

ВИРТУАЛЬНЫЙ МНОГОФУНКЦИОНАЛЬНЫЙ СПЕКТРОАНАЛИЗАТОР.

Князев В.С.

Научный руководитель: к.т.н., доцент Власов А.И.

Иу4, МГТУ им.Н.Э.Баумана

<http://iu4.bmstu.ru>

В последнее время широкое распространение получили виртуальные измерительные средства, которые заменяют стандартные измерительные средства (вольтметры, самописцы, осциллографы, магнитофоны, спектроанализаторы и др.). такие системы состоят из компьютера и нескольких плат сбора данных, причем программная часть виртуального прибора может эмулировать переднюю управляющую панель стационарного измерительного устройства. Сама панель, сформированная на экране дисплея, становится панелью управления виртуального прибора. Пользователь виртуального прибора активизирует объект графической панели с помощью "мыши", клавиатуры или прикладной программы. Несмотря на то, что стационарные приборы старались оснастить дисковыми и т.п., получить на них те же результаты, что и с использованием ПК, оказалось сложно. Кроме того, появление нового стандарта сменных НМГД на 100 Мбайт и 1 Гбайт (применим и на портативных компьютерах), а также повсеместное внедрение сетевых технологий позволяет говорить о виртуальном эксперименте, а все остальные программно-аппаратные средства подбираются, исходя из технических требований эксперимента.

Структура типового аппаратно-программного измерительного комплекса включает в себя следующие составляющие: пользовательский интерфейс, программную и аппаратную части и блок датчиков.

Пользовательский интерфейс предоставляет пользователю графическую информацию об измерениях акустических полей. Контекстно-зависимое меню позволяет использовать тот или иной алгоритм спектрального оценивания, блокируя доступ к запрещенным в данный момент пунктам.

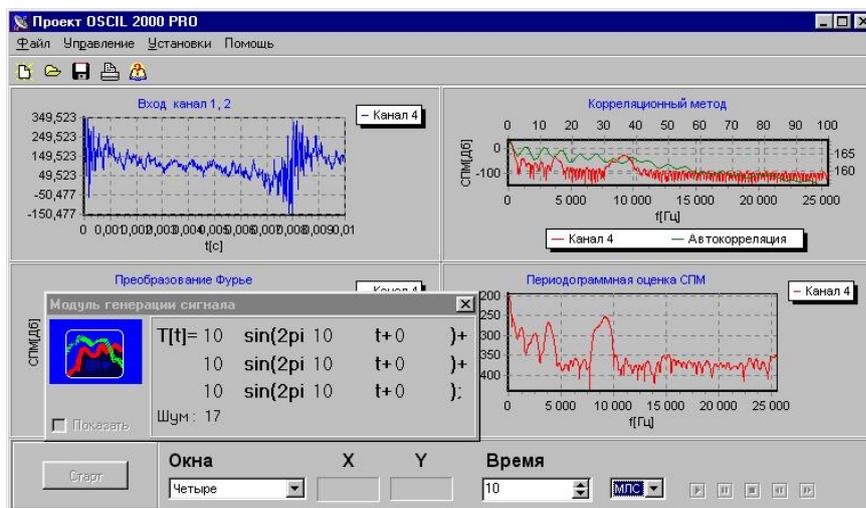


Рис.1. Интерфейс комплекса.

Программная часть комплекса непосредственно обрабатывает полученные сигналы и формирует выходные данные для их последующего вывода на экран и принтер, а также для работы с ними пакетов программ, использующих их в своей работе.

Аппаратная часть комплекса включает в себя плату сбора данных, устанавливаемую в шину данных компьютера PCI/ISA, блок датчиков, установленных вокруг объекта испытаний. В качестве датчиков при виброакустических измерениях используются микрофоны или вибродатчики.

Характеристики виртуальных приборов определяются только техническими характеристиками платы сбора данных. При этом соотношение цена-качество оптимально. Совмещение в ПК возможностей быстрой цифровой обработки, полученной при вводе через плату сбора данных информации, с качественным одновременным изображением результатов

обработки, делает систему виртуальных средств измерения на базе ПК основным инструментом динамических измерений для инженеров разных специальностей.

Один из модулей многофункционального спектроанализатора предназначен для моделирования поличастотных систем активного гашения. Он позволяет проводить исследования поличастотных систем активного гашения, осуществлять рациональный выбор структуры и оптимизацию параметров алгоритма поличастотной фильтрации.

Аппаратно-программный комплекс решает следующие задачи.

- Анализ типовых структур систем активного гашения; исследование и разработка математических моделей и алгоритмов управления поличастотной системой активного гашения; исследование зависимости эффективности компенсации от порядка системы активного гашения, ее структуры, используемых алгоритмов и характеристик объекта управления.
- Детальные экспериментальные исследования эффективности снижения шума при использовании поличастотных систем активного гашения на физической модели; экспериментальное подтверждение эффективности разработанных систем активного гашения. Разрабатываемый комплекс должен удовлетворять техническим требованиям, которые перечислены в техническом задании, о них будет сказано ниже.

Поличастотная система активного гашения, в общем случае, относится к адаптивным, многоканальным, многомерным, узкополосным средствам активной компенсации акустических полей. Область применения поличастотных систем - узкополосные случайные акустические процессы. Для борьбы с шумом и вибрациями в этом случае может быть использована традиционная широкополосная система, однако время настройки такой системы будет в 5-10 раз большее, чем у поличастотной, а при высокой помехоустойчивости – вплоть до 240 раз. При этом требования, предъявляемые поличастотным алгоритмом к аппаратным средствам, существенно ниже, чем аналогичные для широкополосной системы.

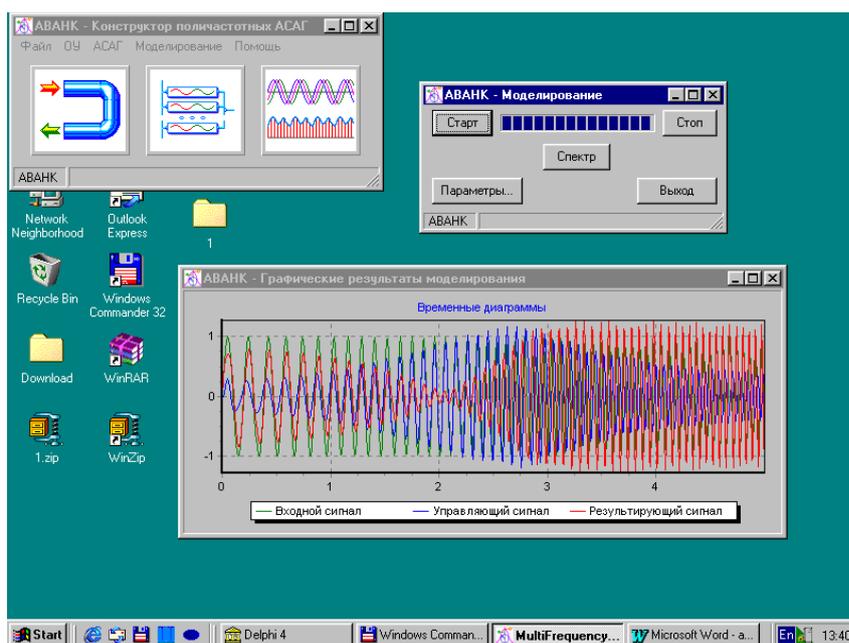


Рис.3.

САГ узкополосных случайных составляющих виброакустических полей используют алгоритмы управления, которые требуют значительно меньшего числа перестраиваемых коэффициентов по сравнению с известными адаптивными алгоритмами в случае их использования для гашения узкополосных составляющих. В общем случае компенсируемая составляющая исходного поля, порождаемая работой цикловых механизмов, не может быть описана периодической или почти периодической моделью. Примером может служить возбуждение исходного виброакустического поля несколькими однопериодными асинхронными двигателями, работающими от одного генератора. При случайном и независимом изменении нагрузок двигателей их оборотные частоты случайным образом изменяются относительно среднего номинального значения, компенсируемая составляющая исходного поля представляет собой узкополосный случайный процесс.

Основные модули аппаратно-программного комплекса представляют собой службы по: формированию данных для ввода, проектированию АСАГ, оптимизации параметров САГ, моделированию процессов активного гашения. Предоставляются также вспомогательные компоненты по выводу графиков исходного сигнала, сигнала компенсации и сигнала ошибки, а также графиков АЧХ и ФЧХ УУ САГ с учетом ОУ, обработчик сообщений, служба подготовки данных для вывода. Основной канал обмена информацией как внутри компонента, так и вовне его реализован через универсальный менеджер БД - BDE, что позволяет легко адаптировать компонент под нужды АПК.

Литература

1. Кузнецов А.П. Моделирование поличастотных АСАГ // Конференция «Информатика и системы управления в XXI dttrt» - М.: МГТУ им.Н.Э.Баумана, 1999 г.
2. Техническое обеспечение цифровой обработки сигналов. Справочни./Куприянов М.С., Матюшин Б.Д., Иванова В.Е. – СПб. «Форт», 2000. – 752 с.
3. Першин Б.Н. Виртуальный измерительный комплекс // Конференция «Информатика и системы управления в XXI dttrt» - М.: МГТУ им.Н.Э.Баумана, 1999 г.

ОПЕРАЦИОННАЯ ПЛАТФОРМА РЕАЛЬНОГО ВРЕМЕНИ – MATRIXREALTIME

Мороз А.А., Михненко А.Е.

Научный руководитель: профессор, д.т.н. Шахнов В.А.

Научный консультант: доцент, к.т.н. Власов А.И.

ИУ4, МГТУ им.Н.Э.Баумана

<http://iu4.bmstu.ru>

Операционные системы реального времени используются главным образом для управления устройствами ввода-вывода. Обычные ОС не подходят для этой цели, когда дело касается мгновенной обработки данных и передачи управляющих сигналов на устройства чтения информации.

Сегодня существует необходимость использования операционных систем реального времени (ОС РВ) при реализации АСУ различных производственных процессов. В основном это высокоточные технологии: АСУ ТП, программы-тренажеры дорогостоящей авто-, авиа- и космической техники, автоматическое управление медицинским оборудованием, системы безопасности, анализирующие аудио- видеoinформацию - это лишь небольшой список, наглядно демонстрирующий характер задач, которые решают ОС РВ.

Рассмотрим подробнее некоторые наиболее перспективные области применения описываемых ОС.

- 1) Научные исследования, требующие обработки данных в реальном режиме времени, требуют универсальной ОС РВ. Реализация автоматизированных систем управления и осуществление контроля за технологическими процессами на базе ОС РВ даст положительный результат. АСУ станут на много более гибкими, производительными и расширяемыми. При правильно выбранных алгоритмах обработки информации и корректной настройке ОС контроль, осуществляемый в реальном времени дает возможность снизить уровень аппаратного обеспечения без потери производительности. Этими возможностями не обладает ни одна ОС построенная без применения технологии реального времени.
- 2) В нынешнее время тренажеры-симуляторы незаменимы при испытаниях автомобилей, самолетов космических кораблей и т.д. Многие компании-гиганты, такие как Dimler Crysler, Toyota, не говоря о правительственных организациях работающих над созданием технических средств нового поколения, уже осознали необходимость их применения. Наиболее широко открыть возможности этих аппаратов поможет рациональное управление. К примеру, современным авиасимуляторам по подготовке летчиков, необходимо реагировать на действия пилота сидящего за штурвалом, мгновенно просчитывая параметры полета, управляемого самолета и, если необходимо, самолетов противника, передавая их на мониторы, а так же на отдельные подвижные части самого тренажера, для создания реальности полета. В настоящее время стало возможным просчитывать поведение самолетов в различных ситуациях с высокой точностью, имея на руках лишь компьютерную модель летательного аппарата, но все расчеты реального поведения самолета требуют больших вычислительных мощностей, правильное использование которых целиком и полностью зависит от ОС, а это значит, что применение ОС РВ наверняка даст положительный результат.
- 3) Анализаторы видео- и аудиоинформации - это тот сектор, в котором системы реального времени наиболее применимы. Во-первых, применение этой аппаратуры с каждым днем возрастает, во-вторых, более широко эта техника распространена в медицине, охранных системах и военной отрасли, т.е. именно там, где требования к управлению оборудованием чрезвычайно высоки. Именно здесь управление устройствами сбора информации имеет наибольшее применение. Например, в охранном секторе контроль над камерами слежения в зависимости от получаемой ими (камерами) информации упростит их установку. Ведь на данный момент только специалисту под силу правильно скорректировать работу всех этих устройств, выбрав для каждого наилучшую позицию для наблюдения. Применение ОС РВ одновременно с удобством увеличивает надежность, т.к. удастся исключить человеческий фактор в процессе контроля; ОС способна анализировать данные постоянно и самостоятельно. Отказавшись от использования ОС РВ, специалисты столкнутся с большим кругом проблем, неправильное решение которых не

гарантирует корректность работы системы. Как следствие, задача, возлагаемая на приборы, не достигается в полной мере.

- 4) Еще одна отрасль, нуждающаяся в применении ОС РВ, - высокоточное производство, где контроль качества должен осуществляться непосредственно во время производства, что существенно экономит время полного производственного цикла, а так же значительно снижает количество бракованных изделий.
- 5) Обеспечение контроля за движением самолетов и космических кораблей, навигация судов и т.д. Эта задача как нельзя лучше описывает применимость ОСРВ. Ведь информация на контролируемые объекты должна поступать незамедлительно иначе последствия могут быть катастрофическими. При разработке навигационных систем технология обработки сигнала в режиме реального времени способна упростить контроль за объектами, практически исключив человеческий фактор из процесса. Соответственно, повышается и точность представления данных.

Нужно признать, что такие системы успешно применяются последние несколько лет, но все дело в том, что практически все из них рассчитаны на выполнение конкретной задачи. Создание проблемно-ориентированной ОС очень повышает стоимость всей разработки и снижает применимость. Для производственных нужд необходимы встраиваемые ОС (уменьшение размера кода), которые помимо всего прочего способны выполнять и другие задачи, т.е. обладать многофункциональностью, так же необходимо помнить о пользователе.

Первое условие выполняется относительно легко (малые размеры свойственны практически всем ОС РВ). Размер кода уменьшается за счет исключения из него некоторых прикладных программ, отсутствие которых не сказывается на работе всей системы и не нарушает ее гибкости. Так же возможно упрощение путем сокращения дополнительных функций в прикладном программном обеспечении. Что же касается многофункциональности, то дело здесь обстоит несколько сложнее, т.к. обычные системы гораздо лучше справляются с этой проблемой. Для обеспечения многозадачности в ОС РВ необходимо большое количество оперативной и сверхоперативной памяти. Связано это с тем, что для обеспечения обработки данных в реальном времени необходимо обеспечить отсутствие задержек. Следовательно, существует необходимость использования самых быстрых видов памяти. Кэширование данных в ПЗУ в этом случае совершенно не приемлемо, т.к. скорости обмена информацией будут на несколько порядков ниже, чем те, которые предоставляет ОЗУ. Возможно, с созданием нового поколения ПЗУ, скорость обмена информацией с которыми будет достаточна для выполнения данной задачи, эта проблема будет решена. Если в управление ОС включаются алгоритмы обработки данных в реальном времени, то задачи, выполняемые вне этих алгоритмов, автоматически имеют пониженный приоритет выполнения, а значит и доступа к памяти. Обойти эти ограничения довольно сложно.

Из вышесказанного видно что, ОС РВ имеют множество недостатков, как, впрочем, и достоинств. Таким образом, в последнее время наметилась тенденция к интеграции обычных ОС и ОС РВ. Существует два основных направления развития технологий реального времени: разработка многофункциональных ОС РВ и написание встраиваемых в обычные ОС модулей для поддержки режима РВ.

Второе направление очень бурно развивается в последнее время. Многие фирмы пошли по этому пути и опередили своих конкурентов. Это связано с тем, что модификация используемых ОС экономит конкретному заказчику деньги на переобучение персонала, а разработчику время на их создание.

В последнее время ОС РВ все чаще интегрируются в системы с открытым кодом (Free BSD, Linux и др.). Преимущество этих систем перед остальными очевидно: отсутствие затрат на приобретение исходного кода, возможность создания собственного программного обеспечения, не обращаясь за описанием необходимых функций к разработчикам. Кроме того стабильность, свойственная UNIX-системам, выделяет их из числа прочих ОС. Стоит отметить, что многие фирмы, занимающиеся созданием ОС РВ, в пакеты с ОС включают специальные языки программирования с поддержкой функций реального времени. Иногда данные функции внедряются в популярные среды разработки ПО, такие как C++. Причем зачастую компиляторы пишутся специально для данной ОС.

Еще одним достоинством такой интеграции является полная совместимость программных продуктов для обеих версий ОС. Преимущество использования дополнительного ядра-модуля

состоит в том, что оно начинает работу лишь тогда, когда запускается программа-клиент, требующая для правильной работы поддержки РВ. Так же возможен вариант альтернативного выбора режима пользователем. В любом случае система способна выполнять обычные программы, используя при этом стандартное ядро, что существенно увеличивает гибкость ОС.

Для полноты картины приведем некоторые наиболее часто встречающиеся недостатки ОС РВ. Во-первых, хотелось бы отметить тот факт, что далеко не для всех устройств, требующих поддержки дополнительного ядра, существуют драйверы, обеспечивающие правильную работу в этом режиме. Но этот недостаток не относится к техническим и не понижает достоинства системы, т.к. создание драйверов является делом времени и осведомленности фирмы производителя. Как правило, такие фирмы регулярно увеличивают базы драйверов, а так же выпускают драйверы по требованию заказчика.

Построение ОС РВ на платформе LINUX

Отличие ОСРВ от обычных систем состоит в том, что сигналы поступающие в систему должны обрабатываться с определенной скоростью, уменьшение которой приведет к сбоям в системе. Сигналы принимаемые в один и тот же промежуток времени должны обрабатываться параллельно.

В ОС Linux очень хорошо реализована многозадачность, что позволяет удовлетворить этому требованию. Как было сказано выше процессы, выполняемые в реальном режиме времени, должны иметь повышенный приоритет, а подпроцессы для этих процессов-родителей должны наследовать его. В Linux этого вполне можно добиться, корректируя ядро. Увеличение уровней исполнения, сделает систему более предсказуемой. Всего этого сложнее добиться на других платформах, таких как Microsoft WindowsNT.

Операционные системы реального времени разделяют на две категории. Это так называемые системы "жесткого" и "мягкого" реального времени. Различие между этими 2-мя классами строится на отношении системы к задержкам во времени исполнения. В "жестких" ОС РВ эти задержки относятся к критическим ошибкам и их наличие свидетельствует о неправильной работе системы. В "мягких" ОС РТ такие ситуации не относятся к критическим, а лишь понижают производительность системы, что существенно не сказывается на ее работе. Естественно, что создание "жестких" систем реального времени требует гораздо больше затрат на разработку т.к. к ним применяются более высокие требования. На основе ОС Linux возможно создание "жесткой" ОС РВ, но для этого необходимо преобразовать ядро таким образом, чтобы система была более предсказуемой.

Для доработки берется стабильное ядро ОС Linux (2.2.16). Следует написать микроядро, в виде модуля, отвечающее за перехватывание прерываний и управление процессами реального времени. В микроядре реализуется налаженная система наследования уровней выполнения, а также передачи ресурсов конкретному процессу. На многопроцессорной платформе дело будет обстоять проще, т.к. системные процессы будут выполняться на одном, а процессы реального времени на другом процессоре. Таким образом достигается параллельность в обработке сигналов. Еще одной проблемой, которую следует решить является управление процессами из вне, т.е. необходимо решить уровень выполнения каких процессов может изменяться пользователем и в каких пределах. Некоторым процессам необходимо самостоятельно определять свою значимость, поэтому нельзя допустить чтобы эти процессы поглотили все системные ресурсы. Изменения, вносимые в исходный код ОС, должны синхронизировать работу основного и микроядра. Если в программу поступает прерывание, сообщающее, что процесс должен обрабатываться в реальном времени, подключается микроядро. Микроядро следит за тем, чтобы время, отводимое на исполнение процесса, не превышало установленного лимита. В противном случае в систему посылается сигнал об ошибке времени исполнения. Для обращения к различным устройствам ввода-вывода необходимо точно знать сколько времени (тактов) затрачивается на тот или иной процесс, для того чтобы установить допустимые временные рамки для работы с этими устройствами. Эти временные показатели должны быть доступны сторонним разработчикам работающим над созданием драйверов устройств.

В качестве основы микроядра выбрана ОСРВ Red Hat ECOS. Объясним этот выбор тем, что достаточно большое количество сигнальных процессоров имеют уже написанные драйверы для этой ОС, что позволит разработчикам и конечным пользователям перейти на MatrixRT без особых

проблем. Для разработки драйверов существует специальный API (Application Programming Interface), в котором отображены все функции и процедуры, необходимые для контроля и обработки потоков с устройств ввода/вывода.

Управление потоком данных в большинстве ОСРВ осуществляется с помощью прерываний. Но так как современные компьютеры не поддерживают достаточного количества IRQ, необходимого для работы с системами, оснащенными большим количеством дополнительных устройств (звуковые карты, карты видеозахвата, SCSI контроллеры, и т. п.), использовать для каждого процесса реального времени отдельного прерывания было бы не оптимальным. Поэтому в ОСРВ MatrixRT реализован немного иной подход. Для этого были созданы так называемые "исключения" (Exceptions). Все исключения привязаны к одному прерыванию, что значительно разгружает ресурсы системы, необходимые для выполнения процессов. Причем, количество исключений ограничено числом 65535, т.е. в принципе количество одновременно выполняемых процессов реального времени ограничено только ресурсами машины (тактовая частота процессора, объем ОЗУ и СОЗУ).

Приведем некоторые примеры. Например, на машине с конфигурацией Intel Pentium 500, 64Mb DRAM, ОСРВ MatrixRT может обрабатывать до 75 системных вызовов без возникновения ошибок времени исполнения, или до 150 системных вызовов с 1% ошибок, что вполне приемлемо для обработки аудио/видео сигнала.

В результате, получаем ОС с открытым кодом, по очень низкой цене (вся цена - стоимость носителя дистрибутива - цена компакт-диска). MatrixRT подойдет как и для автоматических систем контроля, так и для различных мультимедиа-студий.

Литература

- Власов А.И. Аппаратно-программный комплекс автоматизированного проектирования адаптивных и нейроадаптивных систем активного гашения акустических полей //Акустика на пороге 21 века: Сборник трудов 6-ой сессии РАО. - Москва, 1997. - С. 379-382.
- Власов А.И. Автоматизация синтеза систем активного гашения акустических полей //Акустика на пороге 21 века: Сборник трудов 6-ой сессии РАО. - Москва, 1997.-С.375-378.
- Власов А.И. Применение экспертных систем при автоматизации синтеза нейроадаптивных систем активного гашения волновых полей // Нейроинформатика и её приложения: Сборник тезисов докладов 5-го Всероссийского семинара. - Красноярск, 1997. С.34-35.
- Власов А.И. Особенности построения систем автоматизированного синтеза и моделирования средств защиты от влияния волновых полей // Информационные технологии.-1997, №9, - с.31-38.

УНИВЕРСАЛЬНАЯ ЭКСПЕРТНАЯ СИСТЕМА УПРАВЛЕНИЯ ЗНАНИЯМИ И ВЫРАБОТКИ НОВЫХ ТЕХНИЧЕСКИХ РЕШЕНИЙ.

Шмырев В.Н.

Научный руководитель: к.т.н., доцент Власов А.И.

ИУ4, МГТУ им.Н.Э.Баумана

<http://iu4.bmstu.ru>

Модели представления знаний.

Системы искусственного интеллекта, основным представителем которых являются экспертные системы - это системы базирующиеся на знаниях. В базе знаний системы содержится информация как о конкретных фактах и закономерностях соответствующей предметной области (такая информация называется данными или экстенциональной составляющей знаний), так и о более общих законах и правилах (справедливых не только для выделенной предметной области) позволяющих получать (выводить) новую информацию (законы и правила указанного типа называются интенциональной составляющей знаний).

Для создания эффективных ЭС необходимо создать соответствующую базу знаний, содержащую нужную информацию о проблемной области и способах принятия решений в этой области. Существуют различные классы моделей представления знаний.

Декларативные модели базируются на универсальных формальных системах, наиболее распространенными из которых являются исчисление предикатов первого порядка, имеющие эффективную процедуру поиска решения на основе принципа резолюции.

Процедурные модели имеют более эффективные, чем декларативные модели средства поиска решений, но менее универсальны, их сложнее переносить на другие проблемные области, что требует переписки и перенастройки ряда процедур.

Наиболее оптимальным способом представления знаний является сочетание декларативного и процедурного подходов. Такую возможность, а также возможность унифицированного представления как структурных и синтаксических, так и семантических знаний имеют модели представления знаний, к которым относятся фреймовые семантические сети.

Способ представления знаний в данном проекте основывается на концепции фреймов. Поэтому все интенциональные и экстенциональные составляющие реализуются в виде фреймоподобных структур. Знания предметной области, а также встроенные понятия самой системы (такие как слоты, механизмы наследования, эвристики, процессоры задач) имеют явно специфицированные родо-видовые отношения и единый механизм описания отношений, представимый в виде многосвязных списков.

Ядро экспертной системы.

В ядро экспертной системы входят базы данных и знаний, а также решатель, который на основе информации из БД и БЗ осуществляет поиск (вывод) решения. В данном проекте решатель (Процессор Задач) использует механизм вывода AGENDA - управляющий список с динамической коррекцией элементов.

Процессор задач описывается в виде фрейма и реализуется процедурой поиска решения. В общих чертах работу этой процедуры можно описать следующим образом. Из расписания выбирается задача, она объявляется "текущей задачей", затем просматриваются все правила, чтобы выяснить, можно ли (и каким образом) их применить (Затем эти правила ранжируются и исполняются одно за другим). В процессе решения задачи какое-то правило добавляет к расписанию одну или несколько новых задач. В процессе обработки правил программа может использовать метаправила для оптимизации процессов поиска, сортировки списков правил, создания обобщений и т.д. Так как процессор правил представляет собой фрейм-структуру в систему можно добавлять другие процессоры, работающие иначе.

Реализация экспертной системы.

Структурно система состоит из двух слоев. Первый слой представляет собой усовершенствованный LISP-интерпретатор, способный работать с фрейм-структурами. Второй слой - решатель ,плюс все процедуры обработки знаний, описанные в терминах фреймов или LISP-функции реализующие механизмы обработки знаний.

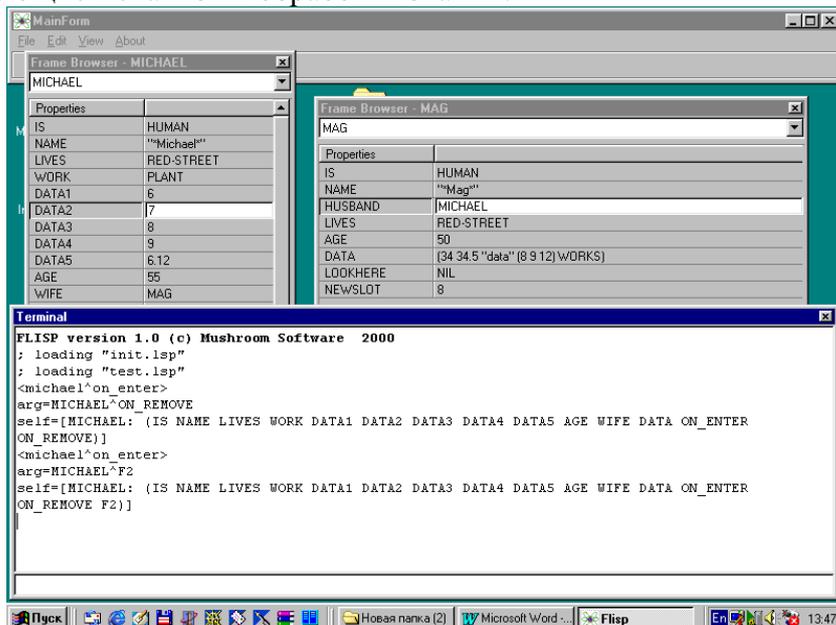


Рис. Вид интерфейса экспертной системы

Программа разработана с использованием Borland C++ Builder и предназначена для работы с Microsoft Windows, вид интерфейса представлен на рисунке.

Литература

- Власов А.И. Применение экспертных систем при автоматизации синтеза нейроадаптивных систем активного гашения волновых полей // Нейроинформатика и её приложения: Сборник тезисов докладов 5-го Всероссийского семинара. - Красноярск, 1997. С.34-35.
- Власов А.И. Особенности построения систем автоматизированного синтеза и моделирования средств защиты от влияния волновых полей // Информационные технологии.-1997, №9, - с.31-38.

АДАПТИВНЫЙ ВЫБОР ПОХОДКИ ШАГАЮЩЕЙ МАШИНОЙ

Кузнецов А.С.

Научный руководитель: профессор, к.т.н. Мысловский Э.В.

ИУ4, МГТУ им.Н.Э.Баумана

Задача выбора типа походки для шагающей машины появилась вместе с самими шагающими машинами.

Для каждого типа поверхности существует конечное множество наиболее подходящих походок. Так, например, для движения по плотной грунтовой дороге с малыми неровностями более подходит галоп, а для движения по пересеченной местности - поочередная походка.

Любая шагающая машина может практически реализовывать лишь ограниченное число походок, связано это, в первую очередь, со сложностью системы управления. Чем большее число походок позволяет реализовывать система управления, тем она более сложна. При движении по опорным точкам вычислительная нагрузка на систему управления более высокая чем при движении периодической походкой. То есть гораздо проще, с технической точки зрения, построить шагающую машину с периодическими походками.

Однако, для технической реализации смены походки необходимо определить зависимости типа походки от условий движения, пользуясь которыми система управления сможет принять решения о смене походки. Такие зависимости будут индивидуальны для каждой машины и потребуют дополнительных исследований.

Однако, решение этой задачи затруднено тем, что у каждой модели шагающей машины свои конструктивные особенности, своя кинематика ноги и свое ограниченное множество реализуемых походок.

