В.А. Русаков

##### **ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ**

Учебное пособие

МОСКВА 2001

В.А. Русаков

##### **ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ**

Учебное пособие

МОСКВА 2001МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ

(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

В.А. Русаков

##### Взаимосвязь открытых систем

*Учебное пособие*

Москва 2001

УДК 004.728(075)

ББК 32.973.202я7

Р.88

Русаков В.А. Взаимосвязь открытых систем: *Учебное пособие.* М.: МИФИ*, 2001. 120. с.*

Описана организация взаимосвязи открытых систем – иерархия уровней эталонной модели ВОС. Рассмотрены функции, протоколы и сервис уровней, использование формальных методов их описания, а также вопросы реализации ПО открытых систем.

Предназначено для студентов пятого курса факультета кибернетики, изучающих курс «Взаимосвязь открытых систем».

Рецензент: Левятов И.Д., к.т.н.

Рекомендовано редсоветом МИФИ в качестве учебного пособия

© В.А. Русаков

© Московский государственный инженерно-физический институт (технический университет), 2001

Редактор и техн. редактор М.В. Макарова

ЛР №020676 от 09.19.97

Подписано в печать 17.05.2001. Формат 60×84 1/16

Печ. л. 7,5. Уч.-изд. л. 7,5. Тираж 200 экз.

Изд. №036-1. Заказ №

Московский государственный инженерно-физический институт

(технический университет). Типография МИФИ

115409, Москва, Каширское ш.,31

###### СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ 4

ВВЕДЕНИЕ 6

1. ЭТАЛОННАЯ МОДЕЛЬ ВОС 9

1.1. Общие положения и понятия 9

1.2. Функции уровня 21

1.3. Сервис уровня 25

1.4. Формализмы описания сервиса и протоколов 31

1.5. Взаимодействие уровней и пользователей служб 35

1.6. Вопросы к разд. 1 40

2. УРОВНИ ЭТАЛОННОЙ МОДЕЛИ ВОС 41

2.1. Сетезависимые уровни 41

2.2. Транспортный уровень 48

2.3. Сеансовый уровень 58

2.4. Уровень представления 68

2.5. Прикладной уровень (общий прикладной сервис) 81

2.6. Вопросы к разд. 2 93

3. СПЕЦИАЛЬНЫЕ ЭЛЕМЕНТЫ ПРИКЛАДНЫХ СЛУЖБ И РЕАЛИЗАЦИЯ ОТКРЫТЫХ СИСТЕМ 94

3.1. О программной реализации 94

3.2. Управление ВОС 107

3.3. Служба справочника 113

3.4. Вопросы к разд. 3 118

СПИСОК ЛИТЕРАТУРЫ 120

# СПИСОК СОКРАЩЕНИЙ

|  |  |
| --- | --- |
| АД | * атомарное действие |
| АПС | * агент пользователя справочника |
| АСС | * агент справочной службы |
| БДП | * блок данных протокола |
| БДС | * блок данных службы |
| БСИ | * база справочной информации |
| БУС | * блок управления событиями |
| БфДП | * буфер данных пользователя |
| ВС | * вычислительная сеть |
| ВТ | * виртуальный терминал |
| ИВС | * информационно-вычислительная сеть |
| ИДС | * информационное дерево справочника |
| ИСУ | * информационная служба управления |
| КА | * конечный автомат |
| КТС | * конечная точка соединения |
| ЛВС | * локальная вычислительная сеть |
| МККТТ | * международный комитет по телеграфии и телефонии |
| МОС (ISO) | * международная организация по стандартизации |
| МФО | * метод формального описания |
| НАС | * нотация абстрактного синтаксиса |
| ОРИ | * относительное различаемое имя |
| ОЭПС | * общий элемент прикладных служб |
| ПДУФ | * передача, доступ и управление файлами |
| ПОУС | * прикладной объект управления системой |
| ПП | * прикладной процесс |
| ППМС | * прикладной процесс менеджера сети |
| ППУС | * прикладной процесс управления системой |
| САС | * системный агент справочника |
| СВОС | * среда взаимосвязи открытых систем |
| СОС | * система обработки сообщений |
| СРС | * среда реальных систем |
| СС | * сетевое соединение |
| СС | * справочная служба |
| СЭПС | * специальный элемент прикладных служб |
| ТДС | * точка доступа к службе |
| ТС | * транспортное соединение |
| УБ | * управление безопасностью |
| УЗПВ | * управление завершением, параллельностью и восстановлением |
| УИП | * управляющая информация протокола |
| УКИ | * управление конфигурацией и именами |
| УО | * управление при отказах |
| УУ | * управление учетом |
| УЭФ | * управление эффективностью функционирования |
| ЭАУС | * элемент агента управления системой |
| ЭМВОС | * эталонная модель взаимосвязи открытых систем |
| ЭМУС | * элемент менеджера управления системой |
| ЭП | * элемент пользователя |
| ЭСДС | * элемент службы доступа к справочнику |
| ЭССС | * элемент системной службы справочника |
| ЭСУА | * элемент службы управления ассоциацией |
| ЭСУО | * элемент службы удаленных операций |

# ВВЕДЕНИЕ

Вначале несколько слов об истории предметной области курса. Системы интересующего нас типа, в которых с помощью средств связи объединены вычислительные машины и терминалы, примерно с начала 70-х годов получают все большее распространение. Они становятся предметом самого пристального изучения как для отдельных исследователей, так и для крупных национальных и международных научно-технических ассоциаций.

О причинах появления таких систем – информационно-вычислительных сетей (ИВС или, короче, вычислительных сетей, ВС) можно говорить достаточно долго. Вкратце движущая сила появления и развития ИВС может быть описана как противоречие. Противоречие между стремлением, с одной стороны, сконцентрировать ресурсы по хранению и обработке информации в небольшом числе мест и, с другой стороны, предоставить эти ресурсы возможно большему числу пользователей.

Концентрация ресурсов вызывается, в свою очередь, целым рядом причин – экономических, исторических, юридических, технологических, стремлением обезопасить и сохранить информацию и т.п. Во многих отраслях человеческой деятельности имеется тенденция укрупнять и усложнять единичные установки (вычислительная техника в этом смысле – не исключение).

Стремление увеличить число пользователей имеет под собой очевидное экономическое основание: расширение использования информации способствует повышению эффективности общественного производства и, в конечном счете, прогрессу общества.

В этом месте стоит отметить, что благодаря использованию связных средств в ИВС термин концентрация (укрупнение) ресурсов приобретает смысл термина объединение (интеграция) распределенных ресурсов. Основная цель создания любой ИВС как раз может быть описана как интеграция информационных, программных и технических ресурсов для оперативного их использования многими пользователями.

Систему, рассчитанную на интеграцию ресурсов произвольного вида и стандарта в рамках большого региона, страны или сообщества стран, часто называют глобальной ВС. Такая система имеет в своем распоряжении все средства для интеграции ресурсов и для своего развития. Локальная ВС, со своей стороны, ориентирована на объединение ресурсов ограниченного вида и на небольшом пространстве. Часто такими ресурсами являются дисковый накопитель большой емкости (информация на нем) и высококачественный принтер, а пользователи находятся друг от друга на расстоянии не превышающем нескольких сотен метров. ЭВМ ЛВС принадлежат обычно одному и тому же классу, программы удовлетворяют одному стандарту и т.п. ЛВС, как правило, создаются для конкретных прикладных задач – управление производством, обработка телеметрии, локальная банковская система и т.п.

Даже приведенная простейшая классификация сетевых систем имеет свои основания – экономические, технологические, архитектурные. Ныне существующие ИВС могут объединять миллионы разнородных пользователей и труднообозримое множество разнотипных ресурсов. Принципы построения и развития столь сложных систем также сформировались не на пустом месте. Концепция архитектуры таких систем возникла как результат интенсивных поисков коренных причин затруднений и тупиков, возникавших ранее в ходе их создания, как результат поисков идей и средств выхода из таких тупиков.

Первые системы, которые предоставляли удаленным пользователям свои ресурсы, создавались в те времена, когда значение многих понятий, составляющих ныне фундамент архитектуры связи систем распределенной обработки информации, не только не осознавалось в полной мере – эти понятия в привычном сегодня виде еще не были сформулированы.

Это обстоятельство имело очень важное проявление – в таких системах смешивались функции управления тремя различными типами объектов: терминалами, линиями связи, данными. Смеши­вание функций означает здесь то, что они делались зависимыми. Как следствие – сис­темы создавались исключительно негибкие, их сложно было моди­фицировать и практически невозможно было совмещать друг с другом в какой-либо части (аппа­ратно/ программно/ информационной). Между тем потребность в таких совмещениях все более возрастала – с ростом вложений в линии связи, терминалы, обрабатывающие и информационные ресурсы вычислительных центров становилось ясно, что невоз­можность использования данного терминала и/или его линии связи для ра­боты с другой системой, а так же невозможность оперативного доступа с этого терминала к другой ЭВМ означает, в конечном счете, сниже­ние эффективности использования этих ресурсов, снижение отдачи от вложенных средств. Негибкость возникающих систем, со своей стороны, означала, что попытки их модификаций в совмещаемых частях практически всегда приводили к столь глубоким трансфор­мациям систем в целом, что возникающие при этом затраты вре­мени и материальных ресурсов оказывались сравнимыми с затра­тами на начальные варианты систем.

Помимо описанных проблем организации эффективной совме­стной работы различных уже существовавших систем, имелись внешне похожие проблемы, обусловленные длительностью времени становления и развития больших ИВС. Такие глобальные ВС – крайне дорогостоящие объекты инфраструктуры – однажды будучи созданными в виде некоторого отправного варианта, эволюционируют в течение целого ряда лет. За это время неоднократным изменениям может подвергнуться и сама ВС и ее окружение – начиная от технологии производства и важных параметров аппаратуры, систем связи, программного обеспечения и кончая стратегией развития самой ВС.

Сказанное хотя бы в некоторой степени может помочь оценить огромную работу международного сообщества специалистов и организаций, в результате которой были созданы архитектурные концепции построения сетевых систем, концепции, рамки которых должны быть, с одной стороны, достаточно мягкими, чтобы обеспечивать потребности длительного развития таких систем в условиях разнообразия предъявляемых к ним запросов и ограничений и, с другой стороны, достаточно жесткими, действенными, чтобы обеспечить проявление положительных сторон широко понимаемых унификации и стандартизации.

# ЭТАЛОННАЯ МОДЕЛЬ ВОС

В разделе рассматриваются основные положения и понятия ЭМВОС. Для большой самостоятельной проработки рекомендуются принципы именования и адресования в среде ВОС, а также методы формального описания и анализа протоколов.

## Общие положения и понятия

Под архитектурой системы здесь и далее будем понимать ее функциональную структуру. Вся совокупность функций взаимосвязи систем чрезвычайно сложна, поэтому она представляется декомпозицией этих функций в виде иерархически упорядоченных компонентов. Компоненты одинаковой иерархической подчиненности образуют уровень. Уровни конструируются независимыми друг от друга, что достигается точным определением поведения уровня – сервиса, предоставляемого данным уровнем. Такой подход позволяет использовать уровневые услуги, образующие сервис, не вникая в подробности функционирования подсистем обеспечения данных услуг. Эти положения зафиксированы в основополагающем документе ISO 7498, принятом в качестве международного стандарта по линии Международной организации по стандартизации (МОС).

В этом документе рекомендована к применению так называемая эталонная модель взаимосвязи открытых систем (ЭМВОС). Модель представляет собой эквивалентную форму описания ИВС, ее структуры, входящих компонентов, функций информационных ресурсов, а также правил взаимодействия элементов ИВС в процессе функционирования. Модель основывается на трех базовых понятиях: системы, которые соответствуют основным элементам ИВС; прикладные процессы, характеризующие информационные ресурсы ИВС; соединения, обеспечивающие обмен информацией между прикладными процессами. Здесь следует оговориться, что соединение в этом месте трактуется максимально широко. Под ним надо понимать кооперацию, которая может принимать самые разнообразные формы. Это и межпроцессорная связь, относящаяся к обмену информацией и синхронизации действий между прикладными процессами; и представление данных, относящееся ко всем аспектам создания и поддержания описаний и преобразований данных; и управление ресурсами, с помощью которых инициируются прикладные процессы; и обеспечение целостности и сохранности данных во время функционирования систем.

Восприятие образа “открытой” системы как системы, готовой и способной к взаимосвязи с другой подобной системой, интуитивно подкреплено ощущением необходимости определенной стандартизации процесса обмена информацией между ними. Это обстоятельство, наряду с атрибутами реально существующих систем, учтено в следующих пояснениях терминов.

Реальная система – совокупность ЭВМ с соответствующим ПО и периферийным оборудованием, терминалов, операторов ЭВМ, физических процессов, средств передачи данных и т.д., которая образует единое целое и способна обрабатывать и/или передавать данные.

Реальная открытая система – реальная система, удовлетворяющая стандарту ISO 7498[[1]](#footnote-1) при ее взаимосвязи с другими системами.

Открытая система – формальное представление в рамках эталонной модели аспектов реальной открытой системы, связанных с взаимодействием открытых систем.

Прикладной процесс выполняет обработку данных для некоторого приложения. При более детальном рассмотрении этих процессов в них можно выделить функции, которые связаны с обеспечением взаимодействия. Эти функции называются прикладными объектами. Такие объекты – части открытых систем.

Наконец, для обеспечения передачи данных между системами необходима некоторая физическая среда. Это могут быть провода, радиоэфир и т.п.

Под средой взаимосвязи открытых систем (СВОС) понимается совокупность взаимодействующих реальных открытых систем вместе с физической средой, предназначенной для передачи данных между ними.

Введенные понятия иллюстрируются на рис 1.1.

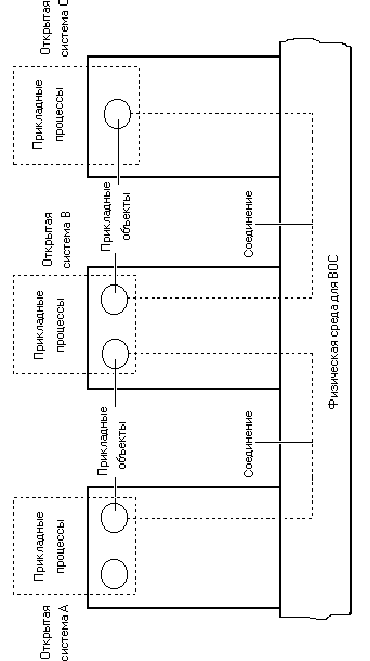


Рис. 1.1. Основные элементы среды ВОС

МОС рекомендовала к использованию семиуровневую иерархию функций, обеспечивающих взаимодействие прикладных процессов, распределенных в различных системах, рис. 1.2.

Прикладной уровень

Уровень представления

Транспортный уровень

Сеансовый уровень

Сетевой уровень

Канальный уровень

Физический уровень

Физическая среда для ВОС

Рис. 1.2. Уровни ЭМВОС

Границы между уровнями установлены так, чтобы взаимодействие между смежными уровнями было как можно меньшим, общее число уровней – сравнительно небольшим, а изменения, проводимые в пределах одного уровня, не требовали бы перестройки смежных уровней.

Значение ЭМВОС заключается в том, что она вводит единый перечень понятий и общепринятый способ разделения функций взаимосвязи на уровни. Однако ЭМВОС не является стандартом протоколов для каждого из уровней, она служит скорее шаблоном, задает рамки, с учетом которых разрабатываются стандарты по протоколам соответствующих уровней.

В соответствии с эталонной моделью открытая система образуется из упорядоченной совокупности подсистем. Подсистема (N-подсистема) – это компонента иерархического разделения функций открытой системы, которая непосредственно взаимодействует только со смежной верхней или со смежной нижней компонентой

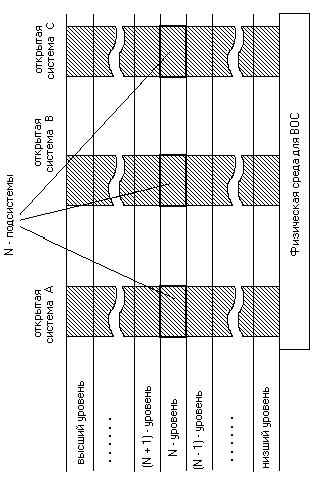


Рис. 1.3. Уровневая организация ВОС

такого разделения. Под N-уровнем понимается подмножество архитектуры ВОС, образованное подсистемами одного и того же ранга N. Под подсистемами одного и того же ранга понимаются такие подсистемы различных открытых систем, которые выполняют одно и то же подмножество функций (рис.1.3).

Для локализации функций (групп функций) используется понятие (логического) объекта (N-объекта). Под N-объектом понимается активный элемент внутри N-подсистемы, выполняющий некоторое подмножество ее функций. В ЭМВОС считается, что объекты имеются на каждом уровне, а N-подсистема состоит из одного или нескольких N-объектов, рис.1.4.

N-подсистемы

N-уровень

Рис. 1.4. N-подсистемы и N-объекты

N-объекты

За исключением самого верхнего уровня каждый N-уровень обеспечивает для (N+1)-объектов N-услуги. Услуга уровня – это функциональная возможность, которую данный уровень ­­­­­­­вместе с нижерасположенными уровнями обеспечивает смежному верхнему уровню. Понятию “служба” соответствует понятие “набор услуг”. Совокупность услуг N-уровня называется сервисом N-уровня, или N-сервисом.

Кроме действий N-объектов, направленных на предоставление услуг вышележащему уровню, в ЭМВОС присутствуют также и внутриуровневые действия, связанные со структурой данного уровня. Возможные действия N-объектов называются N-функциями.

Служба, запрашиваемая (N+1)-объектом, может обеспечиваться как одним, так и совокупностью N-объектов. При совместной работе N-объектов на любом уровне, исключая самый нижний, используется служба, обеспечиваемая (N-1)-уровнем. Предполагается, что объекты самого нижнего уровня взаимодействуют непосредственно через связывающую их физическую среду.

Службы N-уровня предоставляются (N+1)-уровню с использованием N-функций, выполняемых на N-уровне, и при необходимости с использованием (N-1)-служб.

Высший уровень предоставляет непосредственно прикладным процессам полный набор функциональных возможностей, обеспечиваемых всеми уровнями.

Управление взаимодействием N-объектов осуществляется одним или несколькими протоколами. В эталонной модели N-протокол – это набор правил и форматов, который определяет функционирование N-объектов при выполнении ими N-функций (рис.1.5).

N-объекты

N-протоколы

N-уровень

Рис. 1.5. N-протоколы

Пунктирные стрелки, соответствующие N-протоколам, на этом рисунке символизируют всего лишь логический характер связи. Реальная передача информации осуществляется только через физическую среду. Ясно поэтому, что N-объекты могут связываться между собой только с помощью услуг, предоставляемых (N-1)-уровнем. При этом возможны случаи, когда услуги, предоставляемые (N-1)-уровнем, не позволяют напрямую взаимодействовать между всеми N-объектами.

В этих случаях взаимодействие может осуществляться, если некоторый другой N-объект выполняет функцию ретранслятора (рис.1.6). Факт ретрансляции связи N-объектом остается при этом неизвестным как (N-1)-, так и (N+1)-уровню.

Объекты смежных уровней взаимодействуют друг с другом через общую границу. Для локализации мест, в которых происходит взаимодействие, используется понятие точки доступа к N-службе (N-ТДС). N-ТДС – точка, в которой объект N-уровня предоставляет услугу объекту смежного верхнего (N+1)-уровня. Именно через N-ТДС[[2]](#footnote-2) происходит предоставление услуг N-уровнем и потребление услуг (N+1)-уровнем. При этом, если N- объекты представляют собой активные компоненты модели (объект действует, предпринимает...), то N-ТДС – пассивные компоненты модели, служащие как бы для хранения информации при передаче управляющих сигналов и данных между N- и (N+1)-объектами.

Рис. 1.6. Связь N-объектов с использованием услуг (N-1)-уровня и через N-объект-ретранслятор

N-уровень

(N-1)-уровень

N-объект-ретранслятор

N-объект

(N+1)- и N-объекты, имеющие общую ТДС, находятся в одной системе. (N+1)-объект может быть подключен к нескольким N-ТДС, соединенным с одними и теми же несколькими N-объектами. Однако в каждый момент каждая N-ТДС соединена только с одним N-объектом и только с одним (N+1)-объектом, что обусловлено связью ТДС с идентификацией (адресацией) объектов.

Допустимые отношения (N+1)-, N-объектов и N-ТДС проиллюстрированы на рис.1.7.

Местоположение N-ТДС определяется N-адресом. Если (N+1)-объект будет отсоединен от N-ТДС, то N-адрес перестанет обеспечивать доступ к (N+1)-объекту. Если N-ТДС переключается к другому (N+1)-объекту, то N-адрес идентифицирует новый (N+1)-объект, а не старый.

В соответствии с ЭМВОС у объекта имеется постоянный идентификатор – наименование. Пространство наименований – это подмножество наименований в среде ВОС. Имя пространства наименований – идентификатор, однозначно определяющий пространство наименований в среде ВОС. Локальным называется наименование, уникальное в некотором пространстве наименований; глобальным – наименование, уникальное в среде ВОС и состоящее из двух частей: имени пространства наименований и локального наименования.

Под N-справочником понимается N-функция, преобразующая глобальное наименование N-объекта в адрес одной из (N-1)-ТДС, к которой прикреплен N-объект.

Под N-отображением адреса понимается N-функция, обеспечивающая отображение между N- и (N-1)-адресами, связанными с N-объектом.

Использование N-адресов для идентификации (N+1)-объектов наиболее эффективно при постоянной привязке (N+1)-объектов к N-ТДС. В противном случае, для идентификации (N+1)-объектов следует использовать глобальные наименования.

Основной режим взаимодействия в ЭМВОС – режим с установлением соединения. Под N-соединением понимается (логическая) связь, установленная N-уровнем между двумя или более (N+1)-объектами для передачи данных.

В пределах N-уровня между парой N-ТДС, расположенных в различных системах, может быть установлено несколько N-соединений. Для того чтобы (N+1)-объект, использующий эти соединения, мог отличить одно N-соединение от другого, используется идентификатор конечной (оконечной) точки соединения (КТС). Идентификатор КТС уникален в пределах ТДС – его значение выбирается локальным образом.

Соединение является одним из основных понятий ЭМВОС. В рамках модели соединение рассматривается как единственное средство взаимодействия объектов уровня. Вместе с тем ряд положений модели не связан с механизмом взаимодействия объектов. Такие положения (концепция уровней, сервиса, протокола) могут быть соответствующим образом применимы и для режимов, когда соединение не устанавливается, а взаимодействие осуществляется с передачей отдельных независимых блоков данных. Такие режимы используются, например, в ЛВС.

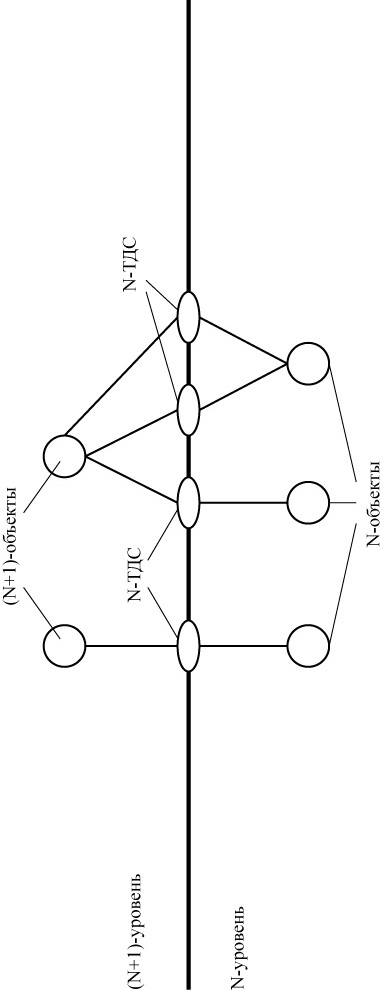


Рис. 1.7. Отношения (N+1)- и N- объектов и ТДС

Передача без соединения означает, что блок данных передается одним независимым действием без установления, поддержания и разрыва соединения. Для лучшего уяснения последствий отказа от соединения рассмотрим его основные свойства.

Соединение представляет собой логическую связь, которая устанавливается между двумя или более объектами для их взаимодействия. Объекты могут принадлежать одному или смежным уровням. В последнем случае говорят об ассоциации пользователя сервиса и поставщика. Возможность установить связь обеспечивается нижележащим уровнем, предоставляющим сервис “с соединением”. В процессе использования этого сервиса различают три различные фазы: установление соединения, передачу данных и разъединение. Каждая фаза, помимо процедурных отличий, имеет четко выделенное начало и конец. Характерно при этом, что:

* в ходе установления соединения производится согласование параметров соединения, режима передачи данных, использования необязательных процедур и т.д.;
* обеспечивается идентификация соединения, которая позволяет при последующей передаче обойтись без некоторых параметров, например, не использовать адрес;
* обеспечивается логическая связь передаваемых по соединению фрагментов данных, что позволяет эффективно сохранять их последовательность и управлять потоком данных.

