ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

ПЕРЕДАЧА ТЕКСТА. СИСТЕМЫ ОБМЕНА ТЕКСТАМИ, ОРИЕНТИРОВАННЫЕ НА СООБЩЕНИЯ (MOTIS)

Часть 3. СОГЛАШЕНИЯ ПО ОПРЕДЕЛЕНИЮ АБСТРАКТНЫХ УСЛУГ

Издание официальное

БЗ 8—98

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Московским научно-исследовательским центром (МНИЦ) Государственного комитета Российской Федерации по связи и информатизации

ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 10 июля 1998 г. № 288

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 10021-3—90 «Информационная технология. Передача текста. Системы обмена текстами, ориентированные на сообщения (MOTIS). Часть 3. Соглашения по определению абстрактных услуг»

4 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 1998

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта России

Содержание

Введение ... IV
Глава первая. ВВЕДЕНИЕ ... 1
1 Назначение ... 1
2 Нормативные ссылки ... 1
3 Определения ... 1
4 Сокращения ... 2
5 Соглашения ... 2
5.1 АСН.1 ... 2
5.2 Термины ... 3
Глава вторая. СОГЛАШЕНИЯ ПО ОПРЕДЕЛЕНИЮ АБСТРАКТНЫХ УСЛУГ ... 3
6 Общее описание ... 3
7 Абстрактные модели ... 3
7.1 Абстрактные объекты ... 3
7.2 Абстрактные порты ... 4
7.3 Абстрактные услуги ... 5
7.4 Абстрактные уточнения ... 5
8 Абстрактные услуги ... 6
8.1 Абстрактные процедуры ... 6
8.2 Операции абстрактной связки ... 7
8.3 Операции абстрактной развязки ... 8
8.4 Абстрактные операции ... 8
8.5 Абстрактные ошибки ... 9
Глава третья. РЕАЛИЗАЦИИ АБСТРАКТНЫХ УСЛУГ ... 9
9 Общее описание ... 9
10 Реализации ВОС ... 10
10.1 Реализации СУО ... 10
10.2 Реализации не-СУО ... 11
11 Собственные реализации ... 11
11.1 Распределенные реализации ... 11
11.2 Нераспределенные реализации ... 11
Приложение А Пример использования нотации абстрактных услуг ... 12
А.1 Назначение объектных идентификаторов ... 12
А.2 Уточнение желтой среды ... 13
А.3 Определение абстрактной услуги желтой системы ... 14
А.4 Уточнение зеленой среды ... 15
А.5 Определение абстрактной услуги зеленой системы ... 17
А.6 Уточнение желтой системы ... 18
А.7 Реализация желтой системы ... 20
А.8 Реализация зеленой системы ... 21
Приложение В Справочное определение объектных идентификаторов ... 23
Приложение С Справочное определение нотации ... 24
Приложение D Различия между ГОСТ Р ИСО/МЭК 10021-3 и Рекомендацией Х.407 МККТТ ... 26
Приложение Е Алфавитный указатель ... 27

Введение

Настоящая часть ГОСТ Р ИСО/МЭК 10021 — одна из совокупности частей ГОСТ Р ИСО/МЭК 10021 [стандарты, распространяющиеся на системы обмена текстами, ориентированные на сообщения (MOTIS)]. Стандарты серии ГОСТ Р ИСО/МЭК 10021 содержат исчерпывающую спецификацию обработки сообщений, охватывающую любое количество взаимодействующих открытых систем.

ГОСТ Р ИСО/МЭК 10021 состоит из нескольких частей, объединенных названием «Информационная технология». Передача текста. Системы обмена текстами, ориентированные на сообщения (MOTIS)»:

- Часть 1. Общее описание системы и службы.
- Часть 2. Общая архитектура.
- Часть 3. Соглашения по определению абстрактных услуг.
- Часть 4. Система передачи сообщений: определение абстрактных услуг и процедуры.
- Часть 5. Хранилище сообщений: определение абстрактных услуг.
- Часть 6. Спецификации протокола.
- Часть 7. Система межперсональных сообщений.

Система обработки сообщений, ставшая возможной благодаря указанным международным стандартам, — сложная задача распределенной обработки информации, многие компоненты которой сами обладают этими характеристиками.

Настоящая часть ГОСТ Р ИСО/МЭК 10021 устанавливает соглашения, позволяющие решать задачи распределенной обработки информации при обработке сообщений, и может быть полезна также в других применениях.

Текст настоящей части ГОСТ Р ИСО/МЭК 10021 является объектом совместного соглашения между МККТТ и ИСО. Соответствующей спецификацией МККТТ является Рекомендация МККТТ X.407.

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ПЕРЕДАЧА ТЕКСТА. СИСТЕМЫ ОБМЕНА ТЕКСТАМИ, ОРИЕНТИРОВАННЫЕ НА СООБЩЕНИЯ (MOTIS)

Ч а с т ь 3. Соглашения по определению абстрактных услуг

Information technology
Text communication Message-oriented text interchange systems (MOTIS)
Part 3 Abstract service definition conventions

Дата введения 1999—01—01

Г л а в а п е р в а я. ВВЕДЕНИЕ

1 НАЗНАЧЕНИЕ

Настоящая часть ГОСТ Р ИСО/МЭК 10021 устанавливает соглашения, используемые для определения задач распределенной обработки информации, возникающих при обработке сообщений.

Настоящая часть ГОСТ Р ИСО/МЭК 10021 имеет следующую структуру. Первая глава — это данная вводная глава. Вторая глава устанавливает соглашения по абстрактному определению задачи распределенной обработки информации. В третьей главе приведены принципы конкретной реализации аспектов обмена данными в таких задачах, например посредством протоколов взаимосвязи открытых систем (ВОС) В приложениях содержится важная дополнительная информация.

В настоящей части ГОСТ Р ИСО/МЭК 10021 отсутствуют требования к соответствию.

2 НОРМАТИВНЫЕ ССЫЛКИ

В настоящем стандарте использованы ссылки на следующие стандарты

ГОСТ 34 973—91 (ИСО 8824—87) Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1)

ГОСТ 34.974—91 (ИСО 8825—87) Информационная технология. Взаимосвязь открытых систем Описание базовых правил кодирования для абстрактно-синтаксической нотации версии 1 (АСН.1)

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

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

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

3 ОПРЕДЕЛЕНИЯ

В настоящей части ГОСТ Р ИСО/МЭК 10021 использованы определения, приведенные в приложении Е, а также приведенные ниже определения.

Настоящая часть ГОСТ Р ИСО/МЭК 10021, основанная на концепциях, разработанных в ГОСТ 28906, использует следующие приведенные в нем термины:

- абстрактный синтаксис;
- прикладной уровень;

Издание официальное

- протокольный блок данных прикладного уровня (ПБДП);
- протокол прикладного уровня;
- сервисный элемент прикладного уровня (СЭП);
- конкретный синтаксис передачи;
- задача распределенной обработки информации;
- услуга уровня;
- уровень;
- открытая система;
- взаимосвязь открытых систем (ВОС);
- реальная открытая система.

В настоящей части ГОСТ Р ИСО/МЭК 10021 использованы следующие термины, определенные в ГОСТ 34.973:

- абстрактно-синтаксическая нотация один (АСН.1);
- тип (данных);
- значение (данных);
- импорт;
- целое число;
- макрокоманда;
- модуль;
- объектный идентификатор;
- тег.

В настоящей части ГОСТ Р ИСО/МЭК 10021 использован следующий термин, определенный в ГОСТ 34.974:

- базовые правила кодирования.

В настоящей части ГОСТ Р ИСО/МЭК 10021 использован следующий термин, определенный в ГОСТ 34.981:

- прикладной контекст (ПК).

В настоящей части ГОСТ Р ИСО/МЭК 10021 использованы следующие термины, определенные в ГОСТ Р ИСО/МЭК 9072-1:

- операция связки;
- ошибка;
- связанный;
- операция;
- служба удаленных операций (СУО);
- удаленные операции;
- операция развязки.

4 СОКРАЩЕНИЯ

В настоящей части ГОСТ Р ИСО/МЭК 10021 использованы сокращения, приведенные в приложении Е.

5 СОГЛАШЕНИЯ

В настоящей части ГОСТ Р ИСО/МЭК 10021 использованы описательные соглашения, указанные ниже.

5.1 А С Н.1

В настоящей части ГОСТ Р ИСО/МЭК 10021 использованы следующие основанные на АСН.1 описательные соглашения:

а) для определения макрокоманд OBJECT, PORT, REFINE — нотация макрокоманд АСН.1 ГОСТ 34.973;

б) для определения макрокоманд ABSTRACT-BIND, -UNBIND, -OPERATION, -ERROR — нотация макрокоманд BIND, UNBIND, OPERATION и ERROR ГОСТ Р ИСО/МЭК 9072-1;

в) для определения абстрактного синтаксиса информационных объектов в примере приложения А — сама АСН.1;

г) для определения различных абстрактных моделей в примере приложения А — макрокоманды OBJECT, PORT и REFINE раздела 7;

д) для определения различных абстрактных услуг в примере приложения А — макрокоманды ABSTRACT-BIND, -OPERATION и -ERROR раздела 8.

АСН.1 использована в основном тексте настоящей части ГОСТ Р ИСО/МЭК 10021 для большей наглядности и в приложениях (в избыточном виде) — для ссылок. При обнаружении между ними различий указана ошибка спецификации.

