



**Современный
Гуманитарный
Университет**

Дистанционное образование

Рабочий учебник

Фамилия, имя, отчество _____

Факультет _____

Номер контракта _____

ОСНОВЫ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

ЮНИТА 3

**ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННЫХ
ИНФОРМАЦИОННЫХ СИСТЕМ**

МОСКВА 1999

Разработано Кузубовым В.Н.

Рекомендовано Министерством общего
и профессионального образования
Российской Федерации в качестве
учебного пособия для студентов
высших учебных заведений

КУРС: ОСНОВЫ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Юнита 1. Теоретические основы автоматических информационных систем.

Юнита 2. Автоматизированные информационные системы (АИС).

Юнита 3. Обеспечение автоматизированных информационных систем.

ЮНИТА 3

Излагаются вопросы программно-технического, правового и нормативно-технического обеспечения автоматизированных информационных систем (АИС). Основой, к которой привязано изложение всех вопросов, является понятие типового жизненного цикла АИС.

Кроме того, детально рассматриваются вопросы, связанные с сертификацией АИС, а также проблемы обеспечения надежности и качества информационных систем. Уделено значительное внимание вопросам информационной безопасности и организации работ при создании современных распределенных АИС по всему жизненному циклу.

Для студентов Современного Гуманитарного Университета

Юнита соответствует профессиональной образовательной программе №1

ОГЛАВЛЕНИЕ

	стр.
ДИДАКТИЧЕСКИЙ ПЛАН	4
ЛИТЕРАТУРА	5
ПЕРЕЧЕНЬ УМЕНИЙ	6
ТЕМАТИЧЕСКИЙ ОБЗОР	7
1. Программно-техническое обеспечение АИС	7
1.1. Классификация, состав и структура программных средств АИС ...	7
1.2. Операционные системы	10
1.3. Средства автоматизации проектирования АИС – языки 4GL и CASE-системы	24
1.4. Основы объектных распределенных технологий – стандарты CORBA, DCOM, RMI	38
1.5. Программное обеспечение интерфейсов АИС	44
1.6. Классификация, состав и структура технических средств АИС ...	49
2. Нормативно-техническое обеспечение качества, эффективности и безопасности АИС	54
2.1. Стандартизация типового жизненного цикла АИС	54
2.2. Показатели качества, эффективность и надежность АИС	57
2.3. Сертификация АИС	60
2.4. Безопасность АИС	62
3. Организационное и правовое обеспечение АИС	67
3.1. Организация работ по этапам жизненного цикла АИС	67
3.2. Защита авторских прав, рынок компонентов АИС и их интеграция	73
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	80
ТРЕНИНГ УМЕНИЙ	84
ГЛОССАРИЙ*	

* Глоссарий расположен в середине учебного пособия и предназначен для самостоятельного заучивания новых понятий.

ДИДАКТИЧЕСКИЙ ПЛАН

Классификация, состав и структура программных средств АИС. Операционные системы. Назначение и состав современных операционных систем. Интеграция сетевых ОС различных типов. Средства автоматизации проектирования АИС. Языки 4GL и CASE-системы. Средства реинжиниринга. Основы объектных распределенных технологий – стандарты CORBA, DCOM, RMI. Программное обеспечение интерфейсов АИС. Классификация, состав и структура технических средств АИС. Рабочие станции. Серверы. Управления дисковыми массивами. Стандартизация типового жизненного цикла АИС. Показатели качества, эффективность и надежность АИС. Сертификация АИС и безопасность АИС. Организация работ по этапам жизненного цикла АИС. Защита авторских прав, рынок компонентов АИС и их интеграция.

ЛИТЕРАТУРА

Базовая

*1. Липаев В.В. Управление разработкой программных средств. Методы, стандарты, технология. М., 1993.

Дополнительная

2. Костогрызов А.И., Липаев В.В. Сертификация качества функционирования автоматизированных информационных систем. М., 1996.

3. Единая система программной документации. М., 1994.

4. Рассохин Д., Лебедев А. World Wide Web - всемирная информационная паутина в сети Internet. М., 1995.

Примечание. Знаком (*) отмечены работы, использованные при составлении тематического обзора.

Современный Гуманитарный Университет

ПЕРЕЧЕНЬ УМЕНИЙ

№	УМЕНИЕ	АЛГОРИТМ
1.	Поиск в Internet информации о технической поддержке по ПС для создания Web-серверов методом ввода ключевых слов через кнопку «Поиск»	<ol style="list-style-type: none"> 1. Запустить проводник (браузер) 2. Открыть меню File (Файл) 3. Открыть подменю Open (Открыть) 4. Набрать в окне “Адрес” URL нужного объекта и “ввод” 5. На открывшейся Web-странице произвести дальнейший поиск вводом ключевых слов 6. Получив в окне проводника найденную информацию, сохранить ее в виде файла на своем ПК, если результат положительный
2.	Поиск в Internet информации о технической поддержке фирм-производителей ИТ через адресную строку вводом ключевых слов	<ol style="list-style-type: none"> 1. Запустить проводник (браузер) 2. Открыть меню File (Файл) 3. Открыть подменю Open (Открыть) 4. Набрать в окне “Адрес” URL нужного объекта и “ввод” 5. Получив в окне проводника главную Web-страницу фирмы - производителя ПС, ознакомиться с ее содержанием 6. Найти информацию о технической поддержке серверов фирмы через адресную строку вводом ключевых слов 7. Распечатать Web-страницу на принтере
3.	Поиск в Internet через адресную строку вводом ключевых слов информации об обучении пользователей фирмами-производителями ИТ	<ol style="list-style-type: none"> 1. Запустить проводник (браузер) 2. Открыть меню File (Файл) 3. Открыть подменю Open (Открыть) 4. Найти информацию о технической поддержке серверов фирмы через адресную строку вводом ключевых слов 5. Получив в окне проводника найденную информацию, сохранить ее в виде файла на своем ПК, если результат положительный

ТЕМАТИЧЕСКИЙ ОБЗОР*

1. ПРОГРАММНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ АИС

1.1. Классификация, состав и структура программных средств АИС

Классификация – один из фундаментальных процессов в научно-технической деятельности. Факты и явления должны быть упорядочены, прежде чем их можно будет понять и разработать общие принципы, объясняющие их группировки, порядок следования и общие свойства. Без классификации нельзя достичь глубокого понимания природных явлений и различных видов интеллектуальной человеческой деятельности.

Классификация – это упорядочение объектов по их схожести, а объектом можно назвать, в том числе, и процессы, и действия – все, чему можно приписать вектор дескрипторов. Классифицировать – значит найти основу для суждений о сходстве, которая достигается с помощью детального описания свойств классифицируемых объектов. Каждому объекту приписываются списки дескрипторов (векторы значений признаков), а классификация проводится по матрице данных, созданной из набора таких векторов. Отсюда следует, что могут создаваться различные классификации на основе различных наборов векторов.

Например, граница между системным и прикладным программированием зависит от конкретной ситуации. С точки зрения системного программиста, который отвечает за работу операционной системы, программист разрабатывающий компилятор, занимается прикладным программированием, то есть является пользователем системного программного обеспечения. Для разработчика функциональных программ автоматизированной информационной системы создатель компилятора является системным программистом, так же, как и специалист по операционной системе.

Под **программным обеспечением АИС** (функциональной компонентой) будем понимать совокупность общесистемных и функциональных программ, а также программ системы обработки данных и программных документов, необходимых для эксплуатации этих программ. Программы, обеспечивающие возможность выполнения АИС основных функций, практически не зависящих от специфики конкретных задач и областей применения входят в состав **программного обеспечения общесистемного назначения**. Тогда **прикладное программное обеспечение АИС** – это программное обеспечение, предназначенное для решения определенной задачи в предметной области или для предоставления пользователю определенных услуг.

Иногда используется термин “**программная система**” – **программное средство (ПС)** - программная продукция, представляющая собой совокупность программ и (или) подсистем, имеющих общее целевое назначение. ПС, оформленное в соответствии с определенными требованиями, называется программным изделием - программой на носителе данных, являющейся продуктом промышленного производства.

На рис.1. представлены состав программного обеспечения АИС, компоненты которого классифицированы согласно функциям, выполняемым ими на различных этапах полного жизненного цикла АИС. В целом, при изложении данного курса мы последовательно придерживаемся концепции

* Жирным шрифтом выделены новые понятия, которые необходимо усвоить. Знание этих понятий будет проверяться при тестировании.

полного жизненного цикла АИС, поэтому и классификация ПО АИС основывается на этой концепции.

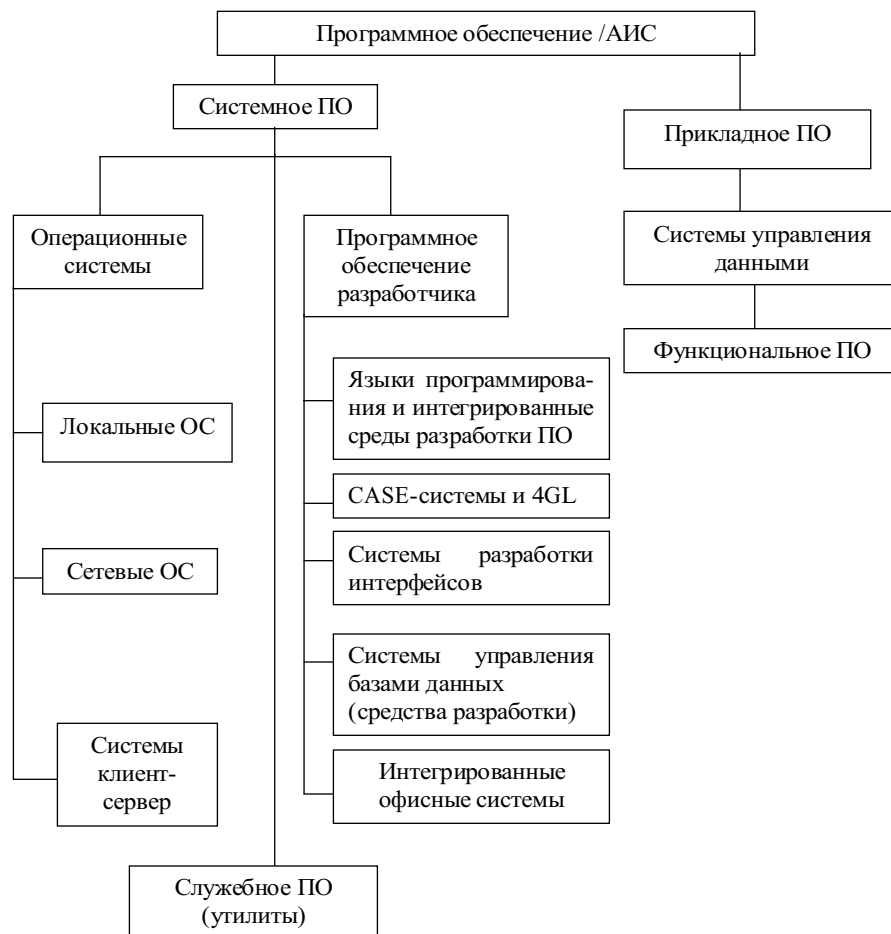


Рис. 1. Состав программного обеспечения АИС.

Базой ПО АИС, услугами которой пользуются все остальные компоненты является **операционная система** – система программ, предназначенная для обеспечения определенного уровня эффективности вычислительной системы за счет автоматизированного управления ее работой и предоставляемых пользователям определенного набора услуг. В группу базового ПО, кроме ОС входят программные средства служебного назначения, которые выполняют частные общесистемные функции:

системы управления базами данных (СУБД) – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами;

служебные программы (утилиты) – набор специальных программ, предназначенных для повышения эффективности рутинных работ по обслуживанию информационно-вычислительной среды, например копирования и сжатия данных, дефрагментация жесткого диска и т.д.

На этапах концептуального, технического и рабочего проектирования наиболее интенсивно используются программные средства разработки, которые в настоящее время оформляются в виде **интегрированных сред** – программных систем, включающих все необходимые пользователю средства и обеспечивающие единообразие с ними. Можно выделить интегрированные среды следующего назначения:

средства автоматизации проектирования и переноса АИС (CASE - технологии) - системы программного обеспечения, которые основываются на методологиях коллективной разработки и сопровождения АИС и обеспечивают автоматизацию всех этапов их жизненного цикла;

интегрированная среда разработки программ (ИСП) – система программ, которая упрощает процесс программирования и делает его более эффективным, ИСП содержит весь комплекс средств, необходимых для написания, редактирования, компиляции, компоновки и отладки программ;

интегрированная система пользователя - несколько взаимосвязанных пакетов прикладных программ, в том числе текстовый редактор, электронная таблица, база данных, деловая графика, средства коммуникации;

инструментальная система - программно - техническая система, позволяющая пользователю, не владеющему языками программирования создавать личные приложения и (или) их модели.

Некоторые средства разработки имеют замкнутый характер, в частности ряд **систем автоматизированного проектирования (САПР)** – системы, имеющие альтернативное программное обеспечение и операционную систему автоматизированного проектирования, позволяющую выбирать совокупность машинных программ применительно к заданному объекту проектирования или классу объектов проектирования.

С учетом быстрого развития в современных АИС средств телеобработки данных – совокупности методов, обеспечивающих пользователям дистанционный доступ к ресурсам систем обработки данных и средств связи – целесообразно выделить в составе ПО АИС **сетевые программные средства**. Это средства разработки и реализации методов, обеспечивающих установление связи, обработку и передачу данных между различными ЭВМ или (и) удаленными или локальными терминалами и абонентскими пунктами.

В состав прикладного ПО АИС входят:

функциональные прикладные программы, предназначенные для решения задачи или класса задач в определенной области применения систем обработки данных;

интерфейсы пользователя – совокупность методов, правил, программных и аппаратных средств, обеспечивающих взаимодействие пользователей с программами или другими видами ресурсов АИС.

Работы по всему жизненному циклу выполняются на **автоматизированных рабочих местах (АРМах)** - программно-технических системах, обеспечивающих возможности доступа пользователя к средствам разработки и ресурсам автоматизированной информационной системы.

1.2. Операционные системы

1.2.1. Основные понятия

Операционные системы осуществляют управление системными ресурсами с целью эффективного, безопасного и надежного выполнения прикладных программ. В основе любой ОС лежит **ядро операционной системы** – резидентная часть, управляющая процессами операционной системы и распределяющая для них системные ресурсы. Резидентная часть программного обеспечения – это часть программного обеспечения, которая располагается постоянно в оперативной памяти ЭВМ, а системный ресурс – любое средство вычислительной системы, которое может быть выделено процессу.

При включении питания ЭВМ **системный загрузчик** (компонента управляющей программы) осуществляет начальную загрузку ядра операционной системы и его инициализацию, после чего все управление системными процессами осуществляет ОС. Взаимодействие между ОС и всеми остальными программно-техническими компонентами АИС осуществляется через систему прерываний. **Прерывание** – это операция процессора, состоящая в регистрации состояния процессора, предшествовавшего прерыванию, и установлении нового состояния, а также реакция процессора на некоторое условие, которое возникло в процессе или вне его; прерывание позволяет обработать возникшее условие специальной программой и вернуться к прерванной программе. Следовательно **система прерываний** – это комплекс технических и программных средств, обеспечивающих возможность прерывать выполнение программы при поступлении внешних или внутренних сигналов прерывания и продолжить выполнение прерванной программы от точки прерывания после обработки сигналов прерывания. Различают следующие основные виды прерываний:

прерывание ввода-вывода – внешнее прерывание, инициируемое устройствами ввода-вывода (в том числе каналами связи) при изменении их состояния, например завершение операции ввода-вывода, ошибка ввода-вывода, готовность аппаратуры;

прерывание по защите памяти – программное прерывание, обусловленное попыткой обратиться к данным, размещенным в области памяти, недоступной данной программе;

прерывание по сбою – один из типов прерываний, возникающий при появлении сигнала от схем контроля ЭВМ.

В системе прерываний устанавливаются **приоритеты прерываний** – числовые значения (ранг), приписываемые прерыванию определенного типа. При одновременном поступлении нескольких прерываний обслуживается прерывание с большим приоритетом (приоритет – характеристика объекта, определяющая его права на предоставление ресурсов).

ОС состоит из набора системных программ, входящих в состав операционной системы и выполняющих управляющие и обслуживающие функции. Системные программы размещаются в **системной библиотеке** – библиотечном наборе данных, содержащем определенные компоненты операционной системы. Каждый ресурс имеет **системное имя** – уникальное имя, идентифицирующее его в данной операционной системе. Все события, происходящие в управляемой ОС системе, регистрируются в системном журнале, специальном наборе данных.

В современных АИС используются следующие виды ОС:

локальные операционные системы, осуществляющие управление ресурсами в пределах конкретного вычислительного комплекса, без

использования каналов связи;

сетевые операционные системы, осуществляющие управление ресурсами в локальных и глобальных сетях передачи данных.

Кроме того могут использоваться специальные ОС, например:

операционные системы автоматизированного проектирования – часть программного обеспечения автоматизированного проектирования, предназначенная для управления проектированием;

операционные системы реального времени – операционная система, обеспечивающая режим работы ЭВМ в реальном времени;

и др.

Система реального времени – это система, осуществляющая информационный обмен с другими системами, периферийными устройствами или датчиками при таких временных характеристиках, которые позволяют немедленно обрабатывать всю поступающую информацию и информацию для вывода. Эти системы обычно используются в управлении техническими объектами (АСУ ТП).

Важными частями любой ОС являются:

файловая система – часть операционной системы, обеспечивающая выполнение операций над файлами;

система планирования – программа упорядочивания последовательности работ в системе, обеспечивающая ее максимальную эффективность.

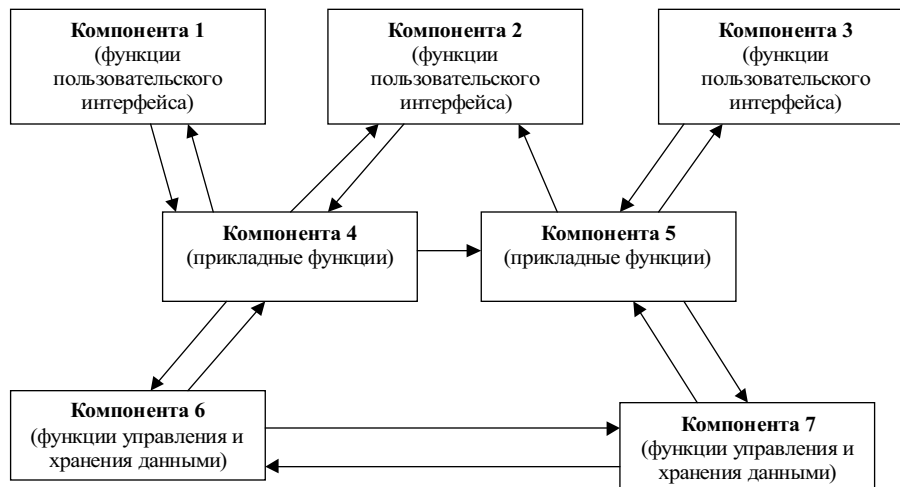


Рис. 2 Схема распределенной АИС

Значительная доля современных АИС имеет распределенный характер (см. рис. 2), то есть их компоненты распределены по различным вычислительным комплексам, которые связаны между собой **каналами передачи данных** – средствами двустороннего обмена данными, представляющими собой совокупность аппаратуры окончания канала данных и линии передачи данных. Используется также термин **передающая среда** – совокупность линии передачи данных и, возможно, коммутаторов данных, концентраторов данных, повторителей и другого оборудования, не

относящегося к станции данных, организация структуры и функционирование которой обеспечивают физическую передачу данных между станциями данных. Распределенные АИС базируются на **сетях ЭВМ** – совокупности сети передачи данных, взаимосвязанных ею ЭВМ и необходимых для реализации этой связи программного обеспечения и технических средств, которая предназначена для организации распределенной обработки информации. Более широким, является понятие **информационная сеть** – совокупность сети ЭВМ и взаимодействующих через сеть ЭВМ удаленных реальных оконечных систем, обеспечивающая доступ прикладных процессов, расположенных в любой из этих систем, ко всем ее информационным, вычислительным, коммуникационным ресурсам, и коллективное их использование.

Логическая структура и принципы работы информационной сети называются архитектурой сети, для описания которой используются следующие основные понятия:

передача данных – пересылка данных с помощью средств связи из одного места для приема их в другом;

функциональный блок сети – система или устройство, выполняющая определенную логически связанную группу функций;

источник данных – функциональный блок, порождающий данные для передачи;

приемник данных – функциональный блок, принимающий переданные данные;

протокол – набор семантических и синтаксических правил, определяющий поведение функциональных блоков при передаче данных;

обмен данными – передача данных в соответствии с установленным протоколом;

соединение – связь, устанавливаемая между функциональными блоками для передачи информации;

ресурсы сети – программное, техническое, информационное и организационное обеспечение информационной сети, предназначенное для решения прикладных задач.

Среди информационных сетей особое значение для построения распределенных АИС имеет концепция открытых информационных сетей.

Открытая информационная сеть – это информационная сеть, взаимодействие всех входящих в состав которой реальных систем, а также взаимодействие которой с реальными системами, не входящими в ее состав, подчиняются требованиям стандартов Международной организации по стандартизации. Совокупность технических и (или) программных средств, входящая в состав открытой информационной сети, реализующая согласованные по горизонтали и вертикали функции некоторого уровня и всех расположенных ниже уровней базовой эталонной модели и выполняющая функции поставщика сервиса для компонентов сети, называется **службой взаимосвязи открытых систем**.

Разновидностью открытых систем являются АИС, реализованные на базе сетей типа **“intranet”** – корпоративных сетей, созданных на базе технологий Internet. В отличие от слова “Internet”, которое имеет смысловой перевод (inter – между, net – сеть), “intranet” является чисто искусственным словообразованием (в английском языке нет слова “intra”), которое принято писать с маленькой буквы. Internet – имя собственное вполне конкретной, хотя и глобальной сети, а intranet – обозначение абстрактного понятия. В русском языке стал использоваться его транслитерный перевод – “интрасеть”.

В принципе все продукты и решения, разработанные для Internet, могут использоваться и для интрасетей. Но тем не менее в их реализации есть некоторые особенности, которые объясняют выделение технологии создания

интрасетей в некоторое самостоятельное направление и, соответственно, появление для них некоторых специфических программных решений. Internet представляет собой сеть из компьютеров-узлов, взаимодействие между которыми выполняется с помощью протоколов TCP/IP. На физическом уровне узлы обычно связаны в некоторые внутренние сети (например, Relcom), поэтому Internet фактически является иерархической структурой: "Internet-сети-узлы". Но с логической точки зрения каждый узел является самостоятельной единицей, имеющей свой уникальный IP (Internet Protocol) адрес, что обеспечивает возможность общения (в режиме on-line или off-line - это неважно) любых двух узлов Internet. В основе организации Internet лежат две основные идеи.

Во-первых, взаимодействие узлов не должно зависеть от типов компьютеров, их архитектур, операционных систем и пр., а также физической реализации связи между ними. Для осуществления этого используются простые и надежные протоколы установления связи и передачи данных. Причем вопрос о том, насколько узел понимает смысл принимаемых или передаваемых данных, является его личной проблемой.

Во-вторых, Internet должен надежно функционировать независимо от изменения своей топологии, в частности, от подключения или отключения его узлов. Причина обеспечения такой надежности заключается в том, что этот процесс никак не контролируется. Единственное, за чем здесь осуществляется контроль - это за обеспечением уникальности IP адресов узлов и соблюдением определенного порядка в их нумерации для решения проблем маршрутизации.

Логика развития корпоративных компьютерных сетей привела к объединению локальных сетей, реализованных, условно говоря, на уровне подразделения (здания), в глобальные. В качестве примера можно представить себе сеть корпораций, имеющих сотни подразделений по всему миру. Или банк, имевший несколько десятков отделений, в каждом из которых были свои собственные локальные сети (а иногда и несколько). Использование идей Internet для создания достаточно больших корпоративных сетей является довольно очевидным (и оно началось еще до того, как появился термин intranet): здесь то же самое разнообразие типов компьютеров, линий связи, а также довольно хаотичное изменение топологии. Причем актуальность данной проблемы проявляется не только на уровне крупных корпораций или банков, но и при создании общей сети даже не самого крупного завода или вуза. По некоторым оценкам схема интрасетей становится приемлемой уже при наличии порядка 100 компьютеров.

Однако создание интрасетей на уровне прямого взаимодействия компьютеров через Internet является просто невозможным. Во-первых, это может быть просто нецелесообразно для относительно небольших сетей, в которых объединение компьютеров выполняется за счет чисто внутренних линий (например, на территории завода). Во-вторых (и это самое главное), из-за необходимости обеспечения информационной защиты при взаимодействии с внешним миром. Ведь открытые протоколы Internet делают прозрачными все включенные в него узлы. Например, узел в принципе не в состоянии сам определить, с каким абонентом – внутренним или внешним - он имеет дело. Естественным решением данной проблемы является то, что внутренняя корпоративная сеть должна быть физически отделена от внешнего мира, а общение между ними выполняется только через специальные шлюзы, которые получили название прокси серверов.

Специфика программных продуктов и технологий для интрасетей заключается в том, что число узлов в них является ограниченным и изменение топологии сети и содержимого информации носит некоторый предсказуемый

характер. А в этом случае организация таких процедур как, например, маршрутизация и поиск информации может быть оптимизирована совершенно другими методами, чем в хаотичном Internet.

При создании АИС на базе интрасетей необходимо решить следующие задачи:

- полностью интегрировать внутренние локальные сети с Internet для расширения возможностей коммуникации предприятий с клиентами и партнерами;

- реализовать новые методы навигации, применяемые в Internet, во всех продуктах, с тем чтобы облегчить операции поиска и анализа информации, а также взаимодействия с партнерами.

- упростить разработку, внедрение и администрирование прикладных программ с целью оптимизации деловых процессов на предприятиях и сокращения циклов разработки;

- интегрировать новые продукты и Internet-технологии с существующими инфраструктурами, чтобы пользователи могли на основе уже имеющихся решений постепенно развивать свои информационные системы.

1.2.2. Назначение и состав современных операционных систем

Современные операционные системы – это системы, обеспечивающие управление ресурсами и координацию функционирования компонентов АИС, распределенных в корпоративных или глобальных сетях передачи данных. В основе их работы лежит технология клиент-сервер (см. рис.3). Клиент – это компьютер (программа), осуществляющий доступ к ресурсам другого компьютера, предоставленным в совместное использование, а сервер – компьютер (программа), предоставляющий свои ресурсы для совместного использования в сети. В качестве примера современной сетевой ОС рассмотрим Windows NT. Сегодня она широко применяется в России и за рубежом.