Режим “без соединения”, в противоположность описанным свойствам соединения, не имеет четко выраженных фаз взаимодействия. Динамическое согласование параметров также отсутствует, хотя предварительная договоренность объектов необходима, иначе объект не сможет проинтерпретировать принятые данные. Вся информация, нужная для доставки данных – адреса, параметры качества сервиса, факультативные процедуры и пр. должны указываться при передаче каждого блока данных. Кроме того, отсутствие логической связи между передаваемыми блоками в ходе их передачи позволяет посылать их по разным маршрутам, а также копировать их для широковещательной рассылки.

Вследствие подобных четко выраженных различий сравнивать эти режимы с целью выявления лучшего бессмысленно, правомерно лишь поставить вопрос об областях их предпочтительного применения.

Режим “с соединением” целесообразно использовать для тех применений, где взаимодействие имеет долговременный характер, конфигурация взаимодействующих объектов постоянна, а поток данных однороден (нет больших пауз).

Режим “без соединения” больше подходит там, где для взаимодействия характерен небольшой объем данных, а временные интервалы между передачами относительно скорости передачи значительны. Легкость тиражирования данных и передачи их по разным маршрутам делают этот режим привлекательным в системах, к которым предъявляются повышенные требования по устойчивости к отказам.

Прикладной

уровень

Уровень

представления

Сеансовый

уровень

Канальный

уровень

Физический

уровень

Транспортный

уровень

Сетевой

уровень

С соединением

Без соединения

Рис. 1.8. Сочетание режимов с/без соединения

N-объекты, поддерживающие режим “без соединения”, могут использовать (N-1)-сервис типа “с соединением”. Возможно и обратное. Взаимодействие между службами различных типов требует преобразования одной службы в другую. Хотя в ЭМВОС в явном виде не налагается ограничений на использование подобных преобразований внутри некоторого уровня, на практике с целью снижения сложности протоколов и увеличения возможностей по взаимодействию систем некоторые ограничения все же вводятся.

Возможное сочетание типов режимов с преобразованиями в рамках отдельных уровней приведено на рис.1.8.

## Функции уровня

Произвольный N-уровень может быть представлен теми функциями, которые в нем выполняются. Среди этих функций, в общем случае, могут быть: выбор протокола; установление и расторжение соединения; мультиплексирование и расщепление соединений; передача нормальных (обычных) данных; передача срочных (внеочередных) данных; управление потоком данных; сегментирование, блокирование и сцепление данных; организация последовательности; защита от ошибок; маршрутизация.

**Выбор протокола.** На N-уровне могут использоваться несколько протоколов. Для организации N-соединения необходимо, чтобы N-объекты выбрали единый N-протокол, иначе они просто не поймут друг друга. Это может быть сделано как до организации соединения, так и во время установления соединения путем использования идентификатора N-протокола.

**Установление и расторжение соединения.** Для установления N-соединения необходимо, чтобы оба N-объекта были способны выполнить обмен данными по протоколу, а так же, чтобы (N-1)-уровень предоставил (N-1)-соединение, т.е. предоставил услуги по передаче данных N-уровня. В свою очередь для выполнения требования по (N-1)-соединению необходимо, чтобы (N-1)-уровень имел соединение, предоставленное (N-2)-уровнем и т.д. – до тех пор, пока не будет найдено подходящее соединение, или не будет достигнута граница с физической средой для ВОС. Организация N-соединения может проводиться совместно с установлением (N-1)-соединения, если (N-1)-протокол позволяет передавать в фазе установления (N-1)-соединения данные, относящиеся к N-соединению.

Расторжение N-соединения в нормальных условиях инициируется одним из связанных с ним (N+1)-объектов. Расторжение N-соединения может также инициироваться одним из поддерживающих его N-объектов в результате возникновения сбоя, ошибок в   
N-уровне или в нижележащих уровнях. В зависимости от условий расторжение N-соединения может привести или не привести к стиранию данных N-пользователя.

Заметим, что N- и (N-1)-соединения могут быть независимыми. Это означает, что возможны следующие ситуации: расторжение   
N-соединения не ведет к расторжению (N-1)-соединения; расторжение (N-1)-соединения не ведет к расторжению N-соединения. В первом случае сохраненное (N-1)-соединение может быть использовано для установления нового N-соединения. Второй случай связан с возможностью восстановления N-соединения даже тогда, когда (N-1)-соединение расторгнуто. Такое восстановление обычно связано с организацией нового (N-1)-соединения и передачей по нему данных, однозначно идентифицирующих сохраненное   
N-соединение.

**Мультиплексирование и расщепление соединений.** Между N- и (N-1)-соединениями возможны следующие соотношения: одно к одному; несколько N-соединений используют одно (N-1)-соединение (мультиплексирование); N-соединение использует несколько (N-1)-соединений (расщепление).

При мультиплексировании необходимо осуществлять функции, связанные с идентификацией данных, относящихся к разным   
N-соединениям, управлять каждым N-соединением в отдельности с тем, чтобы не превысить пропускной способности партнера, и осуществлять планирование – предусматривать появление новых N-соединений, которые отображаются в существующее (N-1)-соединение.

Расщепление связано с управлением (N-1)-соединениями, которые используются для одного N-соединения. Это управление включает в себя принятие решений о том, сколько (N-1)-соединений и с какими характеристиками необходимо запрашивать. Передаваемые по N-соединению данные дробятся и передаются по разным (N-1)-соединениям. Поэтому порядок их поступления может отличаться от порядка передачи. Для сокращения последовательности данных необходимы специальные функции контроля и восстановления.

**Передача нормальных данных.** Взаимодействие N-объектов осуществляется с помощью обмена блоками данных N-протокола (N-БДП[[3]](#footnote-3)), которые содержат управляющую информацию N-протокола, и, возможно, данные пользователя. Последними являются данные, генерируемые (N+1)-объектами. Эти данные передаются по N-соединению прозрачно, т.е. без изменения их структуры.

Данные пользователя могут передаваться как в фазе передачи данных, так и в фазах установления и расторжения N-соединения. В фазе передачи данных используются функции управления потоком, сегментирования, блокирования и сцепления данных, организации последовательности и защиты от ошибок.

**Передача срочных данных.** Срочными данными называются данные, которые обрабатываются с приоритетом по отношению к нормальным данным. Срочные данные обычно используются для целей сигнализации, экстренного уведомления о сбоях и т.п.

Поток срочных данных не зависит от состояния потока нормальных данных. Можно представить, что соединение состоит из двух подканалов: один – для нормальных данных, другой – для срочных. Каждый из подканалов управляется независимо при условии, однако, что на приемном конце срочные данные появляются не позже нормальных данных, перед которыми они были переданы. Поскольку предполагается, что срочный поток будет использоваться сравнительно редко и для передачи небольших количеств данных, для него могут быть применены упрощенные механизмы управления потоком.

**Управление потоком данных.** Различают два типа управления потоком: протокольное, при котором регулируется скорость передачи N-БДП между N-объектами; интерфейсное, при котором регулируется скорость передачи данных между (N+1)- и N-объектом.

При протокольном управлении потоком подразумевается, что протокольная управляющая информация в N-БДП содержит в том или ином виде сведения о способности партнера принять определенное количество данных.

Интерфейсное управление потоком прямо не относится к функциям взаимосвязи. Поэтому при описании механизма такого управления обычно используют локальные, т.е. справедливые в пределах одной системы, соглашения и ограничения.

**Сегментирование, блокирование и сцепление данных.** БДП различных уровней обычно различаются по размерам. Может оказаться, что размер (N+1)-БДП больше максимального размера поля данных в N-БДП. Тогда для передачи (N+1)-БДП по N-соединению необходимо в N-уровне выполнить сегментирование, т.е. разбиение (N+1)-БДП на последовательные сегменты с длиной, равной размеру поля данных N-БДП. Для сохранения идентичности (N+1)-БДП необходимо вставлять в N-БДП, содержащие сегменты (N+1)-БДП, специальные данные, позволяющие произвести сборку (N+1)-БДП при приеме.

Блокирование есть функция N-уровня, позволяющая объединить несколько (N+1)-БДП в один N-БДП. Это может потребоваться в том случае, когда максимальная длина (N+1)-БДП много меньше длины поля данных N-БДП.

Сцепление – функция N-уровня, позволяющая объединить несколько N-БДП в один блок. При этом (N-1)-уровень воспринимает сцепление БДП как один N-БДП.

**Организация последовательности.** Эта функция связана с тем, что (N-1)-услуги, предоставляемые (N-1)-уровнем, могут не гарантировать доставку данных в том же порядке, в каком они были поставлены N-уровнем. Если N-уровень нуждается в том, чтобы сохранить порядок данных, передаваемых через (N-1)-уровень, то N-уровень должен содержать механизмы организации последовательности. Организация последовательности может потребовать дополнительной N-протокольной управляющей информации. Такой информацией могут быть, например, соответствующие порядковые номера.

**Защита от ошибок.** Функция защиты от ошибок состоит из трех компонентов: подтверждения, обнаружения ошибок и уведомления о них, возврат в исходное состояние.

Функция подтверждения может использоваться N-объектами для достижения более высокой вероятности обнаружения потери N-БДП, чем это обеспечивает (N-1)-уровень. Каждый N-БДП, передаваемый между N-объектами-корреспондентами, должен идентифицироваться единственным образом так, чтобы получатель мог информировать отправителя о его приеме. Функция подтверждения также способна установить факт неприема N-БДП и принять соответствующие меры по восстановлению.

Схема однозначной идентификации N-БДП может также использоваться для поддержки других функций, таких как сегментация и организация последовательности.

Функция обнаружения ошибок и уведомления о них может использоваться N-протоколом для обеспечения более высокой вероятности обнаружения ошибок и искажений БДП, чем это обеспечивается (N-1)-услугой. Обнаружение ошибок и уведомление могут потребовать, чтобы в N-протокольную управляющую информацию были включены дополнительные идентификаторы.

Некоторые услуги требуют возврата в исходное состояние для восстановления после потери синхронизации между N-объектами-корреспондентами. Функция возврата в исходное состояние устанавливает N-объекты-корреспонденты в заранее определенное состояние с возможной потерей или дублированием данных.

**Маршрутизация.** Функция маршрутизации в N-уровне обеспечивает прохождение данных через цепочку N-объектов. Тот факт, что передача маршрутизируется промежуточными объектами, не известен ни нижним, ни верхним уровням.

## Сервис уровня

Концепция сервиса, предоставляемого уровнем, является одной из основных в модели ВОС. Вместе с тем понятие сервиса как совокупности услуг уровня формализовать непросто. Это связано с тем, что процедуры и форматы данных, которые передаются на границе уровней, определяются особенностями реализации.

Например, если уровни функционально разнесены по разным устройствам, то интерфейсное взаимодействие между уровнями может включать в себя процедуры защиты от ошибок. С другой стороны, если уровни расположены в одной операционной среде, то эти процедуры не нужны. Поэтому процедуры обмена на интерфейсе не стандартизируются. В результате, например, данные, передаваемые через интерфейс за одно взаимодействие и называемые интерфейсным блоком данных, могут в различных реализациях иметь разную структуру и длину. При рассмотрении сервиса поэтому используют понятие блока данных службы (БДС)[[4]](#footnote-4). БДС – это данные пользователя услуг некоторого уровня, в общем случае несколько интерфейсных блоков данных, идентичность которых при передаче по соединению сохраняется. Это означает, что границы, форма и содержание БДС на двух конечных точках соединения идентичны и не зависят от того, какими порциями БДС передаются по соединению.

В целом следует признать, что на ранних этапах усилий по стандартизации сетевых взаимодействий распределенный характер сервиса осознавался недостаточно. Считалось, что это протокол непосредственно влияет на внешние (с системной точки зрения) взаимодействия, тогда как реализация сервисных взаимодействий является внутренним делом реализаторов в каждом конкретном случае. Между тем нарушения в работе по локальному интерфейсу могут привести к нарушениям в работе протокола и, как следствие, к интерфейсным нарушениям на удаленной взаимодействующей стороне. В наши дни каждый стандарт, разрабатываемый в рамках модели ВОС, содержит описание и протокола, и сервиса, предоставляемого в процессе функционирования соответствующего протокола. В конце концов важность стандартизации предоставляемого сервиса заключается в том, что появляется возможность выполнять независимые реализации протокольных стандартов различных уровней, согласуя их по предоставляемому и используемому сервису, а это – существенный аспект продекларированной независимости уровней.

Необходимое при таком подходе задание внешнего поведения уровня опирается на описанную в стандарте МОС (и соответствующей Рекомендации МККТТ) абстрактную модель, включающую в себя следующие понятия: пользователь службы, поставщик службы и примитивы службы (рис.1.9).

Пользователь службы – объект в некоторой открытой системе, который использует службу через точку доступа к службе (ТДС).

Поставщик службы – некоторое множество объектов, обеспечивающих службу для ее пользователей.

Примитив службы – абстрактное, не зависящее от конкретной реализации представление взаимодействия между пользователем и поставщиком службы.

Использование примитивов не предполагает, что для реализации интерфейса достаточно только описания сервиса. Примитивы служб – это концептуальные понятия, облегчающие описание последовательности событий при доступе к сервису уровня. Тот же сервис может быть описан другим, эквивалентным набором примитивов. Кроме того, примитивы служб не используются при описании чисто локальных действий, которые не связаны с обменом данными между пользователями.

Введенная типизация примитивов разделяет их на примитивы запроса (request), индикации (indication), ответа (response) и подтверждения (confirmation).

Примитивы

службы

Пользователи

N-службы

Поставщик N-службы

Рис. 1.9. Модель службы уровня

Примитив “запрос” – представление взаимодействия, при котором пользователь службы вызывает некоторые процедуры.

Примитив “индикация” – представление взаимодействия, при котором поставщик службы указывает, что:

* им вызываются по его же инициативе некоторые процедуры;
* в другой одноуровневой ТДС была вызвана некоторая процедура.

Примитив “ответ” – представление взаимодействия, при котором пользователь службы указывает, что им закончены некоторые процедуры, вызванные примитивом “индикация”.

Примитив “подтверждение” – представление взаимодействия, при котором поставщик службы в конкретной ТДС указывает на завершение некоторых процедур, ранее вызванных посредством примитива “запрос” в этой ТДС.

Возможно, в будущем список типов примитивов будет расширен, однако в настоящее время указанных четырех типов достаточно для описания сервиса уровней модели ВОС. Тип примитива связан с направлением его передачи. Примитивы запроса и ответа передаются от пользователей службы в сторону его поставщика, а индикации и подтверждения – в обратном направлении (рис.1.10). Обозначение каждого примитива состоит из трех элементов: буквы (или букв), обозначающих уровень модели ВОС; имени примитива, которое соответствует типу описываемой услуги; типа примитива.

Уровень обозначается буквами А или Прк (Пк) (прикладной), Р или Прд (Пд) (представительный), S или Сн (сеансовый), Т (транспортный), N или Ст (сетевой), Dl или K (канальный) и Pl или Ф (физический).

Имя примитива определяется типом услуги. Например, услуга по установлению соединения описывается примитивом с именем CONNECT, по сбросу – RESET, по передаче данных – DATA и т.д.

Например, P-CONNECTresponse есть примитив службы уровня представления, относится к услуге по установлению соединения и является ответом.

Служба состоит из услуг. Услуги могут быть обязательными и факультативными (необязательными), а также подтверждаемыми и неподтверждаемыми. Обязательность услуги означает, что она должна предоставляться во всех реализациях. Факультативные услуги могут предоставляться или нет в зависимости от назначения реализации. Подтверждаемыми называются услуги, предоставление которых связано с обменом парой примитивов – примитивом запроса и примитивом подтверждения. Для некоторых неподтверждаемых услуг обмен примитивами отсутствует – здесь достаточно только передачи запроса от пользователя службы.

Рис. 1.10. Элементы стандартной диаграммы последовательности примитивов (t2>t1)

Пользователь

Пользователь

Поставщик

request

запрос

indication

индикация

t1

t2

response

ответ

confirmation

подтверждение

Упомянутым стандартом оговариваются также формальные правила составления диаграмм последовательностей примитивов. Каждая диаграмма представляется тремя полями, разделенными двумя вертикальными линиями (см. рис.1.10). Центральное поле представляет поставщика службы, а крайние поля – пользователей службы. Вертикальные линии изображают точки доступа к службе, а также течение времени – сверху вниз. Если между примитивами существует явная причинно-временная зависимость, то соответствующие точки на вертикальных линиях соединяются отрезками прямых. Если же такой зависимости нет, то используется знак тильды.

Дальнейшая детализация абстрактной модели службы заключается в формализации представления действий поставщика службы как операций над двумя очередями, соединяющими точки доступа к службе. Эти две очереди представляют одно соединение (рис.1.11).

Использование примитивов не влечет конкретной реализации ТДС. Они не обязательно прямо связаны с элементами протокола, их также не обязательно рассматривать как макровызовы при доступе к службе.

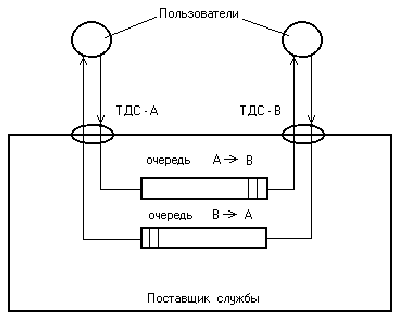


Рис. 1.11. Модель поставщика службы

Взаимодействие между пользователем и поставщиком, описываемое примитивом службы, считается мгновенным событием, которое не может быть прервано другим взаимодействием. Помимо направления взаимодействия, с каждым примитивом связан один или несколько параметров, каждый из которых имеет определенную область значений.

Кроме диаграмм для задания соотношений примитивов на разных концах соединения используются так называемые таблицы соответствия параметров примитивов службы. Такая таблица показывает, какие параметры должны быть включены в примитивы и как эти параметры связаны. Пустая клетка в таблице означает, что параметр, соответствующий табличной строке, в данном примитиве (столбце таблицы) отсутствует. Остальные обозначения имеют следующий смысл: О – наличие параметра обязательно; Н – наличие параметра не обязательно; П – наличие параметра определяется пользователем; (=) – значение параметра идентично значению аналогичного параметра в примитиве, предшествующем порождаемому; У – наличие параметра зависит от условия.

Для иллюстрации допустимости последовательностей примитивов используются таблицы следования (упорядочивания), или помеченные графы (диаграммы состояний-переходов).

В квадратной таблице следования i-й строке (i-му столбцу) соответствует i-й примитив, а пересечение i-й строки и j-го столбца помечается подходящим символом, означающим допустимость (или недопустимость) следования примитива i за примитивом j.

В помеченном графе, описывающем допустимые последовательности примитивов (на одном конце соединения), вершины соответствуют состояниям конечной точки соединения, а дуги – переданным примитивам. Среди вершин всегда имеется вершина “исходное” (“бездействие”). Такое состояние является начальным и конечным для любой допустимой последовательности примитивов. На это должно указывать наличие дуг-примитивов типа “разъединение”, “прекращение” и т.п., ведущих изо всех прочих вершин графа в “бездействие”.

## Формализмы описания сервиса и протоколов

Первоначально описания протоколов носили неформальный, словесный характер. Такие описания вносили иллюзию простоты в решение всех проблем. Однако практика очень быстро убедила разработчиков в обратном – субъективная природа восприятия словесных описаний не позволяет согласовать разрабатываемые стандарты, описания носят неоднозначный характер, не обладают полнотой, не имеют формальной основы для анализа и, в целом, приводят к несовместимости дорогостоящих программно-технических изделий и не могут служить долгосрочной основой для развития сложных распределенных систем.

Выход может быть найдет только при использовании методов формального описания (МФО) протоколов и сервисов, полно и однозначно определяющих все аспекты взаимодействия. К настоящему времени не удалось выработать общепризнанные МФО. Существуют веские основания того, что в обозримом будущем подобное положение сохранится. Среди причин такого положения важнейшими являются:

* сложность, обусловленная громоздкой структурой самих протокольных объектов и комбинаторной сложностью задач взаимодействия;
* большое разнообразие протоколов – от простых стартстопных протоколов передачи данных до сложных протоколов баз данных;
* ряд трудно поддающихся описанию аспектов протоколов, например адресации, мультиплексирования, управления потоком;
* противоречия между требованиями к описаниям всеобщих стандартов и к описаниям для реализации в конкретном программно-аппаратном окружении.

Тем не менее, имеющийся в области описания стандартов прогресс позволяет выделить среди МФО прежде всего такие языки, как Estelle и LOTOS, разработанные в МОС; SDL, разработанный в МККТТ; а среди МФО, предназначенных для реализации в конкретном программно-аппаратном окружении, следует выделить один из самых ранних методов – язык FAPL, используемый для анализа протоколов IBM.

Требования пользователей

Уровень

Рис.1.12. Внешнее поведение уровня

Отправной точкой при описании протоколов является многоуровневый характер архитектуры. При определении сервиса внутренняя природа и структура уровня несущественны – внимание концентрируется на наблюдаемом поведении уровня в терминах входных и выходных событий, происходящих на его границе (рис.1.12). Так как сервис по своей природе является распределенным, то методы его описания должны определять не только события, происходящие в каждой точке доступа к этому сервису, но и связь между событиями различных точек доступа. О важности спецификации сервиса речь уже шла ранее.

Требования пользователей

Объект 1

Объект 2

Протокол

Сервис нижележащих уровней

Рис. 1.13. Внутренняя структура уровня: взаимосвязь одноуровневых объектов

После формальной спецификации требований, предъявляемых к протоколу в виде сервиса, необходимо более детально рассмотреть способ обеспечения этих требований. Для этого определяются внутренняя структура уровня и выполняемые им функции. Как правило, уровень состоит из набора объектов, взаимодействующих друг с другом для согласования своих действий. Описав поведение этих объектов, можно полностью определить протокол (рис.1.13). Как видно, взаимодействия объектов выполняются по запросам пользователей в соответствии с протоколом с помощью сервиса расположенных ниже уровней. Поэтому описания объектов должны включать не только протокольные взаимодействия одного уровня, но и взаимодействия, обусловленные предоставлением и потреблением услуг объектами смежных уровней.

В итоге можно выделить два типа формальных спецификаций: спецификацию сервиса, предоставляемого уровнем, и спецификацию поведения объектов в процессе предоставления сервиса. К МФО этих спецификаций в МОС были выработаны единые требования.

Лежащие в основе МФО модели могут быть разделены на две группы: автоматные модели и модели последовательностей.

Автоматные модели рассматривают внутреннее состояние объекта спецификации и описывают все возможные изменения этого состояния при воздействии на объект. К моделям этого вида относятся традиционные конечные автоматы, регулярные выражения, сети Петри, расширенные автоматы, формальные грамматики, алгоритмические языки.

Модели последовательностей рассматривают только наблюдаемое извне поведение объекта, не делая никаких предположений о его внутренней структуре. К таким моделям относятся абстрактные типы данных; описания, использующие временную логику; исчисление взаимодействующих процессов; переменные истории.

В целом природа моделей последовательностей более пригодна для спецификации сервисов, а автоматных моделей – для спецификации протоколов.

Рассмотрим далее очень кратко первые типы моделей в обеих группах.

**Конечные автоматы.** Спецификации протоколов на основе моделей конечных автоматов в настоящее время используются наиболее широко. Формально конечный автомат (КА) определяется шестеркой объектов КА = {S, I, O, N, M, S0}, где S – конечное множество состояний, I – конечное множество входов, O – конечное множество выходов, N:I×S→S – функция переходов, M:I×S→O – функция выходов, S0 – начальное состояние.

Функции N и M описывают поведение автомата, т.е. если в некотором текущем состоянии si ∈ S на входе появляется сообщение (событие) i ∈ I, то функция переходов определяет новое состояние автомата sk ∈ S, а функция выходов – выходное сообщение (событие) oi ∈ O.

Для описания КА используются различные способы, однако наиболее широко распространены диаграммы состояний-переходов и таблицы решений (состояний-событий).

Диаграмма состояний-переходов представляет собой разновидность ориентированного графа, множество вершин которого соответствует множеству состояний, а множество дуг – множеству переходов. Дуги помечаются соответствующими входными и выходными сообщениями.

Таблица состояний-событий образована элементами – пересечениями столбцов, соответствующих входам i∈I, и строк, соответствующих состояниям s∈S. Элементам таблицы соответствуют пары вида (s, o), где s – новое состояние, а o∈O – выход. В представлениях таблиц смысл строк и столбцов можно, очевидно, поменять местами.

КА наиболее удобны для описания таких ситуаций, как, например, установление или разрыв соединения, восстановление после ошибок, когда количество состояний относительно невелико.

**Абстрактные типы данных.** Возникли из идеи объединить данные и операции манипулирования данными. Спецификацию с использованием абстрактных типов данных называют иногда алгебраической. Более формально под нею понимается тройка (S, Σ, E), где S – конечное множество имен типов; Σ – конечное множество имен операций, замкнутых на S; E – множество аксиом, определяющих результаты операций над определяемыми типами.

Достоинством абстрактных типов является независимость описания от способов реализации и возможность использования в процессе анализа таких спецификаций автоматизированных систем логического вывода. В то же время методы, основанные на абстрактных типах, имеют ряд ограничений, связанных с описанием параллельности и композиции различных подсистем, а также требуют немалых усилий для формулирования набора аксиом.