Заметим, что теги АСН.1 — неявные во всех модулях АСН.1 и в приложениях; в этом отношении указанные модули являются определительными.

5.2 Термины

В настоящей части ГОСТ Р ИСО/МЭК 10021 приняты следующие шрифтовые выделения: определяемые термины — **полужирный шрифт**; термины, на которые даны ссылки (до их определений), — *курсив*; в остальных случаях — светлый шрифт; термины, которые означают имена собственные, напечатаны с прописных букв, общие термины — со строчных.

Глава вторая. СОГЛАШЕНИЯ ПО ОПРЕДЕЛЕНИЮ АБСТРАКТНЫХ УСЛУГ

6 ОБЩЕЕ ОПИСАНИЕ

При определении и описании сложной задачи распределенной обработки информации лучше начать с определения задачи в абстрактных, а не в конкретных терминах. Такой подход гарантирует, что функциональные требования задачи формируются независимо от ее конкретной реализации. Наряду с другими причинами, такое разделение требований важно и потому, что каждый аспект задачи может допускать различные конкретные реализации. В системе передачи сообщений, которая охватывает трех агентов передачи сообщений, первый и второй, например, могут взаимодействовать, используя связь ВОС, а второй и третий — с помощью собственных средств.

В данной главе определены соглашения об абстрактном описании задачи распределенной обработки информации с использованием как макроподхода, так и микроподхода. Описание по первому способу называется *абстрактной моделью,* по второму — *абстрактными услугами.*

В данной главе определены различные формальные средства спецификации *абстрактных моделей* и *услуг.* В приложении А приведен исчерпывающий пример их использования. При чтении данной главы пользователь может обращаться к этому приложению, например к его иллюстрациям.

Данная глава охватывает следующие вопросы:

а) *абстрактные модели;*

б) *абстрактные услуги.*

Примечание — Упомянутые выше формальные средства не являются ни языком формального описания, ни его заменой. Они являются просто нотацией АСН.1, обеспечивающей неформальные описательные соглашения, определенные в данной главе.

7 АБСТРАКТНЫЕ МОДЕЛИ

Макроподход к задаче распределенной обработки информации называется **абстрактной моделью (моделью)** этой задачи и функциональной среды, в которой она выполняется. Он основан на понятиях *абстрактных объектов, портов, услуг* и *уточнений.* (Понятие *абстрактной услуги* более полно рассмотрено в разделе 8).

7.1 Абстрактные объекты

Абстрактный объект (объект) — это функциональный логический объект, один из возможно нескольких взаимодействующих объектов. Объекты подразделены на несколько типов в зависимости от их функций и поведения. Объект одного типа может представлять, например, систему, а множество объектов другого типа — ее пользователей. Объекты взаимодействуют друг с другом посредством *абстрактных портов.*

Тип объекта определяется с помощью макрокоманды OBJECT. Данная спецификация перечисляет типы *абстрактных портов,* обеспечивающих доступ к такому объекту. Для каждого *асим-*

метричного типа порта спецификация определяет, являются ли порты этого типа портами *потребителя* или портами *поставщика.*

```
OBJECT MACRO    :: =
BEGIN
TYPE NOTATION   :: = «PORTS» « {«PortList» } » | empty
VALUE NOTATION :: = value (VALUE OBJECT IDENTIFIER)

PortList        :: = Port « , » PortList | Port
Port            :: = value (PORT) PortType

PortType        :: = Symmetric | Asymmetric

Symmetric       :: = empty
Asymmetric      :: = Consumer | Supplier

Consumer        :: = «[C]»
Supplier        :: = «[S]»

END
```

Значение данных типа OBJECT — это объектный идентификатор, однозначно и недвусмысленно идентифицирующий определенный тип объекта.

Примечание — Ключевое слово «OBJECT» зарезервировано в АСН.1. Выбор подходящей замены для использования в настоящем контексте — предмет дальнейшего изучения.

7.2 Абстрактные порты

Абстрактный порт (порт) — это пункт, в котором один абстрактный объект взаимодействует с другим абстрактным объектом. Порты бывают различных типов, определяющих виды допустимых взаимодействий. Например, порты одного типа могут представлять средство, с помощью которого осуществляется доступ к системе справочника, порты другого типа — средство, с помощью которого он административно управляется. Сами типы портов бывают двух разновидностей:

а) **симметричные:** все порты симметричного типа идентичны;

б) **асимметричные:** каждый порт асимметричного типа относится к одному из двух видов: **поставщик** и **потребитель.**

Примечание — Конкретный смысл терминов «поставщик» и «потребитель» — часто интуитивный. Можно, естественно, рассматривать файловую систему, например для представления портов поставщика его пользователям и администраторам. Но, строго говоря, значение этих двух терминов произвольное.

Два объекта могут взаимодействовать друг с другом через порт одного и порт другого только тогда, когда эти порты взаимодействуют или **связаны** друг с другом. Действия, посредством которых данное состояние инициируется и завершается в одной или нескольких парах портов, называются **связкой** и **развязкой** соответственно.

Два порта могут быть связаны только в случае, если они **совместимы.** Любые два порта одного и того же симметричного типа совместимы. Два порта одного и того же асимметричного типа совместимы только в том случае, если один из них поставщик, а другой — потребитель.

Тип порта определяется посредством макрокоманды PORT. Такая спецификация идентифицирует *абстрактные операции,* представляющие взаимодействия, которые возможны при связке двух таких портов. Если ни одной из них нет в списке, *абстрактные операции* должны рассматриваться как неспецифицированные.

```
PORT MACRO          :: =
BEGIN
TYPE NOTATION       :: = Operations | empty
VALUE NOTATION      :: = value (VALUE OBJECT IDENTIFIER)

Operations          :: = Symmetrical | Asymmetrical
```

```
Symmetrical          :: = «ABSTRACT» «OPERATIONS» « { «OperationList» } »
Asymmetrical         :: = OneSided | TwoSided

OneSided             :: = Consumer | Supplier
TwoSided             :: = Consumer Supplier | Supplier Consumer

Consumer             :: = «CONSUMER» «INVOKES» « { «OperationList» } »
Supplier             :: = «SUPPLIER»  «INVOKES» « { «OperationList» } »

OperationList        :: = Operation « , » OperationList | Operation
Operation            :: = value (ABSTRACT-OPERATION) |

                        — — идентифицирующее абстрактную операцию
                          по типу значения данных

                        — — идентифицирующее абстрактную операцию
                          по типу данных
END
```

Если тип порта — симметричный, оба объекта предлагают все перечисленные *абстрактные операции*. Если тип порта — асимметричный, макрокоманда определяет различие между *абстрактными операциями*, которые предлагает объект с портом потребителя, и абстрактными операциями, которые предлагает объект с портом поставщика.

Значение данных типа PORT — это объектный идентификатор, который однозначно и недвусмысленно идентифицирует определенный тип порта.

7.3 Абстрактные услуги

Абстрактная услуга — это набор возможностей, предлагаемых одним объектом другому посредством одного или нескольких его портов. Первый объект называется **поставщиком абстрактной услуги (поставщиком)**, другой — **пользователем абстрактной услуги (пользователем)**. Каждый рассматриваемый порт может быть либо симметричным, либо асимметричным и в последнем случае — либо потребителем, либо поставщиком.

Абстрактная услуга может иметь любое число пользователей и поставщиков.

Когда порты абстрактной услуги поставщика связаны с совместимыми портами пользователя, считается, что между этими двумя объектами существует **абстрактная ассоциация** (или **ассоциация**).

Абстрактные услуги определены в разделе 8.

Примечание — Абстрактная услуга выполняет почти ту же задачу в прикладном уровне, которую она выполняет в услугах нижних уровней ВОС.

7.4 Абстрактные уточнения

Объект может выглядеть по-разному в разное время. В некоторых случаях удобно считать объект атомарным. Это удобно, например, при описании способа взаимодействия объекта с другими объектами, внешними по отношению к нему, т.е. при определении его абстрактных услуг. В других случаях может быть удобнее считать объект составным, т. е. построенным из других объектов. Это может быть удобно, например, при описании способа реализации объекта.

Как и любые другие объекты, объекты компонентов имеют порты. Некоторые из них видимы на «поверхности» построенного объекта. Другие позволяют объектам компонентов взаимодействовать, поддерживая, таким образом, обеспечение и использование более мелких абстрактных услуг среди объектов компонентов, взаимодействующих для обеспечения абстрактного сервиса построенного объекта.

Функциональное разложение объекта на несколько меньших объектов называется **абстрактным уточнением (уточнением)** этого объекта.

Метод уточнения может быть применен рекурсивно. Сам объект-компонент может быть уточнен для раскрытия его внутренней структуры. Этот процесс может продолжаться, пока не будут достигнуты объекты-компоненты, которые лучше рассматривать как атомарные.

Уточнение определяется посредством макрокоманды REFINE. Она идентифицирует объект, внутренняя структура которого раскрывается, и объекты-компоненты, используемые при построении. Каждый объект-компонент характеризуется или как уникальный, или как повторяющийся.