Основные сведения о походках

При описании походки шагающей машины введем для каждой ноги функцию состояния:

$$q_i(t) = \begin{cases} 0, & \text{если } i \text{ нога в момент времени } t \text{ находится в фазе опоры} \\ 1, & \text{если } i \text{ нога в момент времени } t \text{ находится в фазе переноса} \end{cases}$$

Состоянием N -ногой шагающей машины в момент t назовем упорядоченную совокупность N чисел.

$$Q(t) = \{q_1(t) \quad q_2(t) \quad \dots \quad q_n(t)\}$$

Походкой N -ногой шагающей машины назовем последовательность состояний $\{Q\}$. Походка называется циклической если последовательность состояний - периодическая. Для циклической походки можно ввести матрицу походки или опорную матрицу, которая имеет вид:

$$G = \begin{bmatrix} Q_1 \\ Q_2 \\ \dots \\ Q_m \end{bmatrix}$$

Рассмотрим некоторые периодические походки четырехногой машины (правая задняя нога - 1, левая задняя - 2, левая передняя - 3, правая передняя - 4):

Поочередная походка	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$
Медленная рысь	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$
Иноходь	$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$
Галоп	$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$
Быстрый шаг слона	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

Расписанием для матрицы походки G назовем упорядоченную совокупность чисел $t_{\nu} = \{t_{\nu}^1, t_{\nu}^2, \dots, t_{\nu}^m\}$, где m - число строк матрицы G , t_{ν}^i - длительность существования i -го состояния на цикле ν . Если $t_{\nu} = t_{\nu+1} = t$ походка называется периодической. Более подробно о описании походок и их типах можно узнать в работе [1].

2. Основные сведения о нейронных сетях

Нейроном называется элемент реализующий функцию взвешенного суммирования

$$s = \sum_{i=0}^M x_i \cdot w_i, \text{ где } x - \text{входы(синапсы), } w - \text{веса, и нелинейной функции активации } y = f(s),$$

где y - выход(аксон).

Обычно рассматривают следующие функции активации:

* сигмоид (т.е. функция S-образного вида)[2]: $f(x) = \frac{1}{1 + e^{-\alpha x}}$

* пороговая функция $f(x) = \begin{cases} 1, \text{ при } x \geq T \\ 0, \text{ при } x < 0 \end{cases}$

Связанные между собой нейроны называют нейронной сетью. Нейронные сети принято подразделять слоистые и полносвязные. Слоистыми называют сети у которых каждый вход последующего слоя связан с выходом предыдущего. Полносвязные сети отличаются от слоистых наличием обратных связей.

3. Нейросетевой подход к задаче выбора походки

Задача выбора походки в зависимости от условий движения сводится к задаче классификации. То есть каждой поверхности необходимо поставить в соответствие определенную походку.

Одним из методов решения данного класса задач является обучения с учителем, в качестве учителя будет выступать человек, назовем его испытателем. Организовать обучение можно следующим образом, на испытательном полигоне воссоздаются различные условия движения и испытатель, в зависимости от условий выбирает наиболее подходящий тип походки. После такого обучения машина, а вернее нейронная сеть сможет решать не только задачу распознавания известных ей условий движения, но и классифицировать и относить к известным типам - неизвестные или вернее не входившие в обучающую выборку условия движения.

Одним из наиболее распространенных алгоритмов обучения с учителем является алгоритм «обратного распространения ошибки».

4. Алгоритм обратного распространения ошибки

Суть алгоритма обратного распространения ошибки состоит в том, что функционирование сети разбивается на два этапа: обучения сети, на этом этапе каждому входному вектору ставится известный выходной и сеть настраивается таким образом что бы ошибка была минимальной, и функционирования, на этом этапе сеть решает задачу классификации.

Введем вектор параметров среды $X = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}$ и вектор управления $U = \begin{bmatrix} u_1 \\ \dots \\ u_m \end{bmatrix}$

Вектор параметров среды представляет собой информацию от датчиков о плотности грунта, коэффициенте сцепления, и других параметрах. Вектор управления может представлять собой код походки, передаваемый в систему управления приводами ног.

Пусть $\tilde{U} = \begin{bmatrix} \tilde{u}_1 \\ \dots \\ \tilde{u}_m \end{bmatrix}$ реальное выходное значение вектора управления, U - требуемое при

подаче на вход вектора X . Определим ошибку методом наименьших квадратов:

$$E = \frac{1}{2} \sum (\tilde{u}_i - u_i)^2 \quad (3.1)$$

Минимизацию будем вести методом градиентного спуска

$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}}$, где $\Delta w_{ij}^{(n)}$ приращение коэффициента синаптической идущей от i

нейрона $n-1$ слоя к j нейрону n слоя, η - коэффициент обучения лежащий в пределах от 0 до 1.

Применяя правило дифференцирования сложной функции представим:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \tilde{u}_j} \cdot \frac{d\tilde{u}_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{ij}}, \text{ где } s - \text{ взвешенная сумма } s_j^{(n)} = \sum_{i=0}^M \tilde{u}_i^{(n-1)} \cdot w_{ij}^{(n)}, \tilde{u}_i^{(n-1)} - \text{ выходной}$$

сигнал i нейрона $(n-1)$ слоя. $\frac{d\tilde{u}_j}{ds_j}$ - производная функции активации. Ее можно сразу выразить.

$$\frac{\partial s_j}{\partial w_{ij}} \text{ равна выходу предыдущего слоя } \tilde{u}_i. \text{ Введем невязку } \delta_j^{(n)} = \frac{\partial E}{\partial \tilde{u}_j} \cdot \frac{d\tilde{u}_j}{ds_j}.$$

Представим первый множитель, применив цепное правило $\frac{\partial E}{\partial \tilde{u}_j} = \sum_k \frac{\partial E}{\partial \tilde{u}_k} \cdot \frac{d\tilde{u}_k}{ds_k} \cdot \frac{\partial s_k}{\partial \tilde{u}_j} = \sum_k \frac{\partial E}{\partial \tilde{u}_k} \cdot \frac{d\tilde{u}_k}{ds_k} \cdot w_{jk}^{(n+1)}$, суммирование по k выполняется среди всех нейронов $(n+1)$ слоя.

Подставив $\delta_k^{(n+1)} = \frac{\partial E}{\partial \tilde{u}_k} \cdot \frac{d\tilde{u}_k}{ds_k}$, получим рекуррентную формулу

$$\delta_j^{(n)} = \left[\sum_k \alpha_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{d\tilde{u}_j}{ds_j}$$

Для выходного слоя ее можно записать в виде

$$\delta_p^{(N)} = (\tilde{u}_p^{(N)} - u_p) \cdot \frac{d\tilde{u}_p}{ds_p}$$

Описанный метод получил название «Обратного распространения ошибки». Базовый алгоритм был изложен в диссертации Пола Вербоса (Paul Werbos) 1974 года, но тогда не привлек к себе должного внимания. Рождение алгоритма back-propagation (обратного распространения ошибки) для широкой публики связано с работой группы PDP (Parallel Distributed Processing), освещенной в двухтомном труде 1986г. Именно там в статье Румельхарта, Хинтона и Уильямса была изложена теория обучения многослойного персептрона. Для более глубокого изучения алгоритма «Обратного распространения ошибки» рекомендую обратиться к [2].

5. Некоторые выводы

Основное достоинство данного метода обучения состоит в том, что столкнувшись с неизвестной поверхностью шагающая машина «ищет» в своей памяти похожую поверхность и применяет тот тип походки, который, по ее мнению, наиболее соответствует данному типу поверхности. Однако, если обучающее множество содержало мало элементов, иными словами говоря, испытатель обучал машину на небольшом количестве поверхностей различных типов, то выбранная походка, на самом деле, может мало подходить для априорно неизвестного типа поверхности. В связи с этим встает вопрос о емкости нейронной сети: для сетей с числом слоев больше двух, он остается открытым. Для нейронной сети с двумя слоями, то есть выходным и одним скрытым слоем, детерминистская емкость сети C_d оценивается так:

$$N_w/N_y < C_d < N_w/N_y \cdot \log(N_w/N_y)$$

где N_w – число подстраиваемых весов, N_y – число нейронов в выходном слое.

Следует отметить, что данное выражение получено с учетом некоторых ограничений:

число входов N_x и нейронов в скрытом слое N_h должно удовлетворять неравенству $N_x + N_h > N_y$.

$$N_w/N_y > 1000.$$

Однако вышеприведенная оценка выполнялась для сетей с активационными функциями нейронов в виде порога, а емкость сетей с гладкими активационными функциями обычно больше. Кроме того полученная оценка емкости подходит абсолютно для всех возможных входных образов, которые могут быть представлены N_x входами. В действительности распределение входных образов, как правило, обладает некоторой регулярностью, что позволяет нейронной сети проводить обобщение и, таким образом, увеличивать реальную емкость. В заключении хочу отметить, нейронные сети не являются панацеей от всех бед хотя решаемый ими класс задач

чрезвычайно широк. Думаю что в ближайшее время вряд ли будет построен аналог человеческого мозга на нейронных сетях, тем не менее нейронные сети не являются просто «игрой воображения» ученых уже в настоящее время они позволяют эффективнее иных средств решать многие задачи. Добавлю так же, что при нейросетевом подходе алгоритм решения задачи можно без изменений реализовать на многопроцессорных системах.

Список использованных источников

1. Ф.Уоссермен «Нейрокомпьютерная техника», М., Мир, 1992
1. Шахнов В.А., Власов А.И., Кузнецов А.С., Поляков Ю.В. «Нейрокомпьютеры - архитектура и реализация» //Chip News, - №№5-8
2. А.Н.Горбань, Д.А.Россиев Нейронные сети на персональном компьютере. - Новосибирск: Наука. Сибирская издательская фирма РАН, 1996. - 276 с.
3. В.В.Лапшин, В.В.Дубинин «Шагающие машины. Походки и статическая устойчивость» Препринт ИПМ РАН, 1992, №79
4. В.В.Лапшин, В.В.Дубинин «Динамика и управление движением статически устойчивых шагающих машин» Препринт ИПМ РАН, 1992, №90
5. В.В.Лапшин, В.В.Дубинин «Энергозатраты статически устойчивых шагающих машин» Препринт ИПМ РАН, 1992, №97

ШКОЛЬНЫЙ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНЫЙ ПОРТАЛ: ТЕХНОЛОГИИ РЕАЛИЗАЦИИ

Сегеда С. В. (учащийся 10 – го класса школы)

Научный руководитель: Аврутин В. А.

Школа №906, г.Москвы.

SCHOOL INFORMATIONAL - EDUCATIONAL JACK HOUSE: TECHNOLOGIES of IMPLEMENTATION

Segeda S.V.

The scientific chief: Avrutin V.A.

School №906, Moscow.

Метод доступа.

Метод доступа Ethernet, разработанный фирмой Xerox в 1975 году, пользуется наибольшей популярностью. Он обеспечивает высокую скорость передачи данных и надежность. Изначально

стандарт IEEE 802.3 был рассчитан на обмен данными со скоростью 10Мбит/сек. По мере того как сетевой обмен становился все более и более интенсивным, парки соединенных с помощью ЛВС компьютеров росли, а также появились новые виды информационных потоков (например потоковые аудио и видео), появился новый стандарт Fast Ethernet. Он предусматривает обмен уже на скоростях 100Мбит/сек при том стоимость адаптированной для Fast Ethernet продукции увеличилось незначительно. Более того, в рамках одной сети могут существовать сегменты Ethernet и Fast Ethernet. Не так давно семейство Fast пополнилось еще одним стандартом Gigabit Ethernet, котрой описывает работу в среде Ethernet на скорости 1Гбит/сек.

Для данного метода доступа используются топология "общая шина" и «звезда». Топология общая шина может быть полезна в терминальных классах, но при построении более крупных сетей используют топологию типа «звезда». Поэтому сообщение, отправляемое одной рабочей станцией, принимается одновременно всеми остальными. Но сообщение, предназначенное только для одной станции (оно включает в себя адрес станции назначения и адрес станции отправителя). Та станция, которой предназначено сообщение, принимает его, остальные игнорируют.

Метод доступа Ethernet является методом множественного доступа с прослушиванием несущей и разрешением коллизий (конфликтов) (CSMA/CD - Carrier Sense Multiple Access with Collision Detection).

Перед началом передачи рабочая станция определяет, свободен канал или занят. Если канал свободен, станция начинает передачу.

Ethernet не исключает возможности одновременной передачи сообщений двумя или несколькими станциями. Аппаратура автоматически распознает такие конфликты, называемые коллизиями. После обнаружения конфликта станции задерживают передачу на некоторое время. Это время небольшое и для каждой станции свое. После задержки передача возобновляется.

Реально конфликты приводят к уменьшению быстродействия сети только в том случае, если работает порядка 80-100 станций.

Аппаратура Ethernet

Аппаратура Ethernet обычно состоит из кабеля, разъемов, Т-коннекторов, терминаторов и сетевых адаптеров при исполнении топологии «общая шина», при исполнении ЛВС по топологии «звезда» аппаратура состоит из кабеля, разъемов RJ-45 и концентраторов и др. коммуникационного оборудования (коммутаторы, маршрутизаторы и др). Кабель, очевидно, используется для передачи данных между рабочими станциями. Для подключения кабеля используются разъемы. Эти разъемы подключаются к сетевым адаптерам - специальным платам, вставленным в слоты расширения материнской платы рабочей станции.

Для Ethernet могут быть использованы кабели разных типов: тонкий коаксиальный кабель, толстый коаксиальный кабель и витая пара. Для каждого типа кабеля используются свои разъемы и свой способ подключения к сетевому адаптеру.

В зависимости от кабеля меняются такие характеристики сети, как максимальная длина кабеля и максимальное количество рабочих станций.

Сетевой адаптер Ethernet.

Вне зависимости от используемого кабеля для каждой рабочей станции необходимо иметь сетевой адаптер. Сетевой адаптер - это плата, которая вставляется в материнскую плату компьютера. Она имеет два разъема для подключения к сетевому кабелю.

Сетевые адаптеры Ethernet используют порты ввода/вывода и один канал прерывания. Некоторые адаптеры могут работать с каналами прямого доступа к памяти (DMA).

На плате адаптера может располагаться микросхема постоянного запоминающего устройства (ПЗУ) для создания так называемых бездисковых рабочих станций. Это компьютеры, в которых нет ни винчестера, ни флоппи-дисков. Загрузка операционной системы выполняется из сети, и выполняют ее программа, записанная в микросхеме дистанционной загрузки.

Перед тем как вставить сетевой адаптер в материнскую плату компьютера, необходимо с помощью переключателей (расположенных на плате адаптера) задать правильные значения для портов ввода/вывода, канала прерывания, базовый адрес ПЗУ дистанционной загрузки

бездисковой станции. Рассмотрим подробно состав аппаратных средств Ethernet для различных типов кабеля.

Толстый коаксиальный кабель.

Толстый коаксиальный кабель, используемый Ethernet, имеет диаметр 0.4 дюйма и волновое сопротивление 50 Ом. Иногда этот кабель называют "желтым кабелем". Это самый дорогостоящий из рассматриваемых нами кабелей. Институт IEEE определил спецификацию на этот кабель - 10BASE5.

Каждая рабочая станция через сетевой адаптер (установлен на материнской плате компьютера и на рисунке не показан) специальным многожильным трансиверным кабелем подключается к устройству, называемому трансивером. Трансивер служит для подключения рабочей станции к толстому коаксиальному кабелю.

На корпусе трансивера имеется 3 разъема: два - для подключения толстого коаксиального кабеля, и один - для подключения трансиверного кабеля.

К сожалению, длина одного сегмента ограничена, и для толстого кабеля не может превышать 500 метров. Если общая длина сети больше 500 метров, ее необходимо разбить на сегменты, соединенные друг с другом через специальное устройство - репитер.

Между собой трансиверы соединяются отрезками толстого коаксиального кабеля с припаянными к их концам коаксиальными разъемами.

На концах сегмента подключены специальные заглушки - терминаторы. Это просто коаксиальные разъемы, в корпусе которых установлен резистор с сопротивлением 50 Ом.

Корпус одного из терминаторов должен быть заземлен. В каждом сегменте сети можно соединять только один терминатор.

Существуют и другие ограничения кроме максимальной длины коаксиального кабеля.

Кроме ограничения на длину сегмента существуют ограничения на максимальное количество сегментов в сети (и, как следствие, на максимальную длину сети), на максимальное количество рабочих станций, подключенных к сети и на максимальную длину трансиверного кабеля.

Однако в большинстве случаев эти ограничения не существенны. Более того, возможности толстого кабеля избыточны.

Тонкий коаксиальный кабель

Тонкий коаксиальный кабель, используемый для Ethernet, имеет диаметр 0.2 дюйма и волновое сопротивление 50 Ом. Импортный кабель называется RG-58A/U и соответствует спецификации 10BASE2. Можно также использовать кабель РК-50, выпускаемый нашей промышленностью.

Сеть Ethernet на тонком кабеле существенно проще, чем на толстом.

Как правило, все сетевые адаптеры имеют два разъема. Один из них предназначен для подключения многожильного трансиверного кабеля, второй - для подключения небольшого тройника, называемого T-коннектором. T-коннектор с одной стороны подключается к сетевому адаптеру, а с двух других сторон к нему подключаются отрезки тонкого коаксиального кабеля с соответствующими разъемами на концах. При этом получается, что коаксиальный кабель подключается как бы непосредственно к сетевому адаптеру, поэтому не нужны трансивер и трансиверный кабель.

На концах сегмента должны находиться терминаторы, которые подключаются к свободным концам T-коннекторов. Один (и только один!) терминатор в сегменте должен быть заземлен.

Сети на тонком кабеле имеют худшие параметры по сравнению с сетями на базе толстого кабеля. Но стоимость сетевого оборудования, необходимого для создания сети на тонком кабеле, существенно меньше.

Следует отметить, что некоторые фирмы выпускают адаптеры Ethernet, способные работать при длине сегмента до 300 метров (например, адаптеры фирмы 3COM). Однако такие адаптеры стоят дороже и вся сеть в этом случае должна быть сделана с использованием адаптеров только одного типа.

Витая пара.

Адаптеры Ethernet способны работать с кабелем, называемым витой парой. Витая пара представляет собой 4 парных жилы заключенных в чехол (спецификация 10BASE-T категория 2 и 100BASE-TX категория 5).

Сетевые адаптеры, способные работать с витой парой, имеют разъем, RJ-45.

Концентраторы.

Для сети Ethernet на базе витой пары необходимо специальное устройство - концентратор. Это наиболее простой тип коммуникационного оборудования. Принцип их работы заключается в разветвлении портов; входящий в концентратор по одному каналу сигнал без каких либо изменений транслируется по всем портам, к которым подключены либо компьютеры, либо др. коммуникационное оборудование. Таким образом, концентратор устраняет лишь те проблемы, которые связаны с прокладкой кабеля и подключении к сети новых компьютеров. Трафик же не регулируется, пропускная способность остается на прежнем уровне. Для сети с количеством участников около десяти, это вряд ли является проблемой, но при большем количестве участников широковещание не применимо. Естественный путь повышения быстродействия в таких условиях – поделить единое сетевое пространство на несколько изолированных друг от друга сегментов. В этом случае компьютеру придется фильтровать только те пакеты которые создаются его соседями по сегменту или адресуются из внешних источников. Для соединения сегментов используются несколько способов. Самым простым является использование мостов. Мост представляет собой своего рода «концентратор над концентраторами». Но существуют и более совершенные устройства.

Коммутаторы.

Коммутация является одной из ведущих современных технологий в области компьютерных сетей. Суть ее заключается в следующем. За каждым из портов устройства закреплен определенный сегмент сети со своим уникальным набором MAC-адресов. Эти адреса хранятся в специальной таблице, по которой коммутатор определяет порт назначения полученного пакета, предотвращая тем самым его широковещательное дублирование всем участкам сети. Благодаря молниеносной обработке пакета и столь же быстрой его передаче на нужный порт, коммутатор обеспечивает прирост быстродействия в несколько раз. А так как передача пакета не занимает внутреннюю шину устройства, а передается непосредственно на порт адресата, то одновременно коммутатор может обслужить несколько пакетов сразу. Для заполнения таблицы адресов коммутатор проверяет конфигурацию сети путем рассылки пробных токенов. Еще более высокий уровень управления сетевыми потоками обеспечивают маршрутизаторы.

Маршрутизаторы

Маршрутизаторы способны выполнять множество функций, не ограничиваясь банальной пересылкой пакетов. Дело в том, что они работают не с MAC адресами пакетов, а с так называемые сетевые адреса – составные индексы несущие в себе данные о подсети адресата и его месте в этой подсети. То есть маршрутизатор представляет собой всю сеть в виде набора подсетей, которые в свою очередь могут быть представлены как одним единственным сетевым сегментом, так и целым набором сегментов. Таким образом, подсеть – основная единица управления трафиком, которым оперируют маршрутизаторы. В зависимости от требований предъявляемых к ЛВС, эти подсети можно организовать в сложноподчиненные структуры в рамках локальной сети. Если внутри сети существуют несколько маршрутизаторов, они эффективно обмениваются друг с другом данными об общей топологии сети.

Основная задача маршрутизаторов – определение наиболее короткого пути между узлом, пославшим пакет, и пунктом его назначения. Так как маршрутизаторы применяются в сложных сетях, возможных путей следования пакета бывает несколько, причем оптимальный из них будет тот, который наиболее полно соответствует поставленным системным администратором условиям. Так же применение маршрутизаторов позволяет решить несколько других проблем, с которыми сталкивается любая локальная сеть, и в первую очередь это касается вопросов безопасности. Так как маршрутизатор оперирует сетевыми адресами, он гарантирует скрытость адресов более высокого уровня в рамках своей подсети. В принципе частным случаем маршрутизатора является

firewall. Наконец маршрутизаторы выполняют крайне важную работу по трансляции форматов пакетов. Это оказывается необходимо в тех случаях, когда сеть представляет конгломерат различных топологий: помимо Ethernet в нее могут входить участки, построенные с применением технологий Token Ring или X.25.

ЛВС Информационного портала школы 906

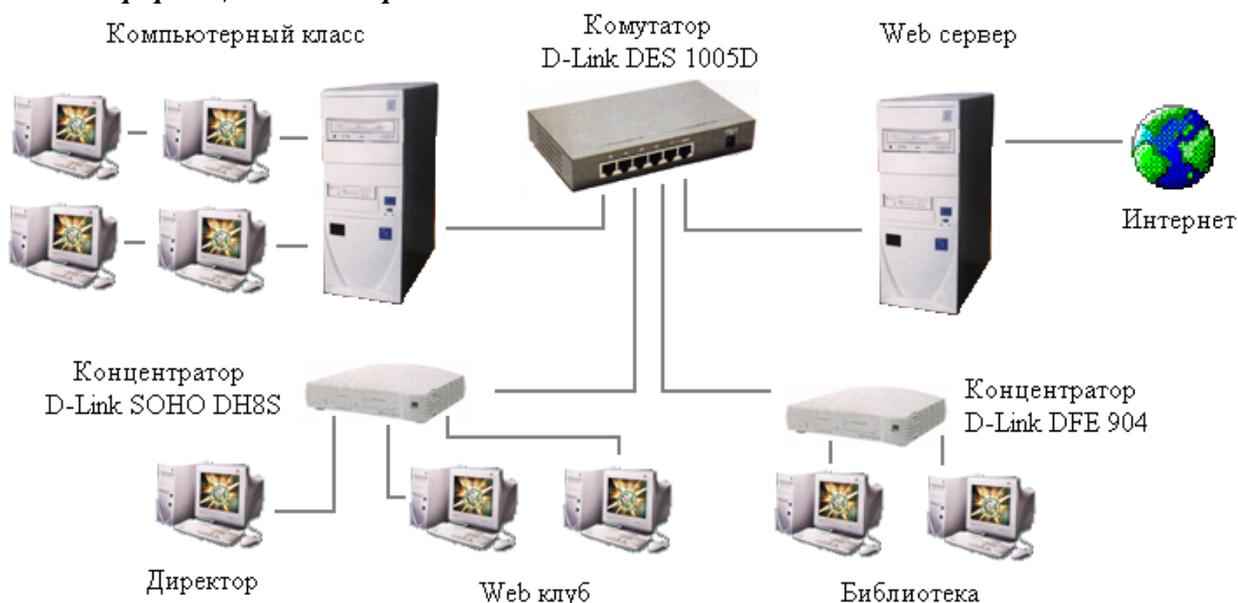


Рис.1. Структурная схема ЛВС школы.

Основой ЛВС школы является коммутатор D-Link DES 1005D и сервер подключенный к интернету, выполняющий функции Linux маршрутизатора и WEB сервера. Оптимальным являлось бы оптоволоконное подключение, но это не осуществимо в рамках финансирования школы, хотя переговоры с компанией Комкор проводятся. Поэтому оптимальным вариантом является выделенная линия.

Коммутатор D-Link DES 1005D обслуживает до 5 подключаемых сегментов, работающих на скорости 10 Мбит/сек или 100Мбит/сек. в полу- или полнодуплексном режиме. Емкость матрицы адресов составляет 8К, что позволяет использовать коммутатор в сетях разных масштабов. Устройство поддерживает режим фильтрации пакетов, а буфер памяти достигает 1Мбайт.

Для соединения с сервером подключенным к интернет, используется сетевая карта D-Link DFE-550TX. Она работает на 32-битной шине PCI в режиме Bus Mastering. Поддерживая тактовую частоту шины независимо от частоты сети, карта осуществляет трансфер данных напрямую с памятью, не привлекая для этих целей ЦП. Карта может быть подключена как к 10Мбит сети 10Base-T, так и к 100Мбит 100Base-TX, при этом использует режим полнодуплексной связи. В нашем случае, так как карта подключена непосредственно к коммутатору, максимальная скорость связи при этом может составлять 100Мбит/сек. Отличительной особенностью DFE-550TX является поддержка контроля за потоком позволяющая компьютеру соединиться с коммутатором. Благодаря этому система получает предупреждение при угрозе перегрузки буфера и своевременно приостанавливает передачу.

Так же к коммутатору подключен сервер терминального класса школы, а через концентраторы компьютеры библиотеки, веб мастерской и администрации школы. По моему мнению не большая стоимость коммуникационного оборудования и высокая производительность такой сети делают ее оптимальной для школ и учебных заведений.

Использование ЛВС в рамках Интернет портала школы 906

Сеть школы используют администрация, учителя и ученики. Администрация школы вносит в сеть информацию, которую необходимо знать учителям и ученикам. Это могут быть приказы, объявления, например, о дате начала каникул, изменения в расписании и многое другое. Учителя могут подойти к компьютеру в своем классе и посмотреть эти объявления. Тоже могут сделать и ученики, правда ученики имеют доступ не ко всей информации потому что некоторые объявления предназначены только для учителей и они закодированы, а код знает только учитель.

Учителя размещают пройденный классом материал, чтобы администрация школы могла следить за успеваемостью класса. Также для этих целей учитель после проведения итоговой контрольной по той или иной теме выставляет оценки за эту контрольную.

Также в школьной сети есть страничка библиотеки, на которой любой учитель или ученик может посмотреть какие книги сейчас находятся в библиотеки. Когда кто-нибудь берет книгу, библиотекарь отображает это в сети. Также на той странички есть услуга заказа книги, с помощью которой ученик может заказать себе книгу, а потом прийти и получить её. Это предназначено для того, чтобы после того как ученик нашел себе книгу и пришел за ней она все ещё была в наличии. Также в сети есть доска объявлений, на которой любой ученик может поместить любую информацию.

Список использованных источников

1. Б.Нанс. “Компьютерные сети”. Москва. Бином. 1996.
2. Молодежная научно-техническая конференция «Наукоемкие технологии и интеллектуальные системы в XXI веке». Москва , МГТУ им. Н. Э. Баумана. 2000.

Kostrа V.V, Prokopchuk Y.A.
Scientific chief d.t.s., prof. Alpatov A.P.

Institute of technical mechanics national academy of science of Ukraine, Dnepropetrovsk
 vladko@rdc.dp.ua

Аннотация

В докладе рассматриваются принципы построения медицинских информационных систем с использованием интеллектуальных модулей. Основу системы составляют следующие модули: конструктор документов, лексический процессор, семантический процессор и проблемный решатель.

Abstract

The report considers question about building medical information systems with use intellectual modules. The modules: designer of documents, lexical processor, semantic processor and problem deciding program assumes as a basis this system.

Анализ информационных и программных компонентов медицинских информационных систем (МИС) [1-4] позволил выделить ряд типовых интеллектуальных модулей, от характеристик которых во многом зависит эффективность работы МИС в целом. К числу таких модулей можно отнести: *Конструктор шаблонов, Лексический процессор, Семантический процессор, Проблемный решатель*. Перечисленные модули позволяют конструировать взаимосвязанные автоматизированные рабочие места (АРМ) медицинских специалистов и целые подсистемы: "СТАЦИНАР", "ПОЛИКЛИНИКА", "ДИАГНОСТ" и т.д. Ниже приведено краткое описание перечисленных модулей.

Конструктор шаблонов / документов, позволяет создавать необходимые формы для описания различных объектов и процессов предметной области (шаблоны являются прототипами документов). Конструктор выполняет три основные функции: создание шаблона; создание документа на основе шаблона; визуализация документа, хранящегося в архиве. Шаблоны документов входят в состав метаинформации. Документы, используемые в системе, могут быть нескольких типов, например: исследования, дневники, хирургическая документация, эпикризы и т.д. Для каждого типа документа имеется свой иерархический справочник, в котором описаны ряд характеристик документов и содержатся ссылки на таблицы базы данных, описывающих шаблоны документов. Конструктор шаблонов / документов позволяет создавать новые шаблоны и модифицировать имеющиеся. С каждым полем документа может быть связано лексическое дерево или семантическое правило.