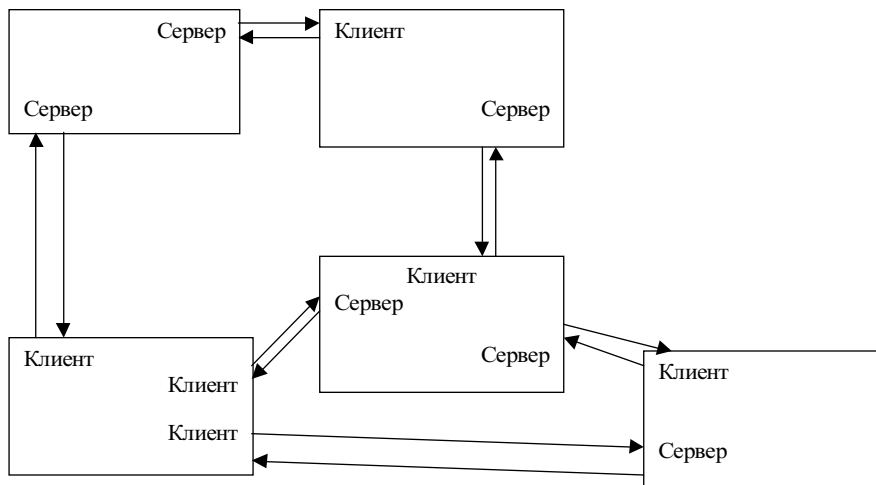


Рис. 3. Схема взаимодействия клиент-сервер.

Microsoft Windows NT – это достаточно мощная, надежная, быстродействующая операционная система, которая обладает высоким уровнем защиты данных, работает с различными файловыми системами, на различных аппаратных платформах и поддерживает многопроцессорные системы. ОС выпускается в двух версиях – это сервер Windows NT Server и Windows NT Workstation 4.0., предназначенная для АРМов (рабочих станций). В принципе NT Workstation может выполнять и функции невыделенного сервера в одноранговой сети, но использовать ее в качестве полноценного сервера нельзя. В ней отсутствуют средства администрирования и управления доменами, оптимизация подсистем для исполнения приложений и др.

Windows NT 4.0. имеет пользовательский интерфейс (оболочку), аналогичный интерфейсу Windows 95. Работая в Windows NT, можно использовать стандартные средства Windows 95, такие как Explorer (Проводник), Network Neighborhood (Сетевое окружение), Briefcase (Портфель) и т.д. Windows NT имеет и специфические функции, которых нет в Windows 95. Например, в Windows NT Workstation есть функции динамического компрессирования файлов при использовании файловой системы NTFS, управление правами доступа к отдельным файлам, каталогам и т.д. **Файловая система NTFS** – это файловая система, разработанная специально для Windows NT. Поддерживает средства восстановления файловой системы, допускает использование чрезвычайно больших носителей данных, а также различных специальных функций. Поддерживает объектно-ориентированные приложения, обрабатывая все файлы как объекты с определяемыми пользователем и системой атрибутами.

В ОС встроена возможность автоматического тиражирования системы на большое число компьютеров, что позволяет избежать рутинной процедуры установки вручную.

В Windows NT Workstation 4.0 поддерживаются технологии, необходимые для работы факс-приложений, системы обмена сообщениями, Internet Explorer, клиента и сценариев входа в сеть NetWare (далее этот вопрос будет рассмотрен подробнее). Система обеспечивает возможность выбрать при загрузке определенную конфигурацию, с которой будет работать конкретный пользователь. Профиль, который выбирается при загрузке, включает в себя настройки разрешения экрана, информацию о запускаемых сервисах и устройствах.

В ее состав входят средства для работы с Internet и создания Intranet-решений. Это Internet Explorer – средство для навигации по серверам в Internet и Peer Web Services, который позволяет создавать персональные Web-страницы.

Большое внимание уделено вопросам безопасности. Имеется целый набор функций, позволяющих разработчикам создавать приложения, способные безопасно исполняться в таких незащищенных сетях, как Internet. Например, такие функции как:

- контекстные** позволяют приложениям подключаться к различным криптографическим системам, выбирая определенную систему;

- генерации ключей** позволяют приложениям создавать и настраивать криптографические ключи;

- обмена ключами** позволяют приложениям передавать или обмениваться ключами;

- шифрования данных** выполняют шифрование/дешифрацию данных в приложениях;

- кэширования и цифровой подписи** позволяют приложениям вычислять криптографически защищенные обзоры данных, а также выполнять цифровую подпись данных.

Windows NT Workstation 4.0 включает универсальный почтовый ящик

Microsoft Exchange, управляющий поступающей почтой и факс-сообщениями. В этот почтовый ящик попадают все приходящие на имя пользователя сообщения из любой системы электронной почты.

Реализован также инструмент, позволяющий администраторам сети управлять АРМами. Это происходит путем ограничения их доступа в сеть, создания определенных настроек и запрещения пользователям изменять начальную конфигурацию. Администраторы, работающие на Windows NT Workstation 4.0, могут управлять как компьютерами с Windows 95 так и компьютерами с Windows NT Workstation.

Windows NT Server может выступать как:

- файл-сервер;
- сервер печати;
- сервер приложений;
- контроллер домена;
- сервер удаленного доступа;
- сервер Internet;
- сервер обеспечения безопасности данных;
- сервер резервирования данных;
- сервер связи;
- сервер вспомогательных служб.

В режиме **файл-сервера** ОС обеспечивает работу компьютера как централизованного хранилища большого количества файлов (баз данных), коллективно используемых в АИС, в том случае, когда хранение этой информации на локальном компьютере нецелесообразно. Все файловые ресурсы независимо от того, на каком диске они расположены (жестком или CD-ROM), сразу могут быть предоставлены для совместного использования.

Как **сервер печати** NT позволяет работать и предоставлять в совместное пользование неограниченное число принтеров. Они могут быть подключены локально или по сети с помощью протоколов TCP/IP. В качестве АРМов (рабочих станций) могут выступать компьютеры с операционными системами: MS-DOS, OS/2, Macintosh, Windows NT Workstation. При работе на АРМе с системой Windows NT Workstation, чтобы подключиться к удаленному принтеру, предоставляемому Windows NT Server, нужно лишь выбрать этот принтер из числа доступных. Система не будет запрашивать дискеты с драйверами и т.п., она использует тот драйвер, что установлен на сервере.

Режим **сервера приложений** возникает в системах, способных исполнять основное ("тяжелое") приложение на мощном высокопроизводительном сервере, а результаты по запросам передавать на маломощные клиентские АРМы, реализуя модель клиент-сервер. Изначально построенная по схеме клиент-сервер, Windows NT приспособлена для работы в системах клиент-сервер в качестве сервера приложений. В первую очередь, такими приложениями являются системы управления базами данных, информационного обмена, системы управления.

В режиме **сервера резервирования** данных NT осуществляет возможность резервного копирования файлов на магнитную ленту. Администратор системы определяет пользователя, ответственного за эту операцию, и только он регулярно выполняет копирование данных на стример. При необходимости эту операцию можно автоматизировать.

Служба **удаленного доступа** (Remote Access Service - RAS) состоит из двух частей: серверной, устанавливаемой на ЭВМ с NT Server, и клиентской, устанавливаемой на АРМах с MS-DOS, Windows95 или Windows NT Workstation. Пользователь АРМа, связанного с сетью через сервер удаленного доступа, чувствует себя работающим непосредственно в сети: он может осуществлять доступ к файлам и данным, печатать документы, подключаться к другим

серверам и обмениваться с коллегами сообщениями по электронной почте.

Такой прозрачный доступ к сети удобен тем, кто постоянно бывает в разъездах, командировках, а также для администраторов системы. Он широко применяется и для связи территориально удаленных филиалов предприятий, при этом одновременно поддерживается до 256 сессий удаленного доступа.

Набор протоколов *Point-to-Point Protocol* (PPP) позволяет осуществлять удаленный доступ в условиях разнородной сети. Поддержка PPP гарантирует возможность удаленного доступа через любой стандартный PPP-сервер удаленного доступа. С другой стороны, Windows NT Server способен соединяться и обеспечивать доступ к сети для пользователей, применяющих средства удаленного доступа сторонних производителей. Он поддерживает любую комбинацию основных протоколов при удаленном доступе и с помощью протокола TCP/IP можно подключаться к Internet.

Режим сервера **связи сетей** подразумевает возможность соединения между собой различных сегментов сети, в том числе сопряжения разнородных сетей.

Windows NT Server 4.0 имеет интерфейс с пользователем идентичный тому, который используется в Windows 95 и Windows NT Workstation. Единство интерфейса всех 32-разрядных платформ снижает затраты на обучение, а также упрощает переход с одной платформы на другую в семействе Windows. В версии 4.0 добавлены новые программы-мастера администрирования, диспетчер заданий и утилиты диагностики.

Диспетчер заданий позволяет просмотреть не только список исполняемых задач, но также и все процессы, работающие в системе. В дополнение к этому можно в наглядной графической форме следить за загруженностью процессора и использованием памяти. Эти возможности позволяют быстро обнаружить некорректно работающую задачу или процесс и, при необходимости, прервать его.

Программы-мастера администрирования позволяют:

- создавать новые учетные записи пользователей и групп;
- управлять предоставлением доступа к сетевым и локальным ресурсам сервера;
- создавать новые принтеры в системе и предоставлять их в совместное использование;
- конфигурировать модемы;
- устанавливать клиентское программное обеспечение;
- добавлять новые приложения или удалять существующие;
- отслеживать правильность соблюдения лицензионной политики.

Программа диагностики предоставляет информацию о драйверах устройств, прерываниях, используемых адресах и настройках сетевых устройств, что упрощает поиск неисправностей в системе. Эта информация выводится в графическом виде и может быть доступна с удаленного компьютера.

Редактор системных правил позволяет определять конфигурацию компьютеров как для отдельных пользователей, так и для всех пользователей **домена** - объединения нескольких компьютеров, использующих единую базу учетных записей и политику безопасности, каждый домен имеет уникальное имя. Использование системных правил совместно с индивидуальными профилями пользователей позволяет как ограничивать пользователей в их действиях, так и предоставлять им максимально допустимую свободу действий. Профиль – это файл, содержащий информацию о параметрах окружения пользователя, например сетевых подключениях, группах программ, положении и размерах окон, сохраняемых при выходе из Windows NT.

Правила загружаются при регистрации пользователя на компьютере и

заменяют собой стандартные установки, хранящиеся в реестре системы. Администратор имеет возможность использовать не только стандартные правила, но и свои собственные, влияющие на конфигурацию приложений, используемых в конкретной системе.

Служба каталогов Windows NT (Microsoft Windows NT Directory Service – NTDS) используется для упрощения работы и управления корпоративными сетями. NTDS является службой каталогов, предназначенной для современных сетей и основанной на защищенной базе каталогов, хранящей идентификаторы пользователей, пароли, права доступа и организационную структуру. База каталогов может автоматически тиражироваться в несколько мест для обеспечения надежного резервирования, балансировки нагрузки и равномерной загрузки сети. Зарегистрировавшись один раз пользователи имеют доступ ко всем ресурсам сети корпорации. Администратор сети использует NTDS для добавления новых пользователей, авторизации доступа к ресурсам сети, а также для отслеживания изменений персонала, организаций и информационных технологий. **Авторизация** – это проверка регистрационной информации о пользователе. Если пользователь регистрируется на Windows NT Workstation, авторизация выполняется на этом АРМе. Все, что нужно помнить пользователю – это его имя и пароль, независимо от того, с какого компьютера он входит в сеть или какой сетевой ресурс он собирается использовать. **Пароль** представляет собой уникальную строку символов, вводимую для авторизации доступа при регистрации. Важное средство защиты, пароль служит для ограничения входа в систему и доступа к компьютерным системам и ресурсам. На каждого пользователя создается администратором **персональный профиль пользователя**, в который записываются все изменения, сделанные пользователем в течение сеанса работы. Эти изменения сохраняются при выходе пользователя из системы. При последующей регистрации на любой рабочей станции с Windows NT Workstation загружается персональный профиль (файл с расширением .USR) и выставляются те параметры окружения, что были перед окончанием предыдущего сеанса.

Администраторы вводят информацию о новых пользователях в графические формы на экране (либо загружая файлы). Определение организационных групп и предоставление прав доступа осуществляется несколькими операциями буксировки. Реорганизация выполняется также легко путем перетаскивания идентификатора пользователя из одной группы в другую. При этом права пользователя соответственно изменяются автоматически. Пользователи могут объединяться в локальные, глобальные и специальные группы. **Локальная группа** – это группа, которой можно предоставлять права и привилегии исключительно для АРМов. Локальные группы предоставляют способ создания объединений пользователей как этой рабочей станции, так и нет, для применения только на рабочей станции. **Глобальная группа** – это группа, которая может использоваться в собственном домене, серверах и рабочих станциях домена и доверяющих доменах. Во всех случаях глобальная группа имеет предоставленные права и привилегии и может становиться членом локальных групп. **Специальные группы** – это группы, члены которых не назначаются администратором. Пользователь становится членом такой группы при выполнении определенных действий в сети. Например, пользователь, выполнивший интерактивную регистрацию, принадлежит к специальной группе Interactive. Под **привилегиями** понимаются возможности пользователя выполнять определенные действия в системе. Привилегии предоставляются системе в целом в отличие от прав доступа, применимых к определенным объектам.

С помощью NTDS можно управлять большим количеством пользователей (вплоть до 40000) входящих в одну организационную единицу (домен), или

создать несколько доменов, которые могут управляться независимо. Объединяя домены, можно практически неограниченно увеличивать число пользователей. Имеется возможность комбинирования управления пользователями или ресурсами, а также разделения полномочий между разными администраторами, расположенными в разных подразделениях или географических точках. В зависимости от того, насколько АИС является централизованной или распределенной NTDS позволяет сконфигурировать каталоги в точном соответствии со структурой АИС и при этом нигде не отойти от правила предоставления однократной регистрации в сети для доступа к ресурсам.

Важной характеристикой является производительность сервера, которая определяется возможностью предоставлять файлы и принтеры в совместное использование, исполнять серверные приложения, а также увеличивать производительность с увеличением числа процессоров.

Windows NT Server обладает следующими возможностями:

- поддержка симметричной мультипроцессорной обработки до 32 процессоров;

- переносимость на основные высокопроизводительные процессоры (Intel, DEC Alpha, MIPS, PowerPC);

- развитые функции дисковой подсистемы, такие как зеркализация (создание точной действующей копии баз данных, которая может быть использована в случае выхода из строя оригинала).

При использовании в качестве линии связи между двумя локальными сетями обычных телефонных каналов, со скоростью обмена данными 9600-19200 Кбит/сек., в Windows NT 4.0 можно соединить два компьютера по нескольким телефонным каналам параллельно. Суммарная пропускная способность такого канала увеличивается пропорционально числу задействованных телефонных линий. Данная функция доступна как для модемной связи, так и для сетей ISDN. В этом случае при инициации связи со стороны клиента будет одновременно осуществляться доступ по всем доступным каналам сразу.

Point-to-Point tunnelling protocol (PPTP) является сетевой технологией, появившейся в Windows NT в версии 4.0, и позволяющей организовывать виртуальные корпоративные сети путем безопасного соединения локальных сетей через Internet. PPTP позволяет клиентам осуществлять доступ к корпоративной сети из любой точки земного шара путем подключения к Internet. В любом случае такое подключение выполняется совершенно безопасно и использует механизмы шифрования. Виртуальная глобальная сеть поддерживается общедоступными каналами, например, Internet. Прямая выгода здесь очевидна – вместо использования дорогостоящих междугородних или международных каналов используется стандартный и более дешевый канал.

Построение глобальной сети невозможно без **маршрутизации**. Для ее осуществления используются **маршрутизаторы**, позволяющие осуществлять взаимодействие между различными локальными и глобальными сетями, а также между сетями с различной топологией (например, Ethernet и Token Ring). В каждом пакете, пересылаемом в локальной сети имеется заголовок, содержащий поля с исходным адресом и адресом назначения. Маршрутизаторы сравнивают заголовки пакетов с сегментами сети и выбирают наилучший путь для их прохождения, что повышает производительность сети. В Windows NT Server сервер удаленного доступа может быть использован для организации маршрутизации между удаленным клиентом и локальной сетью или между двумя локальными сетями.

Windows NT Server обладает встроенной поддержкой русского языка. Эта поддержка выражена не просто в возможности использования русских

шрифтов и ввода русского текста, но и в целом ряде дополнительных возможностей. Так, например, на сервере можно исполнять приложения, имеющие русскоязычный интерфейс, посылать и принимать сообщения на русском языке, просматривать электронную почту, работать с файлами, имеющими длинные русские имена, работать с русскоязычными документами Web.

1.2.3. Операционная система Solaris - коммерческая, открытая, промышленная ОС типа UNIX

Альтернативными Windows NT операционными системами являются сетевые ОС типа UNIX. Рассмотрим одну из наиболее распространенных – ОС Solaris (фирма Sun).

В настоящее время Solaris представляет собой операционную систему, существующую в вариантах для процессоров SPARC (фирма Sun) и Intel x86. Созданная как основа для создания сетевой среды, ОС Solaris предполагает многозадачный, многопоточный, многопользовательский режим функционирования и соответствует большинству признанных международных промышленных стандартов. ОС Solaris предназначена для работы в корпоративных вычислительных сетях. Она обеспечивает эффективный и надежный доступ к системам в целом, серверам, базам данных, принтерам и другим сетевым ресурсам. В нее встроены и оптимизированы на уровне ядра операционной системы средства обеспечения сетевого взаимодействия: от обеспечения взаимодействия прикладных программ и создания распределенных файловых систем, - до поддержки создания полностью распределенной среды вычислений. Solaris является также платформой для разработки сложного программного обеспечения с целым арсеналом средств, позволяющим сократить время разработки приложений, повысить надежность их функционирования, и нацеленных на создание интегрированных систем. Эта ОС относится к классу коммерческих, открытых, промышленных операционных систем типа UNIX.

ОС Solaris в высокой степени соответствует промышленным стандартам и имеет свойство масштабируемости. То есть, одна и та же операционная система используется и на однопроцессорных рабочих станциях (APMax) и на серверах масштаба рабочей группы и на 64-процессорном сервере масштаба предприятия. Для широкого круга приложений ОС Solaris обеспечивает практически линейный рост производительности при увеличении числа процессоров в системе. Масштабируемость представляет собой воплощение концепции симметричных многопроцессорных вычислений (SMP - Symmetric Multi-Processing) и позволяет увеличивать производительность систем без коренной переработки приложений, функционирующих в сети предприятия.

Концепция открытых систем предполагает наличие средств, позволяющих наладить эффективное и надежное сетевое взаимодействие между платформами различных типов и производителей. ОС Solaris, используя в качестве основного сетевого протокола TCP/IP (протокол Internet), может также поддерживать связь по протоколам SNA, DECnet и обеспечивает интеграцию сетей ПК. Серверы с операционной системой Solaris могут встраиваться в сети и домены сетей Novell и Microsoft.

В последних версиях ОС Solaris (2.6) поддерживается концепция "Web-гудка". В этой концепции полагается, что доступ к Web-страницам должен быть очевидным и вошедшим в привычный обиход. Ближайшая аналогия - появление гудка в телефонной трубке, когда практически никто не задумывается о сложной технологии, стоящей за этим событием. Solaris позволяет обеспечить непрерывный и надежный "гудок Web" и реальную

платформу для развертывания технологии Java (язык программирования Web-страниц). При таком подходе создание Web-интерфейса для многих привычных утилит операционной системы просто неизбежен. Так, в Solaris 2.6 реализован Web-старт, значительно облегчающий установку операционной системы. Система подсказок также реализована через Web.

Solaris Web Start - это Java-приложение, которое делает простой и быстрой установку Solaris и приложений Solaris. Стандартная конфигурация может быть установлена простым нажатием кнопки. В то же время гибкость специализированной установки обеспечивается наличием ряда отдельных возможностей, например, выбора структуры файловой системы. Возможности Solaris Web Start включают клиент/серверный режим удаленной установки по сети. Пользователям АРМов операционная среда Solaris с дополнительными Web-возможностями предлагает возможности сохранять, отправлять по электронной почте и "запускать" URL ("Универсальный Указатель Ресурса" в Internet) одним нажатием клавиши "мыши".

Для обеспечения безопасности при работе пользователя с корпоративными сетями и в Internet используются программные решения для Обеспечения Сетевой Безопасности и Электронной Торговли. Эти решения обеспечивают защиту от несанкционированного доступа и организуют безопасное управление внутри корпоративной сети, в том числе защиту для всех отделов и центров внутри организации, что обеспечит безопасные коммуникации, а также функционирование систем различного назначения.

Для обеспечения надежности, выполняемых работ в ОС предусмотрены технологии, предоставляющие клиентам постоянный доступ к файлам даже при отказе сетевого сервера, в том числе горячая замена - возможность заменять системные компоненты без прерывания обслуживания пользователей и программа круглосуточного обслуживания и поддержки пользователей.

Версия ОС Solaris с дополнительными Web-возможностями содержит встроенное программное обеспечение Java в качестве неотъемлемой части операционной среды. В дополнении к браузеру HotJava, стандартно включены Виртуальная машина Java, компилятор и интегрированные Java API (прикладные программные интерфейсы).

Java - это объектно-ориентированный язык программирования, позволяющий быстро создавать компактные, не зависящие от платформы сетевые приложения. Благодаря Java, достаточно простое устройство (Java-клиент, персональный помощник (PDA), интеллектуальный телефон и т.д.), подключенное к сети, может загрузить и исполнить любое приложение Java, которые охватывают все сегменты телекоммуникационной индустрии:

Java поддерживается любой операционной системой и работает на любой компьютерной платформе;

ведущие производители баз данных (Oracle, Informix, Sybase) встраивают Java в свои приложения;

ведущие производители сетевого оборудования используют Java для создания следующего поколения интеллектуальных коммутаторов;

Java используется в качестве платформы для создания Web-телефонов (стационарных и мобильных телефонов с электронной почтой и доступом в Internet);

Java можно рассматривать как универсальную среду, объединяющую все стороны телекоммуникаций: компьютеры, приложения, услуги, сети, сетевое и телекоммуникационное оборудование.

1.2.4. Интеграция сетевых ОС различных типов

Основная проблема, с которой сталкиваются пользователи, работающие в смешанной сети - необходимость установки разнообразных клиентов на рабочие станции и их взаимодействие между собой. Случается, что клиентское программное обеспечение для разных типов сетей, функционирующее по отдельности, отказывается работать будучи установленным на одном компьютере. Часто бывает, что для установки 2-3 разнообразных клиентов просто не хватает памяти. Поэтому пользователи предпочитают использовать интегрированные и согласованные между собой клиентские части, позволяющие не только облегчить доступ к различным серверам, но и унифицировать его. Под унификацией подразумевается единство команд работы с сетью и единство пользовательского интерфейса. Рассмотрим решение этой проблемы при интеграции систем, основанных на сетевых ОС Windows NT и Netware.

В сетях на основе Windows NT Server и Novell Netware используется принципиально разное построение служб управления каталогами. С точки зрения пользователя обе службы предоставляют возможность однократной регистрации в сети, что избавляет пользователя от необходимости регистрироваться дополнительно для доступа к ресурсам того или иного сервера. С точки зрения администратора появляется возможность централизованного управления сетью масштаба предприятия.

И NDS (служба каталогов Novell Netware) и NTDS имеют свои уникальные достоинства. Так в NDS удобнее перенести пользователя из одной части дерева в другую, удобнее просмотреть все ресурсы сети. С другой стороны, NTDS предоставляет администратору более гибкие возможности по установлению отношений между доменами. Можно сделать так, что домен будет иметь либо полную информацию обо всех других доменах, либо только о некоторых, либо вообще никакой.

Windows 95 может быть клиентом как сети Microsoft Windows, так и Novell Netware. С точки зрения пользователя для клиента предоставляется прозрачный доступ как к ресурсам доменов Windows NT, так и к серверам Netware и дереву NDS. Открыв папку Сетевое окружение пользователь увидит все домены сети Microsoft и серверы Netware (при их наличии). Открыв папку, соответствующую файловому тому, пользователь увидит структуру каталогов этого тома и файлы, расположенные в них. В заголовке окна отражается имя контекста и имя тома, при необходимости можно подключить любой из этих каталогов в качестве сетевого диска.

Из сказанного следует, что для пользователя принципиально не существует разницы в способе доступа к сетевым ресурсам. Независимо от того, является ли этот ресурс каталогом, предоставленным в совместное использование сервером Windows NT, каталогом на сервере Netware или объектом в NDS, доступ к нему однотипен.

Windows NT Server и Windows NT Workstation имеют развитые средства взаимодействия с Novell Netware. Они позволяют как пользователям сетей Windows, так и пользователям сетей Netware осуществлять прозрачный доступ к серверам в смешанной сети, а также централизованно управлять пользователями. Часть этих возможностей встроена в NT Server, а часть доступна в виде отдельных продуктов. Client Services for Netware, входящий в Windows NT Workstation позволяет осуществлять взаимодействие с сервером Novell NetWare.

Windows NT Server можно включить в уже существующую сетевую среду с NetWare и использовать в качестве сервера приложений для обеспечения работы баз данных, электронной почты, связи между хост-машинами или

дистрибуции программного обеспечения. Начиная с четвертой версии Windows NT поддерживается NDS. При установке нового компьютера с Windows NT в сети Netware система предложит зарегистрироваться либо на сервере Netware, либо в определенном контексте NDS.

Пользователи Windows NT Workstation получают прозрачный доступ на серверы Netware с помощью Gateway Services for Netware. Windows NT Server также может выступать в качестве шлюза к файлам и принтеру. Этот **шлюз** (средство, обеспечивающее связь между локальными сетями) позволяет клиентам осуществлять доступ к файлам и принтерам на сервере NetWare даже в том случае, если на них не установлен NetWare клиент.

Использование Gateway Services незаменимо в том случае, когда клиент сети Windows не может по каким-либо причинам установить дополнительный протокол или программное обеспечение. Шлюзование также удобно использовать при периодической работе большого числа клиентов с серверами Netware.

Создавая шлюз, администратор сети может разрешить произвольное число одновременных подключений к Netware серверу через шлюз. При этом, независимо от того, на сколько пользователей рассчитан сервер Netware, физически доступ к нему смогут одновременно иметь столько пользователей, сколько указал администратор. Пользователи сети Windows NT осуществляют доступ к ресурсам сервера Netware как к ресурсам NT сервера порой даже и не догадываясь о том, что работают через шлюз.

Какими же правами будут обладать пользователи, получившие доступ к ресурсам Netware-сервера через шлюз? По умолчанию они имеют все права в пределах тех, которые определены для бюджета, используемого самим шлюзом. Однако администратор вправе установить различные привилегии для пользователей.

Диаметрально противоположную задачу решает служба File & Print Services. Она позволяет предоставить прозрачный доступ клиентам Netware к серверу NT. При этом на самих клиентах не требуется делать никаких изменений. Все изменения выполняются только на сервере, после чего он начинает имитировать сервер Netware. Администратор может монтировать тома, создавать очереди печати, управлять привилегиями пользователей. Для этого используются те же самые утилиты, что и для администрирования Windows NT Server. Для администратора сети Netware этот сервер будет управляем с помощью традиционных средств управления серверами Netware. Результаты тестирования показывают, что на имитируемом сервере имеет место снижение производительности, однако оно таково, что на практике пользователь это практически не заметит.

По умолчанию такой сервер предоставляет одновременный доступ практически неограниченному числу пользователей (до 10000).

