

ОСНОВЫ ПОСТРОЕНИЯ ОТКРЫТЫХ СИСТЕМ

Учебное пособие / М., ИРЭ РАН. 1999 - 97 с.

Батоврин В.К. (введ., гл.2, закл.), Дешко И.П. (гл.6), Журавлев Е.Е. (гл.1), Коваленко С.М. (гл.3), Кочемасов А.В. (гл.5), Лебедев А.В. (гл.2), Мордвинов В.А. (гл.6), Олейников А.Я. (введ., гл.7, закл.), Петров А.Б. (гл.1), Смирнов Н.А. (гл.3), Теряев Е.Д. (гл.5), Тювин Ю.Д. (гл.3). Под ред. Олейникова А.Я. -

Излагаются основные подходы к анализу, синтезу и построению открытых информационных, вычислительных и телекоммуникационных систем.

Учебное пособие предназначено для широкого круга студентов и аспирантов, обучающихся по направлениям подготовки 190000, 200000, 210000, 220000 и других.

Печатается в рамках проекта А 0086 "Центр открытых систем" Федеральной целевой программы "Интеграция".

Москва 1999

©ИРЭ-МИРЭА

Содержание.

Введение.....	6
Глава 1. Информационные системы.....	11
1.1. Понятие информационной системы.....	11
1.2. Требования к информационным.....	13
системам.....	13
1.3. Информационная модель и ее элементы.....	16
1.4. Разработка математической и программной моделей ИУС.....	21
Выводы по главе 1.....	23
Глава 2. Информационно-измерительные системы.....	25
2.1. Общие сведения.....	25
2.2. Модель взаимосвязи открытых систем.....	27
2.4. Передача данных в ИИС.....	35
2.5. Магистрально – модульные системы.....	38
Глава 3. Вычислительные машины системы и сети.....	47
3.1. Вычислительные системы.....	49
3.2. Вычислительные сети.....	50
3.3. Сетевые службы и услуги.....	51
3.4. Многоуровневый подход. Интерфейс. Протокол.....	52
3.5. Модель OSI.....	53
Глава 4. Информационно-вычислительные системы и сети.....	57
4.1. Понятие информационно-вычислительные системы и сети.....	57
4.2. Основные требования по обеспечению мобильности программ и данных в открытых ИВС.....	58
4.3. Стандарты, поддерживающие создание мобильных прикладных программ.....	62
4.4. Базы данных в концепции открытых систем.....	65
5.1. Свойства интерфейсовых систем информационного обмена и локальных вычислительных сетей.....	69
5.1.1. Определение СИО и ЛВС.....	69
5.1.2. Анализ СИО и ЛВС как больших систем.....	70
5.1.3. СИО и ЛВС, реализующие заданный класс алгоритмов.....	71
обмена в условиях возникновения неисправностей.....	71
Глава 6. Глобальные информационные и телекоммуникационные сети.....	72
6.1. Согласование протоколов в составных сетях.....	72
6.2. Цифровые сети с интеграцией служб.....	74
6.3. Технология АТМ, широкополосные цифровые сети.....	76
с интеграцией служб.....	76
Глава 7. Основы концепции открытых систем.....	80
7.1. Области применения ТОС.....	82
7.2. Эталонная модель среды открытых систем.....	83
7.3. Понятие профиля.....	86
7.4. Технологический цикл построения открытых систем.....	89
7.5. Разработка профилей.....	93

Выводы по главе 7.....	95
Термины и определения :	95
Список сокращений :.....	96

УДК 681.3

Батоврин В.К. (введ., гл.2), Дешко И.П. (гл.6), Журавлев Е.Е. (гл.1), Коваленко С.М. (гл.3), Кочемасов А.В. (гл.5), Лебедев А.В. (гл.2), Мордвинов В.А. (гл.6), Олейников А.Я. (введ., гл.7), Петров А.Б. (гл.1), Смирнов Н.А. (гл.3), Теряев Е.Д. (гл.5), Тювин Ю.Д. (гл.3). Под ред. Олейникова А.Я. - Теоретические основы построения открытых систем: Учебное пособие / М., ИРЭ РАН. 1999 - 000 с.

Излагаются основные подходы к анализу, синтезу и построению открытых информационных, вычислительных и телекоммуникационных систем.

Учебное пособие предназначено для широкого круга студентов и аспирантов, обучающихся по направлениям подготовки 190000, 200000, 210000, 220000 и других.

Табл.- 000. ил.- 000. библиогр.- 000.

Печатается в рамках проекта А 0086 "Создание Центра открытых систем" Федеральной целевой программы "Интеграция".

Введение

Своевременная, полная и надежная информация представляет собой необходимый компонент для принятия обоснованных решений, являясь фактором, обеспечивающим той или иной группе или организации решающее преимущество в любой из областей современной деятельности. Поэтому развитие информационных и телекоммуникационных технологий в настоящее время следует рассматривать как необходимое условие существования и поддержания функционирования как мировой системы хозяйства в целом, так и отдельных национальных инфраструктур.

Развитие новых технологий в сфере информатизации и телекоммуникаций и формирование глобального информационного пространства обусловливается происходящим в течение последних 20 лет фантастическим прогрессом в области аппаратно-программных компонент компьютерной техники и аппаратуры связи. Прежде всего, к этому следует отнести внедрение в массовое производство дешевых и вместе с тем достаточно мощных микропроцессоров (МП) и других компонент вычислительной техники, предназначенной, в основном, для персональных компьютеров, а также для серверов и рабочих станций. Характеристики этих компонент совершенствуются настолько стремительно, что разработчики зачастую не успевают разрабатывать для них приложения. Так, по заявлению руководства компании Интел, уже к маю 1997 она имела опытные образцы процессоров с тактовой частотой 1 ГГц и специалисты компании уверены, что им удастся увеличить эту цифру на порядок к 2011 г. Не отстают и другие фирмы. Например, в настоящее время проводятся разработки бездисковых массивов энергонезависимой памяти со временем выборки порядка десятков нс, которые смогут заменить обычные жесткие диски. Компании Microsoft, Compaq и Oracle заключили соглашение о согласовании стандартов и начале производства новых типов модемов и связной аппаратуры на основе технологии ADSL (Asymmetric Digital Subscriber Line), что уже сегодня позволяет конечным пользователям увеличить скорости подключения к Internet на порядок и более, используя обычные телефонные линии. Планируется также запуск серий спутников, предназначенных для обеспечения непосредственной связи с конечными пользователями через космос.

Процесс массовой, глобальной компьютеризации последних лет, несмотря на впечатляющее развитие микроэлектроники, не был бы, однако, возможен, если бы в свое время фирма IBM не заложила в своих первых моделях персональных компьютеров IBM PC принцип "открытой" архитектуры¹. Соблюдение этого принципа рядом независимых производителей при-

¹ Под "открытой" архитектурой тогда понималось следующее: полное описание компьютера является открытым и доступным для независимых разработчиков, его шина используется многими адаптерами периферийных устройств, ее протокол также подробно описан, правила взаимодействия программ с периферийными устройствами являются рекомендованными стандартами de-facto, процессор обладает некоторым стандартным набором команд.

водило автоматически к совместимости форматов файлов и обмена данными, взаимозаменяемости деталей и узлов, возможности использования одних и тех же программных продуктов без какой-либо дополнительной адаптации, и т.д. Все это, наряду с рядом удачных технических решений, примененных ПЗМ в своих "персоналках", привело к быстрому росту выпуска PC-совместимых компьютеров и последовавшим за этим процессом массовой компьютеризации, затронувшим практически все сферы человеческой деятельности. В условиях массового выпуска такие системы оказались относительно дешевы², а их дальнейшее развитие в совокупности с необходимостью их интеграции в локальные/глобальные сети и информационно-вычислительные комплексы, имеющие в своем составе аппаратуру от различных производителей, способствовало развитию подхода, называемого ныне "принципами открытых систем".

На сегодня необходимым условием продвижения в сфере информационных технологий является широкое внедрение стандартов и технологий открытых систем, используемых как для аппаратных средств, так и для программных продуктов. Построение программного обеспечения (ПО) вычислительных и информационных комплексов, основанных на идеологии открытых систем, позволяет успешно решать задачи переносимости ПО на платформы различных производителей, проблемы взаимозаменяемости узлов и устройств и, что самое главное, обеспечивает интеграцию устройств и пользователей в различные информационно-вычислительные и телекоммуникационные сети. Следует особо подчеркнуть то обстоятельство, что на сегодняшний день успешная реализация сколько-нибудь существенных проектов в области информационно-вычислительной техники, управления, информатизации и телекоммуникаций не представляется возможной без согласования разработок с существующими стандартами в области открытых систем и, в ряде случаев, разработки новых стандартов.

В условиях перехода к интегрированным вычислительно-телекоммуникационным системам принципы открытых систем составляют основу технологии интеграции, создания отраслевых, региональных и национальных информационных инфраструктур и их взаимодействия в глобальном масштабе. Таким образом можно сделать вывод, что технология открытых систем в настоящее время является той рабочей средой, в рамках которой происходит развитие приоритетных информационно-телекоммуникационных технологий, средств телекоммуникаций и вычислительной техники.

На сегодня не существует однозначно устоявшегося определения термина "открытые системы". Различные организации формулируют его по-разному, исходя из своих конкретных задач. Согласно IEEE, например, "открытая система - это исчерпывающая и последовательная совокупность меж-

² Для сравнения: на июль 1999 розничная цена микросхемы 400-МГц процессора для платформы SUN составляла около 7000 долл., в то время как МП 333-МГц кристалл разработки IBM для PC-совместимых компьютеров стоил менее 40 долл.

дународных стандартов в области информационных технологий и функциональных профилей стандартов, которая специфицирует интерфейсы, службы и поддерживающие форматы для достижения взаимодействия и переносимости приложений, данных и персонала" (подробнее этот вопрос будет рассмотрен в главе 1).

Главное здесь – переход от множества платформ, поддерживаемых только их создателями, к общепринятым стандартам, поддерживаемым всем компьютерным сообществом. Масштабный переход к архитектурам и технологиям открытых систем начался более 10 лет назад и был обусловлен, в основном, двумя проблемами.

- (1) – проблемой мобильности программ и массивов данных,
- (2) - проблемой создания распределенных информационных инфраструктур, обеспечивающих организацию удаленного взаимодействия программно-аппаратных средств и массивов данных.

В качестве классического примера открытой системы можно привести операционную систему UNIX, базовая версия которой была разработана в конце 60-х г.г. К. Томпсоном и Д. Ритчи. Несмотря на большое количество разных реализаций UNIX, их форматы файлов полностью совместимы друг с другом, и, как правило, совместимо также программное обеспечение. На основе UNIX систем легко могут быть построены локальные и глобальные сети практически любой конфигурации.

Принципы открытых систем применяются в настоящее время при построении большинства классов систем: вычислительных, информационных, телекоммуникационных, систем управления в реальном масштабе времени, встроенных микропроцессорных систем. В условиях перехода к интегрированным вычислительно-телекоммуникационным системам принципы открытых систем составляют основу технологии интеграции.

За рубежом работы в области открытых систем ведутся весьма активно и обладают, как правило, приоритетным финансированием. Значительная их часть посвящена вопросам создания и применения профилей (профиль есть согласованный набор базовых стандартов необходимых для решения некоторого класса задач). Идет постоянное обновление документов, обусловленное обновлением старых и разработкой новых стандартов.

На Западе также проводится большой объем работ по сертификации систем и компонентов на соответствие принципам открытых систем, без чего реализация технологии открытых систем (ТОС) невозможна.

К сожалению, объем работ по развитию и применению открытых систем в Российской Федерации не соответствует важности проблемы и значительно меньше зарубежного, хотя вопрос о необходимости форсированного и комплексного развития открытых систем был поставлен Российской академией наук еще в 1993 г. Тем не менее, совместным приказом-постановлением Миннауки России и РАН была создана рабочая группа с участием представителей восьми ведомств, подготовившая Концепцию и проект программы "Развитие и применение открытых систем". Основным выводом рабочей группы состоит в том, что принципы открытых систем должны составить научно-

методическую основу работ по информатизации России. К настоящему времени, несмотря на ряд объективных трудностей, достигнут прогресс и по ряду других вопросов. В то же время некоторые ключевые проблемы (например, проблема сертификации на соответствие принципам ТОС) не имеют пока удовлетворительного разрешения.

К числу нерешенных пока вопросов относятся и проблемы образования, специализации и подготовки кадров в области открытых систем. Для их успешного разрешения, в качестве первого шага в рамках программы 'Интеграция' подготовлен настоящий курс "Открытые информационные, телекоммуникационные и вычислительные системы". При его разработке авторы придерживались рабочей концепции, что разработка спецкурса по открытым системам "с нуля" является достаточно сложной задачей и, кроме того, необходимо дополнительное время на его апробацию. Поэтому, в качестве первого шага, было сочтено целесообразным доработать уже существующие курсы по информационно-вычислительным и телекоммуникационным системам, введя в них дополнительно информацию по теории и технологиям открытых систем. Такое построение курса, с одной стороны, позволяет знакомить студентов с проблематикой открытых систем как интеграционной основы современных информационных технологий, и, с другой стороны, давать базовые знания в области информатизации и информационно-вычислительной техники.

Настоящее учебное пособие состоит из введения, восьми глав и заключения. В главе 1 формулируется понятие информационной системы (ИС), определяются требования к ним, рассматриваются подходы к построению моделей ИС. Глава 2 посвящена информационным измерительным системам, т.е. классу ИС, ориентированных на получение и обработку количественно определённой информации о состоянии реальных объектов. В главе 3 рассматриваются вопросы построения и применения вычислительных машин и систем как технической основы для реализации информационных технологий. Глава 4 посвящена информационно-вычислительным системам и сетям и мобильности программного обеспечения. Глава 5 охватывает круг проблем, посвящённых глобальным информационным и телекоммуникационным сетям и их структуре, стратегии межсетевого взаимодействия и согласованию протоколов. В главе 6 обсуждаются вопросы применения ТОС, как интеграционной основы создания ИС различных классов. Вводится понятие профиля, рассматривается технологический цикл построения открытой ИС. При построении настоящего курса авторы сознательно решили ограничиться рассмотренным кругом вопросов, который охватывает основные аспекты применения технологий открытых систем в современных информационно-вычислительных и телекоммуникационных системах. Расширение тематики привело бы к неоправданному увеличению объема излагаемого материала, без возможности, однако, дать подробное описание всех существующих на данный момент технологий и систем. Поэтому ряд дополнительных вопросов могут быть рекомендованы для самостоятельной углубленной проработки по имеющимся опубликованным материалам и электронным базам данных.

Авторы, тем не менее, были бы благодарны за отзывы, замечания и пожелания по настоящему курсу, которые безусловно будут учтены при последующей доработке данного учебного пособия.

Глава 1. Информационные системы

1.1. Понятие информационной системы

Основу функционирования различных систем составляют процессы преобразования информации. В широком смысле информация - есть некоторые сведения, совокупность каких-либо данных, знаний. В практической деятельности, особенно в технике, при использовании понятия «информация» удобно применять понятие "количество информации". Количество информации определяется как величина, обратно пропорциональная вероятности того события, о котором идет речь в сообщении. Чем более вероятно событие, тем меньше информации несет сообщение о его наступлении, и наоборот.

Дадим ряд определений согласно [1.1].

Система (в предметной области) – это множество взаимосвязанных элементов, каждый из которых связан прямо или косвенно с каждым другим элементом, а два любые подмножества этого множества не могут быть независимыми, не нарушая целостность, единство системы.

Элемент системы – это простейшая структурная составляющая системы, которая в рамках данной системы не структурируется.

Структура системы – это совокупность устойчивых связей, способов взаимодействия элементов системы, определяющая ее целостность и единство.

Среда (в предметной области) – это все, что находится в предметной области за границами системы.

Под информационным процессом будем понимать процесс, связанный с изменением количества информации в системе в результате целенаправленных действий при решении поставленной задачи [1.2].

К основным информационным процессам можно отнести:

- прием информации из внешней среды;
- передача информации внутри системы между ее отдельными элементами
- преобразование информации;
- создание новой информации;
- хранение и накопление информации;
- передача информации из системы во внешнюю среду.

Информационная деятельность связана с созданием информационных моделей всех объектов и явлений природы и общества, участвующих в человеческой деятельности, а также с созданием моделей самой этой деятельности.

Как известно, информация как продукт интеллектуальной деятельности человека является ресурсом и с течением времени накапливается, хотя возможны и потери информационных ресурсов.

В процессе познавательной деятельности мы, так или иначе, сталкиваемся с процессом использования накопленного знания, которое становится ценным лишь тогда, когда становится доступным широкому кругу пользователей. В настоящее время, объем информационных потоков, несущих эти знания, существенно увеличился, поэтому стала актуальной задача информатизации различных видов человеческой деятельности. Данное направление включает в себя развитие аппаратных средств и информационных технологий.

Информационные технологии (ИТ) – совокупность методов и средств реализации информационных процессов в различных областях человеческой деятельности. Иначе говоря, ИТ есть способ реализации информационной деятельности.

К современным ИТ относят:

- развитие глобальных информационных систем;
- внедрение систем автоматизированной обработки информации;
- развитие систем и средств дистанционного доступа;
- интегрирование гетерогенных систем;
- развитие систем искусственного интеллекта и т.д.

Долгое время преобразование информации и принятие решений являлось функцией человека. Сейчас, когда рост объемов информационных потоков привел к тому, что они превысили объемы усвояемости и обрабатываемости информации человеком, возникла проблема повышения эффективности процессов преобразования информации, определяемая следующими причинами:

- любая информация ценна только в процессе ее использования и при резком возрастании объемов информации принятие решений становится затрудненным, а также возрастает время обработки информационного массива;
- усложнение внутренней структуры системы, появление суперсистем, включающих целые совокупности систем, интеграция гетерогенных систем также приводит к резкому увеличению объемов информационных потоков и времени на их обработку;
- расширение сфер применения ИТ приводит к возникновению новых систем, что, в свою очередь, является дополнительным источником увеличения информационных потоков;
- повышение сложности задач, требуемой для их решения точности и оперативности, приводит к опережающему росту сложности управления по отношению к росту возможностей обработки информации [1.2] и так далее.

Определим два основных пути развития ИТ, обеспечивающих повышение эффективности процессов преобразования информации в информационных и информационно-управляющих системах:

- совершенствование технических средств автоматизации на основе применения высокопроизводительных вычислительных устройств

- и систем, что приводит к повышению скорости обработки информации вне зависимости от характера преобразуемой информации;
- совершенствование и расширенное внедрение программного обеспечения.

Для реализации указанных путей необходимо наличие наиболее общих подходов к решению стоящих задач, инвариантных к конкретной содержательной стороне задачи и техническим средствам ее реализации.

Для информационных систем эта задача обостряется в связи с развитием научного знания, существенным увеличением его объемов, когда уже в рамках узких, подотраслевых вопросов объемы процессов восприятия нового знания превышают возможности человека, не говоря уже о возможностях использования межотраслевого опыта. При этом является рациональным решение, когда сочетаются наиболее общие подходы к решению проблемы с их конкретной технической реализацией. Возможность рассматривать любую систему, абстрагируясь от ее технической реализации, возможность переноса опыта по разработке и исследованию систем, решающих один круг задач, к системам, предназначенным для решения задач в иной области, говорит об открытости, как самих систем, так и о принципах и подходах к их построению и исследованию, которые будут сформулированы ниже.

1.2. Требования к информационным системам.

Сами по себе средства вычислительной техники (СВТ) не могут осуществить преобразование информации, для этого необходимо наличие прикладного информационного и программного обеспечения, реализующего функции информационной или информационно-управляющей системы (ИУС). ИУС представляет собой совокупность элементов, находящихся в отношениях и связях между собой и образующих определенную целостность, единство, и предназначенных для осуществления целенаправленного процесса преобразования информации.

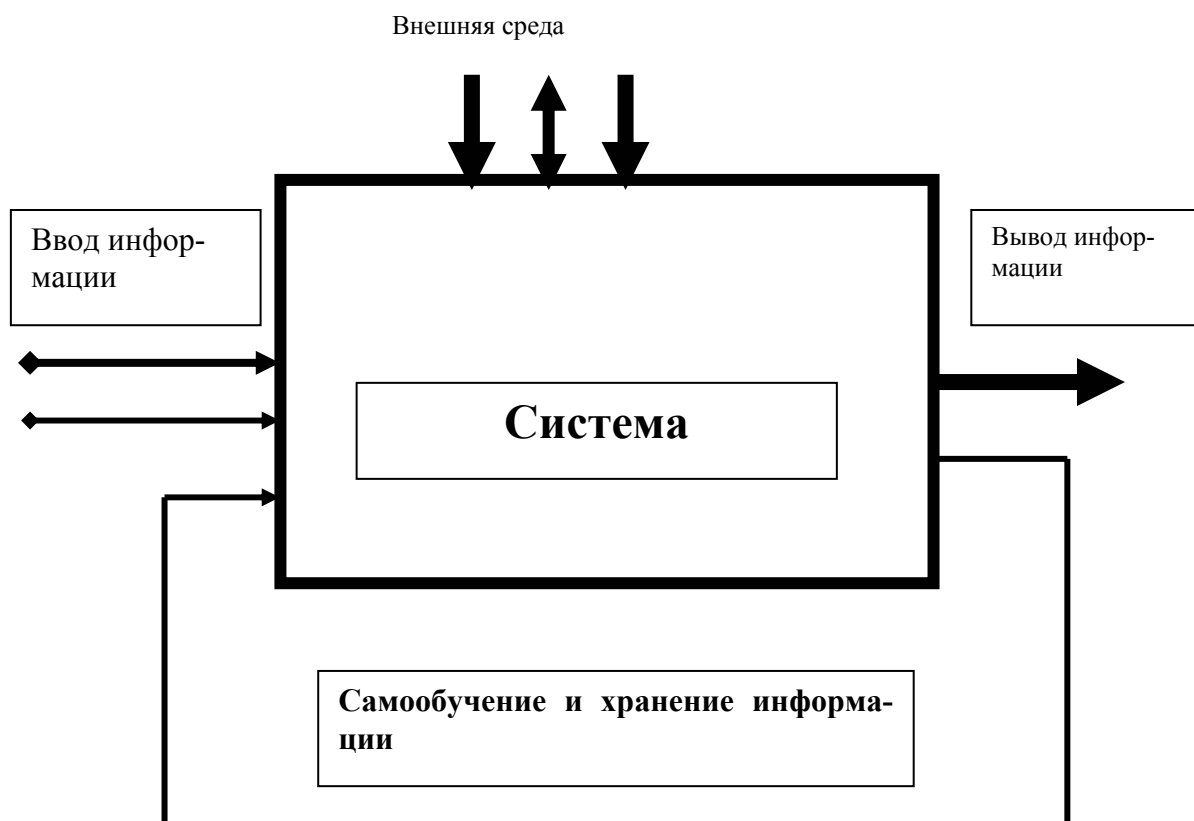


Рис 1.1. Основные функции системы

К основным функциям ИУС можно отнести:

- организация интерфейса обмена между технической и информационной системами, а также между ИУС и внешней средой;
- организация работы и распределение ресурсов собственно ИУС;
- самообучение системы, адаптация к изменяющимся условиям (рис.1.1).

ИУС должны удовлетворять следующим требованиям:

- обеспечивать достоверность соответствия описаний объектов в ИУС по отношению к их реальному состоянию;
- иметь дружественный интерфейс процесса управления,
- обладать возможностями развития и самообучения системы;
- обеспечивать полноту представления информации в системе и во взаимодействии системы с внешней средой, своевременность и обоснованность в выработке соответствующих решений, мобильность ИУС при работе в условиях гетерогенных технических средств, реализующих систему, защиту информации в системе;
- обеспечивать реализуемость заданного алгоритма;
- надежность работы в реальных условиях [1.3].

Реализуемость алгоритма характеризуется следующими свойствами:

- выходные параметры, получаемые алгоритмически, не должны противоречить параметрам, получаемым исходя из величин, задаваемых человеком;
- значения выходных параметров должны быть максимально информативными, а для автоматизированных систем, в которых решение принимается человеком, необходима максимально подробная информация о причинах, процессах и последствиях принятия решений;
- исходные параметры должны обладать свойством полноты и быть реально получаемыми, то есть относиться к классу физически измеряемых величин;
- алгоритм должен правильно (адекватно) отражать физические свойства того процесса, для обеспечения управления которым он создается;
- алгоритм должен обладать способностью к интерактивному общению с человеком.

Необходимость адаптивности ИУС к изменяющимся условиям функционирования вызвана следующими причинами:

- изменением состава применяемых СВТ, приводящим, как правило, к возникновению гетерогенной вычислительной среды;
- изменением состава общего и прикладного программного обеспечения, появлением его новых версий;
- временное возникновение неполноты состава исходных параметров;
- возникновение временных ограничений на ресурсные возможности системы.

Совместимость ИУС заключается в инвариантности к вычислительной среде ее функционирования, к существующим структурам управления, другим ИУС, системному программному обеспечению, имеющейся информационной базе.

Надежность алгоритма заключается в способности правильно выполнять свои функции в реальных условиях. К источникам, которые могут привести к ненадежной работе, можно отнести ошибки алгоритмической реализации физического процесса, ошибки трансляции, ошибки исходных информационных массивов, ошибки, вызванные работой технических средств.

Развитие СВТ, расширенное их внедрение во все сферы науки, техники, сферы обслуживания и быта привели к необходимости объединения конкретных вычислительных устройств и реализованных на их основе ИС в единые информационно-вычислительные системы (ИВС) и среды. При этом возникли следующие проблемы:

- разнородность технических средств ВТ с точки зрения организации вычислительного процесса, архитектуры, системы команд, разрядности процессора и шины данных, ресурсных возможностей, частот синхронизации и так далее, потребовала создания физических интерфейсов, обеспечивающих их совместимость;

- разнородность программных сред, реализуемых в конкретных вычислительных устройствах и системах с точки зрения многообразия операционных систем, различия в разрядности, объемах адресуемой памяти, применяемых языках программирования и так далее, привела к созданию программных интерфейсов между устройствами и системами;
- разнородность реализации одной вычислительной структуры, изготовленной различными производителями, также требовала применения специальных ограничений, либо разработки дополнительных программных и (или) технических средств для интеграции;
- разнородность интерфейсов общения в системе "человек-машина" требовала постоянного переобучения кадров.

Таким образом, необходимость предусмотреть уже на стадии разработки возможность интегрирования разрабатываемого устройства в гомогенные и, особенно, в гетерогенные информационно-вычислительные среды стала актуальной для разработчиков как аппаратных, так и программных средств.

С этой целью, при разработке ИС и ИУС необходимо соблюдать требование системности, включающее в себя:

1. Систематизацию информационной базы, то есть исключение противоречий и дублирования между отдельными ее частями, обеспечение полного представления информации, согласование времени поиска информации в соответствии со структурой.
2. Организацию и упорядочивание внешних связей ИУС и технических средств автоматизации.
3. Учет условий хранения информации в ИУС.
4. Стандартизацию форм представления информации, форм представления документов, структуры информационной базы, структуры и свойств алгоритмов [1.3].

Весь процесс разработки можно условно разделить на этапы: анализ системы и разработка ее информационной модели, разработка математической модели (алгоритма), разработка программной модели, разработка документации на ИУС. Рассмотрим подробнее каждый из перечисленных этапов.

1.3. Информационная модель и ее элементы

Информационной моделью (ИМ) будем называть параметрическое представление процесса циркуляции информации, подлежащей автоматизированной обработке в системе [1.3].

К основным элементам информационной модели относятся:

- блоки преобразования информации (БПИ);
- массивы данных (МД);
- точки диалога (ТД);
- параметрические связи (ПС) [1.3].

Блоки преобразования информации представляются "черными ящиками", имеющими указания цели их функционирования.

Массивы данных представляются связанными совокупностями параметров, одновременно перемещаемыми и хранимыми при функционировании системы.

Точки диалога являются узлами (вершинами) информационной модели, в которых осуществляется взаимодействие между человеком, техническими средствами автоматизации и системой.

Параметрические связи указывают направление переноса информации между отдельными элементами ИМ.