Лексический процессор, использует заготовки словоформ ограниченного профессионального языка для создания текстовых описаний характеристик объектов и процессов и формирования запросов к базе данных. Лексический процессор позволяет также выявлять факторы риска развития угрожающих состояний. Описание стандартной лексики входит в состав метаинформации. Ниже приведен пример стандартной лексики:

ЖАЛОБЫ НА

```
|постоянную; приступообразную| головную боль {
|умеренную; интенсивную|
головную боль
|возникающую; усиливающуюся| при
повышении давления
физической нагрузке
перемене положения |головы; тела|
< ? >
в течение _____ |лет; года; год; мес; дн| }
|постоянное; приступообразное| !головокружение {головокружение}
```

!НАЗНАЧЕНИЯ {

```
@- лизинил |0,025 мг;0,2 мг форте| по |1/4;1/2;1;1,5;2| таб |1;2;3| [
раза в день в непрерывном режиме под контролем уровня пролактина
в крови. Контроль в ___ мес.]
...}
```

Семантический процессор, позволяет формировать заключения на основе семантических правил (операций) отображения реальных значений факторов, фигурирующих в документе, на множество числовых и лингвистических переменных. Семантический процессор позволяет также вводить в документы вычисляемые поля и выделять факторы риска развития угрожающих состояний. Описание семантических операций входит в состав метаинформации. Программная реализация семантических операций разных видов может существенно различаться и постоянно совершенствуется. Один из видов семантических операций реализован с помощью скриптов. Скрипт представляет собой набор текстовых строк на формальном языке, напоминающем Паскаль.

```
var
    FB:real {фракция выброса}
    s_s_m:string { Сократительная способность миокарда }
```

```

endvar
if (FB < 45) then
  s_s_m:='снижена'
else
  s_s_m:='удовлетворительная'
writeln('s_s_m:'s_s_m) {возврат значения}

```

Проблемный решатель, использует информацию, полученную, в частности, от лексического и семантического процессоров (факторы риска), для решения задач диагностики, прогнозирования и выбора такого уровня помощи, при котором минимизируется вероятность реализации угрозы. Причинно – следственные схемы явлений, правила образования производных понятий (синдромов, заключений) входят в состав метаинформации, образуя базу знаний МИС.

Проблемный решатель состоит из двух частей: *логического анализатора* и собственно *решателя*. Логический анализатор можно представить следующим образом:

$$L_H = \{S, P, Y\}_H,$$

где $S = \langle s_1, \dots, s_n \rangle$ - словарь элементарных понятий, факторов, симптомов, которые непосредственно фигурируют в документе; $Y = \langle y_1, \dots, y_m \rangle$ - словарь производных понятий, симптомокомплексов, синдромов, заключений; P - правила образования производных понятий, а также устранения противоречий. Компоненты логического анализатора существенно зависят от решаемой задачи, точнее говоря, от вида проблемной ситуации или угрожающего состояния. Ниже приведены примеры, в которых понятия s_i и y_j одновременно ассоциированы с булевыми переменными, принимающими значения 0/1:

$$P_1: y_1 = s_3 \cdot s_7 + (1 - s_{12}) \cdot s_{37};$$

$$P_2: y_4 = s_{25} \cdot (1 - y_{11}) + s_{21} \cdot y_{27} + (1 - y_2).$$

Проблемный решатель формально может быть представлен следующим образом:

$$R_H = \{S, Y, U, F\}_H,$$

где U - множество допустимых управлений (уровней помощи); F - коллекция эвристических и точных методов, позволяющих диагностировать состояние и прогнозировать развитие процессов в рассматриваемом объекте, исследовать и выявлять те факторы, которые оказывают наибольшее влияние на их развитие. В отличие от семантического процессора проблемный решатель не связан с каким либо одним документом, поэтому он может работать с данными, полученными из разных источников (документов). Схему работы проблемного решателя изобразим следующим образом:

$$f: S \times Y \times U \rightarrow \langle N_i, U_j, p_{ij} \rangle_f,$$

где $f \in F$; p_{ij} - вероятности состояний N_i при управлении U_j . Таким образом, одну и ту же задачу целесообразно решать несколькими способами. Окончательный выбор стратегии и тактики остается за ЛПР с учетом имеющихся критериев K и мотивации M .

Медицинские информационные системы с использованием перечисленных выше модулей внедрены в ряде лечебных учреждений Днепропетровской области.

Литература

1. Алпатов А.П., Прокопчук Ю.А., Костра В.В., Пилипченко И.А. Интеллектуальные управляющие информационные системы. // Сб. трудов межд. конф. по автоматическому управлению «Автоматика - 2000». - Т.6.- Львов: ГосНИИ информационной инфраструктуры, 2000. - С. 20 – 24
2. Прокопчук Ю.А., Алпатов А.П., Огданский Н.Ф., Харченко О.А., Костра В.В. Интеллектуальные модули информационных систем. // Сб. трудов межд. конф. «Искусственный интеллект - 2000» (п. Кацевели). Донецк: Ин-т проблем искусств. интел., 2000. - С 315 - 317.
3. Харченко О.А., Прокопчук Ю.А., Костра В.В., Коноваленко М.Е. Информационная система диагностического отделения медицинского НИИ // Сб. трудов филиала МГТУ им. Н.Э.Баумана в г. Калуге. Специальный выпуск. Материалы межд. Конф. «Приборостроение

- 2000» (п. Симеиз). Калуга, 2000. - С. 349 - 352.
4. Алпатов А.П., Прокопчук Ю.А., Костра В.В. Семантический процессор для медицинских приложений. // Сб. трудов филиала МГТУ им. Н.Э.Баумана в г. Калуге. Специальный выпуск. Материалы межд. Конф. «Приборостроение - 2000» (п. Симеиз). Калуга, 2000. - С. 353 – 357.

**ПРИМЕНЕНИЕ CASE СРЕДСТВ ПРИ ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННОЙ
СИСТЕМЫ УПРАВЛЕНИЯ РАДИОТЕХНИЧЕСКИМ ПРЕДПРИЯТИЕМ
ИЛИ ИНСТРУМЕНТАРИЙ ДЛЯ ИНФОРМАЦИОННОГО АУДИТА ПРЕДПРИЯТИЯ**

Писаревская А.В
научный руководитель: к.т.н. Власов А.И.
ИУ4, МГТУ им.Н.Э.Баумана.

**USING OF CASE FACILITY ON PROJECTION OF AUTOMATED CONTROL SYSTEM RADIO
DEVELOPMENT.**

Pisarevskaja A.V.
the scientific chief: Ph.D. Vlasov A.I.

Аннотация. В докладе рассмотрена возможность применения CASE-средств при проектировании автоматизированной системой управления радиотехническим предприятием. Анализируются варианты построения автоматизированной системы управления радиотехническим предприятием.

Abstract. The paper is about ability of using of case facility on projection of automated control system radio development. Conclusion regarding the necessity of research variants construction of automated control system radio development.

Введение.

С каждым годом расширяются сферы применения информационных систем, они все больше и больше входят в нашу жизнь, помогая нам ориентироваться в потоке информации, быстрее и адекватнее оценивать и реагировать на ситуации, и как результат улучшить качество и повысить производительность труда.

Сегодня многие предприятия проявляют большой интерес к информационным системам, благодаря которым сведения о производстве, сбыте, закупке, бухгалтерии, эксплуатации оборудования, финансовом планировании сразу же после ввода становятся доступными для руководителей предприятия и специалистов.

Индустрия информационных технологий занимает одно из ведущих мест (по данным European Information Technology Observatory (Франкфурт) в 1999г. ожидался прирост европейского рынка информационных систем на 10,7 %) и продолжает увеличивать темп своего развития.

Первые автоматизированные системы-системы материально-технического обеспечения производства появились в еще конце 60-х г. и были ориентированы на решение задач учета и оценки потребностей в сырье и материалах (системы планирования материальных ресурсов MPR(Material Requirements Planning)).

В конце 70-х появилось новое поколение систем. Эти системы обеспечивали более тесную связь и координацию всех подразделений производства, отвечающих за проектирование, изготовление, снабжение, реализацию и маркетинг. Они получили название, системы планирования производственных ресурсов MPR-II (Manufacturing Resource Planning).

Дальнейшее усовершенствование привело к появлению систем планирования ресурсов ERP (Enterprise Resource Planning), что обеспечивало выполнение всех функций учета и контроля для многопрофильных предприятий со сложными административными и территориальными структурами.

Создание корпоративных автоматизированных систем управления сложная задача. Решение, которой требует полной инвентаризации всех видов ресурсов объекта и представления их в едином информационном пространстве. Для упрощения работы системных аналитиков и разработчиков возможно использование инструментальных средств и сред разработки автоматизированных систем, которые вместе образуют, так называемые, CASE(Computer Aided System Engineering)-системы.

Под *Автоматизированной Системой Управления (АСУ)* радиотехническим предприятием будем понимать комплекс аппаратно-программных средств реализующих мультикомпонентную информационную систему, обеспечивающую современное управление процессами принятия решений, проектирования, производства и сбыта в режиме реального времени.

Автоматизированная система управления радиотехническим предприятием, отвечающая современным требованиям производственных отношений может быть построена только на основе целого комплекса современных информационных технологий: аппаратных, программных и телекоммуникационных.

Программный комплекс управления предприятием должен поддерживать как минимум следующие сферы деятельности:

- 1) Управление проектами;
- 2) Управление поставками и сбытом;
- 3) Управление производством;
- 4) Управление финансами;
- 5) Управление тех обслуживанием и тех информационным обеспечением.

Рассмотрим применение CASE-средств на примере построения одного из модулей АСУ радиотехнического предприятия - системы управления производством.

При проектировании АСУ различают системы:

- 1) поведенческого проектирования – построение имитационной модели для поиска оптимального решения, применение аппарата моделирования динамических дискретных систем (сети Петри). Для построения возможно использование методологии IDEF3.
- 2) функциональное проектирование – наиболее известная методика- моделирование бизнес-процессов IDEF0. Язык, используемый для описания модели, предложен более 20 лет назад Дугласом Россом и назывался SATD- Structured Analysis and Design Technique. На данный момент он принят в качестве федерального стандарта США под названием IDEF0.
- 3) информационное проектирование – для построения применяют методологию IDEF1X. Данная методология была разработана для армии США и широко применяется в гос. учреждениях и промышленных предприятиях.

1 Методы проектирования информационных систем.

1.1. Этапы разработки автоматизированных систем управления с применением CASE-средств.

При построении автоматизированной системы управления выделяют следующие этапы:

- 1) Разработка и анализ бизнес модели;
- 2) Формализация бизнес модели, разработка логической модели бизнес-процессов;
- 3) Выбор лингвистического обеспечения, разработка программного обеспечения информационной системы;
- 4) Тестирование и отладка информационной системы;
- 5) Эксплуатация и контроль версий.

Применение CASE-средств возможно на первых двух этапах проектирования.

1) *Разработка и анализ бизнес модели.*

На данном этапе определяют основные задачи АСУ. Производят инвентаризацию всех процессов и ресурсов используемых в процессе производства. Производится декомпозиция задач по модулям. Для проведения анализа и реорганизации бизнес-процессов возможно использовать CASE - средство верхнего уровня VPwin. VPwin поддерживает несколько методологий, определяющих описание системы, а именно методологии IDEF0, IDEF3, DFD, что позволяет полно провести функциональное и поведенческое проектирования. И как результат получить концептуальную модель АСУ, состоящую из описания предметной области, ресурсов и потоков данных, перечня требований и ограничений к технической реализации АСУ.

2) *Формализация бизнес модели.*

На основе формализации полученной концептуальной модели строят логическую модель. Т.е. проводят информационное проектирование с использованием методологии IDEF1X. Для построения возможно применение инструментального средства ER(Entity-Relationship)win. В результате получают разработанное информационное обеспечение АСУ: схемы и структуры данных для всех уровней модульности АСУ, документация по логической структуре АСУ, сгенерированные скрипты для создания объектов БД.

Использование CASE – средств позволяет:

- 1) разбить реализацию проекта на несколько стадий анализа (что обеспечит лучшее восприятие бизнес-логики предметной области, для этого используют Case-средство верхнего уровня VPwin);
- 2) упростить проектирование системы (построение модели данных, для этого применяют, предлагаемое Platinum technology, удобное инструментальным средство ER(Entity-Relationship)win);
- 3) быстро вносить изменения в проект, если в процессе проектирование произошли изменения внешних условий;
- 4) уменьшить время разработки.

Применение CASE-средств позволяет лучше понять работу предприятия и на основе построенной модели произвести информационный аудит, что позволит адекватно оценить

существующие бизнес-процессы и на основе полученной информации произвести изменения в работе предприятия.

1.2. Реализация построения автоматизированной системы управления производством изготовления изделия.

Для построения автоматизированной системы управления технологическим отделом, необходимо проанализировать весь цикл, который проходит изделие. Для этого на основании имеющейся на изделие маршрутной карты составляется концептуальная модель, отражающая протекающие бизнес-процессы (Рис 1).

Для построения модели воспользуемся методологией IDEF3. IDEF3 (workflow diagramming) – методология моделирования использует графическое описание информационных потоков, которое может быть использовано для анализа завершенности процедур обработки информации.

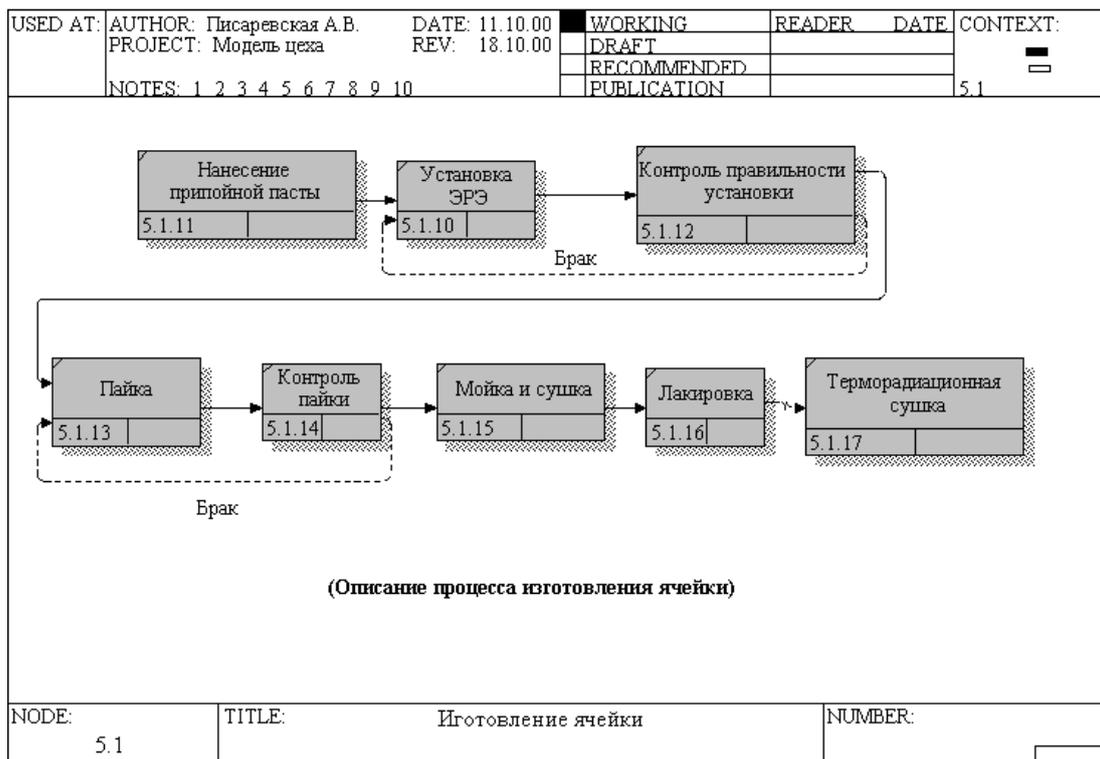


Рис.1. Описание процесса изготовления ячейки на основе методологии IDEF3.

Формализация модели изготовления ячейки с использованием методологии IDEF1X с помощью реляционных методов реализуется с использованием отношений один – ко -многим (Рис 2). Реализацией данной логической модели является совокупностью таблиц, объединенных в единый модуль – АСУ технологического отдела.

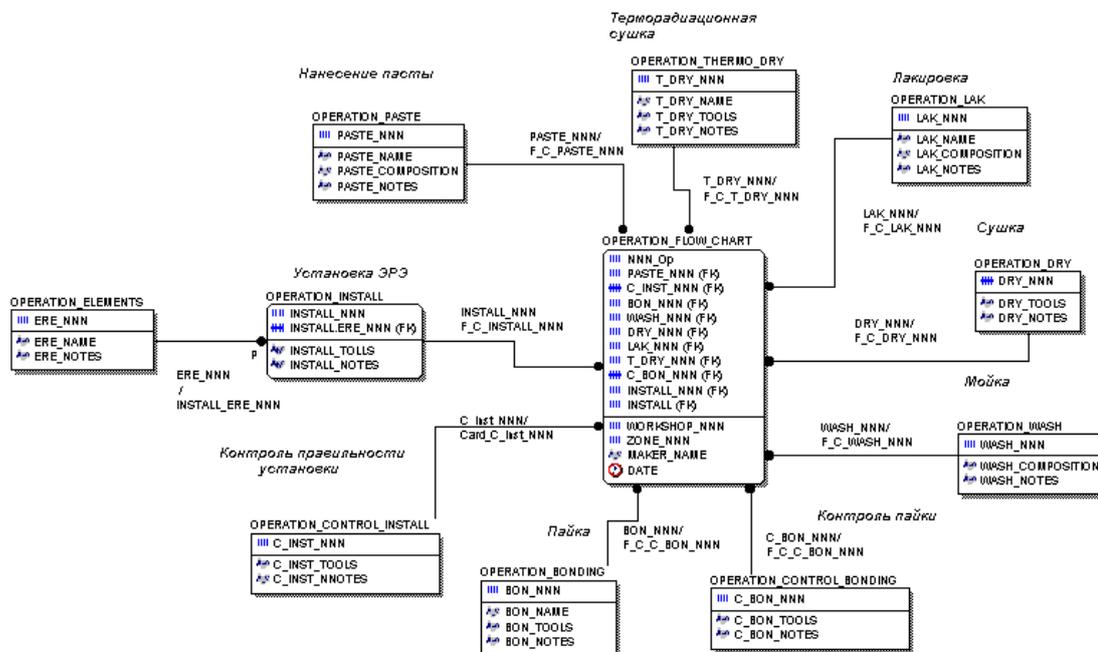


Рис2 Формализация модели процесса изготовления ячейки с использованием методологии IDEF1X.

Ядром логической модели является таблица OPERATION_FLOW_CHART (таблица определяющая структуру маршрутной карты), связывающая таблицы OPERATION_PASTE (Нанесение припойной пасты), OPERATION_INSTALL (Установка ЭРЭ), OPERATION_CONTROL_INSTALL (Контроль правильности установки), OPERATION_BONDING (Пайка), OPERATION_CONTROL_BONDING (Контроль пайки), OPERATION_WASH (Мойка), OPERATION_DRY (Сушка), OPERATION_LAK (Лакировка), OPERATION_THERMO_DRY (Терморационная сушка). Таблица OPERATION_ELEMENT(Элементы) связана с таблицей OPERATION_INSTALL и отражает перечень элементов устанавливаемых на операции «установка элементов».

К недостаткам построенной модели можно отнести то, что вариантов представления бизнес модели может быть много и какой из них наиболее удачный сказать сложно. Это может потребовать дополнительных исследований.

Выводы.

Как уже было сказано применение CASE-средств позволяет произвести информационный аудит имеющихся бизнес процессов. Упростить проектирование системы и предотвратить появление многих ошибок, возникающих из-за неадекватного представления о работе предприятия. Именно возможность упрощения оценки работы предприятия делает применение CASE-средств столь желательными при проектировании автоматизированной системы управления предприятием. И уже после такой оценки перейти к построению модели будущей работы системы. Само же применение АСУ на предприятиях позволяет повысить сократить время на поиски информации, улучшить качество и повысить производительность труда.

Список использованных источников

- 1) С.В. Маклаков «Bpwin Egwin Case-средства разработки информационных систем.» Москва «ДиалогМифи» 2000г
- 2) И.П. Норенков «Основы автоматизированного проектирования» Москва МГТУ имени Н.Э.Баумана 2000г.
- 3) А.И. Власов, С.Л. Лыткин, В.Л. Яковлев «Краткое практическое руководство разработчика по языку PL/SQL». Москва Машиностроение 2000г.

ВЕДУЩИЕ СУБД КАК ОСНОВА ДЛЯ ПОСТРОЕНИЯ ХРАНИЛИЩ ДАННЫХ НА РАДИОТЕХНИЧЕСКОМ ПРЕДПРИЯТИИ

Рассказова С.И.

научный руководитель: к.т.н. Власов А.И.

Московский Государственный Технический Университет им. Н.Э. Баумана,
кафедра “Конструирование и производство ЭА”.

LEADING DBMS AS A BASIC FOR CONSTRUCTION OF DATA WAREHOUSES IN RADIO ENGINEERING MANUFACTURE

Rasskazova S.I.

the scientific chief: Ph.D. Vlasov A.I.

Bauman Moscow State Technical University,
Department of Design and Equipment of Electronic System.

Аннотация. В докладе рассмотрена необходимость построения Хранилищ Данных на радиотехнических предприятиях, а также связанные с этим проблемы. Исследованы решения ведущих компаний-производителей СУБД в области создания Хранилищ Данных и приведена сравнительные характеристики СУБД, лежащих в основе этих решений.

Abstract. This paper is about necessity of Data Warehouses' construction in radio engineering manufactures and connected problems. Leading companies' solutions are investigated. Comparative characteristics of DBMS, which are the basic of these solutions, are adduced.

Введение

Процесс производства на радиотехническом предприятии несет в себе большие объемы информации (например, данные о заказах, партиях выпущенных изделий и т.д.). Необходимо, чтобы результаты деятельности предприятия оказывали влияние на процесс производства с точки зрения его улучшения. Для этого необходимо провести подробный и всесторонний анализ подобной информации за определенный период времени, иногда измеряющийся несколькими годами. Например, имея информацию о браке, можно сделать выводы о причинах его появления и,

изменяя режимы производства, добиться снижения процента брака. На решение проблемы хранения необходимой информации и ее анализа и направлена технология Хранилищ (Data Warehouse) и Витрин Данных.

Однако на предприятии обычно одновременно существуют несколько оперативных информационных систем с собственными базами данных (БД). Так, на предприятии могут быть базы данных, связанных с конкретной операцией или с работой отдельного цеха. Так как эти БД создавались в расчете на решение конкретных задач, то они могут содержать семантически эквивалентную информацию, представленную в разных форматах, с разным атрибутом, иногда даже противоречивую. Информация из таких БД выбирается часто и небольшими порциями. Обычно набор запросов к БД известен уже при проектировании.

Хранилище данных должно содержать единообразно представленную и согласованную информацию, максимально соответствующую содержанию существующих БД. Набор аналитических запросов к базе данных предсказать невозможно. Хранилища данных существуют, чтобы отвечать на нерегламентированные запросы аналитиков. Можно рассчитывать только на то, что запросы будут поступать не слишком часто и затрагивать большие объемы информации. Размеры хранилища стимулируют использование запросов с агрегатами (сумма, минимальное, максимальное, среднее значение и т.д.).

В основе концепции Хранилищ Данных лежат две основополагающие идеи:

1. Интеграция ранее разъединенных детализированных данных;
2. Разделение наборов данных используемых для операционной обработки и наборов данных используемых для решения задач анализа.

Автором концепции Хранилищ Данных (Data Warehouse) является Б.Инмон, который определил Хранилища Данных, как: "предметно ориентированные, интегрированные, неизменяемые, поддерживающие хронологию наборы данных, организованные для целей поддержки управления", призванные выступать в роли "единого и единственного источника истины" обеспечивающего менеджеров и аналитиков достоверной информацией необходимой для оперативного анализа и принятия решений.

Для правильного понимания данной концепции необходимо понимание следующих принципиальных моментов:

- Концепция Хранилищ Данных - это не концепция анализа данных, скорее это концепция подготовки данных для анализа.
- Концепция Хранилищ Данных не предопределяет архитектуру целевой аналитической системы. Она говорит о том, какие процессы должны выполняться в системе, но не о том, где конкретно и как эти процессы должны выполняться.
- Концепция Хранилищ Данных предполагает не просто единый логический взгляд на данные организации (как иногда это трактуется). Она предполагает реализацию единого интегрированного источника данных.

Концепция хранилища данных

Хранилище данных - предметно-ориентированный, интегрированный, неизменяемый, поддерживающий хронологию набор данных, организованный для целей поддержки управления.

Подход построения хранилища данных для интеграции неоднородных источников данных принципиально отличается от подхода динамической интеграции разнородных БД. Реально строится новое крупномасштабное хранилище, управление данными в котором происходит по другим правилам, чем в исходных оперативных БД.

Для того чтобы существующие хранилища данных способствовали принятию управленческих решений, информация должна быть представлена аналитику в нужной форме, то есть он должен иметь развитые инструменты доступа к данным хранилища и их обработки. Поддержка принятия управленческих решений на основе накопленных данных может выполняться в трех базовых сферах [21].

1. Сфера детализированных данных. Это область действия большинства систем, нацеленных на поиск информации. В большинстве случаев реляционные СУБД отлично справляются с возникающими здесь задачами. Общеизвестным стандартом языка манипулирования реляционными данными является SQL. Информационно-поисковые системы, обеспечивающие интерфейс конечного пользователя в задачах поиска детализированной

информации, могут использоваться в качестве надстроек как над отдельными базами данных транзакционных систем, так и над общим хранилищем данных.

2. Сфера агрегированных показателей. Комплексный взгляд на собранную в хранилище данных информацию, ее обобщение и агрегация, гиперкубическое представление и многомерный анализ являются задачами систем оперативной аналитической обработки данных (OLAP). Здесь можно или ориентироваться на специальные многомерные СУБД, или оставаться в рамках реляционных технологий. Во втором случае заранее агрегированные данные могут собираться в БД звездообразного вида, либо агрегация информации может производиться на лету в процессе сканирования детализированных таблиц реляционной БД.
3. Сфера закономерностей. Интеллектуальная обработка производится методами интеллектуального анализа данных (ИАД, Data Mining), главными задачами которых являются поиск функциональных и логических закономерностей в накопленной информации, построение моделей и правил, которые объясняют найденные аномалии и/или прогнозируют развитие некоторых процессов.

Полная структура информационно-аналитической системы, построенной на основе хранилища данных, показана на рис. 1. В конкретных реализациях отдельные компоненты этой схемы часто отсутствуют.

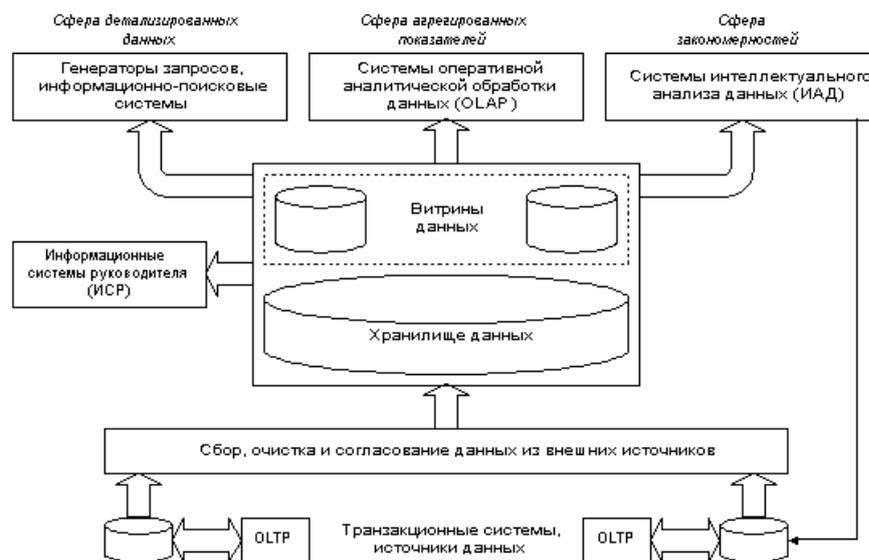


Рис. 1. Полная структура корпоративной информационно-аналитической системы (ИАС)

В соответствии с определением У. Инмона хранилище данных должно иметь следующие свойства:

- Предметная ориентация
- Интегрированность данных
- Инвариантность во времени
- Неразрушаемость - стабильность информации
- Минимизация избыточности информации

Предметная ориентация.

В хранилище данных информация ориентирована на задачи поддержки принятия решений. Для системы поддержки принятия решений (DSS) требуются "исторические" данные. Информация в хранилище данных организована в соответствии с основными аспектами деятельности предприятия (заказчики, продажи, склад и т.п.); это отличает хранилище данных от оперативной БД, где данные организованы в соответствии с процессами (выписка счетов, отгрузка товара и т.п.). Предметная организация данных в хранилище способствует как значительному упрощению анализа, так и повышению скорости выполнения аналитических запросов.

Интегрированность данных

В информационное хранилище обычно помещают данные из самых различных источников. Поэтому они имеют разные имена, атрибуты, единицы измерения и способы кодировки. Исходные данные извлекаются из оперативных БД, проверяются, очищаются, приводятся к единому виду, в

нужной степени агрегируются (то есть вычисляются суммарные показатели) и загружаются в хранилище. Такие интегрированные данные намного проще анализировать.

Инвариантность во времени

Данные в хранилище всегда напрямую связаны с определенным периодом времени. Данные, выбранные из оперативных БД, накапливаются в хранилище в виде "исторических слоев", каждый из которых относится к конкретному периоду времени. Это позволяет анализировать тенденции в развитии производства.

Неразрушаемость - стабильность информации

Из требований временной инвариантности следует, что однажды введенные в хранилище данные никогда не меняются. По отношению к ним возможны только две операции: начальная загрузка и чтение (доступ). Это и определяет специфику проектирования структуры базы данных для хранилищ. Если при создании OLTP-систем разработчики должны учитывать такие моменты, как откаты транзакций после сбоя сервера, борьба с взаимными блокировками процессов (deadlocks), сохранение целостности данных, то для хранилища данные проблемы не столь актуальны - перед разработчиками стоят другие задачи, связанные, например, с обеспечением высокой скорости доступа к данным.

Минимизация избыточности информации

Поскольку информация в хранилище загружается из OLTP-систем, возникает вопрос, не ведет ли это к чрезмерной избыточности данных? Нет, утверждает Билл Инмон. На самом деле избыточность минимальна (около 1%!), что объясняется следующими причинами:

- при загрузке информации из OLTP-систем в хранилище данные фильтруются. Многие из них вообще не попадают в хранилище, поскольку лишены смысла с точки зрения использования в системах поддержки принятия решений;
- информация в OLTP-системах носит, как правило, оперативный характер, и данные, потеряв актуальность, удаляются. В хранилище, напротив, содержится историческая информация, и с этой точки зрения перекрытие содержимого хранилища данными OLTP-систем оказывается весьма незначительным;
- в хранилище содержится некая итоговая информация, которая в базах данных OLTP-систем вообще отсутствует;
- во время загрузки в хранилище записи сортируются, очищаются от ненужной информации и приводят к единому формату. После такой обработки это уже совсем другие данные.