Работа в гетерогенной (смешанной) сети, включающей в себя домены Windows NT и серверы Netware, сопряжена с тем неудобством, что на каждом из серверов Netware имеется своя база пользователей и свои ресурсы, администрирование которых возможно только на этом сервере. Поэтому пользователям сети Microsoft придется дополнительно регистрироваться на каждом из серверов Netware для доступа к его ресурсам (в этом плане они ничем не отличаются от обычных клиентов Netware). Для устранения такого неудобства существует дополнительная утилита, называемая *Directory Service Manager for Netware*, позволяющая синхронизировать пользователей между доменом Windows NT и серверами Netware.

Эта утилита позволяет делегировать группы пользователей домена на произвольно выбранные серверы Netware, обеспечивая тем самым единство пользователей и их профилей. Для администратора Windows NT Server также

предоставляется возможность разрешить каждому конкретному пользователю осуществлять доступ к серверам Netware. Каждому из пользователей можно задать свой сценарий регистрации, а также указать иные параметры обеспечения совместимости с Netware.

1.3. Средства автоматизации проектирования АИС – языки 4GL и CASE-системы

1.3.1. Концепция и основные понятия

К середине 1990-х годов в области информационных технологий четко обозначились следующие факторы:

- развитие электроники значительно увеличило ресурсы ЭВМ, которые можно было использовать для автоматизации проектирования, с тем чтобы уменьшить его трудоемкость и длительность;

- сформировался объектно-ориентированный подход, хорошо структурирующий задачу, как таковую, а также и ее решение в виде прикладной системы;

- на базе объектно-ориентированного подхода стали развиваться визуальные средства быстрой разработки приложений (RAD — Rapid Application Development), основанные на компонентной архитектуре;

- стандартизовались основные методы работы с базами данных;

- все большее количество АИС стало принимать распределенный характер благодаря развитию телекоммуникационных сетей и концепции открытых систем.

Эти и другие факторы явились предпосылками для реальных изменений объектов проектирования - программных средств и баз данных АИС, и в частности возможность их переноса на различные операционные и аппаратные платформы. Все большее значение и массовость приобретают разработки сложных АИС, возросли объемы их программного и информационного обеспечения и требования к их качеству и надежности. Многие АИС стали создаваться и развиваться в виде ряда версий, которые последовательно модифицируясь и совершенствуясь эксплуатируются длительное время. Это вызывает увеличение трудоемкости и длительности проектирования, которые превышают все допустимые пределы. В то же время, происходит накопление проверенных на практике высококачественных компонент, которые можно многократно использовать в различных версиях программного обеспечения АИС и на различных платформах.

Изменились направления наиболее активно развиваемых методов в технологическом процессе разработки программ. Раньше основное внимание уделялось методам, языкам и средствам автоматизации непосредственного программирования (кодирования), относительно небольших программ. В настоящее время основные усилия перенесены на совершенствование и автоматизацию методов системного и структурного проектирования АИС на начальных этапах разработки, а также методов комплексной отладки, развития, переноса и сопровождения программного и информационного обеспечения. В результате методы непосредственного программирования (кодирования) на языках второго и третьего поколений отошли на второй план. Их вытесняют методы и средства автоматизации проектирования сложных программных комплексов на языках четвертого поколения (4 GL).

Основная цель развития современных технологий проектирования АИС состоит в повышении экономической эффективности всего жизненного цикла в различных проблемно-ориентированных областях. Это достигается снижением трудоемкости, ускорением и упрощением проектирования всей

Со совокупность методов и инструментальных средств автоматизации технологического процесса разработки сложных АИС объединяется под названием **CASE (Computer Aided Software Engineering - автоматизированное проектирование программных средств)**.

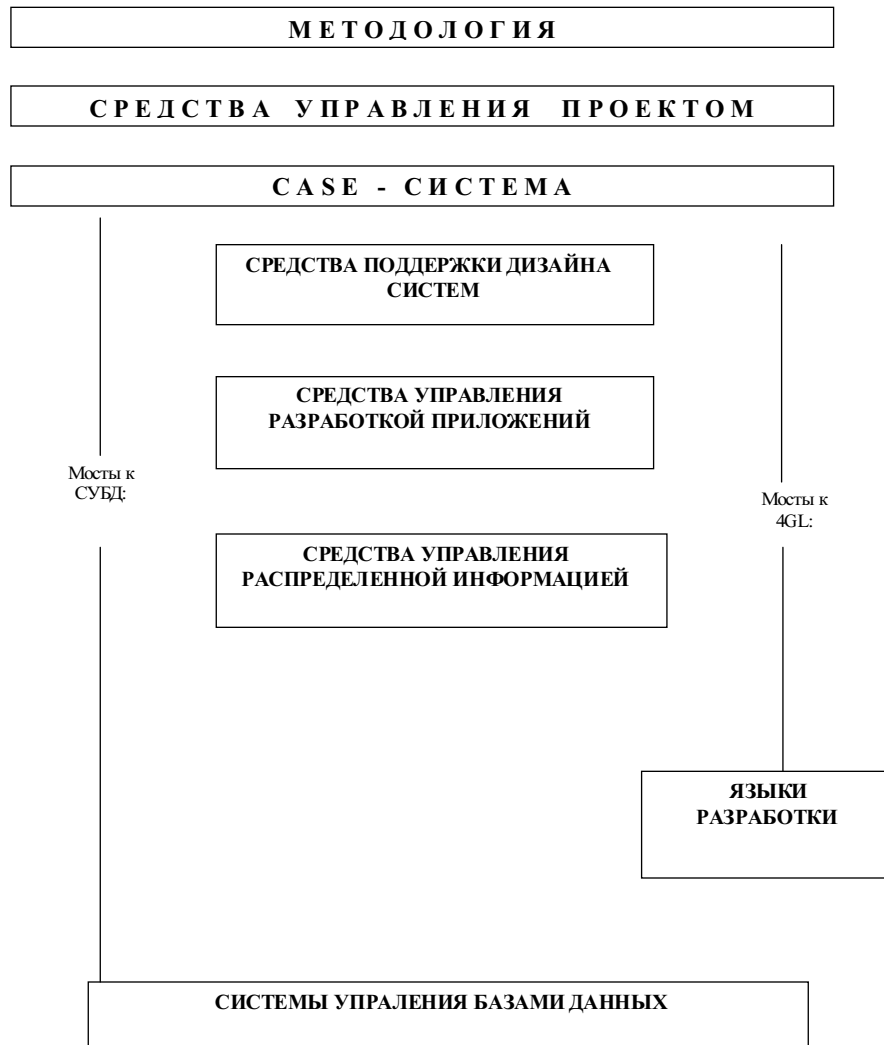


Рис. 4. Концепция разработки АИС с помощью CASE-системы

CASE-технология регламентирует (см. рис. 4) порядок организации и проведения работ, неавтоматизированного и автоматизированного выполнения технологических операций, направленных на получение в имеющихся организационно-технических условиях готовой АИС с заданными функциями и качеством. Для достижения этой цели разработан и активно используется ряд принципов:

- максимально возможное повышение уровня абстракции при описании компонент и действий над ними на различных этапах проектирования;

- сокрытие всей информации, избыточной для данного этапа или объекта проектирования;

- модульность и строгая иерархия в структурном построении программных и информационных компонент;

- унификация правил проектирования, структурного построения и взаимодействия компонент между собой и с внешней средой;

- поэтапный контроль полноты и качества решения функциональных задач.

Особенности предметной области отражаются в специальных алгоритмах прикладной части системы. Эффективность создания и функционирования прикладного ПО АИС определяется качеством ОС, СУБД и CASE-средств. Возможность использования готовых решений, отработанных в ранних версиях продукта или разработках других фирм, определяется степенью поддержки протоколов и интерфейсов с пользователями, с прикладными программами и стандартной документацией.

Интегрированные CASE-средства служат для извлечения знаний из заказчика на этапе проведения обследования, а также для проектирования концептуальной и логической структур баз данных прикладной системы. Кроме того, с их помощью осуществляется сопровождение и развитие проекта (рис. 5). Организация, внедряющая у себя CASE-технологии, должна осуществить определенные затраты на внедрение соответствующей методологии и средств и на обучение персонала. **Основными функциями CASE-средств** являются:

- объектно-ориентированное системное и логическое проектирование программных средств и баз данных;

- планирование и оценка затрат ресурсов на разработку программных средств и баз данных;

- стратегическое планирование и управление проектами на всем жизненном цикле;

- анализ требований, структурное проектирование ПС и БД, разработка и применение спецификаций требований;

- организация и управление базами данных и хранилищами проектов;

- повторное использование отработанных программных компонент, а также перенос их на иные операционные и аппаратные платформы.

Языки четвертого поколения (4GL) являются средством разработки приложений. Они легки в изучении и намного уменьшают размер кода, требуемого программистам для проектирования, записи и тестирования приложений. В результате достигаются значительно меньшие программистские затраты и возросшая более чем в 10 раз производительность, по сравнению с языками низкого уровня. 4GL также облегчают программистам следующие задачи:

- поддержку и модификацию программных приложений;

- написание более строгой программной документации;

- осуществление быстрого макетирования приложений;

- управление проектированием программных средств.

Многие 4GL предоставляют возможность лицам, не являющимся профессиональными программистами, решать самостоятельно некоторые

задачи, например, написание запросов и отчетов БД. На уровне 4GL сообщается приложению, что следует сделать, 4GL автоматически решает, как это сделать.

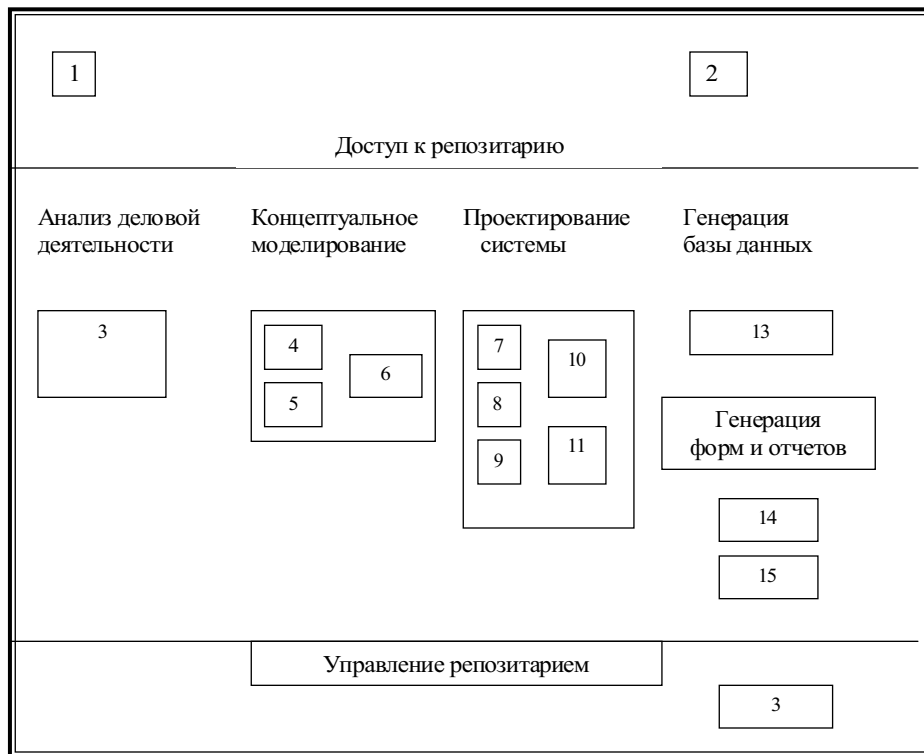


Рис. 5. Компонентная структура CASE-системы: 1-навигатор объектов; 2-матричный диаграммер; 3-средства администрирования; 4-средства моделирования процессов; 5-диаграммер ER-моделей; 6-диаграммер иерархии функций; 7-диаграммер потоков данных; 8-диаграммер структуры приложения; 9-навигатор параметров; 10-навигатор процедурной логики; 11-диаграммер баз данных; 12-диаграммер модулей; 13-генератор сервера; 14-генератор форм; 15-генератор отчетов

Операция, требующая сотни команд кода 3GL, может быть выполнена с помощью нескольких простых предложений на 4GL. Эти предложения затем транслируются в соответствующий код низкого уровня. Под термином 4GL подразумеваются не только конкретные языки программирования, но и множество средств для разработки программного обеспечения, поставляемое вместе с языками. К этим средствам относятся:

- отладчики;
- редакторы текстов и баз данных;
- генераторы приложений и меню;
- средства создания форм и отчетов;

средства форматирования экрана;
системы презентации.

Компонентная структура CASE-системы представлена на рис. 5. Под диаграммерами здесь понимаются средства построения диаграмм, а под навигитарами средства поиска. Эта схема необходима и при изучении следующих подразделов раздела 1.3.

1.3.2. Объектно-ориентированный инструмент для профессиональной разработки приложений в среде клиент/сервер

PowerBuilder (PB) - это объектно-ориентированный инструмент для профессиональной разработки приложений в среде клиент/сервер, позволяющий коллективам разработчиков создавать приложения, которые имеют доступ к базам данных и другой корпоративной информации, хранящейся локально или на сетевых серверах. PB входит в состав инструментальных CASE-средств для разработки масштабируемых приложений в среде клиент/сервер, которые могут быть использованы различными категориями пользователей организации - от разработчиков сложных корпоративных информационных систем до разработчиков на уровне отделов и конечных пользователей.

В состав системы входят следующие компоненты:

PowerBuilder Enterprise, который предназначен для создания сложных многоплатформенных приложений клиент/сервер коллективами профессиональных разработчиков;

PowerBuilder Team/ODBC обеспечивает возможность коллективной разработки и работает с серверами баз данных через ODBC;

PowerBuilder Desktop предназначается индивидуальным разработчикам, создающим автономные приложения под Windows;

Advanced Developer Toolkit включает библиотеку многократно используемых объектов, а также развитые инструментальные средства, такие как редактор изображений и др.

В систему также включен InfoMaker - персональный инструмент разработки в среде клиент/сервер, который позволяет конечным пользователям создавать запросы, формы, отчеты и деловую графику. Пользователи могут манипулировать данными, применяя подход, основанный на формах и не требующий программирования.

Так как все продукты основаны на общей объектной технологии, пользователи могут создавать приложения и передавать их в любое время менеджерам для продолжения разработки, поддержки или сопровождения. Таким образом, разработчики и конечные пользователи получают инструменты, которые позволяют использовать преимущества технологии клиент/сервер в рамках всей организации.

PB поддерживает широкий спектр систем управления реляционными базами данных и полностью использует специфические особенности каждой из них. Разработчики могут использовать встроенную высокопроизводительную реляционную базу данных для создания автономных приложений, а также для обеспечения работы приложений вне сервера. Интеллектуальный объект Окно данных позволяет манипулировать данными из реляционных баз данных без программирования. С помощью Окна данных можно извлекать, обновлять, добавлять, удалять, просматривать, печатать и сохранять данные в любом из 10 форматов файлов. Окно данных непосредственно управляет взаимодействием и манипуляциями с базой данных.

Окно данных упрощает также создание отчетов. PB позволяет создавать широкий спектр деловых отчетов в режиме "укажи и щелкни кнопкой мыши".

Сюда относятся сложные ленточные таблицы, отчеты свободного формата, связанные таблицы, метки, многоколоночные отчеты с многоуровневыми группировками и сортировкой, а также определенные пользователем вычисляемые поля, столбцы и итоговые суммы. Стандартная двухпроходная генерация отчетов позволяет вычислять средние значения, процентные отношения и постраничные суммы. Окна данных также предоставляют возможности встроенной деловой графики для комбинирования текстовой и графической информации.

Кроме того имеются интерактивные средства для создания баз данных **SQL** (язык структурированных запросов, тип реляционной СУБД) и манипулирования ими, избавляющие от необходимости изучения и использования SQL. Разработчики могут создавать таблицы и представления, определять первичные и внешние ключи, запускать командные файлы баз данных, обеспечивать безопасность и редактировать данные из базы данных - и все это в одной интегрированной среде.

В PB используется практический подход к объектной технологии, позволяющий разработчикам информационных систем осуществить быстрый переход к объектно-ориентированной разработке без необходимости знать и использовать специфические языки программирования. Он полностью поддерживает наследование, инкапсуляцию и полиморфизм.

Приложение, созданное с помощью PB, является композицией ряда объектов, таких как окна, меню, функции, структуры и Окна данных. Объекты, выполняющие общие функции, такие как Кнопка печати, могут многократно использоваться в разных приложениях, реально сокращая время разработки, а также повышая продуктивность программистов и качество программ.

Система включает графическую среду для создания определенных пользователем объектов, событий и функций, которая значительно упрощает повторное использование кода и делает более удобным сопровождение. Поддержка многоуровневого наследования облегчает разработку и сопровождение библиотек объектных классов.

Графический подход PowerBuilder к разработке поддерживает большие коллективы разработчиков информационных систем с помощью менеджера общей библиотеки объектов и центрального репозитория дизайна приложений.

Репозиторий представляет собой объектно-ориентированную базу данных. Средства просмотра обеспечивают "навигацию" по проекту, в том числе, перемещение вверх/вниз по иерархиям классов и подсистем, переключение из одного вида диаграмм к другому и т. д. Средства контроля и сбора статистики дают возможность находить и устранять ошибки по мере развития проекта, а не после завершения его описания. Генератор отчетов формирует тексты выходных документов на основе содержащейся в репозитории информации.

Менеджер библиотеки осуществляет проверку для предотвращения одновременного обновления одного объекта несколькими разработчиками, предоставляет возможности поиска в библиотеках, анализирует взаимосвязи, а также создает подробные отчеты для разработчиков по библиотекам и их компонентам. Центральный репозиторий дизайна приложений доступен всему коллективу разработчиков и позволяет им определять расширенные атрибуты таблиц и столбцов, стандартизировать и ускорять процесс разработки приложений.

PB - открытая среда разработки, включающая интерфейсы со многими представителями технологии программного обеспечения в среде клиент/сервер. Средства CASE, системы контроля версий, инструменты соединения узлов, мультимедиа, обработка образов, перьевого ввод, и многие другие технологии полностью интегрируются при помощи открытого интерфейса API.

Все компоненты приложения, такие как окна, меню, логика бизнеса, доступ

к базам данных, создание баз данных, графика и отчеты, можно разрабатывать полностью в рамках PowerBuilder, при этом нет необходимости постоянно покидать среду и возвращаться в нее для выполнения каких-то операций.

Полностью поддерживается MS Windows, включая все сообщения Windows, элементы управления, многооконные приложения, связывание и встраивание объектов OLE, динамический обмен данными и вызовы динамически связываемых библиотек DLL для интеграции с существующими приложениями. Графический интерфейс пользователя GUI может быть создан разработчиком приложения без программирования на низком уровне (например, на языке C).

PowerBuilder содержит PowerScript - язык управления данными 4GL, позволяющий разработчику включать простую и сложную деловую логику в приложения. Этот язык состоит более чем из 100 функций для манипулирования объектами, числами и текстом, функций обработки дат и времени, функций ввода/вывода, а также функций для полной поддержки OLE как в качестве клиента, так и в качестве сервера. Инструмент, входящий в состав PowerBuilder - Художник функций, позволяет разработчику расширять командный язык, добавляя к нему определяемые пользователем функции. Внешние функции можно декларировать, после чего они становятся доступными в приложениях PowerBuilder так же, как и встроенные функции, что позволяет взаимодействовать с внешними процедурами, которые работают на сервере или клиенте.

Инструментальная среда разработки PowerBuilder состоит из ряда интегрированных графических инструментов - **художников**, позволяющих коллективу разработчиков проектировать, создавать, интерактивно тестировать и применять приложения клиент/сервер в режиме "укажи и щелкни кнопкой мыши". Среда разработки может быть расширена для организации непосредственного доступа к набору инструментальных средств разработки других фирм. Интерфейс пользователя позволяет разработчикам открывать окна нескольких художников одновременно, что дает возможность мгновенно получить доступ к нужному режиму работы.

Художник приложений (Application Painter) определяет среду приложения, включая имя приложения и его пиктограмму, установленные по умолчанию цвета и шрифты, программы обработки событий приложений, список библиотек, а также позволяет разработчику графически просматривать всю иерархию объектов приложения. Как только все объекты приложения будут созданы, откомпилированы и протестированы, Художник приложений создаст исполняемый файл (EXE).

Художник окон (Window Painter) создает окна (являющиеся основным элементом интерфейса приложений) и элементы управления, которые могут быть частью окна. **Элементами управления** являются командные кнопки, линейки прокрутки, списки выбора, открывающиеся списки выбора, альтернативы, элементы управления окна данных и многое другое.

Художник меню (Menu Painter) создает меню и линейки инструментов, которые можно связать с любым окном как в процессе определения окна, так и динамически из программы, в ходе ее выполнения.

Художник окна данных (DataWindow Painter) создает интеллектуальный объект данных для просмотра, манипулирования и обновления реляционных баз данных без необходимости программирования. Художник окна данных содержит множество параметров, предоставляемых в режиме "укажи и щелкни кнопкой мыши", которые позволяют изменять оформление внешнего вида окна данных. Не прибегая к программированию, разработчики могут создавать деловую графику, вычисляемые столбцы, автоматические сводки, а также осуществлять межтабличный анализ. Окна данных могут быть динамическими

и поддерживать изменения во время выполнения приложения в виде незапланированных запросов и отчетов, чтобы удовлетворить специфическим требованиям конечных пользователей.

Художник структур (Structure Painter) создает сложные структуры данных для использования их в конструкциях языка PowerScript и для связи с внешними функциями. Структуры помогают разработчику организовать переменные в программах и облегчают взаимодействие с внешними функциями.

Художник настроек (Preferences Painter) позволяет настраивать конфигурацию PowerBuilder в соответствии с привычками и вкусом разработчика.

Художник баз данных (DataBase Painter) предоставляет интерактивные средства для создания и поддержки баз данных SQL. Разработчики могут создавать таблицы и представления, определять первичные и внешние ключи, запускать командные файлы баз данных, обеспечивать безопасность и редактировать данные из базы данных.

Центральный репозиторий дизайна приложений (Central Application Design Repository) предоставляет развитые возможности управления хранением расширенных атрибутов столбцов, таких как заголовки, метки, форматы изображения, правила проверки и графические стили редактирования.

Художник переноса данных (Data Pipeline Painter) обеспечивает разработчикам доступ к данным и перенос данных без программирования, даже если они находятся в различных базах данных. Определения Data Pipeline могут быть сохранены в виде объектов и затем использованы в приложениях.

Художник запросов (Query Painter) предоставляет графические средства для создания запросов к базам данных и сохранения этих запросов в виде объектов. Объекты запросов можно использовать в качестве источника информации для Окна данных и отчетов, обеспечивая еще более тесную интеграцию с базами данных. Художник запросов полностью поддерживает графическое рисование SQL как для таблиц, так и для представлений.

Художник функций (Function Painter) позволяет разработчику создавать и поддерживать функции, которые чаще всего представляют собой общие процедуры, неоднократно используемые в одном или нескольких приложениях, сокращая при этом объем повторяющегося кода.

Художник проектов (Project Painter) позволяет указывать порядок и параметры сборки различных приложений из общего набора объектов в библиотеках PowerBuilder. Определения проектов сохраняются в библиотеках и позволяют автоматизировать большинство действий по подготовке готового приложения.

Художник библиотек (Library Painter) позволяет разработчикам совместно использовать все объекты приложений, такие как окна, окна данных, меню, структуры и объекты пользователей. Эти объекты хранятся в одной или нескольких библиотеках, локализованных в одном месте или распределенных в сети. Возможность проверки при выдаче и возврате объектов позволяет предотвратить одновременное обновление одного объекта несколькими разработчиками. Предоставляются также возможности поиска в библиотеках, анализа взаимосвязи и создания подробных отчетов для разработчиков по библиотекам и их компонентам. Художник библиотек предлагает также прозрачный доступ к множеству систем контроля версий и систем управления конфигурацией, разработанных третьими фирмами. Разработчики, создающие крупные приложения, могут поддерживать несколько версий объектов и отслеживать, кто и когда внес данное изменение.

Художник объектов пользователей (User Object Painter) создает объекты, определенные пользователями и невидимые объекты пользователя,

обеспечивая полное использование объектно-ориентированной техники разработки.

Отладчик выполняет приложение в режиме отладки для поиска ошибок в работе программ и объектов. Отладчик позволяет разработчику устанавливать прерывания, осуществлять выполнение по шагам, просматривать и изменять значения переменных, а также сохранять среду сеанса отладки для использования в будущем.

Возможности художника баз данных (DataBase Painter) обеспечивают тесную интеграцию с широким спектром поддерживаемых баз данных с полным использованием специфических особенностей каждой из них. Он предоставляет доступ к художнику администрирования данных (Data Administration Painter) - интерактивному блокноту для записи и графического представления операторов SQL, которые затем немедленно выполняются СУБД. Художник администрирования баз данных позволяет создавать, удалять и модифицировать пользователей системы управления базами данных, а также указывать привилегии и ограничения доступа в соответствии с возможностями управления доступом выбранной СУБД.

Художник манипулирования данными (Data Manipulation Painter) позволяет предварительно просматривать существующие данные, заполнять новые таблицы, а также тестировать форматы изображений, правила проверки и стили редактирования на реальных данных.

Драйверы для доступа к различным серверам баз данных позволяют организовать связь PowerBuilder с базой данных как на этапе разработки, так и на этапе эксплуатации приложения, что дает возможность использовать все специфические особенности каждой базы данных, включая хранимые процедуры, триггеры, прокручиваемые курсоры, правила проверки ссылочной целостности и т.д. для баз данных, которые предоставляют эти возможности.

PowerBuilder интегрирован с базой данных, в которой хранится репозиторий дизайна приложений. Это повышает производительность разработчика, облегчает создание стандартов приложений и заставляет следовать этим стандартам во всей организации. После определения репозитория, значительно ускоряется создание сложных Окон данных (DataWindows) и отчетов. Централизованное определение репозитория дизайна приложений ускоряет и упрощает поддержку и обновление существующих приложений. Разработчики могут определить в репозитории стандарт создания целостного интерфейса пользователя.

Один из подходов к созданию новых приложений PowerBuilder включает следующие шаги:

- создание объекта приложения (художник приложений);
- проектирование графического интерфейса пользователя (художники окон, меню и объектов пользователя);
- определение поведения объектов (PowerScript);
- определение режимов работы с данными (художник Окон данных);
- генерация отчетов (художник отчетов);
- добавление подсказок в приложение (PowerScript);
- управление приложением (художник библиотек);
- отладка приложения (отладчик);
- поставка законченного приложения (художник приложений).

Разработка в среде PowerBuilder базируется на создании **объектов, элементов управления и событий**. Приложения PowerBuilder состоят из объектов (таких как окна, окна данных, меню и объекты пользователя) и элементов управления (таких как командные кнопки и радиокнопки).

Каждый объект и каждый элемент управления PowerBuilder имеет набор атрибутов, описывающих его размер, расположение, последовательность

размещения и текущее состояние (видимый, разрешенный). Разработчики определяют атрибуты, а также поведение объектов и элементов управления в соответствии с конструкцией приложения.

Объекты, созданные в процессе разработки приложения, хранятся в библиотеках. Разработчики могут использовать в приложении объекты из одной или нескольких библиотек. Обычно каждый член коллектива разработчиков имеет свою собственную тестовую библиотеку, а также использует разделяемые библиотеки общих объектов, хранящиеся на сетевых файловых серверах.