Структура ИМ описывается ориентированным графом, вершинами которого являются блоки преобразования информации, точки диалога и информационные массивы архивного хранения. Дугами графа являются параметрические связи.

БПИ обладают свойством замкнутости, проявляющемся в том, что при выполнении функций, возложенных на них, взаимосвязь каждого блока с остальными элементами ИМ возможна только по фиксированным линиям ПС и внутри блока не могут быть локализованы точки диалога.

Описание блока состоит из описания цели преобразования информации и описания функций, выполняемых блоком.

В описании функций БПИ должны находить отражение временные характеристики его работы, условия синхронизации обработки поступающей информации, описание набора информационных сигналов, без которых невозможно функционирование блока. Описание осуществляется на любом формальном языке: алгоритмическом, математическом и т.д.

Массив данных обладает свойством целостности, проявляющемся в том, что все параметры массива передаются по линиям параметрической связи как связная совокупность. Для выделения части параметров массива должна быть выполнена соответствующая функция, задаваемая (описываемая) блоком преобразования информации, который осуществляет формирование нового массива. Описание МД состоит из перечня идентификаторов параметров и их основных характеристик и описания области допустимых взаимосвязей между ними.

Функционально массивы делятся на три класса: массивы исходной информации, массивы результатов функционирования БПИ, массивы накопления и хранения информации. Массивы первого класса подразделяются на внешние и внутренние. Массивы второго класса подразделяются на внутренние, диалоговые и конечные. Массивы третьего класса делятся на оперативные и документальные (архивные).

Точки диалога являются местом взаимодействия человека и технических средств автоматизации с алгоритмами системы и делятся на ТД первого рода, в которых осуществляется взаимодействие системы и человека, и ТД второго рода, где осуществляется взаимодействие системы и технических средств автоматизации. Описание ТД состоит из описания целей деятельно-

сти в данной точке, перечня задаваемых человеком параметров, описания правил диалога, перечня диалоговых массивов и т.д.

Параметрические связи реализуют взаимодействие между отдельными элементами ИМ. Описание ПС состоит из списка элементов ИМ, взаимодействие между которыми осуществляет данная ПС, включая направление обмена информацией, перечня МД, передаваемых в направлении данной ПС; описания правил, по которым осуществляется передача массивов.

Рассмотрим пример построения системы с одной целью функционирования. В этом случае построения ИМ начинается с уточнения деятельности старшего в иерархии системы органа управления, который на графе ИМ представляется точкой данных (Д1). Далее определяется множество массивов данных $M(\Pi(1))$, значения параметров которого определяется данным органом управления. Они являются управляющими параметрами, ведущими к достижению поставленной цели.

После этого определяется множество МД, содержащих исходные данные, необходимые для получения результата $M(\Pi(2))$.

Контур управления, описывающий результаты первого шага построения ИМ (Рис.1.2), состоит из блока преобразования ин-

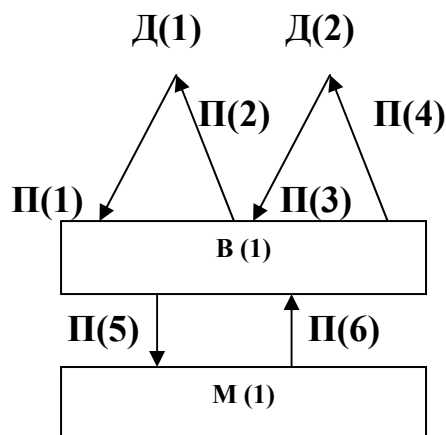


Рис. 1.2. Простейший контур управления.

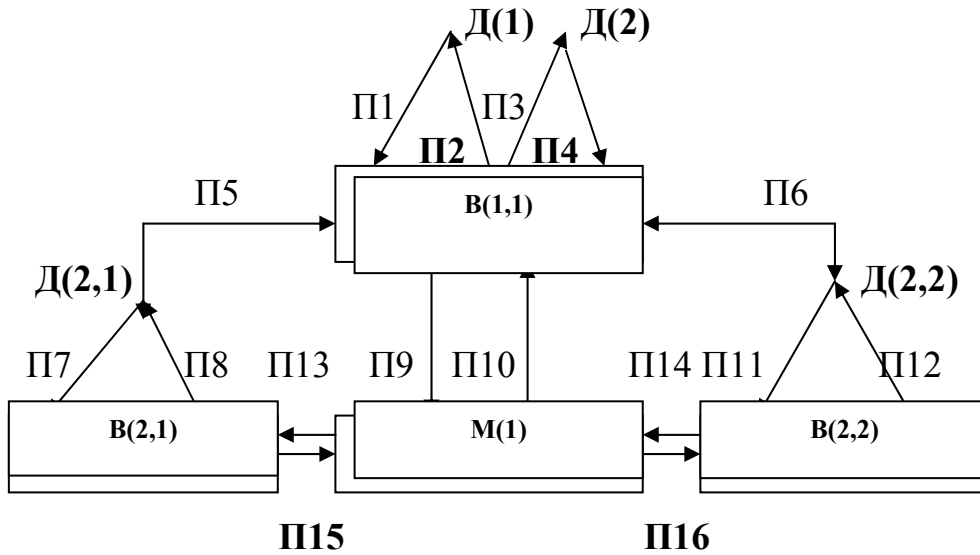
формации $V(1)$, параметрических связей $\Pi(1)$ - $\Pi(6)$ и информационных массивов $M(1)$, $M(\Pi(1))$ - $M(\Pi(6))$ и точек диалога $D(1)$ – точка взаимодействия с человеком и $D(2)$ – точка взаимодействия с техническими средствами автоматизации.

После этого, анализируются возможности автоматизации процесса управления на базе сформированного контура. При этом оценивается способность органа управления принимать решения для формирования значений управляющих параметров $M(\Pi(1))$ в необходимом темпе, а также возможность формализовать процесс преобразования информации, то есть реализовать БПИ $V1$.

В простейших системах процесс построения ИМ на этом может быть завершен. Если хотя бы одно из указанных условий не выполняется, процесс построения ИМ продолжается.

Рассмотрим некоторые типовые ситуации. Орган управления не может выполнять все функции, т.е. в одной точке диалога невозможно обеспечить принятие всех необходимых решений даже при условии автоматизации этого процесса. Тогда необходимо расширить ИМ, вводя новые точки диалога (рис.1.3.).

А)



Б)

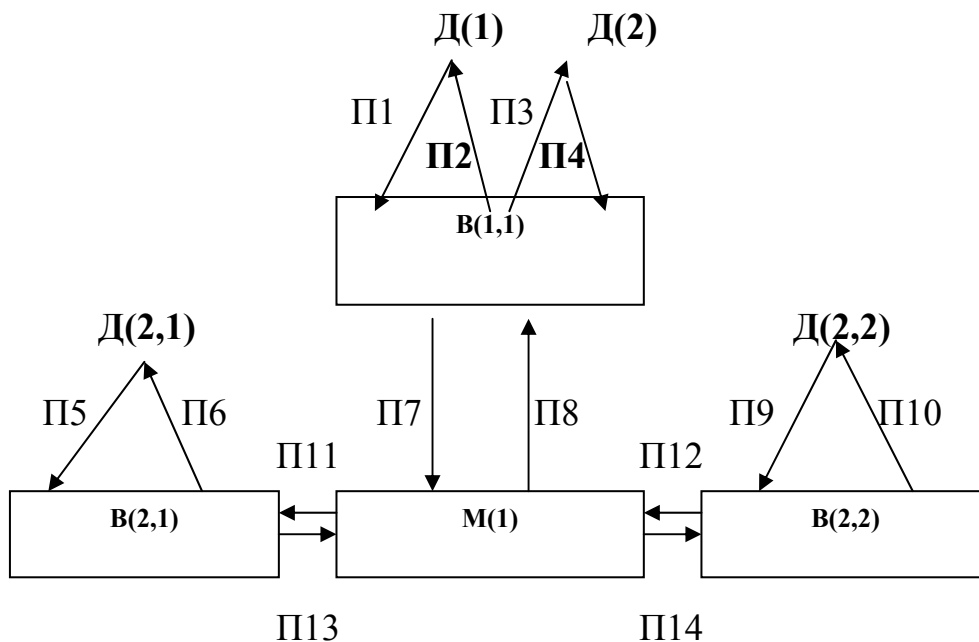


Рис.1.3. Контур управления с двумя уровнями иерархии.

а). Схема централизованного управления.

б). Схема системного управления.

Все функции БПИ (В1) не могут быть полностью возложены на средства автоматизации в связи с отсутствием возможности создания в приемлемое время адекватной математической модели. Тогда необходимо часть функций оставить за людьми, введя новые точки диалога. Для этого вводит-

ся новый (низший) уровень управления и производится членение критерия эффективности (рис.1.2).

На этом уровне располагается несколько подчиненных органов управления, функционирование которых определяется частичным критерием эффективности. Между ними распределяются функции управления. Каждый из этих органов представляется на графе ИМ точкой диалога ($D(1), D(2,1), D(3,1), \dots$). Для каждой из них определяются множества $\{M(1,2,i), M(2,2,i), \dots M(i,2,i)\}$, элементы, которых с точностью до расширения вспомогательными массивами являются подмножествами соответствующих элементов множества $\{M(1,1), M(2,1), M(g,1), M(3,1)\}$. После этого строится новый контур управления, в котором $B(i, k)$ обозначают БПИ i -го уровня управления. Если в точках диалога $D(i,2)$ дано право самостоятельно принимать решения, то вводятся параметрические связи от этой части контура к объекту управления, иначе после подготовки рекомендаций осуществляется доклад вышестоящему органу ($D1$). Для этого в ИМ вводится параметрическая связь с этого уровня управления на внешний.

Процесс заканчивается тогда, когда на всех уровнях иерархии управления (во всех точках диалога) обеспечена возможность выработки решений в темпе течения реального процесса управления.

Сформированная таким образом ИМ, которую будем называть основной, в полной мере отражает содержательную сторону процесса управления (критерии управления, цели управления и пути их достижения). Однако, она в меньшей степени отражает возможность построения математической (ИМ) и программной модели (ПМ).

При построении ММ каждый из БПИ, которые представлены "черными ящиками", должен быть заменен алгоритмом. При этом может возникнуть необходимость введения новых внутренних информационных связей, выполняющих служебные функции. Введение таких связей повлечет за собой расширение множества информационных массивов, увеличение числа промежуточных связей. Все это приводит к корректировке ИМ.

Теперь рассмотрим случай, когда в системе есть не одна, а несколько целей, достижение которых и является задачей системы. Для таких систем применяется метод единичного контура.

Единичным контуром управления называется часть общего процесса циркуляции информации, связанная с достижением только одной цели функционирования системы, имеющая только одну точку диалога. Выделение таким образом цели управления существенно снижает сложность анализа этой части общей системы.

Первым этапом при построении ИМ системы является деление всего контура управления на множество единичных контуров. Последовательно рассматривая цели функционирования формируется свой контур управления для каждой из них. Если он поддается автоматизации, то является единичным и для него строится ИМ. Если автоматизация для него практически невозможна, продолжается деление этого контура на более простые. Для каждой цели функционирования этот процесс заканчивается формированием

системы иерархически связанных единичных контуров управления, для которых может быть построены ИМ.

1.4. Разработка математической и программной моделей ИУС

Рассматривая построенную информационную модель, можно увидеть, что все ее элементы, кроме БПИ, разработаны с достаточной полнотой и формализацией. Для БПИ сформированы цели преобразования информации и определены функции, то есть содержится информация о том, что должен делать каждый блок. Описание того, как он это должен делать, на этапе разработки ИМ не рассматривалось.

Математической моделью блока преобразования информации (ММБПИ) будем называть алгоритмическое представление функций этого блока, обеспечивающее автоматизацию процесса преобразования информации, содержащейся в массивах исходных данных, а так же формирование массивов выходных данных в соответствии с параметрическими связями ИМ [1.3].

На разработку ММ каждого БПИ оказывают влияние три фактора:

- уровень развития математической теории и методов формализации процессов;
- возможности технических средств автоматизации;
- реальные ресурсы труда и времени, имеющиеся для разработки систем.

В составе ММ БПИ выделяют основные и вспомогательные функции. Состав и содержание основных функций БПИ определяется содержанием автоматизированного процесса управления.

Вспомогательные функции обеспечивают выполнение БПИ основных функций. К ним относятся:

- анализ правильности записей исходных данных, осуществление проверки на принадлежность значений исходных данных интервалу допустимых величин;
- выработка реакции на ошибку в исходных данных (обнуление, повторное обращение);
- организация служебного диалога в случаях отклонения работы БПИ от основного режима функционирования;
- анализ полноты множества записей поступивших данных и выработка реакции на возможную неопределённость этих записей;
- организация обучения в системе (самообучение);
- организация вычислительного процесса во времени (синхронизация);
- формирование комментариев.

Разработка ММБПИ начинается с определения ограничений, при которых алгоритм модели должен осуществлять преобразование информации. Эти ограничения представляются в формализованном виде, позволяющем осуществлять алгоритмическую проверку их удовлетворения.

К ним можно отнести следующие ограничения:

- на область допустимых значений параметров;
- на достоверность получаемых значений и их точность (при наличии влияния случайных воздействий);
- на источники значений входных величин;
- на обязательные взаимосвязи между значениями параметров;
- на закон изменения значения параметра.

На следующем этапе определяются допущения, которые могут быть сделаны при разработке алгоритмов ММ и характеризующие степень приближения модели к реальному процессу.

К моменту выбора математических методов для создания алгоритма для каждого из БПИ, разработчик имеет следующие сведения:

- описание целей преобразования информации блоком;
- описание функций, выполняемых блоком;
- перечень МД, содержащих исходную информацию;
- перечень МД, содержащих результаты работы блока;
- описание ограничений;
- описание допущений на разработку алгоритмов ММ.

Обладая этими сведениями, разработчик алгоритма ММ начинает завершающий этап работы. Его задачей является создание алгоритма, удовлетворяющего всем, перечисленным выше, требованиям. Для этого разработчик, используя тот или иной математический аппарат, создаёт ММ, обеспечивающую решение поставленной задачи.

При этом рекомендуется использовать следующие принципы:

1. принцип модульности, когда ММ делится на составные части, между которыми устанавливается информационная связь, а затем алгоритмы этих частей разрабатываются независимо друг от друга;
2. принцип унификации, стандартизации и наращиваемости;
3. принцип оптимизации, то есть выработка рационального решения реализации ММ БПИ;
4. принцип управляемости, то есть управление ходом вычислительного процесса при модульном построении алгоритма;
5. принцип адекватности, то есть достаточно точное отражение тех процессов, для которых она создана;
6. принцип согласованности, то есть согласование множества ММ БПИ по исходной и результативной информации с параметрами МД ИМ, и отсутствие ошибок в МД.

Реализация ММ БПИ осуществляется с применением соответствующих математических методов, с которыми читатель должен был познакомиться в процессе изучения соответствующих разделов высшей математики, поэтому не ставя перед собой задачу описания этих методов отсылаем читателя к соответствующей литературе.

На этапе создания программной модели ИУС должна приобрести все свойства, необходимые для реализации заданных функций с учётом предъявляемых требований.

Программной моделью будем называть отображение информационной и математической моделей на языке технических средств автоматизации.

Основными элементами программной модели являются основные и вспомогательные программные модули, основные и вспомогательные информационные модули.

Общим для основных элементов ПМ является принадлежность к одной и той же категории - категории модулей. Модульное построение программной модели обеспечивает возможность организации параллельного выполнения разработки.

К основным программным модулям относят те модули, которые реализуют функции, определяемые ММ и ИМ. Соотношение между модулями ММ и основными программными модулями может не быть взаимно однозначными: один модуль ММ может отображаться несколькими основными программными модулями и наоборот.

Вспомогательные программные модули реализуют функции сопряжения всех частей (модулей) ПМ с техническими средствами автоматизации, с общесистемным МО и уже внедренной частью ИУС.

Основные информационные модули ПМ предназначены для хранения параметров и массивов ММ и ИМ.

Вспомогательные информационные модули содержат информацию, необходимую для выполнения вспомогательных программных модулей, а также для согласования программной модели с общесистемным ПО, техническими средствами автоматизации и с уже функционирующей частью ИУС.

Программная модель строится из элементов четырех перечисленных типов. Она размещается в памяти ТС ВТ. Её функционирование инициируется входящим потоком, содержащим задания.

К основным принципам разработки ПМ можно отнести:

- автоматизация процесса разработки;
- использование стандартизации и унификации;
- выявление стандартных моделей в составе МП;
- адаптивность к уже существующим МО;
- сокращение затрат.

Собственно реализация программной модели осуществляется на конкретном языке программирования для конкретной программной среды. С путями такой реализации можно познакомиться, обращаясь к специальной литературе.

Выводы по главе 1.

1. Показана необходимость применения информационных технологий на основе развития систем и средств обработки информации.
2. Определены требования, предъявляемые к информационным системам.
3. Описан способ построения информационных моделей ИС и ИУС.

4. Показано, что разработка ИС и ИУС до стадии программной модели является инвариантной к программным и техническим средствам, её реализующим, что позволяет сокращать время разработки и затраты при переносе системы на новые ТСА или в новую программную среду, обеспечивая тем самым мобильность и переносимость ИС и ИУС.

Глава 2. Информационно-измерительные системы.

2.1. Общие сведения.

Важнейшей составной частью информационной индустрии являются информационные технологии. Целью любой информационной технологии является получение и представление информации в такой форме, чтобы наблюдатель мог ее воспринять, проанализировать и принять решение.

По мере развития ИТ наблюдался переход от технологий, направленных только на обработку цифровых данных, к технологиям, включающим возможность автоматического получения и ввода в ЭВМ количественно определенной информации от объекта материального мира с дальнейшей ее обработкой в соответствии с процедурами, указанными оператором или прикладной программой.

Важными элементами такой технологии являются автоматические измерения, в результате выполнения которых с помощью специальных технических средств находят опытным путем численные значения физических величин, характеризующих изучаемый объект или явление.

Процессы автоматических измерений имеют общие черты, которые в первую очередь включают восприятие техническими средствами исследуемых (измеряемых, контролируемых) величин, преобразование этих величин в промежуточные величины, сравнение их опытным путем с известными величинами, формирование и представление результатов в виде именованных чисел, их отношений и суждений, основанных на количественных соотношениях.

Функционально объединенную совокупность средств измерений, вычислительных и вспомогательных устройств, а также каналов (линий) связи, предназначенных для выработки сигналов измерительной информации в форме, удобной для непосредственного восприятия оператором и/или для автоматической обработки, передачи и/или использования в автоматических системах управления принято называть информационно-измерительной системой (ИИС). Схематически типовая ИИС представлена на рис. 2.1.

В зависимости от принципов построения, используется классификация ИИС (табл.2.1). Наиболее интересным классом ИИС являются системы, содержащие в своем составе программно-управляемые вычислительные средства. Такие системы принято называть измерительно-вычислительными системами (ИВС), их ядро составляет ЭВМ или микропроцессор, а также имеются устройства сопряжения с каналами связи, сопряжения с каналами ввода – вывода аналоговых сигналов, аналоговые преобразователи, измерительные коммутаторы, АЦП, ЦАП, устройства, формирующие воздействия на объект и некоторые другие устройства для выполнения функций ИВС. Комплексы агрегированных средств для создания ИВС называются измерительными вычислительными комплексами или просто ИВК.

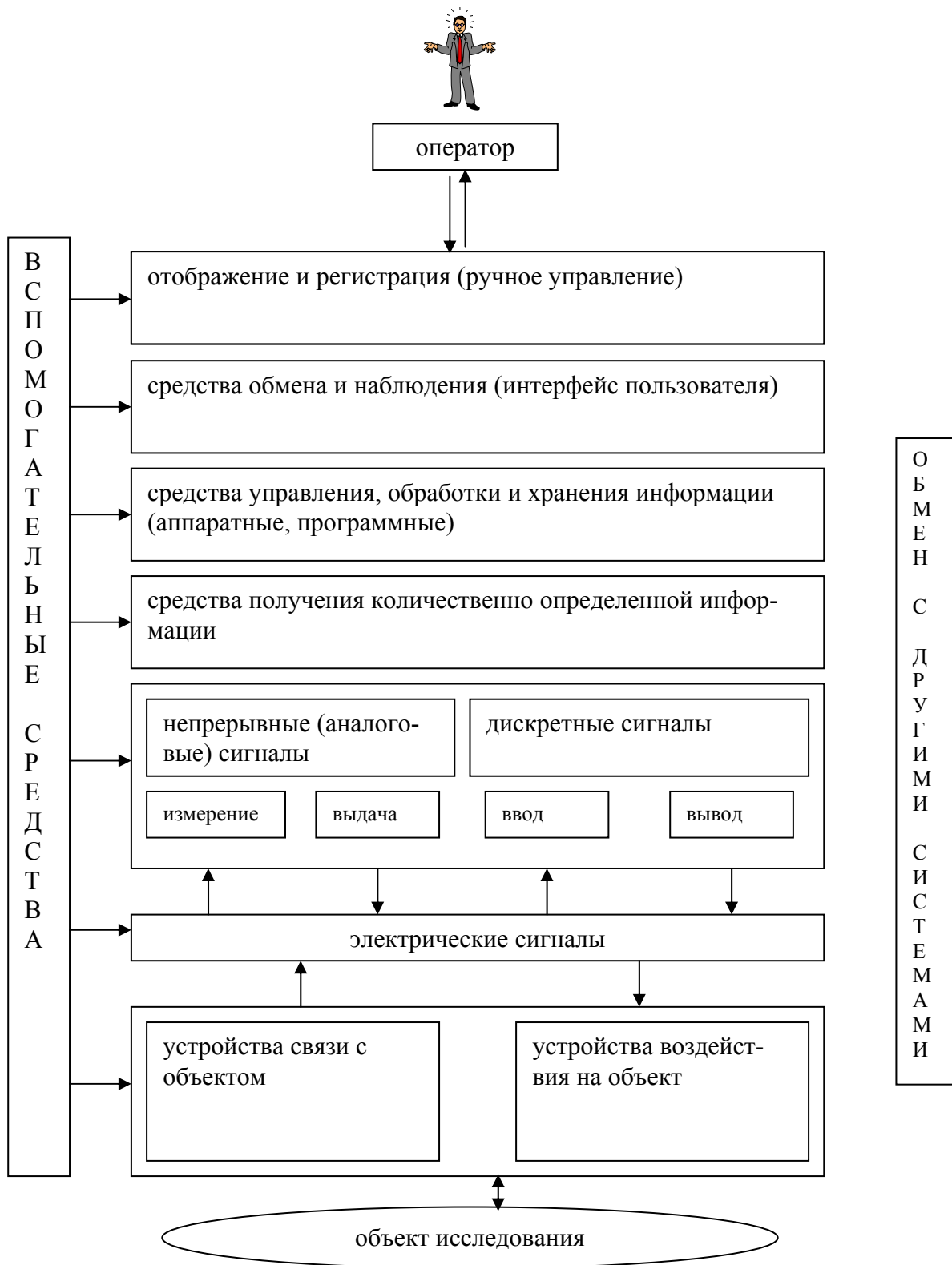


Рис. 2.1. Схематическое представление ИИС.

Таблица 2.1. Классификация ИИС по принципам построения.

Признак классификации	Классы	
Наличие специального канала связи	Отсутствует	Имеется
Порядок выполнения операций получение информации	Последовательный	Параллельный
Агрегатирование состава системы	Агрегатированный	Неагрегатированный
Использование стандартных интерфейсов	Не используется	Используется
Наличие программно-управляемых вычислительных устройств	Отсутствует	Имеется
Использование стандартного базового и прикладного математического обеспечения	Не используется	Используется
Наличие контуров информационной обратной связи	Отсутствует (разомкнутые системы)	Имеются (одно- и многоконтурные компенсационные системы)
Изменение скоростей получения и выдачи информации	Без изменения (системы реального времени)	С изменением
Вид используемых информационных сигналов	Аналоговые	Цифровые

2.2. Модель взаимосвязи открытых систем.

Создание ИИС требует объединения в единую структуру различных элементов, блоков и устройств. Эти составляющие разрабатываются и выпускаются различными производителями. Ясно, что производитель при создании как аппаратных, так и программных средств, обязан соблюдать определенные требования и нормы, соответствие которым гарантирует возможность такого объединения. В первую очередь такие требования и нормы должны быть установлены в отношении свойств и признаков данных, кото-

рыми обмениваются между собой в процессе работы системы ее различные составляющие.

Чтобы организовать процесс взаимодействий, надо данные предварительно подготовить, а также выполнить некоторые дополнительные операции. Ясно, например, что подготовка данных и операции их ввода в компьютер, передачи конечному пользователю или от компьютера к компьютеру качественно различаются.

Лучшим средством решения названных выше проблем является использование существующих национальных и международных стандартов. С точки зрения важности применения в информационных измерительных системах особое место занимает модель взаимодействия открытых систем.

Модель сетевого и межсетевого взаимодействия МОС (ISO) определяет иерархию из 7 уровней взаимодействия компонентов сети.

- 7 уровень – прикладной. Это высший уровень в иерархии. Здесь обеспечивается поддержка прикладных процессов конечных пользователей. Он содержит все необходимые элементы сервиса для прикладных программ пользователя. На этом уровне пользователь имеет свои прикладные программы, где может делать всё, что ему надо, но должен руководствоваться некоторыми установленными правилами при обменах с другим пользователем сети, т.е. выполнять соответствующие протоколы.
- 6 уровень – представительный – обеспечивает преобразование данных пользователя к форматам, принятым в данной системе; преобразует символьные строки и коды и организует файлы с целью обеспечения независимости прикладных программ от форм передачи и получения.
- 5 уровень – сеансовый – обеспечивает установление и поддержку сеансов связи между абонентами при обмене данными, организует двунаправленный обмен данными с размещением во времени, начало и окончание заданий, восстановление связи после ошибок, связанных с отказом канала и отказом сети взаимодействия, восстанавливается или повторно устанавливается соединение.
- 4 уровень – транспортный – обеспечивает управление соединением между различными абонентами, т.е. адресацию конечных абонентов, а также разборку и сборку сообщений, сохранность блоков данных, доставку данных от узла к конкретному адресату, приписанному к узлу и наоборот, выбирает маршрут пересылки данных в сеть. Таким образом, транспортный уровень предоставляет услуги сеансовому уровню. Граница между этими уровнями – это граница между владельцем сети и пользователем.

Принято три верхних уровня определять одним общим названием – процесс или прикладной процесс. Так вот, четвертый уровень обеспечивает взаимодействие между прикладными процессами, устанавливая между ними логические каналы и обеспечивая передачу по этим каналам информационных пакетов, которыми обмениваются процессы. Отметим, что столь популярный сегодня Internet - это транспортный уровень.

Пакет- группа байтов, передаваемых абонентами сети друг другу.

Логические каналы, устанавливаемые транспортным уровнем, называются транспортными каналами.

С другой стороны транспортный уровень не только предоставляет услуги сеансовому уровню, но и обеспечивает взаимодействие по вертикали между 7-5 и 1-3 уровнями. Заранее отметим, что в 1-3 уровнях информация (данные) циркулируют в форме, недоступной для непосредственного восприятия наблюдателем (пользователем).

