопасности. В F.3.3 и H.2 показаны некоторые из этих средств безопасности, скомбинированные в три поддерживающие безопасность приложения. Ниже идентифицированы следующие средства безопасности:

F.3.2.1 *Средство представления пользователя*

Это единственная категория в распределенной системе, которая знает (независимо от любых услуг, которые могут быть использованы) о текущем доступе конкретного субъекта безопасности к защищаемому объекту безопасности. Субъект безопасности обычно является пользователем-человеком, но при некоторой политике, когда контролируются доступы серверов к другим серверам, субъект безопасности может быть *x*-сервером. Ответственность средства включает в себя:

а) передачу сведений для аутентификации;

б) начало выбора сервера;

в) приостановку неактивных пользователей.

F.3.2.2 *Средство аутентификации*

Это средство получает и проверяет полномочия субъекта безопасности, передавая свои заключения другим средствам безопасности. Субъект безопасности является пользователем-человеком, работающим через средство представления пользователя, либо небезопасное приложение, действующее как субъект безопасности (т. е. *x*-сервер, использующий *y*-сервер), либо небезопасное приложение, работающее в оперативном режиме и само ставшее доступным.

F.3.2.3 *Средство атрибутов безопасности*

Это средство обеспечивает согласование относящихся к субъекту атрибутов привилегий доступа и атрибутов управления доступом, которые должны использоваться при авторизации или отклонении запроса доступа субъектов безопасности к объектам безопасности.

F.3.2.4 *Средство авторизации*

Это средство использует контекст доступа, атрибуты привилегий (субъекта безопасности) и атрибуты управления (объекта безопасности) для авторизации или отклонения запроса доступа субъектами безопасности к объектам безопасности.

F.3.2.5 *Средство управления ассоциацией*

Это средство обеспечивает:

а) безопасность поддерживающей коммуникации, включая гарантии идентификации взаимодействующих категорий;

б) авторизацию, через средство авторизации, двух категорий для взаимодействия в интересах управляющего пользователя.

То, как функции архитектуры верхнего уровня относятся или могут быть использованы для поддержки средства управления ассоциацией, может быть предметом последующей стандартизации.

F.3.2.6 *Средство состояния безопасности*

Это средство управляет текущим динамическим состоянием аутентифицированных субъектов безопасности и объектов безопасности в распределенной системе, их ассоциаций и атрибутов привилегий, передаваемых этими ассоциациями.

F.3.2.7 *Средство проверки безопасности*

Это средство получает информацию о событиях от других средств безопасности для записи и немедленного или последующего анализа.

F.3.2.8 *Средство восстановления безопасности*

Это средство действует по информации о событии от средства проверки безопасности в соответствии с правилами, установленными администратором безопасности.

F.3.2.9 *Межобластное средство*

Это средство управляет и отображает интерпретацию идентичности субъектов и объектов безопасности, данных аутентификации и авторизации одной области безопасности в интерпретацию другой области безопасности. Помогает средству управления ассоциацией образовывать ассоциации между категориями в разных областях безопасности.

F.3.2.10 *Средство поддержки шифрования*

Это средство обеспечивает функции шифрования, используемые как другими средствами безопасности, так и приложениями для безопасности данных при хранении и передаче для следующих конкретных целей:

а) конфиденциальность данных;
б) конфиденциальность коммуникаций;
в) целостность коммуникации;
г) аутентификация источника данных;
д) неотрицание источника;
е) неотрицание получателя.

F.3.3 Поддерживающие безопасность приложения

В настоящем стандарте идентифицированы следующие три поддерживающие безопасность приложения:

а) приложение аутентификации и атрибутов безопасности; оно объединяет средства аутентификации и атрибутов безопасности;
б) межобластное приложение; оно является межобластным средством;
в) приложение проверки безопасности; оно является средством проверки безопасности.

Дополнительно могут быть определены другие поддерживающие безопасность приложения, которые реализуют другие комбинации средств безопасности, приведенных в F.3.2. Присутствие средств безопасности, например управления ассоциацией, либо в качестве отдельных категорий, либо как частей прикладных процессов пользователей и серверов, потребует дополнительных элементов протокола для распределенных учрежденческих приложений.

F.3.4 Заместитель (proxy)

В некоторых случаях доступ к *x*-серверу может быть получен *y*-сервером быстрее, чем непосредственно пользователем. Имеются две ситуации:

а) *x*-сервер работает в соответствии со своими собственными интересами;
б) *x*-сервер работает в интересах другого субъекта безопасности (например, пользователя-человека).

Первая ситуация может использоваться, например, для ограничения доступа к объектам безопасности, хранящимся на одном сервере (скажем, файловом сервере), только доступом через другой сервер (скажем, сервер базы данных). Сервер базы данных может выступать в отношении файлового сервера как субъект безопасности со своими собственными привилегиями доступа.

С другой стороны, *x*-сервер (заместитель) может действовать в интересах пользователя и получить некоторые или все из его атрибутов безопасности. Замещение может подразумевать либо доверие пользователя для одного конкретного запроса доступа, либо более широкие полномочия. Заместитель может содержать либо все подробности запроса доступа, либо ссылку на эти подробности. Данный вопрос может быть предметом последующей стандартизации.