## Взаимодействие уровней и пользователей служб

Рассмотрим теперь ряд описанных понятий в их взаимосвязи при отработке элементарных взаимодействий соответствующими механизмами.

Обычно некоторая конкретная передача начинается с того, что пользователь данного N-уровня передает через интерфейс примитив запроса. Реагируя на него, локальный (местный) N-объект генерирует блок данных N-протокола (N-БДП). В него, помимо прочего, входит управляющая информация N-протокола (N-УИП), используя которую локальный N-объект предполагает осуществить элементарное взаимодействие с “удаленным“ равноуровневым N-объектом. С помощью служб нижележащего (N-1)-уровня N-БДП пересылается корреспондирующему N-объекту удаленной системы. Получив N-БДП, этот N-объект формирует примитив индикации и передает его вверх корреспондирующему пользователю. В случае N-службы без подтверждения передача на этом заканчивается.

Пользователь N-службы

Запрос

Подтверждение

Запрос

Корреспондирующий пользователь N-службы

С подтверждением

Индикация

Ответ

С подтверждением

Индикация

N-уровень

(поставщика

N-службы)

Время

Рис. 1.14. Примитивы N-службы во временной последовательности

Для N-службы с подтверждением корреспондирующий пользователь после этого выдает примитив ответа. Локальный N-объект (удаленной системы) генерирует соответствующий N-БДП, который с помощью (N-1)-служб нижележащего уровня пересылается обратно. Получив этот N-БДП, исходный N-объект синтезирует примитив подтверждения и передает его вверх пользователю, завершая тем самым передачу, рис. 1.14.

На рис. 1.15 те же взаимодействия изображены в архитектурно-пространственном представлении. Заметим, что взаимодействие, которое показано на рисунке горизонтальным пунктиром, – логическое взаимодействие (соответствие). Это относится и к взаимодействию показанных на рисунке пользователей N-службы, и к N-протоколу – взаимодействию N-объектов. Передача N-БДП – это также, вообще говоря, передача по логическому “каналу”. Реальная межсистемная передача осуществляется, очевидно, лишь в физической среде для ВОС. Локализованная реальная передача выполняется через интерфейсы, что соответствует вертикальным сплошным линиям на рис. 1.15.

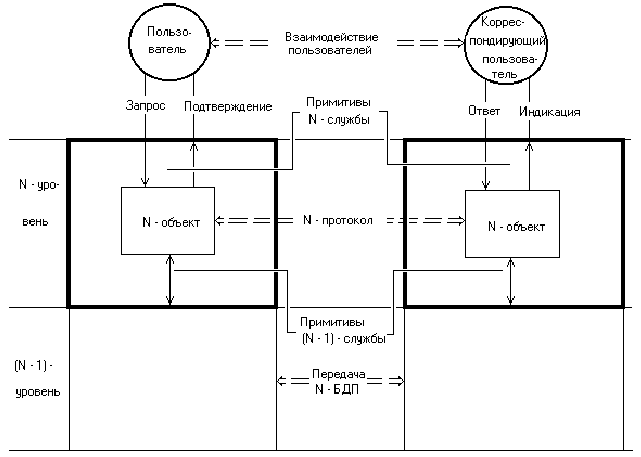


Рис. 1.15. Взаимодействие пользователей N-службы

Полная иерархия вкладываемых друг в друга N-БДП с соответствующими N-УИП изображена на рис. 1.16.

На нем представлены упорядоченные по вертикали все семь функциональных уровней ЭМВОС. На этом рисунке пунктирные горизонтали иллюстрируют взаимодействие в рамках каждого из уровней по соответствующему N-протоколу (ср. с рис. 1.15), т.е. использование (интерпретацию) соответствующей N-УИП. Для физического уровня протокольная сущность пунктирной горизонтали обозначена на рисунке в явном виде потому, что на этом уровне организация логического взаимодействия Ф-объектов производится специфично. Такая организация проявляется в использовании значительного числа управляющих цепей на интерфейсе ООД/АКД[[5]](#footnote-5), в возможном использовании управляющих цепей в сочетании с цепями приема/передачи или в применении специально сформированного подканала в цифровых сетях. Специфика Ф-уровня отчасти обусловлена его особым положением в иерархии уровней ЭМВОС – снизу он граничит непосредственно с физической средой, так что сервисной поддержки нижележащих уровней он лишен.

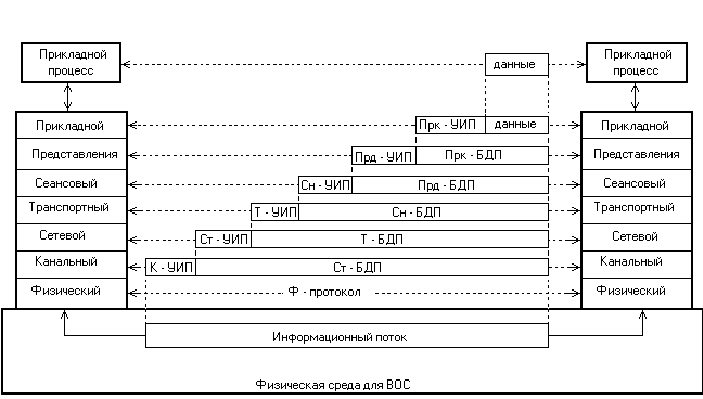


Рис. 1.16. Уровни ЭМВОС и вложения N-БДП

Как видим, блок “данные”, которые на рис. 1.16 символизируют взаимодействие пользователей (прикладных процессов), по мере своего продвижения на передающем конце вниз от уровня к уровню “обрастает” все большим объемом управляющей информации. На приемном конце ситуация обратная – на каждом уровне N-объекты выделяют и интерпретируют свою УИП, а оставшуюся информацию передают далее наверх.

Элементарное действие такого рода на передающем конце изображено на рис. 1.17. С каждым примитивом службы связан некоторый четко описанный набор параметров. Уровень, принимающий параметр “данные пользователя”, трактует этот параметр как БДС. Пользователем услуг N-уровня является уровень N+1, так что N-БДС совпадает с (N+1)-БДП (рис. 1.17). N-объект формирует N-БДП, объединяя N-БДС с N-УИП. Далее образованный БДП передается нижележащему (N-1)-уровню в поле “данные пользователя” соответствующего примитива (N-1)-службы совместно с требуемыми в этой ситуации дополнительными параметрами.

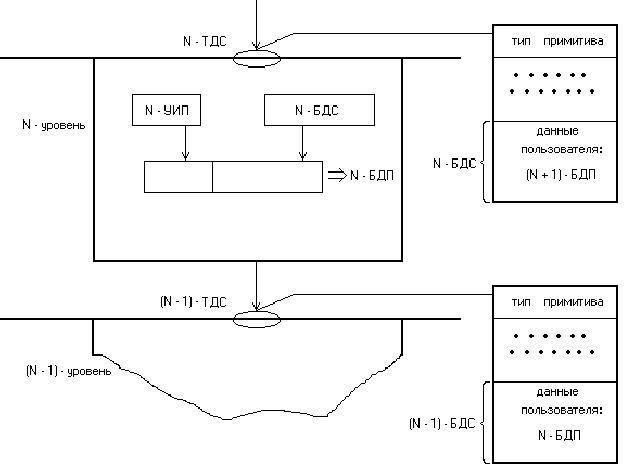


Рис. 1.17. Взаимодействие уровней

Интерпретируемые локально параметры примитивов служб нижних уровней модели без особых трудностей описываются в терминах таких традиционно понимаемых типов, как INTEGER, BOOLEAN и т.д. Более сложная ситуация возникает при пересылке управляющей информации протоколов из одной системы в другую в ходе организации взаимосвязи прикладных процессов, что обусловлено двумя обстоятельствами. Во-первых, один и тот же абстрактный тип может представляться в различных системах по-разному. Например, тот же INTEGER может быть представлен различным числом разрядов и/или неодинаковым расположением знакового разряда. Во-вторых, если на нижних уровнях относительная простота объектов позволяет обходиться при организации их взаимодействия обменом однозначно определенными простейшими битовыми полями, то на верхних уровнях этого уже недостаточно. Здесь начинает ощущаться потребность в значительно более мощных средствах описания, поскольку данные, передаваемые на этих уровнях, могут содержать структуры сложных типов, определяемых прикладными системами.

Оба эти обстоятельства подробнее будут рассмотрены в следующем разделе при описании верхних уровней ЭМВОС, ориентированных на приложения.

## Вопросы к разделу 1

1. Перечислите и раскройте смысл основных понятий эталонной модели ВОС.
2. В контексте эталонной модели ВОС кратко охарактеризуйте смысл и взаимоотношение терминов «реальная система», «реальная открытая система», «открытая система», «прикладной процесс» и «физическая среда».
3. Опишите уровневую организацию ВОС.
4. Почему было принято решение разделить всю совокупность функций обеспечения взаимосвязи на несколько иерархически упорядоченных групп?
5. Охарактеризуйте понятие «функции уровня».
6. Охарактеризуйте понятие «сервис уровня».
7. Опишите модель поставщика службы.
8. Опишите внешнее поведение и внутреннюю структуру уровня.
9. Охарактеризуйте методы формального описания протоколов и сервисов.
10. Составьте схему уровней ЭМВОС и вложения блоков данных протокола.
11. Составьте схему взаимодействия уровней.

# УРОВНИ ЭТАЛОННОЙ МОДЕЛИ ВОС

В разделе описаны уровни ЭМВОС. С учетом специализации основное внимание уделено верхним – начиная с транспортного – уровням, так что для самостоятельной углубленной проработки рекомендуются нижние три (сетезависимые) уровня модели. Более полное представление об общем прикладном сервисе также поможет составить дополнительное изучение удаленных операций и надежной передачи.

## Сетезависимые уровни

Семь уровней ЭМВОС часто принято условно группировать следующим образом. Уровни 1, 2 и 3, т.е. физический, канальный[[6]](#footnote-6) и сетевой, относят к сетезависимым, имея в виду, что детали их функционирования существенно меняются в зависимости от типа рассматриваемых сетей связи и их составляющих. Верхние три уровня – 5, 6 и 7, т.е. сеансовый, представительный[[7]](#footnote-7) и прикладной, относят к уровням, ориентированным на приложения, в силу их функциональной направленности. Четвертый уровень – транспортный – опирается на службы, обеспечиваемые сетевым уровнем, маскируя при этом от пользователей, т.е. объектов верхних уровней, особенности сетевого сервиса, и занимает промежуточное положение относительно обеих упомянутых групп-троек.

Начиная именно с транспортного уровня и далее вверх, мы будем более подробно рассматривать устройство функциональной иерархии ЭМВОС. Нижние три уровня будут рассмотрены очень кратко. Отчасти это объясняется комплементарным содержанием уже прослушанных к этому семестру курсов, в частности, по программно-техническим средствам ЭВМ и локальным сетям. Учитывается также и ориентация выпускаемых специалистов в целом.

**Физический уровень.** Обеспечивает механические, электрические, функциональные и процедурные средства установления, поддержания и разъединения физических соединений для прозрачной передачи битов между канальными объектами. Физическое соединение может включать промежуточные открытые системы, ретранслирующие биты на физическом уровне. Физические объекты связаны посредством физической среды. Физический канал – это путь в физической среде ВОС между физическими объектами вместе со средствами, необходимыми на физическом уровне для передачи по нему битов.

Физический уровень предоставляет канальному уровню следующие услуги: физические соединения; физические БДС; оконечные точки физических соединений; идентификацию физических каналов передачи данных; организацию передачи последовательностей битов; оповещение о неисправности физического уровня; определение параметров качества предоставляемых услуг.

Управление взаимным соединением физических каналов возлагается на канальный уровень.

Физический БДС при последовательной передаче состоит из одного бита и из n битов при параллельной передаче.

Физические соединения могут допускать дуплексную или полудуплексную передачу потока битов.

Физическое соединение может быть двухточечным или многоточечным.

Физический уровень доставляет биты в том же порядке, в каком они были ему предоставлены.

Основные функции, выполняемые внутри физического уровня: установление и разъединение физического соединения по запросу от канального уровня, включая функцию ретрансляции в случае взаимного соединения нескольких физических каналов; синхронная или асинхронная передача физических БДС; управление физическим уровнем.

При передаче данных по аналоговым каналам связи последовательность {Xi} битов на входе преобразуется в устройствах модуляции/демодуляции (модемах) в аналоговые сигналы Si(t), параметры которых согласованы с параметрами физической среды (параметрами физического канала). Принимаемые на выходе аналогового канала сигналы преобразуются в последовательность {Yi} битов, которая может отличаться от переданной последовательности из-за воздействия помехи n(t). Модемы вместе с физическим каналом образуют дискретный канал связи, рис.2.1.

В случае использования цифровых каналов связи преобразование последовательности битов в аналоговые сигналы не производится. При этом вместо модемов используют линейные контроллеры, обеспечивающие сопряжение ООД с физическим каналом.

Дискретный канал связи

Модем

Физическая среда

(аналоговый канал)

Модем

Помеха n(t)

{Xj}

{Yj}

Sj(t)

SJ(t)+n(t)

Рис. 2.1. Структура дискретного канала связи

Дискретный канал связи защищен от воздействия помех только потенциальной помехоустойчивостью передаваемых сигналов – аналоговых или дискретных. Таким образом, поскольку на физическом уровне не решается задача исправления искаженных битов, его следует считать ненадежной системой передачи.

Частота появления ошибок, которые могут возникать в результате таких искажений – одна из характеристик качества услуг, предоставляемых физическим уровнем. К другим характеристикам относятся: доступность услуги; скорость передачи; транзитная задержка.

**Канальный уровень.** Обеспечивает функциональные и процедурные средства установления, поддержания и разъединения канальных соединений между сетевым объектами, а также средства передачи канальных БДС в режимах с установлением и без установления соединения. В первом случае канальное соединение использует одно или несколько физических соединений. Канальный уровень с некоторой вероятностью обнаруживает и, возможно, исправляет ошибки, возникающие на физическом уровне.

Услуги, предоставляемые канальным уровнем, включают в себя: устанавливаемые и разъединяемые динамически канальные соединения; канальные БДС, размер которых может быть ограничен в зависимости от уровня ошибок на физическом уровне и возможностей канального уровня по их обнаружению; идентификаторы оконечных точек канального соединения; упорядоченная доставка канальных БДС; уведомление об ошибках; управление потоком; параметризация качества обслуживания.

С точки зрения пользователей, т.е. сетевых объектов, сервис канального уровня позволяет обеспечить следующие функции:

* независимость от используемых физических средств передачи (пользователи освобождаются от проблем, связанных с конфигурацией физического соединения или его техническими и процедурными характеристиками, например пользователь не знает, какой способ передачи – дуплексный или полудуплексный – используется);
* прозрачную передачу данных (пользователь может передавать данные с любым содержанием, форматом или кодировкой; канальный уровень не интерпретирует эти данные, т.е. доставляет их прозрачно);
* надежный обмен данными (большое число ситуаций (не все), связанных с потерей, переупорядочиванием или искажением данных, обрабатываются без вмешательства пользователей, что означает повышение вероятности безошибочной передачи данных);
* установление соединения по требуемому адресу (если на канальном уровне используется многоточечная конфигурация, т.е. конфигурация, когда есть пользователи с разными адресами, то пользователю дается возможность указать необходимый канальный адрес (т.е. адрес канальной ТДС));
* выбор параметров качества обслуживания (параметры качества сервиса канального соединения разделяются на три группы: параметры, согласуемые во время установления соединения; параметры, значения которых выбираются без согласования с партнером; параметры, значения которых не выбираются, но сообщаются пользователям).

К первой группе принадлежат параметры пропускной способности и транзитной задержки, ко второй – параметры защиты соединения и приоритета, а третью группу образуют коэффициент необнаруженных ошибок и живучесть соединения.

Согласование параметров первой группы не гарантирует сохранения их значений в течение всего времени существования соединения. При ухудшении параметров канальный уровень не сообщает об этом пользователям и не прерывает соединения.

Служба при работе в режиме “без соединения” связана с передачей отдельных независимых К-БДП. В примитивах такой службы передается каждый из используемых в этом режиме показателей качества, именно: транзитная задержка, защита, коэффициент необнаруженных ошибок.

Функциональная поддержка канального сервиса включает в себя установление и разъединение канальных соединений; расщепление канального соединения; синхронизацию; упорядочение; обнаружение ошибок; исправление ошибок; управление потоком; идентификацию канальных объектов и обмен параметрами; управление взаимным соединением физических каналов; управление канальным уровнем.

Как и для физического уровня, стандартизация проце­дур/элементов процедур канального уровня началась задолго до появления ЭМВОС. В начале 70-х годов был разработан целый ряд стандартов, регламентирующих процедуры так называемого основного режима; спустя примерно десятилетие – процедуры, называемые высокоуровневыми процедурами управления каналом. Дальнейшее развитие канальных стандартов происходит в направлении разработки многоканальных процедур и процедур для ЛВС.

Многоканальные процедуры используются для повышения надежности передачи и пропускной способности соединения за счет параллельной работы по нескольким физическим соединениям. Каждое физическое соединение используется для организации подканала, а многоканальная процедура собирает эти подканалы с различными, вообще говоря, характеристиками в одно соединение, предоставляемое объектам сетевого уровня.

Специфика протоколов ЛВС заключается в том, что функции канального уровня здесь расширены за счет добавления процедур управления доступом к среде.

**Сетевой уровень.** Обеспечивает средства установления, поддержания и разъединения сетевых соединений между открытыми системами, содержащими взаимодействующие прикладные объекты, а также средства для обмена сетевыми БДС между транспортными объектами по сетевым соединениям; функциональные и процедурные средства для передачи данных без установления соединения между транспортными объектами; независимость транспортных объектов от маршрутизации и коммутации (ретрансляции), выполняемых на сетевом уровне и связанных с передачей как с установлением, так и без установления сетевого соединения.

Основная услуга сетевого уровня – прозрачная передача данных между транспортными объектами. Сетевой уровень выполняет функции, необходимые для маскирования на своей границе с транспортным уровнем проявлений различий между разными передающими средами и подсетями. Однако в смысле качества обслуживания такая независимость при этом не обеспечивается. Качество обслуживания согласовывается между транспортными объектами и поставщиком сетевой службы в момент установления соединения.

Подсеть – автономный набор из одной или нескольких промежуточных систем, выполняющий функцию ретрансляции, через который оконечные (абонентские) системы могут устанавливать сетевые соединения.

Услуги сетевого уровня – сетевые адреса, используемые для идентификации транспортных объектов; сетевые двухточечные соединения; идентификаторы оконечных точек сетевого соединения; прозрачная передача сетевых БДС любого размера; параметризация качества обслуживания; уведомление об ошибках; упорядоченная доставка сетевых БДС; управление потоком; передача срочных сетевых БДС ограниченного размера; повторная установка; разъединение; подтверждение приема.

Некоторые из этих услуг являются необязательными, т.е. пользователь должен их запрашивать, а поставщик сетевой службы может их или обеспечивать, или нет.

Качество сервиса сетевого уровня определяется качеством отдельных услуг нижележащих подуровней. В фазе передачи данных используются следующие параметры: пропускная способность; транзитная задержка; коэффициент необнаруженных ошибок; живучесть; вероятность отказа. Помимо последнего параметра, специфичного для сетевого уровня в сравнении с уровнем канальным, здесь для фазы передачи данных имеется еще один “новый” параметр – наибольшая приемлемая стоимость соединения. Этот параметр не передается партнеру и имеет, таким образом, влияние на выбор лишь локальных средств.

В сравнении с канальным уровнем, сетевой уровень в части параметров качества сервиса пополнен также характеристиками фаз установления и разъединения соединения. Это, соответственно, задержки установления и разъединения, а так же вероятности неустановления и неразъединения соединений.

Передача Ст-БДС в режиме “без соединения” означает, что каждый блок данных передается во время единичного взаимодействия с поставщиком сервиса. При этом между блоками данных отсутствуют какие-либо зависимости/связи. Последовательность блоков, передаваемых один за другим в один и тот же пункт назначения, необязательно будет доставлена в том же порядке. Более того, не требуется, чтобы поставщик сервиса сообщал о недостаче данных или восстанавливал потерянные данные. В параметры качества сервиса в этом случае входят: транзитная задержка; защита; параметры стоимости; коэффициент необнаруженных ошибок; приоритет.

В целом, дополнительные параметры качества отражают специфику сетевого уровня и позволяют более точно настраивать протоколы транспортного уровня.

Сетевой уровень обеспечивает поддержку сетевых соединений для различных типов сетей – от простых двухточечных сетей до объединения подсетей с различными характеристиками. Сюда относится случай, когда несколько подсетей используются последовательно или параллельно.

На сетевом уровне выполняются следующие функции: маршрутизация и ретрансляция; организация сетевых соединений; мультиплексирование сетевых соединений на канальное соединение; сегментирование и блокирование; обнаружение ошибок; исправление ошибок; упорядочение; управление потоком; передача срочных данных; возврат в исходное состояние; выбор службы; управление сетевым уровнем.

Функция выбора службы может понадобиться для того, чтобы обеспечить одинаковую службу в различных оконечных точках сетевого соединения при использовании подсетей с различным качеством обслуживания.

Если используется объединение нескольких подсетей, то для выполнения функций маршрутизации и ретрансляции сетевой уровень может делиться на подуровни, причем функции маршрутизации и ретрансляции в подсетях и межсетевой маршрутизации могут выполняться независимо.

В целом заметим, что стандартизованы и хорошо известны лишь те протоколы сетевого уровня, которые реализуются в абонентских системах и шлюзах, соединяющих различные подсети: протоколы доступа к подсетям и межсетевые протоколы. Внутрисетевые протоколы, определяющие правила информационного взаимодействия между узлами коммутации, не стандартизованы, а используемые в различных сетях внутрисетевые протоколы зачастую существенно различаются. В то же время продолжается процесс интенсивной разработки новых стандартов, относящихся к сетевому уровню, и дополнений к уже имеющимся. Ситуация объясняется тем, что, с одной стороны, стандарты сетевого уровня создавались в те времена, когда архитектура ВОС не была полностью проработана, с другой – в стандартизации сетевого уровня заинтересовано значительное число организаций, представляющих и производителей сетевых средств, и пользователей сетевых средств, и администрации сетей передачи данных.

## Транспортный уровень

В соответствии с ЭМВОС транспортный уровень выполняет все необходимые процедуры для обеспечения надежной и эффективной прозрачной передачи данных из конца в конец от одного пользователя (сеансового объекта) до другого. Таким образом, все протоколы, определенные на транспортном уровне, функционируют в среде ВОС только между оконечными открытыми системами.

На транспортном уровне не выполняются функции маршрутизации и ретрансляции, так как сетевая служба обеспечивает прозрачную передачу данных между транспортными объектами даже при использовании нескольких промежуточных подсетей.

Транспортный уровень скрывает от пользователей особенности сетевого сервиса. Тип предоставляемого, т.е. транспортного, сервиса может быть не связан с типом потребляемого, т.е. сетевого сервиса. Так, транспортный сервис с соединением может быть представлен над сетевым сервисом любого типа – как с соединением, так и без соединения. Равным образом транспортный сервис без соединения поддерживается протоколом, который может использовать оба типа сетевого сервиса за счет выбора одного из режимов передачи над сетевым сервисом – без соединения и с соединением. Интересно заметить, что для последнего случая в имеющемся стандарте на протокол наличие сетевого соединения не гарантирует отсутствие потери данных даже тогда, когда сетевой уровень работает без сбоев.

При предоставлении сервиса с соединением транспортные со­единения устанавливаются между сеансовыми объектами, иденти­фицируемыми транспортными адресами. Качество обслуживания в транспортном соединении согласовывается между сеансовыми объектами и транспортной службой. В процессе установления транспортного соединения выбирается один из имеющихся классов транспортной службы, характеризуемых различными наборами параметров качества обслуживания: пропускной способности, транзитной задержки, времени установления соединения и т.д.

Между одной и той же парой транспортных адресов может быть установлено несколько транспортных соединений. Для их различе­ния сеансовые объекты используют идентификаторы оконечных точек транспортного соединения.

Если согласованное качество обслуживания не может быть обеспечено и попытки его восстановления терпят неудачу, то транспортное соединение разъединяется и об этом сообщается сеансовым объектам.

В фазе установления соединения могут выполняться следующие функции: выбор сетевого соединения, наиболее полно удовлетворяющего требованиям сеансового объекта с учетом стоимости и качества обслуживания; решение о целесообразности мультиплексирования или расщепления транспортного соединения с целью оптимизации использования сетевых соединений; выбор оптимального размера транспортного БДП; выбор функций, которые будут задействованы в фазе передачи данных; отображение транспортных адресов в сетевые; обеспечение идентификации различных транспортных соединений между одной и той же парой транспортных ТДС; передача данных.