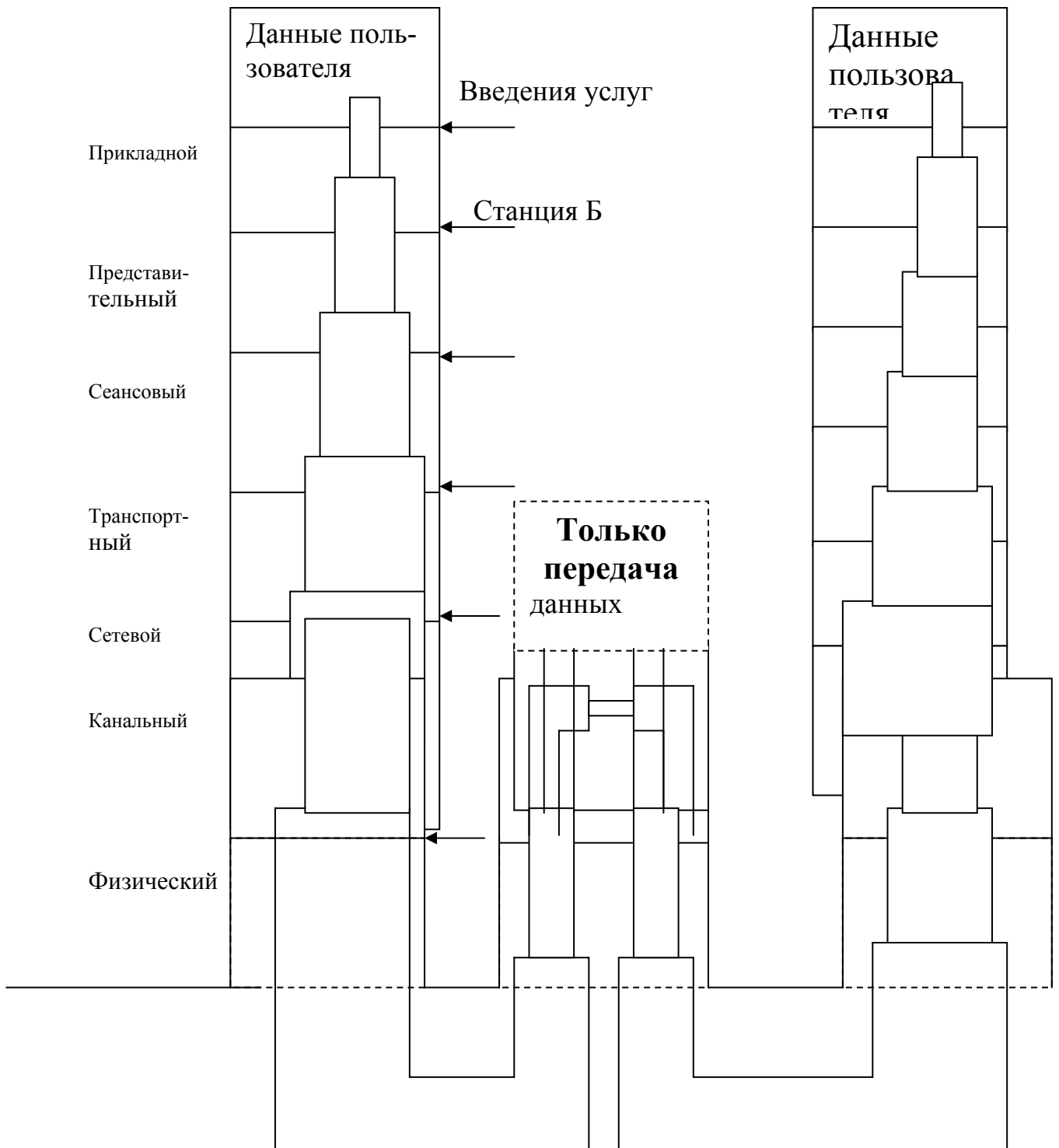
- 3 ступень (уровень) – сетевой, обеспечивает интерфейс окончного оборудования данных с сетью коммутации пакетов, маршрутизацию пакетов в коммуникационной сети, межсетевое взаимодействие. Иначе говоря третий – сетевой уровень обеспечивает функции ретрансляции, в соответствии с которыми данные направляются по маршруту в нужном направлении через устройства пакетной коммутации, т.е. к нужным узлам в соответствии с маршрутными таблицами.
- 2 ступень (уровень) – канальный или уровень звена данных, обеспечивает процесс передачи данных по информационному каналу. Информационный канал это канал логический, который устанавливается между устройствами соединенными физическим каналом. Канальный уровень обеспечивает управление потоком данных в виде кадров, обнаруживает ошибки передачи, реализует алгоритмы восстановления информации в случае обнаружения сбоев или потерь данных. Сейчас принято второй уровень разбивать на два уровня: LLC (Logical Link Control), собственно и обеспечивающий управление логическим звеном данных, т.е. канальный уровень, и MAC (Media Access Control), обеспечивающий управление доступом к среде. MAC таким образом, поддерживает метод, обеспечивающий выполнение совокупности правил, по которым узлы сети получают доступ к ресурсу (память, программа, процесс, и т.д.).
- 1 ступень (уровень) – физический, обеспечивает механические, электрические, функциональные и процедурные средства для осуществления физических соединений, их поддержания и разъединения. В частности, среда распространения сигналов - это физический уровень.

При рассмотрении этих 7 уровней надо обратить внимание на то, что наиболее актуален такой подход при обсуждении работы компьютерных сетей – это, во-первых. А во вторых, совершенно очевидно, что главное, это возможность в рамках такого рассмотрения установить единые общие правила и нормы, по которым надо делать различные устройства, разрабатывать программы, чтобы различные элементы ИТ сделанные разными производителями, работали совместно, стыковались.

Для решения этой задачи ISO разработала эталонную модель ВОС (взаимодействия открытых систем). Международный стандарт МС-7498 «Эталонная модель взаимосвязи открытых систем» был опубликован в 1983 году. Соответствующий ГОСТ 2806-93 был принят впоследствии в нашей стране.

Станция А

Станция Б



Фаза передачи данных.

Рисунок 2.2. Иллюстрация к ВОС модели.

По мере ведения услуг поток данных расширяется.

Чтобы отличить услуги, каждый уровень при обработке добавляет к данным свою протокольную управляющую информацию в виде заголовка и так называемого концевика. Причем заголовки возникают на всех этапах, концевик –

контрольная последовательность бит, используемая для проверки правильности приема сообщения, прибавляет только второй уровень. Физический уровень заголовка не добавляет.



Рисунок 2.3. Иллюстрация обработки сообщений уровнями модели ВОС.

Ясно, что помимо общего стандарта необходимо иметь стандарты на каждый уровень, сегодня на один уровень есть 20 стандартов: 2-41; 3-10; 4-8; 5-5; 6-7; 7-32.

Несмотря на это эталонная модель ВОС облегчает дальнейшие разработки, и мы к ней еще не раз вернемся.

2.3. Структуры и стандартные интерфейсы ИИС.

Мы видим, что ИИС может быть представлена в виде совокупности связанных между собой функциональных блоков (ФБ). В дальнейшем под ФБ будем понимать техническое устройство, которое выполняет информационные и (или) управляющие функции и нуждается в организации совместной, согласованной работы с другими ФБ системы. Будем предполагать, что ФБ это функционально законченное устройство и, соответственно, для организации взаимодействия ФБ системы между собой не требуется знания их внутренних структур и особенностей функционирования.

Для обеспечения работоспособности необходимо организовывать взаимодействие ФБ, составляющих систему, между собой. Для решения этой задачи применяют стандартные интерфейсы, под которыми мы будем понимать совокупность унифицированных правил (протоколов), аппаратных, программных и конструктивных средств, обеспечивающих взаимодействие компонентов системы в процессе ее функционирования. При этом под протоколом (иногда говорят логическим протоколом) будем понимать совокупность правил передачи кодированной информации (данных) между ФБ системы. Применительно к ИИС и ИВК понятие интерфейса шире понятия про-

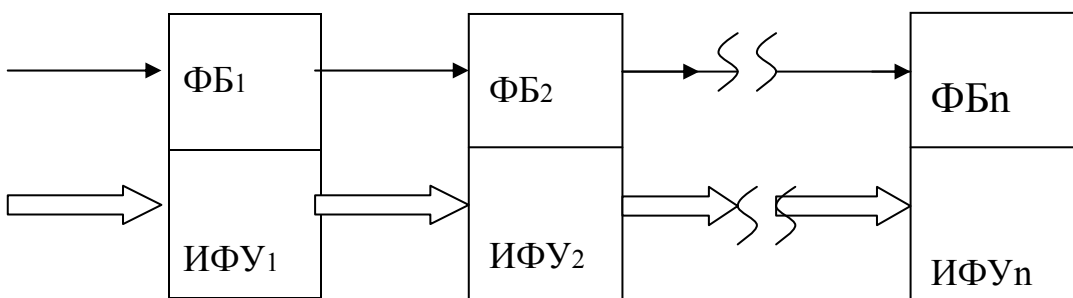
тока. Поэтому используя два понятия, а именно физический интерфейс, определяющий совокупность механических, электрических и конструктивных средств, а также физических средств передачи данных и логического интерфейса, охватывающий всю совокупность логических протоколов. Из сказанного ясно, что физический интерфейс служит материальной основой для реализации логического интерфейса.

Использование стандартных ИФ позволяет обеспечить информационную совместимость различных ФБ между собой, т.е. достичь их согласованного взаимодействия в составе системы в соответствии с заранее определенными логическими условиями и правилами.

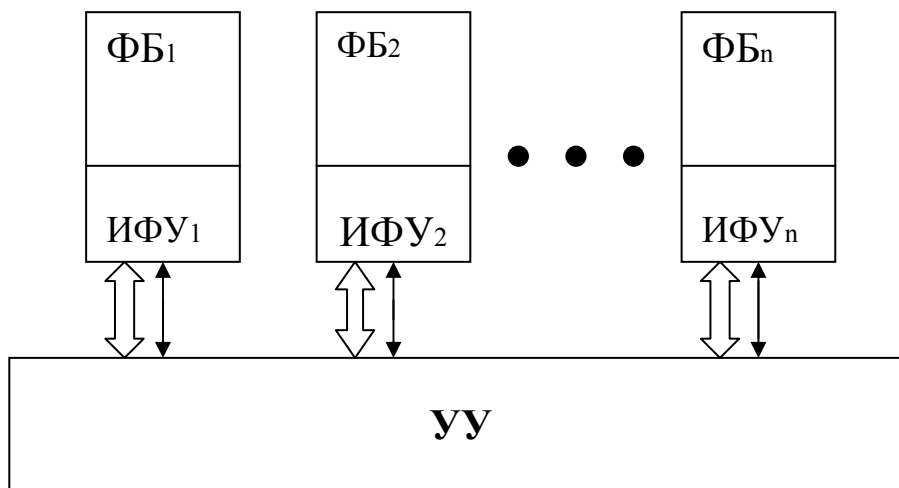
Если взять набор ФБ, снабженных интерфейсами, то их объединение для совместной работы возможно либо в одноступенчатой, либо в многоступенчатой структуре.

Возможные варианты объединения ФБ в ИИС с одноступенчатой структурой представлены на рис. 2.4. Работа ФБ, объединенных в цепочечную структуру происходит последовательно, по мере окончания преобразования в предыдущем ФБ. Объединенные в радиальную структуру ФБ могут взаимодействовать только через устройство управления (УУ). Магистральная структура наиболее удобна.

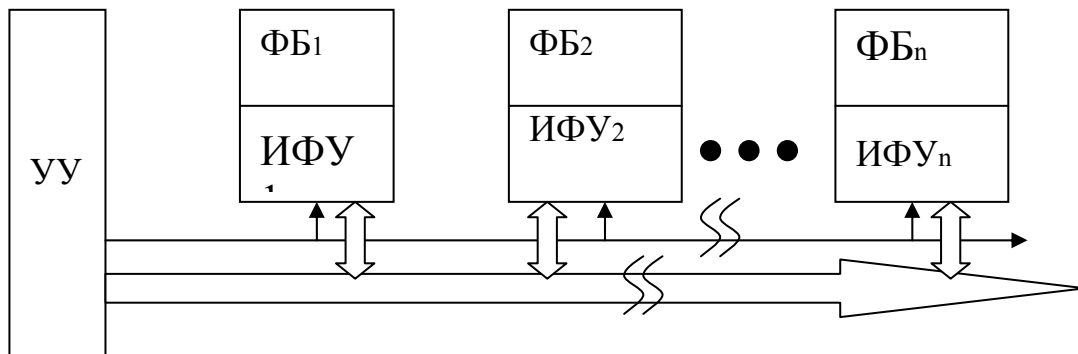
а) Цепочечная



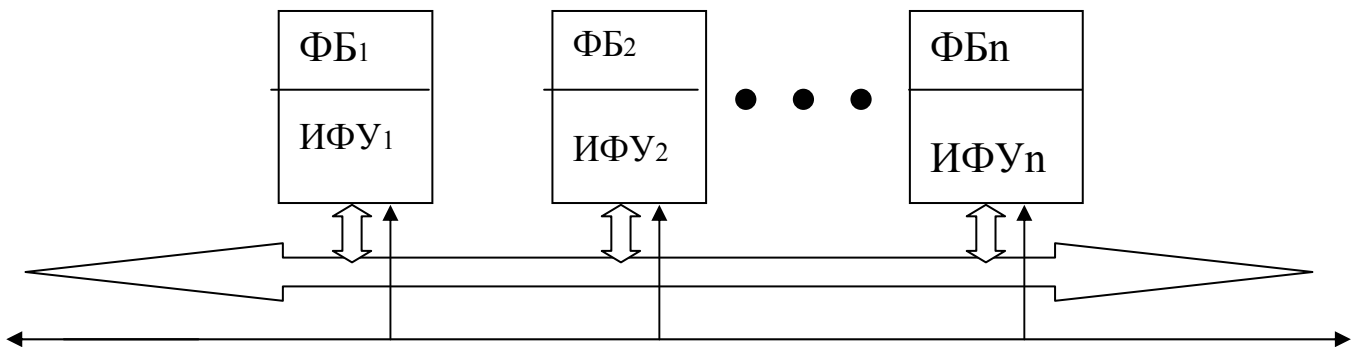
б) Радиальная



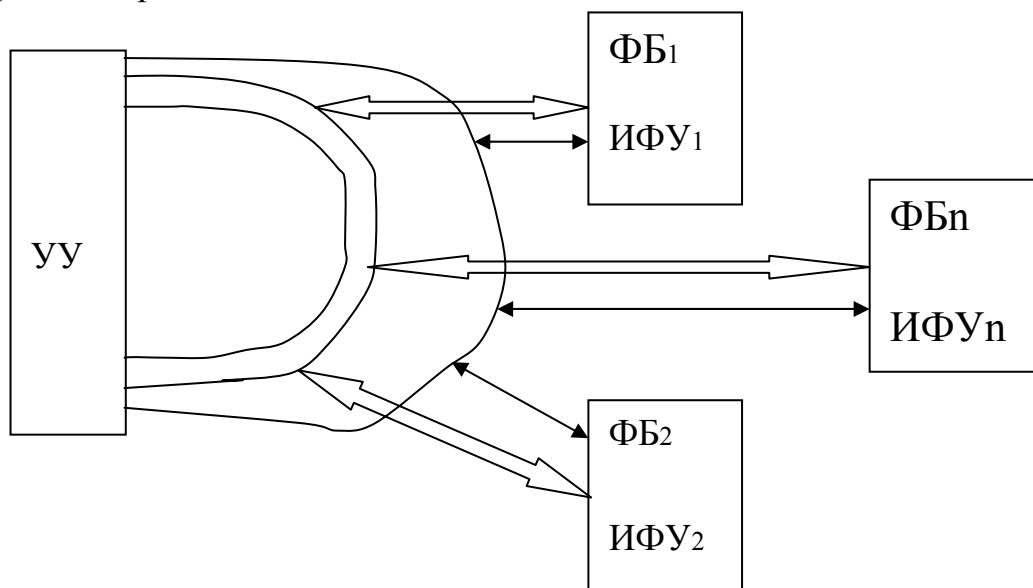
в) Магистральная с централизованным управлением



г) Магистральная с децентрализованным управлением



д) Магистральная петлевая



е) Радиально-магистральная.

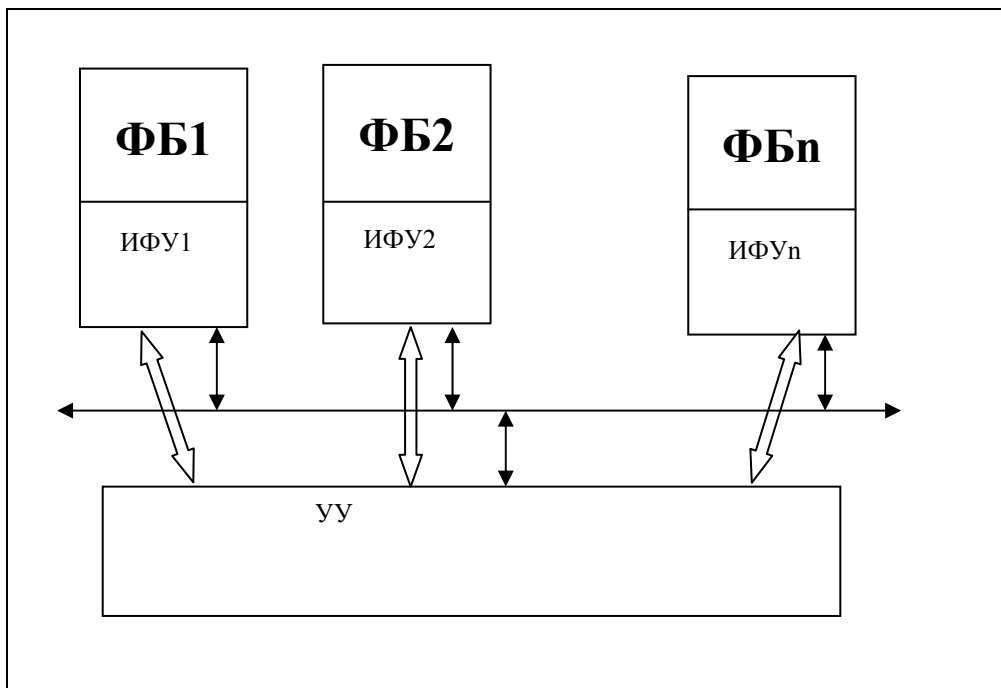


Рис. 2.4. ИИС с одноступенчатой структурой.

При большом количестве ФБ целесообразно организовать совместную работу нескольких объединенных между собой одноступенчатых подсистем. Вариант объединения дан на рис.2.5.

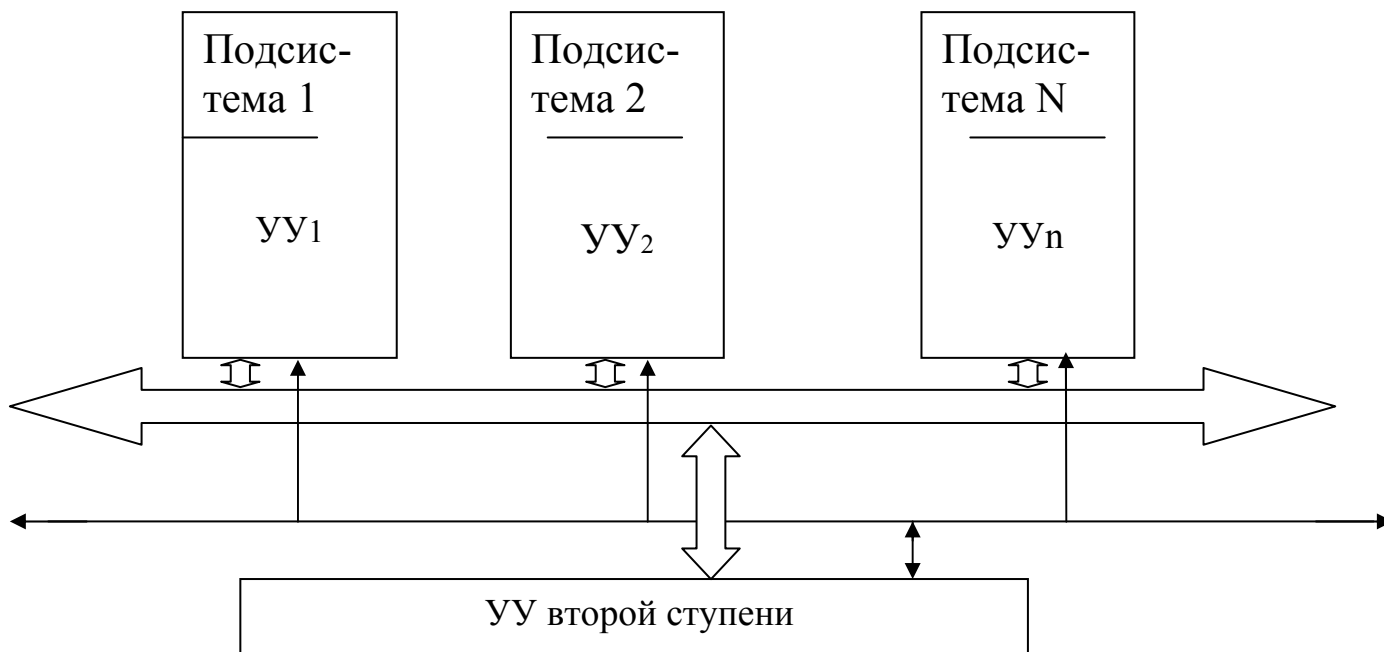


Рис 2.5. Двухступенчатая структура ИИС.

Подсистемы для объединения между собой можно выбрать из числа показанных на рис.2.4. Отметим, что ЭВМ второй ступени в двухступенчатой

структуре выполняет не только функции управления, но и обработки и выдачи информации.

Возможность создания тех или иных структур ИИС и особенностей их функционирования в существенной степени зависят от свойств ИФ.

Рассмотрим (табл. 2.2.) основные признаки стандартных ИФ, от которых в первую очередь зависит информационная совместимость ФБ в системе.

Таблица 2.2. Основные признаки программируемых ИФ.

Признак		Граничные состояния	
		1	2
1	Характер управления	централизованное	децентрализованное
2	Устройство системы шин для обмена информационными и служебными сигналами	объединенная	раздельная
3	Организация системы шин	индивидуальная (радикальная)	коллективная (магистральная)
4	Порядок выполнения операций обмена информацией	последовательный	параллельный
5	Виды информационных ФБ	источники	приемники
6	Способ синхронизации при обмене информацией	синхронный	асинхронный
7	Тип используемой ЭВМ	заданный	любой
8	Унификация конструкции	есть	нет
9	Унификация питания	есть	нет

С момента появления первых стандартов на ИФ, применяемые при создании ИИС прошло около 40 лет. К настоящему времени разработано множество различных стандартов в этой области. Важнейшей задачей при создании ИИС является выбор такого стандартного ИФ, который позволит собрать наиболее эффективную систему при минимальной цене. Ошибки в выборе ИФ могут привести к значительным потерям времени и средств. Чтобы облегчить выбор ИФ рассмотрим как должен быть организован обмен данными в системе.

2.4. Передача данных в ИИС.

При создании ИИС устанавливаются правила, регламентирующие порядок обмена данными в системе, т.е. протокол. Для этого определяются: единица обмена информацией, формат сообщений, система кодирования, способ синхронизации, вид информационных, управляющих и служебных символов, способ обмена сообщениями, набор интерфейсных функций, логические, временные и электрические параметры сигналов, способ формирова-

ния и распознавания запроса на обслуживание, правила адресации, способ обеспечения требуемой достоверности при обмене информацией.

В процессе передачи данных ФБ между собой обмениваются *сообщениями*, состоящими из различных видов *слов*. В первую очередь, это информационные слова, они содержат информационную часть, размер которой применительно к ИИС зависит от точности измерений, и сопроводительную часть, включающую сведения о диапазоне измерений, состоянии ФБ, виде информации (ток, давление, температура, и т.п.). В информационное слово может быть введен и адрес ФБ. Кроме информационных слов должны формироваться командные слова, сообщение может содержать также адресное и временное слова.

Адресное, временное, командное *слова* и одно или группа информационных слов *образуют* информационную *фразу*. *Фразы* в свою очередь объединяются в *кадры*, содержащие *заголовок* (маркер *кадра*), код объекта, код программы, код порядкового номера кадра и информационную часть, состоящую из целого числа *фраз*.

При формировании логического ИФ надо договориться о длине слов, о месте младших и старших разрядов в коде, длине фраз, размерах кадров, признаках кадра и ряде других условий.

Процедура собственно обмена данными также может быть устроена по-разному. Рассмотрим простой пример на.

Пусть ИИС (рис. 2.6.) с магистральной структурой и централизованным управлением состоит из ФБ источника (ФБИ), ФБ приемника (ФБП) и ФБ управления (ФБУ).

Предположим, что обмен данными по инициативе ФБИ, который начинается с того, что запрашивает у ФБУ разрешение на выдачу данных в магистраль. При этом ФБИ передает для ФБУ по магистрали свой адрес $Адр_1 (0)$ и запрос на передачу $Запр_1 (0)$. После получения этих сообщений ФБУ анализирует ситуацию и, если нет запроса, передает по магистрали для ФБИ свой адрес $Адр_0 (1)$ и разрешение на передачу $Разр_0 (R)$, а для ФБП свой адрес $Адр_0 (2)$ и разрешение на прием $Разр_0 (W)$. После получения разрешения на прием данных ФБП анализирует ситуацию и, в случае своей готовности к приему устанавливает в магистрали соответствующее сообщение $ГП_2 (W)$ которое адресуется ФБИ и ФБУ. После получения от ФБП сообщения $ГП_2 (W)$ ФБИ в свою очередь формирует информационное слово, содержащее адрес $Адр_1 (2)$, диапазон и режим измерения $Д (D)$, время $T(t)$ и, соответственно данные (результат измерений) $I (I)$, и через магистраль направляет его ФБП. После передачи этого слова ФБИ устанавливает в магистрали сообщение конец связи $КС! (R)$ для ФБУ и ФБП о том, что он окончил передачу. В заключении ФБП направляет в магистраль для ФБУ и ФБК сообщения о том, что данные приняты $ДП_2 (W)$. На этом цикл обмена заканчивается и устройства переходят в режим ожидания.

Контрольные, синхронизирующие и другие вспомогательные сигналы мы здесь не рассматривали. Если имеется несколько ФБИ и ФБП, то рассмотрение можно вести аналогично с добавлением процедуры установления

приоритета одним ФБ по сравнению с другими. В стандартах на интерфейс устанавливаются конкретные способы реализации логического протокола, ниже мы рассмотрим некоторые из них.

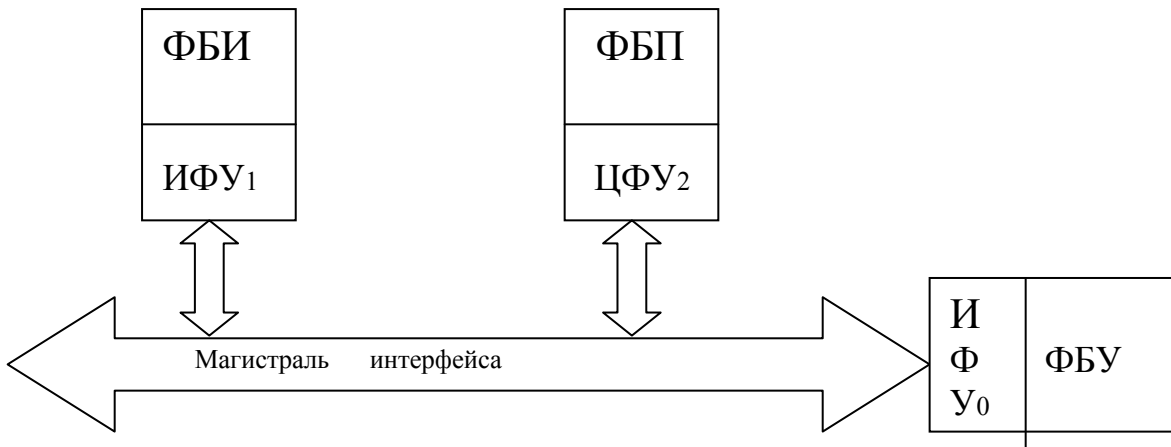


Рис. 2.6. Процедура обмена данными в простой ИИС.

2.5. Магистрально – модульные системы

Как уже указывалось раньше магистрально – модульные системы весьма удобны в эксплуатации, широкий выбор блоков позволяет создавать системы разного назначения, объединять их. Развитый логический протокол также расширяет возможности пользователя. Унификация блоков и каркасов позволяет удешевить производство и подключить к нему различные фирмы помимо разработчика. Короче говоря, блочно – модульный принцип основа всех устройств, применяемых при реализации ИТ. Соответственно магистрально – модульные системы составляют основной парк ИС, включая ИИС. Следствием этого явилось развитие этой группы интерфейсов, начиная с 60-х годов и по настоящее время.

Первой в мире магистрально – модульной системой была система КАМАК. Этот стандарт мы и рассмотрим в качестве примера, т.к. все последующие можно считать его развитием.