F.4 Права доступа для распределенных учрежденческих приложений

Имеются некоторые специфические для РУП характеристики безопасности. Одним из примеров являются права доступа. Права доступа для распределенных учрежденческих приложений должны проектироваться на основе настоящего подраздела.

В таблице F.1 показан пример взаимосвязи между стандартным набором абстрактных операций и одним из возможных наборов прав доступа.

Таблица F.1 — Права доступа и допустимые операции

Операции	Права доступа			
	ВЛА	ЧИУ	ЧИ	Ч
Перечислить	x	x	x	x
Читать	x	x	x	x
Изменить	x	x	x	

Окончание таблицы F.1

Операции	Права доступа			
	ВЛА	ЧИУ	ЧИ	Ч
Копировать	x	x	x	x
Переместить	x	x		
Искать	x	x	x	x
Создать	x			
Удалить	x	x		
Зарезервировать	x	x	x	
Отметить				
Отказаться	x	x	x	x

Знак «x» означает, что соответствующая операция допустима при соответствующих правах доступа.

Примечание — Использованы следующие четыре уровня прав доступа:
ВЛА — владелец;
ЧИУ — читать—изменять—удалять;
ЧИ — читать—изменять;
Ч — только читать.

ПРИЛОЖЕНИЕ G
(справочное)

Управление

Распределенные учрежденческие приложения требуют процедурных, логических и физических мер, которые обеспечивают управление этими приложениями с возможностью планирования, организации, наблюдения, контроля и подсчета использования приложений. Эти меры могут иметь вид единственного или нескольких распределенных учрежденческих приложений в ряде открытых систем. Эти меры называются «управление».

Управление обеспечивается рядом средств, каждое из которых поддерживает один из вопросов требуемого управления. Эти средства включают в себя:

а) управление неисправностями;
б) управление подсчетом;
в) управление конфигурацией и наименованием;
г) управление выполнением;
д) управление безопасностью.

Общие вопросы управления в ВОС рассмотрены в ГОСТ Р ИСО/МЭК 7498-4.

ПРИЛОЖЕНИЕ H
(справочное)

Категории и взаимосвязь приложений

H.1 Введение

В настоящем приложении рассмотрены общие потребности учрежденческих приложений, любое из которых может в разное время играть роль поддерживающего или производящего. В первом случае приложение является поддерживающим (предоставляет услуги для другого приложения). Это другое приложение играет, вообще говоря, производящую роль, обычно предоставляя услуги, видимые пользователю-человеку. В данном приложении описано, как достигается кооперация между приложением, играющем поддерживающую роль, и приложением, предоставляющим производящие услуги.

Примечание — Далее в настоящем приложении термины «поддерживающее» и «производящее» будут использоваться для описания роли, которую некоторое приложение играет в рассматриваемое время в описываемой активности. В данном приложении не подразумевается, что приложение является внутренне поддерживающим или производящим.

Н.2 Работа поддерживающих приложений и средств

Н.2.1 Средство базового времени

При современном распространении международных организаций термин «всемирное» означает, что требуется обеспечение надежного и недвусмысленного базового времени. Это станет еще более важным в будущем, когда большое число узлов, связанных всемирными сетями, будут взаимодействовать друг с другом.

Различные компоненты любой распределенной системы должны иметь возможность получить текущее время. Это время может использоваться другими приложениями, например для установки метки времени в файлах и сообщениях, для возможности аутентификации.

Синхронизация не обязательно должна быть точной, но должна поддерживаться в разумных пределах (например, около 10 мин) по всей распределенной системе. Точность синхронизации устанавливается администраторами.

Если требуется более точное время (например, для меток времени), то оно получается путем использования локального времени узла. Может потребоваться полученные локально значения времени квалифицировать информацией о положении, указывающей источник значения времени.

Методы, которыми различные хосты получают и поддерживают правильное время, находятся вне рассмотрения настоящего стандарта.

Это средство обеспечивает всеобщее международное время, содержащее день, месяц и год в христианском летоисчислении, которое может быть задано с точностью до 1 с или 1 мин.

Н.2.2 Поддерживающие безопасность приложения

Н.2.2.1 *Введение*

В настоящем подразделе описаны поддерживающие безопасность приложения, которые вместе обеспечивают:

- не зависящий от приложения уровень безопасности управления доступом к таким объектам безопасности распределенной системы, как клиенты, серверы, рабочие станции, коммуникационные ресурсы;
- зависящий от приложения уровень безопасности управления доступом к специфическим объектам безопасности в приложении (таким, как документы).

Идентифицированы и описаны в последующих пунктах следующие поддерживающие безопасность приложения:

- приложение аутентификации и атрибутов безопасности (Н.2.2.2);
- межобластное приложение (Н.2.2.3);
- приложение проверки безопасности (Н.2.2.4).

Эти приложения безопасности не несут ответственности за:

- установление безопасной коммуникации между безопасным приложением клиента и безопасным приложением сервера (это может быть достигнуто путем установки главных шифровальных ключей как части установки системы);
- аутентификацию узлов, как подлинных и авторизованных членов распределенной системы.