В фазе передачи данных осуществляется доведение транспортных БДС до сеансовых объектов-получателей по транспортному соединению передачей транспортных БДП. При этом могут быть задействованы следующие функции, использование каждой из которых согласуется в фазе установления соединения:

* упорядочение; сегментирование, блокирование и сцепление;
* мультиплексирование или расщепление; управление потоком;
* обнаружение ошибок; исправление ошибок;
* передача срочных данных; разграничение транспортных БДС;
* идентификация транспортных соединений.
* В фазе разъединения соединения могут выполняться функции:
* оповещения о причине разъединения;
* идентификации разъединяемого транспортного соединения;
* передачи данных.

При установлении транспортного соединения используются следующие примитивы (в скобках – список параметров):

T-CONNECTrequest: (вызываемый адрес, вызывающий адрес, использование срочных данных, параметры качества, данные пользователя);

T-CONNECTindication: (вызываемый адрес, вызывающий адрес, использование срочных данных, параметры качества, данные пользователя);

T-CONNECTresponse: (адрес ответчика, использование срочных данных, параметры качества, данные пользователя);

T-CONNECTconfirmation: (адрес ответчика, использование срочных данных, параметры качества, данные пользователя).

Указанные в параметрах адреса – это адреса Т-ТДС. Адрес ответчика может использоваться тогда, когда на сетевом уровне есть услуга по переадресации. Такая услуга предусмотрена в сетях коммутации пакетов общего пользования. Адрес ответчика, очевидно, может отличаться от вызываемого адреса.

Использование срочных данных указывается в явном виде, но только в том случае, если эта услуга входит в предоставляемый реальной системой сервис. Она может не обеспечиваться даже при использовании стандартных транспортных протоколов.

Данные пользователя передаются прозрачно. Здесь максимальная длина этого поля – 32 октета.

Таблица соответствия параметров примитивов (табл.2.1) не отражает, вообще говоря, всех аспектов согласования.

Установление соединения связано с согласованием параметров качества и порядка использования срочных данных.

Для разъединения уже установленного соединения, а также при отказе установить соединение используется пара примитивов T-DISCONNECTrequest и -indication с соответствующими параметрами. Такая же пара (T-DATA-) с единственным типом параметра – “данные пользователя” – используется при передаче данных по соединению. Длина Т-БДС не ограничена, так как на транспортном уровне есть функция разбиения Т-БДС на последовательность Т-БДП.

Таблица 2.1. Соответствие параметров примитивов T-CONNECT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | Сервисные примитивы | | | |
| request | indication | response | confirmation |
| Вызываемый адрес | O | (=) | - | - |
| Вызывающий адрес | O | (=) | - | - |
| Адрес ответчика | O | - | О | (=) |
| Использование срочных данных | O | (=) | О | (=) |
| Параметры качества | O | О | О | (=) |
| Данные пользователя | П | (=) | П | (=) |

При передаче срочных данных срочный Т-БДС имеет длину до 16 октетов включительно и передается в сервисных примитивах T-EXPEDITED-DATArequest: (данные пользователя) и -indication: (данные пользователя). Срочные Т-БДС имеют собственный механизм управления потоком, например, следующий Т-БДС не будет передаваться, пока не завершиться передача предыдущего Т-БДС.

Продвижение срочных и нормальных Т-БДС должно обеспечиваться так, чтобы срочные Т-БДС не приходили партнеру позже нормальных. В ряде случаев, например при пропаже срочных данных и их повторной передаче, для такого обеспечения нужны дополнительные механизмы. Один из таких механизмов может быть основан на прекращении передачи всех Т-БДС до тех пор, пока на транспортном уровне не будет передан и подтвержден срочный Т-БДС. Задержка передачи на интервал t здесь определяется транзитной задержкой (рис.2.2). С точки зрения пользователя это означает, что передача срочных данных может значительно ухудшить характеристики потока нормальных данных.

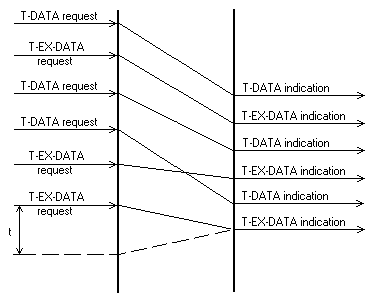


Рис.2.2. Соотношение нормальных и срочных данных

Рис. 2.3 и табл. 2.2 иллюстрируют использование в целях описания соотношений примитивов на одном конце соединения таблиц следования и/или диаграмм состояний-переходов табл.2.2 и рис.2.3 для рассматриваемого транспортного сервиса.

Связь параметров качества сервиса транспортного и сетевого уровней показана на рис.2.4. Параметр приоритета транспортного соединения (ТС) связан с аналогичным параметром сетевого соединения (СтС) косвенно, что обозначено пунктиром. Характер связи обусловлен различием используемых механизмов. Например, если ТС мультиплексируются в одно СтС, то ухудшение параметров СтС одинаково сказывается на всех ТС. Наоборот, ухудшение характеристик одного ТС может быть не связано с приоритетом СтС и определяется исключительно его собственным приоритетом.

Таблица 2.2. Допустимое (+) и недопустимое (-) следование (упорядочивание) примитивов транспортного сервиса

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Может следовать примитив | За примитивом | | | | | | | | | | |
| 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | - | - | - | | - | - | - | - | - | - | - |
| 2 | - | - | - | | - | - | - | - | - | - | - |
| 3 | - | + | - | | - | - | - | - | - | - | - |
| 4 | + | - | - | | - | - | - | - | - | - | - |
| 5 | - | - | + | | + | + | + | + | + | - | - |
| 6 | - | - | + | | + | + | + | + | + | - | - |
| 7 | - | - | + | | + | + | + | + | + | - | - |
| 8 | - | - | + | | + | + | + | + | + | - | - |
| 9 | + | + | + | | + | + | + | + | + | - | - |
| 10 | + | + | + | | + | + | + | + | + | - | - |
| Номер примитива | | | | Примитив | | | | | | | |
| 1 | | | | Т-СОЕДИНЕНИЕзапрос | | | | | | | |
| 2 | | | | Т-СОЕДИНЕНИЕиндикация | | | | | | | |
| 3 | | | | Т-СОЕДИНЕНИЕответ | | | | | | | |
| 4 | | | | Т-СОЕДИНЕНИЕподтверждение | | | | | | | |
| 5 | | | | Т-ДАННЫЕзапрос | | | | | | | |
| 6 | | | | Т-ДАННЫЕиндикация | | | | | | | |
| 7 | | | | Т-СРОЧНЫЕ-ДАННЫЕзапрос | | | | | | | |
| 8 | | | | Т-СРОЧНЫЕ-ДАННЫЕиндикация | | | | | | | |
| 9 | | | | Т-РАЗЪЕДИНЕНИЕзапрос | | | | | | | |
| 10 | | | | Т-РАЗЪЕДИНЕНИЕиндикация | | | | | | | |

Транспортный сервис с соединением обеспечивается протоколом “с соединением” и приложениями к нему. Этот протокол фактически содержит пять различных протоколов, именуемых классами и ориентированных на разный сетевой сервис (с установлением соединения). Определяются три типа такого сервиса:

* тип А – служба с приемлемыми коэффициентами необнаруживаемых и обнаруживаемых ошибок;
* тип B – служба с приемлемым коэффициентом необнаруживаемых ошибок, но с неприемлемым коэффициентом обнаруживаемых ошибок;
* тип C – служба с неприемлемым коэффициентом необнаруживаемых ошибок.



Рис. 2.3. Диаграмма состояний переходов

Каждый класс транспортного соединения имеет различную функциональность (функциональную наполненность). Это связано с необходимостью обеспечивать, с одной стороны, надежный транспортный сервис над сетевым сервисом с ухудшенными характеристиками, а с другой – устранять избыточные транспортные функции при использовании качественного сетевого сервиса. Соотношение используемых классов транспортного сервиса и типов сетевого сервиса приведено на рис.2.5.На нем условно показано, что чем больше функциональность класса, тем хуже может быть качество используемого сервиса при сохранении неизменным транспортного сервиса. Наличие классов 0, 2 для сервиса типа А и 1, 3 – для B связано с оптимизацией использования сетевых ресурсов: классы 2 и 3 отличаются от классов 0 и 1 процедурами мультиплексирования[[8]](#footnote-8) транспортных протоколов в сетевые. Функция мультиплексирования позволяет более рационально использовать сетевые ресурсы, а также снизить затраты на используемые сетевые соединения, особенно в тех случаях, когда стоимость установления нового соединения велика по сравнению со стоимостью передачи объема данных, необходимых для установления транспортного соединения.

Рис. 2.4. Связь параметров качества сервиса транспортного и сетевого уровней

Задержка установления СтС

Задержка установки ТС

Вероятность неустановления СтС

Вероятность неустановления ТС

Пропускная способность СтС

Пропускная способность ТС

Транзитная задержка

Транзитная задержка

Коэффициент необнаруженных ошибок

Коэффициент необнаруженных ошибок

Живучесть СтС

Живучесть ТС

Вероятность отказа

Вероятность отказа

Задержка разъединения СтС

Задержка разъединения ТС

Вероятность неразъединения

Вероятность неразъединения

Защита СтС

Защита ТС

Приоритет СтС

Приоритет ТС

Рис. 2.5. Соотношение классов транспортного протокола и типов сетевого сервиса

**0**

**2**

**1**

**3**

**4**

**A**

B

**C**

Транспортный сервис

Функциональность классов

Сервис сетевого уровня

Получив определенную информацию либо сверху (от пользователя), либо снизу (от поставщика сетевого сервиса), Т-объект генерирует, как это описывалось ранее, соответствующий Т-БДП.

С рассматриваемым транспортным протоколом связано 10 различный типов Т-БДП (изучаются потребности расширения этого множества), среди которых – “запрос на соединение”, “данные”, “отвергнуто”, “ошибка” и т.п. Каждый Т-БДП состоит из заголовка (Т-УИП) и поля данных пользователя, если оно имеется (рис.2.6). Фиксированная часть содержит обычно используемый набор параметров, и, в частности, код типа Т-БДП. Переменная часть используется для определения реже встречающихся параметров. ИД – индикатор длины заголовка. Код типа Т-БДП содержится во втором октете заголовка и используется для определения структуры остальной части заголовка.

ИД

Фиксированная часть

Переменная часть

Поле данных пользователя

Номер

октета 1 2………………….n n+1…………...p p+1……………..

Заголовок

Рис.2.6. Структура Т-БДП

Далее воспользуемся упоминавшимися конечными автоматами для иллюстрации протокольной спецификации фрагмента фазы установления транспортного соединения. В табл.2.3 приведены части списков имен элементов, множеств входных и выходных событий, состояний автомата, разрешающих условий (предикатов).

Предикату соответствует булевская переменная, зависящая от комбинации значений параметров, связанных со входным событием, и текущего состояния одной или нескольких автоматных переменных. Если предикатное условие не выполнено, а альтернатива не определена, то говорят, что имела место ошибка протокола. В этом случае выдается заранее определенное выходное событие, и автомат переходит в соответствующее состояние.

Таблица 2.3. Фрагменты списков имен элементов множеств входных событий, состояний автомата, выходных событий и предикатов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Интерфейс автомата | | Смысл/значение | |
| ТСДНзпр | ТСл-пользователь | | Получен Т-СОЕДИНЕНИЕзапрос | |
| ТСДНотв | ТСл-пользователь | | Получен Т-СОЕДИНЕНИЕответ | |
| СтСДНпдтв | Ст-поставщик | | Получен Ст-СОЕДИНЕНИЕподтверждение | |
| ЗС | Ст-поставщик | | Получен Т-БДП «запрос на соединение» | |
| ПС | Ст-поставщик | | Получен Т-БДП «подтверждение соединения» | |
| … | … | | … | |
|  |  | |  | |
| Имя | | Смысл/значение | | |
| Закрыто | | Транспортное соединение закрыто | | |
| ОЖСтС | | Ожидание установления сетевого соединения | | |
| ОЖПС | | Ожидание Т-БДП «подтверждение соединения» | | |
| ОТКРЫТО | | Транспортное соединение открыто и готово к передаче данных | | |
| ОЖТОТВ | | Ожидание поступления Т-СОЕДИНЕНИЕответ от ТСл-пользователя | | |
| … | | … | | |
|  |  | | | |
| Имя | Интерфейс автомата | | | Смысл/значение |
| ТСДНинд | ТСл-пользователь | | | Послан Т-СОЕДИНЕНИЕиндикация |
| ТСДНпдтв | ТСл-пользователь | | | Послан Т-СОЕДИНЕНИЕподтверждение |
| ТРЗДинд | ТСл-пользователь | | | Послан Т-РАЗЪЕДИНЕНИЕиндикация |
| СтСДНзпр | Ст-поставщик | | | Послан Ст-СОЕДИНЕНИЕзапрос |
| ПС | Ст-поставщик | | | Послан Т-БДП «подтверждение соединения» |
| ЗС | Ст-поставщик | | | Послан Т-БДП «запрос на соединение» |
| ЗР | Ст-поставщик | | | Послан Т-БДП «запрос на разъединение» |
| СтРЗДзпр | Ст-поставщик | | | Послан Ст-РАЗЪЕДИНЕНИЕзапрос |
| … | … | | | … |
|  |  | | | |
| Имя | Смысл/значение | | | |
| Закрыто | Транспортное соединение закрыто | | | |
| ОЖСтС | Ожидание установления сетевого соединения | | | |
| ОЖПС | Ожидание Т-БДП «подтверждение соединения» | | | |
| ОТКРЫТО | Транспортное соединение открыто и готово к передаче данных | | | |
| ОЖТОТВ | Ожидание поступления Т-СОЕДИНЕНИЕответ от ТСл-пользователя | | | |
| … | … | | | |

Использование и интерпретация целого ряда полей заголовка обусловлено не только типом Т-БДП, но и содержимым других определенных полей, в частности выбором того или иного класса. Например, в классе 4 в целях явного управления потоком используется механизм окна, и в Т-БДП, скажем, “подтверждение соединения” соответствующей интерпретации подвергается поле параметра “кредит”.

Таблица 2.4. Фрагмент таблицы состояний-переходов автомата отработки фазы установления транспортного соединения

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Событие | Состояние | | | | | |
| ЗАКРЫТО | ОЖТОТВ | Р4: ОЖСтС | ОЖПС | ОТКРЫТО | … |
| ТСДНзпр | 1 | 0 | 0 | 0 | 0 | … |
| ТСДНотв | 0 | 2 | 0 | 0 | 0 | … |
| СтСДНпдтв | 0 | 0 | 3 | 0 | 0 | … |
| ЗС | 4 | 0 | 0 | 0 | 0 | … |
| ПС | 0 | 0 | 0 | 5 | 0 | … |
| ... | … | … | … | … | … | … |
| 0=ТРЗДзпр, СтРЗДзпр, ЗАКРЫТО (ошибочное условие)  1=Р0: ТРЗДинд, ЗАКРЫТО; Р2:СтСДНзпр, ОЖСтС; Р3: ЗС, ОЖПС  2=ПС, ОТКРЫТО  3=ЗС, ОЖПС  4=Р1:ЗР, ЗАКРЫТО; NOT Р1:ТСДНинд, ОЖТОТВ  5=NOT Р5: ТСДНпдтв, ОТКРЫТО; Р5:ТРЗДинд, СтРЗДзпр, ЗАКРЫТО | | | | | | |

В табл.2.4 дан фрагмент соответствующей таблицы состояний-переходов автомата. Здесь он выписан в виде таблицы ссылок на список комбинаций состояний-событий. Некоторые из этих комбинаций обусловлены предикатами. В имеющемся фрагменте ни в одной из ситуаций никаких специальных действий, например запуска или остановки таймера, не производится, так что ссылочный список таких действий здесь пуст.

## Сеансовый уровень

Три верхних уровня ЭМВОС часто называют уровнями, ориентированными на приложения. В отношении самого нижнего из   
них – сеансового – такая ориентация проявляется, в частности, в том, что его функции сильно связаны с его сервисом, так как собственные, т.е. не инициированные с верхнего уровня, действия на сеансовом уровне практически отсутствуют. В целом, сеансовый уровень (с помощью служб, обеспечиваемых уровнем представления) обеспечивает прикладным объектам следующие средства равноправного, синхронизированного, структурированного взаимодействия:

* установления сеансового соединения, синхронизированного обмена данными, упорядоченного и безусловного завершения сеансового соединения;
* согласования использования маркеров (см. далее) обмена данными, синхронизации и завершения взаимодействия, а также фиксации маркеров на одной из взаимодействующих сторон;
* установления точек синхронизации внутри диалога;
* выполнения ресинхронизации сеансового соединения к согласованной прикладными объектами точке синхронизации;
* прерывания диалога и его возобновления с заранее организованной точки синхронизации.

Пользователям сеансового уровня может быть предоставлена также служба без установления соединения. Ее поддерживает сеансовый протокол, опирающийся, в свою очередь, на транспортную службу без установления соединения. Свойства такой сеансовой службы аналогичны свойствам сетевой и транспортной служб без установления соединения. Она предоставляет своим пользователям средства для прозрачной передачи разграниченных Сн-БДС от одной ТДС до другой за один акт доступа, без предварительной установки и последующего освобождения сеансового соединения. При этом с каждой такой передачей связано определенное качество обслуживания, которое согласовывается между поставщиком сеансовой службы и пользователем-отправителем при инициации сеансовой службы без установления соединения.

На сеансовом уровне вводится ряд понятий, которые широко используются при описании объектов уровня представлений и прикладного уровня. К таким понятиям относятся маркер, точка синхронизации, диалоговый элемент, активность, ресинхронизация, переговоры, функциональная группа и др.

Маркер (жетон, признак, полномочие[[9]](#footnote-9), token) – атрибут сеансового соединения, который динамически назначается одному из взаимодействующих пользователей сеансовой службы (СнСл-пользователей). Владелец маркера имеет исключительное право инициировать выполнение услуги, контролируемой данным маркером. На сеансовом соединении могут быть использованы четыре типа маркера: DK – маркер данных, TR – маркер завершения (освобождения), MI – маркер вспомогательной синхронизации, MA – маркер главной синхронизации. Каждый маркер на сеансовом соединении всегда находится в одном из двух состояний: доступен и недоступен. Маркер доступен тогда, когда СнСл-пользователи в ходе установления сеансового соединения согласовали его применение в процессе предстоящего взаимодействия. В этом случае маркер назначается одной из взаимодействующих сторон, а поставщик сеансовой службы гарантирует отсутствие раздвоения маркера в любой момент существования сеансового соединения. Маркер, по которому не проводилось согласование применения, недоступен и не может быть назначен ни одной из сторон.

Точки синхронизации могут размещаться пользователями сеансовой службы в потоке данных. Эти точки идентифицируются последовательными монотонно возрастающими номерами, которые обеспечиваются поставщиком сеансовой службы. Взаимодействующие прикладные объекты могут связывать с точками синхронизации, а точнее, с их номерами, любую семантику, определяемую приложением.

Фиксацию точек можно выполнить двумя способами, используя технику вспомогательной (малой, minor) или главной (большой, major) синхронизации. Главная точка синхронизации считается зафиксированной после ее явного подтверждения, в то время как вспомогательная точка синхронизации для своей фиксации не требует явного подтверждения и может вообще не подтверждаться. Поставщик сеансовой службы нумерует все (любые) точки последовательно.

Диалоговый элемент (диалоговый блок, единица диалога, dialogue unit) – часть потока данных сеансового уровня, ограниченная двумя последовательными главными точками синхронизации. Это логически выделенное замкнутое взаимодействие, которое можно отработать независимо от предшествующих и последующих процессов обмена данными. Вспомогательные точки синхронизации используются для структуризации обмена данными в рамках диалогового элемента.

Активность (период деятельности, деятельность, activity) – логически завершенный фрагмент работы пользователя службы сеансового уровня, выделенный в общем потоке данных этого уровня. Активность образуется последовательностью диалоговых элементов. Единовременно в рамках сеансового соединения может существовать лишь одна активность. Допускается последовательная обработка активностей на одном сеансовом соединении. В то же время активность может охватывать несколько последовательно устанавливаемых сеансовых соединений.

Ресинхронизация – средство принудительной координации взаимодействующих сторон в рамках сеансового соединения. Может быть инициирована любым СнСл-пользователем в любой момент существования соединения. При этом появляется возможность перераспределить маркеры и согласовать новое значение последовательного номера точки синхронизации для продолжения взаимодействия. В ходе ресинхронизации происходит сброс всех данных, находящихся в это время на стадии передачи по сеансовому соединению. Поставщик сеансовой службы обеспечивает три режима согласования нового номера точки синхронизации: отказ (аннулирование, abandon), рестарт, установка (set).

**Фазы и услуги сеансового уровня.** Сеансовый сервис охватывает три фазы – установление сеансового соединения, передача данных, завершение сеансового соединения. Каждая фаза характеризуется набором услуг, обеспечиваемых сеансовым протоколом.

В фазе установления соединения предусмотрена одна услуга, S-CONNECT, которая позволяет согласовать его параметры, распределить маркеры, выбрать начальный номер точки синхронизации.

В фазе передачи данных используется четыре разновидности услуги: S-DATA, S-EXPEDITED DATA, S-TYPED DATA и S-CAPABILITY DATA, каждая из которых используется для передачи своего типа данных, отличающихся видами приоритета и/или наборами условий/ограничений, учитываемых в процессе передачи.

Управлять расположением маркеров в рамках сеансового соединения, выполнить фиксацию точки синхронизации и ресинхронизации, оповестить об ошибках или неожиданных ситуациях, обеспечить управление активностью позволяют тринадцать других услуг.

Фаза завершения сеансового соединения характеризуется тремя другими услугами: S-RELEASE – упорядоченное завершение (может быть использован маркер завершения TR), S-P-ABORT и S-U-ABORT – безусловное завершение.

Некоторые услуги могут инициироваться и поставщиком (Пс, provider, P), и пользователем (Пл, user, U), например услуги безусловного завершения сеансового соединения – S-P-ABORT и S-U-ABORT соответственно.

Напомним еще раз, что ряд услуг опирается на использование маркеров – управляется или контролируется соответствующими маркерами.

**Функциональные группы (блоки) и стандартные подмножества сеансового сервиса.** Функциональные группы – объединения логически связанных услуг. Вводятся с целью согласования требований пользователей в ходе фазы установления сеансового соединения. Свойство маркерного управления определенной услуги передается функциональной группе, содержащей эту услугу. Если какая-либо услуга, входящая в группу, контролируется маркером, то в функциональную группу включаются услуги управления расстановкой маркеров.

Имеется двенадцать функциональных групп – от базовой до группы управления активностью. Базовая группа объединяет основные сеансовые услуги, позволяющие установить сеансовое соединение, осуществить передачу нормальных блоков данных, завершить сеансовое соединение.

В стандарте вводится понятие сервисного подмножества или сервисного профиля сеансового соединения. Это комбинация базовой функциональной группы с другими группами, подчиняющаяся некоторым условиям.

Стандарт выделяет три сервисных подмножества:

* основное (базовое) комбинированное подмножество (три группы);
* основное (базовое) подмножество синхронизации (восемь групп);
* основное (базовое) подмножество активности – управления деятельностью (семь групп);

В основное комбинированное подмножество помимо базовой функциональной группы входят группы полудуплекса и дуплекса. Дев последние группы поддерживают соответствующие режимы передачи.

Пользовательские прикладные протоколы не ограничиваются в выборе требуемых им сервисных подмножеств тремя перечисленными основными подмножествами. Имеется возможность сформировать тот или иной сервисный профиль сеансового соединения в соответствии с функциональными потребностями приложений.

**Переговоры.** Осуществляются СнСл-пользователями в процессе установления сеансового соединения. При согласовании сервисного профиля сеансового соединения подразумевается, что всегда применяется базовая функциональная группа. Каждый СнСл-пользователь предлагает необходимые для взаимодействия функциональные группы. Группа выбирается лишь тогда, когда оба СнСл-пользователя включили ее в сервисное подмножество, и при этом она может быть обеспечена поставщиком сеансовой службы.

**Качество сеансового сервиса.** Определяет параметры сеансового соединения, которые касаются исключительно поставщика сеансовой службы. Параметры делятся на два типа. Параметры первого типа согласуются в ходе установления сеансового соединения. К ним относятся защита сеансового соединения, его приоритет, темп остаточных ошибок, полоса пропускания и задержка передачи для каждого направления, оптимизация передачи, расширенное управление.

Параметры второго типа не согласуются в фазе установления сеансового соединения, но их значение известно либо изначально, либо в результате проведения предварительных измерений. К ним относятся задержка установления сеансового соединения, вероятность отказа в установлении сеансового соединения, вероятность ошибки передачи, задержка завершения сеансового соединения, вероятность ошибки завершения сеансового соединения, живучесть сеансового соединения.

После установления соединения СнСл-пользователи не могут модифицировать выбранные параметры качества сеансового сервиса в период существования этого соединения. Поставщик сеансовой службы не информирует СнСл-пользователей о каких-либо изменениях параметров качества сеансового уровня.

Стандарт определяет правила согласования параметров качества первого типа.

Каждая сеансовая услуга представляется набором примитивов. Отработка услуги заключается в последовательном исполнении ее примитивов. Подтверждаемая услуга содержит четыре типа примитивов: запрос, индикация, ответ, подтверждение; неподтверждаемая – два: запрос, индикация (в особых случаях – только один – индикация).