Основные компоненты информационного хранилища

ПО промежуточного слоя

Обеспечивает сетевой доступ и доступ к базам данных. Сюда относятся сетевые и коммуникационные протоколы, драйверы, системы обмена сообщениями и пр. Это, например, интерфейсы COM (Component Object Model), CORBA (Common Object Request Broker Architecture), JDBC (Java Database Connectivity - интерфейс взаимодействия с базами данных на языке Java), ODBC (Open Database Connectivity - открытый интерфейс взаимодействия с базами данных), OLE DB (Object Linking and Embedding Database)

Транзакционные БД и внешние источники информации

СУБД представляют собой основной источник информации для Хранилищ Данных. В последних выпусках реляционных и объектно-реляционных СУБД в дополнение к обычным типам индексации, основанной на B-деревьях и хэшировании, поддерживаются битовая индексация, R-деревья и другие виды индексов, играющие важную роль при организации хранилищ.

Уровень доступа к данным

Относящееся сюда ПО обеспечивает общение конечных пользователей с информационным хранилищем и загрузку требуемых данных из транзакционных систем. В настоящее время универсальным языком общения служит язык структурированных запросов (SQL).

Загрузка и предварительная обработка

Этот уровень включает в себя набор средств для загрузки данных из OLTP-систем и внешних источников. Выполняется, как правило, в сочетании с дополнительной обработкой: проверкой данных на чистоту, консолидацией, форматированием, фильтрацией и пр.

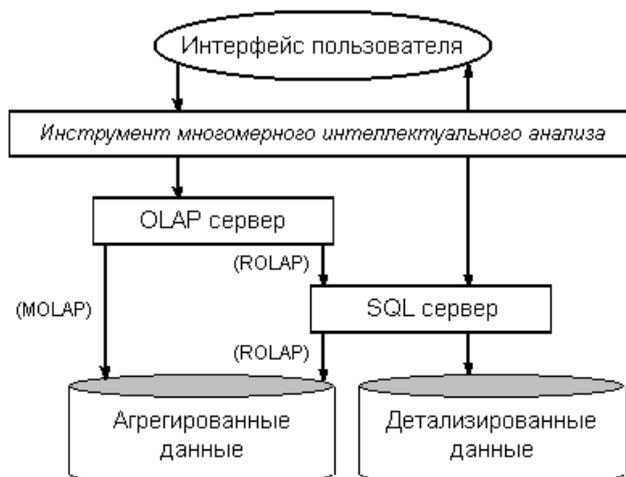


Рис. 2. Архитектура системы многомерного интеллектуального анализа данных

Обработку данных позволяют производить OLAP-системы (On-Line Analytical Processing - интерактивный, исследовательский анализ данных, основанный на выделении нужных слоев многомерных кубов и агрегировании по измерениям). Развитые OLAP-системы позволяют связывать индивидуальные значения с агрегатами и с другими значениями. Для этих систем важно то, что можно делать с данными, а не как они хранятся. Можно хранить кубы в специализированной многомерной базе данных (MOLAP) или с помощью РСУБД (ROLAP).

Информационное хранилище

Представляет собой ядро всей системы - один или несколько серверов БД.

Метаданные

Метаданные (репозиторий, "данные о данных"). Играют роль справочника, содержащего сведения об источниках первичных данных, алгоритмах обработки, которым исходные данные были подвергнуты, и т. д.

Уровень информационного доступа

Обеспечивает непосредственное общение пользователя с данными Хранилища посредством стандартных систем манипулирования, анализа и предоставления данных типа MS Excel, MS Access, Lotus 1-2-3 и др.

Уровень управления (администрирования)

Отслеживает выполнение процедур, необходимых для обновления информационного хранилища или поддержания его состояния.

Проблемы интеграции данных

Приступая к построению аналитической системы, необходимо понимать, что ее реализация невозможна без решения таких вопросов, как:

- Неоднородность программной среды
- Распределенный характер организации
- Повышенные требования к безопасности данных
- Необходимость наличия многоуровневых справочников метаданных
- Потребность в эффективном хранении и обработке очень больших объемов информации

Неоднородность программной среды

Основой Хранилища Данных являются разнообразные внешние источники: базы данных, электронные архивы, публичные и коммерческие электронные каталоги, справочники, статистические сборники и т.д. Все они строятся на различной программной и аппаратной основе. При реализации Хранилища следует учитывать типы взаимодействующих с ним программ, операционных систем и платформ, на которых эти программы функционируют. То есть средства реализации Хранилища должны быть интегрируемы с взаимодействующей с ним неоднородной программной средой. Практически всегда, задача построения Хранилища Данных, это задача построения единой согласованно функционирующей информационной системы, на основе

неоднородных программных средств и решений. И уже сам выбор средств реализации Хранилища Данных становится чрезвычайно сложной задачей. Здесь должно учитываться множество факторов, включая, взаимную совместимость различных программных компонент, легкость их освоения и использования, эффективность функционирования, стабильность и даже формы, уровень и потенциальную перспективность взаимоотношений различных фирм производителей.

Распределенный характер организации

В основе концепции Хранилищ Данных, лежит физическое разделение узлов, в которых выполняется операционная обработка, от узлов, в которых выполняется анализ данных. И хотя, при реализации такой системы, нет необходимости в строгой синхронизации данных в различных узлах (например, на основе средств двух фазной фиксации транзакций), средства асинхронной асимметричной репликации данных являются неотъемлемой частью практически любого решения.

Повышение требований к безопасности данных

Хранилище должно содержать информацию об истории развития радиотехнического предприятия, о его взаимоотношениях с поставщиками и заказчиками, об истории и состоянии рынка. Эта информация дает возможность анализа прошлой и текущей деятельности предприятия и построения прогнозов для будущего. Поэтому при построении Хранилища Данных остро встает вопрос обеспечения безопасности. В системах, основанных на Хранилищах Данных, оказывается недостаточной защита данных в стиле языка SQL, которую обеспечивают обычные коммерческие СУБД (этот уровень защиты соответствует классу С2 в соответствии с классификацией Оранжевой Книги Министерства обороны США). Для обеспечения должного уровня защиты доступ к данным должен контролироваться не только на уровне таблиц и их столбцов, но и на уровне отдельных строк (это уже соответствует классу В1 Оранжевой Книги). Приходится также решать вопросы аутентификации пользователей, защиты данных в местах их преобразования и согласования, в процесс их передачи по сети (шифрование паролей, текстов запросов, данных).

Необходимость наличия многоуровневых справочников метаданных

Наличие метаданных и средств их представления конечным пользователям является одним из основополагающих факторов успешной реализации Хранилища Данных. Для пользователя OLAP-системы требуются метаданные, по крайней мере, следующих типов:

- Описания структур данных, их взаимосвязей.
- Информация о находящихся в Хранилище Данных и поддерживаемых им агрегатах данных.
- Информация об источниках данных и о степени их достоверности.
- Информация о периодичности обновлений данных.
- Информация о владельцах данных.
- Статистические оценки времени выполнения запросов. До выполнения запроса полезно иметь хотя бы приблизительную оценку времени, которое потребуется для получения ответа, и объема этого ответа.

Потребность в эффективном хранении и обработке очень больших объемов информации

Как уже было сказано, Хранилище Данных должно содержать информацию о деятельности предприятия, по крайней мере, за последние несколько лет. Поэтому объем Хранилищ исчисляется сотнями гигабайт или даже терабайтами информации. Проблемой таких больших Хранилищ является то, что накладные расходы на внешнюю память возрастают нелинейно при возрастании объема Хранилища. Исследования, проведенные на основе тестового набора TPC-D, показали, что для баз данных объемом в 100 гигабайт потребуется внешняя память объемом в 4.87 раза большая, чем нужно собственно для полезных данных. При дальнейшем росте баз данных этот коэффициент увеличивается.

Реализация Хранилищ и Витрин Данных

Существует несколько вариантов реализации Хранилищ Данных:

- виртуальное хранилище данных;
- витрины данных;
- глобальное хранилище данных;
- многоуровневая архитектура хранилища данных.

Виртуальное хранилище данных

Основывается на репозитории метаданных, которые описывают источники информации (БД транзакционных систем, внешние файлы и др.), SQL-запросах для их считывания и процедуры обработки и предоставления информации. Непосредственный доступ к последним обеспечивает ПО промежуточного слоя. В этом случае избыточность данных нулевая. Конечные пользователи фактически работают с транзакционными системами напрямую.

Витрина данных

Витрина данных (Data Mart) - это набор тематически связанных баз данных, которые содержат информацию, относящуюся к отдельным аспектам деятельности корпорации. По сути дела, Витрина Данных - это облегченный вариант Хранилища Данных, содержащий только тематически объединенные данные. Целевая база данных максимально приближена к конечному пользователю и может содержать тематически ориентированные агрегатные данные. Витрина Данных, естественно, существенно меньше по объему, чем корпоративное хранилище данных, и для его реализации не требуется особо мощная вычислительная техника.

Глобальное хранилище данных

В последнее время все более популярной становится идея совместить концепции Хранилища и Витрины Данных в одной реализации и использовать Хранилище Данных в качестве единственного источника интегрированных данных для всех Витрин Данных. Тогда естественной становится такая трехуровневая архитектура системы:

На первом уровне реализуется корпоративное Хранилище Данных на основе одной из развитых современных реляционных СУБД. Это хранилище интегрированных в основном детализированных данных. Реляционные СУБД обеспечивают эффективное хранение и управление данными очень большого объема, но не слишком хорошо соответствуют потребностям OLAP-систем, в частности, в связи с требованием многомерного представления данных.

На втором уровне поддерживаются Витрины Данных на основе многомерной СУБД (примером такой системы является Oracle Express Server). Такие СУБД почти идеально подходят для целей разработки OLAP-систем, но пока не позволяют хранить сверхбольшие объемы данных (предельный размер многомерной базы данных составляет 10-40 Гбайт). В данном случае это и не требуется, поскольку речь идет о Витринах Данных. Заметим, что Витрина Данных не обязательно должна быть полностью сформирована. Она может содержать ссылки на Хранилище Данных и добирать оттуда информацию по мере поступления запросов. Конечно, это несколько увеличивает время отклика, но зато снимает проблему ограниченного объема многомерной базы данных.

Наконец, на третьем уровне находятся клиентские рабочие места конечных пользователей, на которых устанавливаются средства оперативного анализа данных.

Основные характеристики СУБД, необходимые для построения Хранилища Данных.

- **Открытость** - интегрируемость с различными платформами и операционными системами. Необходимость наличия такой характеристики вытекает из неоднородности программной среды.
- **Простота использования и управления.** Поскольку немалую часть затрат составят затраты на поддержку и обучение персонала и это обучение может занять большой период времени, то стоит сократить эти расходы до минимума. СУБД должна иметь интуитивно понятный интерфейс и обеспечивать простоту управления. К этой же характеристике можно отнести и поддержку национальных языков.
- **Расширяемость.** Хранилище данных содержит данные различных типов. Поэтому СУБД, на основе которой строится хранилище, должна иметь возможность поддержки мультимедийных и комплексных типов данных, таких как документы, изображения, аудио, видео, пространственные данные, временные ряды и т.д. Кроме того, сюда могут входить специализированные для конкретной отрасли объекты, такие как рентгеновские снимки, отпечатки пальцев, конструкторские чертежи, формы страховых исков и т.д.

- **Поддержка Web.** Возможность обращаться к данным с помощью браузеров Web и использовать сетевые технологии, такие как Java, для работы в составе систем электронной коммерции и корпоративных сетей intranet.
- **Производительность.** Высокая производительность позволяет уменьшить время ответа на запрос. Она обеспечивается следующими механизмами:
 - фрагментация таблиц;
 - асинхронный ввод/вывод;
 - параллельная обработка запросов;
 - распределенная обработка транзакций;
 - наличие хранимых процедур и триггеров.
- **Поддержка OLAP.**

Характеристики СУБД ведущих компаний (Microsoft SQL Server 7.0, Oracle 8i, IBM DB2 Universal Database, Informix Online Dynamic Server, Adabas, Sybase System 11, Ingres II) представлены в таблице 1.

Решения ведущих компаний в области создания Хранилищ Данных

Microsoft

Цель Microsoft - создать всеобъемлющую стратегию хранения данных, чтобы упростить разработку и реализацию экономически эффективных решений для хранения данных с помощью SQL Server. В состав стратегии хранения данных Microsoft входят следующие компоненты:

- Усовершенствование продукта Microsoft SQL Server, включая усовершенствованный процессор запросов, средства доставки информации и преобразования данных, интерфейсы Data Warehousing Framework и интеграцию с Интернетом.
- Microsoft “Plato”. Поддерживает полную многомерную оперативную аналитическую обработку данных, реляционную аналитическую обработку, а также гибридные реализации, предоставляя разработчику выбор модели, наиболее подходящей для нужд организации.
- Microsoft Repository. Репозиторий служит для совместного использования информации, нужной для приложений, связанных с хранением данных: схем, метаданных и преобразований. Новые дополнительные средства репозитория позволяют расширить рынок продуктов для хранения данных, обеспечивая открытую инфраструктуру, общую для разных подходов к проектированию баз и хранилищ данных и для разных поставщиков аналитических инструментальных средств.

SQL Server 7.0 предоставляет следующие средства для хранения данных и создания Витрин Данных:

- Data Transformation Services. Службы преобразования данных Data Transformation Services обеспечивают функции для импорта, экспорта и преобразования данных между Microsoft SQL Server и любыми системами, поддерживающими OLE DB, ODBC, или просто текстовыми файлами. С помощью DTS на базе SQL Server можно строить Хранилища и Витрины Данных, извлекающие и преобразующие данные из многих разнородных источников в диалоговом режиме или автоматически, на основе постоянного расписания (без участия пользователя). Кроме того, DTS интегрированы с Microsoft Repository, где они хранят метаданные, пакеты преобразования, а также историю данных, подвергавшихся преобразованию, включая их исходное состояние. Microsoft “Plato”.
- PivotTable Service. Microsoft PivotTable Service сопутствует “Plato”, обеспечивая клиентским приложениям доступ к данным оперативной аналитической обработки. С использованием Microsoft PivotTable Service могут быть разработаны инструментальные средства для анализа и представления данных оперативной аналитической обработки.
- English Query. Среда Microsoft English Query позволяет разработчикам превратить свои реляционные базы данных в приложения English Query. Это дает пользователям возможность задавать вопросы на английском языке, а не писать запросы с помощью операторов SQL.
- Мастера подсказок и шаблоны-заготовки, входящие в состав Microsoft Management Console. Консоль управления MMC имеет широкий набор удобств и возможностей управления для

улучшения интеграции хранилищ данных. Чтобы помочь пользователям справиться с типичными задачами, предусмотрен специальный набор заготовок задач.

Oracle

Решение компании Oracle в области хранилищ данных основывается на двух факторах: широкий ассортимент продуктов самой компании и деятельность партнеров в рамках программы Warehouse Technology Initiative. Возможности Oracle в области хранилищ данных базируются на следующих составляющих:

- наличие реляционной СУБД Oracle 8, которая постоянно совершенствуется для лучшего удовлетворения потребностей хранилищ данных;
- существование набора готовых приложений, обеспечивающих возможности разработки хранилища данных;
- высокий технологический потенциал компании в области анализа данных;
- доступность ряда продуктов, производимых другими компаниями.

Oracle Data Mart Suite обеспечивает наиболее полное на сегодняшний день решение для построения витрин данных. В состав Oracle Data Mart Suite входят база данных Oracle8, средства проектирования, переноса и преобразования данных, программное обеспечение для анализа и отчетности, ориентированное на конечных пользователей, и документация в стиле "поваренной книги" для быстрого и простого внедрения витрины данных.

Oracle Data Mart Suite — это интегрированный пакет, в состав которого входят следующие компоненты:

- Oracle8 Enterprise Edition.
- Oracle Enterprise Manager
- Oracle Data Mart Designer. Компонента Oracle Data Mart Designer предлагает средства для автоматического представления структуры имеющихся баз данных в виде визуальных диаграмм, извлекая ее либо напрямую из словаря базы данных, либо из файлов описаний, содержащих операторы определений DDL языка SQL. Data Mart Designer может провести "обратное проектирование" (reverse engineering) как базы данных Oracle, так и других. Эта информация записывается в репозиторий Data Mart Designer для последующего использования компонентой Oracle Data Mart Builder.
- Oracle Data Mart Builder является ключевым элементом для быстрого проектирования и внедрения витрины данных.
- Oracle Discoverer Administration Edition
- Oracle Discoverer User Edition
- Oracle Application Server Enterprise Edition
- Oracle Reports
- Developer Server

Компания IBM

Решением компании IBM является **Visual Warehouse** представляет собой набор интегрированных инструментальных средств для построения хранилища данных и включает компоненты для определения отношений между источниками данных и данными в хранилище, преобразования и очистки данных из источников, автоматизации процессов загрузки данных в хранилище и текущего управления хранилищем. Построенный на базе DB2, Visual Warehouse может получать данные из баз данных DB2, Informix, Microsoft, Oracle, Sybase, IMS databases, VSAM и плоских файлов, а также источников, поддерживаемых DataJoiner.

Кроме того, существуют варианты Visual Warehouse, опционально включающие средства Business Objects или Cognos для доступа и анализа информации. Базовый набор для Visual Warehouse содержит:

- **DB2 Universal Database** для хранения метаданных.
- **Visual Warehouse Manager** для определения и управления процедурами выборки данных и загрузки их в хранилище.
- **Visual Warehouse Agent** на различных платформах для выполнения операций по выборке, преобразованию и загрузке данных.

- **Visual Warehouse Information Catalog** (известный также как **DataGuide**) - для доступа к метаданными для пользователей.

Версия **Visual Warehouse OLAP** включает дополнительно DB2 OLAP Server, позволяя пользователям как определять и загружать данные в базы данных со схемой звезды, так и автоматически выполнять при загрузке данных предварительные вычисления и агрегации данных.

Visual Warehouse обладает возможностями, которые позволяют очень эффективно организовать функционирование хранилища данных: использование технологии агентов, использование метаданных и возможность вызывать написанные пользователем и третьими фирмами компоненты для дополнительной расширенной обработки данных.

Informix Software

Стратегия компании в отношении Хранилищ Данных направлена на расширение рынка для ее продукта *On-Line Dinamic Parallel Server*. Предлагаемая архитектура Хранилища Данных базируется на четырех технологиях: реляционные базы данных, программном обеспечении для управления хранилищем данных, средствах доступа к данным и платформе открытых систем. Три последние компонента разрабатываются партнерами компании. После выхода Универсального Сервера, основанного на объектно-реляционном подходе, можно ожидать, что и он будет использоваться для построения Хранилищ Данных.

Software AG

Деятельность компании в области Хранилищ Данных происходит в рамках программы *Open Data Warehouse Initiative*. Программа базируется на таких продуктах компании, как:

- СУБД ADABAS;
- Natural 4GL,
- SourcePoint - средство управления Хранилищем Данных. SourcePoint позволяет автоматизировать процесс извлечения и пересылки данных, а также их загрузки в Хранилище Данных.
- EntireX – предназначен для объединения распределенных приложений и программных компонент в масштабах предприятия. Кроме того, в состав пакета EntireX включена реализация на платформах больших ЭВМ и UNIX стандарта DCOM фирмы Microsoft для платформы Windows, что, в совокупности, позволяет решать задачу интеграции всех информационных технологий, применяемых на предприятии.
- Tamino - обеспечивает доступ ко всем типам накопленных данных (тексты, таблицы, видео и аудио) и их конвертацию в XML-объекты. Tamino делает возможным доступ новых Web-приложений к старым базам данных.

Sybase

Развитие технологий хранилищ данных объясняется необходимостью видеть картину своего бизнеса в целом и принимать информированные решения. Хранение исторических, обобщенных данных о работе компании и их анализ позволяют находить новые источники прибыли, новые резервы повышения эффективности, совершать прорывы в конкурентной борьбе. Стратегия компании в области хранилищ данных основывается на разработанной ей архитектуре Warehouse WORKS.

Sybase предлагает следующие технологии для построения хранилищ данных:

Серверная платформа Adaptive Server IQ - это СУБД, оптимизированная для анализа данных на уровне физического дизайна. Уникальная архитектура IQ позволяет обрабатывать незапланированные аналитические запросы в десятки-сотни раз быстрее, чем традиционные СУБД. При этом вместо разбухания данных в хранилище происходит их сжатие.

Средства проектирования, администрирования и работы с хранилищем Sybase Warehouse Studio значительно упрощает процесс разработки и обслуживания хранилища. В комплект входят Warehouse Architect: CASE-средство проектирования хранилища; Warehouse Control Center: средство управления мета-данными и администрирования хранилища; Infomaker: генератор отчетов и пр.

Среда разработки процессов миграции данных Ardent DataStage - средство, упрощающее и автоматизирующее процесс извлечения, обработки, консолидации, переработки, очистки и загрузки информации в хранилище.

Enterprise Application Studio - объединяет в себе технологию быстрой разработки приложений (RAD) PowerBuilder®, уже получившую признание среду разработки Java™-приложений PowerJ и высокопроизводительную среду развертывания приложений Enterprise Application Server™.

PowerBuilder - объектно-ориентированный инструмент для разработки и развертывания как клиентских, так и серверных приложений.

PowerJ- новый интегрированный инструмент разработки на языке Java. Новая открытая среда приложений, которая предназначена для построения, управления и развертывания промышленных Web-приложений.

PowerDesigner - интегрированный набор CASE-инструментов для проектирования функциональной модели данных, а также логической и физической структур для более чем 40 СУБД.

Промежуточное ПО Sybase включает:

- Доступ к данным (Data access) - подключает любого клиента к корпоративной информационной сети. Семейство программных продуктов Sybase EnterpriseConnect™ включает в себя: DirectConnect™, OmniConnect™, jConnect™, Open ServerConnect™, Open ClientConnect™ и InfoHub.
- Перемещение данных (Data movement) внутри распределенной системы и между географически отдаленными друг от друга точками для обеспечения обмена транзакциями в режиме реального времени как для работы хранилищ данных, так и для систем поддержки принятия решений. Семейство программных продуктов Sybase Data Movement включает: Replication Server™, Replication Agents™, Replication Toolkit™ for MVS, SQL Anywhere™ и SQL Remote™.
- Интеграция и настройка приложений, которые позволяют легко связывать между собой многоуровневые (клиент - промежуточный слой - сервер) приложения.

Заключение

Современные компании предлагают различные варианты решений создания Хранилищ Данных или, по крайней мере, Витрин Данных на основе СУБД и OLAP-систем. Каждый из них содержит свои плюсы и минусы. Однако для принятия правильных и своевременных решений по управлению радиотехническим предприятием необходимо создать для него хорошую информационную инфраструктуру и упростить доступ к данным для оперативного анализа процесса производства. Это может быть достигнуто при совместном применении таких информационных технологий, как:

- реляционные и объектно-реляционные СУБД;
- технология Хранилищ Данных;
- технология оперативной аналитической обработки;
- технология Web.

Список использованных источников

1. А. Сахаров «Концепции построения и реализации информационных систем, ориентированных на анализ» - «СУБД» – 1996 - №4
2. С. Кузнецов «Обзор возможностей применения ведущих СУБД для построения хранилищ данных» - <http://www.citforum.ru>
3. С. Д. Коровкин, И. А. Левенец, И. Д. Ратманова, В. А. Старых, Л. В. Щавелёв «Решение проблемы комплексного оперативного анализа информации хранилищ данных» - <http://www.citforum.ru>
4. Н. Игнатович «DB2 Universal Database – ключевые характеристики» - <http://www.citforum.ru>
5. Н. И. Вьюкова «Архитектура сервера INFORMIX – OnLine Dynamic Server 7.1 и коммуникационные средства» – «JetINFO» - 1995 - №2.

6. Т. В. Грачева «ADABAS – основа технологий SOFTWARE AG» – «СУБД» – 1995 - №2.
7. Джо Салеми «Microsoft революционизирует SQL Server» – «LAN/Журнал сетевых решений» – 1996 - №1.
8. А. В. Лашманов «Oracle – история, состояние и перспективы» - 1995 - №1.
9. А. Тандоев «Архитектура продуктов клиент – сервер фирмы SYBASE» - «СУБД» – 1995 - №1.

Характеристики ведущих СУБД.

Таблица 1

	Microsoft SQL Server 7.0	Oracle 8i	IBM DB2 Universal Database	Informix Online Dynamic Server	Adabas	Sybase System 11
Работа под управлением различных ОС:	Windows NT	Windows NT, Unix	Windows NT, OS/2, Unix	Windows NT, Unix, NetWare, DOS	Unix, OpenVMS, Windows NT и Windows 95/98	Novell NetWare, Windows NT, OS/2, SCO UNIX
Работа на различных платформах	“открыт” для других продуктов Microsoft	Sun, HP, IBM RS/6000, DEC Alpha, NCR, NEC, Sequent, Silicon Graphics, Unisys, Windows NT и др.	AIX, Linux, HP-UX, OS/2, Solaris, Windows NT, Windows 2000, SCO UnixWare	Sequent, HP, Sun, IBM, Siemens Nixdorf, NCR	Sun, HP, IBM RS/6000, DEC Alpha, NCR, NEC, Sequent, Silicon Graphics, Unisys, Windows NT и др.	Sun, HP, IBM RS/6000, DEC Alpha, NCR, NEC, Sequent, Silicon Graphics, Unisys, Windows NT и др.
Простота использования и управления	Отлично	Хорошо	Хорошо	<i>Хорошо</i>	Хорошо	Хорошо
Обработка аудио, видео, изображений	Плохо	Отлично	Отлично	Хорошо	Плохо	Хорошо
Текстовые данные	Хорошо	Отлично	Отлично	Хорошо	Хорошо	Хорошо
Администрирование в оперативном режиме	+	+	+	+	+	+
Поддержка Web	Хорошо	Отлично	Отлично	Хорошо	Хорошо	Хорошо
Фрагментация таблиц	+	+	+	+	+	+
Асинхронный ввод/вывод	+	+	+	+	+	+
Параллельная обработка запросов	+	+	+	+	+	+
Распределенная обработка транзакций	+	+	+	+	+	+
Хранимые процедуры и триггеры	+	+	+	+	+	+
Язык SQL	+	+	+	+	+	+
OLAP	+	+	+	+	+	+

Богданов В.

научный руководитель: к.т.н. Власов А.И.

Московский Государственный Технический Университет им. Н.Э. Баумана,
кафедра “Конструирование и производство ЭА”.

INFORMATIONAL EDUCATIONAL SYSTEM LINUXWORLD

Vladimir Bogdanov

the scientific chief: Ph.D. Vlasov A.I.

Bauman Moscow State Technical University,
Department of Design and Equipment of Electronic System.

Аннотация: После многочисленных попыток работы с платформой Linux, и не особо плодотворных поисков требуемой по данной платформе информации в сети Internet, мы пришли к выводу о необходимости и целесообразности создания сайта для поддержки разработчиков и администраторов информационных систем на базе платформы linux. В качестве катализатора идеи послужило отсутствие подобной системы на сервере нашей кафедры.

Abstract: After numerous attempts of activity with a gantry Linux, and not it is apart of fruitful looking ups of the information, demanded on the given gantry, in a network Internet, we came to conclusions about necessity and expediency of creation of a site for support of the implementators and managers of intelligence systems on the basis gantries linux. As catalyst of idea the absence of a similar system on the server of our stand has served.

В основу разработки были положены идеи интуитивно понятного, дружелюбного, эргономичного интерфейса, а также доступность и простота получения информации. Проанализировав некоторое число реализаций информационно-обозревателей систем, мы эргономичный графический интерфейс интерактивного взаимодействия между пользователем и удаленной информационной системой и организацию структурированного доступа к разделам обозревателя. При разработке графической части сервера мы использовали пакет графической верстки Gnu Image Manipulation Program, известный под своим сокращением the Gimp.

Настоящая графическая оболочка уже вторая на пути развития нашего информ-сервера. Оболочка была дополнена и были внесены небольшие изменения в соответствии с пожеланиями. При конечном оформлении была использована технология Java для достижения оптимального эффекта в привлекательности сайта.

Проделав огромную работу по подбору и систематизации информации мы разработали приятную, с точки зрения простоты поиска необходимых документов, а также навигационную структуру сайта. При помощи структурированности данных обеспечивается легкий доступ ко всем обзорам, посвященным конкретным проблемам, с которыми может столкнуться разработчик и администратор. С помощью технологии HyperText мы обеспечили простую связь с другими информационно-обозревательными системами. В частности с помощью этой технологии мы обеспечили связь с обозревателями серверами разработчиков программного обеспечения, проблемам настройки которого мы посвятили нашу работу. Информацию о результатах исследования рядов вопросов администрирования Linux, мы представили в удобной и доступной форме в разделах нашей системы.

Для работы по созданию самой информационно-обозревательной системы были использованы различные пакеты, такие как Macromedia DreamWeaver 3.0, BlueFish из состава поставки дистрибутива Linux Mandrake 7.0 и Netscape Composer.

В качестве заключения... Наш сайт еще молод, но у нас есть много задумок, по поводу его дальнейшего развития. В процессе написания форум, и в качестве идей, пока, чат и конференция. Все это только вопрос времени.

ВИРТУАЛЬНЫЙ МУЗЕЙ КОМПЬЮТЕРНОЙ ТЕХНИКИ

Харченко А.А.

Научный руководитель к.т.н. Власов А.И.

IU4, BMSTU

ArtemKh@mail.ru, <http://museum.iu4.bmstu.ru>

Аннотация. Статья посвящена Интернет-проекту «Виртуальный музей компьютерной техники», методам и принципам его построения, а также его роли в дистанционном и открытом образовании, обучении школьников и студентов. В статье рассматриваются причины возникновения проекта и пути его дальнейшего развития.

Abstract. This article is devoted to the Internet-project called “Virtual Museum of Computers”, methods and principles of its construction, and the role, which it plays in open and distant education, training of pupils and students. Reasons of the project rise and ways of its further development are also presented in the article.