Эти вопросы находятся в ведении управления безопасностью при установке, конфигурировании и переконфигурировании системы; методы управления могут быть предметом последующей стандартизации.

Эти приложения безопасности не зависят от учрежденческих приложений, которые они поддерживают и обслуживают.

Н.2.2.2 *Приложение аутентификации и атрибутов безопасности*

Это приложение предназначено для аутентификации и работы с атрибутами безопасности пользователя-человека и объектами безопасности уровня приложения, например *x*-серверами.

В общем случае аутентификация относится к пользователю-человеку, *x*-пользователю или *x*-серверу, удостоверяя их идентичность и подтверждая, что некоторая часть (засекречено) хранящейся информации находится в ведении этого пользователя-человека, *x*-пользователя или *x*-сервера. Аутентификация имеет три основных формы.

а) Пользователь-человек, *x*-пользователь или *x*-сервер создают копию порции информации, содержащей секреты этого пользователя-человека, *x*-пользователя или *x*-сервера, которые система ассоциирует с этим пользователем-человеком, *x*-пользователем или *x*-сервером посредством некоторой отображающей таблицы (классический парольный подход). Проверка допустимости этой информации может быть достигнута путем использования другого *y*-сервера (например, проверкой простых операций полномочий в справочнике ИСО).

б) Пользователь-человек, *x*-пользователь или *x*-сервер используют некоторую часть информации без ее явной передачи так, что система (которой так же об этом известно) может подтвердить, что отправитель ведает этой информацией. Идентичность отправителя вновь удостоверяется путем внутреннего отображения. (Методы шифрования, основанные на схемах согласованных ключей).

в) Пользователь-человек, *x*-пользователь или *x*-сервер используют некоторую часть информации без ее явной передачи так, что система (которой об этом неизвестно) может, используя некоторую разделяемую информацию, подтвердить, что отправитель ведает этой информацией. Идентичность отправителя удостоверяется либо путем внутреннего отображения, либо явно, на основе переданной информации. (Методы шифрования, основанные на схемах общих ключей).

Для аутентификации при обмене могут использоваться различные протоколы, например, протокол вызов/ответ, который не позволяет повторять последовательность аутентификации.

Для того чтобы обеспечить гибкий подход, с пользователями-людьми и *x*-пользователями ассоциируются конкретные значения атрибутов безопасности. Рассматриваемое приложение предназначено для обработки и хранения связанных с субъектом атрибутов привилегий и связанных с объектом управляющих атрибутов, используемых для авторизации или отклонения запрошенного доступа к объекту безопасности прикладного уровня, например к *x*-серверу. Проверка этих двух наборов значений атрибутов безопасности осуществляется средством авторизации; эта проверка может быть достаточно сложной и учитывать ряд внешних обстоятельств.

Результат аутентификации и значения атрибутов безопасности не действуют неопределенно долго. Например, при некоторой политике безопасности у пользователей-людей время от времени будут требовать повторно аутентифицироваться и переустановить атрибуты безопасности.

Пользователь приложения аутентификации и атрибутов безопасности, в большинстве вариантов политики безопасности, регистрируется в приложении проверки безопасности.

Н.2.2.3 *Межобластное приложение*

Пользователи-люди и/или компоненты распределенной системы могут находиться в разных областях безопасности. Это влияет на вопросы безопасности при взаимодействии между ними. Например, объекты безопасности, свободно доступные людям-пользователям в одной области безопасности, могут быть доступны только нескольким людям-пользователям вне этой области.

Существенно, что некоторые «секретные» объекты безопасности могут быть защищены от доступа любых пользователей из других областей безопасности. Однако может оказаться необходимым разрешить некоторым пользователям из других областей безопасности доступ к «несекретным» объектам безопасности. Если это так, то в случаях нарушения безопасности удаленных областей «секретные» объекты безопасности в данной области безопасности должны быть защищены.

Выбор объектов безопасности, которые должны быть недоступны для других областей безопасности, может быть различным в зависимости от взаимосвязей между областями безопасности. Этот выбор не является просто иерархическим. При взаимосвязи между двумя областями безопасности конкретное множество объектов безопасности может быть доступно некоторым пользователям из одной области безопасности и не доступно для всех пользователей из других областей, в то время как другие объекты безопасности, недоступные из рассматриваемой области, могут быть доступными для некоторых пользователей из других областей безопасности. Другими словами, одна область безопасности может «доверять» другой области доступ к ограниченному множеству объектов безопасности. Это множество зависит от взаимосвязей между областями безопасности объектов и субъектов безопасности.

При доверии доступа к множеству объектов безопасности между областями безопасности может оказаться необходимым окончательный уровень контроля внутри области объектов безопасности на основе идентичности и атрибутов субъектов безопасности. Область объектов безопасности должна доверять области субъектов безопасности в отношении идентичности субъектов безопасности и может использовать атрибуты субъектов безопасности области этих субъектов.