PowerBuilder полностью поддерживает многоуровневое наследование, инкапсуляцию и полиморфизм. Для разработчиков АИС самым значительным преимуществом объектной технологии является рост производительности благодаря многократному использованию кода. Наследование позволяет усложнить объект-предок и автоматически передавать добавленные характеристики всем его потомкам. Наследование упрощает также поддержку приложений и обеспечивает непротиворечивость.

Инкапсуляция позволяет хранить объект вместе со всеми его функциями или программами в виде инкапсулированного объекта. Инкапсуляция автоматически обеспечивается для любого созданного в PowerBuilder объекта, включая те, которые помещаются внутри других объектов (например, командная кнопка, расположенная в окне).

Централизованная система управления библиотеками объектов PowerBuilder представляет собой открытую среду для групповой разработки приложений и управления проектом. Предоставляются прямые связи из среды PowerBuilder к другим системам управления объектами и контроля версий для управления разработкой объемных приложений.

1.3.3. Разработка проектов распределенных АИС с использованием комплекса средств

В настоящее время, при разработке сложных распределенных АИС обычно используют комплексы инструментальных средств различных производителей. Это связано с тем, что CASE-средства различных производителей взаимно дополняют друг друга. Так, например относительно недорогие и простые в освоении системы могут иметь следующие типичные недостатки:

- недостаточно развитые возможности импорта/экспорта, следовательно, невозможна коллективная разработка модели;
- нет репозитория структур данных;
- сильно ограничены допустимые (возможные) структуры данных;
- отсутствуют средства конструирования базы данных и др.

Рассмотрим на примере, использование комплекса инструментальных средств при разработке распределенной финансовой АИС.

Технические требования к АИС – следующие:

- переносимость приложений на достаточно широкий круг аппаратно-программных платформ и реляционных СУБД; Адаптивность к изменениям административной структуры;

- наращивание выполняемых функций;

- быстрая настройка АИС на конкретных пользователей;

- возможность получать информационные услуги с любых АРМов, в пределах их полномочий;

- интеграция входной, внутренней и выходной информации.

Основные функции АИС – следующие:

- составление отчетов по оборотам финансовых средств;

- составление и анализ выполнения плана оборотов;
- контроль за выдачей финансовых средств;
- анализ просроченной задолженности;
- анализ состояния и объема денежной массы в обороте;
- планирование, учет и анализ эмиссионно-кассовых операций;
- администрирование работы АИС.

В инструментальный комплекс включены: CASE-система, язык 4GL для программирования сервера приложений, сетевая ОС, реляционная СУБД.

CASE-система предоставляет следующие возможности:

- компонента DFD (модуль построения диаграмм) является инструментом создания схем, от диаграмм потоков данных и бизнес-моделей до схем навигации экранов и взаимосвязей функциональных модулей программ;

- компонента ERX (модуль концептуального моделирования) помогает построить концептуальную модель данных проекта;

- компонента RDM (модуль реляционного моделирования) предназначена для построения реляционной модели данных проекта, позволяет создавать базу данных, быстро переносить базу с одной платформы на другую;

- компонента WRM (менеджер репозитория) управляет общим для всех компонент репозиторием для хранения и совместного использования типов данных, доменов, структур данных; имеет возможности формирования отчетов на основе данных и схем проекта.

В процессе **первого этапа** проектирования АИС была разработана концептуальная модель. Диаграммы потоков данных информационно-функциональной модели предметной области АИС были переведены в среду CASE-системы. Построена система диаграмм информационных потоков для этих подсистем с помощью DFD-компоненты. В модели выделено 84 процесса, 9 внешних сущностей, 24 накопителя данных. Определены: 26 базисных типов данных, 26 доменов, 54 структуры данных, информационные потоки связаны со структурами данных, построены описания всех детальных процессов. С помощью средств документирования проекта были получены описания компонент диаграмм и потоков данных.

Данные, полученные в процессе разработки DFD-модели (модели диаграмм потоков данных), были перенесены в репозиторий для их дальнейшего использования при построении реляционной модели данных проекта с помощью RDM-компоненты. На основе базисных типов, доменов и структур данных, определенных в DFD-диаграммах определены:

- 9 основных таблиц;

- 45 подсхем с подчиненными таблицами;

- уникальные ключи и отношения между таблицами;

- а также:

- проведена верификация модели;

- сгенерированы внешние ключи подчиненных таблиц;

- сформирован SQL-код для СУБД Oracle;

- проведен перенос модели на платформу СУБД Informix;

- получена документация по сформированной базе.

На следующем этапе применялась среда разработки приложений (4GL) JAM. Созданный на языке JAM комплекс программ включает в себя АРМ администратора АИС и универсальную программную оболочку для конфигурирования АРМов конкретных пользователей. Разработаны схемы навигации экранов:

- универсальной оболочки конфигурирования АРМов;

- отдельных функциональных модулей;

- оболочки администратора;

- действующего макета подсистемы финансового обращения.

Язык четвертого поколения JAM включает в себя следующие основные компоненты:

- экранный редактор - среду разработки экранов;
- редактор меню;
- отладчик;
- генератор отчетов;
- макетную СУБД JDB;
- модули интерфейса с конкретными СУБД (ORACLE, INFORMIX);
- средство программирования сервисов.

Для хранения описаний информационных объектов и реализации механизма наследования используется репозиторий. Помимо средств визуального программирования JAM имеет встроенный процедурный язык интерпретирующего типа JPL с С-образным синтаксисом. JPL - это процедурный язык интерпретирующего типа, встроенный в среду разработки JAM. На языке JPL реализуется логика работы приложений.

Разработка программного обеспечения началась с переноса функционально-логической модели из CASE-системы в среду визуальной разработки прикладного программного обеспечения JAM. На этой стадии были выполнены следующие работы.

Перенос схемы базы данных для СУБД ORACLE в макетную СУБД JDB. JDB-это встроенная в среду разработки JAM СУБД с SQL языком, которая позволяет выполнять прототипирование и тестирование приложений, реализованных на JAM до создания реальной базы данных в СУБД ORACLE или INFORMIX.

Формирование репозитория АИС в среде JAM. Репозиторий первоначально формируется из схемы базы данных с помощью операции IMPORT. При этом объекты, которые хранятся в репозитории наследуют свойства объектов из базы данных. Затем производится редактирование входов репозитория (вводятся русские наименования данных, шаблоны внешнего представления данных, объекты управления приложениями и т.д.;

Затем были выполнены следующие подготовительные работы:

- сформулированы требования к внешнему оформлению пользовательского интерфейса;

- разработана библиотека типовых программных модулей на языке JPL.

Для обеспечения максимальной гибкости и адаптивности АИС был реализован подход компоновки конкретных АРМов из набора функциональных модулей при помощи универсальных средств компоновки и управления АРМами. Под функциональным модулем понимается группа программных модулей или один модуль, которые выполняют законченную прикладную функцию. Программные модули - это экраны или отчеты JAM. На языке JAM были разработаны АРМ администратора АИС и универсальная программная оболочка для реализации АРМов АИС, включающие в себя средства:

- регистрации пользователей;
- описания типовых программных комплексов;
- конфигурирования АРМов конкретных пользователей;

С помощью АРМа администратора АИС стала возможной практическая реализация концепции рабочего стола пользователя, когда на рабочем месте пользователя под единым управлением интегрируются как конкретные прикладные АРМы, предназначенные для данного пользователя, так и набор типовых средств, таких как текстовые редакторы, электронные таблицы, электронная почта и т.д.

Исходными данными для разработки приложений были функционально-логическая модель, модель базы данных и схема навигации приложений, разработанные в среде CASE-системы.

Прикладное программное обеспечение разрабатывалось средствами JAM. Важным преимуществом данного инструментального комплекса является возможность проведения разработки с построением прототипов.

Первым действующим прототипом системы являлось описание АРМов средствами АРМ администратора, в результате чего пользователь видел функции АРМов в виде меню верхнего уровня до начала программирования. Затем средствами JAM осуществлялось прототипирование экранов.

Коллективная разработка осуществлялась с использованием общего репозитория проекта, на базе которого осуществлялось управление изменениями прикладных экранов, а также контроль состояния разработки. При этом использовался подход, при котором приложение вначале прототипировалось и тестировалось, а затем путем анализа SQL-кода определялись сервисы и соответствующие им сервисные экраны, а также параметры сервисов. Затем вызов SQL в клиентских экранах заменялся на вызов соответствующих сервисов, а из исходного экрана формировались сервисные экраны. По окончании данной работы сервисные экраны и файл описания сервисов переносились на сервисную машину.

В ходе разработки было запрограммировано более 160 экранов и отчетов и на языке JAM и средствами АРМа администратора АИС сконфигурированы 8 прикладных АРМов.

Разработка проводилась 3,5 месяца коллективом в 3 человека и завершилась созданием АИС и комплекта документации из 14 книг общим объемом 450 страниц. Разработки проводились в среде Solaris. По завершении разработки, функционировавшей в среде Solaris - Oracle, проведен перенос в среду UNIX - Informix, включая базу данных, наполненную реальной информацией. Объем базы составлял около 1,5 Мб. Общее время переноса составило менее часа.

Данный комплекс действительно обеспечивает разработку переносимых приложений. Все средства достаточно просты в освоении. Разработка приложений средствами JAM осуществлялась достаточно быстро, причем разработчики не имели большого опыта работы с ним. Использование механизма наследования позволяет значительно снизить трудоемкость внесения изменений при сопровождении.

1.3.4. Средства реинжиниринга

Практика разработки сложных программных комплексов для АИС показывает, что часто такие системы имеют предшественников в виде совокупности программ, которые частично или полностью реализуют требования к системе. Для их развития и дальнейшего сопровождения такой системы именно в силу ее сложности требуется создание совокупности моделей, представляющих в графическом виде разные аспекты системы и отражающих иерархию ее построения. В особенности это касается систем, предшественники которых в той или иной мере основаны на применении объектно-ориентированного подхода.

Для создания подобных моделей современные CASE-системы включают специальные средства **реинжиниринга**, позволяющие восстановить отдельные модели по исходным текстам программ (см. рис. 5).

Исходные предпосылки для проведения реинжиниринга состоят в следующем:

- существует реализованная и оттестированная информационная система;
- имеется перечень задач, которые решает система;
- есть тексты программ, решающих эти задачи, на некотором языке (языках) программирования.

Отдельно от системы может существовать документ, в котором заказчик описывает, что в системе должно быть добавлено или изменено. Построение модели системы включает решение следующих задач:

- описание архитектуры в виде иерархии категорий классов, соответствующей разным уровням абстракции;

- строится логическая модель в виде диаграммы классов, где показываются классы и отношения между ними;

- разрабатывается функциональная модель системы (описание поведения системы) в виде иерархии диаграмм сценариев;

- создается динамическая модель системы, которая описывается с помощью диаграмм состояний для управляющих классов;

- проводится реинжиниринг базы данных системы с построением ER-диаграмм.

Порядок решения этих задач принципиально отличается от порядка действий, выполняемых при разработке новых проектов. Архитектура и логическое описание системы могут быть автоматически получены с помощью реинжиниринга текстов программ и построения диаграмм классов CASE-системой. Таким образом, не функциональное описание системы служит основой для выявления классов и отношений между ними, а наоборот, предварительно полученные диаграммы классов наряду с программными кодами служат для описания поведения системы и динамики ее работы.

CASE – системы, которые используются для реинжиниринга кроме автоматизации этапов анализа и проектирования, генерации кодов на различных языках и выпуска проектной документации, включают анализатор - отдельный программный модуль, обеспечивающий реинжиниринг. Средства автоматической генерации кодов программ на языках третьего и четвертого поколений используя информацию, содержащуюся в логической и физической моделях проекта, формируют файлы заголовков и файлы описаний классов и объектов. Создаваемый таким образом скелет программы может быть уточнен путем прямого программирования.

Анализатор кодов создает модели проектов на основе информации, содержащейся в определяемых пользователем исходных текстах программ. В процессе работы анализатор осуществляет контроль правильности исходных текстов и диагностику ошибок. Модель, полученная в результате его работы, может целиком или фрагментарно использоваться в различных проектах. Это обеспечивает возможность повторного использования программных компонент.

В основе работы таких систем лежит построение различного рода диаграмм и спецификаций, определяющих логическую и физическую структуры модели, ее статические и динамические аспекты. **Диаграммы классов** показывают классы и их иерархии, отображая логику построения прикладной системы. Для больших систем обычно строится не одна а несколько подобных диаграмм. Диаграмма классов определяет только статические аспекты, относящиеся к классам. Динамика их поведения представляется на диаграммах состояний. **Диаграммы сценариев** показывают существующие объекты и их взаимодействие в логическом проекте прикладной системы. **Модульные диаграммы** определяют распределение классов и объектов в модулях, физически реализующих проект. Одна модульная диаграмма представляет всю или часть модульной архитектуры системы. Для решения задачи распределения аппаратных ресурсов используются **диаграммы процессов**.

Не вся необходимая информация о проекте может быть наглядно отражена в диаграммах, поэтому необходимы средства ввода спецификаций, которые дополняют набор диаграмм. Спецификации создаются для классов, класс-утилит, операций, подсистем, объектов, модулей и т. д.

Основные свойства систем обеспечивающих реинжиниринг следующие:

- охват всех этапов жизненного цикла работы над проектом с единой методикой и нотацией;
- возможность повторного использования программных разработок пользователей;
- наличие средств автоматического контроля, позволяющих вести отладку проекта по мере его разработки;
- возможность описания проекта на разных уровнях для различных категорий пользователей;
- удобный для пользователя графический интерфейс;
- автоматическая генерация кодов на наиболее популярных языках программирования;
- наличие средств групповой разработки;
- широкий спектр применения системы - базы данных, банковские системы, телекоммуникация, системы реального времени и т. д.;
- возможность использования различных платформ.

1.4. Основы объектных распределенных технологий – стандарты CORBA, DCOM, RMI

Клиент-серверные приложения, как правило выполняются в рамках открытых систем. **Открытые системы** – это АИС различного назначения, которые размещены в различных узлах телекоммуникационной сети. Работу открытых систем как единого интегрированного целого обеспечивает система отраслевых, государственных и международных стандартов в области информационных технологий, специфицирующих интерфейсы, услуги и поддерживающих форматы данных для достижения взаимодействия и переносимости приложений, данных и персонала. Такие АИС функционируют в условиях информационной и реализационной неоднородности.

Информационная неоднородность ресурсов заключается в разнообразии их предметных областей (используемых сущностей, понятий, словарей; отображаемых реальных объектов, абстракций в информационных системах; семантических правил, определяющих адекватность совокупностей моделируемых объектов реальности; моделируемых процессов; видов данных, способов их сбора и обработки; интерфейсов пользователей и т.д.).

Реализационная неоднородность источников проявляется в использовании разнообразных компьютерных платформ, средств управления базами данных, моделей данных и знаний, языков программирования, операционных систем, и т.п.

При разработке и развитии распределенных АИС объективно необходимо придерживаться соблюдения ряда условий.

1. Интеграция систем. Системы эволюционируют от простых, автономных подсистем к более сложным, интегрированным системам, основанным на взаимодействии разнородных компонентов.

2. Реинженерия систем. Создание системы и ее реконструкция (реинженерия) - непрерывный процесс формирования, уточнения требований и конструирования. Реконструкция систем осуществляется постепенно. Система должна быть спроектирована так, чтобы произвольные ее составляющие могли быть реконструированы при сохранении целостности системы.

3. Перенос унаследованных систем. Любая система после создания противодействует изменениям. Унаследованные системы часто используют

технологии, архитектуры, платформы, программное и информационное обеспечение, при проектировании которых не были предусмотрены нужные меры для их пошагового переноса в новые системы. Необходимо, чтобы переносимые составляющие системы и оставшиеся компоненты унаследованных систем сохраняли совместимость.

4. Повторное использование неоднородных информационных ресурсов. Технология разработки АИС должна обеспечивать повторное использование информационных ресурсов, переходя от технологии программирования, основанной на интенсивном индивидуальном труде по созданию вручную изделий, удовлетворяющих специфическим требованиям одного конкретного применения, к технологии, основанной на планируемых капиталовложениях в разработку повторно -используемых компонентов (ПИК). ПИКи должны обеспечивать производство серий стандартизованных продуктов в определенной прикладной области.

5. Продление жизненного цикла систем. В условиях исключительно быстрого технологического развития требуются специальные меры, обеспечивающие необходимую продолжительность жизненного цикла.

Сложность и объемы работ, необходимых для создания эффективных распределенных АИС требуют объединения усилий различных производителей программного обеспечения. Одним из таких объединений, которое вносит существенный вклад в принимаемые идеологические, архитектурные и технологические решения является OMG (Object Management Group) - крупнейший в мире консорциум разработки программного обеспечения, включающий свыше 700 членов - производителей программного продукта, разработчиков прикладных систем и конечных пользователей. Целью OMG является создание согласованной информационной архитектуры, опирающейся на теорию и практику объектных технологий и общедоступные спецификации интерфейсов информационных ресурсов.

Эта архитектура должна обеспечивать повторное использование и мобильность компонентов, опираясь на коммерческие продукты. В кооперации с OMG работают государственные и международные организации по стандартизации ПО, с целью доведения результатов OMG до официальных стандартов.

OMG на основе объектной технологии вводит концепцию промежуточного слоя последовательно и радикально. Концепция **промежуточного слоя** – это сосредоточение специальных сервисов в слое архитектуры, расположенном между операционной системой и средствами управления компьютерными сетями и прикладными системами, специфическими для конкретных областей применения. Традиционно к такому промежуточному слою относились средства управления и доступа к данным, средства разработки программ, средства управления распределенными вычислениями (включая поддержку необходимых протоколов взаимодействия), средства поддержки пользовательского интерфейса и др. Эта концепция использовалась в международных проектах создания UNIX - ориентированной интеграционной среды.

В 1990 году OMG разработала архитектуру OMA (Object Management Architecture), изначально предназначенную для обмена данными между приложениями. При этом, в частности, понадобился стандарт, определяющий взаимодействие составных частей приложения, или объектов. Этот стандарт получил наименование Общая Архитектура Брокера Объектных Заявок - CORBA (Common Object Request Broker Architecture) (см. рис. 6).

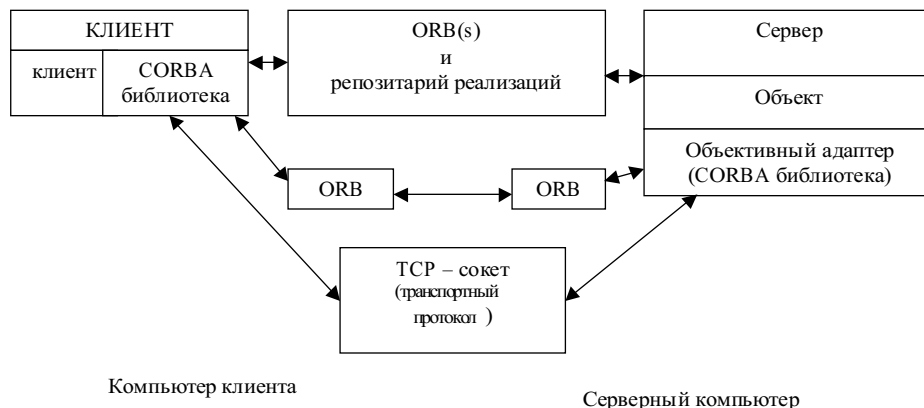


Рис. 6. Общая схема CORBA

Основой ОМА является Объектная модель OMG, которая определяет общую объектную семантику для спецификации характеристик объектов стандартным, независимым от реализации образом. **Объектная модель** OMG определяется в виде объектной модели ядра COM (Core Object Model) и совокупности расширений. **Ядро** специфицирует некоторый набор базовых понятий. Примерами базовых понятий являются объекты, операции, типы, отношение тип/подтип, наследование, интерфейс типа. Каждое **расширение** вводит дополнительный набор понятий. Расширяться может либо COM, либо уже существующие и согласованные расширения. При этом вводится понятие **профиля** - некоторой комбинации COM и одного, или нескольких расширений, вместе поддерживающих определенную целевую архитектуру.

Таким образом решения OMG сводятся к следующему:

вводится базовая объектная модель унифицированного языка спецификаций интерфейсов объектов;

отделяется реализация компонентов от спецификации их интерфейсов;

вводится общий механизм поддержки взаимодействия объектов (брокер объектных заявок, играющий роль "общей шины", поддерживающей взаимодействие объектов).

Тем самым достигается однородность представления компонентов и их взаимодействия.

Далее вводится слой унифицированных служб (сервисов), которые используются при проектировании прикладных систем и для формирования функционально законченных средств промежуточного слоя, предлагающих конкретные виды услуг. Службы и средства представляются однородно своими объектными интерфейсами, что позволяет обеспечить их взаимодействие посредством брокера объектных заявок.

Рассмотрим более подробно назначение каждого компонента ОМА.

Брокер Объектных Заявок (ORB) обеспечивает механизмы, позволяющие объектам посылать или принимать заявки, отвечать на них и получать результаты, не заботясь о положении в распределенной среде и способе реализации взаимодействующих с ними объектов. Он осуществляет поиск реализации объекта, участвующего в заявке, подготовку объектной реализации к приему заявки и передачу данных, являющихся результатом заявки. Интерфейс клиента полностью независим от расположения

вызываемого объекта, языка программирования, на котором он реализован, и любых других аспектов, не отраженных в интерфейсе вызываемого объекта. Стандарт Общей Архитектуры Брокера Объектных Заявок (CORBA) определяет среду для различных реализаций ORB, поддерживающих общие сервисы и интерфейсы. Это обеспечивает переносимость клиентов и реализаций объектов между различными ORB.

Объектные Службы представляют собой набор услуг (интерфейсов и объектов), которые обеспечивают базовые функции, необходимые для реализации других объектов. Операции, предоставляемые Объектными Службами, выступают в качестве базовых строительных блоков для Общих Средств (см. ниже) и прикладных объектов. Имеются следующие службы:

- Уведомления Объектов о Событии;
- Жизненного Цикла Объектов;
- Именования Объектов;
- Долговременного Хранения Объектов;
- Управления Конкурентным Доступом;
- Внешнего Представления Объектов;
- Объектных Связей;
- Транзакций;
- Изменения Объектов;
- Лицензирования;
- Объектных Свойств;
- Объектных Запросов;
- Безопасности Объектов.

Например, функции СУБД представлены рядом таких служб, как долговременное хранение объектов, управление конкурентным доступом к объектам, служба транзакций, службы объектных связей, объектных запросов, изменений объектов и т.п. Эти и другие службы, взятые вместе, реализуют функции как объектных, так и реляционных СУБД. Спецификация служб формируется на основе опыта промышленных корпораций, входящих в состав OMG.

Общие Средства занимают место между ORB и объектными службами с одной стороны, и прикладными объектами с другой. Таким образом, ORB обеспечивает базовую инфраструктуру, Объектные Службы - фундаментальные объектные интерфейсы, а задача Общих Средств - поддержка интерфейсов сервисов высокого уровня. В эту группу входят следующие средства:

- поддержки пользовательского интерфейса;
- управления информацией;
- управления системой;
- управления задачами и др.

Взаимодействие ORB поддерживается Универсальным Межброкерным Протоколом GIOP (General Inter-ORB Protocol). GIOP является универсальным, поскольку он не зависит от конкретной сетевой транспортной среды и может быть отображен в любой транспортный протокол, поддерживающий виртуальные соединения. Отображение GIOP в протокол TCP/IP - определено CORBA в качестве Межброкерного Протокола Internet IIOP (Internet Inter-ORB Protocol). Назначение протокола GIOP/IIOP заключается в том, чтобы поддерживать сети брокеров в рамках Internet и за ее пределами. Согласно GIOP, внутренняя архитектура брокеров предполагается неизвестной. Подход, который может быть выбран конкретным брокером для поддержки GIOP/IIOP, не определяется. Все, что требуется для согласованного включения брокера в компьютерную сеть, - это существование связанных с ним компонентов, способных посылать и принимать сообщения IIOP. **Спецификация GIOP** включает:

- определение общего представления данных, т.е. синтаксиса,

отображающего значения типов данных в формат передачи данных между брокерами;

форматы передаваемых сообщений GIOP, которые введены для поддержки объектных заявок, установления местоположения реализаций объектов и управления транспортными соединениями;

определение ограничений на допустимый сетевой транспорт GIOP.

Принятие стандарта CORBA позволяет обеспечить поддержку глобального объектного пространства в масштабе Internet. Различие назначений WWW-технологий и CORBA заключается в том, что WWW облегчает жизнь поставщиков и потребителей информации, а CORBA облегчает задачу разработчиков систем и фирм-поставщиков инструментальных средств. Поэтому роли WWW и CORBA являются взаимно дополняющими, и в этой связи требуются специальные технологии, обеспечивающие их сопряжение.

В ситуации, когда для работы со сценариями Web-сервера используется форма запроса имени пользователя и пароля для входа в систему, значения переменных Username и Password передаются в сценарий, отвечающий за допуск пользователей только к тем областям Web-сервера, к которым они имеют право обращаться. Для этого используется протокол HTTP, в котором отсутствует понятие состояния. Каждый раз при загрузке новой страницы пользователь фактически отключается от сервера, который, в свою очередь, не имеет представления, кто пользователь такой и что он в данный момент делает. Даже после входа на сервер, при пересылке каждой страницы браузер пользователя должен отправлять на сервер имя пользователя и пароль, чтобы можно было проверить, законно ли обращение к данной странице. Иными словами, ни клиентская, ни серверная части приложения (т.е. ни Web-браузер, ни Web-сервер) не могут оперировать такими понятиями, как локальные переменные, локальные вызовы метода или локальные объекты.

Возможности CORBA, значительно более мощные, чем реализуемая WWW простая модель обмена HTML-страницами (HTML – специальный язык разметки гипертекстовых документов). Интеграция приведет к наилучшему использованию этих двух стандартов. Известны два основных подхода к интеграции CORBA и WWW. Первый из них основан на построении шлюзов между приложениями WWW и CORBA, служащих для трансформации HTTP в протокол CORBA IIOP. Другой подход заключается во встраивании функций CORBA в состав клиентов WWW (программ просмотра) и серверов. Реализация второго подхода возможна либо на основе новых WWW клиентов и серверов со встроенным IIOP, либо при помощи подгрузки из сети модуля поддержки IIOP в клиенте или сервере.

В новом поколении WWW клиентов и серверов, поддерживающих Java, модуль поддержки IIOP реализуется на Java. Так, для любого ресурса, доступного посредством CORBA, может быть разработан пользовательский интерфейс как апплет Java (апплет – небольшие программы, написанные на языке программирования Java). Этот апплет использует модуль IIOP для взаимодействия с сервером CORBA. При первом доступе пользователя к какой-либо услуге, программа просмотра автоматически загружает и устанавливает апплет пользовательского интерфейса. После этого пользователь имеет доступ к этой услуге посредством собственного апплета. Таким образом, услуги объектов-серверов оказываются доступными широчайшей аудитории, независимо от применяемых пользователями платформ и при сохранении для разработчика возможности совершенствования реализации услуг и их интерфейсов.