Кроме того, макрокоманда определяет, какие порты объектов-компонентов связаны с портами других объектов-компонентов и какие видимы на поверхности составного объекта.

```
REFINE MACRO            :: =
BEGIN

TYPE NOTATION           :: = Object «AS» ComponentList
VALUE NOTATION          :: = value (VALUE OBJECT IDENTIFIER)

ComponentList           :: = Component ComponentList | Component
Component               :: = ObjectSpec Ports

ObjectSpec              :: = Object | Object «RECURRING»

Ports                   :: = PortSpecList | empty
PortSpecList            :: = PortSpec PortSpecList | PortSpec
PortSpec                :: = value (PORT) PortSide PortStatus

PortSide                :: = Consumer | Supplier | empty
Consumer                :: = «[C]»
Supplier                :: = «[S]»

PortStatus              :: = «VISIBLE» | «PAIRED» «WITH» ObjectList

ObjectList              :: = Object « , »  ObjectList | Object
Object                  :: = value (OBJECT)

END
```

Значением данных типа REFINE является объектный идентификатор.

Примечание — Порты, которые сами являются объектами, могут выглядеть по-разному в разное время. В некоторых случаях удобно считать порт (пару портов) атомарным. Однако можно представить внутреннюю структуру самого порта, чтобы оценить, каким образом можно обеспечить взаимосвязь. При таком представлении сама пара портов рассматривается как обеспечиваемая совокупность объектов. Это расширит возможность определения возможностей взаимодействия. Понятие «уточнение порта» не рассматривается в данной версии настоящей части ГОСТ Р ИСО/МЭК 10021.

8 АБСТРАКТНЫЕ УСЛУГИ

Микроподход к задаче распределенной обработки информации — это спецификация абстрактных услуг, определяющая, каким образом задача инициируется, управляется и завершается. Он базируется на понятиях *операций абстрактной связки, операций развязки, операций* и *ошибок*, а также на разрешающем понятии *абстрактных процедур*.

Примечание — Определенные ниже макрокоманды предполагают использование АСН.1 для спецификации аргументов, результатов и параметров. Любые специфичные для контекста теги, например назначенные в ходе спецификаций, хотя не имеют значения в этом контексте, но играют важную роль в реализации СУО абстрактной услуги.

8.1 Абстрактные процедуры

Абстрактная процедура (процедура) — это задача, которую выполняет один объект по запросу другого. Выдача запроса и выполнение задачи называются **привлечением** и **исполнением** процедуры. Объекты, выдающие запрос и выполняющие его, называются соответственно **запросчиком** и **исполнителем.**

Процедура может (но необязательно) потребовать, чтобы запросчик при привлечении обеспечил исполнителю один информационный объект предписанного типа, который называется **аргументом** процедуры.

Каждое выполнение каждой процедуры дает успешный или безуспешный результат. Процедура

считается успешной, если она выполнена полностью, и безуспешной, если она закончилась преждевременно.

Процедура может (но необязательно) требовать, чтобы исполнитель проинформировал запросчика об успешности ее выполнения. Она может (но необязательно) требовать, кроме того, чтобы при уведомлении об успешности исполнитель выдал информационный объект предписанного типа, который называется **результатом** процедуры.

Процедура может (но необязательно) требовать, чтобы исполнитель информировал запросчика о безуспешности ее выполнения. Она может (но необязательно) требовать обеспечения определенной информации при уведомлении о безуспешности выполнения.

Примечание — В последующих разделах АСН.1 предписывается как средство спецификации абстрактного синтаксиса аргументов и результатов процедур (а также параметров *абстрактных ошибок).* Такие использования АСН.1 не предполагают, что эти информационные объекты *обязательно* транспортируются между открытыми системами. В частности, то, что информационные объекты посредством их описания в АСН.1 и базовых правилах кодирования имеют конкретные синтаксисы передачи, несущественно в настоящем контексте. АСН.1 — это просто удобное средство формального описания абстрактного синтаксиса информационных объектов.

8.2 Операции абстрактной связки

Операция абстрактной связки — это процедура, успешное выполнение которой связывает одну или несколько пар абстрактных портов. Объект, который запрашивает абстрактную операцию связки, называется **инициатором;** тот, кто ее исполняет, — **ответчиком.**

Операция абстрактной связки приемлема для связки конкретного набора портов инициатора с совместимым набором портов ответчика. Если в наборе один или несколько портов асимметричны, то операция абстрактной связки может быть приемлема для связывания либо только со стороны потребителя, либо только со стороны поставщика, либо с любой из сторон.

Операция абстрактной связки — это полностью обобщенная процедура, за исключением того, что в случае, если информация переносится запросчику при безуспешном результате, она ограничивается одним информационным объектом, называемым **информацией об ошибке.**

Абстрактная операция связки определяется посредством макрокоманды ABSTRACT-BIND, определение которой представляет собой следующее:

```
ABSTRACT-BIND MACRO  ::=
BEGIN

TYPE NOTATION        ::= Ports Bind
VALUE NOTATION       ::= value (VALUE BindType)

Ports                ::= «TO» « { » PortList « } » | empty
PortList             ::= Port « , » PortList | Port
Port                 ::= value (PORT) PortSide
PortSide             ::= Consumer | Supplier | empty
Consumer             ::= «[C]»
Supplier             ::= «[S]»
Bind                 ::= type (BindType) - - должен быть тип BIND
                         | empty <BindType ::= BIND>

END
```

В разделе «Порты», где введено ключевое слово «TO», перечисляются порты ответчика, которые будет связывать эта операция абстрактной связки. Если там перечисляется асимметричный порт без классификации «[Пс]» или «[П]», это означает, что операция абстрактной связки приемлема для использования в связке такого порта в любом направлении.

Заметим, что спецификация аргумента, результата и (или) информации об ошибке выполняется посредством (вложенной) макрокоманды BIND удаленных операций, определенных в ГОСТ Р ИСО/МЭК 9072-1, и это значение такого типа, который выдает макрокоманда. Если информация отсутствует, выдается «BIND» по умолчанию.

Примечание — Взаимосвязь ABSTRACT и BIND помогает выполнять тривиальную реализацию СУО абстрактной услуги (см. 10.1).

Абстрактная услуга обычно содержит операцию абстрактной связки для каждого типа порта, участвующего в его обеспечении. При привлечении нескольких типов порта их операции абстрактной связки могут, но не должны быть различимыми.

8.3 Операции абстрактной развязки

Операции абстрактной развязки — это процедура, выполнение которой (успешное или нет) развязывает два порта. Она вызывается объектом, который вызывал соответствующую операцию абстрактной связи (т. е. инициатором), и выполняется ответчиком.

Операция абстрактной развязки приемлема для развязывания определенного набора портов инициатора из совместимого набора ответчика. Если один или несколько портов в наборе асимметричны, операция абстрактной развязки может быть приемлемой для развязывания только со стороны потребителя, только со стороны поставщика либо с обеих сторон.

Операция абстрактной развязки — полностью обобщенная процедура, за исключением того, что если информация переносится запросчику, при безуспешном результате она ограничивается одним информационным объектом, называемым **информацией об ошибке.**

Операция абстрактной развязки определяется посредством макрокоманды ABSTRACT-UNBIND, определение которой представляет собой следующее:

```
ABSTRACT-UNBIND MACRO  :: =
BEGIN

TYPE NOTATION          :: = Ports Unbind
VALUE NOTATION         :: = value (VALUE UnbindType)

Ports                  :: = «FROM» « { » PortList « } » | empty
PortList               :: = Port « , » PortList | Port
Port                   :: = value (PORT) PortSide
PortSide               :: = Consumer | Supplier | empty
Consumer               :: = «[C]»
Supplier               :: = «[S]»
Unbind                 :: = type (UnbindType) |
                            - - должен быть тип UNBIND
                            empty  <UnbindType :: = UNBIND>

END
```

В разделе «Порты», где введено ключевое слово «FROM», перечислены порты ответчика, от которых эта операция абстрактной развязки будет осуществлять развязку. Если там упоминается асимметричный порт без классификации «[S]» (Пс) или «[C]» (П), это означает, что операция абстрактной развязки приемлема для использования в развязывании такого порта в любом направлении (хотя фактическое направление определяется направлением, в котором осуществляется связка).

Заметим, что спецификация аргумента, результата и (или) информации об ошибке выполняется посредством (вложенной) макрокоманды UNBIND удаленных операций, определенных в ИСО/МЭК 9072-1, и это значение такого типа, которое выдает макрокоманда. Если никакая информация не обеспечивается, выдается «UNBIND» по умолчанию.

Примечание — Взаимосвязь ABSTRACT-UNBIND и UNBIND помогает выполнять тривиальную реализацию СУО абстрактной услуги (см. 10.1).

Абстрактная услуга обычно содержит абстрактную операцию развязки (для) каждого типа порта, участвующего в ее обеспечении. Если участвуют порты нескольких типов, операции абстрактной развязки могут, но необязательно, быть различимыми.

8.4 Абстрактные операции

Абстрактная операция — это процедура, которая может быть привлечена в контексте двух связанных портов. Безуспешность ее выполнения не влияет на связку. Если порты асимметричные, порт предписывает роль запросчика: объект, имеющий порт потребителя, объект, имеющий порт

поставщика, либо то и другое. Если порты симметричные, запросчик может быть любым объектом. Независимо от того, какими являются порты: симметричными или асимметричными, — оставшийся объект является исполнителем.

Абстрактная операция — это полностью обобщенная процедура, за исключением информации, переносимой запросчику при безуспешном результате. Абстрактная операция заканчивается безуспешно, если она сталкивается с *абстрактной ошибкой,* и переносимая информация сводится к информации, необходимой для уведомления об этой абстрактной ошибке. Для каждой абстрактной операции предписываются требование уведомления о безуспешном результате и, в случае необходимости, виды тех абстрактных ошибок, которые могут иметь место.