Взаимосвязи между областями безопасности являются двусторонними (и отражают деловые отношения, на которых они основаны). Следовательно, имеется логическое различие между межобластным приложением и двумя взаимодействующими областями безопасности.

Межобластное приложение отображает интерпретацию:

а) идентичности субъектов безопасности (пользователей-людей и пользователей),

б) идентичности объектов безопасности и

в) атрибутов безопасности

одной области безопасности в интерпретацию другой области.

Для этого межобластное приложение взаимодействует с приложением аутентификации и атрибутов безопасности. При любой политике безопасности межобластное приложение регистрирует активность с помощью приложения проверки безопасности.

Н.2.2.4 *Приложение проверки безопасности*

Прослеживаемость проверки является существенным требованием для любого эффективного мониторинга безопасности.

Политика безопасности определяет события, которые должны регистрироваться производящими и поддерживающими приложениями. Они регистрируют события, используя приложение проверки безопасности.

Прослеженные результаты проверки безопасности надежно хранятся для последующего анализа пользователем-человеком, имеющим соответствующие атрибуты безопасности. (Некоторые события могут вызывать немедленную регистрацию и сигнал тревоги для специально назначенных пользователей-людей или компонентов распределенной системы).

Н.2.3 Приложение справочника

Приложение справочника играет ключевую роль в обеспечении пользователей стабильным «высокоуровневым» представлением среды распределенных учрежденческих приложений в смысле независимости этого представления от изменений сетевых и физических средств нижележащих уровней.

Когда это возможно, распределенные учрежденческие приложения должны использовать приложение справочника, определенное в стандартах серии ГОСТ Р ИСО/МЭК 9594, например для следующих функций:

а) Основной функцией приложения справочника является размещение имен в адреса уровня представления. Пользователь ссылается на категорию в сети по имени, а приложение, через сервер справочника,

разрешает направить его в адрес представления, который обычно тесно связан с «физическим размещением». Тем самым пользователю предоставляется дружественный интерфейс с системой, так как категории могут указываться легко запоминаемыми именами.

б) Приложение справочника также обеспечивает способ управления группами (списками) категорий (например, добавленные или удаленные компоненты, рекурсивное расширение и проверка членства). Эти группы имеют много приложений, например при обработке сообщений.

в) Кроме того, приложение справочника обеспечивает ограниченные, но широко используемые средства для локализации категорий по ссылкам на некоторые их атрибуты (например, поиск в заданной области множества категорий, являющихся *x*-серверами некоторого *x*-приложения, или связь с первым доступным экземпляром *x*-сервера).

Н.2.4 Ссылочный доступ к объекту

Некоторые учрежденческие приложения действуют как источники или стоки объектов, например файлов, документов или частей сообщений Р2.

Передача больших значений объектов данных концептуально привлекает к участию три стороны: пользователя, который запрашивает или дирижирует передачей, источник, создающий значение объекта данных, и сток, потребляющий значение объекта данных.

При двусторонней передаче пользователь является либо источником, либо стоком данных, а вторая сторона, соответственно, — стоком или источником. В некоторых случаях пользователь может быть промежуточным стоком, а в последующем — промежуточным источником некоторого значения объекта данных; т. е. значение объекта данных передается от источника к пользователю, а затем — от пользователя к стоку. В этом случае будет более эффективно, если пользователь будет дирижировать (прямой) передачей данных от источника к стоку.

По аналогии с языками программирования высокого уровня, услуги удаленных операций источника и стока используют в случае двусторонней передачи метод «аргумент или результат по значению» и метод «аргумент или результат по ссылке» — в случае передачи данных по ссылке.

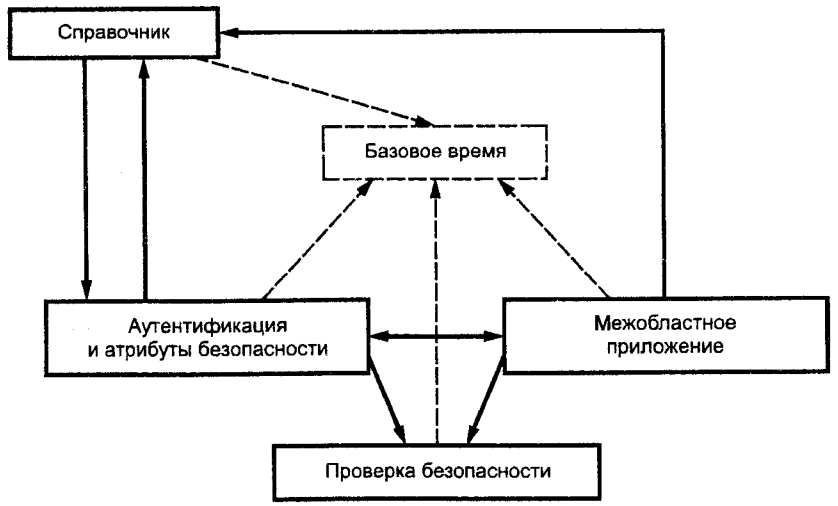
Ссылочный доступ к объекту не заменяет другие типы взаимодействия между серверами. На практике, когда *x*-серверам некоторого *x*-приложения необходимо некоторое взаимодействие, отличное от простой передачи данных, может быть определен конкретный протокол *x*-системы для данного приложения.