Интерфейс КАМАК имеет централизованное управление, причем уровней централизации может быть 2 и более. Система шин для информационных и служебных сигналов раздельная; организация системы шин радиально – магистральная, порядок выполнения операций обмена информацией – параллельный, тип ФБ – активный ФБПИ, метод обмена информацией – синхронный, допускается работа с любой ЭВМ. Конструкция и питание – унифицированные.

Первая ступень централизации управления и обработки информации достигается в крейте, а вторая в ветви, которая может объединять до семи крейтов. Допускается совместная работа нескольких ветвей.

Основой структуры ИС, использующих КАМАК, является крейт. Конструктивно крейт (crate) это ящик, в который можно вставлять платы-модули.

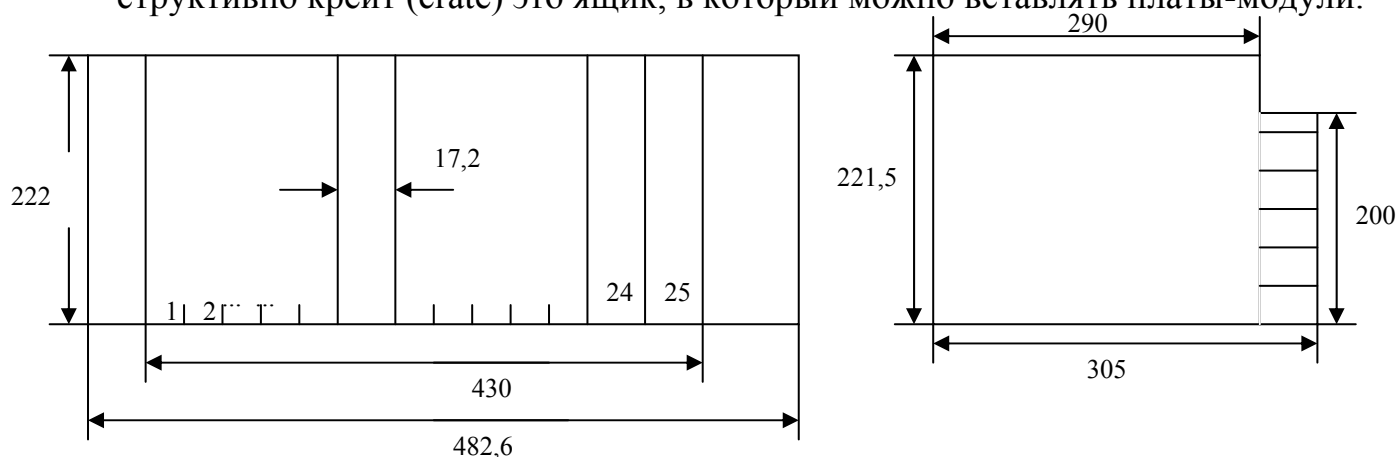


Рисунок 2.6. Основные размеры крейта и платы КАМАК.

Для модулей в крейте (рис. 2.6) имеется 25 гнезд с верхней и нижней направляющими для введения платы модулей станции. Ширина модуля 17.2 мм. Функциональные блоки могут занимать 1,2,3,4,6 или 8 модульных мест. На обеих поверхностях заднего конца платы должно быть по 43 ламели печатного разъема. Шаг между осями ламелей 2.54 мм. Платы модулей входят

в розетки двухрядных 86-контактных разъемов, укрепленных на задней плате крейта. Минимальная глубина крейта с задней платой 360 мм. Крейт может быть снабжен источниками питания, выходы которых соединяются с линиями питания модулей, расположенными на задней плате. Всего линий питания 14 и в них 6 – основных и 8 – дополнительных, включая 1 линию сигнальной земли и 2 свободных резервных линии. Место 24 и 25 станций в крейте обязательно занимает ФБ контроллер крейта (ФБК). На остальных 23 местах размещаются ФБПИ, расположение которых в крейте может быть любым. Входные аналоговые сигналы вводятся через разъемы на передней панели модулей. Допустимый ток питания отдельного модуля ограничен 2А, а рассеиваемая мощность – 8 Вт. Максимальная мощность рассеивания в крейте 200 Вт. Охлаждение воздушное.

Обмен информацией между ФБК и ФБПИ происходит по магистрали крейта. Магистраль крейта для этого имеет 7 шин: адреса, операций, состояний, синхронизации, данных, управления и нестандартных соединений.

Шина адреса – она состоит из 24 индивидуальных линий номера состояния N и 4 сквозных линий субадреса A, обозначаемых A1, A2, A4, A8. Линия номера станции соединяет контакт 9 в гнезде ее разъема и соответствующий контакт N на разъеме ФБК. Так в системе КАМАК осуществляется географическая адресация модуля в целом по радиальной линии. Адресация субблоков внутри ФБПИ логическая по линиям A, видно, что внутренних субадресов в модуле может быть до $2^4 = 16$ субадресов.

Шина операций – она состоит из 5 сквозных линий кодов операций F. Сигналы кодов операций образуются битами F1, F2, F4, F8 и F16. Они определяют до $2^5 = 32$ функций. Интерфейс КАМАК предусматривает 18 типовых функций, например F(0) – чтение регистра, F(24) – запирает и т.п. Набор интерфейсных функций, реализуемых модулей, закладывается при его разработке и помимо типовых может включать оригинальные. Команды, передаваемые по шинам адреса и операций имеют общее наименование – команды NAF. Все сигналы команд NAF формируются и выдаются ФБК.

Шина состояний – она состоит из 24 индивидуальных линий состояния L, по которым запросы на обслуживание передаются от ФБПИ к ФБК. Кроме того шина состояний включает 3 сквозные линии, а именно: Q – по которой от ФБПИ к ФБК сообщается, что соединение состоялось, X – по которой от ФБПИ к ФБК сообщается, что команда принята и B – по которой от ФБК к ФБПИ сообщается, что магистраль занята.

Шина синхронизации – она состоит из 2 сквозных линий. По линии S1 от ФБК к ФБПИ передается синхроимпульс (строб 1), сопровождающий неизменные сигналы на шине данных. По линии S2 от ФБК к ФБПИ передается синхроимпульс (строб 2), позволяющий изменить сигналы на шине данных.

Шина управления она состоит из 3 сквозных линий. По этим линиям передается сигнал пуска – линия Z и сброса – линия C, сопровождаемая синхросигналом по линии S2 и сигналом по линии B, и сигнал запрета выполнения команды – линия – I.

Шина данных – она состоит из 24 сквозных линий W для сигналов записи и 24 сквозных линий R для сигналов чтения данных. Линии W и R объединяют в крейте только ФБПИ и к ФБК не подводятся.

Шина для нестандартных соединений – она состоит из 2 сквозных линий пользователя P1 и P2 и имеет по 3 пустых контакта пользователя P3 – P5 на каждом ФБ.

Для подачи к модулям питания имеется шина питания. Она состоит из 14 линий, 12 из них заняты, а 2 – V1 и V2 являются резервными.

Назначение шин и линий магистрали представлено в таблице 2.3, а контактов в разъеме в таблице 2.4

Таблица 2.3. Назначение шин и линий магистрали КАМАК.

Наименование шины		Назначение линий	Источник сигнала	Число контактов
Шина адреса		Адреса гнезда N Субадрес A1 – A4	ФБК ФБК	1 4
Шина операций		Команда F1 – F5	ФБК	5
Шина данных		Запись W1 – W24 Чтение R1 – R24	ФБК ФБПИ	24 24
Шина состояния (статуса)		Запрос L Ответ Q Команда принята X Занято B	ФБПИ ФБПИ ФБПИ ФБК	1 1 1 1
Шина синхронизации		Строб S1 Строб S2	ФБК ФБК	1 1
Шина управления		Пуск Z Сброс C Запрет I	ФБК ФБК ФБК	1 1 1
Шина Питания	Основное питание	+24В : 1А, 6А	-	1
		+6В: 2А, 25А	-	1
		0В	-	2
		-6В: 2А, 25А	-	1
		-24В: 1А, 6А	-	1
	Дополнительное питание	+200В	-	1
		+12В	-	1
		-12В	-	1
		~117В, фаза	-	1
		~117, нуль	-	1
	Земля E с сигналом	-	1	
	Резерв V1 и V2	-	2	

Таблица 2.4. Назначение контактов в разъеме крейта.

	Номер гнезда	
	От 1 до 24	25
	Сторона А платы	
1 – 5	P1 – P5	P1 – P5
6 – 8	X, I, C	X, I, C
9 – 10	N, L	P6, P7
11 – 12	S1, S2	S1, S2
13 – 24	W24 – W2	L24 – L2
25 – 36	R24 – R2	L12 – L1
37 – 39	-12, +200, ~117	-12, +200, ~117
40 – 43	V1, +12, V2, 0	V1, +12, V2, 0
Сторона В платы		
1	В	В
2 – 6	F16, F8, F4, F2, F1	F16, F8, F4, F2, F1
7 – 10	A8, A4, A2, A1	A8, A4, A2, A1
11 – 12	Z, Q	Z, Q
13 – 24	W23 – W1	N24 – N13
25 – 36	R23 – R1	N12 – N1
37 – 39	- 24, - 6, ~117	- 24, - 6, ~117
40 – 43	Земля Е, +24, +6, 0	Земля Е, +24, +6, 0

Как видно система шин крейта содержит 82 магистральные линии сигналов, проходящих через все стадии и 2 индивидуальных (радиальных) линии идущих от ФБК крейта к каждой станции. Таким образом, в крейте используется смешанная (радиально-магистральная) система шин.

Из 86 контактов разъема стандарт регламентирует использование 81 контакта ФБПИ и 79 контактов ФБК.

Чтобы проиллюстрировать логический поток КАМАК рассмотрим процедуру обмена сигналами на магистрали во время командных операций (рис. 2.7). Штрихованные области соответствуют возможным задержкам сигналов относительно начала очередного цикла на магистрали.

Сигнал занятости ФБК В сопровождает любую командную операцию. Любая командная операция сопровождается сигналами NAF. Это видно на временной диаграмме – линии В, N, А, F переходят в низкое состояние, соответствующее логической единице. Далее ФБПИ выставляет на X и Q низкий потенциал – команда принята и соединение состоялось, а также устанавливает сигналы чтения W и записи R данных. С учетом того, что длительность фронтов не превышает 150 нс, а задержка 100 нс к моменту 400 нс,

когда срабатывает строб 1 на линиях параллельно выставлены двоичные сигналы В, N, А, F, X, Q, и R (или W) всего число 37 сигналов.

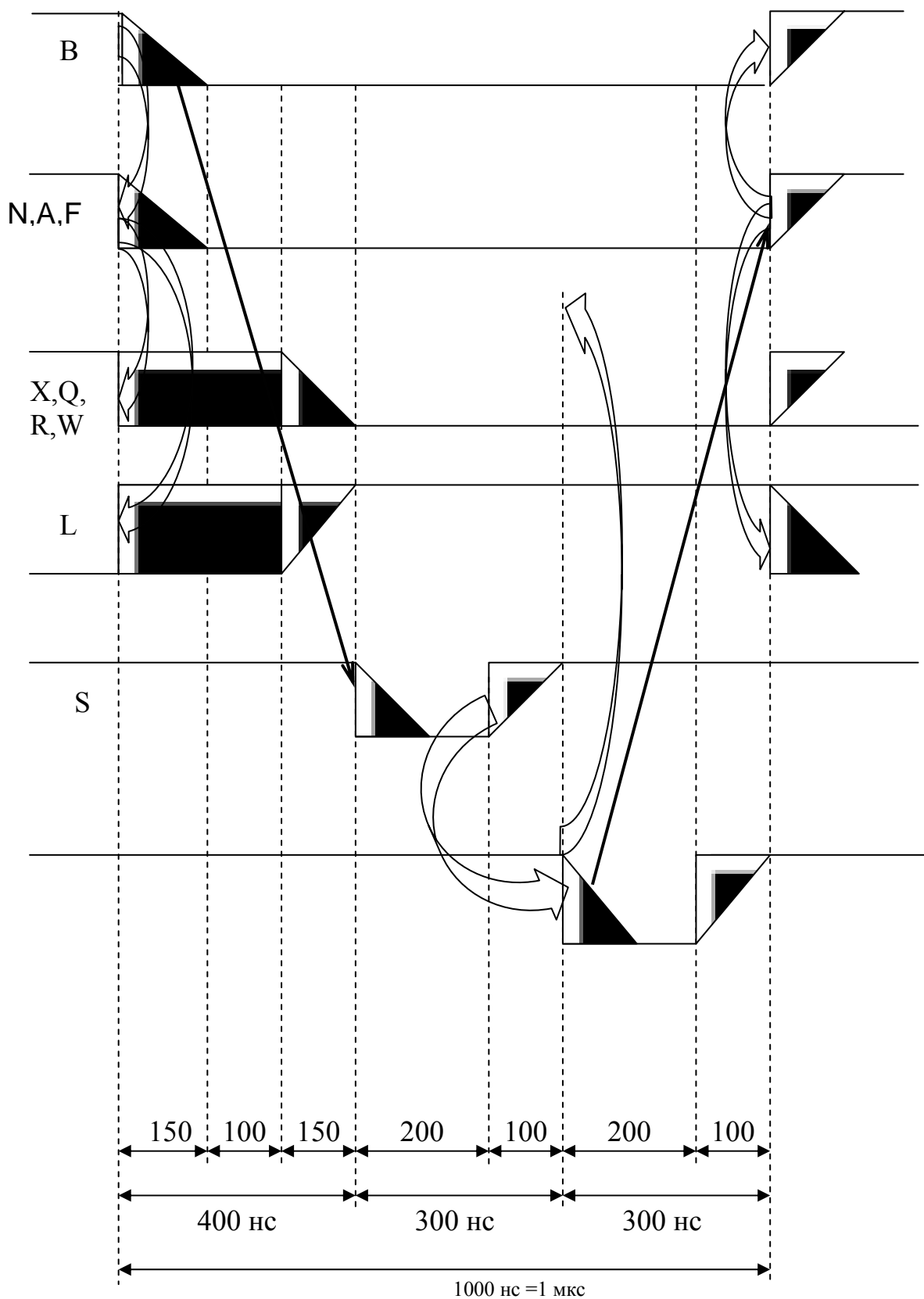


Рисунок 2.7. Обмен сигналами во время командных операций.

Исполнительные строб сигналы S1 и S2 длительностью по 200 нс каждый задержаны относительно начала цикла, для того, чтобы обеспечить достижения на B, N, A, F, R, W, X, Q уровня логической единицы.

С момента появления строба S1 контроллер анализирует сигнал на линии X, причем этот сигнал должен сохраняться неизменным по крайней мере до строба S2.

Действия в модуле, обусловленные функцией F начинается в момент появления одного из стробов S1 или S2 в зависимости от вида операции.

С момента появления строба S1 в течении 300 нс до появления строба S2 происходит действие ФБПИ по чтению данных с линий R или контроллера по записи данных с линий W. По строб сигналу S2 состояние линий R и W может быть изменено.

Сигнал на линии Q устанавливается адресуемым модулем и воспринимается ФБК с момента появления строб сигнала S1.

Сигналы запроса на линии L могут возникать в произвольные моменты времени, но удовлетворение запроса может произойти только лишь в первые 400 нс цикла. Во время действия и исполнения команды, адресованной модулю, как правило реализуется случай $L=0$ при $N=1$ (см. рисунок).

По окончании операции L-запрос вновь появляется на магистрали крейта, если он не был удовлетворен.

Кроме описанных адресуемых командных операций на магистрали крейта могут осуществляться безадресные операции, которые не адресованы конкретным модулям, с использованием управляющих сигналов Z, C и I. Рассмотрение в этом случае проводится аналогично. Длительность цикла в этом случае 750 нс, т.е. несколько короче цикла командной операции.

Если учесть, что передача информации по шине данных происходит только в командном цикле, а число линий R и W равно 24, то магистральный объем полезной информации, передаваемой за один цикл, 24 бита. Следовательно предельная пропускная способность магистрали КАМАК - 3 мегабайта/с.

Если в составе ФБПИ имеется несколько субблоков и один из них выставил флаг (сигнал запроса на линии L), то для нахождения адреса этого субблока ФБК крейта выполняет последовательный опрос всех субадресов A. Только после его нахождения дается сигнал разрешения на обмен информацией по шине данных. В результате пропускная способность КАМАК будет ниже расчетной.

Одного крейта с 23 модулями - станциями как правило недостаточно для автоматизации серьезного эксперимента. Это обстоятельство побудило разработчиков организовать в системе 2^{ой} уровень централизации управления обменом информацией, который носит название ветви (Grauch) или вертикальной магистрали.

Магистраль ветви может объединять до 7 крейтов (рисунок 2.8), которые обычно располагаются в одной стойке.

Ветвь состоит из следующих шин:

Шина адреса и команд ветви - число линий 21 в том числе 7 радиальных линий BCR для передачи адреса крейта, 5 магистральных линий адреса модуля в крейте BN16, BN8, BN4, BN2 и BN1 и 4 магистральных линии субадреса BA8, BA4, BA2 и BA1. Кроме того шина содержит 5 магистральных линий BF16, BF8, BF4, BF2 и BF1 для передачи кода операции в адресованный субблок.

Информационная шина ветви - число линий 24, назначение - запись и чтение информации по магистральным линиям BRW1 - BRW24.

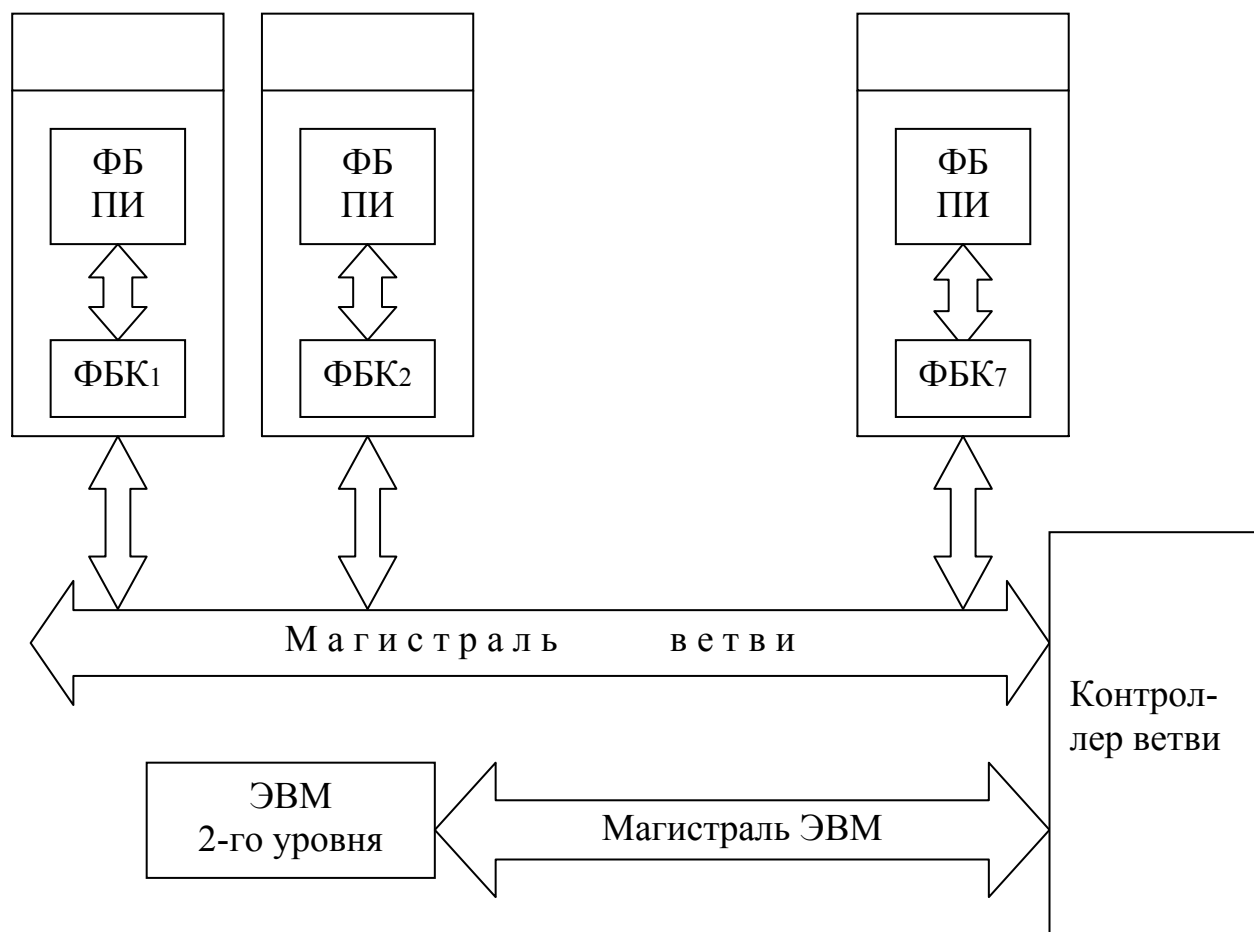


Рисунок 2.8 Структура с вертикальной магистралью.

Шина состояния - число линий 13, в том числе 7 радиальных линий ВТВ1 - ВТВ7 для передачи сигнала запроса на обслуживание крейта, линии VX, VQ, VZ аналогичные по назначению линиям X, Q и Z крейта, линия ВТА для передачи сигналов таймера и линии BD и BG для передачи сигнала запроса на обслуживания крейта и разрешение крейтам на занятие магистрали соответственно.

Резервная шина - число линий 7, из которых BV1 - BV5 используются по усмотрению пользователя, а BV6 и BV7 - зарезервированы.

Таким образом общее число линий ветви 65. Каждая из них представляет собой витую пару проводов. Магистраль ветви использует 132 контактные штырьковые разъемы.

Если экспериментальная установка занимает большую площадь можно перейти от системы с сосредоточенными параметрами к системе с распределенными параметрами. В стандарте КАМАК для этого случая разработали стандарт на последовательную магистраль КАМАК, которая позволяет объединять до 62^x крейтов в кольцо большой протяженности (рисунок 2.9).

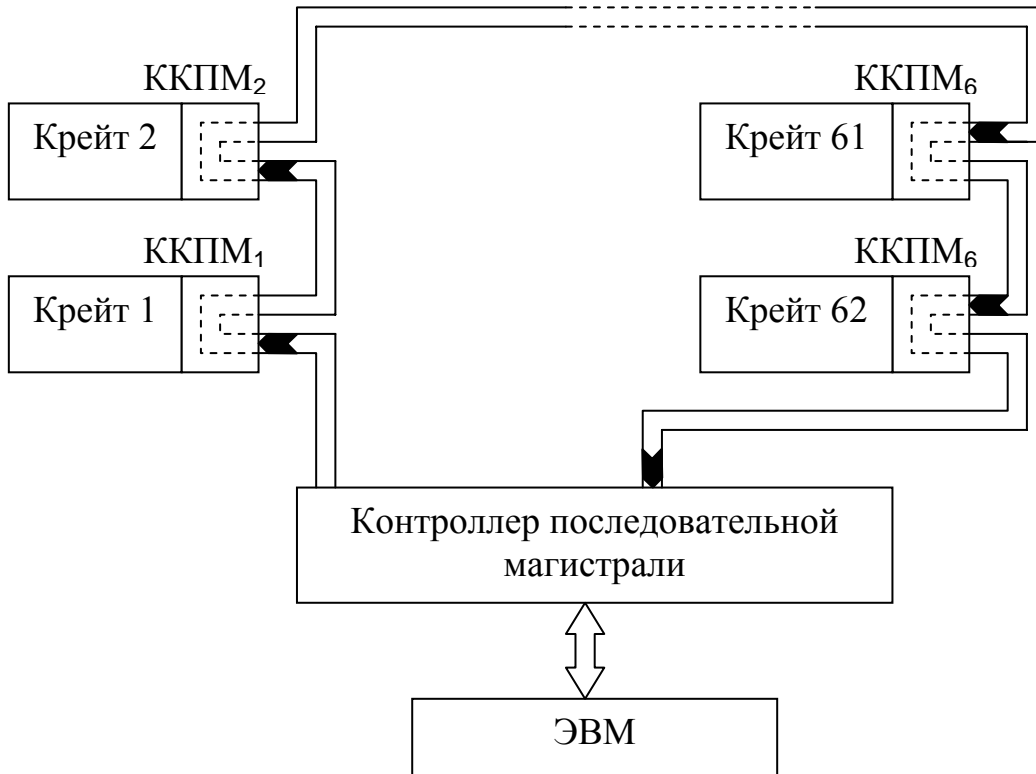


Рисунок 2.9. Последовательная магистраль КАМАК.

Во всех крейтах последовательной магистрали устанавливают одинаковые контроллеры, которые получили название контроллеры типа L. Обмен информацией между ЭВМ и контроллерами типа L происходит через контроллер магистрали - драйвер последовательной магистрали. Таким образом при замене ЭВМ меняется лишь драйвер, все остальные компоненты остаются неизменными.

Последовательная магистраль КАМАК (ПМ) может быть выполнена в 2^x вариантах, а именно: для побитной и побайтной передачи. В обоих случаях в составе магистрали имеется дополнительная линия для передачи тактовых импульсов, частота которых 5 МГц.

При побитной передаче байт информации обрамляется стартовым (логический 0) и стоповым (логическая единица) битами (рис. 2.10) и получается декада.



Рисунок 2.10. Структура сообщения с поразрядно передаваемым байтом.

Для однозначного определения начала и конца декад в сообщении предусмотрены паузы длиной 10 тактов, заполненные логическими единицами.

Таким образом на 20 бит передается всего 8 бит полезной информации, что при $f_{\text{такт}}=5\text{МГц}$, составит 2 Мбит/с.

При побайтной передаче ПМ состоит из $8^{\text{ми}}$ линий и линии для передачи тактовых импульсов. Определены 3 типа сообщений - командные, ответы и заявки на магистраль. Структуру соответствующих сообщений можно найти в литературе.

Литература к главе 2:

1. Блохнин С.М. Шина ISA персонального компьютера IBM PC/AT. – М.: ПК "Сплайн", 1992 – 76с.

2. Петрухин В.С., Степченков Ю.А., Филин А.В. Персональные ЭВМ на основе архитектуры Intel 80386. Книга 2. – Обнинск: ИнвесКо, 1993 – гл. 4.

3. Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC. /Под ред. У.Томпкинса, Дж. Уэбстера; пер. с англ. – М.: Мир, 1992 – 512с.: ил.

4. Новиков Ю.В., Калашников О.А., Гуляев С.Э. Разработка устройств сопряжения для персонального компьютера типа IBM PC. Под общей ред. Ю.В. Новикова. Практ. Пособие. – М.: ЭКОМ, 1997 – 224с.

5. Приборно – модульные универсальные автоматизированные измерительные системы. Справочник. /В.А. Кузнецов, В.Н. Строителев, Е.Ю. Тимофеев и др.; Под ред. В.А. Кузнецова. – М.: Радио и связь, 1993. — 304с.: ил.

Глава 3. Вычислительные машины системы и сети.

Концепция открытых систем возникла в результате эволюции вычислительных сетей и связанных с ними новых информационных технологий. В то же время сами вычислительные сети явились продуктом совершенствования вычислительной техники и программирования. Поэтому, чтобы подойти к центральному понятию этой главы, понятию открытой вычислительной сети, целесообразно кратко рассмотреть исторический путь развития компьютерной отрасли.

Примечание. Технология – реализация научных и технических знаний в практической деятельности. Технологии, направленные на повышение эффективности работы с информацией, называются информационными технологиями. Те из них, которые основаны на использовании компьютеров, получили название новых информационных технологий. Примерами могут служить технологии работы с текстами, таблицами и графикой, электронная почта, мультимедиа, "всемирная паутина" и т. п.

Начнем с определения . **Вычислительная машина (ВМ)**, иначе **компьютер**, представляет собой систему, состоящую из аппаратных (Hardware) и программных (Software) средств и предназначена для приема, обработки , хранения и выдачи информации.

Первые ВМ представляли собой уникальные лабораторные установки, на которых мог работать ограниченный круг высококвалифицированных специалистов, хорошо знающих их устройство. Каждая машина имела свой оригинальный набор команд и специфическую разрядную сетку (например, 27 или 36 разрядов). Программа представляла собой последовательность двоичных кодов команд и данных и вводилась в машину вручную с помощью переключателей. Управление работой машины – ввод программы, ее использование и вывод результатов – так же осуществлялось вручную с помощью кнопок на пульте.