Главные достоинства технологии распределенных объектов CORBA – независимость и поддержка, в частности, независимость от языка, производителя и операционной системы. Брокеры (ORB) для CORBA имеются

практически для любой серьезной операционной системы. Существуют также CORBA ORB для организации взаимодействия между приложениями на различных языках, в частности, C++, Ada, COBOL, Smalltalk, Java. Использование IIOP позволяет ORB, разработанным одним производителем, получать объекты (и затем манипулировать ими) от удаленного ORB, созданного другим производителем. ORB на Java позволяют создавать Java-клиенты, не устанавливая какое бы то ни было специальное ПО на клиентской станции.

Существуют альтернативы CORBA - DCOM и RMI.

Разработанная Microsoft технология DCOM (Distributed Component Object Model) реализована только для операционных систем, Windows 95 и Windows NT. В сотрудничестве с рядом сторонних компаний Microsoft работает над переносом DCOM на другие операционные системы (в частности, на некоторые версии Unix). DCOM, как и CORBA, не зависит от языка; все объекты в нем описываются через интерфейсы, написанные на предложенном Microsoft языке ODL (Object Description Language).

По сравнению с CORBA, у DCOM имеются три главных недостатка. Во-первых, эта технология разработана и поддерживается только одним производителем (Microsoft), что значительно ограничивает разработчика в возможностях выбора (например, функций или инструментальных средств). Во-вторых, DCOM поддерживает небольшое число платформ, и это весьма невыгодно с точки зрения использования существующих программ и масштабируемости приложений. Наконец, DCOM не обладает зрелостью CORBA. В настоящее время Microsoft только разрабатывает средства поддержки обмена сообщениями и транзакциями для DCOM, а в CORBA все это есть начиная со второй версии (1994 год), и многие производители ПО уже обеспечивают такую поддержку в своих продуктах. Разработчик, написавший апплет или приложение Java для обращения к объектам DCOM на сервере, автоматически ограничивает область применения клиентского ПО браузером Internet Explorer от Microsoft и платформами Windows 95/NT. Данное ограничение, безусловно, обедняет возможности использования приложений, размещаемых на Web.

Технология RMI (см. рис. 7) (Remote Method Invocation) предложена корпорацией Sun на основе языка программирования Java.

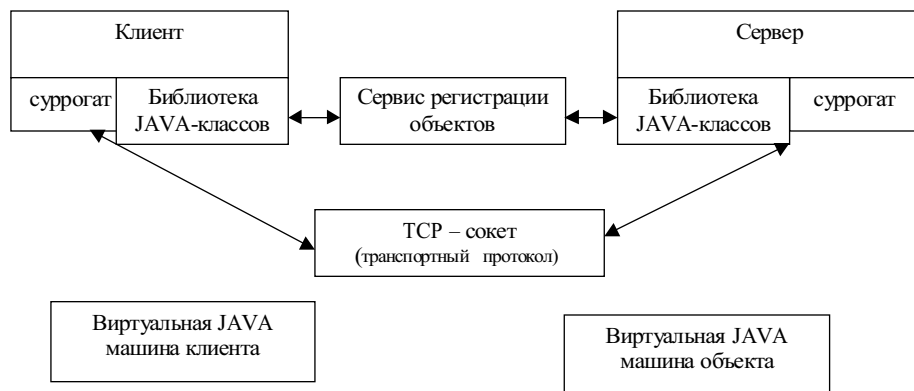


Рис. 7. Общая схема RMI

RMI позволяет клиентам Java создавать экземпляры объектов, находящихся на удаленном сервере, но серверное приложение тоже должно быть написано на Java. Помимо невозможности использовать существующий программный код, у RMI есть и другие ограничения. В отличие от CORBA, в RMI нет понятия сервиса. Кроме того, серверные объекты, написанные на Java с использованием RMI, дают низкую производительность из-за ограничений, связанных с использованием виртуальной машины Java. (проверено, что серверные приложения CORBA на Java работают быстрее, чем аналогичные приложения, использующие RMI.) Кроме того, RMI не дает возможности устанавливать правила активизации объектов, как это делается в ORB.

В перспективе RMI и соответствующие части технологии Java будут приближаться к стандартам OMG, а не наоборот. Sun публично объявила, что будет стремиться к обеспечению возможности для объектов RMI взаимодействовать через IIOP. В целом технология RMI может быть полезна для небольших приложений, целиком написанных на Java, а CORBA обеспечивает возможность интеграции существующих объектов в новые программы и масштабирования приложений.

1.5. Программное обеспечение интерфейсов АИС

По мере роста мощности компьютеров увеличиваются затраты на диалоговую компоненту программного обеспечения. Термин эффективность постепенно изменил свое значение, если раньше он отражал такие характеристики как процессорное время и объем занимаемой памяти, то теперь под ним понимают простоту разработки, легкость сопровождения и удобство работы с программой.

Поэтому затраты на исследование и разработку пользовательского интерфейса являются оправданными. Большинство современных пользовательских интерфейсов основываются на аналогичных идеях:

- активное использование “мыши”;

- ориентированность на объекты;

- графика и имитация процессов и явлений;

- возможность использования алгоритмов, знакомых каждому человеку из его обыденной жизни.

Следовательно существует возможность и необходимость разработки вспомогательного программного обеспечения, предназначенного для создания стандартных интерфейсов, точнее их базисов. С другой стороны, разнообразие аппаратных и системных платформ, на которых должно будет работать это программное обеспечение, требует его переносимости на уровне исходного кода. Вышеизложенные требования логически приводят к идее переносимого унифицированного программного инструмента для создания пользовательских интерфейсов, точнее, программной системы (модуля, блока), которая обеспечивает интерфейс с пользователем. В качестве конкретного примера рассмотрим широко известную X Window.

X Window (далее X) - это система для создания графического пользовательского интерфейса, изначально - на компьютерах, работающих под управлением ОС UNIX. Она построена по схеме клиент-сервер. Взаимодействие X-клиента и X-сервера происходит в рамках соответствующего протокола прикладного уровня. В качестве транспортного протокола может служить как локальный, так и любой сетевой, например, TCP. Это означает, что различия в архитектуре X- клиента и X-сервера не играют никакой роли и их совместимость обеспечивается стандартом X-протокола.

Система обеспечивает графический вывод на экран, воспринимает сигналы от устройств ввода, таких, как клавиатура и мышь, и передает их

программам. Устройство вывода может иметь более одного экрана и X обеспечивает вывод на любой из них. Все это: экран (экраны), устройства ввода (клавиатура, мышь) называется в терминах X Window - дисплей.

Благодаря своей архитектуре X Window свободно используется в распределенных вычислительных системах, например, в сетях TCP/IP (Internet) и позволяет пользователю (за дисплеем) общаться со многими программами одновременно. Чтобы вывод из них не смешивался, система создаёт на экране дисплея “виртуальные подэкраны” - окна. Каждое приложение (как правило) рисует только в своём окне (или своих окнах). X предоставляет набор средств для создания окон, их перемещения по экрану, изменения их размеров, вывода в них и т.п.

Как правило, программы имеют набор конфигурационных параметров - ресурсов. Это может быть цвет окна, различные параметры текстового шрифта и многое другое. Система стандартизует способ задания ресурсов приложений, управления ими, и содержит ряд процедур для работы с ними. Эта совокупность функций называется “менеджер ресурсов”.

X функционирует согласно идеологии управляемости событиями - она организует общение между самими программами и между программами и внешней средой посредством событий. **Событие** есть единица информации, идентифицирующая происходящие в системе изменения или действия. По идентификатору события можно получить информацию о нём: вид события, его характеристики, где оно произошло и т.п..

Система X Window представляет собой совокупность программ и библиотек. Ядром ее является специальная программа - X-сервер. Именно сервер знает особенности конкретной аппаратуры, знает, что надо предпринять, чтобы вывести какой-либо графический объект на экран. Он же умеет воспринимать и обрабатывать сигналы, приходящие от клавиатуры и мыши. Сервер общается с программами-клиентами, посылая им или принимая от них пакеты данных.

Если сервер и клиент находятся на разных машинах, то данные посылаются по сети, а если на одном, - то используется внутренний канал. Например, если сервер обнаруживает, что пользователь нажал кнопку мыши, то он подготавливает соответствующий пакет (событие) и посылает его тому клиенту, в чьём окне находился курсор мыши в момент нажатия кнопки. И наоборот, если программе надо вывести что-либо на экран дисплея, она создаёт необходимый пакет данных и посылает его серверу. Описание этого взаимодействия, форматов пакетов и т.п. и составляет спецификацию вышеупомянутого X-протокола.

Чтобы программировать для X, совсем необязательно знать детали реализации сервера и X -протокола. Система предоставляет стандартную библиотеку процедур, с помощью которых программы осуществляют доступ к услугам X на высоком уровне. Так, для того, чтобы вывести на экран точку, достаточно вызвать соответствующую стандартную процедуру, передав ей требуемые параметры. Эта процедура выполнит всю работу по формированию пакетов данных и передаче их серверу.

Окно - это базовое понятие в X. Оно представляет, обычно, прямоугольную область на экране, предоставляемую системой программе-клиенту. Последняя использует окно для вывода графической информации. Окно имеет внутренность и край. Основными атрибутами окна являются ширина и высота внутренности, а также ширина (толщина) края. Эти параметры называются геометрией окна.

С каждым окном связывается система координат, начало которой находится в левом верхнем углу окна (точнее - его внутренности). Ось X направлена вправо, а ось Y - вниз. Единица измерения по обеим осям - пиксел.

X Window позволяет программе создавать несколько окон одновременно. Они связаны в иерархию, в которой одни являются “родителями”, а другие - “потомками”. Сам сервер на каждом экране создает одно основное окно, являющееся самым верхним “родителем” всех остальных окон. Это окно называется “корневым” (root).

Окна могут располагаться на экране произвольным образом, перекрывая друг друга. X Window имеет набор средств, пользуясь которыми, программа-клиент может изменять размеры окон и их положение на экране. Особенностью системы является то, что она не имеет встроенной возможности управлять окнами с помощью мыши или клавиатуры. Чтобы это можно было осуществить, нужен специальный клиент - менеджер окон. Однако, менеджер не может корректно управлять окнами, ничего о них не зная. Окна могут обладать различными свойствами, которые должен обеспечивать именно менеджер окон. Например, во многих случаях удобно иметь заголовки окон, в других - желательно, чтобы окно нельзя было сделать меньше, или наоборот - больше, определённого размера. Окно может быть “схлопнуто” в пиктограмму (“иконку”) - в этом случае менеджер должен знать, какую пиктограмму использовать и как её назвать. Клиенты могут сообщать менеджеру свои пожелания относительно окон двумя способами:

при создании окна X могут быть переданы “рекомендации” о начальном положении окна, его геометрии, минимальных и максимальных размерах и т.д.;

можно использовать встроенный в X способ общения между программами - механизм “свойств”.

Система X Window предназначена для работы на растровых дисплеях. Число бит на пиксел называют **глубиной** или **толщиной** дисплея. Биты с одинаковыми номерами (одинаковые двоичные разряды) во всех пикселах образуют как бы плоскость, параллельную экрану. Её называют **цветовой плоскостью**. X позволяет рисовать в любой цветовой плоскости (-ях), не затрагивая остальные. Значение пиксела не задаёт цвет точки на экране непосредственно, но задаёт номер ячейки в специальном массиве, в которой и хранится значение цвета, т.е. значение пиксела задаёт номер цвета в текущей палитре. X имеет большой набор процедур, позволяющих рисовать графические примитивы: точки, линии, дуги, текст; работать с областями произвольной формы.

В X Window встроены средства для обмена информацией между программами- клиентами. Для этого используется механизм “свойств”. “Свойство” - это информационная структура, связанная с некоторым объектом, например, окном, доступная всем клиентам X. Каждое свойство имеет имя и уникальный идентификатор - атом. Обычно, имена свойств записываются большими буквами. Атомы используются для доступа к содержимому свойств с тем, чтобы уменьшить объём информации, пересылаемой между клиентами и X сервером. Предусмотрен ряд процедур, позволяющих перевести имя свойства в уникальный атом, и, наоборот, по атому получить необходимые данные. Некоторые свойства и соответствующие им атомы являются предопределёнными и создаются в момент инициализации сервера.

Рассмотрим ряд обобщенных требований к системе управления пользовательского интерфейса - UIMS (User Interface Management System), как составной части системы разработки пользовательского интерфейса (инструментарий) - UIDS (User Interface Development System). Такой комплекс будем обозначать - UIDS/UIMS.

Можно выделить три объекта, для каждого из которых ставятся различные цели при разработке UIDS.

1. Интерфейс с пользователем:

- согласованность;
- поддержка пользователя разного уровня;
- обеспечение обработки ошибок и восстановления.

2. Разработчик программного обеспечения:

- предоставление абстрактного языка для конструирования интерфейса пользователя;
- предоставление согласованных интерфейсов для связанных прикладных задач;
- обеспечение простоты изменения интерфейса на стадии его проектирования (быстрое создание прототипа);
- упрощение разработки повторным использованием программных компонент;
- обеспечение простоты изучения и использования прикладных программ.

3. Конечный пользователь:

- согласованность интерфейса по прикладным программам;
- многоуровневая поддержка сопровождения или функций помощи;
- поддержка процесса обучения;
- поддержка расширяемости прикладных программ.

Эти цели определяют следующие **функциональные характеристики**

UIDS/UIMS:

- работа с входными устройствами;
- проверка допустимости ввода;
- обработка ошибок пользователя;
- реализация обратной связи;
- поддержка обновления/изменения данных прикладной задачи;
- поддержка задач развития интерфейса;
- синтаксическая поддержка.

Наиболее часто используется модель, в соответствии с которой UIMS состоит из трёх компонент:

- система представления, обеспечивающая низкоуровневый ввод и вывод;
- система управления диалогом, обрабатывающая лексические единицы, получаемые в системе представления, в соответствии с синтаксисом диалога;
- модель интерфейса прикладной программы, представляющая семантику диалога и управляющая функциональностью прикладной программы.

На ее основе определяется структура эталонной модели UIDS/UIMS.

Конкретные реализации моделей основываются на различных способах спецификации интерфейса, среди которых можно выделить следующие типы:

- языковая;
- графическая;
- автогенерация по спецификации прикладной задачи;
- объектно-ориентированный подход.

Каждая из этих спецификаций имеет свои особенности. Для **языковой** спецификации применяется специальный язык, чтобы задавать синтаксис интерфейса. Такими языками могут служить:

- сети меню;
- диаграммы состояний и переходов;
- контекстно-свободные грамматики;
- языки событий;
- декларативные языки;
- обычные языки программирования;
- объектно-ориентированные языки.

Этот подход приложим к широкому кругу прикладных задач. Его недостатком является то, что разработчик диалога должен обладать профессиональной программистской подготовкой.

Графическая спецификация связана с определением интерфейса с помощью размещения объектов на экране (визуальное программирование, программирование по примерам). Она проста для использования не программистами, но: трудна в реализации; поддерживает лишь ограниченные интерфейсы. Здесь также существуют разные формы реализации:

- размещение на экране интерактивных средств (меню, кнопки и т.п.) и их привязка к фрагментам, написанным разработчиком интерфейса;

- сеть статичных страниц (кадров), содержащих тексты, графики, интерактивные средства; спецификация по демонстрации.

В методе **автогенерации по спецификации прикладной задачи** интерфейс создаётся (точнее, делается попытка создать) автоматически по спецификации семантики прикладных задач. Этим, в сущности, предпринимается попытка преодолеть сложности использования других технологий, что несомненно является достоинством этого подхода. Однако, ввиду сложности адекватного описания интерфейса, трудно ожидать скорого появления систем, реализующих такой подход в полной мере. Возможные реализации, создание интерфейса:

- на основе списка процедур прикладной программы;

- по типам параметров процедур;

- на основе определения семантики прикладной задачи, описываемой на специальном языке.

Объектно-ориентированный подход связан с принципом, называемым DM (Direct Manipulation). Основное свойство этого подхода состоит в том, что пользователь взаимодействует с индивидуальными объектами, а не со всей системой как единым целым. К конструированию DM интерфейсов предъявляются следующие требования:

- поддержка разных классов интерактивной технологии;

- возможность создания новых типов технологий;

- использование подходящего, простого в изучении языка программирования для описания частей системы, которые трудно специфицировать графически;

- предоставление разной интерактивной технологии одновременно;

- обеспечение семантической обратной связи, проверки семантики и использования в семантике установок по умолчанию;

- гибкость компоненты представления;

- выражение синтаксиса в терминах индивидуальных объектов.

В заключении приведём в качестве примера описание функций системы UIMS одной из фирм, созданной на базе X Window System.

Проектирование интерфейса - пользователь создает интерфейс в интерактивном режиме, используя predetermined элементы – заготовки.

Спецификация ресурсов - реализует простую установку параметров для заготовок. Многообразие ресурсов заготовок и их взаимодействия делает задачу установки параметров чрезвычайно сложной, поэтому везде, где это возможно, предоставляется predetermined установка соответствующего выбора.

Спецификация поведения интерфейса - описывается на C-подобном командном языке.

Связь между интерфейсом и прикладной задачей - реализована вызовом функции прикладной задачи из описания или с помощью разделяемых переменных (активных значений), разделяемые переменные могут быть любого типа.

Тестирование интерфейса и его поведения (режим попытки) - в этом режиме интерпретируется описание связанное с какими-либо событиями, но без вызова функций прикладной задачи.

Эффективность конечного приложения - результат проектирования реализуется одним из двух способов:
интерпретацией описания;
компиляцией интерфейса вместе со всеми описаниями в Си-код.

1.6. Классификация, состав и структура технических средств АИС

1.6.1. Концепция и основные понятия

В развитии технического обеспечения АИС наблюдаются процессы, схожие с процессами развития программного обеспечения:

- интеграция компонентов и систем;
- масштабируемость;

- сотрудничество различных производителей по сближению характеристик продукции на базе стандартов. Под **техническим обеспечением АИС** понимается комплекс технических средств, применяемых для создания АИС, в который входят: средства вычислительной техники, периферийное оборудование, средства связи и др..

Ряд фирм-производителей, которые преобладают на рынке технических средств АИС, стараются выпускать весь спектр средств вычислительной техники (ВТ), который способен полностью решить проблему интеграции технического обеспечения корпоративных и других АИС. Масштабируемые компьютерные системы призваны удовлетворить потребности как разработчиков, так и предприятий, предъявляющих высокие требования по гибкости и надежности приложений. Для этого разрабатываются, производятся и поставляются на рынок рабочие станции, серверы и системы хранения информации уровня предприятия, а также сетевые компьютеры.

Сосредоточивая усилия на ключевых областях информационных технологий - технологиях, продуктах и видах обслуживания, которые составляют основу сетевых ИТ, во всех других областях эти фирмы основываются на стратегическом распределении усилий, полагаясь на дополняющих их партнеров и поставщиков. По мере того, как развивались новые направления деятельности, ключевые технологии претерпевали эволюцию. Например, когда обнаруживались ограничения в существующих микропроцессорных технологиях, увеличивались капиталовложения в разработку процессоров и заключались соглашения о стратегическом партнерстве с производителями полупроводников.

Некоторые производители средств ВТ проникают (диверсифицируют) в смежные области и начинают разрабатывать и производить для своих систем ВТ программное обеспечение, которое содержит как среды программирования, так и операционные системы, а также соответствующие утилиты. Ведутся также собственные разработки сетевых технологий таких, как:

- оконечное оборудование данных** - функциональный блок, осуществляющий подготовку данных, предназначенных для пересылки по каналу передачи данных и служащий источником данных, приемником данных или тем и другим вместе;

- станция данных** - совокупность оконечного оборудования данных и аппаратуры окончания канала данных;
- и др.

Далее, на примере корпорации Sun рассмотрим состав и структуру ее продуктов и технологий в области средств ВТ, которые обеспечивают создание распределенных АИС.

1.6.2. Краткая характеристика основных составляющих физической компоненты АИС

Рабочие станции

Рабочие станции Ultra обладают техническими характеристиками, которые позволяют на их основе создавать эффективные АРМы в АИС различного назначения. Все рабочие станции основаны на архитектуре, позволяющей добиться масштабируемости проектных решений на базе единой сетевой вычислительной платформы. Это обеспечивается благодаря модульной конструкции, что позволяет наращивание конфигурации - новыми процессорами, памятью, средствами ввода-вывода и графики. Достаточно просто подключить их к пакетному коммутатору. А операционная среда Solaris (также продукт Sun) с поддержкой множественных потоков гарантирует, что при добавлении к системе новых ресурсов, будет получен именно такой выигрыш в производительности, который объявлен в соответствующей фирменной документации. Масштабируемость позволяет до минимума сократить количество узких мест при наращивании сети и устранить возникающую в таких случаях потребность полного переоснащения средств ВТ.

Обратите внимание на то, как от модели к модели наращиваются возможности и мощность рабочих станций.

Sun Ultra 5 - рабочая станция на основе архитектуры ПК, оснащенная процессором UltraSPARC-lil представляет собой низший уровень в семействе. Тем не менее она обладает хорошими характеристиками:

- большим запасом дисковой и оперативной памяти;

- возможностью сопряжения через шину PCI;

- реализованной на плате поддержке графики;

- проста в установке, адаптации к индивидуальным требованиям и в администрировании.

Sun Ultra 10 - оснащенная процессором UltraSPARC-III с хорошим соотношением "цена/производительность". Имеет возможность установки до 1 ГБ памяти для особо сложных задач автоматизированного проектирования и четыре PCI-слота для повышения гибкости и расширения возможностей сопряжения с периферийными устройствами.

Графические подсистемы Creator, Creator3D и Elite3D (описаны далее) с поддержкой 24-дюймового цветного монитора и парных мониторов позволяют обеспечивать эффективное решение задач создания 2х- и 3х-мерной цифровой художественной графики, автоматизированного проектирования в области машиностроения, финансового анализа, комплексной автоматизации производства и имитационного моделирования.

Sun Ultra 60 - рабочая станция, которая поддерживает до двух высокопроизводительных процессоров UltraSPARC-II включает также:

- кросс-коммутационную шину (частота 120 МГц) для обеспечения высокой пропускной способности;

- средства ввода-вывода и работы с сетью на базе шины PCI (66 МГц);

- графические подсистемы Creator, Creator3D и Elite3D m6 с поддержкой 24-дюймового цветного монитора и парных мониторов.

Графические подсистемы семейства Ultra Creator, Creator3D и Elite3D обеспечивают потребности в обработке изображений при работе с 2х- и 3х-мерной графикой. Creator используется при редактировании изображений, при подготовке изображений к печати и компоновке страниц. Creator 3D обеспечивает высокопроизводительную работу с 3х- мерной графикой, моделирование объектов, при обработке изображений. При обработке

изображений, Creator3D работает с 3х- мерными объектами и использует средство (VIS) для ускорения обработки изображений, масштабирования, интерполяции и отображения 2х- мерных и объемных образов.

Реализованный на кристалле расширенный набор команд VIS обеспечивает поддержку мультимедиа. Кросс-коммутатор UPA позволяет ускорить передачу больших массивов данных, а арифметико-логический блок попиксельной обработки содержимого оперативной графической памяти еще больше увеличивает быстродействие.

Elite3D использует выделенное устройство для операций с плавающей точкой для достижения еще более высокого уровня профессиональной трехмерной графики. Это позволяет создавать очень сложные системы графической визуализации и анимации и, более того, - обеспечивает уровень производительности и функциональности, доступный ранее только на очень дорогих графических системах. Подсистемы Elite3D бинарно совместимы с API программного обеспечения Creator3D. Elite3D в системах Ultra предпочтительна для высококачественной анимации, при интерактивном создании реалистичных изображений и для создания специальных оптических эффектов. Она позволяет разработчикам в интерактивном режиме работать с большими массивами данных или со сложными моделями при высоком качестве на выходе.

Сетевые технологии создают потребность в новом типе рабочих станций таких, как, например, Java-станции – специальные сетевые компьютеры. Сетевые компьютеры дешевы, не требуют какого-либо администрирования и оптимизированы для системы Java. Java Станция полностью использует возможности сети, которая предоставляет все - от Java приложений и служб до полного интегрированного системного и сетевого управления. Эти системы работают под управлением JavaOS - компактной, расширяемой, операционной системы с полной Java-поддержкой, специально разработанной для реализации Виртуальной Машины Java. Они предлагают модульный, масштабируемый и настраиваемый браузер HotJava и настольную среду, обеспечивая доступ к широкому набору соединительных возможностей и средств повышения производительности, в том числе терминальным эмуляторам, Windows- и Java-приложениям. Установка и управление очень просты - достаточно включить в сеть а программное обеспечение распространяется автоматически. Удаленную загрузку, администрирование и хранение данных обеспечивают серверы, поддерживающие системы Java Станций. Кроме того, встроенная в Java система безопасности обеспечивает надежную и свободную от вирусов работу приложений. Все устройство обеспечивает замену на месте, поэтому нет проблем с поддержкой. С помощью специальной флеш-памяти, загрузка Java-станции происходит быстро и надежно - что делает ее эффективным решением для удаленных установок, подключенных к сети по модему.

Серверы

Семейство **серверных станций** (серверов) фирмы Sun как и семейство рабочих станций представляет собой масштабируемую иерархическую систему.

С учетом особенностей сегодняшних сетевых сред уровень требований к серверу весьма высок. Сервер должен быть:

- достаточно мощным и масштабируемым;
- надежным: - данные и прикладные задачи должны быть всегда доступны для использования;

- способным взаимодействовать со всеми техническими средствами АИС;
- легко устанавливаемым, сопровождаемым и администрируемым.

Сервер должен обеспечивать решение следующих основных задач:

- организация электронной почты;
- работа с базами данных;
- создание и предоставление Web-сервиса;
- обеспечение управления ресурсами АИС и др.

Сервер Sun Enterprise 2 имеет многопроцессорную архитектуру и предназначен для работы в качестве сервера рабочих групп. Сервер имеет следующие характеристики:

- поддерживает до двух UltraSPARC процессоров с тактовой частотой 167 МГц или 200 МГц;

- имеет модульную конструкцию, что позволяет добавлять или обновлять ЦПУ и расширять систему по мере внедрения новых технологий при их появлении на рынке;

- SCSI-2 интерфейс (20 Мб/с) обеспечивает ускоренный ввод-вывод;

- сетевые возможности сервера определяются наличием встроенного сетевого адаптера Fast Ethernet (100Мб/с);

- операционной средой является Solaris, которая предоставляет набор средств управления сетью;

- служба SunService обеспечивает обслуживание клиентов 24 часа в сутки все семь дней в неделю.

Сервер Sun Enterprise 450 включает:

- до четырех процессоров UltraSPARC с тактовой частотой 250 или 300 МГц;

- внутреннюю шину со скоростью 1,6 Гб/с;

- подсистему ввода-вывода на базе шины PCI, обеспечивающую пропускную способность 600Мб/с.;

- оперативную память, расширяемую до 4 Гб;

- встроенные диски Ultra SCSI емкостью до 84 Гб с режимом быстрой замены;

- возможности подключения внешних накопителей общей емкостью до 6 Тб.

Сервер может взаимодействовать с любыми рабочими станциями, Intel и Macintosh, которые могут оказаться в корпоративной сети. Надежность обеспечивается возможностями:

- коррекции ошибок на внутренней шине и в памяти;

- автоматического восстановления системы после сбоя;

- резервируемые с режимом "горячей замены" блоки питания и дисковые накопители;

- все это присутствует в стандартной конфигурации.