Для каждой сеансовой услуги параметрический состав примитивов может быть сведен в свою таблицу соответствия. Подтверждаемая услуга S-CONNECT, например, имеет девять параметров – начиная от идентификатора сеансового соединения, адресов (идентификаторов ТДС) СнСл-пользователей, требований к сеансу и кончая начальной расстановкой маркеров и данными пользователя, позволяющими передавать уточняющую информацию в случае, скажем, неустановленного сеансового соединения.

Стандарт определяет правила, в соответствии с которыми СнСл-пользователи получают право инициировать услуги, контролируемые маркерами.

Услуга ресинхронизации не является контролируемой маркерами, поэтому взаимодействующие СнСл-пользователи, в общем случае, могут одновременно инициировать запросы ресинхронизации или инициация услуги ресинхронизации на одной стороне сеансового соединения может совпасть во времени с инициацией услуг прерывания или сброса активности на другой стороне. Поставщик сеансовой службы разбирается с этими столкновениями (коллизиями ресинхронизации), руководствуясь приоритетной схемой, которая определена стандартом. Основу этой схемы составляет иерархическая упорядоченность потенциально конфликтующих услуг. При столкновении более приоритетной услуги с услугой менее приоритетной первая становится победителем. В результате, например, может происходить следующее явление: СнСл-пользователь, инициировавший услугу ресинхронизации, сброса или прерывания активности и перешедший в состояние ожидания примитива подтверждения соответствующих услуг, будет получать индикации услуг – победителей столкновения.

Кроме учета подобных ситуаций, существуют правила разрешения конфликтов, возникающих при столкновении запросов ресинхронизации, имеющих одинаковые приоритеты.

На рис.2.7 проиллюстрировано развитие одного из многих вариантов коллизии ресинхронизации. К моменту начала рассмотрения на сеансовом соединении зафиксирован ряд точек синхронизации, последняя имеет номер Sn, равный i-1. Распределение маркеров приведено на рисунке. Выписанные на нем названия примитивов соответствуют услугам фазы передачи данных, перечисленным (частично в неявном виде) ранее.

В процессе отработки услуги S-CONNECT поставщик сеансовой службы отображает адреса сеансовых ТДС в соответствующие адреса транспортных ТДС, через которые сеансовые объекты взаимодействуют друг с другом, отображает параметры качества обслуживания на сеансовом соединении в параметры качества обслуживания на транспортном соединении, согласует максимальные размер транспортных БДС с поставщиком транспортной службы.

Поставщик сеансовой службы для обеспечения сеансового соединения использует транспортное соединение. В каждый момент времени существует однозначное соответствие между сеансовым и транспортным соединениями, каждому сеансовому соединению соответствует единственное транспортное соединение. Однако время их жизни может отличаться: одно транспортное соединение может поддерживать несколько последовательных сеансовых соединений, несколько последовательных транспортных соединений могут поддерживать одно сеансовое.

Заметим здесь же, что не существует никаких архитектурных ограничений на число параллельных сеансовых соединений между двумя СнСл-пользователями.

Транспортное соединение, об использовании которого шла речь, либо должно быть установлено путем инициации поставщиком сеансовой службы услуги T-CONNECT, либо захвачено им, если среди зарезервированных сеансовым объектом впрок имеется подходящее. При благополучном развитии ситуации после этого корреспондирующие сеансовые объекты обмениваются по этому транспортному соединению, пользуясь услугами T-DATA, данными, которые необходимы для установления сеансового соединения.

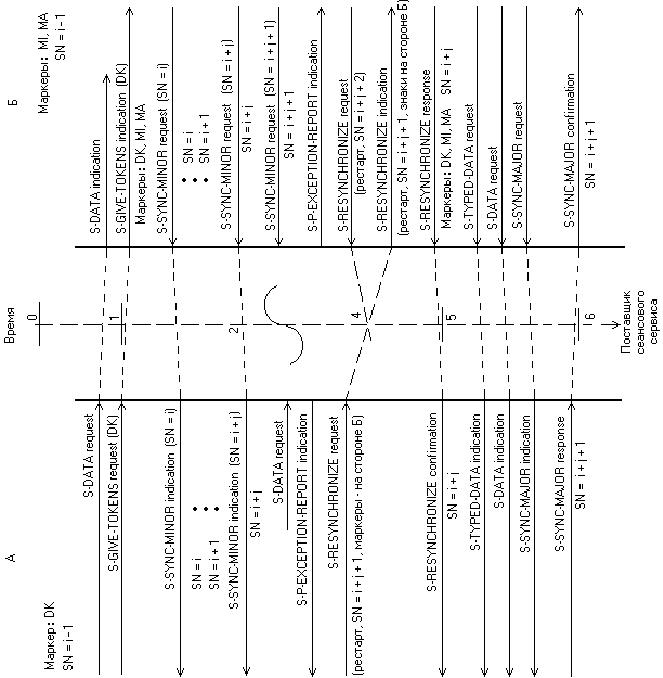


Рис. 2.7. Выход из столкновения запросов ресинхронизации

Таблица 2.5. Фрагменты списков имен элементов множеств входных событий, состояний автомата

|  |  |  |
| --- | --- | --- |
| Имя | Интерфейс автомата | Смысл/Значение |
| СнСДНзпр | СнСл-пользователь | Получен Сн-СОЕДИНЕНИЕзапрос |
| СнСДНотв(+) | СнСл-пользователь | Получен  Сн-СОЕДИНЕНИЕответ(принято) |
| СнСДНотв(-) | СнСл-пользователь | Получен  Сн-СОЕДИНЕНИЕответ(отвергнуто) |
| ТСДНинд | ТСл -поставщик | Получен Т-СОЕДИНЕНИЕиндикация |
| СтСДНпдтв | ТСл -поставщик | Получен  Т-СОЕДИНЕНИЕподтверждение |
| ЗС | ТСл -поставщик | Получен Сн-БДП «соединение» |
| ПС | ТСл -поставщик | Получен Сн-БДП «принято» |
| ПС | ТСл -поставщик | Получен Сн-БДП «отказано» |
| … | … | … |
| СнТМР | Таймер | Истек таймер СнТМР |
|  |  |  |
| Имя | Смысл/значение | |
| Сост 01 | Бездействие; нет транспортного соединения | |
| Сост 01В | Ожидание Т-СОЕДИНЕНИЕ подтверждения | |
| Сост 01С | Бездействие; транспортное соединение имеется | |
| Сост 02 | Ожидание Сн-БДП «принято» | |
| Сост 08 | Ожидание Сн-СОЕДИНЕНИЕответ | |
| Сост 016 | Ожидание Сн-СОЕДИНЕНИЕиндикация | |
| … | … | |
| Сост 713 | Передача данных | |

Воспользуемся здесь формализмом конечных автоматов для протокольной спецификации, как и в случае транспортного уровня, фрагмента фазы установления сеансового соединения. В иллюстративных целях была избрана фаза установления соединения для основного комбинированного подмножества. Ему соответствует набор из 11 типов Сн-БДП, формируемых Сн-объектом.

В табл.2.5 приведены части списков имен элементов множеств входных и выходных событий, состояний автомата, предикатов и специальных действий.

В табл.2.6 дан фрагмент соответствующей таблицы состояний-переходов автомата.

Таблица 2.6. Часть таблицы состояний-переходов автомата отработки фазы установления сеансового соединения

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Событие | Состояние | | | | | | | | | |
| Сост 01 | Сост 01В | | Сост 01С | Сост 02 | Сост 08 | Сост 016 | … | Сост 713 | |
| СнСДНзпр | 1 | 0 | | 2 | 0 | 0 | 0 |  |  | |
| СнСДНотв(+) | 0 | 0 | | 0 | 0 | 3 | 0 |  |  | |
| СнСДНотв(+) | 0 | 0 | | 0 | 0 | 4 | 0 |  |  | |
| ТСДНинд | 5 | 0 | | 0 | 0 | 0 | 0 |  |  | |
| ТСДНпдтв | 0 | 6 | | 0 | 0 | 0 | 0 |  |  | |
| СД | 0 | 0 | | 8 | 0 | 0 | 7 |  |  | |
| ПН | 0 | 0 | | 7 | 9 | 0 | 12 |  |  | |
| ОТ | 0 | 0 | | 7 | 10 | 0 | 12 |  |  | |
| … |  |  | |  |  |  |  |  |  | |
| СнТМР | 0 | 0 | | 0 | 0 | 0 | 11 |  |  | |
| 0=СнПсРинд, ПКР, Сост01  1=ТСДНзпр, [2], Сост 01В  2=Р1: СД, Сост 02  3=ПН, Сост 713  4=ОТ, [4], Сост 16 | | | 5=ТСДНотв, [1], Сост 01С  6=СД, Сост 02  7=ТРЗДзпр, [3], Сост 01  8=Р1: ТРЗДзпр, Сост 01  NOT Р1: СнСДНинд, Сост08 | | | | 9=СнСДНпдтв(+), Сост713  10=СнСДН пдтв(-),  ТРЗДзпр, Сост 01  11=ОТ, [4], Сост16  12=Сост 16 | | |

## Уровень представления

Оконечные системы (абоненты) вычислительных сетей весьма разнообразны и представлены устройствами различных типов – от простых символьно-ориентированных дисплеев до универсальных ЭВМ и систем, ориентированных на базы данных. В разных устройствах используется различное внутреннее представление хранимой информации. Для обеспечения их взаимодействия модель ВОС содержит шестой уровень, называемый уровнем представления (представительным уровнем).

В процессе своего функционирования объекты уровня представления используют протокол, который позволяет согласовать различия в синтаксисе данных с помощью преобразования местного внутреннего представления (локальный синтаксис) в синтаксис передачи. Синтаксис передачи вместе с правилами преобразования составляет образ конкретного соединения уровня представления. Согласование синтаксических различий должно выполняться таким образом, чтобы сохранить семантику передаваемой информации.

Представительный уровень в ходе согласования синтаксических различий имеет дело с двумя аспектами. Первый относится к описанию (заданию) синтаксиса со стороны прикладных объектов, второй – к тому, как описанные заданным синтаксисом данные выражаются в терминах представления их значений в среде ВОС. Первый аспект обозначается термином абстрактный синтаксис, второй – синтаксис передачи.

Абстрактный синтаксис предоставляет возможность прикладным системам выполнять спецификацию передаваемых данных способом, не зависящим от конкретных методов кодирования, используемых для представления данных.

Синтаксис передачи определяет правила кодирования, которые задают специфику представления данных во время их передачи между открытыми системами. Он имеет дело со способом, которым фактически представляются эти данных в виде последовательности нулей и единиц.

Подробнее это можно описать следующим образом. Информация, подлежащая передаче между прикладными объектами, т.е. пользователями представительной службы, указывается в параметре “данные пользователя” примитивов представительной службы. Единица такой информации специфицируется в категориях абстрактного синтаксиса и называется значением представительных данных.

Множество значений представительных данных называется абстрактным синтаксисом, а имя, которое однозначно идентифицирует это множество, – именем абстрактного синтаксиса.

Пользователи представительной службы согласовывают между собой допустимое множество абстрактных синтаксисов и сообщают его представительным объектам. В процессе передачи они могут решить изменить согласованное множество абстрактных синтаксисов.

Для заданного множества абстрактных синтаксисов представительные объекты выбирают взаимоприемлемые синтаксисы передачи.

Синтаксис передачи определяется как те аспекты правил формального описания передаваемых данных, которые задают конкретное их представление. Имя, идентифицирующее однозначное (в рамках заданного имени абстрактного синтаксиса) представление множества значений представительных данных как потока битов, называется именем синтаксиса передачи.

Соответствие между именем абстрактного синтаксиса и именем синтаксиса передачи называется представительным контекстом.

Если согласованное между пользователями и поставщиком представительной службы множество представительных контекстов, которое называется множеством определенных контекстов, пусто, то используется контекст, задаваемый по умолчанию. Контекстом по умолчанию называется представительный контекст, который всегда известен поставщику и обоим пользователям представительной службы. Контекст по умолчанию используется всегда при передаче срочных данных. Контекст по умолчанию может быть либо определен с использованием услуги установления представительного соединения (но не может быть изменен при использовании других услуг), либо установлен по предварительному соглашению.

В общем случае не предполагается однозначного соответствия между конкретными абстрактными синтаксисами и синтаксисами передачи. Сервис представительного уровня позволяет устанавливать такие соотношения динамически. Например, при использовании одного и того же абстрактного синтаксиса передача может происходить с шифрацией или без шифрации, со сжатием или без сжатия передаваемой информации.

Функция согласования синтаксисов передачи поддерживается механизмом переговоров, выполняемым в соответствии с представительным протоколом.

Таким образом, двумя основными функциями уровня представления являются согласование синтаксиса передачи и преобразование (в обе стороны) между абстрактным синтаксисом и синтаксисом передачи. Это преобразование выполняется в рамках представительного объекта невидимым со стороны представительного протокола способом, чем обеспечивается отсутствие влияния таких преобразований на протокол – он имеет дело лишь с использованием системы идентификации синтаксисов передачи.

К другим функциям уровня представления относятся функции запроса на установление и на прекращение сеанса, передачи данных.

При каждом случае передачи без установления соединения уровень представления добавляет к сеансовой службе вызываемые услуги преобразования синтаксиса и выбора синтаксиса. Для их обеспечения уровень представления выполняет функции передачи данных, идентификации синтаксиса и его преобразования.

Возможности, предоставляемые уровнем представления своим пользователям, отнесены к следующим категориями, каждая из которых объединяет ряд функционально схожих услуг:

* установление и завершение соединения,
* управление контекстами,
* передача информации,
* управление диалогом.

Работа с заданным множеством контекстов зависит от выбираемых на этапе установления соединения функциональных групп (блоков) – логических объединений связанных между собой услуг. Такие объединения используются для указания требований пользователя представительной службы во время установления соединения.

Выделяются три функциональные группы: ядра (базовая), управления контекстами, восстановления контекстов.

Функциональная группа ядра доступа всегда, она обеспечивает услуги установления соединения, передачи информации и завершения соединения. Передача информации поддерживается при использовании выбранных сеансовых функциональных групп. Если выбрана только эта группа, то множество заданных на этапе установления представительного соединения контекстов в дальнейшем во время существования этого соединения не может быть изменено. Эта функциональная группа устанавливается по умолчанию.

Функциональная группа управления контекстами должна явно заказываться и согласовываться при установлении представительного соединения. Если она согласована, то множество заданных контекстов может быть изменено во время существования соединения. Услуга, с использованием которой это достигается, может конфликтовать с другими услугами уровня представления, обладающими разрушающими воздействиями. Такие конфликты могут привести к рассогласованию множества заданных контекстов на разных концах соединения. В этом случае возможно получение данных в контексте, неизвестном представительному объекту, о чем он будет сообщать пользователю: “нечитаемые данные”. В то же время пользователи представительного сервиса могут избежать таких конфликтов корректным использованием маркеров.

Функциональная группа восстановления явно заказывается и согласовывается при установлении представительного соединения, причем в этом случае также должна быть заказана функциональная группа управления контекстами. Выбор функциональной группы восстановления дает возможность запоминать множества заданных контекстов в специфицированных точках во время существования представительного соединения. В дальнейшем пользователь представительного сервиса может затребовать возврат к одной из запомненных точек, а сервис представительного уровня обеспечивает корректное восстановление контекстов, существовавших в этой точке. Точками запоминания могут служить точки главной или вспомогательной синхронизации, прерывания активности, установления соединения.

Ход процессов восстановления и управления контекстами во избежание рассогласования контекстных множеств на концах соединения должен подчиняться ряду правил, опирающихся, в частности, на использование маркеров и номеров точек запоминания.

**Услуги уровня представления.** Услуга называется разрушающей, если она может уничтожить другие услуги, находящиеся в процессе исполнения, и неразрушающей – в противном случае. Услуга называется последовательной, если она может применяться только после завершения ранее начатых услуг, и непоследовательной – в противном случае.

Услуга P-CONNECT предназначена для установления представительного соединения с начальным множеством заданных контекстов. Услуга является подтверждаемой, последовательной и неразрушающей. Соответствие параметров примитивов этой услуги дано в табл.2.7; некоторые из этих параметров описаны ниже.

**Список задаваемых представительных контекстов** указывает на то, что прикладная система затребовала во время установления соединения включить в множество заданных контекстов один или более контекстов представления. Параметр состоит из списка с одним или более элементами, каждый из которых содержит две компоненты – идентификатор контекста представления и имя абстрактного синтаксиса.

Таблица 2.7. Соответствие параметров примитивов услуги P-CONNECT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр | P-CONNECT  request | P-CONNECT  indication | P-CONNECT  response | P-CONNECT  confirmation |
| Вызывающий представительный адрес | 0 | 0(=) | - | - |
| Вызываемый представительный адрес | 0 | 0(=) | - | - |
| Альтернативный представительный адрес | - | - | П | П(=) |
| Мультиконтекстность | П | У | - | - |
| Список задаваемых представительных контекстов | П | У(=) | - | - |
| Результирующий список представительных контекстов | - | У | У | У(=) |
| Имя контекста умолчания | П | У(=) | - | - |
| Результат для контекста умолчания | - | У | У | У(=) |
| Качество обслуживания | С | С | С | С |
| Необязательные функциональные группы | П | У | П | У(=) |
| Сеансовые функциональные группы | С | С | С | С |
| Порядковый номер начальной точки синхронизации | С | С | С | С |
| Начальное распределение маркеров | С | С | С | С |
| Идентификатор сеансового соединения | С | С | С | С |
| Данные пользователя | П | У(=) | П | У(=) |
| Результат\* |  |  | О | О(=) |
| \*Если запрос соединения сбрасывается поставщиком, то значение этого параметра устанавливается поставщиком.  О – присутствие параметра обязательно; Н – присутствие параметра необязательно; П – зависит от пользователя; У – присутствие параметра зависит от условия; С – параметр соответствует параметру сеансового сервиса; (=) – значение параметра, эквивалентного значению, указанному в колонке слева. | | | | |

**Результирующий список представительных контекстов** указывает на допустимость или недопустимость контекстов представления из списка задаваемых. Параметр имеет форму списка элементов, однозначно соответствующим элементам списка задаваемых представительных контекстов с указанием допустимости или недопустимости для пользователя или поставщика соответствующего контекста.

**Имя контекста умолчания** – параметр, который явно идентифицирует имя абстрактного синтаксиса контекста умолчания.

**Данные пользователя** содержат данные, выраженные в любой комбинации контекстов, предложенных в списке задаваемых контекстов представления, если он присутствует. Если этот список отсутствует, то данные трактуются в контексте умолчания, который может быть задан явно с помощью параметра “имя контекста умолчания”, или неявно – способом, который стандартом не оговаривается.

**Результат** – параметр, который указывает на результат выполнения услуги P-CONNECT. Может принимать одно из следующих значений: “допустимо”, “отвергнуто пользователем”, “отвергнуто поставщиком (временно)”, “отвергнуто поставщиком (постоянно)”.

Ряд параметров – качество обслуживания, сеансовые функциональные группы, и т.д. – служат для обеспечения доступа пользователя к соответствующим параметрам сеансового сервиса. Связь параметров примитивов услуг верхних трех уровней ЭМВОС будет обсуждаться и далее.

Некоторые параметры услуги P-CONNECT переносятся от пользователя-источника к пользователю-получателю без изменений, другие должны быть согласованы между пользователями сервиса и поставщиком. Это согласование выполняется специальным механизмом, при использовании которого поставщику разрешается модифицировать значения параметров, определенных в примитивах запроса, перед тем, как они будут доставлены в примитивах индикации. При этом те же параметры, определенные в примитивах ответа, доставляются при соблюдении некоторых условий без изменений.

Услуги завершения представительного соединения пользователем P-U-ABORT и поставщиком P-P-ABORT – обе непоследовательные и разрушающие. Упорядоченное завершение (без потерь данных) представительного соединения пользователями представительной службы обеспечивает по завершении всех услуг услуга P-RELEASE.

Услуга управления контекстом представления P-ALTER-CONTEXT обеспечивает задание представительных контекстов и изменение состава множества заданных контекстов представления. Эта услуга неразрушающая и последовательная. Изменение множества заданных контекстов происходит как на стороне инициатора в моменты получения подтверждения, так и на стороне ответчика в момент выдачи ответа.

Назначение четырех услуг передачи информации и тринадцати услуг управления диалогом аналогично назначению соответствующих услуг сеансовой службы.

Последовательности услуг уровня представления должны удовлетворять определенным ограничениям (быть допустимыми). В частности, услуга P-CONNECT не может быть начата на установленном представительном соединении. Одновременные попытки двух прикладных объектов обрабатываются поставщиком сервиса независимо. Прикладные системы сами разрешают конфликты. Услуга P-ALTER-CONTEXT, наоборот, может вызываться только на установленном представительном соединении, и может быть разрушена, как об этом упоминалось ранее, целым рядом других услуг. При отображении на сеансовый сервис дополнительно действуют упоминавшиеся правила последовательностей сеансового сервиса.

В стандарте протокола уровня представления процедуры сгруппированы в соответствии с тремя описанными функциональными группами. Приведем далее протокольную спецификацию (при использовании функциональной группы ядра) фрагмента фазы установления представительного соединения. Группе ядра соответствует набор из шести типов Пд-БДП. Напомним здесь, что имеются примитивы услуг уровня представления, которые без какой-либо модификации отображаются на соответствующие примитивы услуг сеансового уровня без формирования Пд-БДП. Обычно параметры таких примитивов передаются в поле данных пользователя соответствующих примитивов услуг сеансового уровня.

В табл.2.8 выписаны частичные списки имен элементов множеств входных и выходных событий, состояний автомата, предикатов и специальных действий.

В табл.2.9 приведена часть соответствующей таблицы состояний-переходов.

В этом протоколе также определены структура и правила кодирования используемых для передачи данных и управляющей информации. Структура Пд-БДП определена через отображение его параметров в параметры “данные пользователя сеансовой службы” примитивов сеансовой службы, а структура этих последних параметров специфицирована там же с использованием ASN.1 – нотации абстрактного синтаксиса номер 1 (НАС.1) и стандартных правил его кодирования.

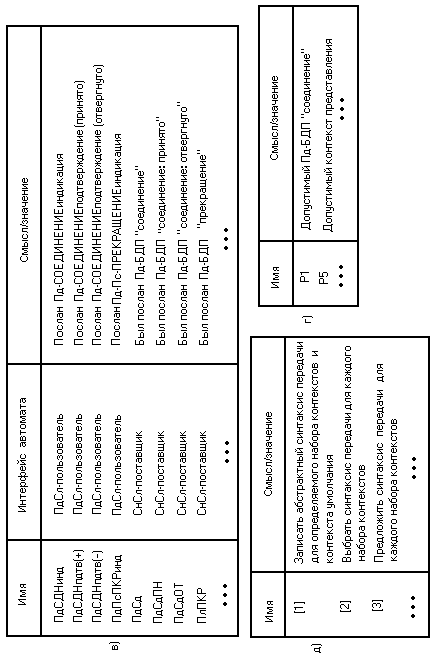
Управляющая информация представительного протокола также специфицирована на ASN.1 и кодируется с использованием стандартных правил.

В том же стандарте определены правила кодирования данных пользователя представительной службы в Пд-БДП для двух режимов – нормального и режима X.410-1984. Режим выбирается в ходе установления ассоциации, о чем речь пойдет несколько позднее.

Таблица 2.8. Фрагменты списков имен элементов множеств входных событий (а), состояний автомата (б), выходных событий (в), предикатов (г) и специальных действий (д)

а)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Интерфейс автомата | Смысл/Значение | |
| ПдСДНзпр | ПдСл-пользователь | Получен Пд-СОЕДИНЕНИЕзапрос | |
| ПдСДНотв(+) | ПдСл-пользователь | Получен Пд-СОЕДИНЕНИЕответ (принято) | |
| ПдСДНотв(-) | ПдСл-пользователь | Получен Пд-СОЕДИНЕНИЕответ (отвергнуто) | |
| ПдСд | СнСл-поставщик | Получен Пд-БДП «соединение» | |
| ПдСдПН | СнСл-поставщик | Получен Пд-БДП «соединение: принято» | |
| ПдСдОТ | СнСл-поставщик | Получен Пд-БДП «соединение: отказано» | |
| … | … | … | |
| б) |  |  | |
| Имя | Смысл/Значение | |  |
| Сост 10 | Бездействие; нет соединения уровня представления | |  |
| Сост 11 | Ожидание Пд-БДП «соединение:принято» | |  |
| Сост 12 | Ожидание Пд-СОЕДИНЕНИЕответ | |  |
| … | … | |  |
| Сост С0 | Передача данных | |  |



В нормальном режиме используются соответствующие синтаксисы передачи, в режиме X.410-1984 – стандартные (базовые) правила кодирования ASN.1.

В предыдущем разделе обсуждались обстоятельства, осложняющие синтаксическую сторону организации взаимосвязи систем. Одно из них – возможное различие локальных синтаксисов и, как следствие, необходимость их согласования и использование преобразований – рассмотрено выше.