Проект «Виртуальный музей компьютерной техники» родился как средство для дистанционного обучения. Но по мере развития проекта, перед его создателями и участниками рождались все новые цели. Сейчас, когда проект уже практически завершен, т.е. сайт готов принимать посетителей, выработаны общие принципы дизайна и построения Интернет-странички, и последующие обновления не повлияют на общее восприятие сайта, можно твердо выделить следующие причины возникновения и цели, преследуемые участниками проекта:

1. Дистанционное и открытое образование, как уже упоминалось ранее. Это, пожалуй, самая главная причина. Наш сайт является одним из немногих проектов в области развития компьютерной техники от самых истоков до нынешних времен. Это позволяет людям, интересующимся компьютерной историей, не бродить в Сети в поисках крупиц информации, сосредоточенных на разных сайтах, и, чаще всего, не являющихся основной темой сайта.
2. «Внутреннее» образование школьников и студентов. Это причина касается другой области образования, в частности, обучение студентов принципам Web-дизайна и Web-мастеринга, и, что немаловажно, самостоятельное обучение. Главная цель – это постановка студентом перед собой задачи. В процессе ее решения он получит необходимые знания самостоятельно, роль научного руководителя здесь – это направить студента в нужное русло, все остальное он получает своими руками. Например, поставив перед собой цель – изучить язык Perl, студент постепенно идет к ней. Научный руководитель обращает внимание на применение Perl на практике, как можно реализовать это в рамках проекта (напр. создать Perl-скрипт гостевой книги), а также на нюансы, связанные с этим языком (такие, как безопасность скриптов). Благодаря такому подходу к проектированию, участник проекта делает максимальный вклад в его развитие, т.к. в выборе технологий он не ограничен и занят интересной ему темой, и таким образом, проект стремится быть как можно более современным.
3. Систематизация информации и восполнение пробелов в истории компьютеров. На момент создания сайта (сентябрь 1999г.) в нашей стране подобных проектов существовало крайне мало (сейчас, впрочем, их тоже немного), информация была разрознена и разбросана по разным сайтам, энциклопедиям и т.п. За рубежом подобной информации гораздо больше, там существуют специальные музеи подобно нашему, а каждая фирма с многолетней историей считает обязательным разделом своего сайта свою историю. Необходимо было как-то систематизировать всю имеющуюся информацию, для этого было создано несколько разделов: «Отечественная техника», где была собрана информация об отечественных разработках; «Зарубежная техника», здесь использовалась информация не столько о разработках, сколько об истории развития тех или иных фирм-производителей техники; «От Абака до Pentium» - это

«биография» компьютера от простейших вычислительных средств древности до теперешних многопроцессорных станций; «Персоналии» - биографии людей, внесших большой вклад в развитие компьютеров, этот раздел впоследствии был разбит на два подраздела – «Отечественные деятели» и «Зарубежные деятели».

Схема построения сайта крайне проста: каждый участник выбирает себе интересную ему тему или «свободный» раздел и постепенно начинает наполнять его информацией. В выборе средств он не ограничен, т.е. может использовать практически все доступные на сегодняшний день средства создания сайтов будь то обычный HTML, или «продвинутое» Flash-технологии. Наиболее сложным моментом создания сайта оказался дизайн. Это неудивительно, поскольку каждый участник имеет свое мнение об общем дизайне сайта. В связи с этим было решено, что каждый участник будет «строить» и оформлять свой раздел так, как он считает нужным, но при этом не уходя далеко от общего дизайна главной странички, т.е. используя подобную цветовую гамму, систему меню и т.п.

Дальнейшее развитие сайта планируется осуществить по трем направлениям:

1. Информационное. Здесь подразумевается наполнение сайта информацией об истории компьютеров, а также об их дальнейшем развитии в будущем. Хотелось бы открыть еще несколько разделов, тем более, что тем для описания и обсуждения предостаточно, но прежде всего, конечно, стоит дополнить уже существующие разделы как можно большим количеством информации.
2. Визуальное. Это выработка общего дизайна и подстройка существующего материала под разные клиентские машины, т.е. разработка «зеркал» музея для пользователей разных платформ, перекодировка в системы mac, iso, dos, win, koï8-r, разработка Flash-версии.
3. Привлечение посетителей и налаживание обратной связи. Планируется создание разделов «ЧАТ», «Гостевая книга», «Форум». В целях привлечения посетителей планируется провести рекламные акции такие, как обмен баннерами, регистрация сайта в популярных поисковых системах.

Планов много и с каждым днем их не становится меньше. Хочется реализовать как можно больше, и мы надеемся, что это осуществимо. Также хочется надеяться на то, что этот проект будет интересен следующим поколениям студентов, и нам будет кому передать его после окончания МГТУ.

МЕТОДЫ ПРОГНОЗИРОВАНИЯ ТЕНДЕНЦИЙ И ПОВЕДЕНИЯ РЫНКА МЕЖБАНКОВСКИХ КРЕДИТОВ.

Баев А. Б.

к.т.н. Яковлева Г. Л.

ИУ5, МГТУ им Н.Э. Баумана, Москва, Россия.

**METHODS FOR FORECAST TENDENCIES AND CHANGES OF
INTER-BANK CREDIT MARKET**

A. Baev

professor-assistant G. Yakovleva

IU5, Bauman Moscow State Technical University, Moscow, Russia

alexbaev@mail.ru

Аннотация. В данной работе рассмотрен круг вопросов, связанных с применением стандартных методов регрессионного анализа и нейросетевых алгоритмов к прогнозированию поведения динамических систем на примере рынка межбанковских кредитов.

Abstract. This article deals with the questions of applying standard methods of analyses and neuronets methods to forecast changes of dinamic system on bases of interbank credit market

Одной из актуальных задач, встающих перед экспертами и аналитиками различных областей, является прогнозирование состояния динамической системы исходя из её поведения в течение некоторого промежутка времени. Математический аппарат решения таких задач представлен широким спектром алгоритмов и методов аппроксимации функций. К классическим относятся: интерполяционные полиномы Лагранжа, многочлены Эрмита, метод наименьших квадратов, естественные кубические сплайны и т.д. Разработка программного обеспечения, реализующего данный математический аппарат, достаточно успешно проводилась рядом независимых организаций и программистов-одиночек. Тем не менее, программная реализация, в силу большого числа параметров, влияющих на прогнозируемую величину, их постоянное обновление, ограниченность конкретных методов и т.д., полностью решить задачу прегнозирования не позволила. Сложилась ситуация, когда для того, чтобы из ряда значений “вытянуть аналитику”, приходилось использовать не только элементы “классического” подхода, но и загружать числовые данные в Excel, или аналогичную программу, пытаясь осуществить какие-либо манипуляции вручную. Такие ситуации имели место, например, в коммерческом банке “Огни Москвы” и ряде других финансово-промышленных компаний.

Задача прогнозирования может быть поставлена следующим образом: имеется объект прогноза и набор числовых характеристик z_q ($q = 1, \dots, m$), описывающих его поведение (считается, что определены наиболее значимые характеристики). Каждая из характеристик в моменты времени $t_i + \Delta t$ принимает значение z_q^i ($i = 1, \dots, n+1$). Таким образом, имеется матрица аргументов прогноза $X = \|z_q^i\|$ ($i = 1, \dots, n$), ($q = 1, \dots, m$) и вектор $x^{(n+1)} = (z_1^{(n+1)}, \dots, z_m^{(n+1)})$. Значения прогнозируемой величины y_i известны в точках $t_i + \Delta t$ ($i = 1, \dots, n$). Необходимо определить $y_{(n+1)}$, т.е. значение прогнозируемой величины Y в точке $t_{(n+1)} + \Delta t$.

При этом для системы любой природы различают краткосрочный и долгосрочный прогнозы. В случае краткосрочного прогноза считается, что все участвующие в нём критерии на прогнозируемую дату известны, в случае долгосрочного - критерии на прогнозируемую дату не известны и должны быть указаны ожидаемые значения и погрешности.

Решение такой задачи (регрессионного анализа) имеет общие независящие от природы прогнозируемой величины этапы:

1. определение оптимальной, исходя из определённых пользователем критериев, аппроксимирующей функции (формирование математической модели процессса);
2. определение значения функции в требуемой точке и вычисление погрешности прогноза ϵ .

При этом ряд задач помимо, собственно, прогнозирования предполагает дальнейшее использование полученной математической модели. В этом случае вид математической модели, её сложность имеют важные значение для исследователя.

Однако, помимо классических методов, сегодня богатейшие возможности для разработки такого рода автоматизированных систем прогнозирования (АСП) представляет аппарат естественных нейронных сетей, реализующих распределённую обучаемую систему.

Оба этих подхода к разработке АСП и рассмотрены в данной работе.

Классический подход в реализованной нами АСП представлен методом наименьших квадратов (МНК), интерполяционными многочленами Лагранжа и естественными кубическими сплайнами.

Нетрудно видеть, при интерполировании происходит повторение ошибок наблюдений, в то время как при обработке экспериментальных данных желательно, напротив, их сглаживание. В связи с этим наибольший интерес из классических методов представляет МНК.

Математическая модель здесь ищется в виде:

$$\Phi_m(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \dots + a_m\varphi_m(x) \quad (1)$$

где: a_i ($i=1..m$) – параметры модели,

$\varphi_k(x)=x_k$ – базисные функции.

Согласно критерию МНК параметры выбираются так, чтобы минимизировать среднеквадратичное отклонение:

$$\delta(\Phi_m, y) = \sqrt{\frac{1}{n+1} \sum_{i=0}^n (\Phi_m(x_i) - y_i)^2} \quad (2)$$

В результате задача сводится к решению системы линейных алгебраических уравнений (нормальной системы уравнений метода наименьших квадратов):

$$\sum_{j=0}^m \left(\sum_{i=0}^n \varphi_j(x_i)\varphi_k(x_i) \right) a_j = \sum_{i=0}^n y_i \varphi_k(x_i) \quad (k=0,1,\dots,m) \quad (3)$$

Для решения этой системы уравнений было предложено использовать метод Гаусса.

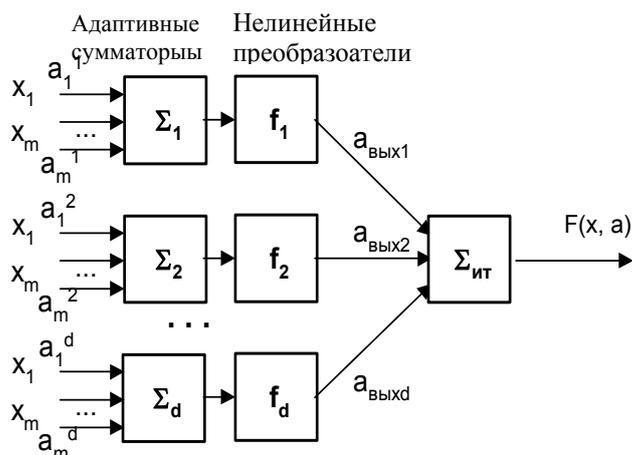
Манипуляция степенью аппроксимирующего полинома в совокупности с возможностью визуального представления модели и автоматического определения погрешности измерения позволяет осуществить подбор оптимальной модели.

Метод Лагранжа также даёт полиномиальную модель. Однако, использование этого метода имеет смысл лишь в том случае, если величина погрешности экспериментальных данных, как и величина шага, с которым производились измерения, достаточно малы.

Использование кубических естественных сплайнов, возможное в случае, когда погрешность экспериментальных данных заведомо мала, даёт (в визуальном отношении) очень хорошие результаты, как правило, по своим характеристикам превосходящие таковые, полученные при использовании метода Лагранжа. В этом случае математическая модель имеет достаточно специфический вид.

Таким образом, поставленная задача прогнозирования может быть решена с применением классических методов. Одним из достоинств в этом случае является “прозрачность” формируемой математической модели. Однако, привлечение классических методов к решению многокритериальных задач прогнозирования имеет результатом значительное усложнение математической модели, сложность её программной реализации, количество критериев и время необходимое АСП для выдачи результата находятся в линейной зависимости. При этом подразумевается знакомство исследователя с используемым им математическим аппаратом.

Так же для рассмотренной выше задачи прогнозирования была разработана и программно реализована нейронная сеть следующей структуры.



Перед подачей аргументов прогноза на вход нейросети выполняется их предобработка: нормирование и масштабирование. Соответственно, после получения выходных сигналов нейросети выполняется их интерпретация.

Рис.1. Реализованная нейронная сеть.

В качестве нелинейных преобразователей использовались функции:

$$f(x, a) = 1/[1 + \text{EXP}-(x, a)], \quad (4)$$

$$f(x, a) = 1/[w + (x, a)^y], \quad (5)$$

где a - вектор параметров размерностью m ; v, w - числовые константы.

При разработке АСП обучение нейронной сети проводили на всех векторах обучающего множества, комбинируя градиентные и стохастические методы: антиградиентного спуска, методы Больцмана и Коши. Критерием качества (целевой функцией) был выбран метод наименьших квадратов.

Суть данных методов заключается в следующем: выполняется поиск локального минимума одним из методов антиградиентного спуска, затем, для выхода из него и поиска глобального осуществляются псевдослучайные изменения величин весов. При этом те изменения, которые ведут к улучшениям, сохраняются. Начальные значения весов выбираются случайным образом. После коррекции веса на небольшое случайное число предьявляется множество входов и вычисляются выходы. Далее необходимо:

1. сравнить эти выходы с желаемыми и вычислить величину разности между ними. Для каждой обучаемой пары вычисляется значение целевой функции. Целью обучения является минимизация целевой функцией;
2. вычислить вес с помощью антиградиентного и стохастического методов. Если коррекция уменьшает целевую функцию, то сохранить ее, в противном случае вычислить вероятность сохранения по формуле (6) или (7);
3. повторять шаги с 1 до 3 до тех пор, пока сеть не будет обучена в достаточной степени.

Для метода Больцмана: если изменение веса приводит к увеличению целевой функции, то вероятность сохранения этого изменения вычисляется с помощью распределения:

$$P(c) = \exp(-c/kT), \quad (6)$$

где $P(c)$ – вероятность изменения c в целевой функции; k – константа, аналогичная константе Больцмана, выбираемая в зависимости от задачи; T – искусственная температура.

Выбирается случайное число r из равномерного распределения от нуля до единицы. Если $P(c)$ больше, чем r , то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению.

Это позволяет системе делать случайный шаг в направлении, портящем целевую функцию, позволяя ей тем самым вырваться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

В методе обучения Коши при вычислении величины шага распределение Больцмана заменяется на распределение Коши, которое имеет вид:

$$P(x) = \frac{T(t)}{T(t)^2 + x^2} \quad \text{где } P(x) \text{ есть вероятность шага величины} \quad (7)$$

Распределение Коши имеет бесконечную (неопределенную) дисперсию, тем самым увеличивается вероятность больших шагов, что резко уменьшает время обучения. Такая связь может быть выражена следующим образом:

$$T(t) = \frac{T_0}{1+t} \quad (8)$$

Эксплуатация системы проводилась для рынка межбанковских кредитов. При этом в качестве параметров выступали ставка рефинансирования ЦБ России и курс доллара. Средняя погрешность прогноза процентных ставок за период с мая по октябрь 2000 года составила: для классических методов 4.8 %, для нейронной сети 3.6%.

Однако, механизм нейронных сетей, реализующий распределённую вычислительную модель, не обладает “логической прозрачностью”, т.е. аналитическое выражение математической модели, используемой при прогнозировании, исследователю недоступно. Если исследование предполагает дальнейшее использование полученных математических моделей, то приходится использовать классические методы.

Решение многокритериальных задач прогнозирования, приводящее к значительному усложнению классических методов аппроксимации и трудности их программирования, значительно проще решается с помощью нейросетевого аппарата.

Эксплуатация системы на рынке межбанковских кредитов показала, что точность прогноза, полученная с применением нейросетевых алгоритмов, выше таковой для классических методов.

Но решение конкретных задач прогнозирования может требовать как нейросетевого, так и классического подходов, имеющих свои “плюсы” и “минусы” (см. выше).

При разработке АСП решающей поставленную задачу прогнозирования изложенными выше методами была разработана система классов С++, позволяющая программисту расширить круг реализованных в системе методов аппроксимации за счёт создания своих классов-наследников или провести моделирование и воспитание нейронной сети необходимой структуры.

АСП реализована на Borland C++ Builder 5.0. Посредством механизма ODBC реализуется доступ к данным MS SQL Server 7.0, возможен ручной ввод данных.

В настоящее время ведётся разработка и воспитание многослойных нейронных сетей от которых ожидаются более высокие показатели прогноза.

Список использованных источников.

1. А.А. Амосов, Ю.А. Дубинский, Н.В.Копченлова. Вычислительные методы для инженеров: Учеб. пособие. – М: Высш. шк., 1994. – 544 с.: ил.
2. А.Н. Горбань, Д.А. Россиев “Нейронные сети на персональном компьютере” – Новосибирск: Наука, Сибирская издательская фирма РАН, 1996 – 276 с.
3. А.Б. Баев. Программно-аналитическая обработка результатов наблюдений. Сборник научных трудов МГУ сервиса – М. 2000 – 61-66 с.
4. Ф. Уоссермен. “Нейрокомпьютерная техника”- М. Мир, 1992 – 126 с.
5. В. Л. Яковлев, Г.Л. Яковлева, Л.А.Лисицкий. “Применение нейросетевых алгоритмов к анализу финансовых рынков” // “Информационные технологии” №9, 48-54 с.

ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ УПРАВЛЕНИЯ ЛОКАЛЬНЫМИ ВЫЧИСЛИТЕЛЬНЫМИ СЕТЯМИ

Пакилев А.Е.

Научный руководитель: к.т.н., доцент Власов А.И.

ИУ4, МГТУ им. Н.Э. Баумана,

INTELLECTUAL METHODS OF CONTROL OF LOCAL AREA NETWORKS

Pakilev A.E.

The scientific chief: Ph.D., senior lecturer Vlasov A.I.

ИУ4, BMSTU

Аннотация: Сегодня даже работа в офисах небольших фирм часто осуществляется с использованием технологий компьютерных сетей, а работа крупных предприятий и корпораций уже не мыслима без возможностей, предоставляемых этими технологиями. Работа посвящена исследованию вопросов интеллектуального управления локальными вычислительными сетями.

Abstract: Today even the activity at offices of small corporations often implements with usage of technologies of computer networks, and activity of major concerns and corporations without capabilities granted by these technologies. The activity is dedicated to research of problems of intellectual control of local area networks.

Введение.

На сегодняшний день существует огромное множество различных способов объединения компьютеров в вычислительные сети и возможностей, предоставляемых ими. Это может быть как тривиальное использование одного принтера несколькими сотрудниками или передача файлов с одного компьютера на другой, так и безопасное получение информации, необходимой для ведения переговоров, сотруднику компании находящемуся за несколько тысяч километров от последней. Такое многообразие может поставить в тупик любого не подготовленного, но желающего использовать передовые технологии, человека. Для решения такой задачи главное – правильно сформулировать требования к создаваемой системе и задачи, которые она должна решать. Так как в простом случае можно использовать уже готовые решения, а в сложном, возможно

многовариантное решение данной задачи, что может даже потребовать использования оптимизационных алгоритмов (например, часто применяют теорию очередей).

1. Иерархическое представление структуры сети.

Любая компьютерная сеть представляет собой набор разнообразных устройств, которые можно разделить на две группы:

- Терминальные / конечные устройства (компьютеры, сетевые принтеры, различные серверы)
- Коммуникационные устройства (концентраторы, мосты, коммутаторы, маршрутизаторы)

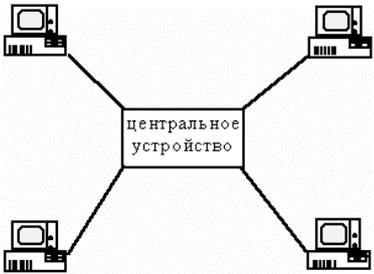
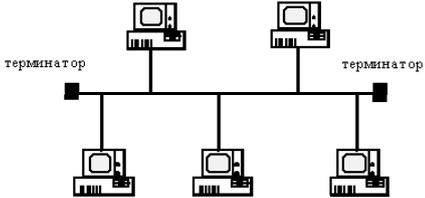
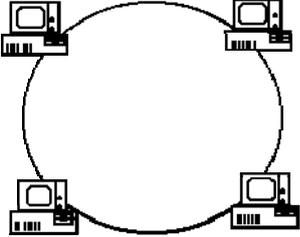
Все эти устройства могут быть объединены между собой различными способами и с использованием различных технологий. Но основными пока все равно являются металлические кабели для небольших расстояний и волоконно-оптические линии связи для создания магистральных каналов с высокой пропускной способностью и при необходимости передавать данные на большие расстояния.

Основные технологии физического уровня, используемые сегодня, рассмотрены ниже в таблице:

Физическая среда	Тип кабеля	Пропускная способность	Ограничения	Особенности	Достоинства	Недостатки
10Base-5	Коаксиальный кабель Ø0.5 дюйма, называемый «Толстый коаксиал».	10 Мбит / с	<p>Максимальная длина сегмента 500 м (без повторителей)</p> <p>Максимально 4 повторителя, но нагруженными могут быть только три из них.</p> <p>На одном сегменте не более 100 трансиверов. Причем расстояние между ними не менее 2.5 м</p> <p>Максимальное число узлов – 297</p>	<p>Образует топологию типа шина.</p> <p>На концах сегмента должны быть согласующие терминаторы сопротивлением 50 Ом.</p> <p>Подключение к сегменту при помощи трансивера (приемопередатчика, устанавливаемого непосредственно на кабеле), соединенного с компьютером кабелем через интерфейс AUI</p>	<ul style="list-style-type: none"> - хорошая защищенность от внешних воздействий - сравнительно большое расстояние между узлами - возможность простого перемещения рабочей станции в пределах кабеля AUI 	<ul style="list-style-type: none"> - высокая стоимость кабеля - сложность его прокладки из-за большой жесткости - потребность в специальном инструменте для заделки кабеля - останов всей сети при повреждении кабеля или плохом соединении - необходимость заранее предусмотреть подводку кабеля ко всем возможным местам установки компьютеров - Сложность поиска неисправности
10Base-2	Коаксиальный кабель Ø0.25 дюйма, называемый «Тонкий коаксиал».	10 Мбит / с	<p>Максимальная длина сегмента 185 м (без повторителей)</p> <p>Максимально 4 повторителя, но нагруженными могут быть только три из них.</p> <p>На одном сегменте не более 30 трансиверов. Причем расстояние между ними не менее 1 м</p> <p>Максимальное число узлов – 87</p>	<p>Образует топологию типа шина.</p> <p>Подключение к сегменту с помощью высокочастотного T-коннектора</p>	<ul style="list-style-type: none"> - Меньшая стоимость - Сравнительная легкость монтажа 	<ul style="list-style-type: none"> - Помехозащищенность ниже - Также ниже механическая прочность - Меньшее возможное количество узлов и суммарное расстояние - Больше количество механических соединений - Сложность поиска неисправности
10Base-T	Кабель на основе неэкранированной пары (Unshielded Twisted Pair, UTP-cat. 3 используются две пары)	10 Мбит / с	<p>Расстояние между конечным узлом и концентратором ≤ 100 м</p> <p>Максимально – 4 концентратора.</p> <p>Узлов ≤ 1024</p>	<p>Образует звездообразную топологию на основе концентратора.</p> <p>Широко распространен (не в России) для телефонии, что позволяет создавать сеть без прокладки дополнительных</p>	<ul style="list-style-type: none"> - Повышение надежности сети, так как отключение одного сегмента не приводит к отключению остальных. - Широкие возможности расширения 	<ul style="list-style-type: none"> - При создании сети требуется большое количество проводов - Необходим специальный инструмент

				кабельных линий		
10Base-F	Многомодовый волоконно-оптический кабель с полосой пропускания 500-800МГц при длине кабеля 1км. Используется два оптоволоконна.	10 Мбит / с	Имеется несколько спецификаций: FOIRL – длина сегмента сети ≤ 1000 м (до 4-х повторителей, максимальная длина сети 2500м) 10Base-FL - длина сегмента сети ≤ 2000 м (до 4-х повторителей, максимальная длина сети 2500м) 10Base-FB - длина сегмента сети ≤ 2000 м (предназначен только для повторителей, между узлами до 5-ти повторителей, максимальная длина сети 2740м)	Образует звездообразную топологию на основе концентратора.	- высокая защита от внешних помех - значительное возможное расстояние между узлами	- высокая стоимость - сложность прокладки, необходим специальный инструмент
100Base-T4	UTP cat. 3, 4, 5 (используются четыре пары)	100 Мбит / с	Максимальная длина сегмента 100м Для 100Base-FX 412м (полудуплекс) 2000м (полный дуплекс) Число повторителей – 1 или 2 в зависимости от их класса.	Образует звездообразную топологию на основе концентратора.	- десятикратное увеличение пропускной способности - 100Base-T4 может использовать уже существующую телефонную кабельную сеть	- увеличение стоимости, сложный монтаж - сокращение диаметра сети - сокращение возможного количества повторителей
100Base-TX	UTP cat. 5 (используются две пары) или STP (Shielded Twisted Pair) Type 1					
100Base-FX	Многомодовый оптоволоконный кабель (используются два волокна)					

В зависимости от целей и задач, предъявляемых к вычислительной сети, будет зависеть сложность ее строения, или, другими словами, *топология* (схема связей). Существует три основных топологии:

Тип	Схематическое изображение	Характеристики	Надежность
Звезда		<ol style="list-style-type: none"> 1. Сложность монтажа – высокая, необходимо большое количество кабеля, процесс монтажа сложен. 2. Стоимость монтажа – высокая, требует наличия дополнительного оборудования и высококвалифицированного монтажа. 3. Максимальная длина сегмента – 100м. 4. Расстояние между узлами – до 100м. 5. Возможности расширения – широкие 	<p>Высокая, неполадки на одной станции не влияют на работу других, однако неполадки в концентраторе выводят из строя всю сеть.</p>
Шина		<ol style="list-style-type: none"> 1. Сложность монтажа – средняя, проста в установке, требует меньше кабеля. 2. Стоимость монтажа – средняя, не требует наличия дополнительного оборудования. 3. Максимальная длина сегмента – 185м 4. Расстояние между узлами – ≥ 0.5м. 5. Возможности расширения – средние 	<p>Средняя, неполадки на отдельной станции трудно изолировать, обрыв магистрального кабеля выводит всю сеть из строя.</p>
Кольцо		<ol style="list-style-type: none"> 1. Сложность монтажа – средняя, 2. Стоимость монтажа – низкая, не требует наличия дополнительного оборудования 3. Максимальная длина сегмента – 185м 4. Расстояние между узлами – 2.5м. 5. Возможности расширения – низкие. 	<p>Низкая, неполадки на одной станции выводят из строя всю сеть.</p>

Любую более сложную сеть можно представить в виде соединенных между собой сегментов, имеющих одну из приведенных топологий.

Рассмотренные выше технологии объединения конечных устройств в сеть и виды топологий сегментов сети вносят определенные ограничения, которые не позволят при необходимости строить большие сети, чем это предусматривает используемая технология.

Многие возникающие проблемы можно решить используя коммутаторы. Эти устройства позволяют сделать следующее:

- 1) Разгрузить линии передачи данных.
- 2) Снять ограничения, накладываемые используемой технологией на число узлов и физическую длину линий связи (Ethernet – 1024 узла, хотя уже при 40-50 компьютерах в сети наблюдается резкое снижение производительности из-за случайного метода доступа к среде передачи данных).
- 3) Передавать данные, требующие приоритизации (передача голоса, видеоданных).
- 4) Использовать гетерогенные сети, при которых требуется трансляция из одного протокола канального уровня в другой (например, Ethernet \leftrightarrow FDDI)
- 5) Предоставляется возможность разделения одной сети на несколько, так называемых, виртуальных сетей (VLAN), что изолирует пользователей разных подсетей. Таким образом, можно, например, изолировать отдел бухгалтерии от отдела разработчиков.
- 6) Предоставляется возможность фильтрации трафика на основе MAC-адресов (например, необходимо запретить печать документов конкретному пользователю на удаленном сервере печати).

Но при построении сети масштабов предприятия коммутаторы также имеют свои ограничения, которые можно преодолеть, используя маршрутизаторы.

Использование последних, позволит следующее:

- 1) Иметь в наличии избыточные связи в топологии сети (петель), т.е. несколько путей достижения конкретного узла, что существенно повысит надежность сети, позволяя при этом балансировать нагрузку на каналы передачи данных.
- 2) Коммутаторы пропускают ширококвещательные сообщения, что при наличии большого числа машин в сети может привести к «широковещательным штормам», создающим ощутимую загрузку линий передачи данных.
- 3) Переход от системы адресации по MAC-адресам к адресации по IP-адресам позволяет более гибко структурировать сеть, облегчая ее настройку и управление.
- 4) Далеко не все коммутаторы поддерживают приоритизацию трафика и трансляцию протоколов канального уровня, а если и поддерживают, то не умеют фрагментировать кадры, что в ряде случаев не приемлемо.

В общем случае большая корпоративная сеть может состоять из десятков маршрутизаторов и еще большего числа коммутаторов, а также другого коммуникационного оборудования, такого как УАТС, мультиплексоры, аппаратура передачи данных на большие расстояния и т. д. Причем, территориально, такая сеть может быть распределенной. Такую сеть, обычно, представляют в виде иерархической структуры, где самым верхним уровнем иерархии является объединение нескольких сетей с помощью маршрутизаторов, подсети объединены коммутаторами, а подсети, в свою очередь, объединяют отдельные рабочие станции, сервера и различное вспомогательное оборудование. Структурированные подобным образом компьютерные сети еще называют *структурированными кабельными системами*.

2. Управление сетями, функции и задачи.

Для работы любой компьютерной сети необходимо наличие аппаратной и программной части. Если в первую могут входить компьютеры, оборудованные сетевыми адаптерами, концентраторы, коммутаторы и т. д., то ко второй части относят набор программного обеспечения, позволяющий управлять работой сетевого оборудования. В зависимости от сложности системы программная часть может быть как набором разрозненных программ, так и единой, интегрирующей в себе множество функций и обеспечивающей комплексное управление распределенной сетью, системой управления.

Независимо от объекта управления и уровня интеграции, желательно, чтобы система управления выполняла ряд функций, которые определены международными стандартами, обобщающими опыт применения систем управления в различных областях. Существуют рекомендации ITU-T X.700 и близкий к ним стандарт ISO 7498-4, которые делят задачи системы управления на пять функциональных групп:

1) управление оборудованием сети.

– настройка параметров работы сетевых устройств, определение физической и логической топологий, определение приоритетов на передачу тех или иных пакетов данных.

2) обработка ошибок работы сети.

– выявление, определение, сигнализация и, если это возможно, устранение последствий сбоев, отказов или, просто, неправильной работы оборудования. В эту группу задач также входит фильтрация, маршрутизация и комплексный анализ сообщений об ошибках.