Абстрактная операция определяется посредством макрокоманды ABSTRACT-OPERATION. Ее определение идентично определению макрокоманды OPERATION удаленных операций, приведенному в ГОСТ Р ИСО/МЭК 9072-1.

ABSTRACT-OPERATION MACRO :: = OPERATION

Абстрактная услуга содержит от нуля до нескольких абстрактных операций для каждого типа порта, участвующего в ее обеспечении. При участии нескольких типов порта они могут, но необязательно, иметь общие абстрактные операции.

Примечание — Эквивалентность ABSTRACT-OPERATION и OPERATION помогает выполнять реализацию СУО абстрактной услуги (см. 10.1).

8.5 Абстрактные ошибки

Абстрактная ошибка — это особый случай, который может возникнуть при выполнении абстрактной операции, приводя к безуспешности ее выполнения.

При уведомлении об абстрактной ошибке исполнитель передает запросчику идентичность абстрактной ошибки и, возможно, один информационный объект, называемый ее **параметром.** Для каждой абстрактной ошибки предписываются необходимость выдачи параметра и, при необходимости, его тип.

Абстрактная ошибка определяется посредством макрокоманды ABSTRACT-ERROR. Ее определение идентично определению макрокоманды ERROR удаленных операций, установленному в ГОСТ Р ИСО/МЭК 9072-1.

ABSTRACT-ERROR MACRO :: = ERROR

Абстрактная услуга содержит от нуля до нескольких более абстрактных ошибок, о которых сообщается в абстрактных операциях.

Примечание — Эквивалентность ABSTRACT-ERROR и ERROR помогает выполнять тривиальную реализацию СУО абстрактной услуги (см. 10.1).

Глава третья. РЕАЛИЗАЦИЯ АБСТРАКТНЫХ УСЛУГ

9 ОБЩЕЕ ОПИСАНИЕ

Поскольку задача распределенной обработки информации охарактеризована и определена в абстрактных терминах, должен быть предписан способ конкретной реализации каждого аспекта задачи. Как отмечено выше, каждый аспект может допускать несколько конкретных реализаций.

В данной главе устанавливаются принципы конкретной реализации абстрактных моделей и услуг. **Реальный** *x* —это вычислительный процесс или система, или реальная открытая система, конкретно реализующая абстрактный объект типа *x*.

В данной главе рассматриваются следующие вопросы:

а) реализация ВОС;

б) собственные реализации.

Примечание — Рассматриваемые здесь аспекты абстрактной модели являются абстрактными портами и их связками. Это обусловлено тем, что абстрактные порты отмечают границу не только между абстрактными объектами, но и между физическими системами, которые представляют собой конкретную

реализацию этих абстрактных объектов. Таким образом, абстрактные порты и связки — это части абстрактной модели, которые должны строиться или которые можно построить с помощью средств ВОС, если предполагается взаимодействие открытых систем.

10 РЕАЛИЗАЦИИ ВОС

Основная цель рекомендаций МККТТ и стандартов ИСО — определить способ реализации задач распределенной обработки информации, выполняемых несколькими взаимодействующими реальными открытыми системами.

В функциональной среде ВОС объекты реализуются посредством прикладных процессов, характеризующихся в общем случае неоднозначным отображением объектов на прикладные процессы. Связь между объектами, реализуемая прикладными процессами в различных открытых системах, выполняется с помощью прикладных протоколов ВОС (состоящих из прикладных контекстов). Таким образом, прикладной контекст реализует связку, использование и развязку многих пар портов.

Спецификация прикладного контекста осуществляется в понятиях скоординированной работы многих сервисных элементов прикладного уровня. Поэтому реализация будет чрезвычайно прямолинейной, чтобы установить, определяется ли сервисный-элемент-прикладного-уровня, соответствующий каждому порту, с которым предполагается обеспечить связь.

Реализация абстрактных портов и связок посредством СЭП и прикладных контекстов (ПК) рассматривается ниже. Рассматриваются реализации и СУО, и не-СУО.

10.1 Реализации СУО

Конкретная реализация портов и связок часто тривиальна при выполнении посредством удаленных операций.

Это действительно так, потому что она прямолинейна для определения такой абстрактной услуги, в которой существует основанный на СУО прикладной протокол, функционально ей идентичный. Это правильно, в свою очередь, потому, что основа спецификации абстрактных услуг изоморфна для спецификации прикладных протоколов, основанных на СУО. В таблице 1 перечислены соответствия вне изоморфизма.

Таблица 1 — Соответствия абстрактных услуг и основанных на СУО протоколов

Аспект абстрактной услуги	Аспект протокола, основанного на СУО
Операция абстрактной связки	Операция связки
Операция абстрактной развязки	Операция развязки
Абстрактная операция	Операция
Абстрактная ошибка	Ошибка

Соответствия, приведенные в таблице 1, обусловлены тем, что соответствующие аспекты формально определены с использованием тесно связанных или эквивалентных макрокоманд, как показано в таблице 2.

Таблица 2 — Эквивалентные абстрактные услуги и макрокоманды СУО

Макрокоманды абстрактной услуги	Макрокоманды СУО
ABSTRACT-BIND	BIND
ABSTRACT-UNBIND	UNBIND
ABSTRACT-OPERATION	OPERATION
ABSTRACT-ERROR	ERROR

Определение СЭП, основанных на СУО, и ПК, которые конкретно реализуют абстрактные порты, поясняется в приложении А на примере.

Для тривиальности реализации необходимо наличие операции абстрактной связки, связывающей все порты, которые должны взаимодействовать попарно.

Примечание — Когда в абстрактной услуге участвуют несколько портов (пар портов), это требует, чтобы операция абстрактной связки предназначалась для конкретных участвующих портов. Не предусматривается (в настоящее время) автоматический синтез приемлемой абстрактной связки, основанный, например, на определениях операций абстрактной связки для отдельных портов.

10.2 Реализации не-СУО

Конкретная реализация портов и связок — это более существенная задача, когда делается попытка использования иных средств, кроме удаленных операций; немногое можно сказать и об общей проблеме.

Несмотря на сказанное, уместны следующие два соображения:

а) конкретная реализация абстрактной услуги в качестве прикладного протокола очень упрощена при использовании АСН.1 для определения ее блоков ПБДП. Это объясняется тем, что спецификация протокола может просто импортировать соответствующие типы и значения из спецификации абстрактной услуги;

б) конкретная реализация абстрактной услуги, абстрактные операции которой не уведомляют о своих результатах, концептуально проста. Это объясняется тем, что каждая такая абстрактная операция представляет взаимодействие, содержащее один ПБДП. Из этого простейшего из всех возможных взаимодействий можно построить произвольно сложные.

11 СОБСТВЕННЫЕ РЕАЛИЗАЦИИ

Вторичная цель рекомендаций МККТТ и стандартов ИСО — добиться того, чтобы указанные части задачи распределенной обработки информации, выполняемые собственными средствами, решались таким образом, чтобы обеспечивалась заданная общая функциональность системы.

Реализация абстрактных портов и связок собственными средствами кратко рассматривается ниже. Учитываются и распределенные, и нераспределенные реализации.

11.1 Распределенные реализации

Конкретная реализация портов и связок посредством протоколов обмена собственных вычислительных машин — вопрос локального характера. Спецификация наглядной функциональности, воплощенная в абстрактной услуге, обеспечивает руководство для изготовителей собственных реализаций с тем, чтобы в приемлемых случаях такие реализации могли бы играть соответствующую роль в решении общей задачи.

11.2 Нераспределенные реализации

Конкретная реализация портов и связок посредством механизмов полностью одной вычислительной машины — вопрос локального характера. Как и в случае, рассмотренном в 11.1, спецификация абстрактной услуги служит руководством для изготовителей в обеспечении того, чтобы собственная реализация могла, тем не менее, выполнять соответствующую роль в решении общей задачи.

ПРИЛОЖЕНИЕ А
(информационное)

ПРИМЕР ИСПОЛЬЗОВАНИЯ НОТАЦИИ АБСТРАКТНЫХ УСЛУГ

В данном приложении на примере показано использование нотации абстрактной модели и услуг. В примере показаны две системы: «желтая» и «зеленая» — и их функциональные среды: «желтая» и «зеленая».

Нотация абстрактной модели используется для отдельного описания сред (разделы А.2 и А.4) и для демонстрации связи их систем: одна строится из другой (раздел А.6). Нотация абстрактной услуги используется для описания возможностей каждой системы (разделы А.3 и А.5). Пример заканчивается реализацией совокупных портов в виде ПК и СЭП, использующих нотацию СУО ГОСТ Р ИСО/МЭК 9072-1, что может соответствовать обмену данными ВОС (разделы А.7 и А.8).