Таблица 2.9. Часть таблицы состояний переходов автомата отработки фазы установления представительного соединения

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Событие | Состояние | | | | | |
| Сост 10 | Сост 11 | | Сост 12 | … | Сост С0 |
| Пд СДНзпр | 1 | 0 | | 0 | … | - |
| Пд СДНотв(+) | 0 | 0 | | 2 | … | - |
| Пд СДНщтв(-) | 0 | 0 | | 3 | … | - |
| ПдСд | 4 | 0 | | 0 | … | - |
| ПдСдПН | 0 | 5 | | 0 | … | - |
| ПдСдОТ | 0 | 6 | | 0 | … | - |
| … | … | … | | … | … | … |
| 0=ПдПсПКРинд, ПлПКР, Сост 10  1=Р5: ПдСд, [3], Сост 11  2=ПдСдПн, [1][2], Сост С0  3=ПдСдОТ, Сост 10 | | | 4=Р1: ПдСДНинд, Сост 12;  NOT Р1: ПдСдОТ, Сост 10  5=ПдСДНпдтв(+), [1], Сост С0  6=ПдСДНпдтв(-), Сост 10 | | | |

Далее остановимся на втором обстоятельстве – потребности в достаточно мощных по выразительности средствах для описания сложных структур данных, участвующих в информационном обмене. Абстрактный синтаксис может быть описан множеством способов. Наиболее широко в рассматриваемой области используется стандарт, определяющий ASN.1, который используется, в частности, для задания синтаксиса, применяемого в разработках верхних уровней модели ВОС, проводимых МОС.

Система абстрактного синтаксиса дополняется правилами кодирования, определяющими значения потока октетов. Спецификации на ASN.1 в принципе могут быть представлены различными способами кодирования. Стандартный способ МОС называют еще базовыми правилами кодирования.

Дадим далее краткий обзор ASN.1. Целью спецификаций с использованием его средств является строгое определение правил формирования информационного потока из произвольных достаточно сложных значений и выделении этих значений при обработке потока. Центральную роль в этом процессе играет понятие типа.

Множества однородных значений характеризуются типом, а конкретного представителя из таких множеств будем называть значением соответствующего типа.

В ASN.1 имеется базовый набор простых (встроенных) типов. Это – BOOLEAN, INTEGER, BITSTRING, OCTETSTRING и NULL. Используя встроенные типы и применяя небольшой набор способов построения, можно получать сложные типы. Способы можно применять в любой последовательности и не только к встроенным типам, но и к типам, полученным из встроенных типов с помощью тех же способов. Полученные сложные типы называются структурированными. Имеется пять способов структурирования: списком (SEQUENCE), списком из (SEQUENCE OF); множеством (SET), множеством из (SET OF); выбором (CHOICE); вырезкой (SELECTION); и, наконец, типы, множество значений которых определено иначе. Такие типы называются внешними.

Каждый тип, используемый в стандарте ASN.1, имеет специальный признак – тег. Он определяется либо стандартом, либо пользователем и вводится скорее для машинного использования, чем для человека. Один и тот же тег может назначаться различным типам. В этом случае теги будут трактоваться в соответствии с текущим контекстом. В то же время для обозначения одного и того же типа пользователь может использовать различные теги. В этом случае типы являются изоморфными, но отличаются друг от друга тегами. Тег состоит из идентификатора класса и неотрицательного целого, различающего теги внутри класса. Имеется четыре класса тегов: UNIVERSAL, APPLICATION, PRIVATE и CONTEXT-SPECIFIC.

ASN.1 представляет собой средство со строгой типизацией, что требует явного описания используемых типов.

При структуризации возможно использование указаний OPTIONAL или DEFAULT. Первое означает, что значение данного типа может быть опущено. Указание DEFAULT, за которым должно следовать “значение” данного типа, говорит о том, что если значение типа опущено, то по умолчанию восстанавливается указанное “значение”.

Для обозначения значений, типы которых не специфицированы, используются служебные слова ANY и EXTERNAL. Первое соответствует значениям любого типа ASN.1, второе указывает на значения, типы которых описаны иначе, не средствами ASN.1.

При передаче значения тегированного типа передается как тег, определенный пользователем, так и тег самого типа. В тех случаях, когда тег самого типа передавать не надо (однозначность разбора обеспечивается явным тегированием), можно использовать служебное слово IMPLICIT.

В ASN.1 имеется механизм макроопределений, описывающий синтаксис новых типов и значений, вводимых пользователем. Для введения макроопределений используется служебное слово MACRO, а для определения новых типов и новых значений – слова TYPE NOTATION и VALUE NOTATION. Получаемый новый тип можно использовать в дальнейших определениях.

Для структуризации описаний ASN.1, относящихся к одной предметной области, вводятся модули. Тело модуля охватывается скобками BEGIN и END и содержит определения ASN.1. Модуль имеет имя, за которым следует служебное слово DEFINITION. Имена модулей должны быть уникальными.

Базовые правила кодирования для ASN.1 формируют синтаксис передачи. В соответствии с этими правилами возможно двоякое кодирование всех типов, за исключением внешних (рис.2.8). В одном из форматов размер содержимого указывается явно, в другом – определяется по признаку конца содержимого (два нулевых октета).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| а) | Идентификатор | Длина | Содержимое |  |  |
|  |  |  |  |  |  |
| б) | Идентификатор | 1000 0000 | Содержимое | 0000 0000 | 0000 0000 |

Рис. 2.8. Общие форматы кодирования: с явным указанием длины (а) и с признаком конца содержимого (б)

Идентификатор – это тег типа значения, т.е. класс и номер типа. Правила его кодирования описывают размещение кодов класса и номера в нужном количестве октетов в определенном порядке.

Поле “содержимое” состоит из последовательности октетов, начиная с нулевого, которые представляют значения данных. Для представления используется простое или сложное кодирование. При простом кодировании октеты содержимого непосредственно представляют кодируемое значение. Этот метод применяется для кодирования значений простых типов BOOLEAN, INTEGER и NULL.

Сложное кодирование требует специального представления значения одним или несколькими октетами содержимого. Этот вид кодирования применяется для представления значений структурированных типов. Значения типов BITSTRING и OCTETSTRING могут представляться по усмотрению разработчиков любым из описанных методов. Рекомендуется использовать сложное кодирование в тех случаях, когда передается часть строки, которая еще полностью не сформирована.

## Прикладной уровень (общий прикладной сервис)

Прикладной уровень является последним (седьмым) уровнем ЭМВОС. С этим связан ряд особенностей, отличающих его от других уровней эталонной модели. На прикладном уровне завершается наращивание сервисной мощности модели ВОС по архитектурной вертикали и начинается своего рода горизонтальное наращивание. Сервис, поставляемый различными протокольными прикладными объектами, с учетом определенных ограничений можно объединять, формируя различные профили сервиса прикладного уровня в интересах конкретных специальных прикладных систем.

На прикладном уровне осуществляется окончательное и естественное погружение механизмов взаимосвязи, объявленных в модели, в вычислительную среду с ее понятийным построением. Очевидна поэтому потребность в уточненном описании понятия прикладного процесса (application process, AP). Прикладной процесс – это идентифицируемый объект в рамках реальной открытой системы, ведущий обработку информации и ответственный за согласование правил среды своего существования с законами модели ВОС. Прикладной процесс представляется в рамках модели ВОС одним или более прикладными объектами (Application Entity, AE).

Прикладной объект, далее, представляется совокупностью элементов прикладных служб (Application Service Elements, ASE). Такая совокупность, в свою очередь, может быть разделена на набор общих элементов прикладных служб (ОЭПС, ACSE), некоторый набор специальных элементов прикладных служб (СЭПС, SASE), а также элемент, создаваемый разработчиком прикладной системы – элемент пользователя (ЭП, UE)[[10]](#footnote-10) (рис.2.9).

Целью выделения общего прикладного сервиса в отдельный стандарт является обеспечение разработчиков прикладных распределенных систем обработки информации наиболее общими унифицированными услугами, которые в совокупности с услугами представительного уровня составляют понятийную основу построения и работы приложений.

В наборе общих элементов прикладных служб (ОЭПС) имеется элемент службы управления ассоциацией (ЭСУА) – множество функций, обеспечивающих минимальные требуемые услуги службы прикладного уровня[[11]](#footnote-11). Прикладная ассоциация устанавливается между двумя вызовами[[12]](#footnote-12) прикладных объектов при необходимости обмена информацией между прикладными объектами с целью выполнения задания по распределенной обработке информации.

Свойства и цель информационного обмена по прикладной ассоциации определяются в зависимости от устанавливаемого для нее контекста обработки информации и прикладного контекста. Первый определяется конкретным заданием по распределенной обработке информации, для которого установлена ассоциация. Прикладной контекст формируется явно определенным набором элементов прикладной службы, которые могут быть вызваны пользователем ассоциации.

Каждый элемент прикладной службы требует описания предоставляемых им услуг и поддерживающего их выполнение протокола.

Спецификация протокола задает правила информационного обмена между равноправными элементами прикладной службы. Она также может содержать описание используемой службы представительного уровня или услуг, предоставляемых другими элементами прикладной службы.

Определение услуг аналогично определению служб уровней ЭМВОС, оно описывает функции, выполняемые данным элементом.

Помимо этого, для некоторых элементов прикладной службы устанавливаются также концептуальные модели, например, такие, как виртуальное файлохранилище для специального элемента передачи, доступа и управления файлом.

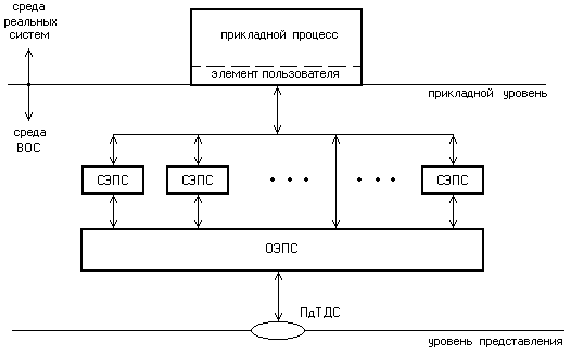


Рис. 2.9. Элементы прикладных служб

Прикладное взаимодействие базируется на представительном соединении, которое создается и завершается поставщиком общего прикладного сервиса. Поэтому у пользователя сервиса нет необходимости устанавливать представительное соединение прямым выходом на представительный уровень с помощью примитива P-CONNECT. Доступ к ЭСУА, обеспечивающему услугу образования ассоциации, может быть осуществлен опосредованно – вызовом специального элемента прикладной службы (СЭПС) или элемента пользователя, что иллюстрируется структурой связей элементов на рис.2.9.

**Услуги, предоставляемые ЭСУА.** Для управления одной ассоциацией используются следующие услуги: A-ASSOCIATE (Пк-АССОЦИИРОВАНИЕ), A-RELEASE (Пк-ОСВОБОЖДЕНИЕ), A-U-ABORT (Пк-Пл-ПРЕКРАЩЕНИЕ (Пк-Пл-РАЗРЫВ)), A-P-ABORT (Пк-Пс-ПРЕКРАЩЕНИЕ (Пк-Пс-РАЗРЫВ)).

Подтверждаемая услуга A-ASSOCIATE позволяет прикладному объекту установить прикладное соединение (ассоциацию) с другим прикладным объектом. В ходе установления ассоциации прикладные объекты обмениваются параметрами прикладного соединения и согласовывают их. Подтверждаемая услуга A-RELEASE позволяет прикладному объекту произвести упорядоченное завершение существующего прикладного соединения без потери передаваемой информации. Две оставшиеся неподтверждаемые услуги используются для безусловного завершения ассоциации с возможной потерей информации.

В число параметров, обрабатываемых в ходе установления ассоциации, входят такие, как имена принимающего (ведомого) и инициирующего (ведущего) прикладных объектов, вызывающий и вызываемые адреса объектов, имя прикладного контекста, данные пользователя, результат, а также целый ряд параметров, семантика которых определяется сервисом сеансового и представительного уровня. Среди последних – такой, как список представительных контекстов. В общем случае параметр определяет набор абстрактных синтаксисов, для которых требуется создание представительных контекстов.

Установив ассоциацию, ЭСУА не вмешивается в дальнейший диалог, ведущийся СЭПС, пока те (либо поставщик прикладного сервиса) не запросят (не сообщат) о завершении ассоциации.

В параметрическом отношении общая функциональная ориентация всех трех верхних уровней ЭМВОС (на приложения) проявляется в том, что многие параметры, связанные с примитивами, отображаются непосредственно с одного уровня на другой. На рис.2.10 приведена схема, в соответствии с которой осуществляется параметрическая комплектация сеансовой услуги S-CONNECT и представительной услуги P-CONNECT на основе прикладной услуги A-ASSOCIATE.

Для того чтобы некоторый вызов прикладного объекта установил ассоциацию с другим вызовом прикладного объекта, ему должно быть известно наименование этого прикладного объекта. Используя такие наименования, пользователь прикладной службы сначала с помощью СЭПС справочной службы (СС) должен получить адреса прикладных объектов. Все последующие обращения к другим СЭПС уже, как правило, содержат в качестве параметров как имена, так и соответствующие адреса обоих прикладных объектов.

Протокол управления ассоциацией обеспечивает услуги ЭСУА, опираясь на службу представительного уровня. Использование этой службы может быть двоякое. В случае, если доступна только ее функциональная группа ядра (базовый функциональный блок), то ЭСУА-БДП не используются, а примитивы ЭСУА и представительной службы отображаются друг в друга непосредственно. Если же протокол функционирует в режиме, называемом нормальным, то функциональная группа ядра представительной службы используется для обмена ЭСУА-БДП между равноправными протокольными объектами, при этом использование дополнительных функциональных групп представительной службы согласовывается между пользователями ЭСУА. Все ЭСУА-БДП (их пять типов) специфицированы с использованием ASN.1.

Далее следует фрагмент автоматной протокольной спецификации фазы установления ассоциации. Как и ранее, вначале следуют описания множеств имен событий, состояний и предикатов (см. рис.2.10) и далее – часть таблицы состояний-переходов автомата (см. рис.2.11).

Опишем далее еще один общий элемент прикладной службы – элемент управления завершением (присвоением, фиксацией, commitment), параллельностью (соревнованием, concurrency) и восстановлением (recovery) – УЗПВ (CCR).

Вне рамок рассмотрения здесь останутся общие элементы прикладных служб надежной передачи и удаленных операций.

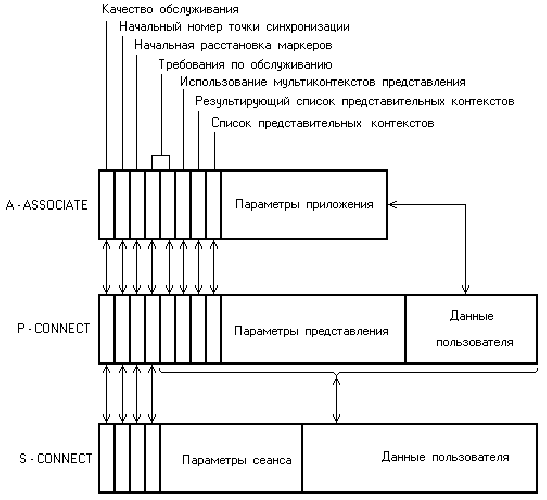


Рис. 2.10. Схема формирования параметров услуг A-ASSOCIATE, P-CONNECT и S-CONNECT

В основе УЗПВ лежит понятие атомарного (неделимого) действия (АД). АД – последовательность операций, выполняемых распределенным приложением, которая обладает следующими свойствами:

* АД прямо или косвенно управляется единственным прикладным объектом;
* на выполнение АД не влияют внешние воздействия;
* части АД, выполняемые различными прикладными объектами и, возможно, в других открытых системах, либо все успешно завершаются, либо все завершаются без изменения соответствующих данных, а управляющий прикладной объект получает диагностические сообщения от вовлеченных в это АД прикладных объектов (все данные, вовлеченные в атомарное действие, называются связанными).

Таблица 2.10. Фрагменты списков имен элементов множеств входных событий, состояний автомата, выходных событий, предикатов и специальных действий

|  |  |  |
| --- | --- | --- |
| Имя | Интерфейс  автомата | Смысл/Значение |
| ПкАССзпр | ОСл-пользователь | Получен  Пк-АССОЦИИРОВАНИЕзапрос |
| ПкАССотв(+) | ОСл-пользователь | Получен  Пк-АССОЦИИРОВАНИЕответ(принято) |
| ПкАССотв(-) | ОСл-пользователь | Получен Пк-  АССОЦИИРОВАНИЕответ(отвергнуто) |
| ПкАЗПР | ПДСл -поставщик | БДП-ПкАЗПР(запрос ассоциации) |
| ПкАОТВ(+) | ПДСл -поставщик | БДП-ПкАОТВ+(принято) |
| ПкАОТВ(-) | ПДСл -поставщик | БДП-ПкАОТВ-(не принято) |
| ПДСДНпдтв(-) | ПДСл -поставщик | Пд-СОЕДИНЕНИЕподтверждение  (отвергнуто) |
| … | … | … |
|  |  |  |
| Имя | Смысл/значение | |
| Сост 0 | Бездействие (ассоциации нет) | |
| Сост 1 | Ожидание ответа на отосланный БДП-Пк-АЗПР | |
| Сост 2 | Ожидание Пк-АССОЦИИРОВАНИЕответ | |
| Сост 5 | Ассоциация установлена | |
| … | … | |
|  | | |
| Имя | Интерфейс  автомата | Смысл/Значение |
| ПкАССинд | ОСл-пользователь | Получен  Пк-АССОЦИИРОВАНИЕиндикация |
| ПкАССпдтв(+) | ОСл-пользователь | Получен Пк-АССОЦИИРОВАНИЕ  подтверждение(принято) |
| ПкАССпдтв(-) | ОСл-пользователь | Получен Пк-АССОЦИИРОВАНИЕ  Подтверждение(отвергнуто) |
| ПкАЗПР | ПДСл–поставщик | Был послан БДП-ПкАЗПР |
| ПкАОТВ(+) | ПДСл–поставщик | Был послан БДП-ПкАОТВ+ |
| ПкАОТВ(-) | ПДСл–поставщик | Был послан БДП-ПкАОТВ- |
| ПкПКРинд | ОСл–пользователь | Послан Пк-ПРЕКРАЩЕНИЕиндикация |
| ПКР | ПдСл-поставщик | Был послан БДП(прекращение) |
| … | … | … |
|  |  |  |
| Имя | Смысл/значение | |
| Р1 | Пк-объект может обеспечить соединение | |
| … | … | |

Завершение АД переводит данные, над которыми оно производится, в конечное состояние и завершает взаимосвязь между прикладными объектами по поддержанию службы УЗПВ (прекращает ЗПВ-отношения между ними). Откат (возврат) АД возвращает данные в исходное (начальное) состояние и прекращает ЗПВ-отношение между прикладными объектами.

Управление параллельностью гарантирует, что АД не завершится до тех пор, пока не выполняться следующие условия:

* не завершатся все АД, которые обрабатывали те же данные до их периода использования рассматриваемым АД;
* никаких изменений значений данных в течение их периода использования не произойдет, за исключением тех, которые планируются данным АД.

Таблица 2.11. Часть таблицы состояний-переходов автомата обработки фазы установления ассоциации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Событие | Состояние | | | | 0=ПкПКРинд, ПКР, Сост0  1=Р1:ПкАЗПР, Сост1  2=ПкАОТВ(+), Сост5  2=ПкАОТВ(-), Сост0  4=Р1:ПкАССинд, Сост2 NOT Р1: ПкАОТВ(-), Сост0  5=ПкАССпдтв(+), Сост5  6=ПкАССпдтв(-), Сост0 |
| Сост 0 | Сост 1 | Сост 2 | … |
| ПкАССзпр | 1 | 0 | 0 | … |
| ПкАССотв(+) | 0 | 0 | 2 | … |
| ПкАССотв(-) | 0 | 0 | 3 | … |
| ПкАЗПР | 4 | 0 | 0 | … |
| ПкАОТВ(+) | 0 | 5 | 0 | … |
| ПкАОТВ(-) | 0 | 6 | 0 | … |
| ПдСДНпдтв(-) | 0 | 6 | 0 | … |
| … | … | … | … | … |

Механизм управления параллельностью может базироваться на технике блокирования используемых ресурсов. Однако возможно применение других алгоритмов, связанных с менее строгим упорядочением АД, допущением большего числа случаев возврата (отката). Выбор техники управления параллельностью остается за реализаторами прикладных распределенных систем.

Управление восстановлением гарантирует правильное выполнение АД даже при наличии отказов прикладных объектов и среды передачи.

Установление ЗПВ-отношения подразумевает разделение его участников на ведущего (старшего) (прикладной объект, инициировавший ЗПВ-отношение) и ведомого (младшего) (прикладной объект, принявший ЗПВ-отношение). АД, вовлекая в свой ход множество прикладных объектов, образует дерево атомарного действия, ветвями которого являются ЗПВ-отношения, развивающиеся на существующих двунаправленных ассоциациях. Таким образом, АД охватывает множество ведущих и ведомых, причем среди ведущих существует источник АД – корень АД. В ходе развития АД один и тот же прикладной объект может иметь несколько вхождений в дерево АД, причем в различных качествах (рис.2.11).

A

B

C

D

E

F

G

F

G

J

Корень

Ведущие

Ведомые

A

B

C

E

G

B,C

D

E,F,G

F,G

J

Рис. 2.11. Дерево АД

Один из методов выхода из возможных тупиковых ситуаций, возникающих при организации подобных деревьев АД, – использование таймеров.

Придание свойства атомарности некоторому действию достигается с помощью выполнения этого действия в две фазы. В первой фазе источник действия определяет, все ли ведомые способны выполнить его завершение. Только после получения от всех ведомых отчетов о готовности к выполнению завершения действия принимается решение о том, выполнять ли завершение. В ходе второй фазы ведомые осуществляют либо завершение, либо откат в соответствии с решением (рис.2.12).

Эта схема обеспечения свойства атомарности действует при соблюдении следующих условий:

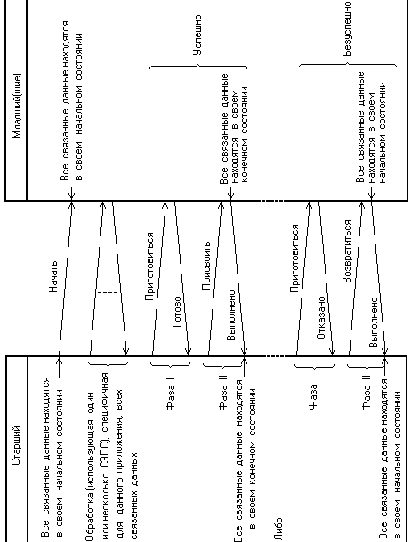
* в фазе завершения АД не должно существовать возможностей изменения ее результатов;
* ведущий может потребовать выполнения возврата к начальному состоянию в любое время до выдачи запроса на завершение действия;
* ведущий не может требовать выполнения завершения от ведомого до тех пор, пока не получит от него согласия на это;
* если ведомый сообщил о своей готовности завершить действие, то он не может отказаться от требования выполнить завершение;
* ведомый может отказаться от завершения в любое время вплоть до момента выдачи своего согласия на него;
* ведущий должен потребовать выполнить возврат к исходному (начальному) состоянию от ведомого, указавшего на невозможность завершения.

Корень несет ответственность за завершение или возврат. Имя корня используется для идентификации АД в целях управления параллельностью.

Содержательная сторона АД определяется прикладными процессами и СЭПС, которые манипулируют вовлеченными в действие данными в рамках АД от момента начала действия и до момента завершения (см. рис.2.12). УЗПВ лишь обеспечивает средства для инициации АД и управления им. Вовлеченные в АД прикладные процессы сами, локально, решают вопрос о том, какие шаги следует предпринять, чтобы обеспечить неукоснительное выполнение правил атомарного действия, как об этом говорилось ранее на примере механизма управления параллельностью.

Перед инициацией АД ведущий прикладной объект с помощью ЭСУА устанавливает ассоциацию с каждым из ведомых. После этого с помощью услуги УЗПВ C-BEGIN ведущий устанавливает с каждым из ведомых ЗПВ-отношение. Дерево АД всякий раз при этом приобретает еще одну ветвь. Отказ партнера образовать ЗПВ-отношение выполняется с помощью услуги C-REFUSE. Действие C-BEGIN связано с установлением главной точки синхронизации в рамках используемой ассоциации.

Рис. 2.12. АД от начала до двухфазного завершения



Вслед за услугой C-BEGIN прикладной пользователь может обращаться к другим прикладным услугам, что, собственно, и определяет сущность (содержательную сторону) вводимого АД.

Помимо указанных двух, в состав услуг УЗПВ входят услуги C-PREPARE, C-READY, C-COMMIT, C-ROLLBACK, C-RESTART. Примитивы трех последних услуг используются в разных ситуациях при завершении ЗПВ-отношения. В целом сервисные ЗПВ-последовательности развиваются в рамках ассоциаций, образованных приложениями. В каждый момент на ассоциации может существовать лишь одна сервисная ЗПВ-последовательность.

Разрешенные сервисные ЗПВ-последовательности для ведущего и ведомого собраны в две таблицы, занимающие несколько страниц, и сопровождаются объемными комментариями и уточнениями.