3) анализ производительности и надежности.

– накопление некоторых статистических данных работы сети, например, время реакции системы, пропускная способность каналов, интенсивность трафика и др. Данные показатели необходимы для оперативного управления сетью и оптимального планирования загрузки и развития сети.

4) управление безопасностью и полномочиями пользователей.

– контроль использования ресурсов сети, сохранение целостности данных, поддержка процедур аутентификации пользователей и проверка их полномочий при доступе к ресурсам.

Часто эти задачи решаются средствами отдельных программных и программно-аппаратных комплексов

5) учет работы сети.

– регистрация времени работы сети, причем, она может осуществляться дифференцировано для разных сетевых служб или типов передаваемой информации. Эта группа задач тесно связана с таким понятием как биллинг, т. е. с оплатой за использование тех или иных ресурсов сети.

Наряду с функциональными группами задач управления сетью выделяют также функциональные группы задач управления системами:

1) Учет используемых аппаратных и программных средств

– сбор таких данных, как используемое оборудование, установленные операционные системы и прикладные программы. Такая информация позволяет получить полную картину существующих ресурсов сети. Например, системный администратор может быстро выяснить, на каких компьютерах необходимо обновить драйверы принтеров.

2) Распределение и установка программного обеспечения

– рассылка пакетов необходимого программного обеспечения на группы машин, установка и конфигурация сетевых приложений, запускаемых с файловых серверов.

3) Удаленный анализ производительности и возникающих проблем.

– сбор таких сведений как загрузка процессора, коэффициент использования физической памяти, интенсивность транзакций и т. д.

Приведенные задачи управления системами схожи с задачами управления сетью, только управление происходит над другими объектами. Вследствие этого, наблюдается интеграция программных продуктов управления сетью и управления системами. Примерами таких интегрированных программных продуктов являются CA Unicenter TNG и TME-10 IBM/Tivoli.

3. Многоуровневое представление задач управления.

Как уже говорилось выше, большая сеть, например, сеть корпорации, строится иерархически, фактически, отражая иерархию самого предприятия. Нижний уровень такой структуры будет рассматривать элементы сети, такие как отдельные компьютеры, коммуникационные устройства, каналы передачи данных. Следующий уровень – группы компьютеров, принадлежащих отделам предприятия. Далее, уровень отделения предприятия. И, наконец, уровень самого предприятия. На каждом уровне иерархии можно выделить те же функциональные группы задач, что и в целом, для всей сети. Только для каждого уровня они будут иметь свою специфику, т. е. чем выше уровень иерархии, тем более общий характер будет иметь эти задачи.

Для построения иерархических вычислительных сетей был выработан стандарт Telecommunication Management Network (TMN), разработанный ITU-T, ISO, ANSI и ETSI. Изначально этот стандарт предназначался только для телекоммуникационных сетей, но ориентация на использование общих принципов делает его полезным для построения любой крупной интегрированной системы управления сетями. Стандарты TMN состоят из большого числа рекомендаций ITU-T (и стандартов других организаций), но основные принципы данной модели описаны в рекомендации M.3010.

Согласно модели TMN выделяют следующие уровни:

1) Нижний уровень – уровень элементов сети.

– состоит из отдельных устройств сети (каналы, оконечное оборудование, коммутаторы, мультиплексоры и т. д.) Как правило, большинство устройств имеют встроенные средства дистанционного управления. Некоторые технологии, такие как FDDI, ISDN, frame relay, SDH, требуют реализации хотя бы минимальных средств управления. Таким образом, если устройство даже не имеет собственного встроенного блока управления, базовые операции по контролю его работы все равно можно осуществлять.

2) Уровень управления элементами сети.

– представляет элементарные системы управления. Элементарные системы управления автономно управляют отдельными устройствами. Данный уровень изолирует верхние уровни управления от деталей и особенностей управления конкретным оборудованием, реализуя единую модель управления. Обычно элементарные системы управления разрабатываются и поставляются самим производителем оборудования. Примером таких систем может служить CiscoView от Cisco Systems, Optivity от Nortel Networks, RADView от RAD Data Communication и т. д.

3) Уровень управления сетью.

– этот уровень координирует работу элементарных систем управления, позволяя контролировать координацию составных каналов, согласовывать работу транспортных подсетей разных технологий. Данный уровень обеспечивает централизованное управление всей сетью, обеспечивая работу сети как единого целого.

4) Уровень управления услугами.

– занимается управлением и контролем транспортных и информационных услуг, которые представляются конечным пользователям. В задачи этого уровня входит настройка сети на обеспечение работы конкретной услуги (необходимая средняя

пропускная способность, создание виртуальных каналов, указание максимальных величин задержек и т. д.). После начала работы некоторого сервиса данный уровень занимается контролем качества, надежности и безошибочности работы.

5) Уровень бизнес-управления.

– занимается вопросами долговременного планирования сети с учетом финансовых аспектов деятельности организации, владеющей сетью. Подсчитываются эксплуатационные расходы и доходы. Определяется экономическая эффективность сети и необходимость дальнейшего ее расширения.

4. Методы управления сетью.

Первая функциональная группа задач управления сетью рассматривает задачи управления оборудованием. Существует несколько способов управления оборудованием сети:

- 1) «Вручную», используя для настройки отдельные программы для каждого устройства.
- 2) С использованием терминала, непосредственно подключенного к устройству.
- 3) С использованием удаленного терминала.
- 4) С использованием протоколов управления.

Настройка «вручную».

При объединении в сеть нескольких компьютеров и, быть может, выделенного сервера достаточно средствами операционной системы каждого устройства указать необходимые параметры работы. Например, при построении такой одноранговой сети между компьютерами, работающими под управлением Windows, достаточно указать IP-адреса каждой машины, маску подсети и, при необходимости, еще настроить сетевые адаптеры. Файловый сервер или сервер печати в таком случае настраивается не намного сложнее. Грамотный специалист в течении пары часов настроит такую сеть.

Более сложные в настройке устройства, например, коммутатор, часто поставляются вместе с фирменной программой настройки, обычно, имеет удобный интерфейс и хорошую справочную систему.

Настройка с использованием терминала, подключенного к устройству.

Другой способ – управление с помощью некоторого терминального устройства, обычно, настольного или переносного компьютера, подключенного через стандартный последовательный интерфейс RS-232. Хотя могут использоваться и другие, «экзотические», терминальные устройства.

Настройка с использованием удаленного терминала.

Основной недостаток двух предыдущих методов очевиден. Если устройств много или они просто находятся далеко друг от друга (например, в соседнем здании), то придется потратить много времени для того, чтобы обойти все устройства. На сегодняшний день большинство устройств поддерживают возможность удаленного управления с помощью встроенных модулей управления. Существуют стандартные программы (например, программа telnet, поставляемая вместе с Windows), которые могут дистанционно через сеть общего пользования или специальную выделенную сеть управления подключиться к управляемому ресурсу. При этом, для оператора, использующего данный метод управления, работа с устройством, практически, не отличается от предыдущего метода с той лишь разницей, что, не вставая с места, он может управлять несколькими устройствами.

Настройка с использованием протоколов управления.

Все описанные выше методы предоставляют возможность работы индивидуально с каждым сетевым устройством, причем какие-либо средства автоматизации очень ограничены. Все это очень затруднило бы работу в крупных сетях, имеющих сотни различных коммуникационных устройств. Дальнейшее развитие коммуникационного оборудования привело к разработке устройств со встроенными средствами дистанционного управления и протоколов взаимодействия систем управления и этих устройств. Это позволило разрабатывать системы автоматического и автоматизированного управления, интегрирующие в себе многие функции управления, а, главное, предоставляющие возможность управлять сетью комплексно и централизованно, т. е. управлять не отдельными устройствами, а целиком всей сетью.

Simple Network Management Protocol (SNMP)

SNMP был разработан Университетом Тенесси (США) в 1988 году и распространяется свободно. Он широко поддерживается производителями оборудования Ethernet, расширяется его использования в сетях Token Ring и в телекоммуникациях. В значительной степени популярность SNMP задержала принятие CMIP, варианта управляющего протокола по версии OSI.

Коротко говоря, SNMP - это запрос-ответный протокол, используемый для получения от сетевых устройств информации об их статусе, производительности и их характеристиках.

Его главные достоинства - простота, доступность, независимость от производителей. Набор команд маленький, реализовать его просто. Поскольку для него требуются минимальные ресурсы, им можно пользоваться в сетях персональных компьютеров, CMIP для этой области применения не подходит.

Для использования этого протокола устройство должно содержать в себе так называемый "агент", обрабатывающий элемент, который обеспечивает менеджерам, размещенным на управляющих станциях сети, доступ на управление и мониторинг устройства.

Основные операции по управлению вынесены в управляющую станцию. Устройства же заняты, прежде всего, своим непосредственным делом. Более того, для устройства, которое не в состоянии обслужить даже простейшие действия, либо не воспринимает SNMP, может быть установлено промежуточное устройство - отвечающее за работу агента-посредника.

SNMP непосредственно опрашивает управляемые устройства. Изредка, когда происходит значительное событие, агенты инициируют управляющие действия. Такими действиями в SNMP являются: инициализация агента, рестарт агента, разрыв связи, рестарт связи, неудачная аутентификация, потеря маршрутизатора. Если подобное событие произошло, агент выставляет прерывание. Производители могут добавить собственные события, требующие обработки.

Информация, существенная для управления сетью, накапливается в MIB (Management Information Base), являющейся, по существу, базой данных объектов. Спецификация MIB1 определяет выборку и чтение информации, MIB2 - изменение и установку значений объектов. MIB является спецификацией OSI. RMON (remote network monitoring MIB - удаленный мониторинг MIB) - последнее дополнение к SNMP, предназначенное для удаленного использования протокола.

SNMP использует транспортный протокол UDP без соединений, что гарантирует малое время отклика и удобно при неполадках сети. Протоколы, ориентированные на соединения, типа TCP, слишком загружают устройства и поэтому не подходят для эффективного управления сетью, кроме того, они не помощники, если сеть "осыпалась". А управление сетью наиболее актуально именно в таких ситуациях. Впрочем, SNMP

работает и под TCP, OSI TP0 и TP4, OSI CNLS, и также в рамках NetWare, начиная с версии 3.11.

SNMP обслуживает передачу данных между агентами и станцией, управляющей сетью. Администраторы "видят" сеть через управляющую станцию. Такая станция - это мощный компьютер, работающий, как правило, под управлением UNIX, реже - DOS. Обычно обеспечивается графический интерфейс пользователя, основанный на X Window или MS Windows, иногда - ASCII-интерфейс.

Управляющая система, как минимум, обеспечивает автоматическое конфигурирование карты сети, отображение в реальном времени состояний сети, мониторинг производительности, событий, тестирование соединений дублированных IP-адресов, генерацию предупреждений. Может предоставляться реляционная база данных и средства генерации отчетов.

При условии, что SNMP является основой большинства систем управления, он имеет ряд существенных недостатков:

- Каждый производитель коммуникационного оборудования добавляет в свои устройства какие-нибудь индивидуальные особенности, что сильно затрудняет сделать универсальный протокол. Производители объединяются вокруг попыток создания более развернутого стандарта.
- Большинство специалистов сходятся на том, что используемая схема централизованной регистрации предупреждений и сбора статистики слишком громоздка для корпоративных решений. Простой мониторинг тысячи клиентских станций может переполнить даже весьма мощный управляющий компьютер. Решение, к которому склоняются производители и аналитики, состоит в распределенной среде управления, основанной на системе доменов, отвечающих за определенные группы клиентов и серверов или за конкретное приложение.
- Отсутствуют средства взаимной аутентификации агентов и менеджеров. Единственное средство, которое можно было бы отнести к средствам аутентификации, является использование в сообщениях так называемой строки сообщества - "community string". Эта запись предназначена для структуризации системы агентов и менеджеров на отдельные группы, что плохо подходит для аутентификации. Версия SNMPv.2 должна была ликвидировать этот недостаток, но из-за разногласий между разработчиками стандарта новые средства аутентификации хотя и появились в этой версии, но как необязательные.

Но пока выработка стандарта для управления системами клиент/сервер далека от заключительной стадии. Правда, уже существуют предложения по стандартизации для SNMP-платформ, позволяющей собирать определенного рода статистику о приложениях, работающих в таких системах (например, о повреждениях файлов в реляционных СУБД).

SNMP2

В 1993 году в качестве замены SNMP предложен протокол SNMP2. SNMP2 улучшает характеристики производительности и безопасности SNMP, определяет дополнительные типы данных, обеспечивает операции типа "менеджер менеджеров" и определяет, как протокол отображается на транспортные уровни.

SNMP2 поддерживает домены управления и более эффективный сбор данных от узлов сети. Однако производители не проявляют особого энтузиазма относительно его поддержки. Объясняясь, говорят, что заказчики этого пока не просят. Однако без поддержки стандартов нельзя управлять разнородными конфигурациями.

Впрочем, может оказаться, что и SNMP2 не решит проблем управления. В нем сохранен механизм регистрации, требующий, чтобы выделенная станция периодически опрашивала управляющие устройства.

Некоторые производители (Open Vision, IBM, HP, Legent) обращаются к спецификациям CORBA, описывающим распределенную объектную технологию управления, когда агенты, расположенные на управляемой системе, самостоятельно и по собственной логике обрабатывают важные события. Однако единой реализации этих спецификаций пока нет.

Common Management Information Protocol (CMIP)

В отличие от SNMP, CMIP представляет собой весьма детальный стандарт, создатели которого стремились охватить как можно больше проблем.

Но, равно как и SNMP, CMIP отвечает за общение менеджера и агента, обеспечивая при этом куда более мощные средства управления. Сравним хотя бы наборы ключевых слов двух протоколов.

SNMP: get, getnext, set, trap.

CMIP: get, set, action, event-report, create, delete.

Тем не менее, дебатов типа "CMIP или SNMP" практически нет - выбор ясен. SNMP - это общепринятый протокол сегодняшнего дня, а CMIP некоторыми признается за долговременную перспективу.

5. Модели управления сетью.

5.1. Схема менеджер-агент.

Элементарная схема управления коммуникационным устройством представляет из себя взаимодействие между агентом и менеджером. Агент, в данном случае, является посредником между управляемым устройством и программой-менеджером. В общем случае менеджер занимается управлением несколькими устройствами и для получения некоторого однообразия в управлении устройствами строится некоторая модель, описывающая управляемый ресурс только теми



характеристиками, которые необходимы для контроля и управления. Соответственно, агент посылает менеджеру уже обработанную информацию в соответствии с построенной моделью, освобождая программу-менеджер от особенностей конкретного устройства, а также особенностей взаимодействия агента с управляемым ресурсом.

Модель устройства, которую используют агент и менеджер, представляет собой базу данных, содержащую информационные и управляемые характеристики устройства. Другое название – база данных управляющей информации (Management Information Base, MIB). Существует разница в том, как MIB используется агентом и менеджером. Агент заполняет эту базу данных текущими значениями характеристик ресурса, а менеджер на основе базы данных получает информацию о том, что он может узнать об устройстве и чем может управлять.

Менеджер взаимодействует с агентами по стандартному протоколу. Этот протокол позволяет менеджеру запрашивать информацию у агента и получать ответ. Как уже говорилось выше, взаимодействие менеджера и агента может происходить как по сети общего назначения, так и по специально выделенной сети управления. Очевидно, что управление по сети общего пользования более экономично, так как не требует построения отдельной, дополнительной сети, но, с другой стороны, существует опасность, что ошибочными действиями можно прервать работу канала и таким образом

потерять контроль над устройством. Также немаловажным является тот факт, что передача управляющей информации по сети общего пользования может быть перехвачена злоумышленником, что позволит осуществить даже изменение настройки ресурсов сети.

Агенты могут встраиваться в управляемое оборудование, а могут работать на отдельном компьютере, связанном с управляемым оборудованием по какому-либо стандартному интерфейсу. Менеджер, обычно, работает на отдельном компьютере, выполняющем также и функции консоли оператора управления.

Агенты также различаются “уровнем интеллекта”, т. е. они могут и просто подсчитывать количество проходящих пакетов, а могут и самостоятельно выполнять некоторые действия в аварийных ситуациях или строить временные зависимости на основе собираемых данных.

Такая модель менеджер-агент лежит в основе таких популярных стандартов управления, как стандарты Internet на основе SNMP протокола, и стандартов ISO\OSI на основе протокола CMIP.

5.2. Структуры распределенных систем управления.

В крупной корпоративной сети полностью централизованная система управления, построенная на базе единственного менеджера, вряд ли будет работать хорошо по нескольким причинам. Во-первых, такой вариант не обеспечивает необходимой масштабируемости по производительности, так как единственный менеджер вынужден будет обрабатывать весь поток сообщений от всех агентов, что при нескольких тысячах управляемых объектов потребует очень высокопроизводительной платформы для работы менеджера и перегрузит служебной управляющей информацией каналы передачи данных в той сети, где будет расположен менеджер. Во-вторых, такое решение не обеспечит необходимого уровня надежности, так как при отказе единственного менеджера будет потеряно управление сетью. В-третьих, в большой распределенной сети целесообразно располагать в каждом географическом пункте отдельным оператором или администратором, управляющим своей частью сети, а это удобнее реализовать с помощью отдельных менеджеров для каждого оператора.

Схема «менеджер - агент» позволяет строить достаточно сложные в структурном отношении распределенные системы управления.

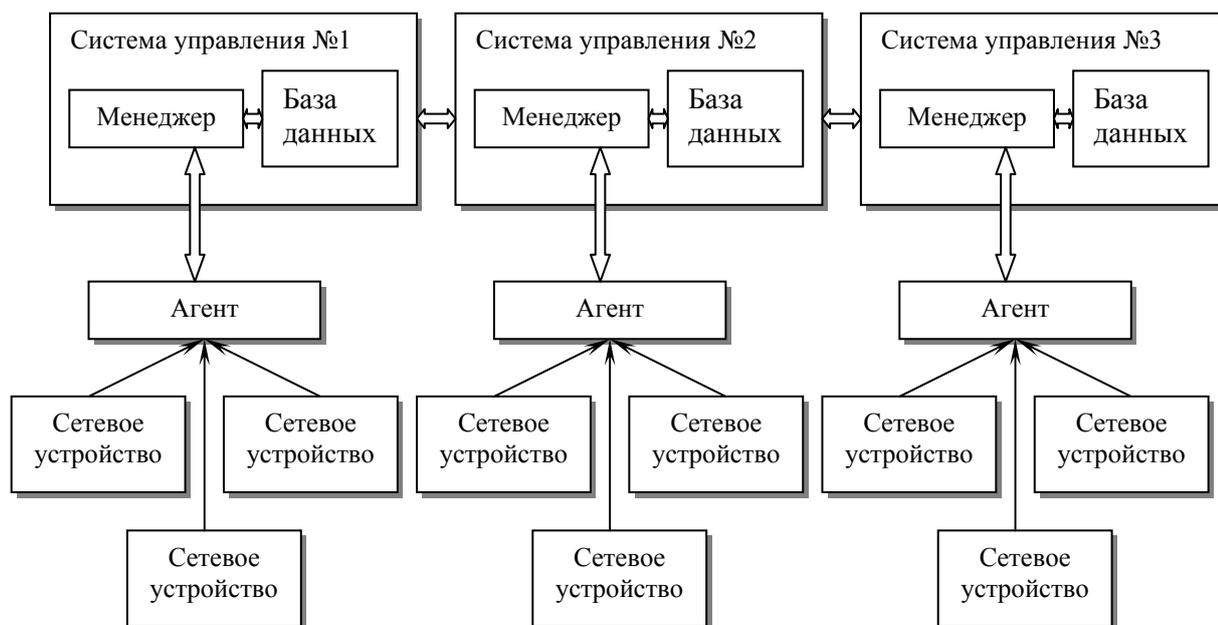
Обычно распределенная система управления включает большое количество связок менеджер - агент, которые дополняются рабочими станциями операторов сети, с помощью которых они получают доступ к менеджерам

Каждый агент собирает данные и управляет определенным элементом сети. Менеджеры, иногда также называемые серверами системы управления, собирают данные от своих агентов, обобщают их и хранят в базе данных. Операторы, работающие за рабочими станциями, могут соединиться с любым из менеджеров и с помощью графического интерфейса просмотреть данные об управляемой сети, а также выдать менеджеру некоторые директивы по управлению сетью или ее элементами.

Наличие нескольких менеджеров позволяет распределить между ними нагрузку по обработке данных управления, обеспечивая масштабируемость системы. Чаще всего используются два подхода к их соединению:

- одноранговый
- иерархический

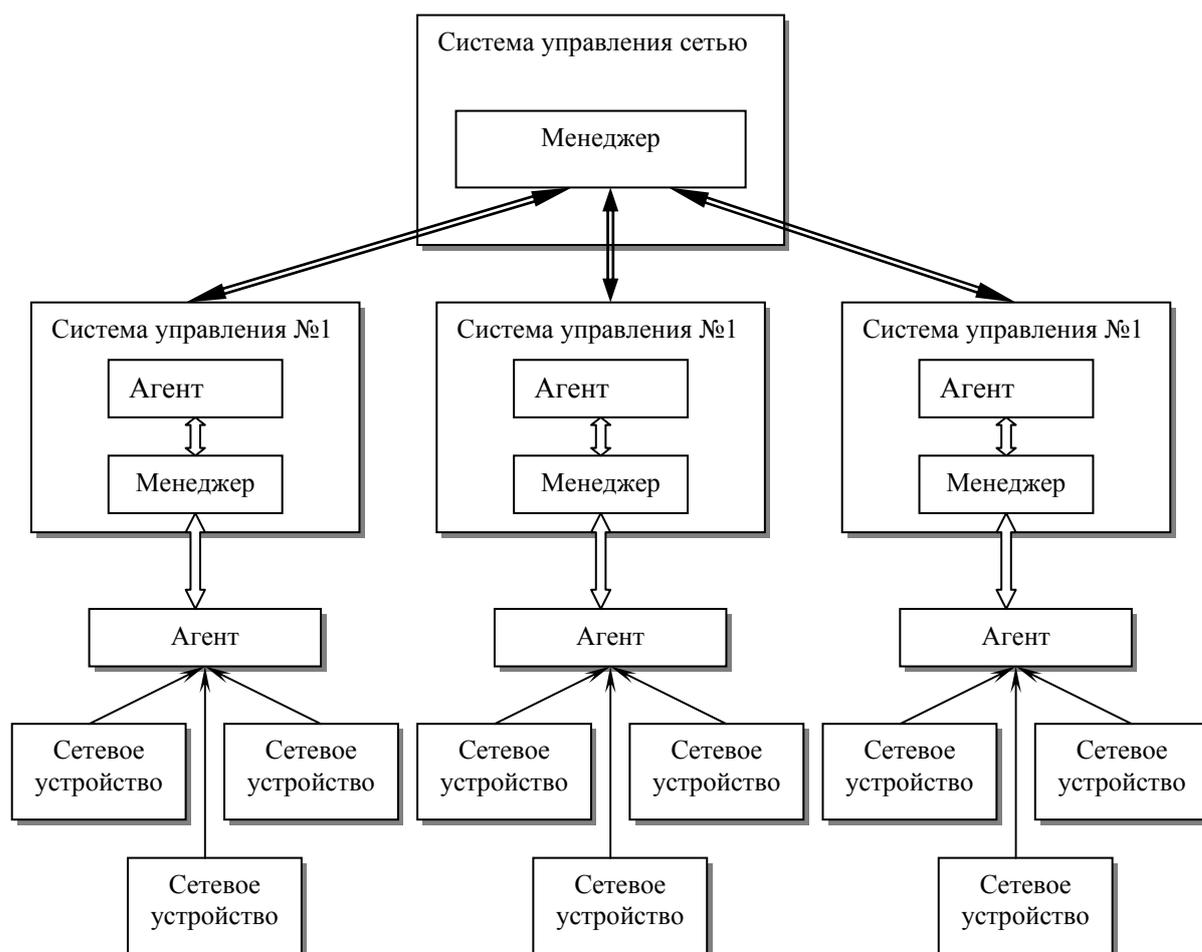
В случае одноранговых связей каждый менеджер управляет своей частью сети на основе информации, получаемой от нижележащих агентов. Центральный менеджер отсутствует. Координация работы менеджеров достигается за счет обмена информацией между базами данных каждого менеджера.



Одноранговое построение системы управления сегодня считается неэффективным и устаревшим. Обычно оно вызвано тем обстоятельством, что элементарные системы управления построены как монолитные системы, которые первоначально не были ориентированы на модульность системы (например, многие системы управления, разработанные производителями оборудования, не поддерживают стандартные интерфейсы для взаимодействия с другими системами управления). Затем эти менеджеры нижнего уровня стали объединяться для создания интегрированной системы управления сетью, но связи между ними оказалось возможным создавать только на уровне обмена между базами данных, что достаточно медленно. Кроме того, в базах данных таких менеджеров накапливается слишком детальная информация об управляемых элементах сети (так как первоначально эти менеджеры разрабатывались как менеджеры нижнего уровня), вследствие чего такая информация малопригодна для координации работы всей сети в целом. Такой подход к построению системы управления называется подходом «снизу вверх».

Гораздо более гибким является иерархическое построение связей между менеджерами. Каждый менеджер нижнего уровня выполняет также функции агента для менеджера верхнего уровня. Такой агент работает уже с гораздо более укрупненной моделью (MIB) своей части сети, в которой собирается именно та информация, которая нужна менеджеру верхнего уровня для управления сетью в целом. Обычно для разработки моделей сети на разных уровнях проектирование начинают с верхнего уровня, на котором определяется состав информации, требуемой от менеджеров-агентов более низкого уровня, поэтому такой подход назван подходом «сверху вниз». Он сокращает объемы информации, циркулирующей между уровнями системы управления, и приводит к гораздо более эффективной системе управления.

Модель TMN в наибольшей степени соответствует иерархической архитектуре связей между менеджерами, хотя известны реализации принципов TMN и в одноуровневых архитектурах.



5.3. Платформенный подход.

Вышеописанные схемы построения имеют один существенный недостаток – они не являются универсальными, что очень существенно при наличии в сети устройств разных производителей.

Поэтому наиболее популярным на сегодняшний день является использование платформенного подхода. В таком случае индивидуальные программы управления разрабатываются не «с нуля», а используют службы и примитивы, предоставляемые специально разработанным для этих целей программным продуктом — платформой. Примерами платформ для систем управления являются такие известные продукты, как *HP Open View*, *SunNet Manager* и *Sun Soltice*, *Cabletron Spectrum*, *IMB/Tivoli TMN10*.

Эти платформы создают общую операционную среду для приложений системы управления точно так же, как универсальные операционные системы, такие как Unix или Windows NT, создают операционную среду для приложений любого типа, таких как MS Word, Oracle и т. п. Платформа обычно включает поддержку протоколов взаимодействия менеджера с агентами — SNMP и реже CMIP, набор базовых средств для построения менеджеров и агентов, а также средства графического интерфейса для создания консоли управления. В набор базовых средств обычно входят функции, необходимые для построения карты сети, средства фильтрации сообщений от агентов, средства ведения базы данных. Набор интерфейсных функций платформы образует интерфейс прикладного программирования (API) системы управления. Пользуясь этим API, разработчики из третьих фирм создают законченные системы управления, которые могут управлять специфическим оборудованием в соответствии с пятью основными группами функций.

Обычно платформа управления поставляется с каким-либо универсальным менеджером, который может выполнять некоторые базовые функции управления без программирования. Чаще всего к этим функциям относятся функции построения карты сети (группа Configuration Management), а также функции отображения состояния управляемых устройств и функции фильтрации сообщений об ошибках (группа Fault Management). Например, одна из наиболее популярных платформ HP Open View поставляется с менеджером Network Node Manager, который выполняет перечисленные функции.

Чем больше функций выполняет платформа, тем лучше. В том числе и таких, которые нужны для разработки любых аспектов работы приложений, прямо не связанных со спецификой управления. В конце концов, приложения системы управления — это прежде всего приложения, а потом уже приложения системы управления. Поэтому полезны любые средства, предоставляемые платформой, которые ускоряют разработку приложений вообще и распределенных приложений в частности.

6. Исследование систем управления сетью.

IBM NetView

IBM первым взялась за интегрированное управление разнородными сетями, но по ряду оценок в наименьшей степени ориентирована на работу с сетями других производителей. IBM NetView - классический инструмент для управления сетями предприятия, организованными в стиле "центр обработки данных". В 1986 году IBM выработала Открытую Архитектуру Управления Сетью (Open NetWork Management Architecture), которая лучше подходит к централизованным мэйнфрейм-ориентированным (чем клиент/сервер) архитектурам и которая определила три компонента: фокусные точки, входные точки и сервисные точки. Фокусные точки – это менеджеры, устанавливаемые на мэйнфреймах. Входные точки агенты, встроенные в управляемые ресурсы. Входные точки, пользуясь NMVT-протоколом семейства SNA, передают данные фокусной точке. Сервисная точка обеспечивает шлюз между NMVT и протоколами управления сетями, не принадлежащими к SNA. Таким образом, в частности, поддерживается общение с SNMP- и CMIP-устройствами.

Более поздняя разработка IBM NetView/6000, учитывающая особенности распределенной среды UNIX-станций, лишена многих ограничений, связанных с иерархической архитектурой своей предшественницы.

AT&T Accumaster Integrator

AT&T бросила вызов IBM и взялась за производство альтернативной системы управления сетью, также выбрав классическую иерархическую модель. Поскольку AT&T не испытывала груза зависимости от собственного оборудования, в предложенной в 1987 году Унифицированной Архитектуре Управления Сетью (Unified Network Management Architecture) были заложены возможности управления устройством от разных поставщиков. За основу был взят интерфейс менеджер/элемент модели OSI. Однако спецификации OSI были неполны, а их развитие отставало от практических потребностей, AT&T действовала на свой страх и риск. В дальнейшем многие из предложений AT&T стали общепринятыми.

AT&T Accumaster Integrator - первая интегрированная управляющая система, основанная на предложениях OSI. В ней используется цветная графика, интегрированная реляционная СУБД, даже экспертная система. К сожалению, стандарт управления сетью по версии OSI до сих пор не завершен, а производители ориентируются на SNMP. В результате, Accumaster в настоящий момент используется, в основном, только с оборудованием AT&T.

Еще один недостаток Accumaster - централизованный подход: в его иерархической архитектуре только один главный Integrator.