А.1 Назначение объектных идентификаторов

Для определяемых в данном приложении модулей АСН.1 требуется присвоение различных объектных идентификаторов. Все они определяются ниже с использованием АСН.1. Эти присвоения — определяющие, за исключением присвоений для модулей АСН.1 и самого предмета соглашений по определению прикладных услуг. Определяющие присвоения для модулей АСН.1 имеются в самих модулях, другие ссылки на них используются в разделах IMPORT. Для предмета соглашений по определению прикладных услуг присвоения фиксированные.

```
ExampleObjectIdentifiers {joint-iso-ccitt mhs-motis(6) asdc(2)
    example(1) modules(0) object-identifiers(0)}
DEFINITIONS IMPLICIT TAGS :: =
BEGIN
- - Пролог
- - Экспортирует все
IMPORTS - - ничего - - ;
ID :: = OBJECT IDENTIFIER
- - Пример соглашений по определению абстрактной услуги (неопределяющий)
id-asdc-ex ID :: = { joint-iso-ccitt mhs-motis(6) asdc(2) example(1) }
                    - - неопределяющий
- - Категории
id-mod    ID :: = {id-asdc-ex 0} - - модули; неопределяющие
id-ot     ID :: = {id-asdc-ex 1} - - типы объектов
id-pt     ID :: = {id-asdc-ex 2} - - типы портов
id-ref    ID :: = {id-asdc-ex 3} - - уточнения
id-ac     ID :: = {id-asdc-ex 4} - - прикладные контексты
id-ase    ID :: = {id-asdc-ex 5} - - сервисные элементы прикладного уровня
id-as     ID :: = {id-asdc-ex 6} - - абстрактные синтаксисы

- - Модули
id-mod-object-identifiers   ID ::= {id-mod 0} - - неопределяющий
id-mod-ye-refinement        ID ::= {id-mod 1} - - неопределяющий
id-mod-y-abstract-service   ID ::= {id-mod 2} - - неопределяющий
id-mod-ge-refinement        ID ::= {id-mod 3} - - неопределяющий
id-mod-g-abstract service   ID ::= {id-mod 4} - - неопределяющий
id-mod-ys-refinement        ID ::= {id-mod 5} - - неопределяющий
id-mod-ys-realization       ID ::= {id-mod 6} - - неопределяющий
id-mod-gs-realization       ID ::= {id-mod 7} - - неопределяющий

- - Типы объектов
id-ot-y-environment         ID ::= {id-ot 0}
id-ot-y-user                ID ::= {id-ot 1}
id-ot-y-system              ID ::= {id-ot 2}
id-ot-g-environment         ID ::= {id-ot 3}
id-ot-g-user                ID ::= {id-ot 4}
id-ot-g-manager             ID ::= {id-ot 5}
id-ot-g-system              ID ::= {id-ot 6}
id-ot-agent                 ID ::= {id-ot 7}
```

```
- - Типы портов
id-pt-y-use                ID ::= {id-pt 0}
id-pt-g-use                ID ::= {id-pt 1}
id-pt-g-management         ID ::= {id-pt 2}

- - Уточнения
id-ref-y-environment       ID ::= {id-ref 0}
id-ref-g-environment       ID ::= {id-ref 1}
id-ref-y-system            ID ::= {id-ref 2}

- - Прикладные контексты
id-ac-y-use                ID ::= {id-ac 0}
id-ac-g-use                ID ::= {id-ac 1}
id-ac-g-management         ID ::= {id-ac 2}

- - Сервисные элементы прикладного уровня
id-ase-y-use               ID ::= {id-ase 0}
id-ase-g-use               ID ::= {id-ase 1}
id-ase-g-management        ID ::= {id-ase 2}

- - Абстрактные синтаксисы
id-as-y-use                ID ::= {id-as 0}
id-as-g-use                ID ::= {id-as 1}
id-as-g-management         ID ::= {id-as 2}

END - - Конец примера объектных идентификаторов
```

А.2 Уточнение желтой среды

Желтая среда, изображенная на рисунке А.1, формально уточняется ниже с использованием макрокоманд OBJECT и REFINE.

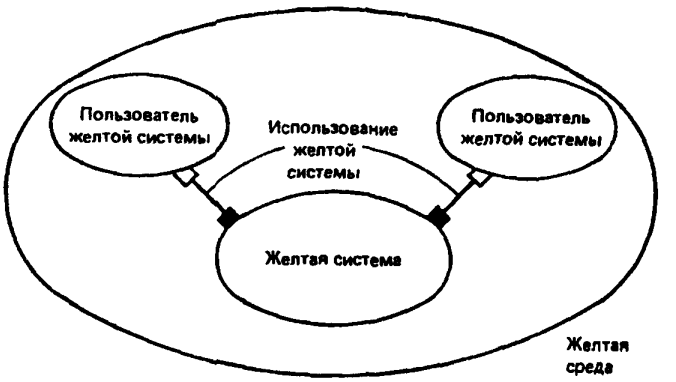


Рисунок А.1 — Желтая среда

Как показано на рисунке А.1 и подтверждено ниже спецификацией ACH.1, желтая среда может моделироваться как объект, который может быть разложен на один центральный объект — **желтую систему** и любое количество других периферийных объектов — **пользователей желтой системы.** Желтая система взаимодействует со своими пользователями посредством ее портов **использования желтой системы.**

- - - - - - - -

```
YellowEnvironmentRefinement { joint-iso-ccitt mhs-motis(6) asdc(2)
    example(1) modules(0) ye-refinement(1)}
DEFINITIONS IMPLICIT TAGS :: =
BEGIN
- - Пролог
EXPORTS
    yellow-environment, yellow-environment-refinement,
```

```
        yellow-system, yellow-user;
IMPORTS
        - - Абстрактная услуга желтой среды
            yellow-use
            - - - -
            FROM YellowAbstractService { joint-iso-ccitt
                mhs-motis(6) asdc(2) example(1) modules(0)
                y-abstract-service(2) }
        - - Пример объектных идентификаторов
            id-ot-y-environment, id-ot-y-system, id-ot-y-user,
                id-ref-y-environment
            - - - -
            FROM ExampleObjectIdentifiers { joint-iso-ccitt
                mhs-motis(6) asdc(2) example(1) modules(0)
                object-identifiers(0)}
        - - Нотация абстрактной услуги
            OBJECT, REFINE
            - - - -
            FROM AbstractServiceNotation { joint-iso-ccitt
                mhs-motis(6) asdc(2) modules(0) notation(1)} ;

- - Желтая среда
yellow-environment OBJECT
        :: = id-ot-y-environment
- - Уточнение желтой среды
yellow-environment-refinement REFINE yellow-environment AS
        yellow-user-RECURRING
        yellow-system
                yellow-use [S] PAIRED WITH yellow-user
        :: = id-ref-y-environment

- - Типы объектов компонентов
yellow-user OBJECT
        PORTS {
                yellow-use [C]}
        :: = id-ot-y-user

yellow-system OBJECT
        PORTS {
                yellow-use [S]}
        :: = id-ot-y-system

END - - Конец уточнения желтой среды
```

А.3 Определение абстрактной услуги желтой системы

Ниже приведено формальное определение абстрактной услуги, обеспечиваемой желтой системой ее пользователям, с использованием макрокоманд PORT и ABSTRACT-BIND, -OPERATION, -ERROR.

Как указывает спецификация АСН.1, абстрактная услуга, обеспечиваемая желтой системой, включает в себя порты одного типа: использование-желтой-среды. Каждый порт охватывает некоторое число абстрактных операций, которые в совокупности уведомляют об абстрактных ошибках. Желтая система защищает свои порты посредством операции абстрактной связки — **YellowBind**, требующей, чтобы пользователи удобным образом идентифицировали себя перед последующим взаимодействием. Операция абстрактной развязки — **YellowUnbind**, составляющая этап завершения, необходима для завершения взаимодействия.

```
                                        - - - - - - - - - -
YellowAbstractService { joint-iso-ccitt mhs-motis(6) asdc(2) example(1)
        modules(0) y-abstract-service(2) }
DEFINITIONS IMPLICIT TAGS :: =
BEGIN

- - Пролог
EXPORTS
        AuthenticateUser, Yellow-operation-1, . . . yellow-use;
```

```
IMPORTS
      - - Пример объектных идентификаторов
            id-pt-y-use
            - - - -
            FROM ExampleObjectIdentifiers { joint-iso-ccitt
                  mhs-motis(6) asdc(2) example(1) modules(0)
                  object-identifiers(0)}
      - - Нотация абстрактных услуг
            ABSTRACT-BIND, ABSTRACT-ERROR, ABSTRACT-OPERATION, PORT
                  - - - -
            FROM AbstractServiceNotation { joint-iso-ccitt
                  mhs-motis(6) asdc(2) modules(0) notation(1) };

- - Тип порта

yellow-use PORT
        CONSUMER INVOKES {
                  Yellow-operation-1, . . . }
        :: = id-pt-y-use

- - Операция абстрактной связки
Credentials :: = SET {
        name      [0] IA5String,
        password [1] IA5String}

YellowBind :: = ABSTRACT-BIND
          TO {yellow-use[S]}
       BIND
          ARGUMENT credentials Credentials
             BIND-ERROR ENUMERATED {
                name-or-password-invalid(0) }

- - Операция абстрактной развязки
YellowUnbind :: = ABSTRACT-UNBIND
    FROM {yellow-use[S] }
- - Абстрактные операции
Yellow-operation-1 :: = ABSTRACT-OPERATION
        ARGUMENT . . .
        RESULT . . .
        ERRORS {
                yellow-error-1, . . . }
. . .

- - Абстрактные ошибки
yellow-error-1 ABSTRACT-ERROR
        PARAMETER . . .
        :: = 1
. . .

END - - Конец абстрактной услуги желтой среды
```

А.4 Уточнение зеленой среды

Зеленая среда, изображенная на рисунке А.2, формально уточняется с использованием макрокоманд OBJECT и REFINE.