Вместе с тем опыт использования даже таких несовершенных машин показал их высокую эффективность при решении сложных научно-технических задач и указал пути их совершенствования. Одновременно было осознано ,что развитие ВМ затрагивает широкий круг людей, организаций и предприятий (как производителей так и потребителей) и оно немыслимо без широкой унификации и стандартизации аппаратных и программных решений. Унификация и стандартизация в области ВМ и программирования породила целый ряд новых понятий, таких как интерфейс, протокол, спецификация, формат, кадр и т.д., о чем мы будем говорить подробнее в дальнейшем.

Примечание. Стандартизация –это деятельность, направленная на достижение оптимальной степени упорядочения в определенной области

посредством установления положений для всеобщего и многократного использования в отношении реально существующих или потенциальных задач.

Стандарт – нормативный документ по стандартизации.

Унификация предшествует стандартизации и заключается в перенесении удачных решений на вновь создаваемые изделия. Унификацию иногда именуют стандартом "де-факто", а стандарт как нормативный документ – стандартом "де-юре".

Следующим шагом в эволюции ВМ было появление мощных и надежных ВМ универсального назначения. Программирование было облегчено и унифицировано за счет разработки алгоритмических языков высокого уровня: Фортран, Алгол, PL/1 и т.д.. Как стандарт "де-факто" появился байт – единица количества информации в восемь бит. Чтобы автоматизировать управление работой ВМ, были разработаны операционные системы. Для удовлетворения потребностей разных категорий пользователей были созданы ряды ВМ – семейства программно совместимых ВМ различной производительности.

***Примечание. Операционная система (ОС)** – совокупность системных программ, основными функциями которых являются управление режимами работы ВМ, процессами и ресурсами. К **режимам работы** относятся ввод-вывод, трансляция и исполнение программы, тестирование и диагностика машины и т.д. Понятие **процесса** возникло с появлением мультипрограммных машин, и его наиболее распространенное определение – программа в стадии выполнения. Процесс может создаваться, прерываться, возобновляться, блокироваться, менять свой приоритет и т.д. **Ресурсом** называется любая часть аппаратного или программного обеспечения машины, которая может быть выделена процессу. Главными ресурсами являются процессоры, память, устройства ввода-вывода, файлы, базы данных.*

В операционной системе выделяют ядро, которое резидентно размещается в оперативной памяти и состоит из наиболее часто исполняемых программ, и нерезидентные компоненты, размещаемые во внешней памяти и вызываемые по мере надобности.

Расширился спектр периферийных устройств (устройства ввода и вывода информации), что поставило задачу разработки унифицированного сопряжения между этими устройствами и ЭВМ. Так возникло важное понятие интерфейса как унифицированной системы технических соединений (провода, шины, разъемы), логических сигналов (информационные и управляющие биты и байты с фиксированными электрическими уровнями "0" и "1") и алгоритмов обмена (процедуры установления связей, обмена данными и завершения передачи).

Машины универсального назначения имели большие размеры, требовали значительного штата обслуживающего персонала и поэтому эксплуатировались, как правило, в составе вычислительных центров (ВЦ), которые

предоставляли услуги пользователям в режиме пакетной обработки. Пакетная обработка обеспечивает наибольшую загрузку процессора, но лишает пользователя интерактивного взаимодействия с ВМ, что резко снижает эффективность его работы. Так, например, сдав программу с данными в ВЦ, на другие сутки пользователь узнавал, что программа не выполнялась, так как одна из перфокарт содержала ошибку.

Интересы пользователей стали учитываться с шестидесятых годов, когда начали развиваться интерактивные многотерминальные системы разделения времени. В пределах небольшого временного интервала не слишком заметного для человеческого восприятия, каждому пользователю выделяется свой квант времени для работы с компьютером. Возникал эффект единоличного диалога с ВМ.

Терминалы вышли за пределы ВЦ и рассредоточились по отделам, лабораториям, цехам. Хотя обработка данных осталась централизованной, но некоторые функции, такие как ввод и вывод информации, стали уже распределенными. Был сделан первый шаг к вычислительным сетям.

3.1. Вычислительные системы

Вычислительные системы стали возникать с появлением задач, которые было невозможно решить даже на самых мощных компьютерах. Для повышения производительности стали создавать многомашинные вычислительные системы (ММВС) и многопроцессорные вычислительные системы (МПВС). Отметим, что ММВС и МПВС могут также строиться для решения другой задачи – повышения надежности вычислительных машин, но здесь мы ограничимся только вопросом повышения производительности.

Многомашинная вычислительная система – включает в себя несколько компьютеров, каждый из которых работает под управлением собственной операционной системы, а также программные и аппаратные средства, обеспечивающие работу всех компьютеров как единого целого. Связь между компьютерами может осуществляться через общую дисковую память или через специально построенную общую оперативную память (ООП) и адаптер канал-канал (АКК), связывающий каналы ввода-вывода. ООП и АКК часто называют устройствами комплексирования машин, а саму многомашинную систему с такими устройствами называют **вычислительным комплексом**. Производительность ММВС является высокой на слабосвязанных задачах и падает, если параллельно выполняемые задачи тесно связаны между собой по данным.

Многопроцессорная вычислительная система включает в себя несколько центральных процессоров (ЦП), работающих под управлением единой операционной системы, имеющих общую оперативную память и общие периферийные устройства.

Примечание. Центральный процессор выполняет в ВМ основные функции по обработке данных и вместе с тем управляет работой всех других частей ВМ. Состоит из арифметико-логического устройства, устройства управления и процессорной памяти.

Высокая производительность МПВС достигается в следствии параллельной работы процессоров и быстрого их взаимодействия за счет общей оперативной памяти. Вместе с тем МПВС обычно имеет, при прочих равных условиях, более высокую стоимость чем ММВС.

3.2. Вычислительные сети

Как уже отмечалось, появление режима разделения времени на ВМ дало возможность вынести терминалы пользователей за пределы ВЦ и тем самым сделать первый шаг к вычислительным сетям. Следующим шагом стало появление телеобработки данных – доступа к компьютеру с терминалов, удаленных на большие расстояния, с помощью телефонных линий связи. Создание систем телеобработки данных позволило наряду с удаленными связями типа терминал-компьютер реализовать удаленные связи типа компьютер-компьютер. Так исторически возникли первые глобальные вычислительные сети. В этих сетях были реализованы такие ставшие теперь обычными сетевые службы, как обмен файлами, электронная почта и другие.

Локальные сети появились несколько позднее и своим появлением они обязаны мини-ЭВМ. Низкая стоимость мини-ЭВМ привела к их распространению во многих подразделениях предприятий. Далее предприятия стали объединять свои компьютеры и разрабатывать программное обеспечение для их взаимодействия. В результате широким фронтом стали возникать локальные вычислительные сети с разнотипным сопряжением и программным обеспечением. Стандартные технологии объединения в сеть утвердились только к середине 80-х годов, когда на смену мини-ЭВМ пришли персональные микро-ЭВМ.

Рассмотрим общие принципы построения вычислительных сетей. Сеть объединяет множество компьютеров. Но если в многомашинной системе компьютеры работают как единое целое и территориально находятся рядом, то в вычислительной сети компьютеры в значительной степени автономны и территориально могут быть распределены. Связь между компьютерами осуществляется с помощью специальных периферийных устройств – сетевых адаптеров, соединенных каналами связи.

После приведенных пояснений можно перейти к определению. **Вычислительной сетью** называется множество компьютеров, соединенных средствами связи и работающих под управлением сетевого программного и аппаратного обеспечения. Основная цель создания сети – обеспечения совместного использования всех ресурсов сети.

Различают локальные и глобальные сети. К **локальным сетям** относят сети компьютеров, сосредоточенных на небольшой территории (обычно в радиусе до 1 - 2 км). **Глобальные сети** объединяют компьютеры или локальные сети, расположенные на больших расстояниях, например, в разных городах и даже странах.

Основные отличия локальных сетей от глобальных сводятся к следующему. В локальных сетях прокладывают собственные качественные линии связи: коаксиальный кабель, витые пары, оптоволоконный кабель. Такие каналы позволяют вести передачу данных немодулированными сигналами с высокой скоростью (10, 16 и 100 Мбит/с), сравнимой со скоростью работы дисковых устройств, внутренних шин обмена ВМ и т.п. В результате, у пользователя сети, подключенному к удаленному разделяемому ресурсу, например диску, возникает ощущение, что он пользуется этим диском, как своим.

В отличие от локальных сетей, в глобальных сетях применяют уже существующие каналы связи (телефонные, телеграфные, радиорелейные, спутниковые и т.п.). Из-за низкой надежности этих каналов передача данных ведется модулированными сигналами с контролем передачи и подтверждением получения, а также повторной передачей искаженных или неполученных данных. Для глобальных сетей характерны значительно более низкие скорости передачи данных – 2400, 9600 бит/с, 28,8; 33,6; 56 и 64 Кбит/с – и только на магистральных каналах – до 2 Мбит/с.

Отличаются локальные сети от глобальных и набором услуг, предоставляемых пользователю. В локальных сетях набор услуг несравненно шире.

Далее в этой главе в основном будут рассматриваться локальные сети.

3.3. Сетевые службы и услуги

Облик той или иной сети для пользователя определяется, прежде всего, тем набором услуг, которые он может получить, например, прочитать удаленный файл, послать почтовое сообщение или распечатать свой документ на "чужом" принтере. Сетевой компонент (обычно программный), который предоставляет в распоряжение пользователя некоторый определенный набор услуг, называется **сетевой службой**. В качестве примеров можно назвать службу удаленного доступа, службу электронной почты, службу печати, а также столь необходимую файловую службу.

Рассмотрим принципы организации служб сети. При создании некоторой службы, например, файловой, помимо соединения машин каналами связи и снабжения их сетевыми адаптерами, необходимо еще сделать добавления к операционным системам компьютеров. На тех компьютерах, файлы которых должны быть доступны пользователям сети, необходимо добавить программные модули, которые будут постоянно находиться в режиме ожидания запросов, поступающих по сети от других машин. Обычно такие модули называются **серверами**. На компьютерах, пользователи которых хотят получить доступ к этим файлам, также нужно добавить к операционной системе

уже другие программные модули, которые должны вырабатывать запросы на доступ к удаленным файлам и передавать по сети на нужный компьютер. Такие модули обычно называются программными **клиентами**.

Каждая пара модулей "клиент-сервер" обеспечивает совместный доступ пользователей к определенному виду ресурсов, в нашем примере – к файлам. В этом случае говорят, что пользователь имеет дело с файловой службой. По этой же схеме организуются другие виды служб (рис. 3.1).

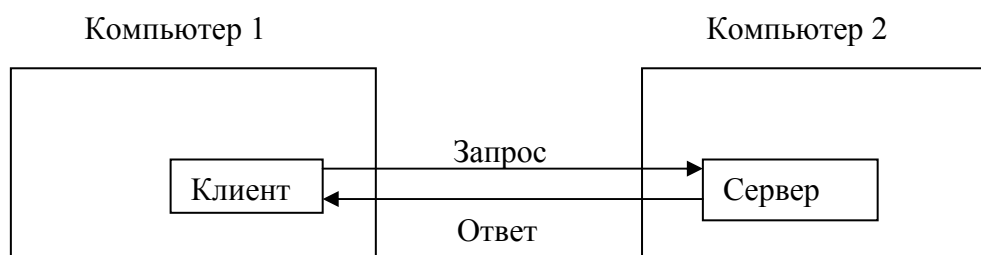


Рис. 3.1. Взаимодействие "клиент-сервер"

Термины "клиент" и "сервер" используются не только для обозначения программных модулей, но и компьютеров сети. Компьютер, предоставляющий свои услуги, называют сервером, а компьютер, потребляющий услуги, – клиентом. В частности, один и тот же компьютер может выступать и клиентом, и сервером.

Мы рассмотрели самые общие вопросы организации вычислительных сетей. Однако даже на таком уровне описания ясно, что создание реальной сети упирается в проблему совместимости разнообразного оборудования и программного обеспечения. Совместимость же, как известно, наиболее успешно реализуется путем стандартизации, к которой мы и переходим.

3.4. Многоуровневый подход. Интерфейс. Протокол. Стек протоколов

Организация взаимодействия между машинами в сети является сложной задачей. Как известно, для решения сложных задач успешно применяется метод декомпозиции, т.е. разбиение одной сложной задачи на несколько более простых задач – модулей. Использование метода декомпозиции становится еще более эффективным, если все множество модулей удастся разбить на иерархически взаимодействующие уровни. Это означает, что для выполнения своих задач каждый уровень обращается только к соседнему нижележащему уровню, а результаты своей работы передает только соседнему вышележащему уровню. Такая иерархическая декомпозиция предполагает четкое определение функций каждого уровня и интерфейсов между соседними уровнями.

Если данную многоуровневую концепцию применить к взаимодействию компьютеров в сети, то, во-первых, нужно учесть, что каждый компьютер, с точки зрения сети, представляет собой многоуровневую систему, уровни которой взаимодействуют в соответствии с четко определенным набором правил, называемым **интерфейсом**, а во-вторых, следует принять во внимание, что в процессе обмена сообщениями участвуют две машины, и следовательно, каждая пара одноименных уровней обеих машин также должна согласованно работать по единым правилам, которые получили название **протокола**.

В сущности, протокол и интерфейс выражают одно и то же понятие – набор формализованных правил, определяющих последовательность и формат сообщений, – но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия моделей одного уровня в разных узлах, а интерфейсы – модулей соседних уровней в одном узле.

Средства каждого уровня должны отрабатывать свой собственный протокол, а также интерфейсы с соседними уровнями. Иерархически организованный набор протоколов, достаточный для организации машин в сети, называется **стеком коммуникационных протоколов**.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Программный модуль, реализующий некоторый протокол, часто для краткости также называют "протоколом". При этом соотношение между протоколом – процедурой и протоколом – программой аналогично соотношению между алгоритмом и реализующей его программой.

Теперь, после рассмотрения основных идей многоуровневого подхода, перейдем к конкретной многоуровневой модели, играющей важную роль в теории и практике компьютерных сетей.

3.5. Модель OSI

Многоуровневый подход к задаче передачи сообщений давно и хорошо известен. На этом подходе основана работа знакомой всем почтовой связи. Поэтому целесообразно, прежде чем переходить к изложению сетевой многоуровневой модели, кратко рассмотреть упрощенный пример взаимодействия двух корреспондентов, живущих в разных городах, с помощью обычной почты. В этом примере можно выделить семь уровней модели связи (рис.3 2).

На рисунке представлены названия уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический. А также описаны функции каждого уровня для отправителя и получателя. Для трех верхних уровней приведены также протоколы и интерфейсы. Читателю предлагается в качестве упражнения определить протоколы и интерфейсы других оставшихся уровней.

Обратим внимание, что в рассмотренном примере реальная физическая связь существует только на нижнем физическом уровне. Все остальные связи между уровнями, хотя и регламентируются соответствующими протоколами, но являются не физическими, а виртуальными. Так, например, соединение между отправителем и получателем могло бы быть физическим, если бы отправитель лично съездил в другой город к получателю и сам вручил ему письмо. В реальной жизни отправитель опускает письмо в почтовый ящик в своем городе, т.е. довольствуется виртуальной связью.

Вернемся теперь к многоуровневой модели. В начале 80-х годов ряд международных организаций разработали модель, которая до сих пор играет значительную роль в развитии сети. Эта модель называется **моделью взаимодействия открытых систем (Open System Interconnection, OSI)** или моделью OSI. Модель OSI так же определяет семь уровней взаимодействия систем, аналогично рассмотренным в примере, дает им такие же стандартные имена и указывает, какие функции должен выполнять каждый уровень (рис. 3.3).

На рисунке 3.3 показаны принципы организации обмена сообщениями между двумя компьютерами сети на всех семи уровнях взаимодействия, структура сообщений на каждом уровне, места расположения протоколов и интерфейсов в процессе обмена. Рассмотрим пример. Пусть какому-либо приложению компьютера 1 необходимо записать данные в удаленный файл, находящийся на компьютере 2.

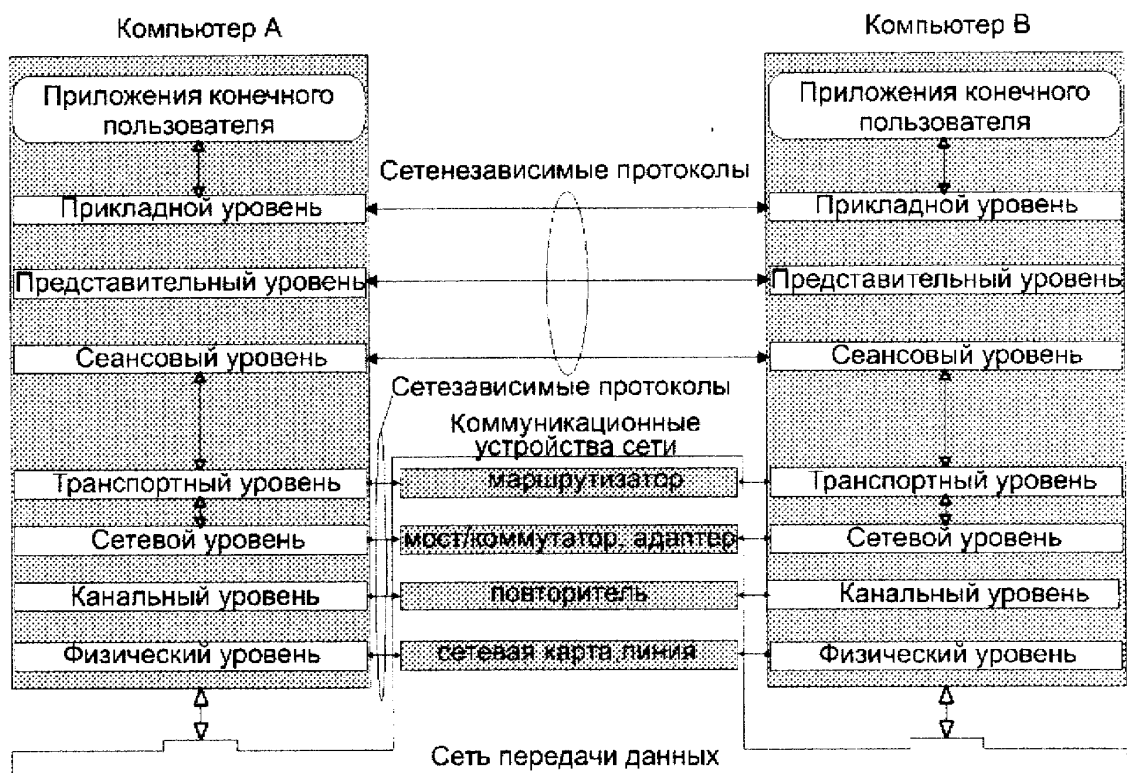


Рис. 3. Сетезависимые и сетезависимые уровни модели OSI

теории передачи данных формат этого сообщения состоит из одного поля-поля данных. Далее приложение обращается с запросом к прикладному уровню, в нашем примере - к его файловой службе. На основании этого запроса на прикладном уровне формируется сообщение, состоящее из двух полей: поля данных, несущего исходную полезную информацию, и поля заголовка, более кратко – заголовок, несущего служебную информацию. В заголовке сообщается о месте нахождения файла (в нашем примере - в компьютере 2) и о типе операции, которую нужно выполнить с файлом. Сформированное сообщение прикладной уровень направляет вниз по стеку представительному уровню. Представительный уровень добавляет к сообщению собственную служебную информацию – заголовок представительного уровня, в котором содержатся указания для представительного уровня компьютера 2. Наконец, на физическом уровне сообщение содержит заголовки всех уровней и в таком виде передается по линиям связи машине-адресату. В машине-адресате обработка сообщения идет в обратном порядке, с перемещением вверх с уровня на уровень путем последовательной обработки заголовков вплоть до выполнения нужной операции в файле компьютера 2.

Дадим краткую функциональную характеристику уровням модели OSI. Прежде, однако, сделаем существенное замечание. Модель OSI не включает в себя средства взаимодействия приложений и соответствующих им процессов. Взаимодействие приложений является важным направлением современных исследований, которые нами не рассматриваются.

Модель OSI начинается с прикладного уровня, на котором с помощью приложений пользователь создает сообщение. Для данного уровня сообщение - это блок структурированной информации. Примерами сообщений могут служить документ, рисунок и т.п. На прикладном уровне находятся сетевые службы, о которых уже говорилось.

Представительный уровень позволяет преодолевать синтаксические различия в представлении данных или же различия в кодах символов, например, кодов ASCII и EBCDIC. На этом же уровне выполняется шифрование и дешифрование, обеспечивающее секретность обмена данными.

Сеансовый уровень обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации, может проверять права пользователя на выход в сеть. На практике этот уровень используется редко. Функции сеансового уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Транспортный уровень обеспечивает трем перечисленным уровням надежную передачу данных, т.е. обнаружение и исправление ошибок передачи.

Как правило, протоколы транспортного уровня и более высоких уровней реализуются программными средствами - сетевыми операционными системами.

В качестве наиболее известного транспортного протокола можно назвать протокол TCP стека TCP/IP сети Internet. Протокол TCP нарезает из по-

тока байтов данных **сегменты** равной длины и передает их нижележащему уровню. После того, как эти сегменты будут доставлены (возможно, разными путями), протокол TCP снова соберет их в непрерывный поток байт.

Сетевой уровень определяет маршрут движения данных в сети. Сообщения сетевого уровня принято называть **пакетами**. Если объединяются несколько сетей, то адрес получателя состоит из старшей части – номера сети и младшей – номера узла в этой сети.

Примером протокола сетевого уровня является протокол межсетевого взаимодействия IP стека TCP/IP сети Internet. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на узел-получатель или на маршрутизатор, ведущий к следующей сети. Маршрутизатор - это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения.

Канальный уровень – это предпоследний сверху уровень сети и последний сетевой уровень компьютера. Средства физического уровня лежат за пределами машины. В компьютерах функции канального уровня реализуются сетевыми адаптерами и их драйверами.

В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. К типовым топологиям, поддерживаемым протоколами канального уровня, относятся общая шина, кольцо и звезда.

На канальном уровне обычно передача идет битами. Биты группируются в наборы, называемые **кадрами**. Для выделения кадра помещается специальная последовательность бит в его начало и конец. Кроме того, для кадра вычисляется его контрольная сумма, которая позволяет определить правильность передачи или зафиксировать ошибку.

Физический уровень лежит вне компьютера и имеет дело с передачей битов по физическим каналам связи, таким, например, как витая пара, коаксиальный кабель и т.п. Примером протокола физического уровня может служить один из протоколов технологии Ethernet, который определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100м, манчестерский код для представления данных, а также некоторые другие характеристики среды и сигналов.

При характеристике уровней модели OSI следует также отметить, что три нижних уровня – физический, канальный и сетевой – являются сетезависимыми, т.е. протоколы этих уровней тесно связаны с технической реализацией сети. Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и мало зависят от технических особенностей построения сети. Транспортный уровень является промежуточным и скрывает все детали функционирования нижних уровней от верхних.

Рассмотренная сетевая модель OSI вплотную подводит нас к такому важному понятию, как открытая система, и связанными с нею другими поня-

тиями: открытый характер стандартов и спецификаций, расширяемость и масштабность, мобильность, управляемость, прозрачность и т.п. Эти вопросы рассматриваются в последующих разделах пособия.

Литература к главе 3:

1. Кузнецов С.Д. Четверть века операционной системы UNIX.- Открытые системы, 1995, вып.1.Сс14-17.
2. Филинов Е.Н. Выбор и разработка концептуальной модели среды открытых систем.- Открытые системы, 1995, вып.6.С.71-77.
3. Липаев В.В., Филинов Е.Н. Мобильность программ и данных в открытых информационных системах.- М.: Научная книга, 1997. – 368с.

Глава 4. Информационно-вычислительные системы и сети

4.1. Понятие информационно-вычислительные системы и сети.

Информационно-вычислительные системы (ИВС) и сети представляют собой некий комплекс аппаратных средств (аппаратную платформу) и программных средств, позволяющих обеспечить выполнение информационных, вычислительных или управленческих процедур в самых разных областях современной жизни на основе информационных технологий с использованием компьютеров. Широкое распространение ИВС в самых разных областях современной жизни: промышленности, финансах, образовании и культуре, - делает актуальным вопрос о создании некоторых стандартов выполнения ИВС. Наиболее комплексно эти вопросы ставятся в научном направлении, которое определяет концепцию «открытых систем».

Определение понятия открытой системы дано, например, комитетом IEEE POSIX 1003.0. Из этого определения вытекает понятие открытой системы для ИВС как среды для прикладных программ, базирующейся на стандартных интерфейсах и обеспечивающей мобильность прикладных систем, персонала и взаимодействие (интероперабельность) систем. Таким образом важнейшими свойствами открытой ИВС будут:

- мобильность прикладных программ, т.е. возможность переноса программ с одной аппаратной платформы на другую с минимальными доработками или даже без них;
- мобильность персонала, т.е. возможность подготовки персонала для работы на ИВС с минимальными временными и трудовыми затратами;
- четкие условия взаимодействия частей ИВС и сетей с использованием открытых спецификаций.

Ключевой момент в обеспечении свойств открытых ИВС – использование открытых спецификаций, т.е. общедоступных спецификаций,

которые поддерживаются открытым, гласным согласительным процессом, направленным на постоянную адаптацию новым технологиям.

Важным моментом в концепции открытых систем для ИВС является то, что открытая система – это система, построенная на стандартных технических средствах – микропроцессорах и использующая стандартную операционную систему. Примером такой системы может служить персональный компьютер IBM PC или компьютер, совместимый с IBM PC. Большинство этих платформ используют одну и ту же операционную систему, и на них могут исполняться одни и те же прикладные программы.

4.2. Основные требования по обеспечению мобильности программ и данных в открытых ИВС.

Основными целями создания и применения концепции, методов и стандартов открытых систем является повышение общей экономической эффективности разработки и функционирования ИВС, обеспечение мобильности программ и данных. При создании сложных, распределенных ИВС, формировании их архитектуры, выборе компонентов и их связей следует учитывать ряд современных концептуальных требований:

- архитектура ИВС должна соответствовать текущим и перспективным целям, стратегическим функциональным задачам создаваемой системы;
- в структуре и компонентах следует предусматривать обеспечение максимально возможной сохранности инвестиций в аппаратные и программные средства при длительном развитии, сопровождении и модернизации ИВС;
- для обеспечения перспективы развития ИВС следует предусматривать возможность интеграции гетерогенных вычислительных компонентов и мобильность приложений в отношении различных аппаратных и операционных платформ на основе стандартов открытых систем.

Гибкость модификации ИВС при развитии обеспечивается рядом принципов и правил структурного построения ИВС и ее компонентов. Эти правила направлены на стандартизацию и унификацию структуры и взаимодействия компонентов разного ранга и назначения в пределах проблемной области. Основные принципы и правила можно объединить в группы, которые отражают:

- стандартизированную структуру программных средств определенного класса;
- унифицированные правила построения прикладных программных компонентов;
- унифицированные правила построения и организации межмодульного интерфейса прикладных программ;

- унифицированные правила внешнего интерфейса и взаимодействия компонентов прикладных программных систем с внешней средой, с операционной системой и другими типовыми средствами организации вычислительного процесса.

Таким образом для эффективной разработки прикладных программных систем (ППС) необходимо формулировать и соблюдать ряд принципов и правил структурного построения ИВС. Одним из важнейших и эффективных путей решения этой проблемы является концепция и совокупность стандартов открытых систем.

Первоначально концепция открытых систем развивалась на непосредственной поддержке разработчиками операционной системы UNIX с широким набором функций и ее модификациями. Однако в последнее время унифицированные интерфейсы открытых систем воспринимаются многими разработчиками новых операционных систем массового применения. Поэтому в последующем изложении предполагается, что ядро операционной системы позволяет использовать стандартизированные интерфейсы с приложениями, а расширения операционной системы выполняют самостоятельные функции и организационно в нее не входят. Таким образом в понятие операционной системы далее включается ее ядро и некоторые расширения, непосредственно поддерживающие управление вычислительным процессом и памятью.