DECmcc

DEC был первым производителем, описавшим распределенную архитектуру управления сетью – она называлась ЕМА (Enterprise Management Architecture) - и ее реализацию DECmcc. Эта архитектура описывается в терминах гибких взаимосвязей между менеджерами и агентами, допуская взаимодействие менеджеров между собой и управление различных менеджеров пересекающимися наборами агентов. DECmcc использует распределенный сервис имен и каталогов.

Кроме общепринятой базы данных для данных, связанных с управлением сетью, предложенная DEC платформа поддерживает функциональные модули, модули представления и доступа. Ядро менеджера - объектно-ориентированный прикладной программный интерфейс, основанный на RPC (Remote Procedure Call - Удаленный Вызов Процедуры). Модули доступа предоставляют интерфейсы к системам управления других производителей, поддерживая многочисленные протоколы, фирменные и стандартные. Доступ к удаленным устройствам строится в объектном стиле. Функциональные модули реализуют специфические приложения, типа конфигурирования и обработки ошибок. Модули представления отвечают за пользовательский интерфейс - он может быть графическим и простым символьным.

DECmcc работает в рамках VMS и Ultrix, а теперь и в DEC OSF/1.

HP OpenView

Компания Hewlett-Packard вступила в круг поставщиков систем управления сетями на год позже AT&T. Для того чтобы конечным пользователям не нужно было заботиться о разнообразии различных новых протоколов, HP опубликовала семейство прикладных программных интерфейсов, предоставив другим производителям возможность надстраивать их продукты над базовыми сервисами OpenView. Как это ни удивительно, первоначальная версия OpenView работала под DOS и имела довольно ограниченные возможности. Современная же UNIX-версия весьма развита и работает на разнообразных платформах, в том числе, конечно, на оборудовании HP и рабочих станциях Sun.

OpenView состоит из четырех основных компонентов:

- сервисы представления, включающие прикладные интерфейсы как для Windows, так и для OSF Motif;
- средства управления данными - база данных и управление событиями;
- коммуникационные протоколы, реализующие SNMP и, в определенной мере, CMIP;
- надстроенные над тремя первыми компонентами прикладные управляющие программы, представляющие ряд специфических сервисов.

Распределенный прикладной программный интерфейс достаточно хорошо скрывает детали сетевого оборудования. На основе OpenView свои управляющие приложения реализуют не только традиционные поставщики компьютерного и сетевого оборудования, но и те, кто занят телекоммуникациями.

Масштабируемость и гибкость OpenView доказывают, что архитектура платформы является адекватным решением проблем распределенного предприятия.

SunNet Manager

В число наиболее популярных платформ управления сетью, ориентированных на SNMP, в последнее время вошел SunNet Manager. В 1992 году больше всего SNMP-приложений, разработанных независимыми производителями, предназначалось для SunNet Manager и OpenView. Sun сообщил о 120 подобных разработках, HP - о 110.

Отличительная особенность SunNet Manager - ориентация на потенциально очень большое число рабочих станций в управляемой сети, измеряемое, быть может, десятками тысяч. Управляющая консоль взаимодействует с SNMP-клиентами и клиентами-"посредниками" (прошу agents). Каждый такой посредник может взаимодействовать со многими узлами. У него есть и другая функция - общение с агентами, "не понимающими" SNMP-протокол.

Компании, которые производят коммуникационное оборудование, разрабатывают дополнительные менеджеры для популярных платформ, которые выполняют функции управления оборудованием данного производителя более полно. Примерами таких менеджеров могут служить менеджеры системы Optivity компании Bay Networks и менеджеры системы Trancsend компании 3Com, которые могут работать в среде платформ HP Open View и SunNet Manager.

CiscoWorks2000

CiscoWorks2000 – это семейство программных продуктов, базирующихся на стандартах Интернет и предназначенных для управления сетевыми устройствами Cisco.

CiscoWorks2000 (CW2000) состоит из ядра - Resource Manager Essentials (RME) + Internet Performance Monitor (IPM) и дополнительных приложений:

- CWSI Campus,
- Access Control List (ACL) Manager,
- User Registration Tool (URT).

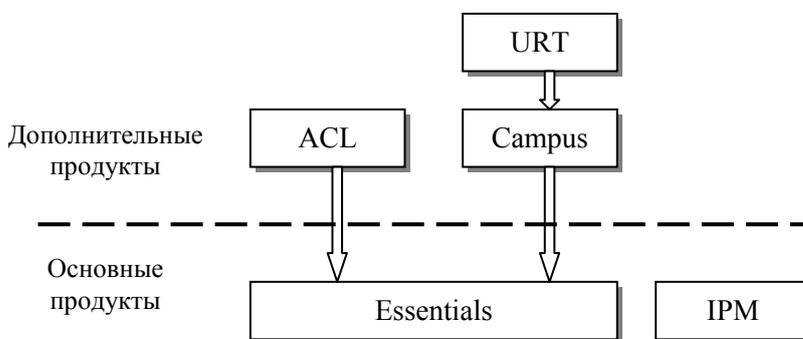
Resource Manager Essentials представляет собой программное средство, предоставляющее WEB интерфейс для управления устройствами Cisco, интеграции сетевой информации и обеспечения возможности для управления устройствами других производителей.

Для WAN каналов и сетей, CW2000 предлагает *Internet Performance Monitor*. IPM применяется для быстрого поиска неисправностей и мониторинга WAN сетей.

CWSI Campus добавляет набор приложений разработанных для управления (прежде всего) коммутаторами локальных сетей Cisco Catalyst и ATM коммутаторами LightStream, а так же другими сетевыми устройствами Cisco. CWSI Campus предоставляет возможности для удаленной настройки, изображения общего вида сети, управления потоками и управления производительностью, что значительно облегчает управление коммутируемой (свитчеванной) сетью.

Access Control List (ACL) Manager обеспечивает инструментарий для конфигурирования, редактирования и оптимизации ACL (фильтров) на маршрутизаторах Cisco.

User Registration Tool (URT) расширяет возможности по наблюдению за пользовательскими компьютерами в сети, имеющиеся в CWSI Campus. URT предоставляет дополнительные возможности по идентификации пользователей в сети путем формирования



т. н. "user registration policy bindings" для установки правил регистрации, перемещения и наблюдения за пользовательскими компьютерами в сети.

CW2000 может быть установлен под NT, Solaris, AIX, HP-UX.

Resource Manager Essentials

Resource Manager Essentials состоит из следующих компонент:

- Inventory Manager
- Change Audit Service
- Device Configuration Manager
- Software Image Manager
- Availability Manager
- Syslog Analyzer
- Cisco Management Connection
- CCO Service Tools

Inventory Manager содержит базу данных по сетевым устройствам Cisco, начиная от небольших ISDN маршрутизаторов 700 серии до крупных высокопроизводительных коммутаторов 8000 серии. Данная база данных постоянно пополняется. *Inventory Manager* значительно облегчает задачи сетевого управления предоставляя точные отчеты об аппаратуре и программном обеспечении развернутом в вашей сети.

Change Audit Service предоставляет всесторонний отчет о программных, аппаратных и конфигурационных изменениях в сети.

Device Configuration Manager поддерживает архив активных конфигураций маршрутизаторов и коммутаторов установленных в сети.

Software Image Manager значительно ускоряет анализ текущего программного обеспечения сетевых устройств и заливку нового.

Availability Manager предназначен для быстрого определения текущего рабочего состояния критических (узловых) маршрутизаторов и коммутаторов и их доступности.

Syslog Analyzer предназначен для анализа сообщений, поступающих с сетевых устройств и выдачи сетевому администратору рекомендаций по реакции на них.

Cisco Management Connection позволяет строить быстрые ссылки к наиболее важным приложениям сетевого менеджмента.

Integrated Access to Cisco Connections Online (CCO) осуществляет быстрое подключение к соответствующим разделам на CCO.

CWSI Campus

CWSI Campus обеспечивает всестороннее решение по управлению коммутаторами локальных сетей Cisco Catalyst и ATM коммутаторами LightStream. CWSI Campus предоставляет следующие возможности:

- Всестороннее исследование сети и построение топологической карты на физическом и логическом уровне, интеллектуально отображая до 500 сетевых устройств
- Интуитивно понятный графический интерфейс для легкой и быстрой настройки маршрутизаторов и коммутаторов
- Мониторинг и анализ производительности используя собранные RMON/RMON2 данные трафика
- Конфигурационные и аналитические компоненты для оптимизации производительности LAN и WAN
- Возможности по конфигурированию VLAN для логического сегментирования LAN на бродкастовые группы

- Компоненты для конфигурирования и диагностики ATM и LAN.
- Наблюдение за пользовательскими станциями

CWSI Campus начинает свою работу с исследования вашей сети. Опираясь на заданный узловой коммутатор, CWSI выполняет процесс обнаружения и идентификации устройств в сети и связей между ними. В результате данного исследования строится топологическая карта сети. Полученная карта может быть изменена и сохранена.

Администратор сети так же может получить детальное изображение отдельных устройств и соединений. Администратор сети может указать на какое-либо иконку какого-либо устройства и двойным кликом мышки получить полное графическое изображение данного прибора и активизировать процесс его настройки или мониторинга загрузки.

Топологическая карта сети может предоставлять изображения “multilayer switching (MLS)” устройств (многоуровневых коммутаторов), их “route processor” и “switch engine” компонент (плат маршрутизации и коммутации) и отношений между MLS устройствами в сети.

Вид топологии сети служит так же как точка запуска для следующих приложений:

VLAN director – приложение для конфигурирования VLAN, которое позволяет создавать, изменять или удалять виртуальные сегменты посредством простого манипулирования мышью. Порт коммутатора может быть добавлен в какой-либо VLAN путем обычного указания мышкой.

ATM director – графический интерфейс для конфигурирования ATM интерфейсов и LANE служб. Инструментарий управления ATM и LAN службами позволяет проводить всестороннее исследование ATM устройств (которые поддерживают ILM1), устанавливать и постоянные (PVC) и коммутируемые (SVC) виртуальные соединения и проводить в реальном режиме времени сквозной анализ пути для проверки соединения конечных устройств. Дополнительные возможности управления ATM устройствами включают в себя:

- анализ производительности компонент LAN,
- конфигурирование PNN1 маршрутизации
- мониторинг трафика.

User Tracking – табличный список активных станций подключенных к коммутаторам Catalyst и маршрутизаторам Cisco, обнаруженным CWSI Campus в сети. Данный список представляет собой мощное средство для решения проблем с подключением и расположением пользовательских станций в конкретном виртуальном сегменте. CWSI Campus определяет местонахождение всех активных станций в сети путем просмотра таблиц форвардинга на каждом обнаруженном коммутаторе Catalyst. Информация, включая MAC адрес каждой станции,

коммутатор и порт на коммутаторе к которому подключена станция поддерживается приложением User Tracking. Данная информация, представленная в табличном виде, легко может быть отсортирована и обработана. В список может быть включена дополнительная информация об IP адресе и имени (DNS) станции.

Traffic director - средство для собирания и отображения актуальной RMON информации прямо с коммутатора либо через RMON2-совместимое устройство, как например SwitchProbe. Используя графический интерфейс можно настраивать и проводить мониторинг RMON агента на коммутаторе. С помощью данного приложения можно наблюдать и анализировать трафик в сети и определять реакцию на изменение его параметров.

CiscoView – система отображения и управления, которая позволяя выбирая иконку Cisco маршрутизатора, коммутатора или другого устройства на топологической карте сети, можно получить вид данного устройства. На данном графическом изображении, в режиме реального времени, цветом показывается статус соответствующих портов и модулей устройства. Выбирая изображение соответствующего модуля или порта можно произвести его перенастройку или получить отображение динамического графика его загрузки.

Optivity Enterprise

Bay Networks Optivity Enterprise как и CiscoWorks2000 представляет собой семейство продуктов, предоставляющих при простоте в использовании всеобъемлющее, централизованное и масштабируемое управление всевозможными коммуникационными устройствами от BayNetworks. Optivity Enterprise включает в себя следующие продукты:

- Optivity LAN
- Optivity Internetwork
- Optivity Design & Analysis

Optivity LAN поддерживает RMON и SuperRMON, что предоставляет полноценное управление сбоями, производительностью и конфигурацией сетей Ethernet, FastEthernet, TokenRing и ATM. Optivity LAN позволяет легко следить за состоянием корпоративной сети и быстро обнаруживать и устранять возникшие проблемы.

Optivity Internetwork прекрасное приложение для настройки, мониторинга и диагностики маршрутизаторов от Bay Networks, Cisco и других производителей, которые совместимы с MIB-II. Optivity Internetwork представляет собой набор взаимосвязанных программ, которые облегчают и улучшают управление сложной сети. Интуитивно понятный графический интерфейс предоставляет в реальном времени возможность всеобъемлющего управления маршрутизаторами, их мониторинг и получение информации об их состоянии в одном окне.

Optivity Design & Analysis предоставляет RMON утилиты, предоставляющие полную поддержку RMON. Совместно с такими программами как DesignMan, TrendMan, и TrafficMan Optivity Design & Analysis позволяет собирать, суммировать и наблюдать корреляцию параметров производительности сети за определенные промежутки времени.

Все продукты семейства Optivity Enterprise легко интегрируются с основными платформами управления, такими как HP OpenView, IBM NetView для AIX, и Solstice SunNet Manager.

Optivity Enterprise предоставляет следующие возможности:

- Ускорение решения сетевых проблем

Optivity Enterprise позволяет управлять всей сетевой инфраструктурой с помощью простых встроенных сетевых управляющих систем, с высоким коэффициентом готовности и высокой производительностью, предоставляющих широкий диапазон возможностей многоуровневого сетевого управления. Enterprise Health Advisor обеспечивает полный контроль сбоев, позволяя отслеживать состояние как коммуникационного оборудования так и сегментов локальных и глобальных линий связи. Состояние устройства легко определяется по содержимому окна приложения, разделенному на несколько областей по важности сообщения – критические сбои, опасность, уведомления и рабочие сообщения. Соответствие проблемы определяет Bay Networks Fault Correlator, позволяя тем самым быстрее определить состояние устройства.

- Облегчение планирования сети

Optivity Enterprise также содержит приложения для разработки дальнейшего развития сети, что позволит обеспечить максимальную эффективность эксплуатации сети и использование полосы пропускания. DesignMan используется для моделирования сегментов сети, позволяя протестировать новую структуру локальной сети не внося изменений в уже существующую сеть. TrendMan позволяет анализировать статистику работы сети для определения конфигурации сети, которая предоставит оптимальную производительность.

- Упрощение управления сети

Optivity Enterprise включает в себя приложения, облегчающие управление сетью. Enterprise Command Center определяет и отображает структуру корпоративной сети, группируя сетевые ресурсы по местоположению. Перетаскивание сетевых ресурсов в интеллектуальную палитру утилит позволяет выполнять однотипные действия над группой маршрутизаторов, локальных или глобальных сегментов сети.

– Защищает сделанные инвестиции.

Открытость и масштабируемость Optivity Enterprise позволяет обеспечивать управление расширяющихся, гетерогенных сетей. Предоставляя возможность управления концентраторами, коммутаторами и маршрутизаторами с одной консоли Optivity Enterprise позволяет расширить возможности таких популярных платформ управления как HP OpenView, IBM NetView для AIX и Solstice SunNet Manager. В то время как продукты Optivity Enterprise отвечают за конфигурирование, мониторинг, управление сбоями и диагностированием сетевых устройств, то нижележащая платформа управления может быть использована для физического и логического отображения всей сети, включая концентраторы, коммутаторы, маршрутизаторы и другие управляемые устройства.

Выводы: Результаты исследований получены в ходе проведения научно-исследовательской работы по созданию распределенной гетерогенной сетевой инфраструктуры кафедры ИУ4 МГТУ им.Н.Э.Баумана. В дальнейшем с ростом структуры сети и увеличением обрабатываемого трафика вопросы оптимизации и интеллектуализации управления выйдут на первый план.

ПРОБЛЕМА БИЛЛИНГА В СЕТЯХ ETHERNET И ЕЁ ВОЗМОЖНОЕ РЕШЕНИЕ.

Волков А.Б.

научный руководитель: к.т.н. Власов А.И.

Московский Государственный Технический Университет им. Н.Э. Баумана,
кафедра “Конструирование и производство ЭА”.

BILLING PROBLEM IN NETWORKS ETHERNET AND THE POSSIBLE SOLUTION.

Volkov A.B.

the scientific chief: Ph.D. Vlasov A.I.

Bauman Moscow State Technical University,
Department of Design and Equipment of Electronic System.

Аннотация: На сегодняшний день всё более и более острее становится проблема биллинга и подсчета трафика в сетях Ethernet.

Abstract: For today more also is more acute to become a billing problem and inventory of traffic in networks Ethernet.

Введение

Быстрое развитие сетей, большая популярность Internet, делают систему подсчёта трафика всё более сложной. Для того чтобы не хорошо разбираться в системе работы данных программ, наглядно разберём все аспекты данной проблемы.

Поскольку в основе всего процесса лежит пропускание пакетов, адресованных в интернет, через сервер: маршрутизация пакетов, то прежде всего надо разобраться в сути происходящего.

Определим два понятия: программный маршрутизатор, программа запущенная на компьютере, и аппаратный маршрутизатор, некое устройство, с помощью которого ведётся подсчёт трафика.

Общие сведения о работе Linux-маршрутизатора.

Сервер между локальной сетью и Internet выполняет следующие функции:

- Поддержка постоянной (или по требованию) связи с IP-провайдером
- Маршрутизация IP пакетов между локальной сетью и сетью Internet (routing) для выхода локальных пользователей в сеть Internet.
- Обеспечение IP-сервиса (ftp,telnet,talk и т п) для пользователей локальной сети и сети Internet.
- Обеспечение защиты от несанкционированного доступа к локальной сети из сети Internet .

Настройка routing'a (маршрутизации пакетов между интерфейсами) может быть как статической (однократно настраиваемая таблица), так и динамической.

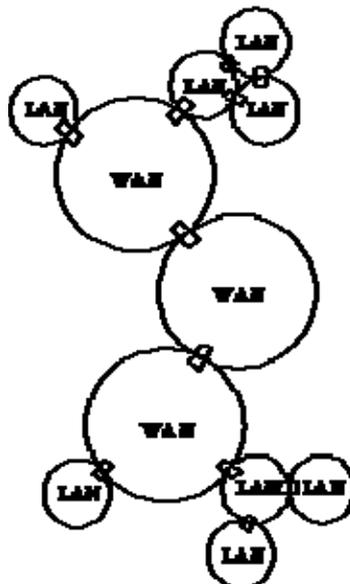
Динамическая настройка бывает необходима в том случае, если у Вас сложная, постоянно меняющаяся структура сети и одна и та же машина может быть доступна по различным интерфейсам (например через разные Ethernet или SLIP интерфейсы). Для динамической настройки routing'a служат команды routed и gated.

Как правило в большинстве ситуаций бывает достаточно статической настройки интерфейса командой route.

Настройка routing'a производится в файле *rc.inet1*. Как правило Вам бывает необходимо настроить routing по вышеперечисленным трем интерфейсам:

- Локальный интерфейс (lo).
- Интерфейс для платы Ethernet (eth0).

- Интерфейс для серийного порта - SLIP/CSLIP (sl0).



Краткое описание IPX/SPX и TCP/IP протокола.

Рассмотрим протокол TCP/IP: Пакет IP состоит из заголовка и поля данных. Заголовок пакета имеет следующие поля:

- Поле Номер версии (VERS) указывает версию протокола IP. Сейчас повсеместно используется версия 4 и готовится переход на версию 6, называемую также IPng (IP next generation).
- Поле Длина заголовка (HLEN) пакета IP занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байт в поле Резерв (IP OPTIONS).
- Поле Тип сервиса (SERVICE TYPE) занимает 1 байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе приоритета пакета (PRECEDENCE). Приоритет может иметь значения от 0 (нормальный пакет) до 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Поле Тип сервиса содержит также три бита, определяющие критерий выбора маршрута. Установленный бит D (delay) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T - для максимизации пропускной способности, а бит R - для максимизации надежности доставки.
- Поле Общая длина (TOTAL LENGTH) занимает 2 байта и указывает общую длину пакета с учетом заголовка и поля данных.
- Поле Идентификатор пакета (IDENTIFICATION) занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля.
- Поле Флаги (FLAGS) занимает 3 бита, оно указывает на возможность фрагментации пакета (установленный бит Do not Fragment - DF - запрещает маршрутизатору фрагментировать данный пакет), а также на то, является ли данный пакет промежуточным или последним фрагментом исходного пакета (установленный бит More Fragments - MF - говорит о том пакет переносит промежуточный фрагмент).
- Поле Смещение фрагмента (FRAGMENT OFFSET) занимает 13 бит, оно используется для указания в байтах смещения поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами максимальной длины пакета.
- Поле Время жизни (TIME TO LIVE) занимает 1 байт и указывает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеряется в секундах и задается источником передачи средствами протокола IP. На шлюзах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица; единица вычитается также при каждой транзитной передаче (даже если не прошла секунда). При истечении времени жизни пакет аннулируется.

- Идентификатор Протокола верхнего уровня (PROTOCOL) занимает 1 байт и указывает, какому протоколу верхнего уровня принадлежит пакет (например, это могут быть протоколы TCP, UDP или RIP).
- Контрольная сумма (HEADER CHECKSUM) занимает 2 байта, она рассчитывается по всему заголовку.
- Поля Адрес источника (SOURCE IP ADDRESS) и Адрес назначения (DESTINATION IP ADDRESS) имеют одинаковую длину - 32 бита, и одинаковую структуру.
- Поле Резерв (IP OPTIONS) является необязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей, каждое из которых может быть одного из восьми predefined типов. В этих подполях можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки. Так как число подполей может быть произвольным, то в конце поля Резерв должно быть добавлено несколько байт для выравнивания заголовка пакета по 32-битной границе.

Максимальная длина поля данных пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65535 байтов, однако при передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной в 1500 байтов, уместающиеся в поле данных кадра Ethernet.

Самым важным полем для нас, является поле Общая длина (TOTAL LENGTH). Путём простого вычитания из него объёма служебной информации мы получим размер Объёма Данных. А так же поле Адрес источника (SOURCE IP ADDRESS) для вычисления “хозяина” пакета.

Теперь перейдём к протоколу IPX/SPX, но так как этот стек состоит из 2ух частей, то мы будем рассматривать только IPX, т.к. именно в нём содержится интересующая нас информация.

Рассмотрим подробнее назначение отдельных полей пакета.

Поле *Checksum* предназначено для хранения контрольной суммы пакета или другой служебной информации. В прикладных программах обычно не используется.

2	Checksum	-	контрольная сумма
2	Length	-	общая длина пакета
1	TransportControl	-	счетчик пройденных маршрутизаторов
1	PacketType	-	тип пакета
4	DestNetwork	-	номер сети получателя пакета
6	DestNode	-	адрес станции-получателя
2	DestSocket	-	гнездо программы-получателя
4	SourceNetwork	-	номер сети отправителя пакета
6	SourceNode	-	адрес станции-отправителя
2	SourceSocket	-	гнездо программы-отправителя
длина	Data	-	Передаваемые данные

Поле *Length* определяет общий размер пакета вместе с заголовком. NetWare поддерживает следующие максимальные длины пакетов: Token Ring и ARCnet - 4202 байта, Ethernet - 1514 байтов. Это поле устанавливается протоколом IPX передающей станции.

Поле *TransportControl* служит как бы счетчиком маршрутизаторов, которые проходит пакет на своём пути от передающей станции к принимающей. В начале это поле устанавливается в 0 протоколом IPX передающей станции.

Поле *PacketType* определяет тип передаваемого пакета. Программа, которая передаёт пакет средствами IPX, должна записывать в это поле значение 0x04.

Поле *DestNetwork* определяет номер сети, в которую передаётся пакет. Устанавливается в прикладной программе. Если в поле указывается нулевое значение, то пакет передаётся в сеть (сегмент), к которой подключена станция.

Поле *DestNode* определяет адрес станции, которой предназначен пакет. Устанавливается прикладной программой. Если пакет предназначен всем станциям в сети (сегменте), то в поле указывается значение FFFFFFFh.

Поле *DestSocket* предназначено для определения программы, которая запущена на станции-получателе и должна принять пакет. Это поле устанавливается в прикладной программе.

Поля *SourceNetwork*, *SourceNode*, *SourceSocket* содержат соответственно номер сети, из которой посылается пакет, адрес передающей станции и гнездо программы, которая передаёт пакет. Эти поля заполняются протоколом IPX передающей станции.

Поле *Data* в пакете IPX содержит передаваемые данные. Это поле формируется протоколом IPX передающей станции на основании описания блока ECB

В данном случае, нам важны поля общая длина пакета, после преобразования которой мы получим размер передаваемых данных, и адрес станции-отправителя.

После того, как мы познакомились с устройством протоколов и работой программного маршрутизатора, приступим к созданию конкретной схемы работы “счётчика”.

Возможные решения и их практическая реализация.

Рассмотрим 2 простейшие схемы. Первая – самая простая и подходит только для очень слабо нагруженных сетей, без возможности контроля доступа(использую только распараллеливание потока).И вторая – более сложная, подходящая к большим сетям, с большей нагрузкой на сервер и требующие большей защиты.

Схема первая: Данная схема основана на простом захвате пакета, после чего извлекается поле объёма данных. После этого данные приплюсовываются к уже имеющимся данным, сопоставленным с данным адресом(IP). Работа данной схемы схожа с работой Sniffer’a и по этому не обязательно, чтобы данная программа была запущена на маршрутизаторе, только в том случае, когда сеть ширококвещательная.

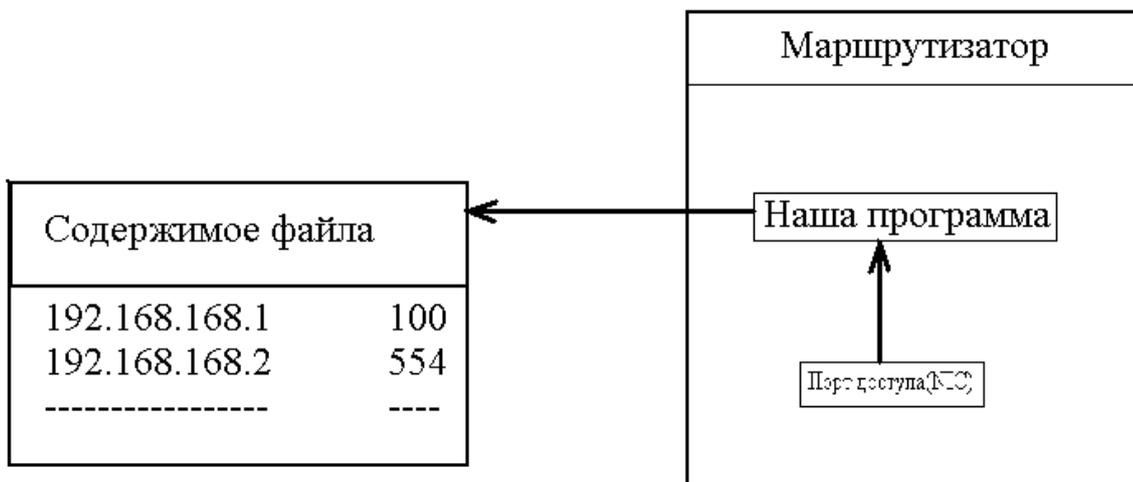
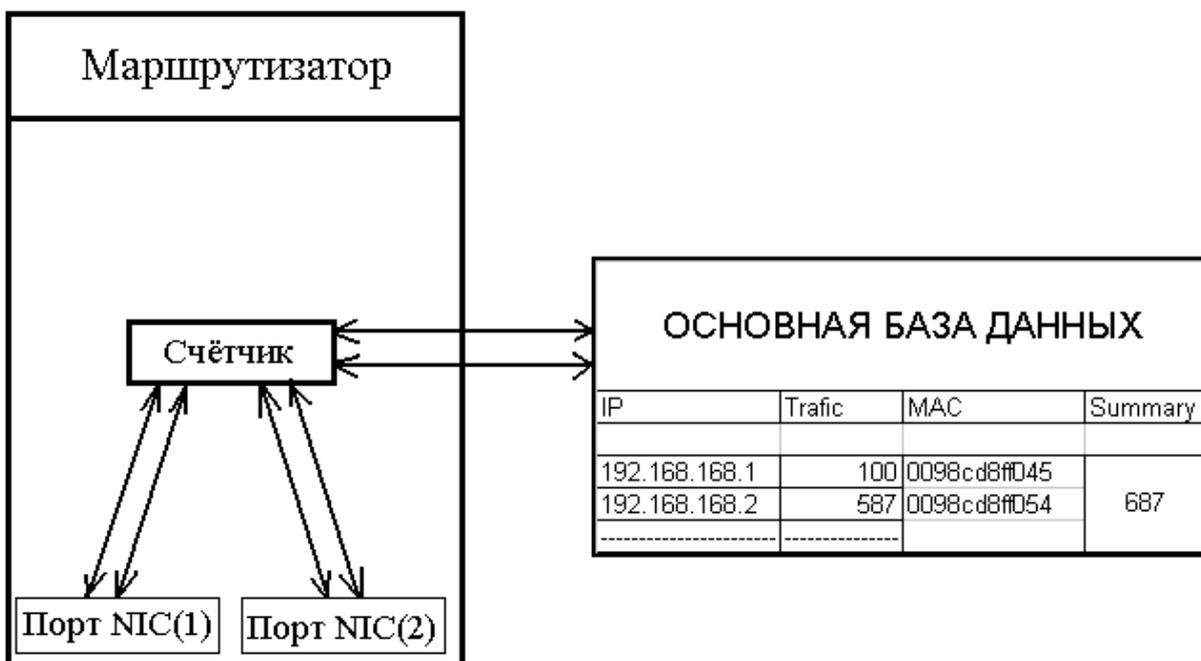


Схема вторая: Это более сложная схема, нежели предыдущая. Здесь реализован контроль доступа(прохода пакетов сквозь маршрутизатор). Так же здесь используется База Данных, содержащая все данные и настройки

относительно работы счётчика. А так же сама программа полностью осуществляем маршрутизацию, исходя из содержимого Базы Данных.



Выводы.