Как показывает рисунок А.2 и подтверждает спецификация ACH.1, зеленая среда может моделироваться в виде объекта, который можно разбить на один центральный объект — **зеленую систему**, любое количество других периферийных объектов — **пользователей зеленой системы** и любое количество дополнительных объектов — **диспетчеров зеленой системы**. Зеленая система взаимодействует с пользователями и диспетчерами зеленой системы посредством ее портов **использования зеленой системы** и с (одними) диспетчерами зеленой системы посредством ее портов **управления зеленой системой**.

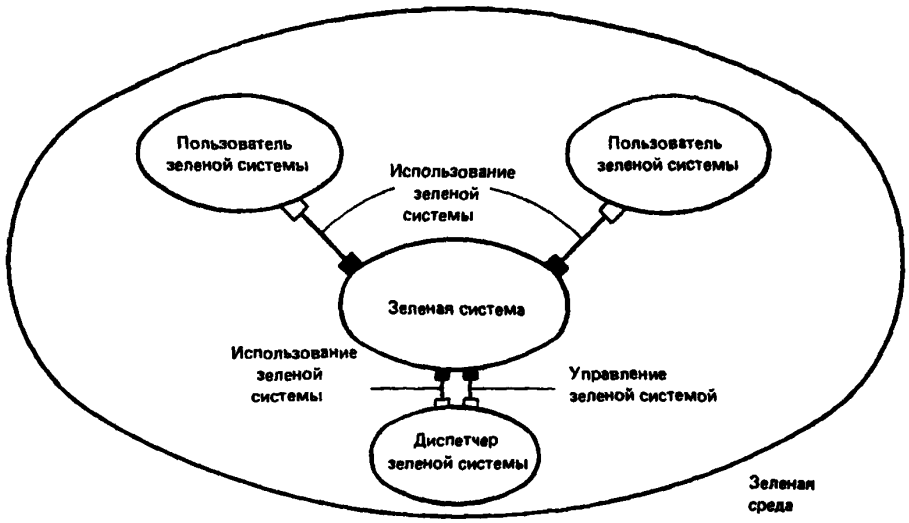


Рисунок А.2 — Зеленая среда

- - - - - - - - -

```
GreenEnvironmentRefinement { joint-iso-ccitt mhs-motis(6) asdc(2)
      example(1) modules(0) ge-refinement(3) }
DEFINITIONS IMPLICIT TAGS :: =
BEGIN

- - Пролог
EXPORTS
      green-environment, green-environment-refinement,
      green-manager, green-system, green-user;
IMPORTS
      - - Абстрактная услуга зеленой среды
            green-use, green-management
            - - - - -
            FROM GreenAbstractService { joint-iso-ccitt
                  mhs-motis(6) asdc(2) example(1) modules(0)
                  g-abstract-service(4)}
      - - Пример объектных идентификаторов
            id-ot-g-environment, id-ot-g-manager, id-ot-g-system,
               id-ot-g-user, id-ref-g-environment
            - - - -
            FROM ExampleObjectIdentifiers { joint-iso-ccitt
                  mhs-motis(6) asdc(2) example(1) modules(0)
                  object-identifiers(0) }
      - - Нотация абстрактных услуг
            OBJECT, REFINE
            - - - -
            FROM AbstractServiceNotation { joint-iso-ccitt
                  mhs-motis(6) asdc(2) modules(0) notation(1) };

- - Зеленая среда
green-environment OBJECT
      :: = id-ot-g-environment

- - Уточнение зеленой среды
green-environment-refinement REFINE green-environment AS
      green-user      RECURRING
      green-manager RECURRING
      green-system
```

```
            green-use           [S] PAIRED WITH green-user, green-manager
            green-management [S] PAIRED WITH green-manager
    :: = id-ref-g-environment

- - Типы объектов компонентов
green-user OBJECT
    PORTS {
        green-use {C} }
    :: = id-ot-g-user
green-manager OBJECT
    PORTS {
        green-use           [C],
        green-management  [C] }
    :: = id-ot-g-manager
green-system OBJECT
    PORTS {
        green-use           [S],
        green-management  [S] }
    :: = id-ot-g-system

END - - Конец уточнения зеленой среды
```

А.5 Определение абстрактной услуги зеленой системы

Ниже приведено формальное определение абстрактной услуги, обеспечиваемой зеленой системой для ее пользователей и администраторов, с использованием макрокоманд PORT и ABSTRACT-BIND, -OPERATION, -ERROR.

Как указывает спецификация АСН.1, абстрактная услуга, обеспечиваемая зеленой системой, включает в себя порты двух видов: использование-зеленой-системы и управление-зеленой-системой. Порт любого из этих видов охватывает некоторое число абстрактных операций, которые в совокупности уведомляют об абстрактных ошибках. Зеленая система защищает свои порты посредством операций абстрактной связки **AuthenticateUser** и **AuthenticateManager**, требующей, чтобы пользователи удобным образом идентифицировали себя и диспетчеров перед последующим взаимодействием. Операции абстрактной развязки, указывающие на то, что для завершения взаимодействия не требуется никакого этапа завершения, не определяются.

- - - - - - - -

```
GreenAbstractServise { joint-iso-ccitt mhs-motis(6) asdc(2)
    example(1) modules(0) g-abstract-service(4) }
DEFINITIONS IMPLICIT TAGS :: =
BEGIN

- - Пролог

EXPORTS
    AuthenticateManager, AuthenticateUser, green-management,
    Green-management-operation-1, . . . green-use,
    green-use-operation-1, . . . ;

IMPORTS
    - - Пример объектных идентификаторов
        id-pt-g-use, id-pt-g-management
        - - - -
    FROM ExampleObjectIdentifiers { joint-iso-ccitt
            mhs-motis(6) asdc(2) example(1) modules(0)
            object-identifiers(0)}
    - - Нотация абстрактных услуг
      PORT, ABSTRACT-BIND, ABSTRACT-OPERATION, ABSTRACT-ERROR
        - - - -
      FROM AbstractServiceNotation { joint-iso-ccitt
            mhs-motis(6) asdc(2) modules(0) notation(1) };
    - - Типы портов

    green-use PORT
```

```
        CONSUMER INVOKES {
                Green-use-operation-1, ... }
        ::= id-pt-g-use

green-management PORT
        CONSUMER INVOKES {
                Green-management-operation-1, ... }
        ::= id-pt-g-management
- - Операции абстрактной связки
Credentials ::= SET {
        name     [0] IA5String,
        password [1] IA5String}

AuthenticateUser ::= ABSTRACT-BIND
        ARGUMENT credentials Credentials
        BIND-ERROR ENUMERATED {
                name-or-password-invalid(0) }

AuthenticateManager ::= ABSTRACT-BIND
        ARGUMENT credentials Credentials
        BIND-ERROR ENUMERATED {
                name-or-password-invalid (0),
                not-a-manager(1)}

- - Абстрактные операции
Green-user-operation-1 ::= ABSTRACT-OPERATION
        ARGUMENT ...
        RESULT ...
        ERRORS {
                green-error-1, ...}
. . .

Green-management-operation-1 ::= ABSTRACT-OPERATION
        ARGUMENT ...
        RESULT ...
        ERRORS {
                green-error-1, ...}
. . .

- - Абстрактные ошибки
        green-error-1 ABSTRACT-ERROR
        PARAMETER ...
        :: = 1

. . .

END - - Конец абстрактной услуги зеленой среды
```

А.6 Уточнение желтой системы

Желтая система, изображенная на рисунке А.3, формально уточняется при использовании макрокоманд OBJECT и REFINE.

Как показано на рисунке А.3 и подтверждено спецификацией АСН.1, желтая система, при ближайшем рассмотрении, имеет компоненты. В частности, желтая система включает в себя зеленую систему, диспетчера зеленой системы, дополненного разнообразными объектами, и *агентов*. Агент является промежуточным звеном между зеленой системой и пользователем желтой системы. Он как бы прибавляет значимость зеленой системе. В любом случае это поставщик порта использования-желтой-системы и потребитель порта использования-зеленой-системы.