Сервер Sun Enterprise 4500 - это компактный сервер среднего уровня с большой вычислительной мощностью. Он имеет общую модульную архитектуру семейства серверов от Sun Enterprise 3500 до Sun Enterprise 6500, что облегчает модификацию и позволяет производить локальное наращивание вычислительных возможностей путем подключения до 30 процессоров. В сервере возможна установка до 14 процессоров, до 4 серверов Sun Enterprise 4500 можно установить в стойку центра данных.

Самым мощным в семействе серверов является Sun Enterprise 10000, который может использоваться в качестве:

- центрального информационного сервера корпоративной АИС;

- информационного хранилища;

- сервера мощных (тяжелых) приложений с онлайн-обработкой транзакций (OLTP).

Enterprise 10000 включает:

- до 64 процессоров UltraSPARC с тактовой частотой 250-Мгц;

- до 64 Гб разделяемой памяти, с шириной пропускания внутрисистемной

магистрала до 12 гигабайт в секунду;
до 20 ТБ дискового пространства, что существенно для проектов крупнейших хранилищ данных;
возможности “горячей замены” без необходимости перезагрузки или выключения питания.

Управления дисковыми массивами

Для **создания хранилищ данных в корпоративных системах** Sun выпускает специальное высокопроизводительное и надежное семейство машин для управления дисковыми массивами - StorEdge. StorEdge представляет собой универсальную платформу, которая может одновременно работать с вычислительными машинами разных типов, совместимыми на уровне внешнего интерфейса. Специальное ПО позволяет организовать совместное использование одной и той же копии данных разнородными вычислительными машинами.

Чтобы еще более повысить гибкость системы и защищенность данных предусмотрена возможность быстрого переназначения дискового пространства с одной машины на другую. Высокий уровень готовности системы к выполнению своих функций достигается благодаря архитектуре с двойным резервированием, обеспечивающей поддержку дисковых массивов и зеркалированных устройств. Средства автоматической связи с администратором позволяют осуществлять дистанционную диагностику системы и своевременно принимать необходимые меры по устранению ошибок и неисправностей - как правило, без необходимости высылать на объект обслуживающий персонал.

Дисковый массив Sun StorEdge A3000 обеспечивает 99.99% времени работоспособности. Все компоненты:

- контроллеры;
- блоки питания;
- блоки охлаждения;
- диски;

соответствуют требованиям избыточности и “горячего подключения”, что обеспечивает высокий уровень надежности, отказоустойчивости и сервисного обслуживания. В основе Sun StorEdge A3000 лежат двойные, активные, интеллектуальные, аппаратные контроллеры. Интерфейс с хост-системой (ЭВМ, которая предоставляет пользователю основные информационные и другие ресурсы) осуществляется по двойному каналу (40 Мбт/сек). Это улучшает пропускную способность системы ввода/вывода, поскольку контроллер разгружает ЦП от вычислений контроля четности в операциях ввода/вывода. Тем самым ускоряются приложения, зависящие от операций произвольного ввода/вывода, такие как информационные хранилища, серверы баз данных и файлов, системы принятия решений. В то же время приложения, использующие последовательные операции ввода/вывода (обработка изображений, мультимедиа, видео и сбор данных), могут воспользоваться более высокой производительностью хост-связи по двойному дифференциальному каналу.

Система может масштабироваться до 11 терабайт дискового хранения, а поставляется с 15 или 35 дисками, что обеспечивает до 318 Гбт дисковой памяти в едином корпусе. Там же может быть расположена интегрированная 400 Гбт библиотека, обеспечивающая скорость передачи 6 Мбт/сек с использованием двух приводов. В результате получается законченное решение для хранения данных в едином корпусе.

Работоспособность системы поддерживается автоматическим

восстановлением и дружелюбным графическим интерфейсом администратора, что упрощает как восстановление, так и управление системой. Дифференциальный Ultra SCSI интерфейс обеспечивает подключение к хост-компьютерам производства других фирм, что является существенным для неоднородных сетей.

Следующим членом семейства является Sun StorEdge A5000, который представляет собой второе поколение дисковых массивов Sun и обладает повышенной надежностью, готовностью к работе и обслуживанию. Его производительность и емкость наращиваются параллельно, поэтому система пригодна для использования как в масштабе отдела предприятия, так и в крупном ВЦ, где требуется хранение многих терабайт данных.

Следующее поколение - интеллектуальный сервер хранения данных Sun StorEdge A7000 разработан специально для крупных центров обработки и хранения данных и для коллективного использования информации.

Для автоматизированного резервного копирования и восстановления данных в корпоративных АИС предназначено компактное устройство Enterprise Tape Library 4/1000 - ленточная библиотека, которая может сохранять более терабайта данных без сжатия и обеспечивает скорость и надежность, необходимые для широкомасштабного резервного копирования. При использовании сжатия данных может осуществиться резервное копирование более 300 Гб данных за восемь часов. Надежность системы определяется тем, что сбои встречаются не чаще, чем раз в течение двух миллионов средних рабочих циклов.

2. НОРМАТИВНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КАЧЕСТВА, ЭФФЕКТИВНОСТИ И БЕЗОПАСНОСТИ АИС

2.1. Стандартизация типового жизненного цикла АИС

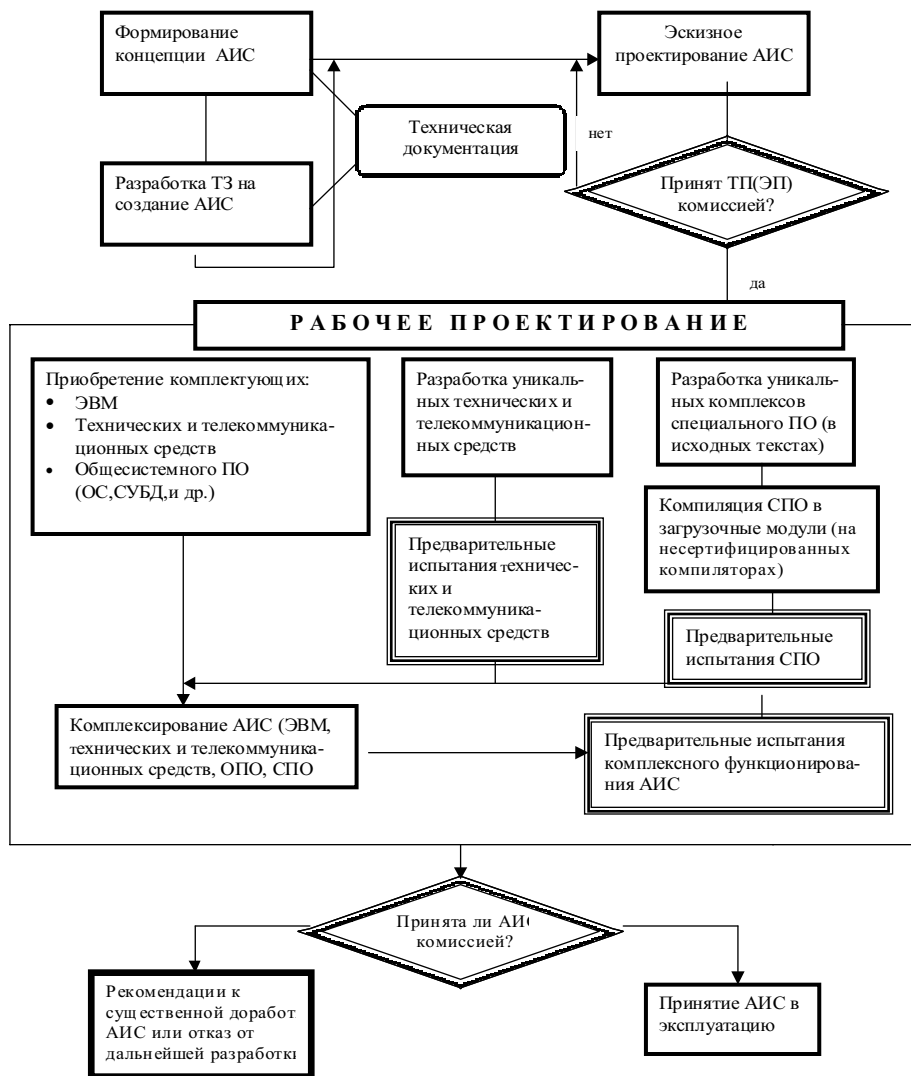
Методической основой технологии, регламентирующей деятельность специалистов в процессе разработки, является типовой жизненный цикл АИС (см. рис. 8) и технологический процесс, который отражает этапы и систему операций в последовательности их выполнения и взаимосвязи, обеспечивающих ведение разработки от подготовки технического задания до завершения испытаний и ввода в эксплуатацию.

Типовой жизненный цикл АИС отображается в системе **нормативно-технических документов**, которые содержат правила, общие принципы или характеристики, касающиеся различных видов деятельности или их результатов, стандарты, технические условия, инструкции и регламенты по применению. Правила оформления, содержание, структура и привязка к конкретному этапу типового жизненного цикла АИС (ЖЦ АИС) нормативно-технических документов содержатся в международных и национальных стандартах. Стандартизация типового жизненного цикла АИС проходила постепенно в течение многих лет и к настоящему времени ее еще нельзя считать завершенной. Тем не менее, создание эффективных сложных АИС было бы сегодня невозможно без следования требованиям стандартов. Далее рассматриваются основные стандартные нормативно-технические документы, которые необходимо оформлять при разработке современных АИС.

В результате проведения первого этапа ЖЦ АИС "Системный анализ проекта ПС" должны быть разработаны концепция и техническое задание на создание АИС.

Концепция АИС должна содержать в себе:

определение целей, задач и функциональных характеристик системы;



————— Признак потенциальных возможностей реализации угроз качеству функционирования АИС;
 ===== Признак потенциальных возможностей пропуска реализованной угрозы (вследствие отсутствия соответствующих требований в ТЗ, заранее известных методик проверки выполнения требований ТЗ, отсутствия независимости проверяющих, недостатков их квалификации, ограниченного времени на проверку и пр.);
 ТП(ЭП)-технический (эскизный) проект; ОС- операционная система; ОПО – общесистемное ПО; СПО-специальное (прикладное) ПО; СУБД – система управления базой данных.

Рис. 8. Схема типового жизненного цикла АИС с учетом угроз безопасности

прогноз способа использования системы, квалификации кадров, ресурсов и времени полного жизненного цикла (концепции применения);

определение степени риска при реализации проекта, структуры коллектива разработчиков, специальных требований (концепции разработки);

непротиворечивая совокупность высказываний, истинных для данной предметной области, включая возможные состояния, классификации, законы, правила (концептуальная схема).

Техническое задание (ТЗ) – это основной исходный для разработчиков и заказчиков документ, который содержит систематизированное описание технических требований к проектируемой системе, а также порядок ее разработки и приемки в эксплуатацию.

Второй и третий этапы ЖЦ АИС – это этапы эскизного и технического проектирования, условно выделенная часть процесса разработки АИС, сводящаяся к выполнению одной или нескольких проектных процедур, общность которых определяется принадлежностью получаемых проектных решений к одному иерархическому уровню описания.

Эскизное проектирование – это предварительное проектирование, описывающее крупноблочную модель АИС на уровне описаний пользовательских интерфейсов и логических моделей данных. После согласования с заказчиком предварительного (эскизного) проекта и его технико-экономических показателей, могут быть произведены уточнения условий контракта и ряда требований ТЗ.

Современный подход предполагает на этапе эскизного проектирования создание **прототипа** - действующей модели АИС, созданной на базе эскизного проекта средствами CASE-системы. Прототип АИС предназначен для анализа, корректировки и согласования с заказчиками решений эскизного проекта и требований ТЗ на действующей модели АИС. Испытания прототипа - это совместное с заказчиками проведение имитационных прогонов АИС на базе прототипа методом проведения экспериментов, моделирующих основные моменты реальной эксплуатации АИС.

После уточнения ТЗ по результатам испытания прототипа переходят к этапу детального (технического) проектирования. **Технический проект АИС (ТП)** – это нормативно-технический документ (система документов), содержащий (щих) детальные технические решения по архитектуре АИС, всем компонентам АИС, точное описание функций АИС. В состав ТП АИС, как минимум, включаются следующие разделы (документы): общесистемная документация, функциональная часть, информационное обеспечение, программно-техническое обеспечение (платформа), организационно-правовое обеспечение.

После согласования технического проекта начинается этап рабочего проектирования, в который входит:

кодирование (программирование) и отладка программных компонент включает следующие работы;

документирование исходных и объектных текстов компонент, результатов их тестирования, качества и технических характеристик;

интеграция (комплексирование) и комплексная отладка программного обеспечения;

разработка и отладка информационной и физической компонент АИС;

разработка комплекта эксплуатационной документации для пользователей.

В рабочий проект входят следующие документы, содержащие сведения, необходимые для разработки, изготовления, сопровождения и эксплуатации программ:

спецификация программного обеспечения - точное и подробное

описание действий, которые должна выполнять система программного обеспечения АИС, причем основное внимание уделяется тому, что ПО должно выполнять, а не тому, как это должно выполняться;

программная спецификация - точное описание того результата, который должен быть достигнут с помощью программы, может иметь форму отображения набора входных данных в набор выходных данных;

программное изделие - программа на носителе данных, являющаяся продуктом промышленного производства;

исходные тексты программ - запись программ на исходном языке программирования с необходимыми комментариями;

описание программы - сведения о логической структуре и функционировании программы;

пояснительная записка к программе - схема алгоритма, общее описание алгоритма и функционирования программы, обоснование принятых технических решений;

описание применения программы - сведения о назначении программы, области и методах применения, классе решаемых задач и ограничениях и о программно-технической платформе;

руководство программиста - сведения, необходимые для установки, настройки и эксплуатации программы;

руководство пользователя - программный документ, содержащий сведения для обеспечения процедуры общения пользователя с исполнительной системой в процессе выполнения программы.

Этап испытания и документирование АИС включает следующие работы: подготовка к приемке АИС - разработка программы приемо-сдаточных испытаний на соответствие характеристикам, заданным в ТЗ и контракте; документирование результатов приемо-сдаточных испытаний актом приемки АИС заказчиком;

оформление полного комплекта документов; проведение сертификации на соответствие спецификации требований и эксплуатационной документации.

2.2. Показатели качества, эффективность и надежность АИС

Качество любого изделия представляется набором показателей, отражающих его свойства и определяющих возможность и эффективность его применения по прямому назначению. Чем сложнее изделие, тем большее число характеристик описывают его свойства и тем больше необходимо показателей для адекватного отражения его качества. Качество АИС описывается совокупностью показателей - критериев, для каждого из которых должны быть определены метрики и методы их измерения. Эти критерии и метрики АИС позволяют описывать их свойства как конечного продукта независимо от способа их достижения.

Качество АИС – это система показателей, отражающих свойства АИС и определяющих возможность и эффективность применения АИС по прямому назначению, зафиксированному в техническом задании на разработку. Оно достигается, прежде всего, за счет высокого качества технологических процессов при их создании и управления этими процессами.

Затраты на разработку резко возрастают, когда показатель качества приближается к пределу, достижимому при данной технологии и уровне автоматизации процесса разработки. При переходе к принципиально иной, более прогрессивной технологии при тех же затратах достижимо более высокое качество. Следовательно сложность АИС и допустимые затраты ресурсов становятся косвенными критериями или факторами, влияющими

на качество.

Основными показателями и критериями качества АИС являются следующие:

научно-технический уровень - степень использования в системе технических решений, отвечающих современным научно-техническим достижениям;

уровень автоматизации - степень использования автоматизированных информационных технологий при выполнении функций АИС;

эффективность - свойство системы, заключающееся в выполнении предписанных функций, с учетом соотношения затрат с результатами;

экономичность - количество и степень занятости ресурсов: реализующей ЭВМ, АРМов пользователей, каналов сети, и др.;

функциональные критерии качества - показатели, отражающие специфику областей применения и степень соответствия АИС их основному целевому назначению;

конструктивные критерии качества - показатели, отражающие эффективность использования информационными технологиями ресурсов вычислительных средств, а также надежность и другие общие характеристики функционирования АИС.

В существующих стандартах описывается до 30 показателей и приводятся их определения. Однако значения многих показателей могут определяться только экспертно. В процессе разработки контроль всей совокупности показателей нецелесообразен, так как конструктивное влияние на процессы разработки и применения оказывают только некоторые из них. В зависимости от класса, функций и особенностей АИС целесообразно выбирать различные наборы критериев, адекватные свойствам конкретной АИС. Таких критериев, которые должны наиболее полно отражать назначение и функциональные характеристики АИС, обычно не более 10. Каждый из основных критериев может быть описан еще несколькими частными показателями, характеризующими основные свойства объекта или процесса в разных аспектах.

В теории надежности работоспособным называется такое состояние объекта, при котором он способен выполнять заданные функции с параметрами, установленными требованиями технической документации. Другими словами, **надежность АИС** – это свойство системы сохранять во времени в установленных пределах значения всех характеристик, определяющих способность системы выполнять требуемые функции в условиях заданных режимов эксплуатации.

Надежность является внутренним свойством систем, проявляющимся только во времени, а следовательно критерии качества становятся динамическими и преимущественно стохастическими, характеризующими функционирование АИС в целом. Измеряемые интегральные показатели качества программ в этом случае более определенные и могут достаточно точно оцениваться экспериментально.

Ключевым понятием в теории надежности является **отказ** - событие, заключающееся в нарушении хотя бы одного из требований к качеству выполнения одной или нескольких функций, установленных в нормативно-технической и (или) конструкторской документации на систему. Соответственно **устойчивость АИС** – это способность АИС к безотказному функционированию при наличии сбоев и отказов и реагирование на проявления ошибок так, чтобы это не отражалось на показателях надежности, безопасности и эффективности, **работоспособность АИС** – это состояние АИС, при котором она способна выполнять заданные функции с параметрами, установленными требованиями технической документации.

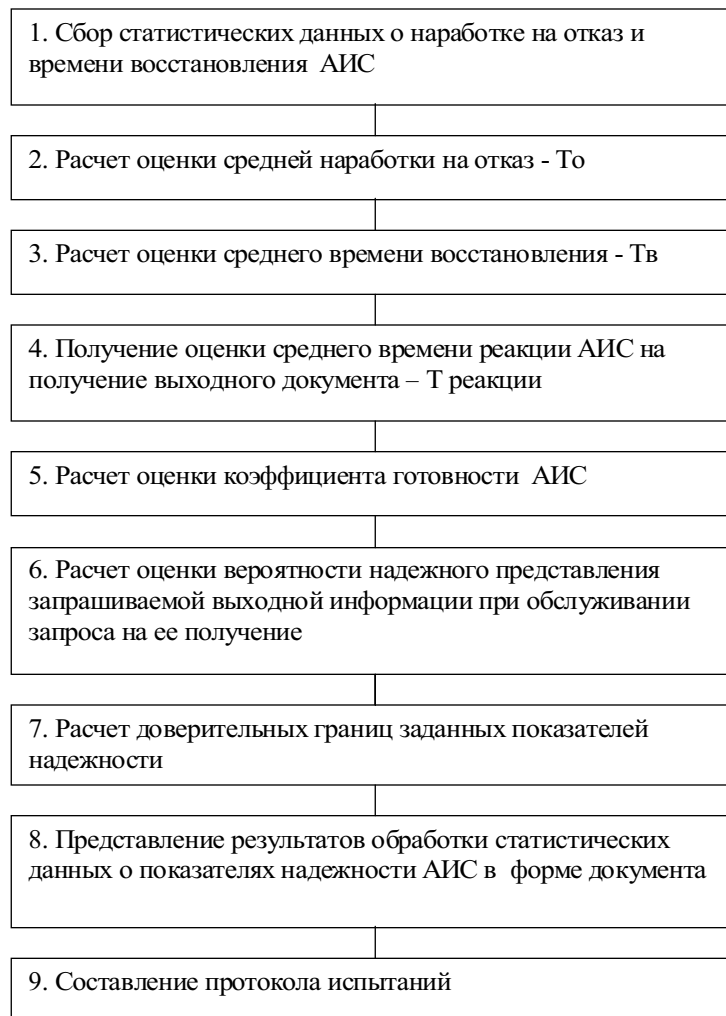


Рис. 9. Блок схема испытаний АИС для получения характеристик надежности

Устойчивость зависит от уровня неустраненных ошибок и способности АИС реагировать на проявления ошибок так, чтобы это не отражалось на показателях надежности. Последнее определяется:

- эффективностью контроля за доступом к данным;
- степенью обеспечения их секретности и сохранности;
- отбором достоверных данных, поступающих из внешней среды;
- средствами обнаружения аномалий процесса функционирования.

Показатели надежности АИС сходны с аналогичными характеристиками, принятыми для других промышленных изделий. Широко используется критерий **длительности наработки на отказ** - время работоспособного состояния системы между последовательными отказами или началами нормального функционирования системы после них. Вероятностные характеристики этой величины в нескольких формах используются как разновидности критериев надежности.

Наработка на отказ учитывает ситуации потери работоспособности АИС, когда длительность восстановления относительно велика. Качество АИС более полно отражает **наработка на отказовую ситуацию** - длительность между потерями работоспособности независимо от того, насколько быстро произошло восстановление. Обобщение отказов и восстановлений производится в критерии **коэффициент готовности**, который отражает вероятность иметь восстанавливаемую систему в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы системы на достаточно большом интервале, содержащем отказы и восстановления.

Применение основных понятий теории надежности для оценки качества сложных АИС позволяет получить ряд четких, хорошо измеряемых показателей качества. Приведенные критерии используются в основном при испытании АИС (см. рис. 9) и на завершающих фазах комплексной отладки.

Информационная компонента АИС (базы данных) имеет следующие показатели качества:

полнота - относительное число описаний объектов, хранящихся в базе данных, к общему числу объектов соответствующей предметной области или по отношению к числу объектов в аналогичных БД по той же предметной области;

достоверность - степень соответствия данных об объектах в БД реальным объектам в соответствующей предметной области в данный момент времени;

актуальность - относительное число морально устаревших данных об объектах в базе данных к общему числу накопленных и обрабатываемых данных.

оперативность - степень соответствия динамики изменения данных при обновлении базы данных состоянию объектов соответствующей предметной области или величина запаздывания между появлением или изменением характеристик реального объекта и их отображением в базе данных.

2.3. Сертификация АИС

Сертификация информационных технологий – это испытание ИТ в наиболее жестких условиях особо выделенным коллективом специалистов, имеющим право на официальный государственный или ведомственный контроль функций и качества ИТ, с целью установления соответствия сертифицируемых ИТ стандартам или другим нормативным документам.

Эти специалисты имеют право на расширение условий испытаний и на создание различных критических и стрессовых ситуаций в пределах нормативной документации, при которых должно обеспечиваться заданное качество и безопасность результатов решения предписанных задач.

Если все испытания проходят успешно, то оформляется специальный документ - **сертификат соответствия**. Этот документ официально подтверждает соответствие стандартам, нормативным и эксплуатационным документам функций и характеристик испытанных средств, а также допустимость их применения в определенной области.

Основная цель сертификации – это защита интересов пользователей, государственных и ведомственных интересов на основе контроля качества продукции, обеспечение их высоких потребительских свойств, повышение эффективности затрат в сфере их разработки, производства, эксплуатации и сопровождения, повышение объективности оценок характеристик и обеспечение их конкурентоспособности.

Методически процесс сертификации представляет особую совокупность испытаний компонент АИС, для которых необходимы специальные стандарты, методики, средства автоматизации и подготовленные специалисты. Работы по сертификации объединяются в технологический процесс, на каждом этапе которого регистрируются документы, отражающие состояние и качество результатов разработки. Процесс сертификации отличается от обычных испытаний более высоким уровнем формализации и документального оформления всех условий и результатов испытаний, проводимых специальным испытательным органом.

Процесс сертификации требует решения ряда следующих научных и методических задач:

- необходимо определить представительный набор характеристик и их значений, требуемую достоверность измерения показателей качества и организационный уровень удостоверения сертификата;

- в соответствии с требованиями к достоверности показателей качества должны определяться и минимизироваться содержание и объемы сертификационных испытаний;

- должны быть разработаны эффективные методы и методические нормативные документы, регламентирующие процессы сертификации;

- типовой технологический процесс сертификационных испытаний должен поддерживаться эффективными средствами автоматизации.

Исходные данные для сертификации опираются на совокупность документов, выбираемых и адаптируемых с учетом конкретных объектов сертификации. Наиболее общие исходные данные сосредоточены в стандартах, посвященных сертификации, аттестации, тестированию, испытаниям, документированию и обеспечению качества различных изделий и в частности компонент АИС.

Методология принятия решений о допустимости выдачи сертификата на АИС основывается на оценке степени ее соответствия действующим и/или специально разработанным документам:

- действующим международным и национальным стандартам на тестирование, испытания, аттестацию программ и баз данных, требования которых не ниже требований, регламентируемых отечественными документами;

- международным и государственным стандартам на технологию создания компонент АИС;

- стандартам на сопровождающую документацию с учетом необходимости и достаточности номенклатуры документов, семантической полноты и однозначности понимания содержания документов;

- нормативным документам - техническим условиям, техническим описаниям, спецификациям требований и другим регламентирующим документам на АИС по выбору заказчика, разработчика и испытателя.

Заключительные сертификационные испытания АИС в реальной системе или имитированной внешней среде специально аттестованным центром являются основанием для выдачи сертификата соответствия.

2.4. Безопасность АИС

2.4.1. Концепция и основные понятия безопасности АИС

В процессе сертификационных испытаний необходимо также решить задачи обеспечения безопасности АИС различного назначения. **Безопасность** – это понятие, характеризующее способность системы обработки данных обеспечивать защиту, достоверность и надежность хранения информации. В случае непредумышленных угроз, вводится понятие **технологической безопасности АИС**, т.е. системы методов и средств предотвращения отказов и выявления безопасности функционирования при случайных, дестабилизирующих воздействиях и отсутствии злоумышленного влияния на АИС, а также снижения воздействия этих угроз до допустимого уровня, который определяется соответствующими нормативно-техническими документами.

Проблемы **защиты** (средства для ограничения доступа или использования всей АИС или ее части) имущественных прав на информацию возникли в связи с превращением информации в объект собственности. В этом случае формируются следующие цели защиты информации в АИС:

- предотвращение утечки, хищения, утраты, несанкционированного уничтожения, искажения, модификации (подделки), несанкционированного копирования, блокирования информации;

- сохранение полноты, достоверности, целостности информации и программ обработки, установленных собственником или уполномоченным им лицом;

- обеспечение конституционных прав граждан на сохранение личной тайны и конфиденциальности персональной информации, накапливаемой в базах данных;

- сохранение секретности, конфиденциальности информации в соответствии с действующим законодательством;

- соблюдение прав авторов программной и информационной продукции, используемой в информационных системах.

В деятельности по применению мер и средств защиты информации выделяются следующие самостоятельные направления:

- защита информации от несанкционированного доступа;

- юридическая защита электронных документов;

- защита от компьютерных вирусов и других опасных воздействий по каналам распространения программ;

- защита от несанкционированного копирования и распространения программ и ценной компьютерной информации.