Применение таймеров в ЗПВ-сервисе необязательно (значения таймеров входят в виде параметров в некоторые примитивы). Их назначение заключается в обеспечении приемника услуги дополнительной информацией, которую он может использовать наряду с множеством других факторов для определения своего поведения.

Протокол УЗПВ применяется в тех случаях, когда другие прикладные протоколы вызывают его процедуры, используя примитивы службы УЗПВ. Он специфицирует УЗПВ-БДП в терминах ASN.1, а также указывает, каким образом эти БДП отображаются в представительные БДС. Количество типов УЗПВ-БДП – десять основных и шесть вспомогательных. Особенность протокола УЗПВ заключается в том, что его процедуры используют представительный сервис для обеспечения своих функций наряду со специальными прикладными протоколами. Отсюда исходит требование к тем приложениям, которые пользуются ЗПВ-сервисом. Представительные контексты таких приложений должны содержать абстрактный синтаксис, описывающий УЗПВ-БДП. Включение такого представительного контекста осуществляется в момент образования приложением соответствующей ассоциации или непосредственно перед началом АД путем инициации представительной услуги P-ALTER-CONTEXT.

На сеансовом уровне должны быть доступны следующие функциональные группы: базовая сеансовая, сеансовая типизированных данных, сеансовая главной синхронизации и сеансовая ресинхронизации.

## Вопросы к разделу 2

1. Опишите услуги, предоставляемые канальным уровнем.
2. Раскройте функциональное наполнение сетевого уровня.
3. Опишите сервис сетевого уровня.
4. Охарактеризуйте связь параметров качества сервиса транспортного и сетевого уровней.
5. Опишите услуги, предоставляемые транспортным уровнем.
6. Охарактеризуйте фазу установки транспортного соединения.
7. Охарактеризуйте фазу передачи данных транспортного соединения.
8. Верно ли, что транспортный сервис с соединением нуждается в поддержки аналогичного сетевого сервиса. Ответ обоснуйте.
9. Определите понятия «маркер», «точка синхронизации», «диалоговый эле­мент», «активность», «ресинхронизация», «переговоры», «функциональная группа».
10. Опишите услуги, предоставляемые уровнем представления.
11. В контексте эталонной модели ВОС кратко охарактеризуйте понятия «абстрактный синтаксис», «синтаксис передачи» и взаимосвязь между ними.
12. Составьте схему элементов прикладных служб.
13. Опишите услуги, предоставляемые элементом службы управления ассоциацией.
14. Опишите службу управления завершением, параллельностью и восстановлением.

# СПЕЦИАЛЬНЫЕ ЭЛЕМЕНТЫ ПРИКЛАДНЫХ СЛУЖБ И РЕАЛИЗАЦИЯ ОТКРЫТЫХ СИСТЕМ

В данном разделе пособия рассмотрены фрагменты реализации ПО открытых систем, организация управления ВОС и служба справочника. Для самостоятельной проработки здесь оставлены СЭПС передачи, доступа и управления файлами (ПДУФ); виртуального терминала (ВТ); системы обработки сообщений (СОС) и др.

## О программной реализации

Различным аспектам реализации, и не только программным, обсуждавшихся идей ВОС посвящены целые монографии. По необходимости кратко остановимся далее лишь на немногих моментах – с учетом специализации именно программных, – позволяющих лучше уяснить архитектурно-декомпозиционный подход, принятый в модели ВОС, при взгляде со стороны программного обеспечения реальной открытой системы. Ясно также, что многое зависит от конкретных аппаратурных и программных средств такой системы, поэтому дальнейшее следует рассматривать скорее как ориентировочную иллюстрацию общего подхода.

Хорошо известно, что реализация функций взаимосвязи весьма ресурсоемка. Определенное представление об этом можно было составить из уже усвоенного материала. Поэтому для выполнения связных процедур и для обработки обычных (прикладных) задач чаще всего используют отдельные подсистемы. Как правило, полная подсистема связи – это некоторый отдельный конструктив. В нем объединены: локальный процессор связи; память, имеющая объем, достаточный для хранения программных реализаций протоколов, участвующих в обработке данных; схемы, необходимые для реализации физического интерфейса с сетью. Данные включают в себя информацию о (локальном) состоянии протокола каждого из уровней и сообщения, передаваемые между уровнями.

Прикладные процессы пользователей выполняются на главной (host) системе. Взаимодействие главной системы и ее подсистемы связи опирается на программное обеспечение межпроцессорных связей, имеющееся в обеих системах. Синхронизация обмена данными между ними обеспечивается системой прерываний.

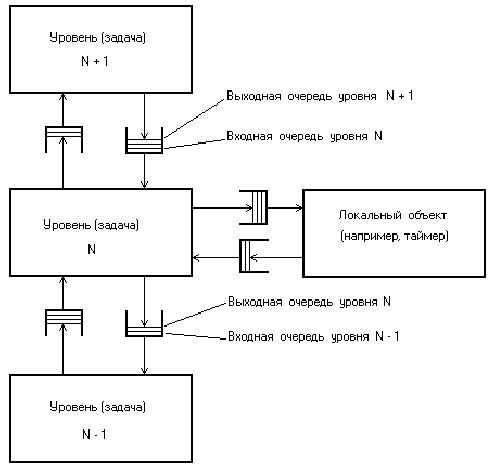


Рис. 3.1. Структура очередей взаимодействующих задач

Программные реализации протоколов смежных уровней должны быть автономны в том же смысле, что и сами эти уровни, ибо в противном случае будут утрачены все преимущества многоуровневой архитектуры. Поэтому полная подсистема связи реализуется в виде пакета модулей задач (процессов), по одному на каждый уровень, с добавленными задачами для выполнения (локальных) функций управления и обработки прерываний таймера. Задачи взаимодействуют друг с другом с помощью набора очередей или почтовых ящиков, как это показано на рис.3.1, используемая дисциплина – FIFO (первым пришел – первым обслужен). Межзадачное взаимодействие осуществляется через локальное ядро, функционирующее в режиме реального времени. Это же ядро производит планирование задач и обработку прерываний, поступающих от таймеров.

Смежные уровни взаимодействуют друг с другом с помощью примитивов. Это взаимодействие имеет атомарный (неделимый) характер, что проявляется двояким образом. С одной стороны – сами примитивы – это первичные, неделимые элементы описания сервиса. С другой – ход обработки протокольным автоматом одного из них не может быть прерван обработкой другого, несмотря на асинхронность моментов их появления. Для обеспечения атомарности обработки каждого входного события (принятия примитива к обработке) прочие локальные события передаются протокольному объекту тем же способом, что и те, которые поступают сверху и снизу от смежных уровней. С таким подходом мы встретимся позже при рассмотрении организации управления ВОС и услуг СЭПС информационных служб управления с доступом их примитивов непосредственно на нужный уровень. Здесь же чуть ниже увидим, что аналогично обрабатываются примитивы таймера.

Каждый примитив службы совместно со связанными с ним параметрами формируется в буфере памяти, называемом блоком управления событиями (БУС). Форма представления этих параметров локальна и соответствует языку реализации протокольных объектов (задачи в целом).

Как правило, число и тип параметров зависят от примитива, однако с каждым протокольным объектом связывают единую структуру (БУС) символьных данных. При этом в заголовке каждого БУС имеется поле типа примитива, за ним следует поле всех возможных параметров, связываемых с полным набором примитивов одного уровня. При передаче примитивов между уровнями посылающая задача указывает тип передаваемого примитива в заголовке БУС, а получающая задача использует это поле, чтобы определить ту конкретную процедуру выходного события, которая должна быть инициирована в данном случае. Эта процедура выделяет лишь те параметры, которые касаются ее, т.е. лишь те, которым присвоены содержательные значения.

Структура типичного БУС приведена на рис.3.2. Этот БУС относится к транспортному уровню, т.е. используется при взаимодействии между сеансовым и транспортным уровнями для всех транспортных примитивов запроса, индикации, ответа и подтверждения. Полный БУС имеет структуру записи (record), а переменной ТипСобытия присваивается тип примитива службы. Об использовании поля УказательНаБфДП и связанного с ним ДлинаБфДП см. далее. Поля вызываемых и вызывающих СнТДС, ТТДС и СтТДС применяются в примитивах Т-СОЕДИНЕНИЕ, а поля ИдПриемника и ИдИсточника – в последующих примитивах Т-ДАННЫЕ для установления соответствия между транспортным соединением и данными пользователя, связанными с примитивами.

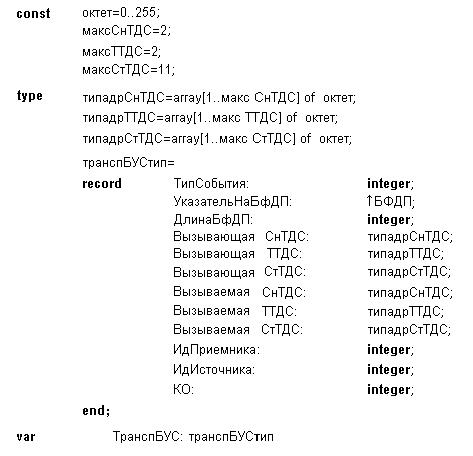


Рис. 3.2. Пример структуры межуровневых данных: транспортный БУС

Чтобы передавать примитивы службы между уровнями (задачами), инициирующая задача формирует для локального ядра примитив взаимодействия задач, выдавая в качестве параметра указатель адреса БУС. В результате этого ядро помещает указатель в конец соответствующей межуровневой очереди. Задача в начале работы опрашивает все свои очереди между ее и выше-, и нижележащим уровнями, а также очереди с задачами управления и таймера в отношении наличия БУС, ожидающего обработки. Если таковой имеется, то задача считывает указатель из начала соответствующей очереди и приступает к обработке входного события. В результате, как правило, формируется БДП и строится должным образом форматированное сообщение, которое затем помещается в одну из выходных очередей задачи.

Описанный механизм оказывается рациональным, когда в прикладном уровне реализуется всего одна функция обработки и, следовательно, в каждый данный момент может выполняться всего один запрос на службу. При параллельной обработке нескольких запросов на службу все межуровневые сообщения либо могут передаваться с помощью одного и того же набора очередей, либо для каждой активной ТДС организуется отдельный набор.

Первый подход привлекателен в силу своей простоты, однако его применение грозит либо значительным снижением эффективности обработки, либо малообозримыми затруднениями в случае попыток разобраться с причинами, вызвавшими временную (возможно, надолго) блокировку активной ТДС – особенно в случае, когда нужно передавать срочные данные. Задержка может быть обусловлена, например, регулировкой потока, а посылка срочных данных – задействованием механизма обработки отказов. В итоге на практике с каждой ТДС ассоциируется свой набор очередей. Если теперь какая-либо активная ТДС окажется временно закрытой, то следует просто отложить обработку элементов в соответствующей очереди, пока не будет снят временный запрет. Поток сообщений через другие активные ТДС не будет задержан.

С каждым примитивом бывают связаны данные пользователя, которые, как правило, являются конкатенацией УИП, относящихся к вышележащим уровням. Протокольный объект, получив примитив службы (БУС), использует параметры, связанные с примитивом, а также информацию о текущем состоянии протокола, связанную с данным соединением, чтобы сформировать УИП данного уровня. Затем эта УИП объединяется с данными пользователя, связанными с поступившим примитивом, образуя БДП данного уровня. Именно этот БДП передается нижележащему уровню в поле данных пользователя требующегося в данном случае примитива.

Данные пользователя, связанные с примитивом, хранятся в буфере данных пользователя (БфДП). Именно в БфДП хранятся накопленные УИП (БДП) каждого из вышележащих уровней, так что содержимое именно БфДП передается физическим уровнем. Как известно, УИП каждого уровня определяется так, что в любой системе ее интерпретация однозначна. Поэтому вне зависимости от того, что из себя представляют БДП как структуры данных, они имеют в конечном счете форму цепочки октетов. Другими словами, каждый БфДП описывается как одномерный массив (array) октетов. Поле УказательНаБфДП каждого БУС является указателем адреса начала БфДП, содержащего данные пользователя, которые связаны с примитивом. Величина ДлинаБфДП используется в качестве индикатора числа октетов, содержащихся в данный момент в буфере. Каждый раз, когда очередной уровень присоединяет свою УИП к уже имеющемуся содержимому буфера, значение ДлинаБфДП увеличивается на требуемую величину. После поступления БфДП на физический уровень полученное в конечном счете число октетов пересылается из БфДП удаленной системе.

Структура отдельного протокола уровня в контексте всего пакета подсистемы связи изображена на рис.3.3.

Для обеспечения доступности на каждом уровне все БУС и все связанные с ними БфДП объявляются глобальными структурами. Обычно весь пул из БфДП для каждого уровня формируется при начальной инициализации системы и при этом указатели буферов объединяются в список. Каждый раз, когда появляется потребность в новом буфере, из списка извлекается указатель на очередной свободный буфер; каждый раз, когда буфер перестает быть нужным, указатель в список возвращается.

Для того, чтобы уровень мог обрабатывать несколько запросов на службу параллельно, с каждой активной ТДС должен ассоциироваться отдельный набор автоматных переменных (массив ТаблицаСостоянийСобытий нужен в единственном экземпляре). Для простоты на рис.3.3 показан только один набор. Для примера был выбран транспортный уровень, так что можно сопоставить фрагменты его описаний в форме на рис.3.3 и в представленной ранее форме конечного автомата.

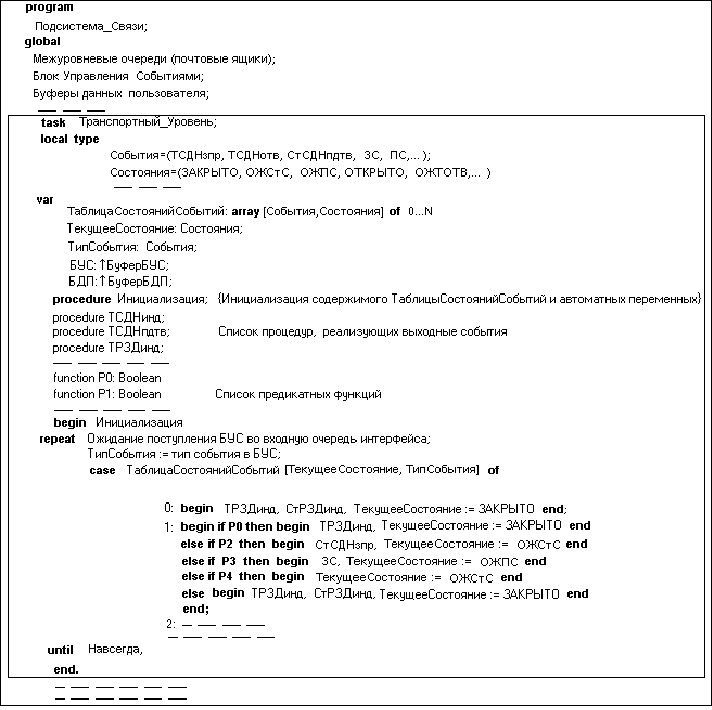


Рис. 3.3. Структура программа протокола уровня

Планирование задач осуществляется ядром реального времени. Если в данный момент некоторая задача (уровень) бездействует – ждет наступления входного события, то после того, как указатель на БУС помещается в одну из входных очередей задачи, ядро автоматически ее запускает. Сначала тип события из БУС (тип примитива) присваивается переменной ТипСобытия, которая в сочетании с переменной ТекущееСостояние используется для входа в массив ТаблицаСостоянийСобытий. Выбранная таким образом клетка массива определяет подлежащую запуску процедуру выходного события и новое ТекущееСостояние. Если же в клетке содержатся предикаты, то определяется список альтернативных комбинаций “выходное событие/новое состояние”. Предикаты устанавливаются или сбрасываются либо в процессе обработки отдельных событий, либо в результате обращения к специально написанным булевым функциям.

Прикладной процесс (ПП) пользователя выполняется на главной системе. Он получает доступ к службам распределенной информации с помощью полной подсистемы связи, применяя для этого имеющийся комплект примитивов, который соответствует элементу пользователя (ЭП). Подробнее о таких примитивах (примитивах пользователя) речь пойдет несколько позже. ЭП – пакет библиотечных процедур, которые подключаются к ПП до начала его работы в целях взаимного преобразования примитивов, применяемых в среде реальных систем (СРС) и примитивов, связанных с конкретным СЭПС в среде ВОС (СВОС).

Чтобы установить связь с определенным СЭПС, ЭП сначала получает свободный БУС, относящийся к данному СЭПС. Обычно это указатель адреса начала буфера в совместно используемой памяти. Затем ЭП записывает в БУС необходимые параметры, связанные с примитивом, после чего инициирует передачу указателя адреса БУС выбранному СЭПС с помощью программного обеспечения межпроцессорных связей. Любые данные пользователя, связанные с примитивом, заносятся в БфДП. Указатель адреса этого БфДП записывается в БУС, куда заносится также значение объема данных, содержащихся в БфДП. Получив БУС, выбранный СЭПС действует далее уже описанным образом.

Ясно, что объем данных пользователя, содержащийся в БфДП, по мере прохождения вниз сквозь все уровни будет изменяться и в конце может составить величину примерно от нескольких сотен октетов, если пересылается только УИП, до нескольких тысяч октетов, если передается, например, содержимое файла (по частям). Длина БфДП обычно фиксирована, так что в подобных случаях для обработки составных блоков используются списки буферов. Если очередной БфДП заполняется, то из списка свободных буферов выбирается указатель на свободный БфДП, и тот присоединяется к предыдущему.

Задача таймера атомарным образом взаимодействует с задачами (уровнями) подсистемы связи, используя механизм очередей. О применении такого механизма при асинхронных локальных взаимодействиях речь шла ранее. Так как всю подсистему обслуживает единственная задача таймера, эта задача имеет единую входную очередь. Кроме того, как и в случае межуровневых связей, для задачи таймера имеется всего один тип БУС.

Услуги, предоставляемые пользователю задачи таймера, включают в себя запуск и отмену таймера, а также индикацию тайм-аута. Подходящим набором примитивов представляется следующий: ТАЙМЕРстарт (ИДуровня, ИДтаймера, Время), ТАЙМЕРотмена (ИДуровня, ИДтаймера), ТАЙМЕРсрок (ИДтаймера).

Первые два (входных для задачи таймера) примитива имеют параметр ИДуровня, идентифицирующий в единой очереди уровень, порождающий эти примитивы. Кроме того, поскольку с каждым уровнем может быть связано несколько одновременно работающих таймеров, то используется, в данном случае для идентификации таймера в рамках уровня, параметр ИДтаймера. На практике с этой целью обычно используется идентификатор соединения. Параметр Время описывает то время, которое должно пройти, прежде чем задача таймера известит соответствующий протокольный объект о том, что закончился заданный временной интервал.

Для инициирования операций таймера протокольный объект сначала выбирает свободный БУС таймера и вписывает в него в качестве параметров идентификаторы уровня и таймера, а также требуемый интервал времени. Далее он инициирует передачу указателя БУС задаче таймера, выдавая для этого соответствующий межзадачный примитив локальному ядру реального времени. Получив запрос, задача таймера формирует строку таблицы, в которой указывается идентификатор таймера и связанный с этим таймером интервал времени (переменная-счетчик).

Задача таймера управляется прерываниями, поступающими от системных часов. Через определенный период она сначала проверяет наличие в ее входной очереди каких-нибудь БУС, ожидающих обработки. Если таковые имеются, то они подвергаются обработке. Далее, временные интервалы, связанные со всеми активными таймерами, уменьшаются на единицу. Если после этого оказывается, что временной интервал, связанный с каким-либо таймером, истек, то в свободный БУС записывается примитив ТАЙМЕРсрок с параметром, идентифицирующим этот таймер, и указатель БУС с помощью примитива межзадачной связи помещается во входную очередь (от таймеров) соответствующего уровня. В определенный момент по назначении задачи этого уровня к работе этот элемент входной очереди будет выбран и неделимым образом обработан.

Если после инициирования таймера протокольный объект, исходя из приема соответствующих ответов, принимает решение о выключении этого таймера, то он посылает примитив ТАЙМЕРотмена с надлежащими параметрами, используя БУС и примитив межпроцессорной связи, обеспечиваемый ядром реального времени. Когда задача таймера будет в следующий раз назначена к работе, она удалит соответствующий таймер из таблицы таймеров.

Вопросы, связанные с программной реализацией собственно прикладных процессов (ПП) пользователей, выходят далеко за рамки обсуждаемой тематики ВОС. Тем не менее, перед изучением функционирования различных СЭПС полезно хотя бы коротко рассмотреть организацию взаимодействия между средами ВОС (СВОС) и реальных систем (СРС).

К настоящему времени на разработку программного обеспечения и операционных систем, общение с которыми не подчиняется правилам ВОС, затрачены исключительно большие средства. Очевидна поэтому целесообразность такого подхода, который позволил бы за счет относительно малых дополнительных затрат (средств) осуществить эволюцию таких закрытых систем к системам открытым. Подход, принятый в МОС для достижения этой цели, опирается в основном на отделение СВОС от СРС.

Сначала для определенного приложения описывается виртуальная система и задается набор примитивов службы пользователя, который будет пользоваться этой системой. Все запросы и ответы пользователя на интерфейсе со СВОС относятся к этой виртуальной системе.

Далее между СВОС и ПП, т.е. СРС, создается дополнительное программное обеспечение (ЭП), которое выполняет преобразования в обе стороны между наборами примитивов служб, относящихся к виртуальной системе и к тем, с которыми работает пользователь в СРС, если такие преобразования необходимы. На практике ЭП является набором библиотечных процедур, подключаемых к ПП пользователя.

Различия в типах СЭПС и особенности реализаций приложений приводили, приводят и, очевидно, будут приводить в дальнейшем к большому разнообразию наборов примитивов пользователя. Обычно можно, во-первых, разделить ПП пользователей на ПП-инициаторы и ПП-респонденты (запрашивающая и отвечающая стороны соответственно) и, во-вторых, провести типизацию примитивов, которыми пользуются те и другие.

Примитивы на запрашивающей стороне можно отнести к одному из следующих трех типов: Посылка Сообщения (ПсС), Посылка сообщения с Извещением (ПсСИ), Посылка сообщения с Ответом (ПсСО). Параметры этих примитивов позволяют подключенному ЭП формировать соответствующие примитивы СЭПС. Кроме того, с каждым примитивом связывается параметр состояния, что позволяет ЭП информировать ПП об успешности или безуспешности запроса, и в последнем случае – о причине неудачи.

После передачи примитива ЭП операционная система главной системы приостанавливает ПП пользователя. Трем указанным типам примитива соответствует различная предыстория возобновления этого ПП.

Для примитива типа Посылка Сообщения (ПсС) ПП активизируется сразу после того, как подсистема связи возвращает параметр состояния, связанный с этим примитивом, который указывает либо на успешность посылки (из локальной системы), либо на ее безуспешность, сообщая о причине неудачи (рис.3.4, а).

Для примитива ПсС не поступает никаких сведений о том, получил ли предполагаемый корреспондирующий пользователь переданное сообщение. Поэтому обычно этот тип используется в различных связанных с СЭПС службах без подтверждения, выполняемых по установленной ранее ассоциации.

Для примитивов типа ПсСИ ПП пользователя активизируется только после того, как сообщение было доставлено указанному удаленному ПП, и извещение об успешной доставке поступило в систему запрашивающей стороны (рис.3.4, б). Поэтому обычно этот тип используется тогда, когда примитивы пользователя относятся непосредственно к службам СЭПС с подтверждением.

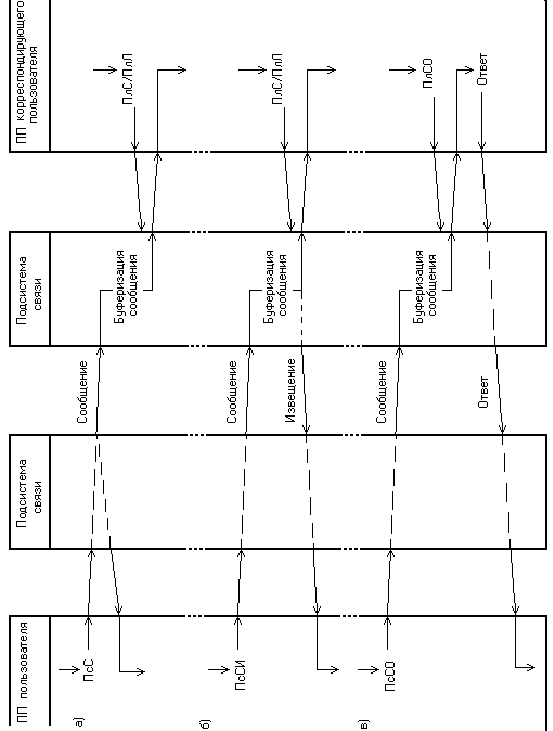


Рис. 3.4. Типы примитивов пользователя на запрашивающей и отвечающей сторонах

Для примитивов типа ПсСО ПП пользователя активизируется только после того, как сообщение было доставлено указанному удаленному ПП и последний прислал ПП запрашивающей стороны ответное сообщение (рис.3.4, в). Обычно этот тип используется тогда, когда примитив пользователя охватывает несколько примитивов СЭПС. При этом ПП активизируется только после поступления ответа, связанного с посланным сообщением (запросом). Поскольку параметры, связанные с примитивами этого типа, могут быть использованы для пересылки данных в обоих направлениях, этот тип называют также удаленным вызовом процедур (УВП).