- - - - - - - -

```
YellowSystemRefinement { joint-iso-ccitt mhs-motis(6) asdc(2)
        example(1) modules(0) ys-refinement(5) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

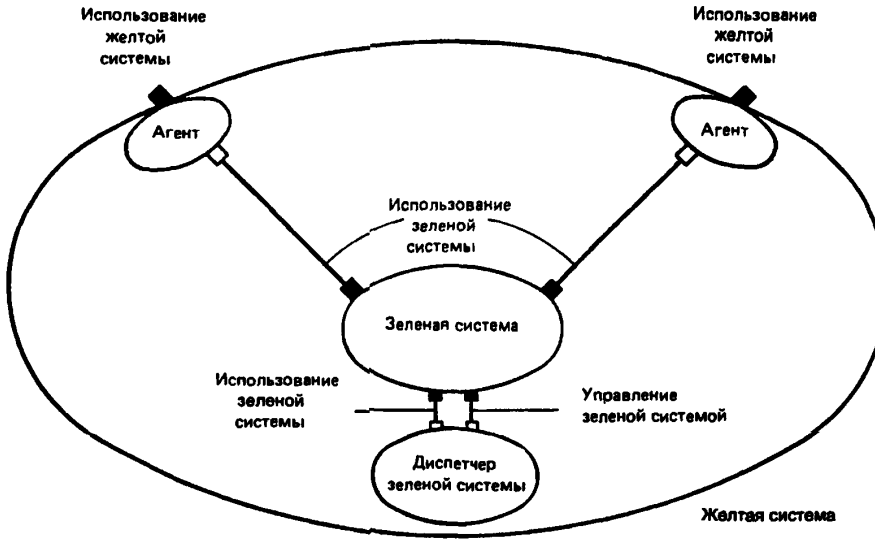


Рисунок А.3 — Желтая система

```
- - Пролог

EXPORTS
      agent, yellow-system-refinement;

IMPORTS
      - - Уточнение желтой среды
            yellow-system, yellow-use
            - - - -
            FROM YellowEnvironmentRefinement { joint-iso-ccitt
                  mhs-motis(6) asdc(2) example(1) modules(0)
                  ye-refinement(1) }
      - - Уточнение зеленой среды
         green-management, green-manager, green-system, green-use
            - - - -
            FROM GreenEnvironmentRefinement { joint-iso-ccitt
                  mhs-motis(6) asdc(2) example(1) modules(0)
                  ge-refinement(3) }
      - - Пример объектных идентификаторов
            id-ot-agent, id-ref-y-system
            - - - -
            FROM ExampleObjectIdentifiers {joint-iso-ccitt
                  mhs-motis(6) asdc(2) example(1) modules(0)
                  object-identifiers(0)}
      - - Нотация абстрактной услуги
            OBJECT, REFINE
            - - - -
            FROM AbstractServiceNotation { joint-iso-ccitt
                  mhs-motis(6) asdc(2) modules(0) notation(1) };

- - Уточнение желтой среды

yellow-system-refinement REFINE yellow-system AS
      agent RECURRING
            yellow-use [S] VISIBLE
      green-manager RECURRING
      green-system
```

```
            green-use [S] PAIRED WITH agent, green-manager
            green-management [S] PAIRED WITH green-manager
        ::= id-ref-y-system

- - Тип объекта компонента
agent OBJECT
        PORTS {
            yellow-use [S],
            green-use [C]}
        ::= id-ot-agent

END - - Конец уточнения желтой среды
```

А.7 Реализация желтой системы

Ниже приведена формальная реализация абстрактной услуги желтой системы посредством СУО, с использованием макрокоманд APPLICATION-CONTEXT и APPLICATION-SERVICE-ELEMENT ГОСТ Р ИСО/МЭК 9072-1.

Как указывает спецификация АСН.1, абстрактная услуга, обеспечиваемая желтой системой, реализуется как единичный **СЭП-использования-желтой-системы** и соответствующий **ПК-использования-желтой-системы** Каждая операция абстрактной связки, абстрактная операция или абстрактная ошибка в абстрактной услуге имеет соответствующую и эквивалентную операцию связки, операцию или ошибку, соответственно, в ее реализации, основанной на СУО.

Заметим, что операциям присваиваются целочисленные значения; соответствующие абстрактные операции не требуют и не получили таких значений.

\- - - - - - -

```
YellowSystemRealization { joint-iso-ccitt mhs-motis(6) asdc(2)
        example(1) modules(0) ys-realization(6) }
DEFINITIONS IMPLICIT TAGS : : =
BEGIN

- - Пролог

EXPORTS
        yellow-use-AC, yellow-use-ASE;

IMPORTS
        - - Абстрактная услуга желтой среды
        Yellow-operation-1, ... yellow-use, YellowBind, YellowUnbind
            - - - -
            FROM YellowAbstractService { joint-iso-ccitt
                    mhs-motis(6) asdc(2) example(1) modules(0)
                    y-abstract-service(2) }
        - - Пример объектных идентификаторов
            id-ac-y-use, id-as-y-use, id-ase-y-use
            - - - -
            FROM ExampleObjectIdentifiers { joint-iso-ccitt
                    mhs-motis(6) asdc(2) example(1) modules(0)
                    object-identifiers(0) }
        - - ПБДП удаленных операций
            rOSE
            - - - -
            FROM Remote-Operations-APDUs {joint-iso-ccitt
                    remote-operations(4) apdus(1)}
        - - Управление ассоциацией
            aCSE
            - - - -
            FROM Remote-Operations-Notation-extension {joint-iso-ccitt
                    remote-operations(4) notation-extension(2)}
        - - Расширение нотации удаленных операций
            APPLICATION-CONTEXT, APPLICATION-SERVICE-ELEMENT
```

```
        - - - -
        FROM Remote-Operations-Notation-extension { joint-iso-ccitt
            remote-operations(4) notation-extension(2) }
aCSE-AS OBJECT IDENTIFIER    = { joint-iso-ccitt association-
control(2) abstractSyntax(1) apdus(0) version1(1) }

    - - Прикладной контекст

yellow-use-AC APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS {aCSE}
        BIND YellowBind
        UNBIND YellowUnbind
        REMOTE OPERATIONS (rOSE)
        INITIATOR CONSUMER OF {yellow-use-ASE}
        ABSTRACT SYNTAXES {yellow-use-AS, aCSE-AS}
          = id-ac-y-use

    - - Сервисный элемент прикладного уровня

yellow-use-ASE APPLICATION-SERVICE-ELEMENT
    CONSUMER INVOKES {
    yellow-operation-1,    }
      = id-ase-y-use

yellow-operation-1 Yellow-operation-1    = 1


- - Абстрактный синтаксис

yellow-use-AS OBJECT IDENTIFIER   = id-as-y-use

END - - Конец реализации желтой среды
```

А.8 Реализация зеленой системы

Ниже приведена формальная реализация абстрактной услуги зеленой системы посредством СУО, с использованием макрокоманд APPLICATION-CONTEXT и APPLICATION-SERVICE-ELEMENT ГОСТ Р ИСО/МЭК 9072-1

Как определено в спецификации АСН 1, абстрактная услуга, обеспечиваемая зеленой системой, реализуется в виде двух СЭП (**СЭП-использования-зеленой-системы** и **СЭП-диспетчера-зеленой-системы**) и двух соответствующих ПК (**ПК-использования-зеленой-системы** и **ПК-диспетчера-зеленой-системы**). Каждая операция абстрактной связки, абстрактная операция или абстрактная ошибка в абстрактной услуге имеет соответствующую и эквивалентную операцию связки, операцию или ошибку, соответственно, в ее реализации, основанной на СУО

Заметим, что операциям присваиваются целочисленные значения; соответствующие абстрактные операции не требуют и не получили таких значений

- - - - - - - - -

```
GreenSystemRealization { joint-iso-ccitt mhs-motis(6) asdc(2)
    example(1) modules(0) gs-realization(7) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

- - Пролог

EXPORTS
    green-management-AC, green-management-ASE, green-use-AC,
    green-use-ASE,

IMPORTS
    - - Абстрактная услуга зеленой системы
    AuthenticateManager, AuthenticateUser, green-management,
    Green-management-operation-1, . .  green-use,
```

```
        Green-use-operation-1,
        - - - -
                FROM GreenAbstractService {joint-iso-ccitt
                        mhs-motis(6) asdc(2) example(1) modules(0)
                        g-abstract-service(4)}
        - - Пример объектных идентификаторов
                id-ac-g-management, id-ac-g-use, id-as-g-management,
                  id-as-g-use, id-ase-g-management, id-ase-g-use
                - - - -
                FROM ExampleObjectIdentifiers {joint-iso-ccitt
                        mhs-motis(6) asdc(2) example(1) modules(0)
                        object-identifiers(0)}
        - - ПБДП удаленных операций
                rOSE
                - - - -
                FROM Remote-Operations-APDUs { joint-iso-ccitt
                        remote-operations(4) apdus(1) }
        - - Управление ассоциацией
                aCSE
                - - - -
                FROM Remote-Operations-Notation-extension { joint-iso-ccitt
                        remote-operations(4) notation-extension(2) }
        - - Расширение нотации удаленных операций
                APPLICATION-CONTEXT, APPLICATION-SERVICE-ELEMENT
                - - - -
                FROM Remote-Operations-Notation-extension { joint-iso-ccitt
                        remote-operations(4) notation-extension(2)}

aCSE-AS OBJECT IDENTIFIER    = { joint-iso-ccitt association-
control(2) abstractSyntax(1) apdus(0) version1(1) }

- - Прикладной контекст
green-use-AC APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS { aCSE }
        BIND AuthenticateUser
        UNBIND NoOperation
        REMOTE OPERATIONS {rOSE}
        INITIATOR CONSUMER OF { green-use-ASE }
        ABSTRACT SYNTAXES { green-use-AS, aCSE-AS }
          = id-ac-g-use
green-management-AC APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS { aCSE }
        BIND AuthenticateManager
        UNBIND NoOperation
        REMOTE OPERATIONS {rOSE}
        INITIATOR CONSUMER OF { green-management-ASE }
        ABSTRACT SYNTAXES { green-management-AS, aCSE-AS }
          = id-ac-g-management
NoOperation    = UNBIND

- - Сервисный элемент прикладного уровня

green-use-ASE APPLICATION-SERVICE-ELEMENT
      CONSUMER INVOKES {
    green-use-operation-1,    }
      = id-ase-g-use

green-management-ASE APPLICATION-SERVICE-ELEMENT
      CONSUMER INVOKES {
    green-management-operation-1,    }
      = id-ase-g-management

green-use-operation-1 Green-use-operation-1    =1
```

```
. . .

green-management-operation-1 Green-management-operation-1 ::= 50
. . .