Когда серверам различных приложений требуется некоторое взаимодействие, отличное от простой передачи данных, они используют протоколы доступа, как описано в D.3.3.2, D.6 и D.7.

Н.2.5 Взаимодействия между поддерживающими приложениями

Имеется много различных взаимодействий между поддерживающими приложениями. Ниже описаны некоторые наиболее важные из них.

Каждое поддерживающее приложение использует базовое время. Эти взаимодействия изображены пунктирными стрелками на рисунке Н.1. Как именно получается значение времени, находится вне рассмотрения настоящего стандарта. Когда время доступно локально, это делается локальными методами; единственным требованием является синхронизация с базовым временем, как описано в Н.2.1.



Примечание — Стрелки и содержимое прямоугольников соответствуют следующей взаимосвязи клиент—сервер:

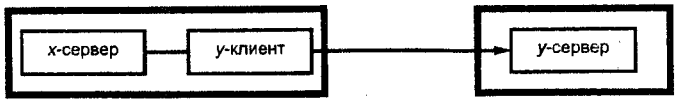


Рисунок Н.1 — Взаимодействия между поддерживающими приложениями

Приложение справочника использует приложение аутентификации и атрибутов безопасности для того, чтобы справочник мог аутентифицировать своих пользователей. Это взаимодействие типа 2.

Если приложение аутентификации и атрибутов безопасности распределено по ряду серверов, то любой их этих серверов может взаимодействовать как пользователь с приложением справочника для того, чтобы найти адрес представления другого сервера этого приложения (взаимодействие типа 2).

Н.3 Поддержка производящих приложений

Н.3.1 Введение

В настоящем разделе даны указания о том, как производящие приложения используют поддерживающие приложения. Примеры включают в себя приложения передачи сообщений, хранения сообщений, записи и поиска документов и печати документов.

Описанные ниже поддерживающие взаимодействия, в общем случае, не видны человеку—пользователю этих производящих приложений, т. е. пользователи-люди не обязаны явно специфицировать требования к этим взаимодействиям.

Упоминаемые ниже типы взаимодействий описаны в D.6.

Н.3.2 Поддержка приложения передачи сообщений

Н.3.2.1 *Описание приложения передачи сообщений*

Приложение передачи сообщений позволяет пользователям отправлять и получать сообщения различной длины и содержания. Оно непосредственно доступно всем пользователям системы распределенных учрежденческих приложений, позволяя обмениваться сообщениями между пользователями, разделенными как большими, так и малыми расстояниями. Передача сообщений от отправителя к получателю осуществляется методом сохранения и последующей передачи.

Приложение передачи сообщений определено в ГОСТ Р ИСО/МЭК 10021-3.

Приложение передачи сообщений может обеспечивать специфические возможности, такие как поддержка списков рассылки группе получателей под одним именем.

Н.3.2.2 *Работа приложения передачи сообщений*

Приложение передачи сообщений требует поддержки базового времени, приложения аутентификации и атрибутов безопасности и приложения справочника.

Следующие взаимодействия могут быть вовлечены в типичное использование приложения передачи сообщений:

а) пользователь приложения передачи сообщений обращается к базовому времени;

б) пользователь приложения передачи сообщений обращается к приложению справочника для получения адреса представления сервера передачи сообщений (взаимодействие типа 1);

в) пользователь приложения передачи сообщений обращается к приложению аутентификации и атрибутов безопасности для получения атрибутов безопасности при доступе к серверу передачи сообщений (взаимодействие типа 1);

г) пользователь приложения передачи сообщений представляет сообщение на сервер передачи сообщений;

д) сервер передачи сообщений обращается к приложению аутентификации и атрибутов безопасности для аутентификации пользователя приложения передачи сообщений (взаимодействие типа 2);

е) сервер передачи сообщений обращается к базовому времени для создания временной метки для каждого сообщения;

ж) сервер передачи сообщений обращается к приложению справочника для раскрытия «получателей», которые являются списками рассылки (взаимодействие типа 2);

и) сервер передачи сообщений обращается к приложению справочника для получения адресов представления всех серверов хранения сообщений получателей (взаимодействие типа 2).

Н.3.3 Поддержка хранилища сообщений

Н.3.3.1 *Описание доступа к хранилищу сообщений*

Приложение хранения сообщений тесно связано с приложением передачи сообщений. Приложение передачи сообщений фактически опускает сообщение в «почтовый ящик», связанный с пользователем, а приложение хранения сообщений позволяет этому пользователю получить письмо. Приложение хранения сообщений определено в ГОСТ Р ИСО/МЭК 10021-5.

Н.3.3.2 *Работа приложения хранения сообщений*

Приложение хранения сообщений требует поддержки базового времени, приложения аутентификации и атрибутов безопасности и приложения справочника.