В составе коммерческих операционных систем используются следующие крупные группы программных средств:

- поддерживающие функционирование и оперативное взаимодействие прикладных программ с операционной и внешней средой;
- обеспечивающие телекоммуникацию и обмен данными между территориально распределенными компонентами ИВС;
- обеспечивающие автоматизацию разработки и всего жизненного цикла мобильных ППС.

Средства поддержки оперативного взаимодействия ППС с внешней средой и пользователями доступны для стандартизации. В их состав входят программные компоненты взаимодействия с пользователями, управления заданиями, управления файлами и базами данных. Разумная глубина стандартизации позволяет консервировать средства непосредственного взаимодействия приложений с операционными системами и сохраняет возможность развития процессов и средств комфортного взаимодействия с пользователями, управления заданиями, а также управления файлами и базами данных.

Средства, обеспечивающие автоматизацию разработки и жизненный цикл ППС в наибольшей степени допускают унификацию в части языков программирования, а также структуры и состава документов, сопровождающих программы и описания данных. Наиболее унифицируемыми в этой группе являются современные CASE-технологии.

Средства обеспечения телекоммуникации и транспортировки данных имеют большое значение для обеспечения согласованного взаимодействия компонентов в распределенных ИВС. В этой области допустима и успешно проводится глубокая стандартизация не только концепций и методов взаимо-

действия компонентов, но также конкретных протоколов и профилей локальных и распределенных сетей. Компоненты являются открытыми друг для друга в том случае, если, несмотря на их программно-аппаратные различия, они могут взаимодействовать с помощью определенных процедур «прозрачных» для пользователя.

В решении поставленных выше задач по созданию открытых ИВС в общей концепции открытых систем развиваются два основных направления:

- открытых вычислительных систем (open computing systems – OCS), обеспечивающее возможность относительного простого и эффективного по трудоемкости переноса апробированных программных средств на различные типы аппаратных платформ за счет стандартизации процессов и интерфейсов взаимодействия ППС с операционными системами ЭВМ;
- взаимосвязи открытых систем (open systems interconnection – OSI), унифицирующее структуру, процессы и интерфейсы для совместимости методов и средств обмена данными между разнотипными удаленными ЭВМ.

Важнейшими объединяющими целями развития указанных двух направлений являются снижение трудоемкости и длительности создания, а также повышение качества и функциональных возможностей современных ИВС. Для каждого направления характерно развитие методов и стандартов, ориентированных преимущественно на его поддержку и реализацию. В данном разделе основное внимание будет уделено первому направлению.

Рабочей группой POSIX P1003.0 Института инженеров по электронике и электротехнике США – IEEE предложена Референсная Модель Среды Открытых Систем (OSE/RF). Данная модель предусматривает разбиение среды на три составные части: прикладное обеспечение, прикладная платформа, внешняя среда. Под прикладным обеспечением понимаются собственно прикладные программы, данные, а также документация и средства обучения пользователей. Прикладная платформа состоит из аппаратной платформы и системного программного обеспечения. К внешней среде относятся все системные компоненты, которые являются внешними по отношению к прикладной платформе и прикладному обеспечению. Это утилиты и подсистемы, реализуемые на других (удаленных) платформах, а также периферийные устройства.

Взаимодействие между прикладными средствами и платформой осуществляется с помощью интерфейса прикладных программ (API). Предусматривается четыре интерфейсных компонента для взаимодействия с различными услугами: системными службами, коммуникационными службами и службами, обеспечивающими человеко-машинный интерфейс.

Взаимодействие между прикладной платформой и внешней средой производится через область интерфейсов внешней среды (EET). Эта модель описывает также системы, построенные на основе архитектуры клиент-сервер, которые сейчас получили широкое распространение.

Наиболее актуальна стандартизация при коллективной разработке критических ППС для систем, к которым предъявляются высокие требования по качеству и живучести. Активная стандартизация за рубежом процессов жизненного цикла таких ППС проводится в основном по трем направлениям.

Первое направление организуется и стимулируется Международной организацией стандартизации (International Standards Organization – ISO) и Международной комиссией по электротехнике (International Electrotechnical Commission – IEC). Эти организации совместно создали и издали большое число стандартов, которые для краткости маркируются одной аббревиатурой ISO. На этом уровне осуществляется стандартизация наиболее общих технологических методов и процессов, имеющих значение для международной кооперации и разделения труда.

Второе направление активно развивается в США Институтом инженеров электротехники и электроники (IEEE) совместно с Американским национальным институтом стандартизации (ANSI). По этому направлению разработано наибольшее число стандартов в рассматриваемой области. Во многих случаях они служат базой для последующего создания стандартов уровня ISO/IEC. Стандарты ISO/IEC и ANSI/IEEE в основном имеют рекомендательный характер.

Третье направление стимулируется министерством обороны США (DOD). Создаваемые по этому направлению стандарты имеют во многих случаях характер обязательных для фирм, работающих по заказам министерства обороны США. Для разработки критических ППС наибольшее значение получил стандарт DOD-STD-2167A, регламентирующий процессы и документы от анализа требований к системе до завершения ее испытаний.

Профиль – это совокупность нескольких базовых стандартов (и других нормативных документов) с четко определенными и гармонизированными подмножествами обязательных и факультативных возможностей, предназначенных для реализации заданной функции или группы функций. Функциональная характеристика (заданный набор функций) объекта стандартизации является исходной информацией для формирования и применения профиля этого объекта или процесса. Профиль не может противоречить использованным в нем базовым стандартам и нормативным документам. На базе одной и той же совокупности стандартов могут формироваться и утверждаться различные профили для разных проектов и сфер применения.

В профилях сосредотачиваются наборы базовых стандартов из перечисленных выше групп, предназначенных для регламентации конкретных прикладных функций открытых систем. При этом могут использоваться не все положения, требования и параметры отобранных для профиля базовых стандартов, а также могут дополнительно применяться некоторые стандарты де-факто и формализованные спецификации, не отраженные в международных стандартах. При сертификации приложений могут также использоваться применяющиеся базовые стандарты и технические документы в соответствии с назначением и функциями ППС.

На стадиях жизненного цикла информационных систем выбираются и затем применяются основные функциональные профили: профиль прикладного программного обеспечения, профиль среды информационной системы, профиль защиты информации в системе, профиль инструментальных средств.

Особую роль в развитии проблемы обеспечения переноса программ на различные аппаратные платформы сыграла унификация операционной системы UNIX. Современные версии этой системы ориентированы на стандартизированное взаимодействие с прикладными программами и внешней средой, поэтому современный UNIX может рассматриваться в качестве хорошей иллюстрации ППС, построенной в соответствии с требованиями открытых систем [1, 2]. Это также объясняет преобладание в разработках и публикациях по мобильности программ систем на основе UNIX.

4.3. Стандарты, поддерживающие создание мобильных прикладных программ.

Опыт создания мобильных программных средств, обобщавшийся в последние годы, привел к необходимости разработки концепции и комплекса стандартов, обеспечивающих эффективную по трудоемкости переносимость прикладных программ между различными аппаратными и операционными платформами. Ядром стала группа стандартов, созданная специалистами США под эгидой IEEE под общим названием – Интерфейсы переносимых операционных систем (Portable operating system interface – POSIX). Проблему переноса программ сосредоточили на унификации интерфейсов операционных систем ЭВМ с различными прикладными программами, а также с окружающей средой. Эти стандарты не ориентированы на определенную конкретную архитектуру ЭВМ, однако предполагают использование современной операционной среды и прежде всего UNIX, как стандарта де-факто, а также международных стандартов на языки программирования и стандартов верхних уровней взаимосвязи открытых систем. В совокупности они образуют нормативную базу открытых компьютерных систем – OCS, обеспечивающих разработку переносимых программных средств.

Все стандарты POSIX имеют рекомендательный характер. Они не должны служить препятствием для переноса объектного кода, ограничивать или ухудшать исполнение приложений при стандартизированных интерфейсах и ограничивать формирование новых унифицированных интерфейсов по мере необходимости. Стандарты группы POSIX, регламентирующие интерфейсы мобильных программных средств с операционной средой, включают:

1. IEEE 1003.0 – Руководство по POSIX окружению открытых систем. Набор POSIX стандартов.
2. ISO 9945-1:1990 (IEEE 1003.1) – Информационная технология. Интерфейсы переносимых операционных систем. Часть 1: Интерфейсы систем прикладных программ (язык Си).

3. ISO 09945-2:1992 (IEEE 1003.2) – Информационная технология. Интерфейсы переносимых операционных систем. Часть 2. Команды управления и сервисные программы (Shell and utilities).
4. IEEE1003.3 – Методы тестирования для определения соответствия стандартам POSIX.
5. IEEE 1003.4 – Расширение к IEEE 1003.1 для переносимых операционных систем и прикладных программ реального времени.
6. IEEE 1003.5 – Интерфейсы переносимых приложений на языке Ада с привязкой к системным средствам POSIX 1003.1.
7. IEEE 1003.6 – Расширения для обеспечения защиты информации и безопасности переносимых прикладных программ.
8. IEEE 1003.7 – Расширения, связанные с системным администрированием.
9. IEEE 1003.8 – Сети передачи данных и сетевые интерфейсы для прозрачного доступа к файлам.
10. IEEE 1003.9 – Интерфейсы переносимых приложений на языке Фортран.
11. IEEE 1003.10 – Профиль прикладного окружения для суперкомпьютеров.
12. IEEE 1003.11 – Профиль прикладного окружения для распределенной диалоговой обработки транзакций.
13. IEEE 1003.12 – Протоколно независимые интерфейсы для коммуникации.
14. IEEE 1003.13 – Профиль прикладного окружения систем реального времени.
15. IEEE 1003.14 – Профиль прикладного окружения мультипроцессорных систем.
16. IEEE 1003.15 – Расширение для очередей пакетов прикладных программ, определяющие интерфейсы пользователей и администраторов при пакетной обработке.
17. IEEE 1003.16 – Развитие интерфейсов для переносимых приложений на языке Си.
18. IEEE 1003.17 – Справочные службы, обслуживание каталогов и пространство времен.
19. IEEE 1003.18 – Общие характеристики функциональной среды и профиль окружения прикладной платформы, соответствующей стандартам POSIX.
20. IEEE 1003.19 – Развитие приложений для переносимых приложений на языке Фортран.
21. IEEE 1003.20 – Развитие интерфейсов для переносимых приложений на языка Ада.
22. ISO 13210:1994. – Информационные технологии. Методы тестирования для измерения соответствия стандартам POSIX.

По функциональному назначению документы POSIX можно разделить на четыре группы стандартов [3]:

- базовые, определяющие общие принципы построения, директивы, основы реализации и методологию тестирования интерфейсов мобильных приложений, а также общее административное управление программами и данными (IEEE 1003.1, -2,-3,-4,-7);
- конкретизирующие интерфейсы с операционной платформой прикладных программ, разрабатываемых на трех базовых языках программирования – Си, Фортран и Ада (IEEE 1003.5, -9,-16,-19,-20);
- определяющие взаимодействие в распределенных открытых системах, телекоммуникацию в сетях, а также защиту информации (IEEE 1003.6, -8,-12,-15,-17);
- регламентирующие процессы создания, основные компоненты и структуру профилей прикладного окружения для интерактивного взаимодействия с пользователями, мультипроцессорных систем, суперкомпьютеров, систем реального времени (IEEE 1003.10, -11,-13,-14,-18).

Рассмотрим первую группу стандартов несколько подробнее. Идеология и модель создания мобильных программных средств изложена в документе IEEE 1003.0 – POSIX Guide –Руководство по POSIX окружению открытых систем, которое детализирует для пользователей модель комплекса стандартов POSIX, а также взаимодействующих с ними стандартов де-юре и де-факто и спецификаций, необходимых для создания переносимых приложений. Модель отражает принципы построения интерфейсов прикладных программ с платформой – операционной системой, через которую осуществляется взаимодействие с компонентами внешнего окружения. Прикладные программы непосредственно не взаимодействуют с внешним окружением, а связаны с ним только через операционную систему.

Таким образом определяющими являются два интерфейса между тремя базовыми компонентами: между прикладными программами и платформой – операционной системой (API) и между платформой и внешним окружением (EEI). Определены общие функции – услуги платформы для этих взаимодействий. Внешнее окружение включает компоненты человеко-машинного взаимодействия с пользователями, компоненты информационного взаимодействия с внешними устройствами ЭВМ и компоненты, обеспечивающие телекоммуникацию. Эти компоненты и услуги более детально должны быть формализованы соответствующими частными стандартами POSIX.

Детализация интерфейсов начинается в стандарте IEEE 1003.1 – Интерфейсы систем прикладных программ, в котором уточняется концепция переносимости и принципы ее обеспечения путем унификации интерфейсов прикладных программ с операционными системами. Основные команды управления и сервисные программы, обеспечивающие переносимость, изложены в стандарте IEEE 1003.2. В стандарте ISO 9945-2:1992 (1996), который является развитием стандарта IEEE 1003.2, изложено конкретное представле-

ние команд операционной системы и утилит, обеспечивающих унифицированное взаимодействие с мобильными прикладными программами.

Стандарт IEEE 1003.3 содержит общую методологию тестирования для проверки соответствия интерфейсов прикладных программ стандартам POSIX. Представлены принципы формирования наборов тестов и предлагается методика развития системы тестов для контроля создаваемых приложений.

Стандарт IEEE 1003.4 дополняет и конкретизирует интерфейсы программ, которые для мобильных приложений, функционирующих в реальном времени, требуют расширения административного управления. Стандарт охватывает три основных направления: файловые системы реального времени, согласование функционирования многокомпонентных архитектур ЭВМ, распределение памяти посредством семафоров.

Стандарт IEEE 1003.7 содержит системные интерфейсы и сервисные программы для управления при инсталляции, конфигурировании и сопровождении версий прикладного окружения операционных систем. Рассматривается административное управление со стороны пользователей системными файлами, сопровождением и развитием версий прикладных программ.

4.4. Базы данных в концепции открытых систем

В современных информационных системах хранение информации осуществляется в базах данных. Под базой данных понимают структурированную форму записи и хранения информационных ресурсов, позволяющую упростить поиск и сортировку данных. Требования концепции открытых систем обуславливают целый ряд особенностей построения баз данных (БД) и файловых систем современных компьютеров.

Стремление эффективно использовать память и производительность ЭВМ при различном объеме, содержании и взаимосвязи данных привело к разработке различных методов их структурирования и организации. Соответственно с организацией данных изменялись методы, средства и интерфейсы манипулирования ими со стороны пользователей, обеспечения их целостности, сохранности и защиты. При этом значительную роль играла необходимость обеспечения возможности экономного по трудоемкости переноса больших объемов информации на различные аппаратные и операционные платформы. Основная задача стандартов взаимодействия пользователей с данными, разрабатываемых в рамках концепции открытых систем, состоит в упорядочении и регламентировании организации и структуры данных, а также правил манипулирования ими: накопления, хранения, изменения и транспортировки для эффективного их использования.

В результате для управления и использования относительно небольших объемов слабо структурированных данных развиты методы и стандарты файловых систем, а для манипулирования сложными, взаимосвязанными данными созданы и развиваются методы и стандарты управления базами данных.

Между данными, характеризующими объекты, могут существовать связи, имеющие различный содержательный смысл и степень сложности. Для эффективной работы с разнообразными по структуре и содержанию данными необходимо учитывать эти особенности, что привело к созданию многих типов систем управления базами данных (СУБД). Основными из них являются сетевые, иерархические и реляционные, которые в свою очередь делятся на локальные и распределенные.

Внимание разработчиков БД направлено на достижение такой ясности и дружелюбности интерфейса пользователей ЭВМ, которые позволила бы создавать базы данных без специальных знаний из области СУБД. Одним из результатов является предпочтение многими разработчиками и пользователями реляционной модели данных в отношении сетевой или иерархической. Другой результат — создание объектно-ориентированного интерфейса конечного пользователя с использованием техники меню и окон для выполнения основных функций. Наряду с ориентацией на прикладного пользователя, большинство СУБД учитывает и потребности пользователя-программиста, предоставляя ему встроенный язык программирования. Используя этот язык, программист может реализовать нетривиальные приложения по обработке данных, ориентированные на конкретные потребности.

Международные стандарты, определяющие построение сетевых, реляционных и распределенных файловых систем и баз данных, исторически вошли в разные разделы системы международных стандартов (системы обработки информации — СОИ, информационные технологии — ИТ, взаимосвязь открытых систем — ВОС), однако тематически весьма близки. Среди этих стандартов наиболее известны и активно используются базовые стандарты построения файловых систем (FTAM) и стандарты, регламентирующие описание реляционных баз данных (SQL). Учитывая их большое, принципиальное значение для систем обработки данных, более подробно ниже представлены концептуальные основы этих стандартов.

В области стандартизации баз данных наиболее известны следующие международные стандарты:

1. ISO 8907:1987. СОИ. Язык описания сетевой базы данных (NDL).
2. ISO 9075:1989. СОИ. Язык описания реляционной базы данных (SQL): Дополнение 1:1992.
3. ISO 9579:1991. ИТ. ВОС. Доступ к удаленной базе данных (RDA).

В стандарте ISO 8907 определены синтаксис и семантика трех языков сетевой базы данных:

- язык описания схемы для определения структуры и условий целостности сетевой базы данных (СБД),
- язык описания подсхемы для определения пользовательской структуры базы данных;
- язык манипулирования данными для определения процедур базы данных и исполнительных операторов в конкретной прикладной программе, работающей с СБД.

Заданы логические структуры данных и базовые операции для СБД. Стандарт обеспечивает переносимость определений баз данных и прикладных, программ между аттестованными реализациями языка. Установлены два уровня языка, причем уровень 1 является подмножеством полного языка (уровень 2). Определены коды возврата. В трех информационных приложениях приведены примеры применения языка.

В стандарте ISO 9075 представлен язык SQL. Язык SQL, служит для удобного и понятного пользователям формулирования обращений к реляционным БД и манипулирования данными. Язык обеспечивает общее управление данными в приложениях, где требуется гибкость структур данных и маршрутов доступа, специфическое манипулирование данными и изменение степени ограничения доступа. Внимание акцентировано на обеспечении свойства целостности данных, т.е. глубокой взаимосвязи данных, в которой наибольшее значение имеет функциональная зависимость, что отличает их от набора независимых, совместно используемых файлов. Особенно полно механизм контроля целостности формализован на основе произвольных логических утверждений.

Язык SQL, в основном, является языком запросов и манипулирования данными, однако его функции и возможности значительно шире. Он обеспечивает построение эффективных диалоговых систем обработки транзакций со стандартизированными наборами параметризуемых запросов. В языке имеются средства:

- определения и манипулирования схемой БД;
- определения ограничений и контроля целостности;
- определения и уничтожения структур физического уровня для эффективной реализации транзакций;
- авторизации и защиты доступа к данным;
- использования точек сохранения транзакций и откатов при сбоях;
- средства динамической компиляции запросов, на базе которых возможна диалоговая их обработка.

Язык имеет строгую математическую основу на базе предикатов первого порядка. Группы по стандартизации и разработчики СУБД обеспечивают совместимость версий языка снизу вверх в последовательных расширениях языка. Стандартизованы синтаксис и семантика операторов выборки и манипулирования данными. Имеется возможность указания в запросе потребности группирования отношения — результата по указанным полям с поддержкой условий выборки на всю группу целиком. Формализованы средства ограничения целостности БД, включающие возможность определения первичных и внешних ключей отношений и проверочных ограничений целостности. В случае нарушения ограничений целостности предусмотрен откат транзакций в точку, непосредственно предшествующую операции манипулирования данными, или полный откат транзакции к ее началу.

Стандарт ISO 9579 регламентирует доступ к удаленной базе данных (RDA). Он предназначен для унификации установления дистанционного соединения между клиентом, действующим по запросу прикладной программы

или администратора данных, и сервером, который управляет обменом данными с базой данных. Это обеспечивает взаимодействие систем управления базами данных в гетерогенной информационной среде. Услуги RDA охватывают:

- административное управление диалогом между клиентом и сервером;
- управление ассоциацией;
- управление ресурсами;
- языки доступа к данным между отдельным клиентом и отдельным сервером.

Стандарт состоит из двух частей. Часть 1 определяет обобщенную модель доступа к удаленной базе данных и частную модель для доступа к базам данных с использованием языка баз данных SQL, а также операции над моделью базы данных в виде абстрактного взаимодействия между источником запроса и средством обслуживания базы данных. Приведены основные концепции RDA. В стандарте определяются услуги RDA, предоставляющие пользователю возможности управления транзакцией и манипулирования данными. Описан протокол, обеспечивающий эти услуги. Операторы SQL передаются в виде символьной строки с отдельным списком входных параметров с описанием особых случаев.

Часть 2 определяет специализацию удаленного доступа к базам данных с использованием языка баз данных SQL, операции над моделью данных в виде абстрактного взаимодействия между источником запроса и средством обслуживания базы данных. Определяются услуги RDA, предоставляющие пользователю возможности управления ассоциацией, управления ресурсом, управления транзакцией и манипулирования данными. Описан протокол, обеспечивающий эти услуги. Показано отображение услуг RDA на соответствующие элементы процедур. Приведены аттестационные требования к реализациям протокола.

Глава 5. Построение высокоэффективных информационных систем, относящихся к классу больших (открытых) систем

5.1. Свойства интерфейсовых систем информационного обмена и локальных вычислительных сетей.

5.1.1. Определение СИО и ЛВС.

В развитии современных подвижных объектов (ПОб) наблюдаются следующие тенденции:

- непрерывное увеличение объема, функциональной насыщенности и сложности задач, возлагаемых на ПОб, что приводит к резкому увеличению на борту вычислительных средств: БВМ, БЦВМ, БЦВС и, как следствие, к усложнению СИО и ЛВС;

- резкое увеличение количества бортового электронного оборудования различных модификаций, решающего задачи на различных типах ПОб, что требует вариантного применения последовательных модификаций, а также выработки единых критериев оценки эффективности использования выбранных вычислительных средств (БВМ, БЦВМ, БЦВС) и средств информационного обмена (СИО, ЛВС);

- качественное изменение элементной базы (появление сверхбольших и сверхбыстродействующих интегральных схем) и увеличение цены отказа вычислительных средств и средств информационного обмена, что требует применения системного подхода к построению СИО и ЛВС.

За последнее десятилетие коренным образом изменился облик вычислительных средств информационного обмена на борту подвижного объекта. Вместо разрозненных бортовых приборов на борту ПОб стали применяться высокоинтегрированные вычислительные модули, машины и системы, объединенные сложными интерфейсами, системами информационного обмена и локальными вычислительными сетями.

Интерфейс - это совокупность линий, шин, сигналов, протоколов взаимодействия (алгоритмов их реализации) электронных схем, предназначенных для организации информационного обмена между территориально распределенными абонентами.

Система информационного обмена - это комплекс аппаратно-программных средств, работающих под управлением единой операционной системы и ориентированных на реализацию функций обмена информацией (ввода и вывода информации) между территориально распределенными абонентами.

Локальная вычислительная сеть - совокупность вычислительных средств: бортовых вычислительных модулей, бортовых цифровых вычислительных машин, средств информационного обмена: интерфейсов с различными протоколами обмена и различной пропускной способностью, обеспечивающих реализацию информационного трафика на борту подвижного

объекта в реальном масштабе времени под управлением единой операционной системы.

К современным и перспективным интерфейсам, СИО и ЛВС предъявляются следующие требования:

- минимизация количества физических линий связи при обеспечении полной связности абонентов информационного обмена;
- высокая пропускная способность при обеспечении требуемого уровня надежности и достоверности передачи информации;
- минимальный коэффициент затрат при перераспределении потоков информации в случае модификаций и модернизаций бортового оборудования;
- высокий уровень унификации, применяемых для построения СИО и ЛВС интерфейсов.

Современные СИО и ЛВС позволяют реализовать алгоритмы информационного обмена чрезвычайно широкого спектра и эти возможности в последнее время возросли благодаря внедрению сверхбольших и сверхбыстродействующих интегральных схем.

5.1.2. Анализ СИО и ЛВС как больших систем.

Исходя из состава функциональных элементов, их сложности и взаимозависимости, СИО и ЛВС могут быть отнесены к классу больших систем.

Большим системам вообще, а СИО и ЛВС, в частности, присущи общие черты, среди которых можно выделить следующие:

- целенаправленность и управляемость системы, т.е. наличие у всех её элементов общей цели и общего назначения;
- системный характер реализуемых алгоритмов обмена, требующий совместной обработки информации от различных источников;
- сложная иерархическая архитектурная организация, предусматривающая сочетание централизованного управления с автономностью функциональных подсистем;
- наличие различных уровней переработки информации, самоорганизации и адаптации;
- целостность и сложность поведения, сложные связи между переменными, сложные петли обратной связи;
- большое количество входящих в систему функциональных подсистем, входов и выходов, разнообразие выполняемых функций;
- наличие информационных связей между функциональными элементами, а также внешних связей с другими функциональными подсистемами;
- наличие широкого спектра дестабилизирующих воздействий;
- высокая степень автоматизации процессов проектирования.

Эти особенности наложили определенный отпечаток на формы постановки и методы решения задач анализа и синтеза СИО и ЛВС. Задача анализа состоит в определении необходимых вероятностных характеристик, ока-

зывающих существенное влияние на работу СИО и ЛВС. Для решения задач анализа используются математические методы статистических оценок, специальные методы информационного, операционного, точностного и системного анализа алгоритмов.

Задачи синтеза: разработка концепций построения СИО и ЛВС, выбора оптимальной архитектурной организации СИО и ЛВС. Для решения задач синтеза используются более современные методы алгоритмического, информационного и надежностного синтеза по сравнению с методами, применяемыми при исследовании СИО и ЛВС на основе электронно-технологических характеристик, не отражающих особенностей, связанных с выработкой и принятием решений.

Эти особенности привели к необходимости анализа сложных систем не только в нормальных условиях эксплуатации, но и в, так называемых, аномальных или экстремальных условиях, которые возникают в результате выхода за допустимые пределы величин внешних воздействий.

Таким образом, актуальной стала проблема синтеза СИО и ЛВС, способных корректно реализовывать заданные алгоритмы информационного обмена в условиях возникновения неисправностей.

5.1.3. СИО и ЛВС, реализующие заданный класс алгоритмов обмена в условиях возникновения неисправностей.

Для удобства дальнейшего описания введем общее понятие "Информационная система" для системы информационного обмена и локальной вычислительной сети. Информационная система, реализующая заданный класс алгоритмов обмена в условиях возникновения неисправностей, должна обладать тремя основными взаимодействующими, определяющими и дополняющими друг друга свойствами: информационно-вычислительным потенциалом, надежностью и эксплуатационным совершенством.

Как известно, и теоретически и практически невозможно построить информационную систему, в которой не наблюдалось бы отказов. Важно проанализировать характер и влияние этих отказов на информационно-вычислительный процесс и разработать концепции, позволяющие минимизировать степень эффективности отказов, т.е. высокоэффективную систему. Эффективность является интегральной мерой информационно-вычислительных возможностей системы, уровня ее надежности и эксплуатационного совершенства. Эти свойства взаимосвязаны и взаимообусловлены, поэтому они не могут анализироваться обособленно друг от друга. При анализе альтернативных вариантов информационной системы предпочтение будет отдано той системе, у которой при прочих равных показателях, стоимость жизненного цикла будет минимальной.

Жизненный цикл - это общая календарная продолжительность существования системы, включая стадии: разработки, серийного производства и эксплуатации. Жизненный цикл определен временным интервалом решения о создании системы до наступления предельного состояния, т.е. состояния

при котором дальнейшая эксплуатация системы должна быть прекращена из-за неустранимых нарушений требований нормативно-технической документации (НТД).

В информационной системе при реализации заданного класса алгоритмов обмена будут наблюдаться неисправности, тем не менее, за счет внутренних ресурсов системы и организации ее функционирования система должна выполнить поставленную задачу. Речь в данном случае может идти только о снижении уровня эффективности реализации заданного класса алгоритмов обмена, которое характеризуется переходом системы из одного вида технического состояния в другой.