Защита конфиденциальной и ценной информации от несанкционированного доступа и модификации технически сводится к задаче разграничения функциональных полномочий и доступа к АИС. При этом под словом “доступ” следует понимать не только возможность потенциального нарушителя “читать” хранящуюся в компьютере информацию, но и возможность модифицировать ее. Требования по защите в различных приложениях могут существенно различаться, однако они всегда направлены на достижение трех основных свойств защищаемой информации:

- конфиденциальность** - засекреченная информация должна быть доступна только тому, кому она предназначена;

- целостность** - информация, на основе которой принимаются важные решения, должна быть достоверной и точной, а также защищена от возможных непреднамеренных и злоумышленных искажений;

- готовность** - информация и соответствующие автоматизированные

службы должны быть доступны, готовы к обслуживанию всегда, когда в них возникает необходимость.

Защита юридической значимости электронных документов оказывается необходимой при обработке, хранении и передачи сообщений, файлов, баз данных, содержащих в себе приказы, платежные поручения, контракты и другие распорядительные, договорные, финансовые документы. Их общая особенность заключается в том, что в случае возникновения споров (в том числе и судебных), должна быть обеспечена возможность доказательства истинности факта того, что автор действительно фиксировал акт своего волеизъявления в отчуждаемом электронном документе, а не был фальсифицирован каким - либо третьим лицом.

Для решения данной проблемы используются современные **криптографические** методы проверки подлинности информационных объектов, связанные с применением так называемых “цифровых подписей”. Эти методы основаны на включении в документы специальных меток, логически неразрывно связанных с его текстом, для порождения которых используется индивидуальный секретный криптографический ключ. При наличии индивидуальных криптографических ключей исключается возможность “подделки” таких меток со стороны других лиц. Цифровая подпись может неопровержимо свидетельствовать об авторстве того или иного конкретного лица и этот факт может быть проверен получателем.

Защита информации и ПС от **компьютерных вирусов** и других опасных воздействий по каналам распространения программ приобрела за последнее время особую актуальность. Масштабы реальных проявлений “вирусных эпидемий” оцениваются сотнями тысяч случаев “заражения” программ во всех странах. Хотя некоторые из вирусных программ оказываются вполне безвредными, многие из них имеют разрушительный характер. Особенно опасны вирусы для компьютеров, входящих в состав информационно-вычислительных сетей. Некоторые особенности современных АИС создают благоприятные условия для распространения вирусов. К ним, в частности, относятся:

- необходимость совместного использования ПС и БД многими пользователями;

- трудность ограничения в использовании программ и данных;

- ненадежность существующих механизмов защиты и разграничения доступа к информации в отношении противодействия вирусу.

Как правило, рассматриваются два направления в методах защиты от вирусов:

- применение “иммуностойких” программных средств, защищенных от возможности несанкционированной модификации;

- применение специальных программ-анализаторов, осуществляющих постоянный контроль возникновения “аномалий” в деятельности прикладных программ, входную и периодическую проверку наличия других возможных следов вирусной активности.

Защита от несанкционированного копирования и распространения программ и ценной информации является самостоятельным видом защиты имущественных прав, ориентированным на охрану интеллектуальной собственности. Эта защита обычно осуществляется с помощью специальных программ, подвергающих защищаемые программы и данные предварительной обработке (вставка парольной защиты, проверок по обращению к устройствам хранения ключа и ключевым дискетам, блокировка отладочных прерываний, проверка рабочей ЭВМ по ее уникальным характеристикам и т.п.), которая приводит объектный код защищаемой программы в состояние, препятствующее его выполнению на “чужих” ЭВМ. Могут применяться дополнительные

аппаратные блоки (ключи), подключаемые к разъему принтера или к системной шине ЭВМ, а также производится шифрование файлов, содержащих объектный код программы.

Общим свойством средств защиты программ от несанкционированного копирования является ограниченная стойкость такой защиты, так как в конечном случае объектный код программы поступает на выполнение в центральный процессор в открытом виде и может быть прослежен с помощью аппаратных отладчиков. Однако это обстоятельство не снижает потребительские свойства средств защиты до нуля, так как основной целью их применения является в максимальной степени затруднить, хотя бы временно, возможность массового тиражирования новых версий ПС до появления последующих версий.

2.4.2. Методы и средства защиты в распределенных АИС

Количество распределенных АИС, которые основываются на компьютерных сетях постоянно растет. Сети позволяют совместно работать с информацией и другими ресурсами большому количеству пользователей в организациях самых разных масштабов. Часто информация, хранящаяся на сетевых серверах, является секретной и предназначена для ограниченного круга лиц. Предотвращение несанкционированного доступа к такого рода информации становится основой защищенности и конкурентоспособности организации. Основой защиты информации является структура безопасности: ее определение, реализация и управление. Система контроля над безопасностью АИС должна соответствовать риску, связанному с компонентами информационных технологий. Она должна отражать структуру деятельности и информационные потоки. Персонал должен знать процедуры и правила защиты информационных ресурсов и поддерживать высокую степень целостности системы.

Основой, для создания системы безопасности распределенной АИС является защищенная сетевая операционная система. Функции защиты должны быть включены в систему повсеместно. Файловая система, каталог учетных записей пользователей, система аутентификации пользователей, подсистемы, управление памятью, — все требует серьезной проработки и четкого планирования. Например, защита удаленного доступа: дозвон, сетевая маршрутизация, фильтрация сетевых протоколов или сервисов, сетевая регистрация и аутентификация, передача файлов, электронная почта, связь с Internet, - везде администратор системы должен решать проблемы защиты. Использование разработок сторонних фирм нередко усугубляет эти проблемы: дополнительно встают вопросы совместимости, мощности и наращиваемости.

В частности, для навигации в Internet возникает потребность в поддержке следующих возможностей:

- цифровая подпись программного кода, которая позволяет определить, кто является издателем программного обеспечения, еще до его загрузки (аналогично тому, как сегодня для этого применяются эмблемы и логотипы);

- отключение загрузки программного кода, если пользователь предпочитает не видеть объектов на Web-странице, таких как Java-приложения, сценарии или элементы управления;

- поддержка стандартов защиты Internet, так как надежные коммуникации и транзакции без непроизводительных потерь чрезвычайно важны для пользователей программы просмотра, должны поддерживать стандартные протоколы защиты;

- цифровые сертификаты для идентификации серверов и пользователей, что позволяет пользователям уникальным образом идентифицировать себя

на Web-узлах и обращаться к той информации, на которую они имеют соответствующие полномочия, а также идентифицировать сервер, запрашивая его сертификат;

возможности надежного и эффективного наращивания средств защиты самими пользователями и разработчиками, для чего программа просмотра должна иметь открытую архитектуру шифрования и обеспечивать замену средств защиты для разработки приложений и интеграции с новыми технологиями защиты.

Такие способы защиты, как запрос пароля с последующей передачей его в открытом виде по коммуникационному каналу, и простые списки доступа на серверах, становятся малоэффективными в условиях прогрессивной коммерциализации Internet. Примером технологии, предлагающей необходимые для применения в масштабах Internet универсальность и общность, может служить существующая на сегодняшний день в виде промышленного стандарта Sun Microsystems и проекта стандарта Internet спецификация, сокращенно называемая SKIP (Simple Key management or Internet Protocol - Простой протокол управления ключами в интерсети). Эта спецификация была разработана компанией Sun Microsystems и предложена в качестве стандарта Internet.

Протокол SKIP обладает следующими свойствами:

совместим с протоколом Internet - IP;

встраивается в IP-стек выше аппаратно-зависимой его части (аппаратно независим) и работает на всех тех каналах, на которых работает IP;

принадлежность SKIP к IP-стеку обеспечивает прозрачность этого протокола для приложений;

обрабатывает IP-пакеты, не накладывая никаких ограничений на вышележащее программное обеспечение, в свою очередь, приложения никак не "чувствуют" SKIP;

сеансонеависим: - для организации защищенного взаимодействия между парой абонентов не требуется никакого дополнительного информационного обмена и не требуется передачи по каналам связи какой-либо открытой информации;

обеспечивает шифрование (путем помещения пакетов, подлежащих защите, в SKIP- пакеты) и аутентификацию (достоверную идентификацию источника) информации.

Режимы инкапсуляции и шифрования могут применяться как совместно, так и раздельно.

Имеются программные реализации SKIP для ОС Solaris. Предлагаются версии продукта различного назначения от простой программы для защиты трафика оконечного устройства, работающего с одним выделенным сервером, до полнофункционального продукта защиты станции в корпоративной сети, который обеспечивает учет топологии сети и индивидуальную настройку правил взаимодействия с различными ее узлами. Кроме того, предлагается аппаратно-программный комплекс для коллективной защиты локальных сетей.

Другой пример – это система защиты, применяемая в Windows NT Server в соответствии с требованиями уровня защиты C2, определенными в издании Национального центра защиты компьютеров Министерства обороны США, известном как "Оранжевая книга". Независимо от того, являются ли ОС отдельно стоящими или сетевыми, в США они оцениваются по критериям, установленным в Оранжевой книге. Windows NT Server изначально разрабатывался в соответствии с требованиями Оранжевой книги.

Согласно требованиям уровня защиты C2 защищенная сетевая система характеризуется рядом параметров. Важнейшими требованиями уровня защиты C2 являются следующие:

владелец ресурса (например, файла) должен иметь возможность контроля

доступа к ресурсу;

операционная система должна защищать находящиеся в памяти компьютера и принадлежащие одному процессу данные от случайного их использования другими процессами (например, NT Server защищает участок памяти, занятый процессом, так, что его содержимое не может быть прочитано даже после того, как процесс освободил его);

при удалении файла с диска пользователи не должны иметь доступа к его данным, если дисковое пространство, ранее занятое удаленным файлом, выделяется для использования новым файлом;

каждый пользователь должен быть уникальным образом идентифицирован в системе, а система - иметь возможность применения этой идентификации для отслеживания всей деятельности пользователя;

администраторы системы должны иметь возможность аудита всех событий, связанных с защитой системы, а также действий отдельных пользователей, правами доступа к данным аудита должен обладать ограниченный круг администраторов;

система должна защищать себя от вмешательства, такого как модификация работающей системы или файлов, хранящихся на диске.

Кроме того, вводятся дополнительные требования, определяемые самой практикой - они относятся к управлению защитой и ее использованию. Среди них следующие:

контроль со стороны администратора за тем, какие и кем используются ресурсы;

возможность централизованного управления привилегиями и правами; возможность включения пользователей в группы, установки допустимого времени работы и т. п.;

аудит таких событий, как попытка регистрации, доступа к файлам, принтерам и т.д.;

блокировка учетных записей при неверной регистрации;

установление срока жизни и правил использования пароля.

Windows NT Server, разработанный в соответствии с требованиями уровня C2, предлагает эти дополнительные средства, как для управления, так и использования этих дополнительных требований. Имеется набор инструментов, облегчающих администраторам управление и поддержку системы защиты. Например, администратор может контролировать круг пользователей, имеющих права доступа к сетевым ресурсам: файлам, каталогам, серверам, принтерам и приложениям. Правами, определяемыми для каждого ресурса, можно управлять централизованно.

Учетные записи пользователей также управляются централизованно. С помощью простых графических инструментов администратор задает принадлежность к группам, допустимое время работы, срок жизни и другие параметры учетной записи. Администратор получает возможность аудита всех событий, связанных с защитой доступа пользователей к файлам, каталогам, принтерам и иным ресурсам. Система способна блокировать учетную запись пользователя при определенном числе неудачных попыток регистрации. Администраторы вправе устанавливать срок жизни паролей, принуждать пользователей к периодической смене паролей и вводу паролей, которые сложно вскрыть.

Простая процедура регистрации обеспечивает доступ к соответствующим ресурсам. Для пользователя невидимы такие процессы, как шифрование пароля на системном уровне, подразумевающее отсутствие передачи пароля в открытом виде по сети. Шифрование препятствует обнаружению пароля при несанкционированном просмотре сетевых пакетов. Пользователь определяет права доступа к ресурсам, которыми он "владеет". Например,

чтобы разрешить совместное использование своего документа, он может указать, кто и как именно может с ним работать. Разумеется, доступ к ресурсам АИС контролируется только администраторами с соответствующими правами.

Пример более глубокого использования защиты - способность системы защищать данные, находящиеся в физической памяти компьютера. Если данные больше не содержатся на диске, система предотвращает несанкционированный доступ к той области диска, где они содержались, и никакая программа не "подсмострит" информацию, с которой оперирует в данный момент другое приложение в физической памяти машины.

Добавлен набор функций, позволяющих разработчикам создавать приложения, способные безопасно исполняться в таких незащищенных сетях, как Internet. Специфицируются функции, которые позволяют:

приложениям подключаться к различным системам криптографии (CSP), выбирая определенный CSP по имени или в соответствии с требуемым классом функциональности;

приложениям создавать и настраивать криптографические ключи;

приложениям обмениваться ключами;

выполнять шифрование/дешифрацию данных в приложениях;

приложениям вычислять криптографически защищенные наборы данных, а также выполнять цифровую подпись данных.

3. ОРГАНИЗАЦИОННОЕ И ПРАВОВОЕ ОБЕСПЕЧЕНИЕ АИС

3.1. Организация работ по этапам жизненного цикла АИС

3.1.1. Организация разработки и эксплуатации АИС

Работы по созданию АИС различного назначения проводятся в тысячах фирм и государственных организаций по всему миру. Для подавляющего большинства АИС важнейшим требованием (наряду с требованиями эффективности и безопасности) является соблюдение единых для всех разработчиков требований к АИС с целью их интеграции в единую мировую информационную среду. С этой целью технология создания современных АИС включает в себя концепцию и стандарты открытых систем. Далее перечислены важнейшие признаки открытых систем, отражающих их основные (программно-технический, нормативный, информационный и пользовательский) аспекты.

Открытая система - это:

а) совокупность разнородных компьютеров, объединенных в сеть, которые могут работать как единое целое, независимо от того, где они расположены, как в них представлена информация и под управлением какой операционной системы они работают;

б) совокупность международных стандартов в области информационных технологий, которая специфицирует интерфейсы, услуги и поддерживающие форматы данных для достижения взаимодействия и переносимости приложений, данных и персонала;

в) система, которая реализует открытые спецификации на интерфейсы, службы и форматы данных, достаточные для того, чтобы обеспечить: возможность переноса прикладных систем с минимальными изменениями; совместную работу с другими прикладными системами на локальных и удаленных платформах; взаимодействие с пользователями в стиле, облегчающем переход от системы к системе.

Основными целями создания и применения концепции, методов и стандартов открытых систем являются:

повышение общей эффективности разработки и функционирования

информационных систем, а также логической и технической совместимости их компонент;

снижение трудоемкости, стоимости и длительности разработки сложных распределенных информационных систем, программных средств и баз данных;

сохранение инвестиций, вложенных в реализованные информационные системы, их программные средства и базы данных, в процессе развития архитектуры и ресурсов аппаратных и операционных платформ;

обеспечение высокого качества и надежности функционирования программных средств и баз данных в информационных системах (ИС).

Приступая к разработке АИС, необходимо принять одно из двух решений: разработать полностью новую информационную систему или найти соответствующий аналог и перенести его компоненты в требуемую операционную среду и на новую аппаратную платформу.

Обоснованное решение может быть принято в результате сопоставления основных технико-экономических показателей:

трудоемкости по всему жизненному циклу АИС;

длительности этапов разработки и внедрения;

необходимого числа специалистов для альтернативных вариантов.

Важным направлением сокращения затрат и повышения качества является активное использование методического, технологического, алгоритмического и программного задела из предшествующих проектов.

Проектирование АИС начинается с системного анализа предметной области и выявления потребности в создании АИС с определенными функциями. Методами математического моделирования создаются варианты, фрагменты и компоненты прототипа АИС и выделяются возможные методы реализации предполагаемых функций. Используются близкие по функциям аналоги, которые можно рассматривать как прототипы ПС в целом или его отдельных компонент. CASE-средства обеспечивают унификацию процессов моделирования, автоматизированный анализ системных требований и выработку первичных требований к предполагаемому проекту АИС.

На базе первичных требований появляется возможность оценить объем, подлежащих разработке компонентов и некоторые дополнительные характеристики возможного объекта и среды разработки. На этом этапе CASE-средства обеспечивают оценку возможной трудоемкости и длительности разработки, необходимого числа специалистов и других ресурсов. Полученные при системном анализе данные позволяют руководителям разработки и заказчикам принимать решение о целесообразности продолжения проектирования и осуществлять стратегическое планирование проекта, которое формализуется в техническом задании на ПС.

План управления разработкой АИС должен быть связан с планом управления обеспечением качества АИС по всему жизненному циклу. Эта связь выражается в одинаковом поэтапном представлении планов и в наличии в них значительной части идентичных процессов и документов. При подготовке планов и проведении работ целесообразно по возможности разделять их цели и функции. План управления разработкой следует ориентировать на организацию специалистов, непосредственно создающих компоненты и АИС в целом, на эффективное распределение и использование ими ресурсов и средств автоматизации. В плане управления обеспечением качества внимание специалистов направляется на анализ качества достигнутых результатов разработки, методах и средствах достижения заданных заказчиком характеристик ПС. Хотя план управления разработкой также должен способствовать созданию ПС с необходимым качеством, тем не менее практика показала, что разделение этих планов позволяет значительно повышать

эффективность процессов разработки ПС и достигаемое их качество.

Оценки проекта позволяют осуществить выбор основных CASE методов и средств для проведения последующего проектирования.

Затем выполняются следующие работы:

- производится необходимая адаптация средств автоматизации, применительно к особенностям объекта и среды проектирования;

- определяется структура базы данных проектирования и объемы основных массивов данных;

- разрабатываются руководства для специалистов, выделяемых на данный проект, и осуществляется их обучение;

- конкретизируется технология проектирования и оценивается потенциальная эффективность.

Анализ требований технического задания и проведенных технико-экономических оценок позволяет выполнить техническое проектирование АИС и предварительную оценку необходимых ресурсов физической компоненты для решения функциональных задач. **Разработка технического проекта** – это разработка спецификаций требований на функциональные группы программ и модули, выбор и освоение готовых, апробированных компонент из состава предыдущих проектов, удовлетворяющих разработанным спецификациям требований. CASE средства на этом этапе поддерживают автоматизированное проектирование спецификаций требований. Повышению эффективности разработки могут значительно способствовать заимствование из предыдущих проектов спецификаций прототипов, версий и отдельных компонентов. Для обеспечения удобства проектирования на этом этапе большое значение имеют графические методы визуализации технических решений и логического контроля проекта. **Разработка спецификации требований на АИС** заключается в формализации функций, условий внешней среды, требований к характеристикам и качеству решения задач.

Спецификации являются базой для детального планирования процесса разработки АИС и компонентов. На основе такого плана разрабатывается подробный график работ и выделяются ресурсы для реализации каждого этапа. График в последующем уточняется и корректируется в течение всего времени проектирования. Задача CASE-средств состоит в обеспечении удобства работы с такими графиками, их изменения, выявления критических путей и этапов работ. Руководители проектов должны иметь для управления полную и наглядную информацию о состоянии и развитии всех компонент проекта, об используемых ресурсах, об объектах или процессах, имеющих риск нарушения планов или технико-экономических показателей.

Техническое проектирование и создание основных спецификаций обеспечивают возможность выбора системы управления базой данных проектирования и распределения ресурсов хранилища данных проекта. Базы данных современных систем автоматизации проектирования (БДП) АИС имеют сложную структуру и несколько уровней защиты от несанкционированного доступа. Регламентирование доступа к БДП на запись обеспечивает сохранность наиболее важной информации и гибкость изменения данных в процессе проектирования. Наиболее важными информационными массивами в БДП являются:

- исходные документы - стандарты, руководства и документы регламентирующие и организующие весь процесс проектирования, развития версий и интерфейсы;

- прогнозируемые показатели начатого проекта;

- техническое задание, конкретные планы, доступные ресурсы, состояние и прогнозы процесса проектирования АИС и ее компонентов;

- описание структуры, спецификации на весь комплекс программ, на

группы программ, на информационные и программные модули и их интерфейсы;

рабочие версии текстов программ на языке программирования и в объектном коде для программных модулей, групп программ и ПС в целом;

модели и имитаторы для генерации тестов, программы обработки результатов, конкретные тестовые и эталонные значения, а также результаты испытаний АИС и компонентов, допустимые для переноса и повторного использования;

испытанные и аттестованные базовые версии ПС, пригодные для производства, с полным комплектом эксплуатационной и технологической документации;

журнал предполагаемых и реализованных изменений, и все данные необходимые для конфигурационного управления развитием АИС.

Далее начинается **разработка рабочего проекта** - разработка исходных текстов программных модулей, функциональных компонент и описаний данных в соответствии со спецификациями требований, методиками и стандартами, трансляция исходных текстов и устранение синтаксических и семантических ошибок, разработка комплекта эксплуатационной документации для пользователей и описаний компонентов характеристик АИС для предъявления заказчику на приемо-сдаточные испытания.

Современная технология рабочего проектирования делится на технологию разработки компонентов и на технологию сборки АИС из готовых компонентов.

Первая технология должна обеспечивать возможность создания модулей и групп программ в качестве комплектующих изделий с четким описанием функций и интерфейсов, а также гарантии качества, стандартов и соглашений о связях. Для этого компоненты подвергаются автономным испытаниям, а также снабжаются документацией, достаточной для их правильного применения в сборных ПС и при переносе.

Вторая технология использует программные и информационные компоненты (ПИК) как законченные комплектующие изделия и должна обеспечивать их корректную сборку и сопряжение в АИС. АИС подлежат аттестации и испытаниям на соответствие заданным требованиям заказчика. В этой технологии принципиально изменяются масштабы и сложность объектов, являющихся базовыми в технологическом процессе. Ими становятся программные, технические и информационные модули, в которых внутренняя структура скрыта и недоступна для изменения при сборке. Для обеспечения переноса и повторного использования испытанных модулей возрастает значение стандартов, правил и ограничений, определяющих сопряжение их в единую систему.

При рабочем проектировании АИС номенклатура и функции ПИК определяются в процессе системного и структурного проектирования. При этом могут отсутствовать прямые прототипы АИС и основой для проектирования являются концепция и техническое задание, а также системные прогнозы возможного развития аналогичных АИС. Проектирование ведется по принципу сверху-вниз с последовательной детализацией функций компонентов и каждого модуля. Стандартизация и унификация компонентов и их интерфейсов базируется на системе правил и ограничений их структурного построения, обращения с данными, межмодульного интерфейса, оценки качества и документального оформления.

Для однозначного понимания и описания функций программных, физических и информационных компонент, функциональных групп программ и структур данных необходима унифицированная система именования и

идентификации функций ПИК. Эта система должна обеспечивать полное и корректное выделение необходимых ПИК из множества подобных, различающихся отдельными характеристиками, без необходимости дополнительного анализа конкретных алгоритмов, текстов программ и детальных описаний данных.

Описания спецификаций требований должны содержать иерархически упорядоченную совокупность характеристик компонентов, обеспечивающую каталогизацию ПИК, автоматический поиск группы наиболее подходящих из них по их описаниям и окончательный подбор компонентов с учетом всей номенклатуры функциональных и технических показателей. Информационные модули распределяются по уровням доступности (видимости) из программных компонент и должны иметь описания, стандартизированные по форме и структуре содержания. Целесообразно использовать унифицированные описания типовых структур данных, выделяемых в информационные повторно используемые компоненты.

Сборки АИС должна начинаться, когда разработаны и имеются в базе данных проектирования большинство ПИК, отлаженных в составе базовых версий, основой технологии является их перенос и сборка в новой версии ПС. Почти всегда выявляется необходимость создания некоторого количества новых ПИК, но наличие основной части отработанных ПИК значительно облегчает и процесс разработки необходимых новых ПИК.

Основой технологии сборочного проектирования АИС является развивающаяся база данных проектирования АИС и используемых программных и информационных компонент, апробированных и пригодных для многократного применения. База данных проектирования по мере необходимости пополняется новыми или обновленными версиями старых компонент, с расширенными или модифицированными функциями. В отдельных разделах базы данных проектирования накапливаются и хранятся базовые версии компонентов. Эти версии оформляются вместе с тестами и результатами испытаний, а также с полной документацией, необходимой для применения, переноса и сопровождения.

Иерархическая модульная структура АИС обеспечивает технологию проектирования по принципу сверху-вниз с позиции учета функционального назначения и наилучшего решения целевой задачи всей системы. Это способствует концептуальному единству компонентов и возможности рационального распределения ограниченных ресурсов проектирования по мере декомпозиции компонентов. Иерархическое построение АИС позволяет ограничить и локализовать на каждом из уровней соответствующие ему компоненты, вследствие чего облегчается анализ и синтез компонентов более высоких уровней и системы в целом.

Сборка базовой версии АИС завершается комплексной отладкой и испытаниями в реальной внешней среде или при ее имитации. Для этого необходимо создавать модели внешней среды, способные генерировать детерминированные и стохастические тестовые данные. Примененные при испытаниях модели внешней среды, исходные данные при генерации базовой совокупности тестов, а также результирующие данные испытаний каталогизируются и накапливаются в базе данных.

После отладки оформляется комплект эксплуатационной документации, обеспечивающей ее поставку пользователям и квалифицированное использование. Затем, по заранее подготовленной программе, проводятся **приемо-сдаточные испытания** на соответствие функциональным и техническим характеристикам, заданным в контракте и согласованным с заказчиком.

Поддержка процесса эксплуатации АИС заключается в накоплении

и обработке отчетов пользователей о результатах эксплуатации, анализе категорий выявленных ошибок и разработке предложений по совершенствованию и развитию функций АИС. Развитие, сданной в эксплуатацию АИС, заключается в оперативном и долгосрочном, контролируемом изменении при расширении и модификации требований к ней, а также с целью повышения качества функционирования. При недостаточной системной разработке, в процессе эксплуатации возможны значительные отклонения реальных требований от требований ТЗ. В этом случае важно установить приоритет, чтобы упорядочить процессы внесения изменений в различные компоненты. Неупорядоченные корректировки ПИК могут приводить к необходимости значительного изменения АИС в целом. Поэтому процессы изменения необходимо поддерживать методами и средствами конфигурационного управления, обеспечивающими анализ, учет и регистрацию всех изменений.

Причины для изменений должны конкретизироваться, формализоваться и накапливаться в базе данных вместе с характеристиками источников их появлений. Необходимо оценивать, насколько данное изменение может расширить и улучшить эксплуатационные характеристики АИС в целом, а также каковы возможные затраты на выполнение изменений.

При системном анализе трудно регламентировать необходимый перечень работ и график их выполнения. Этот этап имеет наиболее творческий характер и сильно зависит от функционального назначения и степени новизны создаваемой АИС. Наиболее четкому регламентированию во времени доступны работы от формирования технического задания до завершения испытаний и сдачи системы заказчику. Для их проведения необходима поддержка методами и средствами автоматизации технологического процесса.

3.1.2. Организация коллектива разработчиков и взаимодействие с Заказчиком

В настоящее время (в отличие от недавнего прошлого) в нашей стране появилось понятие “информационный бизнес”. Организация информационного бизнеса направлена, в первую очередь, на сохранение и повышение конкурентоспособности фирмы-производителя новых информационных технологий. Основу таких фирм составляют постоянные или временные коллективы разработчиков. Организационные структуры таких фирм упрощаются, а потоки управления “спрямляются”, делая возможным эффективное принятие решений. Так как технология является стратегической составляющей их бизнеса, следует начинать с выяснения того, как должна быть устроена структура информационных потоков и управления проектами.