Следующие взаимодействия могут быть вовлечены в типичное использование приложения хранения сообщений:

а) агент пользователя (см. ГОСТ Р ИСО/МЭК 10021-2) обращается к базовому времени;

б) агент пользователя обращается к приложению справочника для получения адреса представления хранилища сообщений (взаимодействие типа 1);

в) агент пользователя обращается к приложению аутентификации и атрибутов безопасности для получения атрибутов безопасности для доступа к хранилищу сообщений (взаимодействие типа 1);

г) агент пользователя запрашивает сообщения из хранилища сообщений;

д) хранилище сообщений может обратиться к приложению аутентификации и атрибутов безопасности для аутентификации агента пользователя (взаимодействие типа 1);

е) хранилище сообщений использует средство авторизации для проверки допустимости доступа;

ж) хранилище сообщений возвращает пользователю сообщение.

Н.3.4 Поддержка приложения записи и поиска документов

Н.3.4.1 *Описание приложения записи и поиска документов*

Приложение записи и поиска документов обеспечивает возможности записи и поиска документов для нескольких пользователей в распределенной системе.

Приложение записи и поиска документов также обеспечивает управление доступом к хранению документов.

Н.3.4.2 *Работа приложения записи и поиска документов*

В пределах области безопасности приложение записи и восстановления документов требует поддержки базового времени, приложения аутентификации и атрибутов безопасности и приложения справочника.

Следующие взаимодействия могут быть вовлечены в использование (в данном примере — для поиска) приложения записи и поиска документов:

а) пользователь приложения записи и поиска документов обращается к базовому времени;

б) пользователь приложения записи и поиска документов обращается к приложению справочника для получения адреса представления требуемого ему сервера записи и поиска документов (взаимодействие типа 1);

в) пользователь приложения записи и поиска документов обращается к приложению аутентификации и атрибутов безопасности для получения атрибутов безопасности для доступа к серверу записи и поиска документов (взаимодействие типа 1);

г) пользователь приложения записи и поиска документов запрашивает документ от сервера записи и поиска документов;

д) сервер записи и поиска документов может обратиться к приложению аутентификации и атрибутов безопасности для аутентификации пользователя приложения записи и поиска документов (взаимодействие типа 2);

е) приложение записи и поиска документа возвращает документ пользователю данного приложения.

Н.3.5 Поддержка приложения печати документов

Н.3.5.1 *Описание приложения печати документов*

Приложение печати документов обеспечивает возможности совместного использования дорогих, с большими возможностями, устройств отображения для нескольких пользователей в распределенной системе.

Н.3.5.2 *Работа приложения печати документов*

В пределах области безопасности приложение печати документов требует поддержки базового времени, приложения аутентификации и атрибутов безопасности и приложения справочника.

Следующие взаимодействия могут быть вовлечены в использование (в данном примере — для печати документа) приложения печати документов:

а) пользователь приложения печати документов обращается к базовому времени;

б) пользователь приложения печати документов обращается к приложению справочника для получения адреса представления требуемого ему сервера печати (взаимодействие типа 1);

в) пользователь приложения записи и поиска документов обращается к приложению аутентификации и атрибутов безопасности для получения атрибутов безопасности для использования сервера печати (взаимодействие типа 1);

г) пользователь приложения печати документов передает документ серверу печати документов;

д) сервер печати обращается к приложению аутентификации и атрибутов безопасности для аутентификации пользователя приложения печати документов (взаимодействие типа 2);

е) сервер печати помещает документ в очередь для печати;

ж) приложение печати документов извещает пользователя о завершении запроса.

ПРИЛОЖЕНИЕ J
(справочное)

Модель объекта

J.1 Введение

Как клиент, так и сервер рассматриваются как содержащие объекты.

Все внешне наблюдаемое поведение объекта описывается в спецификации типа объектов. Объекты, описанные одной и той же спецификацией типа объектов, называются экземплярами этого типа объектов. Спецификация типа объектов является множеством операций типа, каждая из которых отлична от других и логически полна. Каждая операция типа задается на уровне абстракции, на котором определяется, что происходит, но не как это происходит.

Результат операции типа может зависеть от состояния объекта; состояние является результатом подмножества предшествующих осуществленных операций типа. Изменения состояния объекта и их влияние на операции типа описываются в спецификации типа объектов.

X-сервер моделируется совокупностью одного или нескольких объектов различных типов объектов. Эти объекты называются объектами *x*-сервера. Внешне наблюдаемое поведение *x*-сервера описывается спецификациями типов объектов из указанной совокупности объектов *x*-сервера. Аналогично, внешне наблюдаемое поведение *x*-клиента может быть описано рядом спецификаций типов объектов. Эти объекты называются объектами *x*-клиента. Все взаимодействия между клиентом и сервером описываются определением *x*-услуг, которое полностью выводится из спецификаций типов объектов клиента и сервера.

Соединение между вызовом клиента и вызовом сервера устанавливается операцией связывания. Операция связывания включает в себя ряд взаимодействий:

а) с прикладным контекстом, который определяет подмножество доступных типов операций из определения *x*-услуг и вытекающие из них правила, установленные в определении *x*-услуг;

б) с конкретным множеством экземпляров объектов *x*-сервера и, факультативно, *x*-клиента.