Информационная система (ИС) может быть в следующих видах технического состояния: исправном и неисправном, работоспособном и неработоспособном, правильного и неправильного функционирования.

ИС исправна, если в ней нет ни одной неисправности и ни одного несоответствия требованиям НТД, в противном случае ИС находится в неисправном состоянии.

ИС работоспособна, если значение всех ее параметров, характеризующих способность выполнять заданные функции, соответствуют требованиям НТД, в противном случае система не работоспособна.

ИС правильно функционирует, если в текущий момент времени способна выполнить заданные функции, в противном случае она находится в состоянии неправильного функционирования.

Исправное состояние включает работоспособное состояние правильного функционирования, т.е. исправная ИС всегда работоспособна и функционирует правильно, хотя правильно функционирующая система в данный момент времени может быть неработоспособна, а значит и неисправна.

Переход ИС из неисправного состояния в работоспособное, а из работоспособного - в состояние правильного функционирования представляет собой процесс постепенной деградации параметров, в результате которого технические возможности системы ухудшаются по мере отказа ее элементов, однако, сама система продолжает реализацию заданной функций. Таким образом, при разработке ИС в ее архитектуру вводят элементы, обеспечивающие возможность работы системы в режиме постепенной деградации параметров. ИС, характеризуемая постепенной деградацией параметров, способна выполнить поставленные задачи при возникновении отказов. Создание ИС, способной деградировать постепенно, и является основным методом построения высокоэффективной информационной системы.

Глава 6. Глобальные информационные и телекоммуникационные сети

6.1 Согласование протоколов в составных сетях

Развитие мирового информационного пространства, необходимость

включения локальных и территориальных сетей различного масштаба в глобальные телекоммуникации поставило задачу организации межсетевого взаимодействия в гетерогенной среде. Источником гетерогенности является исторически сложившееся использование различных протокольных стеков, необходимость интеграции ранее разработанных и находящихся в эксплуатации сетевых технологий с современными решениями.

Согласование различных протокольных стеков производится следующими методами: трансляцией; мультиплексированием; инкапсуляцией (туннелированием).

Трансляцию осуществляет специально выделенное устройство, которое с помощью аппаратных или программных средств выполняет преобразования сообщений из формата одной сети в формат другой. Преимущества этого метода заключаются в его простоте, а недостатки - в плохой масштабируемости.

При мультиплексировании разнородные сети соединяются через компьютер, операционная система которого содержит все необходимые протокольные стеки и программное обеспечение мультиплексора протоколов. Здесь запросы обрабатываются быстрее, чем при трансляции, но достигается это за счет дополнительных ресурсов мультиплексирующего компьютера.

Инкапсуляция решает более частную задачу - объединение однородных сетей через промежуточную, транзитную систему с иной транспортной технологией. В этом случае транзитная сеть содержит пограничные маршрутизаторы, выполняющие функции упаковки и извлечения пакетов транспортного протокола объединяемых сетей в пакеты транспортного протокола транзитной сети. При этом методе возможность взаимодействия узлов объединяемых сетей с узлами, входящими в транзитную сеть. Характерными примерами являются сопряжение территориальных сетей X.25 или передача пакетов IPX через сеть TCP/IP.

Средства согласования протоколов существенным образом зависят от уровня, на котором оно выполняется.

На физическом, самом нижнем, уровне задача сводится к согласованию физического, а иногда и логического кодирования, что выполняется концентраторами с соответствующими портами.

На следующем, канальном уровне, выполняются операции изменения формата кадра, вычисления нового значения контрольной суммы, при необходимости изменение порядка следования бит в байте. Адресная информация просто переносится из одного кадра в другой, но максимальный размер кадров у соединяемых сетей должен быть одинаковым. Эти процедуры выполняют транслирующие коммутаторы или мосты.

Наиболее сложной задачей является согласование сетей на сетевом уровне. На предыдущих уровнях соединялись сети или их фрагменты, построенные на основе одной и близких базовых сетевых технологий. Однако составная сеть может содержать части, в которых используются разные сетевые протоколы, а следовательно, разная система адресации и форматы адресного пространства. Кроме этого, могут быть и другие отличия. Это ус-

ложняет трансляцию и требует применения специализированных программных или аппаратных платформ, называемых шлюзами. В их качестве могут выступать многопротокольные маршрутизаторы с возможностью мультиплексирования протоколов сетевого уровня.

6.2 Цифровые сети с интеграцией служб

Принципы и технологии открытых систем нашли свое выражение в развитии интеграционных процессов сетей телефонии и передачи данных как составных частей современных телекоммуникаций. На первом этапе их базой послужило развертывание полностью цифровых информационных потоков без радикальной перестройки первичных сетей и концепция многоцелевых интерфейсов пользователя. В результате была разработана и стандартизована технология ISDN, получившая в отечественной литературе название узкополосной цифровой сети с интеграцией служб (У-ЦСИС) [.....].

Заложенные в технологии ISDN возможности не позволяют полностью описать их в терминах эталонной модели взаимодействия открытых систем OSI. В связи с этим архитектура ISDN предусматривает использование в качестве основообразующих модель плоскостей, а не уровней. К ним относятся:

- управляющая плоскость (С-плоскость), ответственная за вызовы и обусловленными ими соединениями, поддержку системы сигнализации,
- плоскость администрирования (М-плоскость), выполняющая как управление различными наборами протоколов, так и административное управление всей системой;
- транспортная плоскость (Т-плоскость), составляющие ее протоколы описывают характер физического взаимодействия,
- пользовательская плоскость (U-плоскость), генерирующая собственно информационный поток. Все вышеперечисленные плоскости обязаны ее обеспечивать в полном объеме.

Функционально плоскости ISDN выполняют роль физического, канального и сетевого уровней модели OSI. Выполнение функций более высоких уровней выходит за рамки ISDN и возлагается на взаимодействующие оконечные системы.

В стандартах ISDN определены несколько типов каналов, отличающихся скоростью передачи и типом передаваемой информации. Приведем их краткую характеристику:

- В-канал является основным пользовательским каналом для передачи речевых сигналов и данных. Для него определена скорость в 64 Кб/с, для соединений используется метод коммутации каналов,
- D-канал служит для сигнализации и сокращения времени установления соединения, работая на скорости 16 или 64 Кб/с и используя метод коммутации каналов или пакетов;

- Н-канал, предназначены для обслуживания приложений, требующих большей пропускной способности, чем обеспечивает В-канал.

Определены два основных интерфейса доступа: для малых компаний или домашних приложений и для крупных организаций с развитыми сетевыми потребностями.

В первом случае используется интерфейс передачи данных с номинальной скоростью (BRJ-интерфейс), образованный двумя В-каналами и одним D-каналом (2В+D) с общей скоростью 144 Кб/с, а во-втором - интерфейс основного уровня (РЫ-интерфейс), образованный 23 или 30 В-каналами и D-каналом на скорости 64 Кб/с (23В+D), что обеспечивает общую скорость, соответствующую линиям T1 или E1. При этом интерфейс ВY поддерживает соединения «точка-точка» или «точка-многоточка», а интерфейс PRJ только «точка-точка».

Функциональные устройства ISDN распределены по следующим группам:

1. Оконечная станция типа 1 (NT1), реализующая функции физического уровня;
2. Оконечная станция типа 2 (NT2), способная реализовывать функции физического, канального и сетевого уровня;
3. Терминальное оборудование типа 1 (TE1), к которому относится любое ISDN-устройство у конечного пользователя,
4. Терминальное оборудование типа 2 (TE2), сюда относятся устройства, не входящие в предыдущую группу, т.е. неспособные непосредственно работать в сетях ISDN,
5. Терминальный адаптер (ТА), использование которого позволяет оборудованию из группы TE2 взаимодействовать с сетью ISDN.

Взаимосвязь между различными функциональными устройствами определяется с помощью так называемых опорных (эталонных) точек R, S, T, U, описывающих типы соединений и являющихся, по сути, соответствующими интерфейсами:

- точка R описывает интерфейс между оборудованием из группы TE2 и терминальным адаптером ТА;
- точка S описывает интерфейс между оборудованием из групп TE1 или ТА и оконечными станциями NT1 или NT2,
- точка T описывает интерфейс между оборудованием из групп NT2 и NT1;
- точка U описывает интерфейс между оборудованием из группы NT1 и телефонной сетью.

Необходимо отметить, что имеются и другие опорные точки, описывающие взаимодействие ISDN с сетями других типов.

Как отмечалось ранее, ISDN реализует три нижних уровня модели OSI. Физический уровень ISDN определен рекомендациями I430, I431 и выполняет следующие основные функции:

- кодирование данных;
- дуплексная передача по В и D каналам,
- мультиплексирование соединений типа BRI или PRI;
- установление и разъединение виртуальных каналов;
- идентификацию терминального оборудования.

На канальном уровне основным протоколом является протокол доступа к D-каналу, определенный в рекомендациях I440 и I441. Он организует бит-ориентированную передачу в кадровой структуре, обеспечивая вышележащий уровень сервисом передачи информации, как с подтверждением, так и без него. Отметим, что на этом уровне имеются также средства, обеспечивающие взаимодействие с пакетными сетями X.25.

Сетевой уровень определен рекомендациями 1450, 1451 и определяет процедуры управления вызовами (соединениями с коммутацией, как пакетов, так и каналов) по каналу D и передачу информации по ним. На этом уровне также обеспечена поддержка взаимодействия с сетями X.25.

С развитием технологий Frame Relay на базе волоконно-оптических линий связи появилась возможность существенного повышения скорости передачи за счет отказа от протоколов X.25 с их развитыми средствами контроля ошибок и обусловленными этим накладными расходами. Соответствующие стандарты ISDN предусматривают такую возможность.

6.3. Технология АТМ, широкополосные цифровые сети с интеграцией служб

Растущая потребность в ресурсоёмких сетевых приложениях, таких как видеоконференции, высококачественное видео и аудио, мультимедиа среды обуславливает необходимость повышения скоростей передачи данных. С другой стороны назрела необходимость разработки новой транспортной подсистемы, которая единым образом передавала бы все виды информации, обеспечивая при этом []:

- гибкость и адаптацию подсистемы к изменению уровня требований пользователя к объему, скорости и качеству доставки информации;
- эффективное использование транспортных ресурсов за счет выделения пользователю в каждый момент времени только той части пропускной способности сети, который ему необходим,
- возможность создания единой универсальной сети (без подразделения на локальные или глобальные) для всех существующих и будущих служб,
- снижение общих затрат на проектирование, строительство и обслуживание телекоммуникационной подсистемы.

Очевидно, что эти требования вполне отвечают и проистекают из концепции и самой логики построения открытых систем.

Технической основой для реализации такой подсистемы служат воло-

конно-оптические линии связи, достижения в технологии сверхбольших интегральных схем, а также разработка нового режима переноса информации, получившего название АТМ или асинхронный режим переноса в отечественной транскрипции.

Сущность этого режима заключается в асинхронном мультиплексировании потоков ячеек от разных пользователей в едином цифровом тракте. Принципиальным является фиксированная длина каждой ячейки, равная 53 байтам, из которых 5 байт занимает заголовок, а 48 байт отведены под полезную информацию. Малая длина заголовка и ограничение возложенных на него функций позволяет достичь очень высоких скоростей в коммутаторах АТМ. В сочетании с малой и постоянной длиной информационного поля это обеспечивает необходимые условия для служб функционирующих в режиме реального времени.

Таким образом, асинхронный режим переноса стал основой для создания широкополосных цифровых сетей с интеграцией служб Ш-ЦСИС (В-ISDN).

Архитектура Ш-ЦСИС является расширением модели У-ЦСИС и включает в себя следующие плоскости (рисунок 6.1):

- управляющую плоскость (С-плоскость), имеющую уровневую структуру и отвечающую за протоколы установления, контроля и разъединения соединений;
- плоскость администрирования (М-плоскость), выполняющую управление и координацию между всеми плоскостями и уровнями,
- пользовательская плоскость (U-плоскость), имеющая уровневую структуру и обеспечивающая передачу всех видов информации с механизмами контроля и управления потоком, защиты от ошибок и т.д. Предыдущие плоскости обязаны обеспечить ее поддержку.

Самым нижним уровнем протокольного стека является физический уровень. Он состоит из двух подуровней:

- зависящего от физической среды, на котором обеспечивается передача битового потока через физическую среду, синхронизация, кодирование и т.д.,

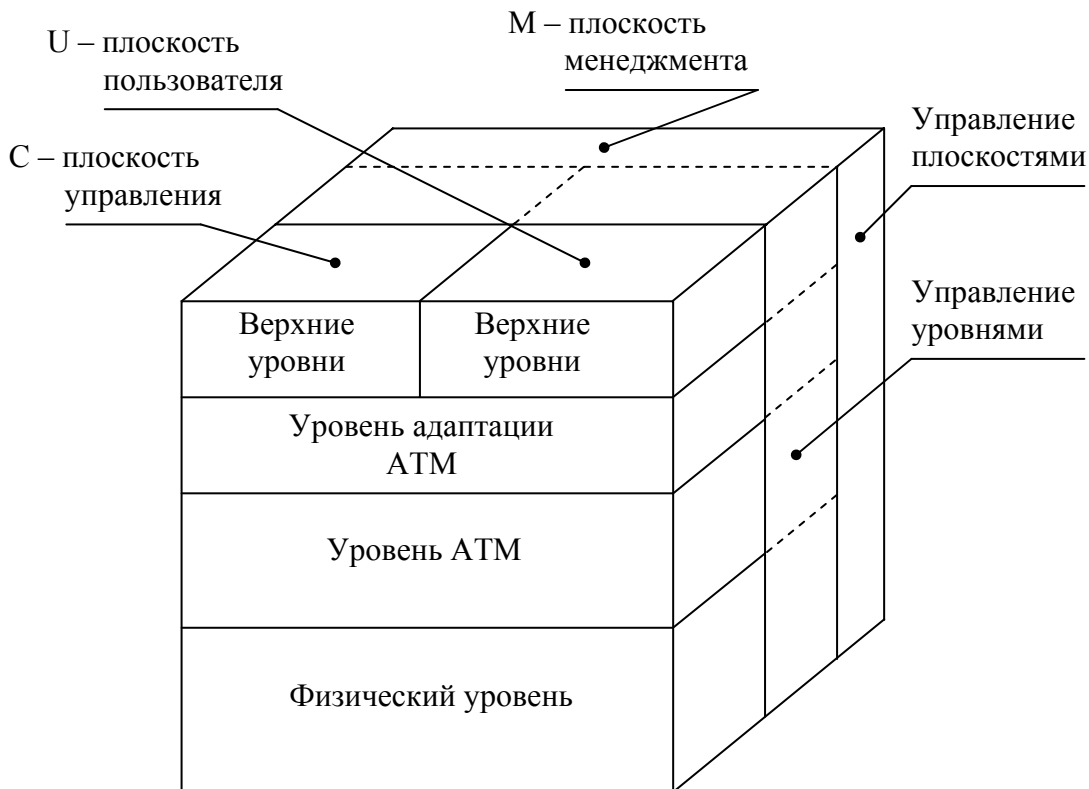


Рис.6.1. Плоскости архитектуры Ш-ЦСИС.

- подуровня конвергенции передачи, который помещает ячейки в кадры битового потока и выделяет ячейки их них.

Следующий уровень получил название АТМ. На нем выполняются операции мультиплексирования и демупльтиплексирования ячеек, генерация или удаление заголовков ячеек, направление потока ячеек по виртуальным каналам и виртуальным путям. Их идентификаторы располагаются в заголовке ячеек.

Наконец, верхний уровень - уровень адаптации АТМ. Он состоит из двух подуровней и выполняет две основных функции: сегментацию и сборку пакетов для их размещения в ячейках и восстановления из них; обеспечение сетевых сервисов приложениям и протоколам более высоких уровней.

Стандартизованы два типа интерфейса «пользователь-сеть» на скоростях 155 Мб/с и 622 Мб/с, соответственно. Они предусматривают использование кадров синхронной цифровой иерархии или систем передачи ячеистого типа. Также возможно размещение ячеек в кадрах систем плезиохронной цифровой иерархии. Эти операции, а также контроль, обнаружение и исправление ошибок в заголовках ячеек, их разграничение и согласование скорости выполняются на физическом уровне.

На уровне АТМ базовым элементом выступает ячейка. В интерфейсе «пользователь-сеть» ее заголовок содержит следующие поля:

- общее управление потоком (4 бита) служит для контроля нагрузки;
- идентификатор виртуального пути (8 бит);

- идентификатор виртуального канала (16 бит) - вместе с предыдущим полем служит для маршрутизации ячейки;
- тип полезной нагрузки (3 бита) - несет информацию о типе ячеек;
- поле приоритета (1 бит) - указывает приоритет ячейки, т.е. возможность ее отбрасывания в случае перегрузок.

Заголовок ячейки в сетевом интерфейсе имеет одно отличие. В нем отсутствует поле «общее управление потоком», а поле «идентификатор виртуального пути» занимает 12 бит, что расширяет возможности маршрутизации в сети.

На уровне адаптации АТМ и образующих его подуровнях выделено 4 типа: тип1, тип2, тип 3/4 и тип 5. Это обусловлено тем, что различные службы требуют различного обслуживания, классы которого зависят от: постоянной или переменной битовой скорости; режима с соединением или без него; наличия или отсутствия требований к синхронности тактовых частот источника и приемника.

Тем самым обеспечивается качество обслуживания, необходимое пользователю.

Технологии АТМ и построенные на их основе сетевые системы находятся в стадии интенсивного внедрения. Можно утверждать, что именно с ними будет связано достижение нового уровня в развитии телематики.

Глава 7. Основы концепции открытых систем

Переход к информационному обществу, создание единого информационного пространства, требуют создания информационной инфраструктуры (ИИ) (рисунок.7.1). Информационная инфраструктура любого уровня (глобальная, национальная, отраслевая и т.д.) состоит из информационных ресурсов, вычислительных ресурсов и средств телекоммуникаций. Информационные и вычислительные ресурсы формируются на основе аппаратно-программных платформ различных классов (персональные ЭВМ, рабочие станции, мини-ЭВМ, мейнфреймы, супер-ЭВМ), изготовленных различными производителями, работающими под управлением различных операционных систем, т.е. осуществляется переход к гетерогенным системам. В системах телекоммуникаций используются различные протоколы и форматы данных, и неизбежно встает вопрос о переносе прикладных программ с одной программно-аппаратной платформы на другую, о доступе к различным базам данных, о взаимосвязи удаленных систем. При этом следует помнить, что любая система рано или поздно требует модернизации, расширения, и эта модернизация должна произойти с минимальными потерями, в том числе с минимальными затратами на переобучение персонала. Таким образом, возникает вопрос о создании и применении технологии, решающей эти проблемы. Такой технологией выступает технология открытых систем (ТОС) [1,2].

Существо технологии открытых систем состоит в формировании среды, включающей программное обеспечение, аппаратные средства, службы связи, интерфейсы, форматы данных и протоколы, обеспечивающей переносимость, взаимосвязь и масштабируемость приложений и данных [1]. Совокупность указанных качеств достигается за счет использования развивающихся, общедоступных и общепризнанных стандартов на продукты информационных технологий, составляющих среду открытой системы.



Рисунок 7.1. Соотношение понятий: информационное общество, информационное пространство, информационная инфраструктура, технология открытых систем.

Примечание:

Здесь сразу требуется отметить, что понятие "открытая система" не означает, что она является незащищенной в смысле доступа к содержащейся в ней информации. Сохранение конфиденциальности информации, представляющей собой государственную, коммерческую, военную и личную тайну, является обязательным условием любой информационной системы.

Возвращаясь к рис.7.1, отметим, что чем выше по вертикальной шкале, тем больше доля социальных, политических и экономических аспектов, а чем ниже, тем больше доля технических аспектов. Соответственно, меняется характер законодательно-нормативной базы. Идя сверху вниз, это - сначала международные соглашения, потом национальные законы, региональные законы, постановления правительства, различные подзаконные акты и, наконец, внизу - стандарты и руководства по их применению.

7.1. Области применения ТОС

ТОС применяется при построении систем всех классов и назначений, входящих как компоненты в информационную инфраструктуру:

- вычислительных;
- баз данных;
- телекоммуникационных;
- систем реального времени ;
- встроенных микропроцессорных систем.

Разница заключается лишь в том, что для систем различных классов используются различные группы стандартов.

Это и естественно, потому что к общей концепции открытых систем специалисты, работающие в разных областях, шли своим путем. Для специалистов, работающих в области вычислительных систем, история развития концепции открытых систем возникла с того момента, когда возникла проблема переносимости прикладных программ и данных между компьютерами с различной архитектурой. Частичное решение проблемы переносимости обеспечили ранние стандарты языков, таких как Фортран и Кобол. Следующим важным шагом послужило появление операционной системы UNIX, которая устанавливается на большое количество аппаратных платформ и на которой идет множество приложений.

В области телекоммуникаций уже достаточно давно встал вопрос о согласовании используемых протоколов.

В системах реального времени также уже достаточно давно встал вопрос о разработке стандартного интерфейса между компьютером и объектом управления. Классическим примером можно считать реализацию этого интерфейса в международном стандарте КАМАК.

В развитии и применении ТОС заинтересованы все участники процесса информатизации :

- пользователи;

- проектировщики систем и системные интеграторы;
- производители технических и программных средств вычислительной техники и средств телекоммуникаций;
- разработчики стандартов.

7.2. Эталонная модель среды открытых систем

Число продуктов информационных технологий, как мы все знаем, составляет многие и многие тысячи, соответственно, велико и число стандартов. Для облегчения взаимопонимания между указанными выше группами специалистов целесообразно использовать какую-то единую модель среды открытых систем. Такой моделью служит т.н. эталонная модель среды открытых систем (Open System Environment Reference Model - OSE/RM) (см. рис 7.2.) [1].

Эта модель может модернизироваться в зависимости от класса системы. Например, для телекоммуникационных систем хорошо известна 7-уровневая модель взаимосвязи открытых систем ISO/IEC 7498 . Можно сказать и по другому, модель OSE/RM выросла как расширение модели взаимосвязи открытых систем с детализацией верхнего прикладного уровня.

Как видно из рис.7.2., в эталонной модели можно выделить следующие основные компоненты(горизонтальный разрез):

- приложение;
- платформу;
- внешнюю среду;
- интерфейс приложения с платформой;
- интерфейс платформы с внешней средой.

и следующие элементы. или функциональные области :

- службы операционной системы;
- службы интерфейса "человек-машина";
- служба управления данными;
- служба обмена данными;
- служба машинной графики;
- служба сетевого обеспечения;

Подробное описание функций, выполняемых этими службами можно найти в [2, 3, 4]

Службы операционной системы являются узловыми службами, необходимыми для функционирования платформы, и они обеспечивают интерфейс между приложением и платформой.

Службы интерфейса "человек-машина" определяют методы, с помощью которых человек может взаимодействовать с приложением. В зависимости от функциональных возможностей, требуемых пользователем, и от приложения эти интерфейсы могут включать в себя различные функции.

Службы управления данными. В большинстве систем центральным элементом является административное управление данными, которое

может быть определено независимо от процессов создания и использования этих данных, может обслуживаться неопределенным образом и коллективно использоваться многими процессами.

Служба обмена данными представляет собой специальную поддержку (включая форматы и семантику элементов данных) для обмена информацией между приложениями одной и той же или различных платформ.

Службы машинной графики обеспечивают функции, необходимые для создания и манипулирования выводимыми на экран изображениями. К этим службам относятся определение выводимых на экран элементов и управление ими, определение атрибутов изображения. Эти службы определены в спецификациях по описаниям многомерных графических объектов и изображений в независимой от устройства форме.

Сетевые службы обеспечивают функциональные возможности и механизмы поддержки распределенных приложений, требующих доступа к данным, а также поддержки взаимодействия приложений в неоднородной сетевой среде.

Теперь следует обратить внимание(рис. 7.2.) на то, что существуют функции, общие для всех вышеперечисленных элементов, и, таким образом, модель становится трехмерной.

К этим элементам относятся :

- службы поддержки разработки программного обеспечения;
- службы защиты информации;
- интернационализация;
- служба поддержки распределенной системы;

Службы поддержки разработки программного обеспечения позволяют создавать программное обеспечение, в соответствии с требованиями открытых систем. Эти службы включают стандартные языки программирования, инструментальные средства и функциональную среду программной инженерии.

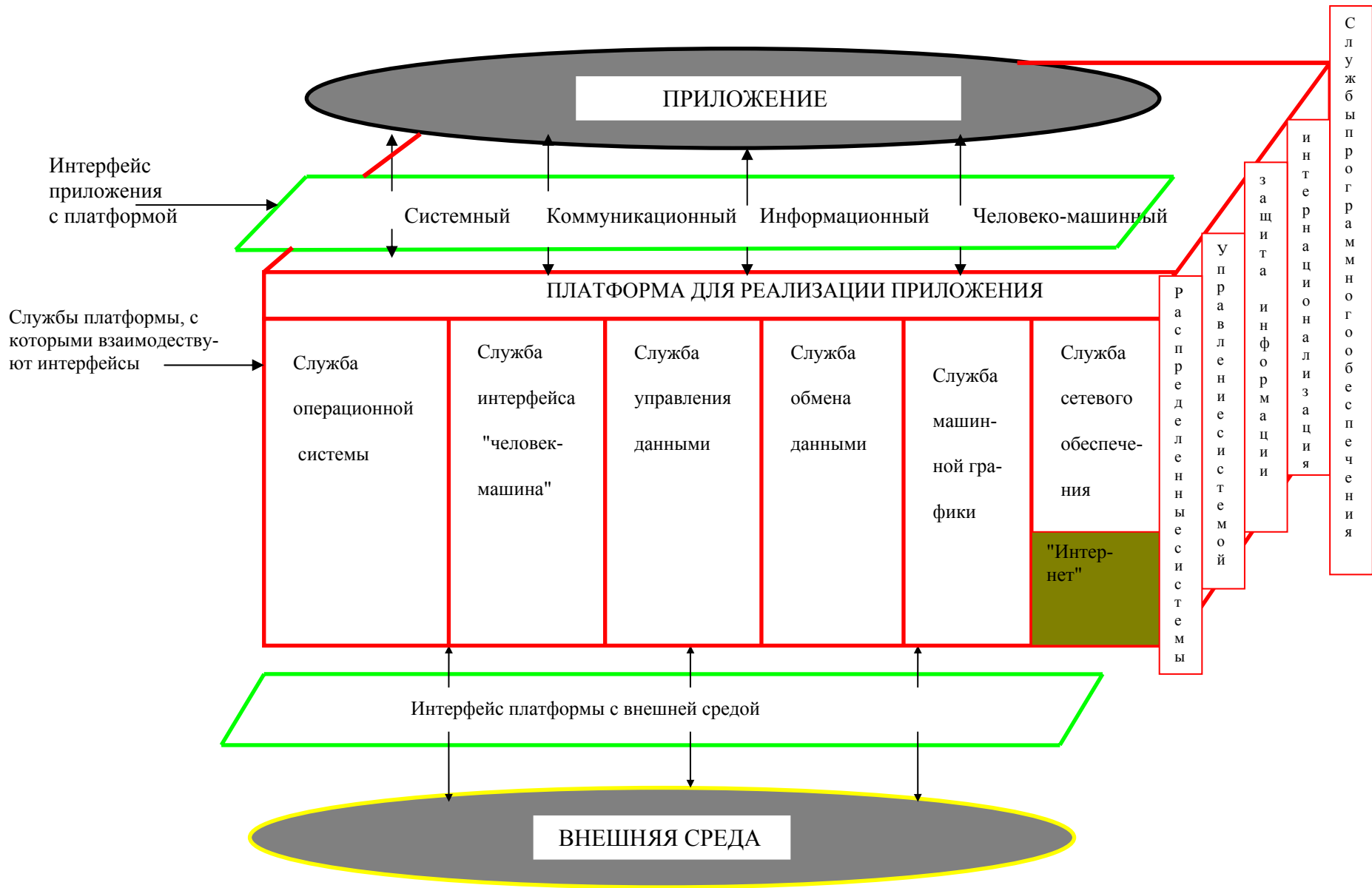


Рис. 7.2. Эталонная модель среды открытой системы OSE/RM

Службы защиты информации предназначены для обеспечения защищенного распространения и целостности информации и для защиты системы от несанкционированного доступа.