- - Абстрактный синтаксис

green-use-AS OBJECT IDENTIFIER ::= id-as-g-use

green-management-AS OBJECT IDENTIFIER ::= id-as-g-management

END - - Конец реализации зеленой среды
```

ПРИЛОЖЕНИЕ В
(обязательное)

СПРАВОЧНОЕ ОПРЕДЕЛЕНИЕ ОБЪЕКТНЫХ ИДЕНТИФИКАТОРОВ

Данное приложение определяет (в справочных целях) различные объектные идентификаторы, приведенные в модулях ACH.1 приложения С. Здесь используется ACH.1.

Все объектные идентификаторы, присвоенные в настоящей части ГОСТ Р ИСО/МЭК 10021, присвоены в данном приложении, за исключением тех, которые присвоены в приложении А. Данное приложение является определяющим для всего, кроме модулей ACH.1 и самого субъекта соглашений по определению услуг прикладного уровня. Определяющие обозначения для формы встречаются в самих модулях; другие ссылки на них используются в разделах IMPORT. Последние являются фиксированными.

```
ASDCObjectIdentifiers {joint-iso-ccitt mhs-motis(6) asdc(2)
            modules(0) object-identifiers(0)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

- - Пролог

- - Экспортирует все

IMPORTS - - ничего - -

ID ::= OBJECT IDENTIFIER

- - Соглашения по определению абстрактных услуг (неопределяющие)

id-asdc ID ::= {joint-iso-ccitt mhs-motis(6) asdc(2) - - неопределяющие

- - Категории

id-mod ID :: = (id-asdc 0) - - модули; неопределяющие
id-ex  ID :: = (id-asdc 1) - - пример; неопределяющий

- - Модули

id-mod-object-identifiers ID ::= (id-mod 0) - - неопределяющие
id-mod-notation           ID ::= (id-mod 1) - - неопределяющие

END - - Конец объектных идентификаторов COAY
```

ПРИЛОЖЕНИЕ С
(обязательное)

СПРАВОЧНОЕ ОПРЕДЕЛЕНИЕ НОТАЦИИ

Данное приложение, являющееся дополнением ко второй главе, определяет для справочных целей нотацию специфицирования абстрактных моделей и услуг. В нем используется АСН.1.

```
AbstractServiceNotation { joint-iso-ccitt mhs-motis(6) asdc(2)
        modules(0) notation(1) }
DEFINITIONS IMPLICIT TAGS :: =
BEGIN

- - Пролог

EXPORTS
    ABSTRACT-BIND, ABSTRACT-ERROR, ABSTRACT-OPERATION,
    ABSTRACT-UNBIND, OBJECT, PORT, REFINE;

IMPORTS
    - - Нотация удаленных операций
      BIND, ERROR, OPERATION, UNBIND
      - - - -
      FROM Remote-Operation-Notation { joint-iso-ccitt
                remote-operations(4) notation(0) };

- - Макрокоманды объекта

OBJECT MACRO :: =
BEGIN

TYPE NOTATION   ::= «PORTS» « { » PortList « } » | empty
VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

PortList        ::= Port «,» PortList | Port
Port            ::= value (Port) PortType

PortType        ::= Symmetric | Asymmetric

Symmetric       ::= empty
Asymmetric      ::= Consumer | Supplier

Consumer        ::= «[C]»
Supplier        ::= «[S]»

END

- - Макрокоманда порта


PORT MACRO ::=
BEGIN

TYPE NOTATION     ::= Operations | empty
VALUE NOTATION    ::= value (VALUE OBJECT IDENTIFIER)

Operations        ::= Symmetrical | Asymmetrical

Symmetrical       ::= «ABSTRACT» «OPERATIONS» «{» OperationList «}»
Asymmetrical      ::= OneSided | TwoSided
```

```
OneSided            ::= Consumer | Supplier
TwoSided            ::= Consumer Supplier | Supplier Consumer

Consumer            ::= «CONSUMER» «INVOKES» « { » OperationList « } »
Supplier            ::= «SUPPLIER» «INVOKES» « { » OperationList « } »

OperationList       ::= Operation «,» OperationList | Operation
Operation           ::= value (ABSTRACT-OPERATION) |
        - - значение данных, идентифицирующее абстрактную операцию
                type
        - - тип данных, идентифицирующий абстрактную операцию

END

- - Макрокоманды уточнения

REFINE MACRO        ::=
BEGIN

TYPE NOTATION       ::= Object «AS» ComponentList
VALUE NOTATION      ::= value (VALUE OBJECT IDENTIFIER)

ComponentList       ::=Component ComponentList | Component
Component           ::= ObjectSpec Ports

ObjectSpec          ::= Object | Object «RECURRING»

Ports               ::= PortSpecList | empty
PortSpecList        ::= PortSpec PortSpecList | PortSpec
PortSpec            ::= value (PORT) PortSide PortStatus

PortSide            ::= Consumer | Supplier | empty
Consumer            ::= «[C]»
Supplier            ::= «[S]»

PortStatus          ::= «VISIBLE» | «PAIRED» «WITH» ObjectList

ObjectList          ::= Object «,» ObjectList | Object
Object              ::= value (OBJECT)

END

- - Макрокоманды абстрактной связки, развязки, операции и ошибок

ABSTRACT-BIND MACRO :: =
BEGIN

TYPE NOTATION       ::= Ports Bind
VALUE NOTATION      ::= value (VALUE BindType)

Ports           ::= «TO» «(» PortList «)» | empty
PortList        ::= Port «,» PortList | Port
Port            ::= value (PORT) PortSide
PortSide        ::= Consumer | Supplier | empty
Consumer        ::= «[C]»
Supplier        ::= «[S]»

Bind            ::= type (BindType) - - должен быть типа BIND
                  | empty <BindType ::= BIND>

END
```

```
ABSTRACT-UNBIND MACRO ::=
BEGIN

TYPE NOTATION     ::= Ports UnBind
VALUE NOTATION    ::= value (VALUE UnBindType)

Ports             ::= «FROM» « ( » PortList « ) » | empty
PortList          ::= Port «,» PortList | Port
Port              ::= value (PORT) PortSide
PortSide          ::= Consumer | Supplier | empty
Consumer          ::= «[C]»
Supplier          ::= «[S]»
Unbind            ::= type (UnbindType) - - должен быть типа UNBIND
                      | empty <UnbindType ::= UNBIND>

END

ABSTRACT-OPERATION MACRO :: = OPERATION

ABSTRACT-ERROR    MACRO :: = ERROR

END - - Конец нотации абстрактных услуг
```

ПРИЛОЖЕНИЕ D
(информационное)

РАЗЛИЧИЯ МЕЖДУ ГОСТ Р ИСО/МЭК 10021-3 И РЕКОМЕНДАЦИЕЙ Х.407 МККТТ

В данном приложении перечислены все, кроме чисто стилистических, различия между настоящей частью ГОСТ Р ИСО/МЭК 10021 и соответствующей рекомендацией МККТТ.

Между этими двумя спецификациями никаких различий нет.

ПРИЛОЖЕНИЕ Е
(информационное)

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

В данном приложении содержится алфавитный указатель для настоящей части ГОСТ Р ИСО/МЭК 10021. В нем указаны номера разделов, в которых определен каждый элемент каждой категории.

В приложении указаны элементы (при их наличии) следующих категорий:

а) сокращения;
б) термины;
в) единицы информации;
г) модули АСН.1;
д) макрокоманды АСН.1;
е) типы АСН.1;
ж) значения АСН.1.

Е.1 Сокращения

АСН.1 3
ВОС 3
ПБДП 3
ПК 3
СУО 3
СЭП 3

Е.2 Термины

абстрактная ассоциация 7.3
абстрактная модель 7
абстрактный объект 7.1
абстрактная операция 8.4
абстрактная ошибка 8.5
абстрактный порт 7.2
абстрактная процедура 8.1
абстрактная услуга 7.3
абстрактное уточнение 7.4
аргумент 8.1
асимметричный 7.2
ассоциация 7.3
выполнение 8.1
запросчик 8.1
инициатор 8.2
информация об ошибке 8.3
исполнитель 8.1
модель 7
операция абстрактной развязки . . 8.3
операция абстрактной связки 8.2
ответчик 8.2
объект 7.1
параметр 8.5
пользователь 7.3
пользователь абстрактной услуги . 7.3
порт 7.3
поставщик 7.2
поставщик абстрактной услуги . . . 7.3
потребитель 7.2
привлечение 8.1
процедура 8.1
развязка 7.2
связка 7.2
реальный 9
результат 8.1
симметричный 7.2
совместимый 7.2
уточнение 7.4

УДК 681.324:006.354 ОКС 35.100.70 П85 ОКСТУ 4002

Ключевые слова: обработка данных, обмен информацией, взаимосвязь сетей, взаимосвязь открытых систем, передача данных, передача текста, сообщения, определения

Редактор *Л.В. Афанасенко*
Технический редактор *О.Н. Власова*
Корректор *О.В. Ковш*
Компьютерная верстка *Е.Н. Мартемьяновой*

Изд. лиц. № 021007 от 10.08.95. Сдано в набор 07.09.98. Подписано в печать 10.12.98. Усл. печ. л. 3,72. Уч.-изд. л. 3,35. Тираж 238 экз. С1074. Зак. 673.

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

gost.e-ecolog.ru