В рамках этого ряда взаимодействий и ограничений, установленных операцией связывания, могут быть дальнейшие ограничения, упрощения и изменения множества рассматриваемых экземпляров объектов.

Операция связывания и последующая операция развязывания описаны в спецификациях типов объектов *x*-сервера и являются частью определения *x*-услуг.

Объект *x*-клиента или *x*-сервера может, при более близком рассмотрении, состоять из нескольких подчиненных объектов. Эти подчиненные объекты могут иметь соответствующие типы операций. Подмножество взаимодействий может быть ограничено конкретным множеством этих подчиненных объектов. Идентификация этих объектов и типов операций над ними выражается в параметрах типов операций спецификаций типов объектов.

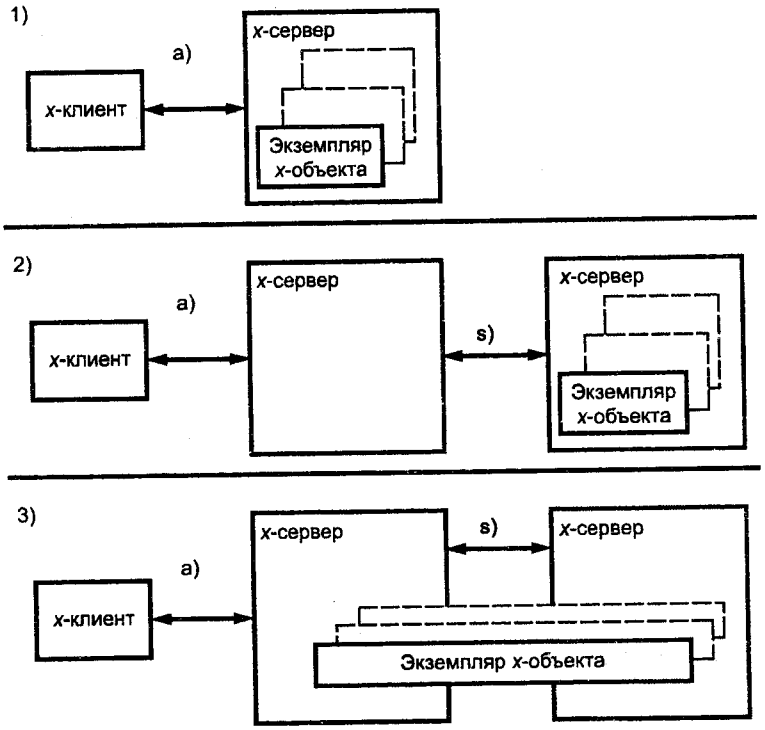
J.2 Взаимосвязь между *x*-сервером и экземплярами объектов

Объекты сервера *x*-системы далее называются *x*-объектами. В зависимости от природы *x*-объектов в осуществление операции над экземпляром *x*-объекта в интересах *x*-клиента может быть вовлечено один или несколько *x*-серверов.

На рисунке J.1 показаны три различных ситуации отношений экземпляров *x*-объектов и *x*-серверов:

а) один или несколько экземпляров типа *x*-объектов целиком хранятся на *x*-сервере, с которым взаимодействует *x*-клиент (например, экземпляры сообщений хранятся на сервере хранения сообщений);

б) один или несколько экземпляров типа *x*-объектов целиком хранятся на *x*-сервере, отличном от того, с которым взаимодействует *x*-клиент (например, часть информации справочника в информационной базе сервера справочника). Доступ между *x*-серверами к экземплярам *x*-объектов осуществляется либо с помощью протокола *x*-системы, либо через *x*-клиента и протокол *x*-доступа;



а) Протокол *x*-доступа; s) Протокол *x*-системы или *x*-клиент и протокол *x*-доступа

Рисунок J.1 — Сервер и экземпляры объектов

в) один или несколько экземпляров типа *x*-объектов хранятся на нескольких *x*-серверах (например, вся или часть распределенной базы данных). Доступ к объектам координируется совместно используемыми *x*-серверами либо с помощью протокола *x*-системы, либо через *x*-клиента и протокол *x*-доступа.

С точки зрения *x*-клиента, между этими тремя ситуациями нет различия. *X*-клиент просто задает типы операций относительно *x*-сервера, с которым он взаимодействует, используя протокол *x*-доступа. Не обязательно все три рассмотренных варианта должны поддерживаться каждым распределенным учрежденческим приложением.

ПРИЛОЖЕНИЕ К
(справочное)

Стандартный набор операций

К.1 Введение

В данном приложении подробнее рассмотрен пример стандартного набора абстрактных операций РУП, введенный в 6.6.

Набор абстрактных операций содержит следующие операции:

а) перечислить;
б) читать;
в) изменить;
г) копировать;
д) переместить;
е) искать;
ж) создать;
и) удалить;
к) зарезервировать;
л) отметить;
м) отказаться.

К.2 Описание операций

Примечания

1 «Идентификатор объекта» является именем объекта или ООС. В случае операции потребления или операции доступа используется ООС.

2 Все операции могут содержать специфическую для данной операции информацию управления безопасностью.