Задачи интернационализации состоят в обеспечении взаимопонимания специалистов различных стран. Эти задачи состоят не только в общности используемого языка, но в решении таких аспектов, как человеко-машинный интерфейс, учет специфики рынка различных стран и организации информационного обеспечения в различных странах.

Следует обратить внимание, что сеть Интернет, построенная на основе протоколов TCP/ IP, также является частью среды открытой системы как часть сетевых служб, входящих в одну из функциональных областей среды, и далеко не решает всех проблем открытых систем.

7.3. Понятие профиля

Таким образом, получается, что основу ТОС составляет стандартизация. Стандартизация зачастую представляется достаточно скучным предметом. Но ее необходимость заставляет относиться к ней крайне серьезно.

Здесь уместно напомнить историю создания в Америке Национального Бюро Стандартов (НБС) в прошлом веке. Решение о создании НБС было принято после того, как однажды пожарники, приехавшие тушить пожар, не смогли привернуть шланг к крану из-за отличия в резьбе. Ныне НБС преобразовано в Национальный институт стандартов и технологии (National Institute of Standards and Technology - NIST), который является одним из главных разработчиков в области открытых систем.

Уместно также провести сравнение с хождением к зубному врачу - если Вы отдаете себе отчет в том, что качество ваших зубов рано или поздно определит качество всей вашей жизни и ее длительность - следует относиться к своим зубам серьезно и регулярно ходить к врачу, хотя приятного мало, но необходимо. Также и стандартизация. По определению, любой стандарт предназначен для достижения оптимальной степени упорядоченности в определенной области и служат средством управления качеством продукции. Не хотите же Вы, чтобы создаваемая или используемая вами информационная система была неупорядоченной и низкого качества. Так что стандартизации следует придавать первостепенное значение. Считается, что высокий уровень стандартизации будет одним из отличительных признаков 21-го века. Это и понятно - идут интеграционные процессы во всех областях человеческой деятельности и без соблюдения стандартов эта интеграция затруднена. Но и

здесь есть своя романтика - она связана с понятием профиля, которое является узловым понятием ТОС.

Профиль, это - согласованный набор базовых стандартов, предназначенный для решения какой либо задачи или класса задач.

В соответствии с этим существует определенная классификация профилей (см. рис.7.3).

На верхнем уровне находятся международные стандартизованные профили (International Standardized Profiles –ISP), признанные комитетом ИСО. ISP имеют в международном сообществе такой же статус, что и международные базовые стандарты и направлены на широкую область применения. Например, ISP POSIX 1003.13 предназначен для описания работы систем реального времени, независимо от конкретной области применения.

Ниже следуют национальные профили, в соответствии с которыми должна строиться Национальная ИИ. Примерами национальных профилей могут служить Профиль Переносимости Приложений Правительства США (Application Portability Profile - APP) [5] и входящий в него Государственный профиль взаимосвязи открытых систем США (Government Open System Interconnection Profile - GOSIP) [6].

Далее следуют т.н. профили индустрии информационных технологий, которые ориентированы на области применения, общие для различных отраслей. К таким профилям можно отнести, например, профиль делопроизводства (Office Documentation Architecture -ODA), поскольку электронный обмен документацией в различных отраслях имеет одни и те же принципы. Можно говорить также о профиле электронных библиотек, независимо от их проблемной ориентации.

Профили следующего уровня - отраслевые профили, для каждой отрасли может и должен быть построен свой профиль, например профиль банковской деятельности, профиль военного назначения , профиль научных исследований и т.д.

Ниже следуют профили уровня предприятия и подразделения.

Все описанные профили не должны противоречить профилю более высокого уровня, т.е. вкладываться друг в друга по принципу "матрешки".

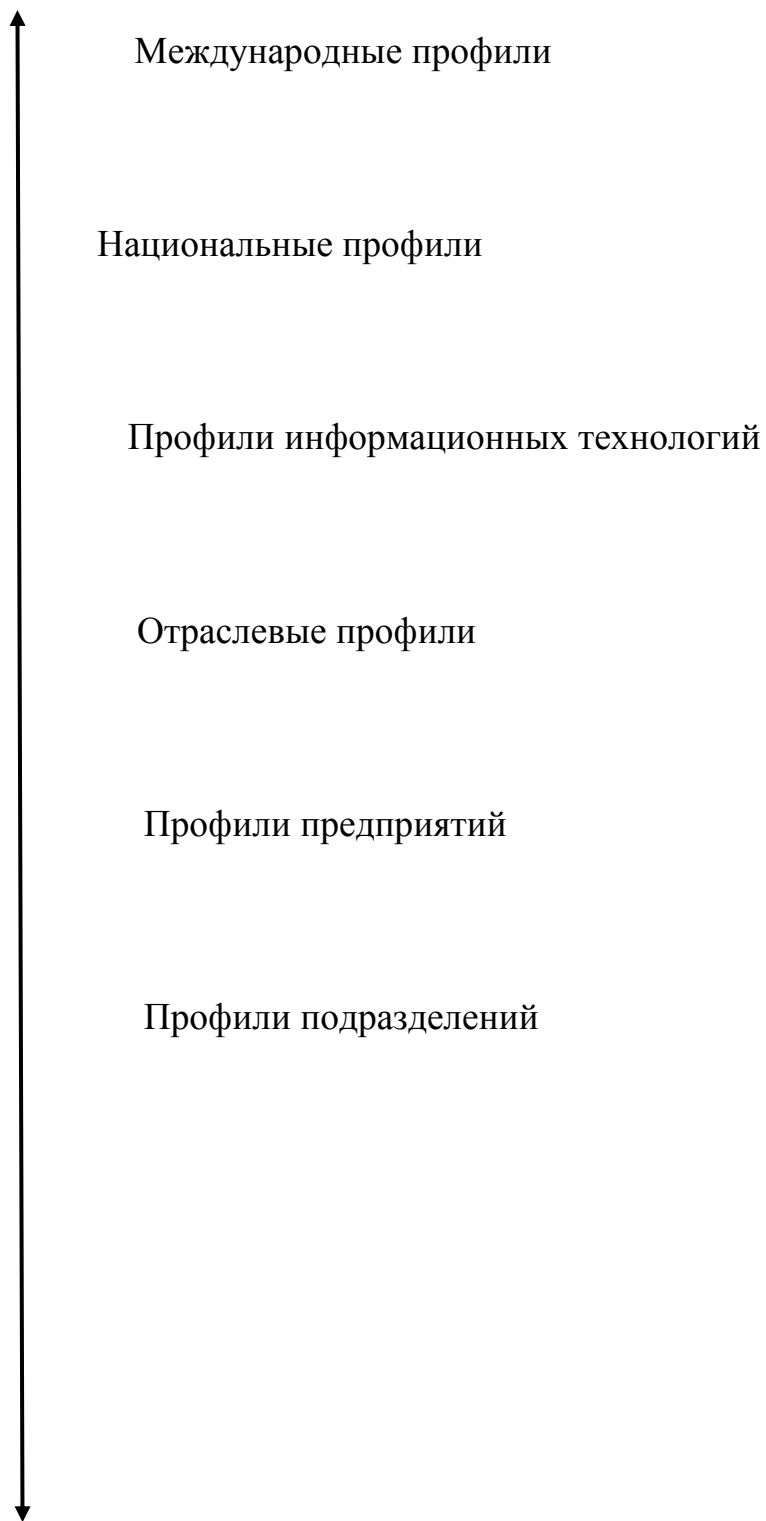


Рисунок 7.3 Классификация профилей

В целом же, можно сказать, что чем выше уровень профиля, тем шире область его применения, а чем ниже уровень, тем уже его функциональная направленность.

Общие правила построения профилей описаны в документе ISO/IEC TR-10000 [7].

7.4. Технологический цикл построения открытых систем

Как известно, традиционный жизненный цикл информационных систем регламентируется стандартом **ГОСТ 34.601-90** "Информационная технология. Комплект стандартов на автоматизированные системы. Стадии создания." -- Взамен ГОСТ 24.601-86, ГОСТ 24.602-86. - Утв . 1990. - Тем. код Е. Настоящий стандарт распространяется на автоматизированные системы (АС), используемые в различных видах деятельности (исследование, проектирование, управление и т. п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях. Стандарт устанавливает следующие стадии создания АС:

- 1) формирование требований к АС;
- 2) разработка концепции АС;
- 3) техническое задание;
- 4) эскизный проект;
- 5) технический проект;
- 6) рабочая документация;
- 7) ввод в действие;
- 8) сопровождение АС

В приложении к стандарту приведено содержание работ на каждой стадии.

Примечание В случае создания простых систем ряд стадий могут быть опущены.

Согласно этому ГОСТу средства обеспечения информационных систем состоят из компонентов :

- технического обеспечения;
- программного обеспечения;
- информационного обеспечения;
- организационно-правового обеспечения;

Требование к "открытости" систем, состоит, как следует из вышесказанного, в необходимости построения профиля - согласованного набора стандартов на все компоненты системы. Это означает необходимость введения еще одного вида обеспечения - нормативного обеспечения.

Технологический цикл построения открытых систем показан на рис. 7.4.

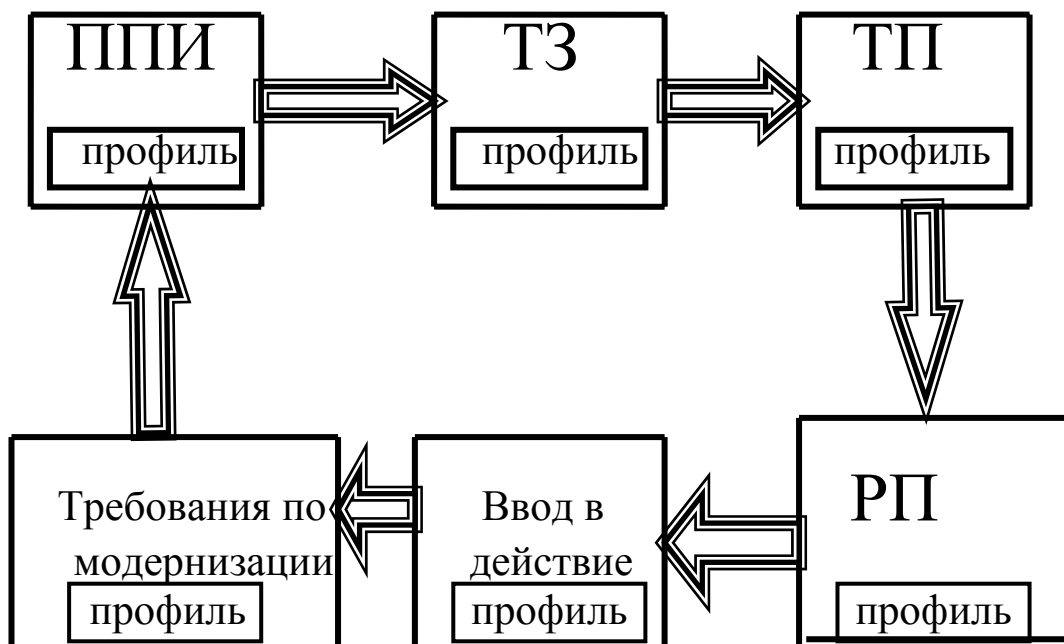


Рисунок. 7.4. Основные стадии создания открытых систем

ППИ - предпроектные исследования

ТЗ - техническое задание

ТП - технический проект

РП - рабочий проект

Его основное отличие от традиционного цикла заключается в построении профиля системы, зафиксированного еще на стадии предпроектных исследований. Состав профиля зависит от уровня системы и не должен противоречить профилю соответствующего уровня. На основе построенного профиля должны реализовываться все последующие стадии жизненного цикла системы.

В соответствии с мировой тенденцией, жизненный цикл современных информационных систем как и других видов продукции, для обеспечения качества и особенно для обеспечения конкурентоспособности на мировом рынке, должен соответствовать международным стандартам серии ИСО 9000 и состоять из 11 стадий (петля качества) (см. рис. 7.5.).

Подчеркнем еще раз роль профилей в реализации принципов открытых систем. Построение профиля на самых ранних стадиях создания информационной системы и следование ему на всех дальнейших стади-

ях жизненного цикла дает пользователю (заказчику) независимость от конкретного поставщика при создании и модернизации системы.

Разработчики приложений, в свою очередь, следуя профилю, обеспечивают возможность повторного применения разработанных приложений при смене платформ.

Поставщики средств вычислительной техники обеспечивают себе расширение рынка сбыта и при желании могут специализироваться в одном конкретном рынке сбыта и виде продукции.

Наконец, разработчики стандартов могут оценить отсутствие необходимых стандартов и сосредоточить внимание на их разработке.

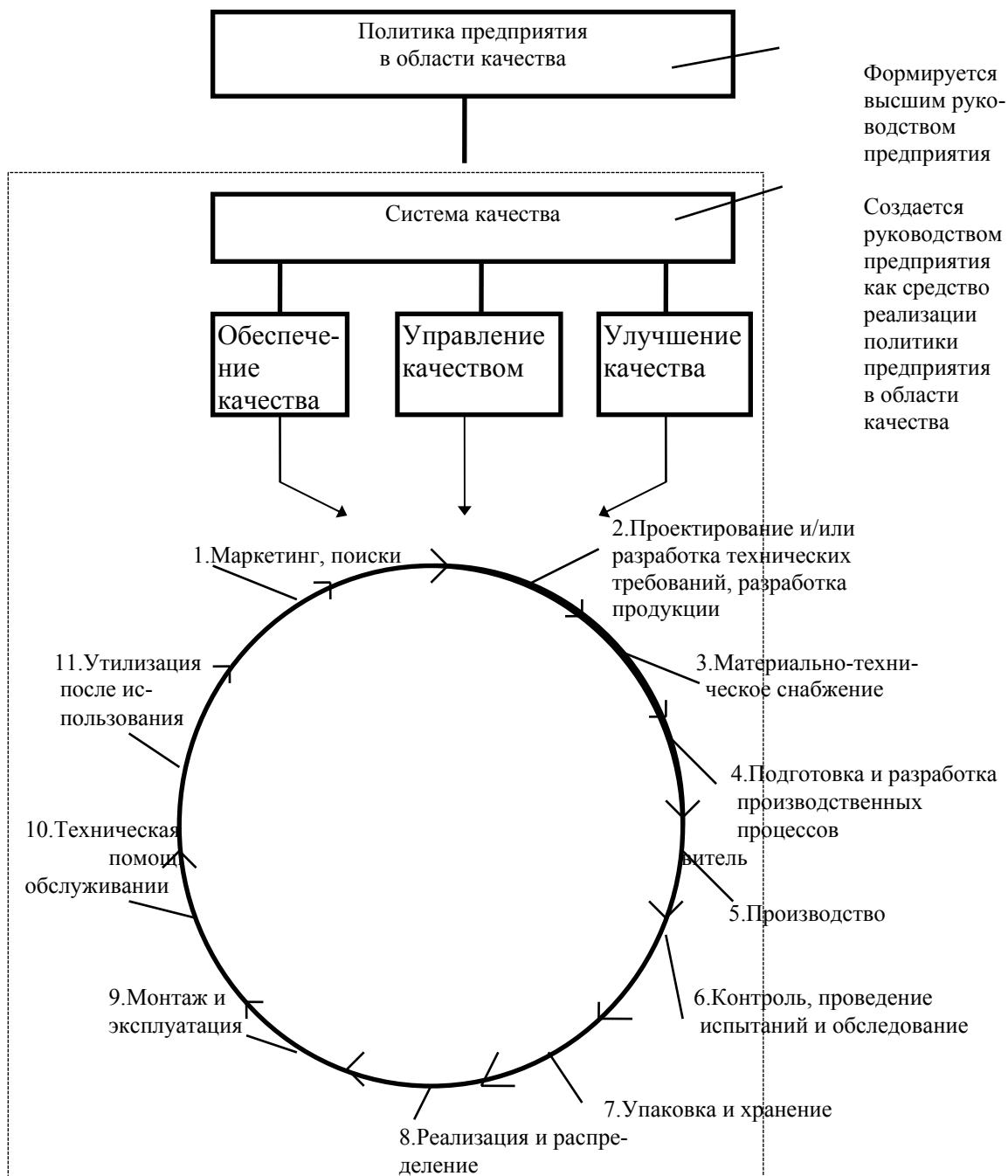


Рисунок. 7.5. Жизненный цикл системы - "петля качества"

7.5. Разработка профилей

Вопросами разработки стандартов и спецификаций в области информационных технологий занимаются во всем мире более 300 организаций. Эти организации можно разделить на три категории :

- аккредитованные организации по стандартизации;
- производители ;
- группы пользователей;

Внутри каждой из этих трех категорий, организации объединяются между собой, в том числе, в различные ассоциации и консорциумы, организации всех этих категорий участвуют в сложном и дорогостоящем процессе выработки стандартов [8] по принципам Workshop (рис. 7.6.). Workshop дословно переводится как мастерская, что очень точно передает смысл весьма точно регламентированного процесса, аналога в русском языке пока нет, и поэтому мы будем пользоваться этим термином). Следует подчеркнуть, что в условиях рыночной экономики стандарты играют совершенно другую роль, чем они играют при централизованно распределительной экономике, их живучесть и эффективность достигаются за счет согласительного, сбалансированного процесса их создания с участием пользователей, производителей и разработчиков.

К аккредитованным организациям в нашей стране относится Госстандарт.

Следует отметить, что механизм выработки профилей в целом такой же, как и при выработке базовых стандартов, создание профилей называется функциональной стандартизацией. Просто, ввиду необходимости согласования очень большого количества стандартов, этот процесс гораздо более трудоемкий и дорогостоящий. В этом смысле функциональная стандартизация представляет собой в настоящее время высшую форму стандартизации.

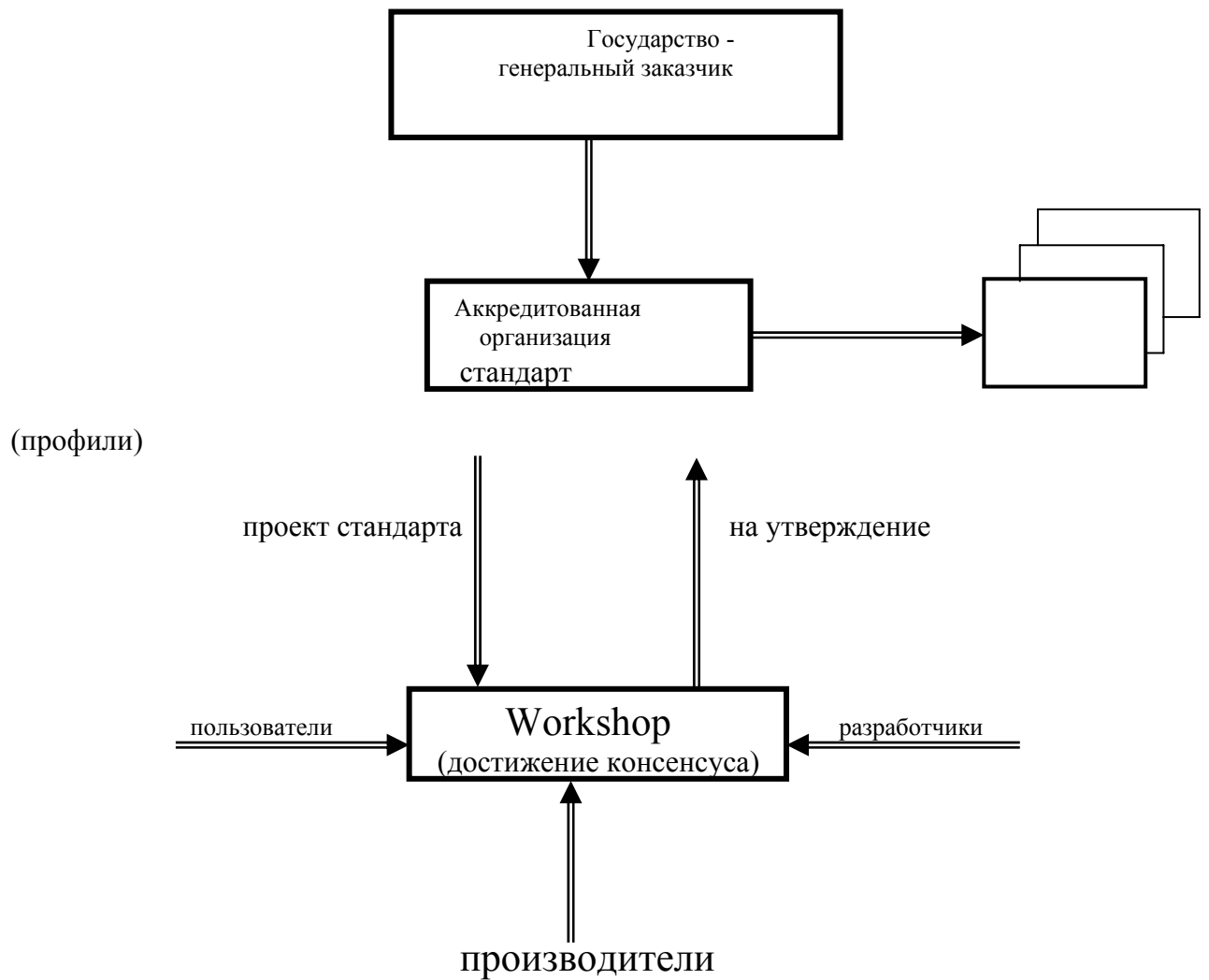


Рисунок. 7.6. Процесс принятия стандарта(профиля) методом достижения консенсуса

Выводы по главе 7.

1. При создании современных информационных систем всех классов и назначений, в том числе рассмотренных в предыдущих разделах данного пособия, необходимо придавать первостепенное значение вопросам стандартизации.

2. На сегодня процесс применения стандартов в области информационных технологий реализуется с помощью Технологии Открытых Систем, существо которой состоит в создании аппаратно-программной среды, обеспечивающей переносимость приложений, взаимодействие и масштабируемость систем.

3. Узловым моментом в Технологии Открытых Систем выступает построение и применение профиля (функционального стандарта) на основе которого осуществляются все стадии жизненного цикла систем.

4. Технология Открытых Систем обеспечивает значительную экономию за счет возможности повторного использования приложений отсутствия необходимости создания промежуточных согласующих устройств.

5. Можно сказать, что Технология Открытых Систем представляет собой основную технологию при построении Информационной Инфраструктуры любого уровня (глобальной, национальной, отраслевой, предприятия и т.д.).

Термины и определения :

Взаимодействие(interoperability) - способность двух и более систем обмениваться информацией и однозначно использовать информацию, полученную при обмене.

Взаимосвязь(interconnection) - способность двух и более систем обмениваться информацией без искажений.

Внешняя среда(external environment) - набор программно - аппаратных средств внешних по отношению к прикладной платформе которую они поддерживают

Гетерогенная система - система, использующая ресурсы различной архитектуры

Интерфейс - программно - аппаратный посредник между аппаратными или программными средствами

Масштабирование(scalability) - способность приспособлять функциональность в ряду архитектур с малыми и большими скоростями и разрядностью

Мейнфрейм(mainframe) - большая вычислительная машина

Нормативное обеспечение - стандарты, технические условия, технические требования, рубрикаторы и классификаторы

Открытая система - Система, содержащая исчерпывающую и последовательную совокупность международных стандартов в области информационных технологий и функциональных профилей стандартов, которая специфицирует интерфейсы, службы и поддерживающие форматы для достижения взаимодействия и переносимости приложений, данных и персонала.

Переносимость приложений [portability application software]- качество прикладной программы, характеризующее объем доработок программы в исходных кодах при постановке приложения на отличающихся платформах.

Приложение(application) - применение совокупности ресурсов, предоставляемых информационной системой и данных для нужд пользователя

Среда открытой системы - вычислительная среда, включающая программное обеспечение, аппаратные средства, службы связи, интерфейсы, форматы данных и протоколы, которая в основе имеет развивающиеся доступные и общепризнанные стандарты и обеспечивает значительную степень переносимости (portability), взаимосвязи (interoperability), и масштабирование (scalability) приложений и данных.

Список сокращений :

ИИ - Информационная Инфраструктура

НБС - Национальное Бюро стандартов

ТОС - Технология открытых систем

Литература

К главе 1:

- 1.1. Отраслевой стандарт высшей школы “Информационные технологии”.
- 1.2. Айламазян А.К., Стась Е.В. Информатика и теория развития. - М., Наука, 1989.
- 1.3. Гвардейцев М.И., Морозов В.П., Розенберг В.Л. Специальное математическое обеспечение управления. - М., "Советское радио", 1978.

К главе 2:

- 2.1. Айламазян А.К., Стась Е.В. Информатика и теория развития. - М., Наука, 1989.
- 2.2. Шеннон К. Работа по теории информатики и кибернетики. - М., Из-во иностранной литературы, 1963.
- 2.3. Гвардейцев М.И., Морозов В.П., Розенберг В.Л. Специальное математическое обеспечение управления. - М., "Советское радио", 1978.
- 2.4. Блохнин С.М. Шина ISA персонального компьютера IBM PC/AT. –М.: ПК "Сплайн", 1992 – 76с.
- 2.5. Петрухин В.С., Степченков Ю.А., Филин А.В. Персональные ЭВМ на основе архитектуры Intel 80386. Книга 2. – Обнинск: ИнвесКо, 1993 – гл. 4.
- 2.6. Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC. /Под ред. У.Томпкинса, Дж. Уэбстера; пер. с англ. – М.: Мир, 1992 – 512с.: ил.
- 2.7. Новиков Ю.В., Калашников О.А., Гуляев С.Э. Разработка устройств сопряжения для персонального компьютера типа IBM PC. Под общей ред. Ю.В. Новикова. Практ. Пособие. – М.: ЭКОМ, 1997 – 224с.
- 2.8. Приборно – модульные универсальные автоматизированные измерительные системы. Справочник. /В.А. Кузнецов, В.Н. Строителей, Е.Ю. Тимофеев и др.; Под ред. В.А. Кузнецова. – М.: Радио и связь, 1993. — 304с.: ил.

К главе 3:

1. Кузнецов С.Д. Четверть века операционной системы UNIX.- Открытые системы, 1995, вып.1.Сс14-17.
2. Филинов Е.Н. Выбор и разработка концептуальной модели среды открытых систем.- Открытые системы, 1995, вып.6.С.71-77.

3 Липаев В.В., Филинов Е.Н. Мобильность программ и данных в открытых информационных системах.- М.: Научная книга, 1997. – 368с.

К главе 6:

1. Щербо В.К., Козлов В. А. Функциональные стандарты в открытых системах. Ч.1. Концепция открытых систем. Справочное пособие. М.: Международный центр научной и технической информации, 1997.- 124с.
2. Бакланов И.Г. ISDN и FRAME RELAY: технология и практика измерений/Под ред. А.Б. Иванова. М.: ЭКО-ТРЕНДЗ, 1999.- 186с.
3. Халсалл Ф. Передача данных, сети компьютеров и взаимосвязь открытых систем: Пер, с англ.-М.: Радио и связь, 1995.- 408с.
4. Назаров А.Н., Симонов М.В. АТМ: технология высокоскоростных сетей. М.:Эко-Трендз, М.: 1999. - 251 с

К главе 7:

1. Ю.В. Гуляев, А.Я. Олейников Технология открытых систем - основное направление информационных технологий/Информационные технологии и вычислительные системы, М.:ИТиВС №3 1997, стр.4-14
- 2.<http://opensys.ire.ras.ru>, Проекты, кн.2,4
- 3.И.В. Байкова и др. Распределенные информационно - вычислительные системы, Вып.1, Локальная сеть ВЦ РАН, Вычислительный центр РАН, М.: 1995, 112 с.
- 4.James Isaak et al. Open Systems Handbook. IEEE Standards Press, 1994, p.p.197
- 5.Application Portability Profile. v.3. NIST SPEC PUB 500-230 1996, p.p.220
- 6.Government Open System Interconnection. FIPS 146-3. NIST. USA
- 7.Carl F. Cargill. Information Technology Standardization Theory, Process and Organisations. Bedford, MA: Digital Press, 1991
- 8.Workshop.