Осознание новых правил ведения информационного бизнеса на базе систем, которые обеспечивают разработку гибких решений и быстрого реагирования на изменения в окружающей среде, требует нового подхода к организации дела в индустрии программных средств. В распоряжении разработчиков имеются концепции и стандартные модели, испытанные и проверенные группами разработки крупнейших мировых фирм-производителей ПС. Эти методы отличает альтернативный подход, обеспечивающий гибкость, масштабируемость и ясность. Так, для улучшения организации разработки и повышения производительности необходимо выяснить:

масштабируемую модель группы по реализации проекта, определяющую равнозначные роли и распределение ответственности, необходимые для полноты и успешной работы группы;

модель процесса разработки, определяющую основные этапы проекта, являющуюся итеративной, нацеленной на результат, и управляющую планированием рисков;

сервисную модель приложения, предлагающую рассматривать приложение как множество сервисов или услуг, каждый из которых содержит повторно используемые свойства и функциональные возможности;

процесс разработки приложения, определяющий жизненный цикл через создание сценариев использования, определение бизнес-объектов и построение физических компонентов.

Специалистов, которые включаются в группы разработчиков можно оценивать численностью, а также тематической и технологической квалификацией. В группах создания, испытания и сопровождения сложных АИС участвуют системные аналитики, руководители различных рангов, программисты, администраторы баз данных и вспомогательный обслуживающий персонал в некотором рациональном сочетании. Например, для системного анализа проекта требуются специалисты наиболее высокой квалификации (системные аналитики), так как их возможные ошибки наиболее сильно отражаются на эффективности и качестве всего жизненного цикла ПС. Потребность в разработке и функции АИС могут также определяться независимо заказчиком, что позволяет сокращать системный анализ у разработчика.

Определяющими являются совокупная численность и структура коллектива, а также его подготовленность к совместной работе по созданию конкретного типа АИС.

Тематическую квалификацию специалистов приблизительно можно характеризовать средней длительностью работы в данной проблемной области основной части коллектива, непосредственно участвующей в процессах разработки. Важную роль играет квалификация руководителей и системных аналитиков в конкретной прикладной области. Особенно важна не индивидуальная характеристика каждого специалиста, а прежде всего интегральный показатель бригады, выполняющей реализацию некоторой, достаточно крупной функциональной задачи. При низкой тематической квалификации допускаются наиболее грубые системные ошибки, требующие больших затрат при последующей доработке АИС или делающие их практически не приемлемым для пользователей.

Технологическая квалификация коллектива характеризуется опытом и длительностью работы с регламентированными технологиями и комплексами автоматизации. Особое значение имеет коллективный опыт организации и выполнения сложных проектов на базе современных автоматизированных технологий. Опыт применения конкретного программно-технического комплекса может являться существенным фактором при последующем выборе технологии для создания новых версий АИС или компонентов. Характеристика групп специалистов-разработчиков во многом справедлива и в отношении коллективов **центров сертификации АИС** - проблемно-ориентированных, независимых от разработчиков, ведомственных или государственных организаций, которые аккредитованы с правом проведения испытаний и выдачи сертификатов соответствия.

3.2. Защита авторских прав, рынок компонентов АИС и их интеграция

Под **интеллектуальной собственностью** понимаются результаты интеллектуального творчества человека, выраженные в материальной форме (записанные или нарисованные на бумаге, воплощенные в конструкции или

изделии и т.п.). Технические решения по информационным технологиям, в том числе права в отношении результатов интеллектуальной деятельности при разработке и совершенствовании АИС и ее компонентов, также являются интеллектуальной собственностью. Эти права определяются **правовым обеспечением АИС** - системой юридической документации, определяющей правовые нормы - как в коллективах, разрабатывавших и эксплуатирующих АИС, так правовые отношения между этими коллективами и внешней средой.

Объекты авторского права в АИС - это все результаты процесса создания и развития АИС: данные системного анализа, программные средства и спецификации, базы данных и т.п. Авторское право на компоненты АИС, как и на другие объекты авторского права, возникают в силу их создания.

В новой редакции Уголовного кодекса РФ есть статьи, позволяющие применить более суровые меры наказания к лицам, которые совершают хищение интеллектуальной собственности. Незаконное использование объектов авторского права или смежных прав, а равно присвоение авторства, если эти деяния причинили крупный ущерб, - наказываются штрафом в размере от двухсот до четырехсот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до четырех месяцев, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо лишением свободы на срок до двух лет.

Те же деяния, совершенные неоднократно либо группой лиц по предварительному сговору или организованной группой, наказываются штрафом в размере от четырехсот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от четырех до восьми месяцев, либо арестом на срок от четырех до шести месяцев, либо лишением свободы на срок до пяти лет.

Неправомерный доступ к охраняемой законом компьютерной информации, то есть информации на машинном носителе, в электронно-вычислительной машине (ЭВМ) или их сети, если это деяние повлекло уничтожение, блокирование, модификацию либо копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети, наказывается штрафом в размере от двухсот до пятисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до пяти месяцев, либо исправительными работами на срок от шести месяцев до одного года, либо лишением свободы на срок до двух лет.

То же деяние, совершенное неоднократно либо группой лиц по предварительному сговору или организованной группой либо лицом с использованием своего служебного положения, а равно имеющим доступ к ЭВМ, системе ЭВМ или их сети, - наказывается штрафом в размере от пятисот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от пяти до восьми месяцев, либо исправительными работами на срок от одного года до двух лет, либо арестом на срок от трех до шести месяцев, либо лишением свободы на срок до пяти лет.

В настоящее время этими проблемами занимается Управление по экономическим преступлениям МВД России. Законодательная база и судебная практика уже сегодня позволяют пресечь деятельность фирм, занимающихся распространением ворованных программ.

Главными источниками поступления ворованного программного обеспечения потребителю являются торговые организации. Это прежде всего продавцы компьютеров и компакт-дисков CD-ROM. Прилавки многочисленных ларьков завалены подделками, организации используют программное обеспечение без необходимых лицензий на него, продавцы компьютеров в качестве "бесплатного приложения" устанавливают программы без

соответствующего разрешения правообладателей.

В то же время, приобретение не лицензионных ПС может обернуться техническими проблемами и дополнительными расходами. В итоге, получив “в подарок” операционную систему, например, с неполным набором драйверов или системой подсказок, пользователь первым делом обвиняет в недоработках производителя. Неприятности происходят и в случае установки программ с дисков, купленных на рынке, а не у сертифицированных дилеров (по данным Службы технической поддержки Microsoft, в России примерно 30% звонков с вопросами приходится на пользователей нелегального ПО и такие пользователи технической поддержки не получают).

Легализация программных продуктов позволяет решать как текущие, так и перспективные задачи компании, при этом **лицензионно чистый (легальный) продукт** – это программный продукт, который содержит лицензионное соглашение, являющееся первым признаком, по которому можно определить легальность продукта.

Если при осмотре компьютера вы не найдете специального пакета с документацией на установленное ПО и сертификата подлинности на него, то вам, скорее всего, установили нелегальную операционную систему. А это потенциальная возможность получить у такого торговца или неполную версию продукта, или дефектный диск, или букет вирусов. Слишком велика вероятность того, что ПО, приобретенное таким образом может стать причиной сбоев в работе информационной системы или потери важных данных.

Легальные покупатели получают преимущества при покупке дополнительных лицензий на право установки программ по мере роста масштабов сети, при обновлении ПС по мере выхода новых версий, техническую и информационную поддержку фирмы изготовителя (будущий специалист также может стать изготовителем и продавцом собственных ПС). Таким образом, приобретение лицензий экономически более выгодно, чем покупка коробок с продуктами.

Вот почему знание “правил игры” на рынке ПС является важной составляющей в системе знаний специалиста по информационным технологиям.

Для защиты интересов производителей ПС создана Ассоциация производителей программного обеспечения (BSA), одной из целей которой является борьба с использованием и распространением ворованного ПО. В случае выявления фактов использования нелегальных продуктов юристы BSA подают иски на компании-нарушители. Проводятся рейды в фирмах, продающих диски CD-ROM с незаконными копиями программ и фирмах, устанавливающих “пиратские” копии на продаваемые компьютеры. Ниже приводится ряд положений, определяющих “правила игры” на рынке ПС.

Лицензионное соглашение является первым признаком, по которому можно определить легальность продукта. Язык лицензионного соглашения соответствует языковой версии купленного продукта. Так, например, купив русскую версию ОС Microsoft Windows 95, вы найдете в коробке лицензионное соглашение на русском языке. Иногда в комплект поставки входит лицензионное соглашение сразу на нескольких языках.

Независимо от формы приобретения продукта лицензия всегда сопровождает продукт или в форме отдельного документа, или в виде врезки в Руководство Пользователя, или в виде специально выделенного текста на упаковке с диском (дискетами), или в виде специального файла. Например:

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ MICROSOFT

ВАЖНО – ПРОЧИТАЙТЕ ВНИМАТЕЛЬНО!

“Настоящее лицензионное соглашение: (далее Соглашение) является

юридическим документом, оно заключается между вами (физическим или юридическим лицом) и корпорацией Microsoft Corporation (далее: корпорация Microsoft) относительно указанного выше программного продукта корпорации Microsoft (далее: Программа или Программное обеспечение), включающего в себя программное обеспечение, записанное на соответствующих носителях, любые печатные материалы и любую встроенную или электронную документацию.

Устанавливая, копируя или иным образом используя Программу, вы тем самым принимаете на себя условия настоящего Соглашения. Если вы не принимаете условий данного Соглашения, то незамедлительно верните неиспользованную Программу туда, где вы ее приобрели, и вам будут полностью возмещены все уплаченные суммы.

Программа защищена законами и международными соглашениями об авторском праве, а также другими законами и договорами, относящимися к интеллектуальной собственности.

Программа лицензируется, а не продается.

Настоящее Соглашение дает вам следующие права.

Программное обеспечение.

Вы можете установить и использовать одну копию Программы или вместо нее любую предыдущую версию этой же Программы на отдельном компьютере. Основной пользователь того компьютера, на котором установлена Программа, имеет право установить вторую копию на переносной или домашний компьютер исключительно для своего использования.

Хранение и использование в сети.

Разрешается также хранить или установить копию Программы на устройстве (например, на сетевом сервере) для того, чтобы дать возможность установить или запустить Программу на других компьютерах через внутреннюю сеть. Однако для каждого компьютера, на котором Программа установлена или на котором она запущена с сетевого сервера, необходимо приобрести отдельную лицензию. Лицензия на Программу не допускает совместного использования или одновременного использования на разных компьютерах.

Лицензионный Пакет.

Если Соглашение входит в состав Лицензионного пакета Microsoft, то разрешается сделать дополнительные копии программного обеспечения, количество которых указано в Соглашении. Каждой копией разрешается пользоваться так, как это описано выше. Пользователь имеет право установить дополнительные копии на переносной или домашний компьютер, как это описано выше".

Итак, приобретая лицензированное программное обеспечение пользователь:

получает гарантию того, что купил полную версию, и производитель гарантирует ее работоспособность и обеспечивает техническую поддержку;

получает ПС без вирусов, которые часто присутствуют на незаконно изготовленных копиях;

в коробке с ПС есть Руководство Пользователя, которое значительно сокращает время освоения продукта и помогает приобрести основные навыки в работе и настройке продукта под ваши конкретные задачи.

Регистрация и последующее получение Свидетельства о регистрации позволит пользоваться услугами Службы технической поддержки по телефону. Зарегистрированные пользователи имеют возможность приобретения последующих версий продукта со скидкой.

Прежде чем купить компьютер с предустановленным программным обеспечением, необходимо проверить:

наличие в комплекте поставки ПК наличие отдельного пакета,

включающего сертификат подлинности и документацию на продукт;
на сертификате должны быть нанесены защитная голограмма и надпись
“Только для поставки (или продажи) с персональным компьютером”;
продавец должен подтвердить факт наличия договорных отношений с
производителем ПО и показать копию соответствующего контракта, на
основании которого производится установка ПО.

Существует специальное программное обеспечение, позволяющее
сетевым администраторам вести текущий сбор данных о программном
обеспечении, установленном на сетевых ПК, и проводить их анализ.

Сбор данных осуществляется автоматически через установленные
промежутки времени и в соответствии с заданными параметрами. Результаты
сбора представляются в виде специального отчета, по которому можно
определить факт наличия на сетевых ПК тех программных продуктов, на
которые в организации не имеется соответствующих корпоративных лицензий.

Сетевой администратор должен иметь точные данные о количестве ПК в
организации и о программном обеспечении, которое на них установлено. В
первую очередь это касается операционных систем и офисных приложений
(текстовых редакторов, электронных таблиц, баз данных), а также
специализированного программного обеспечения. Если у фирмы уже имеется
одна или несколько купленных коробок с ПО, достаточно закупить только
корпоративные лицензии на право его установки в организации.

Составляется карта, в которой будет учтено уже имеющееся ПО
(легальное и нелегальное). Затем необходимо рассчитать требуемое количество
лицензий, которое будет приобретаться в рамках программы легализации
компаний. Необходимо рассчитать примерную стоимость закупаемого ПО и
лицензий и познакомиться с условиями лицензионного соглашения. После
этого стороны проводят детальный расчет стоимости контракта и его
оптимизацию.

Специальные скидки предусмотрены для учебных заведений. Существует
тип лицензий — Academic. Организации, подпадающие под статус учебных
заведений, имеют право на приобретение таких лицензий по специальным
ценам. Право приобретать академические лицензии имеют следующие
организации:

все учебные заведения, имеющие лицензии на ведение образовательной
деятельности, выданные центральным органом государственного управления,
курирующим образование;

академические институты, финансируемые из государственного бюджета;
публичные библиотеки;

музеи;

больницы и клиники при высших учебных заведениях.

Для эффективного использования информационных технологий часто
необходимы консультации, обучение и даже прямое участие представителей
разработчика в решении наиболее сложных и масштабных задач клиентов.
Поэтому у производителей ПС во всем мире существуют сервисные службы,
помогающие многим миллионам клиентов практически во всех странах. Любой
легальный пользователь ПС может воспользоваться высокопрофессиональной
помощью служб технической поддержки пользователей программных
продуктов.

Сервисные службы предоставляют следующие услуги:

бесплатную техническую поддержку, предназначенную для ответов на
требующие немедленной реакции вопросы по конкретным ПС;

периодически публикуется свежая информация, необходимая для оценки,
развёртывания, эксплуатации и поддержки продуктов производителя на
рабочих местах и в рамках организации в целом;

поддержка, планирование, развёртывание и обслуживание систем клиент-сервер, построенных на продуктах производителей.

установка, гарантийное и послегарантийное сервисное обслуживание аппаратных и программных комплексов и сетей;

обеспечение удаленного доступа пользователей к техническим информационным базам данных;

обновление программного обеспечения и документации.

Кроме того, на месте установки оборудования предлагается ряд дополнительных услуг:

поддержание оптимального функционирования и настройка системы;

планирование, разработка и интеграция вычислительных сетей;

предоставление различных информационных услуг и технических консультаций;

обучение пользователей и системных специалистов заказчика.

Кроме коммерческого рынка ПС в мире существует, значительный по объему фонд бесплатно (свободно) распространяемого программного обеспечения.

Очевидно, что это очень важный фактор для развития ИТ во всем мире и особенно важный для России. Слишком часто у нас не хватает денег, чтобы приобрести действительно нужное программное обеспечение.

Например, Фонд свободного программного обеспечения (FSF - Free Software Foundation) представляет собой во многих отношениях исключительное явление в современном мире программирования. FSF - это программистская организация, задачей которой является устранение ограничений по копированию, распространению, изучению и модификации программ для компьютеров. Для достижения этой общей задачи FSF стимулирует разработку и использование свободного программного обеспечения. В "Манифесте GNU", написанном в 1985 г., в качестве основной идеи, приведшей к возникновению FSF, выдвигается неприятие права собственности на программы.

Особенности взаимоотношений в сообществе программистов часто ставят людей перед выбором следования естественному чувству дружбы и взаимопомощи или подчинения препятствующего этому закону о собственности. При использовании свободного программного обеспечения необходимость такого выбора исчезает.

Создание интегрированной свободной программной системы позволяет избежать дублирующей работы программистов (которая часто требуется только по причине наличия программ в чьей-либо собственности). Свободное распространение исходных текстов программ облегчает их сопровождение и приспособление к нуждам конкретного пользователя (не требуется прибегать к услугам только компаний - владельцев лицензий на исходные тексты). Появляется дополнительная и очень важная возможность использования хорошего программного обеспечения в учебных целях.

Ограничение на копирование программ - это не единственный способ зарабатывать деньги. Основная идея состоит в том, что нужно продавать не программы, а труд программиста. В частности, источником дохода может быть сопровождение программных систем или их настройка для использования на новых компьютерах или в новых условиях, преподавание и т.д.

Программное обеспечение FSF является "свободным" в двух смыслах.

Во-первых, любую программу можно свободно копировать и передавать кому угодно.

Во-вторых, наличие исходных текстов программ обеспечивает возможность свободного изучения программ, их улучшения и распространения доработанных вариантов.

Судьба конкуренции программного обеспечения различных производителей, в значительной степени зависит от позиции и поддержки сообщества независимых разработчиков и пользователей. Свобода выбора разработчиком своего инструмента довольно сильно ограничена следующими условиями:

- его личным опытом;
- необходимостью преемственности с предыдущими наработками;
- пожеланиями и возможностями заказчика;
- прочими обстоятельствами.

Один из подходов по выбору систем программного обеспечения различного назначения содержит следующие рекомендации.

1. Одним из главных элементов является выбор программных средств, которые развиваются эволюционным путем. Что означает - в основе всех инноваций лежат уже существующие решения, область действия которых либо расширена (например, переход от локальных моделей к сетевым), либо дополнена новыми функциями. При этом соответствующее развитие должны иметь практически все основные технологии:

- операционные системы;
- локальные средства разработки;
- сетевые решения;
- офисные пакеты для конечных пользователей.

2. Должна соблюдаться комплексность и высокая степень интеграции решений. Необходимо, чтобы имелся полный набор средств для реализации практически любых задач, ориентированных как на пользователей, так и на разработчиков разного уровня. Конечно каждый конкретный компонент может быть лучше или хуже своих аналогов, но комплексность и интеграция между компонентами должна быть четкой.

3. Важным элементом должны быть принципы открытости и поддержки альтернативных технологий, в том числе и конкурирующих.

4. Важным фактором являются решения по локализации общесистемных ПС и адаптации их для национальных условий.

5. В области ПС для Internet необходимо уделять внимание возможностям как клиентской, так и серверной части ПС, в том числе средствам создания Web-серверов. Хотя очевидно, что по мере расширения Internet основные возможности конечного пользователя будут определяться не типом его браузера (функциональные возможности которых уже сейчас сильно сблизились), а именно сервером.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 1. Составьте логическую схему базы знаний по теме юниты.

Задание 2. Изобразите на схеме основные компоненты программного обеспечения разработчика АИС

Задание 3. Впишите в представленную таблицу содержание работ на стадии разработки технического задания на АИС по этапам

Стадия разработки	Этапы работ	Содержание работ
Техническое задание (ТЗ)	1. Обоснование необходимости разработки АИС 2. Исследования в предметной области АИС 3. Разработка и утверждение ТЗ	

Задание 4. Впишите в представленную таблицу содержание работ на стадии разработки эскизного проекта АИС по этапам

Стадия разработки	Этапы работ	Содержание работ
Эскизный проект (ЭП)	1. Разработка ЭП АИС 2. Утверждение ЭП	

Задание 5. Впишите в представленную таблицу содержание работ на стадии разработки технического проекта АИС по этапам

Стадия разработки	Этапы работ	Содержание работ
Технический проект (ТП)	1. Разработка ТП АИС	
	2. Утверждение ТП	

Задание 6. Впишите в представленную таблицу содержание работ на стадии разработки рабочего проекта АИС по этапам

Стадия разработки	Этапы работ	Содержание работ
Рабочий проект (РП)	1. Разработка компонент АИС	
	2. Разработка рабочей документации	
	3. Испытания компонент АИС и сборка	

Задание 7. Впишите в представленную таблицу содержание работ на стадии сопровождения АИС у заказчика по этапам

Стадия разработки	Этапы работ	Содержание работ
Сопровождение АИС у заказчика	1. Накопление данных в ходе эксплуатации АИС 2. Модификация АИС и переход на новую версию	

ТРЕНИНГ УМЕНИЙ

Пример выполнения упражнения на умение №1

1. Поиск в Internet информации о технической поддержке по ПС для создания Web-серверов методом ввода ключевых слов через кнопку “Поиск”.

Задание

Найти в Internet исходные коды классов JDK на языке Java по адресу: <http://java.sun.com> и произвести дальнейший поиск вводом ключевых слов через кнопку Поиск.

(Может быть использован проводник Microsoft Internet Explorer или Netscape)

Сохранить найденную информацию в файле `jav_klas.htm`.

Решение

№ п/п	Алгоритм	Конкретное соответствие данной ситуации предложенному алгоритму
1.	Запустить проводник (браузер)	Двойной щелчок “мыши” по значку проводника
2.	Открыть меню File (Файл)	Двойной щелчок “мыши” по кнопке «Файл»
3.	Открыть подменю Open (Открыть)	Двойной щелчок “мыши” по кнопке «Открыть»
4.	Набрать в окне “Адрес” URL нужного объекта и “ввод”	Набрать в окне “Адрес” http://java.sun.com и нажать Enter
5.	На открывшейся Web-странице произвести дальнейший поиск вводом ключевых слов	Двойной щелчок “мыши” по кнопке Поиск, ввести: Java, JDK, class и нажать Enter
6.	Получив в окне проводника найденную информацию сохранить ее в виде файла на своем ПК, если результат положительный	В окне проводника появилась информация, если результат положительный в меню File (Файл) двойным щелчком “мыши” по имени Save as... (Сохранить как...). В открывшемся диалоговом окне набрать <code>jav_klas.htm</code> и нажать Enter.

Решите самостоятельно:

Задание 1.1.

Найти в Internet данные о последних новостях в использовании языка Java по адресу: <http://java.sun.com> и произвести дальнейший поиск вводом ключевых слов через кнопку Поиск.

(Может быть использован проводник Microsoft Internet Explorer или Netscape)

Сохранить найденную информацию в файле `jav_new.htm`.

Задание 1.2.

Найти в Internet информацию об инструментальных средствах (компиляторы, средства просмотра и т.д.) для языка Java по адресу: <http://java.sun.com> и произвести дальнейший поиск вводом ключевых слов через кнопку Поиск.

(Может быть использован проводник Microsoft Internet Explorer или Netscape)

Сохранить найденную информацию в файле `jav_kurs.htm`.

Пример выполнения упражнения тренинга на умение №2

2. Поиск в Internet информации о технической поддержке фирм-производителей ИТ через адресную строку вводом ключевых слов

Задание

Найти в Internet главную Web-страницу фирмы “Sun microsystems” по адресу:

<http://www.sun.ru> ;

(Может быть использован проводник Microsoft Internet Explorer или Netscape).

Провести дальнейший поиск информации о технической поддержке серверов фирмы “Sun microsystems” через адресную строку вводом ключевых слов.

Изучить содержание и вывести на печать.

Решение

№ п/п	Алгоритм	Конкретное соответствие данной ситуации предложенному алгоритму
1.	Запустить проводник (браузер)	Двойной щелчок “мыши” по значку проводника
2.	Открыть меню File (Файл)	Двойной щелчок “мыши” по имени File (Файл)
3.	Открыть подменю Open (Открыть)	Двойной щелчок “мыши” по имени Open (Открыть)
4.	Набрать в окне “Адрес” URL нужного объекта и “ввод”	Набрать в окне “Адрес” http://www.sun.ru и нажать Enter
5.	Получив в окне проводника главную Web-страницу фирмы - производителя ПС, ознакомиться с ее содержанием	В окне проводника появилась информация - главная Web-страница фирмы “Sun microsystems”. Ознакомьтесь с ее содержанием
6.	Найти информацию о технической поддержке серверов фирмы через адресную строку	В адресную строку введите команду go и далее через пробел – server, service. Internet Explorer начнет поиск с использованием заранее определенной системы поиска.
7.	Распечатать Web-страницу на принтере	Двойной щелчок “мыши” по “горячей” кнопке с изображением принтера.

Решите самостоятельно:

Задание 2.1

Найти в Internet главную Web-страницу фирмы “Microsoft” по адресу:
<http://www.microsoft.ru> ;
 (Может быть использован проводник Microsoft Internet Explorer или Netscape)
 Провести дальнейший поиск информации о технической поддержке программной продукции фирмы через адресную строку вводом ключевых слов
 Изучить содержание и вывести на печать.

Задание 2.2.

Найти в Internet главную Web-страницу фирмы “Compaq” по адресу:

http://www.compaq.ru ;
 (Может быть использован проводник Microsoft Internet Explorer или Netscape)
 Провести дальнейший поиск информации о технической поддержке рабочих станций фирмы через адресную строку вводом ключевых слов
 Изучить содержание и вывести на печать.

Пример выполнения упражнения тренинга на умение №3

3. Поиск в Internet через адресную строку вводом ключевых слов информации об обучении пользователей фирмами-производителями ИТ

Задание

Найти в Internet через адресную строку вводом ключевых слов информацию об обучении пользователей фирмой “Sun microsystems” языку Java
 (Может быть использован проводник Microsoft Internet Explorer или Netscape)
 Сохранить найденную информацию в файле jav_educ.htm.

Решение

№ п/п	Алгоритм	Конкретное соответствие данной ситуации предложенному алгоритму
1.	Запустить проводник (браузер)	Двойной щелчок “мыши” по значку проводника
2.	Открыть меню File (Файл)	Двойной щелчок “мыши” по кнопке «Файл»
3.	Открыть подменю Open (Открыть)	Двойной щелчок “мыши” по кнопке «Открыть»
4.	Найти информацию о технической поддержке серверов фирмы через адресную строку вводом ключевых слов	В адресную строку введите команду go и далее через пробел – educate, service, Java и нажать Enter. Internet Explorer начнет поиск с использованием заранее определенной системы поиска.
5.	Получив в окне проводника найденную информацию сохранить ее в виде файла на своем ПК, если результат положительный	В окне проводника появилась информация, если результат положительный в меню File (Файл) двойным щелчком “мыши” по имени Save as... (Сохранить как...). В открывшемся диалоговом окне набрать jav_educ.htm и нажать Enter.

Решите самостоятельно:

Задание 3.1.

Найти в Internet через адресную строку вводом ключевых слов информацию об обучении пользователей фирмой Microsoft поддержке ОС фирмы
(Может быть использован проводник Microsoft Internet Explorer или Netscape)
Сохранить найденную информацию в файле ms_OS.htm.

Задание 3.2.

Найти в Internet через адресную строку вводом ключевых слов информацию об обучении пользователей фирмой SYBASE использованию инструментального средства разработки PowerBuilder
(Может быть использован проводник Microsoft Internet Explorer или Netscape)
Сохранить найденную информацию в файле PB_kurs.htm.

ОСНОВЫ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

ЮНИТА 3

ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Редактор Л.А. Савина
Оператор компьютерной верстки Е.М. Кузнецова

Изд. лиц. ЛР № 071765 от 07.12.1998
НОУ “Современный Гуманитарный Институт”
Тираж

Сдано в печать
Заказ