К.2.1 Перечислить

Операция «перечислить» используется для получения списка компонентов в заданном объекте.

Аргумент операции «перечислить» может содержать следующие компоненты:

а) идентификатор объекта;
б) селектор;
в) запрошенные атрибуты;
г) индикация запроса (запрашивается значение объекта или ООС).

Результат операции «перечислить» может содержать компонент:

значение списка или ООС списка.

К.2.2 Читать

Операция «читать» используется для получения значения или ООС и атрибутов заданного(ых) объекта(ов).

Аргумент операции «читать» может содержать следующие компоненты:

а) идентификатор объекта;
б) селектор;
в) запрошенные атрибуты;
г) индикация запроса (запрашивается значение объекта или ООС).

Результат операции «читать» может содержать компонент:

значение или ООС объекта, который должен быть прочитан.

К.2.3 Изменить

Операция «изменить» используется для изменения значения и/или атрибутов заданного объекта.

Аргумент операции «изменить» может содержать следующие компоненты:

а) идентификатор объекта;
б) изменения (удаление, замену или добавление).
Результат операции «изменить» может не содержать никаких компонентов.

К.2.4 Копировать

Операция «копировать» используется для копирования объекта.
Аргумент операции «копировать» может содержать следующие компоненты:
а) идентификатор объекта;
б) селектор;
в) запрошенные атрибуты;
г) идентификатор объекта назначения.
Результат операции «копировать» может не содержать никаких компонентов.

К.2.5 Переместить

Операция «переместить» используется для перемещения объекта.
Аргумент операции «переместить» может содержать следующие компоненты:
а) идентификатор объекта;
б) селектор;
в) запрошенные атрибуты;
г) идентификатор объекта назначения.
Результат операции «переместить» может не содержать никаких компонентов.

К.2.6 Искать

Операция «искать» используется для идентификации объектов, которые соответствуют заданным условиям, и помещения результата в заданном объекте назначения.
Аргумент операции «искать» может содержать следующие компоненты:
а) идентификатор объекта;
б) селектор;
в) запрошенные атрибуты;
г) идентификатор объекта назначения;
д) индикация запроса (запрашивается значение объекта или ООС).
Результат операции «искать» может содержать компонент:
идентификатор объекта назначения.

К.2.7 Создать

Операция «создать» используется для создания объекта. Факультативно может быть допустимо присвоение значения и атрибутов.
Аргумент операции «создать» может содержать следующие компоненты:
а) идентификатор объекта;
б) поддерживаемые атрибуты;
в) значение объекта и/или атрибутов;
г) индикация запроса (запрашивается значение объекта или ООС).
Результат операции «создать» может содержать компонент:
идентификатор объекта.

К.2.8 Удалить

Операция «удалить» используется для удаления объекта.
Аргумент операции «удалить» может содержать следующие компоненты:
а) идентификатор объекта;
б) селектор.
Результат операции «удалить» может не содержать никаких компонентов.

К.2.9 Зарезервировать

Операция «зарезервировать» используется для «фиксации» объекта для предотвращения его изменения или удаления другими пользователями.
Аргумент операции «зарезервировать» может содержать следующие компоненты:
а) идентификатор объекта;
б) запрошенное действие (зарезервировать или освободить).
Результат операции «зарезервировать» может не содержать никаких компонентов.

К.2.10 Отметить

Операция «отметить» используется для задания информации о статусе изменений заданного объекта.
Аргумент операции «отметить» может содержать компонент:
идентификатор объекта.
Результат операции «отметить» может не содержать никаких компонентов.

К.2.11 Отказаться

Операция «отказаться» используется для отказа от выполнения ранее запрошенной задачи (например, «найти»).
Аргумент операции «отказаться» определяет данную задачу.
Результат операции «отказаться» может не содержать никаких компонентов.

ПРИЛОЖЕНИЕ L
(справочное)

Библиография

[1] ИСО/МЭК 10031-2—91 Информационная технология. Текстовые и учрежденческие системы. Часть 2. Модель приложений распределенного учреждения

[2] ИСО/МЭК 9594-2—95 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 2. Модели

УДК 681.324:006.354 ОКС 35.100.70 П85 ОКСТУ 4002

Ключевые слова: взаимосвязь открытых систем, прикладной уровень, текстовые и учрежденческие системы, распределенные системы, распределенная обработка данных, общая модель, протокол, проектирование протоколов

Редактор *В.П. Огурцов*
Технический редактор *В.Н. Прусакова*
Корректор *В.С. Черная*
Компьютерная верстка *В.И. Грищенко*

Изд. лиц. № 02354 от 14.07.2000. Сдано в набор 19.12.2000. Подписано в печать 29.01.2001. Усл. печ. л. 5,12.
Уч.-изд. л. 4,80. Тираж 300 экз. С 174. Зак. 105.

ИПК Издательство стандартов, 107076, Москва, Колодезный пер., 14.
Набрано в Издательстве на ПЭВМ
Филиал ИПК Издательство стандартов — тип. "Московский печатник", 103062, Москва, Лялин пер., 6.
Плр № 080102