На отвечающей стороне примитивы распадаются на следующие три типа: Получение Сообщения (ПлС), Получение от Любого (ПлЛ), Получение Сообщения с Ответом (ПлСО).

В примитиве типа ПлС ПП отвечающей стороны описывает тот (удаленный) ПП, которому он готов отвечать. После выдачи такого примитива ПП либо приостанавливается, пока от указанного ПП не поступит сообщения, либо немедленно активизируется, если уже имеется соответствующее ждущее сообщение. Обычно такой тип примитивов связывают со службами (с подтверждением или без подтверждения) конкретного СЭПС после того, как ассоциация уже установлена.

Примитивы типа ПлЛ подобны примитивам типа ПлС с той лишь разницей, что не осуществляется спецификация ПП, которого отвечающая сторона готова обслужить. Обычно этот тип используется тогда, когда ПП на отвечающей стороне, например, ПП-сервер, готов установить ассоциацию с новым ПП-клиентом. После выдачи примитива ПП либо приостанавливается, пока не поступит новое сообщение, либо немедленно активизируется, если уже имеется ждущее сообщение. Оба типа проиллюстрированы на рис.3.4, а и б.

Примитив типа ПлСО используется совместно с ПсСО для установления двусторонней связи между двумя ПП, т.е. для организации УВП. После получения из входной очереди сообщения (запроса) ПП активизируется и спустя некоторое время вырабатывает ответ, который передается подсистеме связи и доставляется удаленному ПП на запрашивающей стороне (рис.3.4, в).

Ясно, что между двумя взаимодействующими сторонами (процессами) должна быть реализована какая-то форма буферов для хранения сообщений. В этих целях может быть использована память с произвольной выборкой, применение которой для БУС и БфДП описано выше. Эта память входит в состав подсистемы связи и непосредственно адресуется как процессором связи, так и главным процессором.

## Управление ВОС

В рамках управления ВОС выделены следующие дисциплины: управление при отказах (УО), управление учетом (УУ), управление конфигурацией и именами (УКИ), управление эффективностью функционирования (УЭФ), управление безопасностью (УБ). В этих рамках услуги по обмену информацией предоставляют информационные службы управления (ИСУ), которые являются службами прикладного уровня.

К ресурсам ВОС относятся средства, необходимые для функционирования протоколов ВОС, и данные управления, обеспечивающие информацию о состоянии функционального окружения ВОС.

УО – совокупность средств, инициируемых в результате ненормальной работы функционального окружения ВОС. Отказы проявляются в виде сбоев[[13]](#footnote-13) при функционировании открытой системы. УО предоставляет средства для обслуживания и анализа файлов регистрации сбоев, приема и обработки уведомлений об обнаружении сбоев, административного сопровождения сбоев, выполнения последовательностей тестов, исправления отказов.

Ресурсами ВОС, имеющими отношение к отчетам о сбоях, являются N-объекты и N-соединения. Тип сбоя определяется независимо от типа ресурса. Имеется четыре класса сбоев, различаемых по степени воздействия сбоя на ресурс: невосстанавливаемый, функциональный, временный и исправимый.

УУ – совокупность средств, обеспечивающих определение стоимости ресурсов и оплаты за их использование. УУ предоставляет средства для оповещения пользователя об оплате или объеме потребления ресурсов, установки учетных лимитов на использование ресурсов, определения стоимости использования совокупности ресурсов. Служба УУ работает с двумя уровнями, на которых доступна информация учета: с сетевым, учитывающим использование среды связи, и с прикладным, учитывающим оплату за использование среды связи и ресурсов оконечных систем.

УКИ – совокупность средств управления, идентификации, сбора и предоставления данных, обеспечивающих непрерывное функционирование служб взаимосвязи. Включает средства установки параметров открытых систем, инициализации и закрытия ресурсов ВОС, сбора данных о состоянии открытых систем, обеспечения конкретными данными по запросу.

УЭФ – совокупность средств, необходимых для оценки поведения ресурсов ВОС и эффективности деятельности по взаимосвязи. Сюда относится сбор статистических данных, необходимых для обслуживания и анализа файлов регистрации состояний систем.

УБ – совокупность средств защиты ресурсов ВОС, т.е. средств санкционирования, контроля доступа, шифрования и управления ключами, аутентификации, обслуживания и анализа регистрационных файлов безопасности.

Реализованные в открытых системах функции управления системами, которые используют ИСУ, обобщенно называются прикладными процессами управления системами (ППУС, SMAP). Часть таких процессов, относящаяся к передаче данных в рамках ВОС, определяется как прикладной объект управления системами (ПОУС).

Элементы ИСУ являются примером специальных элементов прикладной службы (СЭПС), определенных в ЭМВОС.

ПОУС может быть представлен несколькими элементами разных типов: ОЭПС, СЭПС, другими элементами прикладной службы. Взаимодействие двух ПОУС разных открытых систем предполагает наличие между ними ассоциации, устанавливаемой с помощью ЭСУА. Обмен информацией управления по своей природе является в основном двусторонним, где каждая сторона при взаимодействии в рамках одного элементарного обмена выполняет роль инициатора или ответчика.

По крайней мере, один ППУС должен быть локализован в открытой системе, участвующей в обмене информацией управления. Связь данного ППУС с удаленным ППУС с целью передачи этой информации обеспечивается через ПОУС. По соглашению с удаленным ППУС может использоваться также любая другая служба прикладного уровня. Предполагается, что ППУС обеспечивает также интерфейс с локальным системным администратором, если тот существует в данной открытой системе. Такой интерфейс может использоваться в ходе выполнения чисто локальных функций, например инициализации системы.

В зависимости от поддерживаемого в каждой открытой системе подмножества функций управления взаимоотношения между ППУС могут изменяться от полностью симметричных до асимметричных. Взаимодействующие при выполнении некоторой задачи управления ППУС могут предварительно осуществлять процедуру согласования своих возможностей по управлению во время фазы установления прикладной ассоциации.

Управление ассоциацией прикладных объектов управления системами (ПОУС) обеспечивается услугами У-ИНИЦИАЛИЗАЦИЯ, У-ЗАВЕРШЕНИЕ, У-РАЗРЫВ, опирающимися на поддержку услуг ЭСУА Пк-АССОЦИИРОВАНИЕ, Пк-ОСВОБОЖДЕНИЕ, Пк-Пл-РАЗРЫВ и Пк-Пс-РАЗРЫВ. Помимо этих трех услуг управления ассоциацией, обмен управляющей информацией обеспечивают еще несколько услуг СЭПС ИСУ, чье применение также опирается на использование услуг ОЭПС, конкретнее, на услуги элемента службы удаленных операций (ЭСУО).

Для передачи запроса на пересылку информации (обычно статистической) от одного ПОУС к другому используется подтверждаемая услуга У-ПОЛУЧЕНИЕ.

Подтверждаемая услуга У-УСТАНОВКА обеспечивает возможность передачи требования другому ПОУС на установку значений атрибутов открытой системы. Установка является основным механизмом управляющих воздействий на ресурсы.

Подтверждаемая услуга У-ДЕЙСТВИЕ обеспечивает возможность передачи требований ПОУС другой открытой системы на выполнение операций. Эта услуга должна использоваться в тех случаях, когда на выполнение операций нельзя воздействовать с помощью установки значений атрибутов.

Индицируемая (неподтверждаемая) услуга У-СОБЫТИЕ обеспечивает уведомление о событии. Используется ПОУС для передачи асинхронных сообщений о ресурсах другому ПОУС. Подтверждаемый вариант этой услуги используется ПОУС для передачи другому ПОУС таких асинхронных сообщений о ресурсах, на которые необходим ответ.

Подтверждаемая услуга У-СРАВНЕНИЕ обеспечивает возможность передачи требования ПОУС другой открытой системы на сравнение значений атрибутов этой открытой системы с заданными значениями и возвращения результатов сравнения.

В число параметров примитивов описанных услуг входят идентификаторы устанавливаемых параметров систем, идентификаторы действий, идентификаторы уровней, идентификаторы событий, величины соответствующих параметров и характеристики действий.

Применяемые услуги ЭСУО – это услуги УО-ВЫЗОВ, УО-РЕЗУЛЬТАТ и УО-ОШИБКА. Услуги ЭСУО предполагают, естественно, использование услуг уровня представлений.

Обычно в каждой подсети имеется система, решающая задачу управления этой подсетью, в то время как остальные ее системы имеют лишь минимальный необходимый набор функций управления. Менеджер (оператор) всей сети управляет подсетью с помощью этой системы, опираясь на централизованный для подсети ППУС – прикладной процесс менеджера сети (ППМС). В его состав входя функциональные компоненты, отвечающие дисциплинам управления: УО, УУ, УКИ, УЭФ и УБ. Соответствующий СЭПС – это элемент менеджера управления системой (ЭМУС), СЭПС в остальных системах подсети – элементы агентов управления системой (ЭАУС) (рис.3.5). Специфика этих элементов, как видим, состоит в том, что они имеют интерфейс не только с ОЭПС (такой интерфейс имеет всякий СЭПС, и его назначение понятно), но и интерфейс с каждым уровнем. Через этот последний интерфейс осуществляется взаимодействие с функциями управления уровнем, упоминавшимися при рассмотрении каждого из них.

Каждый уровень помимо автоматных переменных и прочей информации, относящейся к его текущему функциональному состоянию, поддерживает еще целый ряд переменных, касающихся различных статистических данных и требуемых для целей управления.

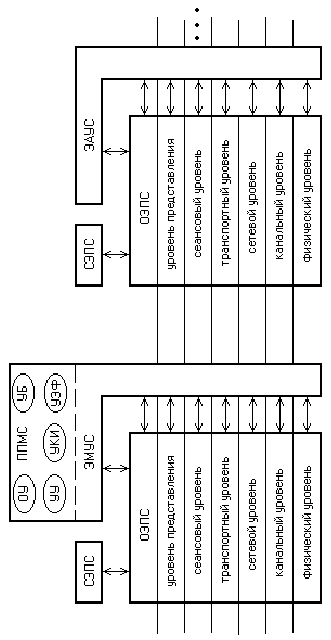


Рис. 3.5. Организация управления в подсети

К примерам переменных такого рода относятся:

* число полученных отрицательных Пк-АССОЦИИРОВАНИЕ-ответ, число полученных и посланных ЭСУА-БДП “прекращение” – для ЭСУА (ОЭПС);
* число посланных и полученных ПдСДНотв(-) (ответов “отвергнуто” на запрос на соединение уровня представлений) и Пд-БДП, содержащих код причины; число полученных непонятных Пд-БДП – для уровня представления;
* число полученных и посланных Сн-БДП “отказано”, число полученных и посланных Сн-БДП “прекращение” – для сеансового уровня;
* число имевших место ошибок протокола; число случаев тайм-аута, связанных с переданными Т-БДП, число Т-БДП с неверными контрольными суммами – для транспортного уровня;
* число посланных и полученных Ст-БДП; число аннулированных в силу неизвестной Ст-ТДС – для сетевого уровня;
* число столкновений и попыток повторных передач (МДКП/ОК, CSMA/CD); число безуспешных передач маркера (шина, управляемая маркером) – для уровня логического звена (подуровня управления логическим каналом) – в рамках ЭМВОС соответствует подуровню канального уровня.

ЭАУС должен обеспечивать средства доступа к таким специальным функциональным параметрам уровня, как:

* предел окна – транспортный уровень;
* временные интервалы таймеров – все уровни;
* содержимое таблиц маршрутизации – сетевой уровень;
* допустимый максимум повторных передач (МКДП/ОК);
* желательное время обращения маркера (шина, управляемая маркером).

С дисциплинами управления связана совокупность действий (директив), с помощью которых ППМС выполняет операции с отдельными переменными уровней, относящимися к состоянию и статистике системы. Так, на уровне всей системы ППМС может запросить у ЭАУС вернуть в исходное состояние объекты уровней в связи с реконфигурацией. В рамках одного уровня ППМС может запросить у ЭАУС конкретную статистику и т.д.

Ясно, что все передачи, связанные с выполнением таких действий, инициируются ППМС. Возможны также ситуации, при возникновении которых ЭАУС должны немедленно информировать ППМС о наступлении в присоединенной системе некоторых событий. Например, о достижении в некотором уровне определенными статистическими параметрами заранее установленных пороговых значений.

О сути протокола взаимосвязи менеджера и агента с опорой на услуги компонентов ОЭПС – ЭСУА и ЭСУО – уже шла речь ранее.

## Служба справочника

Справочная служба (служба справочника) ВОС играет значительную роль во взаимодействии открытых систем. Справочник обеспечивает получение информации, необходимой прикладным процессам ВОС, процессам управления ВОС, другим уровневым объектам и телекоммуникационным средствам. К обеспечиваемым службой справочника возможностям относятся, например, использование ориентированных на пользователя имен для обращения к объектам, поддержание соответствия между именем и адресом.

Одна из причин использования имени – стремление освободить пользователя от необходимость знать конфигурацию используемой сети, ее изменений в ходе подключений/отключений подсетей, например, ЛС, и их компонентов, перемещений прикладных процессов из одного пункта сети в другой. Имена объектов, очевидно, должны быть уникальными, так что в больших межнациональных сетях “имя” может состоять из нескольких компонентов (атрибутов), образующих формат “Страна.Подсеть.Система.Имя”. В итоге, если Имя уникально в системе, то “имя” с гарантией будет уникальным и для всей СВОС.

Возможны различные варианты размещения справочной информации в сети – начиная от варианта организации централизованного единственного справочника на всю сеть и кончая вариантом полной его децентрализации, когда каждая система хранит и поддерживает свою копию такого справочника. Оба крайних варианта имеют свои достоинства и недостатки. Обычно на практике применяется некоторый компромисс. Принятая МОС регламентация логической структуры базы справочной информации позволяет не учитывать тот факт, что справочник является скорее распределенным, нежели централизованным.

База справочной информации (БСИ) состоит из “входов справочника”, каждый из которых содержит информацию об одном объекте. Имеется два вида входов: объектные и альтернативные. Для каждого конкретного объекта существует только один объектный вход, содержащий первичную информацию об этом объекте. Дополнительно для данного объекта могут присутствовать несколько альтернативных входов.

Входы справочника организованы в дерево, называемое информационным деревом справочника (ИДС). Вершины ИДС, кроме корня, являются входами. Альтернативные входы всегда являются листьями ИДС. Дуги определяют взаимоотношения между вершинами. Дуга от вершины А к вершине В означает, что вход в вершине А является “непосредственно старшим” для входа в вершине В и, наоборот, вход в вершине В является “непосредственно подчиненным” для входа в вершине А.

Каждый вход состоит из множества атрибутов, имеющих заданный тип и одно или более значений. Атрибуты представляют собой отдельные элементы информации, каждый из которых описывает конкретную характеристику объекта. Тип атрибута идентифицирует класс представляемой данным атрибутом информации, например телефонный номер. В каждом входе имеется не более одного атрибута конкретного класса; при этом один атрибут может иметь несколько значений.

Некоторые типы атрибутов регламентируются международными стандартами, другие определяются национальными или частными организациями. Несколько типов атрибутов справочник знает и использует для своих целей, однако для большинства из них он семантики не знает. При этом тип атрибута включает интерпретацию, которая содержит информацию, достаточную справочнику для сравнения значений атрибутов.

Примерами обязательных для каждого входа атрибутов является информация именования, такая, как персональное имя, и адресная информация, такая, как номер телефона.

Каждый вход имеет атрибут “относительное различаемое имя” (ОРИ), значение которого выбирается таким образом, чтобы относительные различаемые имена всех входов с некоторым одним старшим входом были различными. ОРИ выбирается при создании входа и может быть при необходимости модифицировано.

“Различаемое имя” данного объекта определяется как последовательность ОРИ соответствующего объектного входа и всех его старших входов в порядке возрастания старшинства. Интерпретация различаемого имени относится к регламентируемой стандартом МОС и входит в группу интерпретаций идентификации объекта.

Каждый пользователь при доступе к справочнику представлен агентом пользователя справочника (АПС). Сам справочник представлен совокупностью системных агентов справочника (САС). Опуская детализацию описания, оба типа агентов называют также просто агентами справочной службы (АСС).

В состав прикладного объекта АПС входит СЭПС – элемент службы доступа к справочнику (ЭСДС), а также ОЭПС: ЭСУА – элемент службы управления ассоциацией и ЭСУО – элемент службы удаленных операций. В состав прикладного объекта САС входит СЭПС: элемент системной службы справочника (ЭССС) и, возможно, ЭСДС, а также ОЭПС: ЭСУА и ЭСУО.

Доступ к справочнику обеспечивается парой ЭСДС, один из которых является частью прикладного объекта АПС, а другой – прикладного объекта САС. Вместе они образуют поставщика услуг по доступу к справочнику. Системная служба справочника обеспечивается парой ЭССС, которые являются частями прикладных объектов САС двух открытых систем.

Предоставляемые пользователям возможности собраны в функциональные группы:

* ядро – работа с именами (верификация имен: проверка уникальности в системе имени, представленного пользователем), работа с множеством объектов, фильтрование, управление службой, контроль доступа (проверка права пользователя на выполнение конкретного запроса), чтение входа;
* исследование справочника – идентификация подчиненных объектов, описание подчиненных имен;
* справки по спискам – перечисление членов группы, проверка на принадлежность группе, проверка множества на принадлежность группе;
* управление объектом – добавление и исключение объекта, модификация ОРИ, типов атрибута, значения атрибута;
* управление контролем доступа – модификация списка контроля доступа, предоставление списка контроля доступа;
* управление альтернативой – добавление и исключение альтернативного входа, модификация указателя альтернативного входа, предоставление списка альтернативных входов;
* дублирование – инициация, завершение, возобновление дублирования.

Работа с именами, с множеством объектов, фильтрование, управление службой и контроль доступа проводятся в ходе предоставления многих услуг справочной службы. Остальные возможности связаны с конкретными услугами.

Услуги функционального ядра всегда доступны. Остальные возможности в любой комбинации могут обеспечиваться поставщиком услуг дополнительно. Если доступна какая-либо функциональная группа, то доступны все входящие в нее возможности.

Доступность возможностей определяется уровнем сложности конкретного АСС. В типичном варианте множества подтверждаемых услуг справочной службы в набор примитивов входят примитивы определения адреса и добавления/удаления/изменения элемента содержимого БСИ. В ряду параметров соответствующих примитивов запросов и ответов, помимо специфицированных имен и адресов, могут использоваться:

* идентификатор запроса (позволяет прикладному процессу (элементу пользователя) различать запросы, ожидающие ответа справочника);
* идентификатор подлинности (позволяет удостовериться в том, что пользователь имеет право инициировать внесение изменений в БСИ);
* код ошибки.

Принята следующая типизация ошибочных ситуаций: ошибка контроля доступа, ошибка в атрибуте, ошибка аутентификации, ошибка имени, ошибка службы, ошибка при обновлении, ошибка дублирования, ошибка соединения, неверный САС.

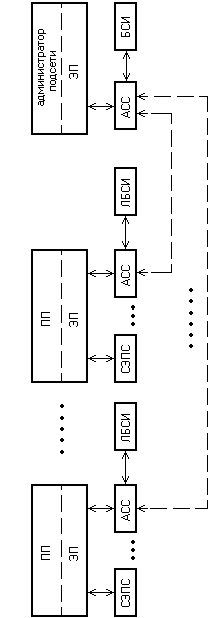


Рис. 3.6. Схема справочной службы подсети

В реализуемой на практике компромиссной схеме размещения справочной информации в сети АСС, получив запрос на определение адреса по специфицированному символическому имени, обращается к своей локальной БСИ (ЛБСИ). Кроме того, обычно в каждой подсети имеется еще и АСС, ответственный за хранение и поддержание экземпляра полной общесетевой БСИ. К этому АСС обращаются прочие агенты подсети в случае, когда они оказываются не в состоянии самостоятельно ответить на запрос (рис.3.6).

Для обеспечения взаимосвязи между АСС (между АПС и САС или между двумя САС) должна быть установлена ассоциация между соответствующими прикладными объектами. Общий прикладной сервис (ЭСУА) обсуждался ранее (см. разд. 2).

Так как каждый локальный АСС поддерживает свою версию БСИ, то центральный АСС подсети, произведя по запросу изменение БСИ и послав подтверждение об этом агенту-инициатору, должен информировать о выполненном изменении и все прочие локальные АСС.

Для обеспечения взаимодействия АПС с САС (и САС с другим САС) протокол доступа к справочнику (и, соответственно, системный протокол справочника) использует службу удаленных операций (ЭСУО). В свою очередь, служба УО обеспечивается протоколом УО в сочетании со службой и протоколом управления ассоциацией, представительной службой и, возможно, при использовании службы и протокола надежной передачи.

## Вопросы к разделу 3

1. Приведите пример структуры межуровневых данных.
2. Составьте схему организации управления в подсети.
3. Дайте общее описание службы справочника.
4. Предложите основу практической реализации свойства атомарности примитивов.
5. Оцените плюсы и минусы полностью централизованной и полностью децентрализованной службы справочника.
6. Каким образом можно оценить затраты ресурсов, необходимых для поддержки взаимосвязи прикладных процессов?
7. Перечислите основные задачи, решаемые управлением ВОС.
8. В чем заключается и чем обусловлена специфика элементов информационных служб управления?
9. Можно ли ограничится лишь адресацией или лишь именованием объектов? Ответ обоснуйте.
10. Иерархизируйте по важности функциональные возможности службы справочника.
11. Рассмотрите последствия отказа какой-либо конкретной функциональной группы средств управления ВОС.

# СПИСОК ЛИТЕРАТУРЫ

1. Протоколы информационно-вычислительных сетей: Справочник / С.А.Аничкин, С.А.Белов, А.В.Бернштейн и др.; под ред. И.А.Мизина, А.П.Кулешова. – М.: Радио и связь, 1990. – 504 с.
2. Сервис открытых информационно-вычислительных сетей: Справочник / С.С.Зайцев, М.И.Кравцунов, С.В.Ротанов. – М.: Радио и связь, 1990. – 240 с.
3. Халсалл Ф. Передача данных, сети компьютеров и взаимосвязь открытых систем: пер. с англ. – М.: Радио и связь, 1995. – 408 с.
4. Взаимосвязь открытых систем: пер. с англ. / Под ред. С.И.Самойленко. // Тематический выпуск ТИИЭР. Т. 71. N° 12. – М.: Мир, 1983. – 175 с.
5. Архитектура, протоколы и тестирование открытых информационных сетей: Толковый словарь / В.Ф.Баумгарт, С.П.Волкова, А.В.Гнездовский и др.; под ред. Э.А.Якубайтиса. – М.: Финансы и статистика, 1989. – 192 с.

1. Или требованиям Рекомендации X.200 MKKTT (CCITT) – Международного консультативного комитета по телеграфии и телефонии. [↑](#footnote-ref-1)
2. Используется также аббревиатура N-СТД - сервисная точка доступа N-уровня. [↑](#footnote-ref-2)
3. Используется также аббревиатура N-ПБД - N-протокольный блок данных. [↑](#footnote-ref-3)
4. Используются также аббревиатуры БДИ - блок данных интерфейса и СБД - сервисный блок данных. Границе между уровнями N+1 и N соответствуют N-БДИ и N-БДС. [↑](#footnote-ref-4)
5. ООД - оконечное оборудование [обработки] данных, соответствует DTE - data terminal equipment;

   АКД - аппаратура [окончания] канала данных, соответствует DCE - data circuit - terminating equipment. [↑](#footnote-ref-5)
6. В отечественной литературе вместо термина “канальный уровень” часто используется термин “уровень звена передачи данных”. [↑](#footnote-ref-6)
7. Точнее отвечающим функциональной сущности является термин “уровень представления”, а не часто применяемый по созвучию с названиями других уровней термин “представительный уровень”. [↑](#footnote-ref-7)
8. Элементы процедур различных классов не всегда одинаковы. [↑](#footnote-ref-8)
9. В скобках здесь и далее приводятся варианты термина, встречающиеся в литературе. [↑](#footnote-ref-9)
10. В ряде источников используется аббревиатура CASE - common application service elements, термину ASE может ставиться в соответствие термин «множество прикладных услуг», термину UE - термин «услуги пользователя», «интерфейс пользователя», «агент пользователя». Элемент пользователя также может не определяться, что подразумевает выполнение его функций прикладным процессом. [↑](#footnote-ref-10)
11. Используется также термин базовое ядро ACSE. [↑](#footnote-ref-11)
12. Вызов прикладного объекта выполняет функции прикладного объекта для конкретного случая обмена информацией. [↑](#footnote-ref-12)
13. Под сбоем понимается факт несоответствия функционирования системы ее спецификации, под отказом - механическая или алгоритмическая причина неправильного функционирования. [↑](#footnote-ref-13)