Теперь, немного разобравшись в некоторых аспектах данной проблемы, мы можем более серьёзно подойти к решению проблемы. Но, исходя из рассмотренных моделей и их всевозможных вариаций, приходим к выводу, что использование программных счётчиков не всегда удобно и надёжно. Но в применении к относительно небольшому кол-ву пользователей, небольшому трафику и не необходимости хорошей защиты.

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ НА ПРИМЕРЕ СИСТЕМ РАСПОЗНАВАНИЯ ОБРАЗОВ.

Новиков И.С.

Московский Государственный Технический Университет им. Н.Э. Баумана,
кафедра “Конструирование и производство ЭА”.

ARTIFICIAL INTELLIGENCE ON AN EXAMPLE OF PATTERN RECOGNITION SYSTEMS

Novikov I.S.

Bauman Moscow State Technical University,
Department of Design and Equipment of Electronic System.

Аннотация: В работе исследованы методы обработки изображений и технического зрения.

Abstract: In activity the processing techniques of the maps and technical vision is investigated.

Исторический обзор.

С конца 40-х годов ученые все большего числа университетских и промышленных исследовательских лабораторий устремились к дерзкой цели: построение компьютеров, действующих таким образом, что по результатам работы их невозможно было бы отличить от человеческого разума. Вскоре многие из них поняли, что создателям вычислительных машин есть чему поучиться у биологии. Среди них был нейрофизиолог и поэт-любитель Уоррен Маккаллох, обладавший, философским складом ума и широким кругом интересов. Он занимался изучением философии и психологии в Йельском университете, а затем поступил в медицинскую школу. Посвятив некоторое время исследованиям эпилепсии и травм черепа, Маккаллох углубился в изучение центральной нервной системы. Потом он перебрался в Иллинойский университет, где получил должность руководителя лаборатории фундаментальных исследований факультета психиатрии. В 1942 г. Маккаллох, участвуя в научной конференции в Нью-Йорке, услышал доклад одного из сотрудников Винера о механизмах обратной связи в биологии. Высказанные в докладе представления об обратной связи в чем-то перекликались с собственными идеями Маккаллоха относительно работы мозга. В течение следующего года Маккаллох в соавторстве со своим 18-летним протеже, блестящим математиком Уолтером Питтсом, разработал теорию деятельности головного мозга. Эта теория и явилась той основой, на которой сформировалось широко распространенное мнение, что функции компьютеров и мозга в значительной мере сходны. Исходя отчасти из предшествующих исследований нейронов (основных активных клеток, составляющих нервную систему животных), проведенных Маккаллохом, они с Питтсом выдвинули гипотезу, что нейроны можно упрощенно рассматривать как устройства, оперирующие двоичными числами. Двоичные числа, состоящие из цифр единица и ноль, – рабочий инструмент одной из систем математической логики. Маккаллох и Питтс предложили конструкцию сети из электронных «нейронов» и показали, что подобная сеть может выполнять практически любые вообразимые числовые или логические операции. Далее они предположили, что такая сеть в состоянии также обучаться, распознавать образы, обобщать, т.е она обладает основными элементами интеллекта.

Теории Маккаллоха – Питтса в сочетании с книгами Винера («Кибернетика» и вышедшая через два года после нее работа «Человеческое использование человеческих существ») вызвали огромный интерес к разумным машинам. В 40 – 60-е годы все больше

кибернетиков из университетов и частных фирм запирались в лабораториях и мастерских, напряженно работая над теорией функционирования мозга и методично припаявая электронные компоненты моделей нейронов.

Из этого кибернетического, или нейромодельного, подхода к машинному разуму скоро сформировался так называемый «восходящий метод» – движение от простых аналогов нервной системы примитивных существ, обладающих малым числом нейронов, к сложнейшей нервной системе человека и даже выше. Основной трудностью, с которой столкнулся «восходящий метод» на заре своего существования, была высокая стоимость электронных элементов. Слишком дорогой оказывалась даже модель нервной системы муравья, состоящей из 20 тыс. нейронов, не говоря уже о нервной системе человека, включающей около 100 млрд. нейронов. Даже самые совершенные кибернетические модели содержали лишь несколько сотен нейронов. Столь ограниченные возможности обескуражили многих исследователей того периода.

Одним из тех ученых, кого ничуть не испугали трудности, был Фрэнк Розенблат, труды которого, казалось, отвечали самым заветным устремлениям кибернетиков. В 1956 г. в Корнеллском университете защитил докторскую диссертацию по экспериментальной психопатологии. Поступив в Корнеллскую лабораторию авиации в качестве психолога-исследователя, Розенблат, проявлявший повышенный интерес к механизмам работы мозга, стал искать способ их электронного моделирования. В середине 1958 г. молодой ученый продемонстрировал компьютерную модель электронного устройства, названного им перцептроном, которое должно было имитировать процессы человеческого мышления. Он заявил, что при мерно через год предполагает построить действующую модель.

При всей своей примитивности модель Розенבלата произвела сильное впечатление на собравшихся. Он запрограммировал один из самых мощных компьютеров того времени, ИБМ-704, так, чтобы тот моделировал действие электронной схемы перцептрона, столь сложной, что мощному компьютеру требовалось около получаса на решение задачи, с которой реальный перцептрон должен был справиться за тысячные доли секунды. Перцептрон должен был передавать сигналы от «глаза», составленного из фотоэлементов, в блоки электромеханических ячеек памяти, которые оценивали относительную величину электрических сигналов. Эти ячейки соединялись между собой случайным образом в соответствии с господствовавшей тогда теорией, согласно которой мозг воспринимает новую информацию и реагирует на нее через систему случайных связей между нейронами. В предложенной модели система просматривала два изображения, составленные из квадратиков, сортируя сигналы и различая изображения каким-то неведомым способом, который, по словам Розенבלата, можно было описать только при помощи сугубо специальных технических терминов.

ПРИНЦИПЫ МАШИННОГО ЗРЕНИЯ

Начиная с 1958 г., когда Фрэнк Розенблат предложил так называемый перцептрон, специалисты по искусственному интеллекту не ослабляют попыток воспроизвести с помощью компьютера сложные процессы зрительного восприятия. Нет сомнения, что всякая попытка создать автономные машины, умеющие решать разнообразные задачи в условиях изменяющейся внешней обстановки, неизбежно приведет к необходимости наделить такие машины способностью не только видеть предметы, но и распознавать их, т.е. понимать, что именно они видят. Люди легко справляются с этим, узнавая знакомые предметы на различных расстояниях, при различном освещении и практически при любых углах зрения. То, что процессы зрительного восприятия у человека почти не затрагивают сознания, порождает серьезные проблемы с точки зрения создания ИИ: эти процессы плохо поддаются исследованиям, а поэтому их очень трудно взять за основу при разработке систем машинного зрения.

В ниже мы рассмотрим ключевой аспект машинного зрения – распознавание образов. Имей мы машины, обладающие способностью распознавать, им можно было бы поручить,

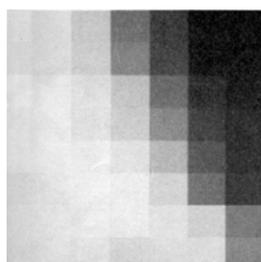
как минимум, операции технического контроля и сортировки на производстве; они могли бы также интерпретировать изображения, передаваемые спутниками, которые следят, например, за погодой или состоянием сельскохозяйственных угодий.

Любая система машинного зрения должна уметь формировать, анализировать и интерпретировать изображения. Задача формирования изображений решается достаточно просто: компьютер принимает от «глаз» системы (телекамер) результаты измерений силы света, отраженного от различных точек поверхностей объектов трехмерного мира. Хотя компьютер мог бы без особых затруднений превратить полученную информацию в изображение на экране, это не приблизит его к распознаванию самого объекта. Поэтому система обрабатывает исходный массив чисел в соответствии с определенным алгоритмом, позволяющим выделить области, где интенсивности меняются резко или по какому-то конкретному закону. Цель такого анализа – получить описания объектов, которыми можно будет пользоваться в процессе распознавания. Определив такие свойства объекта, как цвет, форма и текстура поверхности, машина пытается сопоставить их с информацией, заранее заложенной в ее память. Сопоставление – наиболее ответственный элемент машинного зрения. Множество исследований было посвящено поискам наиболее совершенных способов представления информации в компьютере и разработке процедур, позволяющих машине на основе этой информации понять, что представляет собой наблюдаемый объект и как следует с ним поступать.

Преобразование изображения в массив чисел.

Из всех операций, выполняемых при машинном зрительном восприятии, лучше всего отработан процесс формирования изображений внутри системы. Телевизионная камера регистрирует значения силы света, отраженного от различных точек поверхностей трехмерных объектов. Эта информация преобразуется в электрический сигнал, величина которого пропорциональна интенсивности отраженного света. Далее полученный таким образом аналоговый электрический сигнал преобразуется в цифровой, воспринимаемый компьютером. С этой целью величину сигнала измеряют через равные интервалы, и каждое полученное значение переводится в число, которое показывает положение данной точки в выбранном диапазоне яркостей (интенсивностей), называемом шкалой серого. Эти числа образуют двумерную решетку – массив уровней серого, каждый элемент которой соответствует некоторой точке преобразованного в цифровую форму изображения – пикселу (*pixel*, производное от *picture element* – элемент изображения).

В устройствах машинного зрения для систем искусственного интеллекта шкала серого обычно охватывает диапазон от 0 до 255: нуль соответствует самым темным точкам изображения, 255 – самым светлым. В системах цветного зрения отдельно измеряются интенсивности по каждому из основных цветов: красному, зеленому и синему. Они преобразуются в три массива значений яркости, где каждое значение может меняться в пределах от 0 до 255. Таким образом, в системах цветного зрения на каждый пиксел приходится обрабатывать втрое больше информации, чем в черно-белых.



53	50	43	33	27	24	24
50	49	43	34	27	23	23
55	52	48	40	31	25	23
52	52	47	41	33	26	24
52	52	50	46	37	28	24
48	50	51	49	42	31	25
51	52	51	49	46	41	32
51	51	48	45	45	44	34

Рис.1. Сильно увеличенное изображение

Числа в показанной выше матрице уровней серого цвета соответствуют значениям яркости в сетке 7*8 пикселей (Рис. 1), которая была выделена из преобразованного в преобразованного в цифровую форму изображения. Резкое изменение значение вдоль главной диагонали матрицы соответствует перепаду яркости на данном увеличенном фрагменте изображения.

Поиск линий и границ.

Чтобы выделить зафиксированный телекамерой объект, компьютер системы машинного зрения анализирует изображение, преобразованное в матрицу уровней серого цвета. Программа находит начертания частей объекта, которым соответствуют резкие перепады яркости.

Получив массив чисел – значений интенсивности света, отраженного от объектов исходной трехмерной сцены, машина должна расшифровать их, т.е. выяснить, что же означают эти числа. Процесс обычно начинается с так называемого поиска контуров – процедуры, позволяющей машине найти очертания объектов или их частей. Алгоритм поиска контуров заставляет компьютер искать резкие перепады яркости, которые обычно наблюдаются на границах объекта, при резких изменениях рельефа или цвета поверхности. Например, на черно-белом изображении (Рис.1) наиболее резкие переходы от темного к светлому соответствуют границам между темный фоном и светлым объектом. Переходы отчасти стараются так называемыми шумами - небольшими колебаниями интенсивности, обусловленными шероховатостями поверхности, а также неизбежными электронными помехами, возникающими в процессе преобразования аналогового сигнала в цифровую форму. Чтобы свести к минимуму ошибки при определении контуров, компьютер должен предварительно исключить (или снизить до допустимого уровня) шумовые эффекты, используя для этого процедуру, называемую сглаживанием (Рис. 2). Оставшиеся после сглаживания существенные изменения яркости вычерчиваются в виде контурного изображения, или контурной карты.

На графике слева представлены значения яркости, полученные при сканировании строки пикселей, выделенной из изображения. На кривой видны как малые изменения яркости на поверхности объекта, так и ее значительные перепады. В результате сглаживания получается график, показанный справа. При сглаживании значение яркости каждого пиксела заменяется взвешенным средним яркостей данного пиксела и его ближайших «соседей» слева и справа. Чем по большему числу соседних пикселей производится усреднение, тем более гладким получается график и тем меньше шумовых эффектов содержит окончательная контурная карта изображения.



Рис. 2

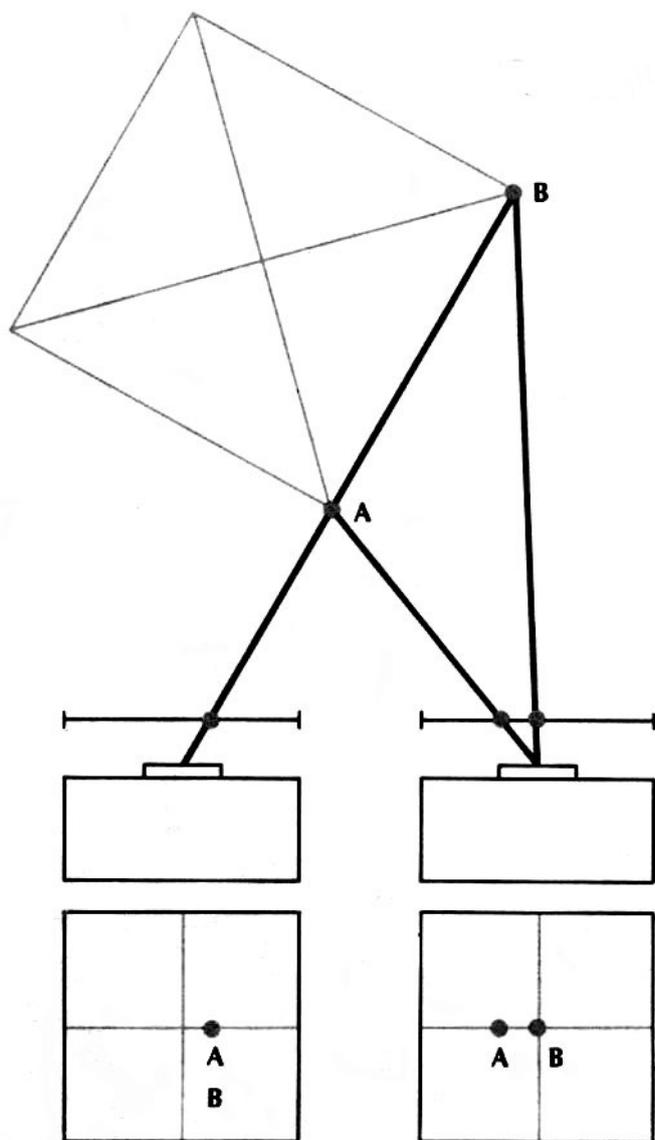
Проблема восприятия глубины.

При всей полезности процедуры поиска контуров для распознавания образов сама по себе она дает лишь двумерный контур объекта, который меняется в зависимости от точки наблюдения. А чтобы воспринимать глубину изображения, машина должна получать информацию, достаточную для надежного распознавания трехмерных объектов.

Наша способность к объемному восприятию отчасти обусловлена бинокулярным зрением. Мозг, получив изображения от каждого глаза, по смещениям соответственных точек изображений создает ощущение глубины. Чтобы применить подобный прием в системах машинного зрения, необходимо решить одну важную проблему. Оснастить машину двумя камерами, которые, подобно глазам человека, будут давать изображения с разных точек, совсем не сложно. Гораздо труднее найти в этих двух изображениях соответственные точки для определения смещения между ними.

Эта так называемая проблема соответствия вызвана тем, что компьютер в системе со стереоскопическим зрением не имеет о внешних объектах никаких других данных, кроме двух числовых массивов яркостей, полученных от каждой из камер. Уровни серого одной и той же точки объекта в двух массивах могут не только оказаться в разных позициях, но и отличаться друг от друга из-за разницы в положениях камер.

Одно из решений проблемы соответствия состоит в сведении массивов уровней серого к контурным картам при помощи процедуры поиска контуров. Затем система сканирует карты в поисках похожих участков. Выделив соответственные точки в таких участках, можно определить их положения относительно центров плоскостей изображений каждой из камер. По разнице их положений в двух наборах данных система определяет расстояния от камер до каждой из точек объекта и так, точка за точкой, восстанавливает объемную форму.



На схеме представлен вид сверху на пирамиду. На виде плоскости изображения камер показаны горизонтальными отрезками перед объективами. Для наглядности эти плоскости показаны отдельно в нижней части схемы. Точки А и В, соответствующие переднему и правому углам пирамиды, в правой камере регистрируются как раздельные, а в левой камере точки А и В сливаются.

Определив, что точка А в правой камере соответствует точке А в левой камере компьютер может рассчитать, насколько и в каком направлении эти точки смещены относительно центров соответствующих плоскостей изображений. Аналогичный расчет выполняется и для точки В. Исходя из того, что смещение между точками А в камерах больше, чем между точками В, компьютер определяет, что точка А находится ближе к камере, чем точка В.

Описание текстуры поверхности.

Один из методов, с помощью которых система машинного зрения «конструирует» объект из мешанины данных о его внешнем изображении, – анализ текстуры – применяется и при распознавании предметов. Конкретная текстура поверхности представляется в массиве уровней серого определенными комбинациями значений яркости. Резкие изменения текстуры чаще всего указывают на переходы от одной физической поверхности к другой, например, от луга к лесу на аэрофотоснимках. Плавные же изменения текстуры могут дать ключ к распознаванию трехмерной формы объекта.

Для описания текстуры поверхности применяются два основных метода: структурный и статистический анализы. При структурном анализе система ищет характерные особенности, или приметы, объекта и соотношения между ними. При выявлении примет система обычно пользуется для нахождения линий раздела методом поиска контуров в массиве уровней серого.

Статистический подход применяется для анализа поверхностей, в текстуре которых трудно выделить характерные приметы, или изображений, в которых такие детали не видны, как, например, при получении некоторых изображений со спутников. Здесь внимание концентрируется на соотношении яркостей данного пиксела и его окружения в массиве уровней серого, и в результате определяется вероятность того, что данный пиксел близок по яркости своим соседям. Например, статистический анализ изображения меха должен

показать, что по яркости более близки соседние пиксели по вертикали, чем по горизонтали. Текстура меха с его тонкими набегающими друг на друга волосками обладает свойством направленности.

Выявление цвета.

Основная трудность, с которой сталкиваются системы машинного зрения при попытках распознать объект только по его цвету, обусловлена освещением: воспринимаемый цвет сильно зависит от освещения, а оно в реальном мире непрерывно меняется. Даже такая вроде бы простая задача, как отличить яблоко от апельсина, становится сложной, если оба фрукта сильно затенены.

Всякий цвет обладает тремя независимыми признаками: собственно цвет, т.е. то, что обычно описывают словами «красный», «оранжевый», «синий» и т. д., интенсивность, или степень яркости, и насыщенность, показывающая, насколько цвет чистый. Освещение изменяет интенсивность и насыщенность воспринимаемого цвета, тогда как сам цвет относительно независим от освещения.

Телекамера в системах машинного зрения в состоянии регистрировать только красный, зеленый и синий компоненты света (RGB). Таким образом, чтобы распознать собственно цвет объекта, система должна сначала преобразовать информацию о яркости трех основных компонентов в данные о цвете, интенсивности и насыщенности. Этот процесс, который включает несколько математических преобразований, проиллюстрирован на рисунке 3, где показана трехмерная модель цветов, применяемая в системе машинного зрения. Чистые красный, зеленый и синий цвета расходятся вдоль трех ребер куба из общей черной вершины, в которой значения яркости всех трех цветов нулевые. Так называемая «ось интенсивности» проходит по диагонали куба от черного (не освещено) – через все варианты серого – до белого (максимальная освещенность).

	Красный	Зеленый	Синий
Черный	0	0	0
Красный	255	0	0
Зеленый	0	255	0
Синий	0	0	255
Желтый	255	255	0
Пурпурный	255	0	255
Морской волны	0	255	255
Белый	255	255	255

В таблице приведены доли красного, зеленого и синего в различных цветах, воспринимаемых камерой системы машинного зрения. Например, чистый красный цвет регистрируется как максимальное значение красного и нулевые - зеленого и синего; желтый - как равные части зеленого и красного. Черный цвет представляется как отсутствие всех трех основных цветов, а белый - как максимальные значения по всем трем шкалам.

Чтобы установить основной оттенок предмета, система машины всего определить яркость и насыщенность цвета точки поверхности, выраженного в тройке значений интенсивностей красного, зеленого и синего. Здесь показана тройка КЗС некоторой точки. Взяв среднее от этих трех значений, система находит общую интенсивность и откладывает ее на оси интенсивности. Затем через полученную точку проводится плоскость, перпендикулярная оси интенсивности, на которой отмечается положение точки КЗС.

После того, как данное значение тройки КЗС отмечено на плоскости, определяется расстояние от него до центра (уровень серого на оси интенсивности), которое соответствует насыщенности - чистоте цвета. Затем измеряя угол между линией насыщенности и одной из базовых линий, проведенной, например, из красной вершины в центр, находят оттенок.

Поиски подходящей модели.

Чтобы распознать объект, после того как определены его характеристики, машина должна сопоставить описания его формы, контуров, цвета и текстуры поверхности с особенностями хранящихся в ее памяти моделей. Мы рассмотрим три способа такого сопоставления. В

наипростейшем методе сопоставления с шаблоном машина отыскивает среди ограниченного числа заранее известных ей объектов (шаблонов) наиболее похожий на распознаваемый. В некоторых системах, например в системах оптического считывания символов, распознающих печатные знаки, хранится только по одному шаблону для каждого типа объектов. В других системах, в частности у промышленных роботов, которые должны выбирать детали с конвейера, в память заложено несколько вариантов. В любом случае объект должен представать перед камерой в одном из ограниченного множества положений: если его повернуть так, что он не будет похож ни на один из шаблонов, распознавание оказывается невозможным.

Выявление признаков допускает большее отклонение от «идеала». В этом случае система классифицирует объекты по нескольким параметрам, или признакам, которых обычно должно быть не менее двух. Еще более сложный реляционный метод позволяет машине распознавать составные объекты, которые могут представать в различных комбинациях; этим методом исследуют структурные взаимосвязи между составными частями объекта, а не описания самих частей. Например, люди могут быть высокого или низкого роста, худыми и толстыми, мужского или женского пола, они могут лежать, стоять, положив руки на бедра, сидеть, скрестив ноги. Заложённая в память машины модель называется картой связности; на языке этой карты объект представляется как набор компонентов и взаимосвязей между ними типа «связан с...», «состоит из...», «является частью...».

Когда идеалы сталкиваются с действительностью.

Хотя системы машинного зрения уже используются во многих областях, большинство из них работает лишь в строго определенных условиях. Например, оптические устройства считывания символов (сканеры) воспринимают знаки, отпечатанные определенным шрифтом, но абсолютно бессильны перед рукописным текстом. Автомобили-автоматы способны двигаться по дорогам, но лишь при условии, что разница между дорогой и обочиной выражена достаточно четко; не исключено, что такой автомобиль врежется в машины на стоянке, если та вплотную примыкает к дороге. Точно так же машины, запрограммированные для распознавания объекта в некотором определенном ракурсе, не сработают, когда тот же объект предстает перед ними в других положениях. Чтобы машинное зрение оказалось действительно полезным, исследователи ИИ должны найти способ задавать машине реальный мир во всем его разнообразии.

Изменение ракурса – лишь часть проблемы: вид объекта может зависеть от множества других причин. Человек без труда понимает, что очищенный и разделенный на дольки апельсин – все равно апельсин, а система машинного зрения обнаружит лишь, что такой объект не имеет ничего общего с целым фруктом. Наложение на конвейере одной металлической детали на другую может привести к тому, что эта пара будет зарегистрирована как некий третий, неизвестный объект. Системы машинного зрения до сих пор испытывают затруднения при выборе деталей из бункера или из неупорядоченной кучи. При изменении: освещения меняются цвет и текстура поверхности, возникают кажущиеся границы объектов, например тени, которые не имеют никакого отношения к реальным особенностям или объемной форме объекта. И наконец, представители определенных классов объектов, таких как люди, собаки, автомобили, деревья, нередко так сильно отличаются друг от друга, что любой из них может оказаться не соответствующим «идеалу» машины.

Список использованных источников:

1. Компьютер обретает разум. Под ред. В.Л. Стефанюка. -Изд. Мир, 1990.
2. Кибернетика и общество. Винер Н. -Изд. ИЛ, 1958
3. Перцептроны. Минский М., Пейперт С. -Изд.Мир,1971

ВИРТУАЛЬНЫЙ КЛУБ PERL ПРОГРАММИСТОВ И ВОЗМОЖНОСТИ ЗАВТРАШНЕГО ДНЯ

Раубель Александр

Научный руководитель: к.т.н. Власов А.И.

ИУ4, Московский государственный Технический Университет им. Н.Э. Баумана

VIRTUAL CLUB PERL OF THE PROGRAMMERS AND CAPABILITY OF TOMORROW'S DAY

Raubel A.

The scientific chief: Ph.D. Vlasov A.I.

ИУ4, BMSTU

Аннотация: Данная работа посвящена созданию виртуального клуба программистов. Особое внимание уделено проблемам создания дружественного пользовательского интерфейса и удобной системы навигации по сайту. Также рассмотрены методы и технология создания подобных ресурсов.

Abstract: the given activity is dedicated to creation of virtual club of the programmers. The special attention is given to problems of creation of amicable user interface and friend navigation system on a site. Methods and technology of creation of similar resources also are reviewed.

Введение.

На сегодняшний день без технологии CGI не может обойтись не один известный сайт интернета. Поисковые системы, форумы и конференции, гостевые книги и счетчики посещений - это далеко не полный список того, что создается с помощью этой технологии. Язык Perl, на котором преимущественно и пишутся CGI-скрипты, знает большинство программистов, работа которых имеет непосредственное отношение к глобальной сети. Однако необходимо заметить, что все те, кто решил освоить этот язык даже частично, непременно сталкиваются с проблемами, решить которые составляет для них практически невозможным. Поэтому создание "Виртуального клуба Perl-программистов" не могло не найти положительных откликов не только от начинающих, но и от опытных программистов, работающих в сфере WEB-технологий.

Возникшие проблемы и методы их решения.

Итак, главная задача, которая стояла перед создателями Клуба - это оперативно и быстро предоставить квалифицированную помощь программисту в решении его проблемы. Именно на эту формулировку опирались создатели при проектировании общего дизайна и структуры сервера. Внедрение языков JAVA и JAVASCRIPT в документы HTML сделало дизайн виртуального клуба максимально удобным для его посетителей и обеспечило удобную навигацию по сайту.

Первой же, неотъемлемой частью сервера стал форум, зайти в который может каждый посетитель сайта. Если раньше программист оставался наедине со своими задачами, то благодаря возможности общаться в форуме, любой человек может оставить там свои комментарии или интересующий его вопрос. Во многих ситуациях спросить что-нибудь у многочисленных клиентов клуба гораздо проще, чем покупать специализированные книги. Что касается возможности стать членом "Виртуального клуба Perl-программистов" и воспользоваться его услугами, то надо сказать, что раздел "регистрация" также был создан одним из первых.



Зарегистрировавшись, программист получает ряд значительных преимуществ. Во-первых, это возможность получения по почте последних новостей сайта, а также некоторых новых программ и анонсов только что поступившего аналитического материала. Благодаря внедрению cgi-скриптов, каждый состоящий в клубе программист может за считанные секунды оформить себе рассылку, указав свое имя, адрес и клубный пароль. Второе преимущество, которое дает регистрация, которое стоит отметить, заключается в том, что виртуальный клуб всегда заботится о своих клиентах, поэтому каждый зарегистрировавшийся программист может всегда рассчитывать на быструю помощь в решении любого его вопроса.

Третье, что составляет основу "Виртуального клуба Perl-программистов" - это большое количество аналитической информации, собранной на сервере. Благодаря возможности поиска по сайту, созданной также при помощи технологии CGI, представляется возможным моментально найти интересующую Вас статью. На сервере собрано огромное количество материала не только о Perl и некоторых других языках программирования, но и о средствах создания собственных сайтов и о других новейших технологиях всемирной сети. Все статьи оценены по пятибалльной шкале, что на много облегчает задачу выбора материала для начинающих программистов. Надо заметить, что база данных виртуального клуба постоянно пополняется новыми материалами.

Одним из главных моментов, что выгодно отличает данный проект от других серверов с похожей тематикой, - это обширный выбор разнообразных готовых CGI-скриптов и программ. Большинство программ написано студентами различных ВУЗов и другими опытными программистами, работающими в сфере интернет-технологий.

Заключение.

Таким образом, Виртуальный клуб Perl-программистов представляет собой огромный узел интернета, основная задача которого - это предоставить максимальную помощь своим клиентам. Благодаря ему, любой программист может пообщаться со своими коллегами в режиме реального времени и получить ответы на свои вопросы за несколько минут...

ПРОБЛЕМНО-ОРИЕНТИРОВАННАЯ БАЗА ДАННЫХ ДЛЯ ПОДДЕРЖАНИЯ
ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ С ИСПОЛЬЗОВАНИЕМ НЛП.

Лапшин М.О., Резчикова Е.В., Цвилева Е.С.

Научный руководитель: кандидат технической наук, доцент Резчикова Е.В.

ИУ4, МГТУ им. Н.Э. Баумана

USING NLP TECHNOLOGIES IN PROBLEM-ORIENTED DATABASE FOR SUPPORTING
INTELLECTUAL ACTIVITY.

Lapshin M.O., Rezchikova E.V., Tsvileva E.S.

Scientific leader: candidate of technical science, assistant professor Rezchikova E.V.

**IU4, Moscow State Technical University named after Bauman N.
E.**

Аннотация. В данной работе можно выделить три основных раздела. Первый содержит основные термины, необходимые для полноценного использования методик теории НЛП для усвоения технических дисциплин, а также для дальнейшего изучения и применения подходов НЛП в творческой инженерной деятельности. Во втором разделе приводится аксиоматический аппарат НЛП. Данные утверждения стимулируют творческое мышление, позволяя увидеть решаемую проблему в различных аспектах, через определенный фильтр восприятия. Третий раздел посвящен понятию «предикаты», использование которых позволяет осуществлять хорошую коммуникацию, являющуюся одним из основных условий учебного процесса, а в инженерной деятельности - эффективно формировать команды, улучшая взаимодействие внутри группы и, как следствие, повышать результат совместной работы.

Abstract. There are three main parts in this article. The first one contains main terms of NLP technic. They are needed for sterling using NLP methods in learning technical subjects and for the future study and use NLP in engineering. The second contains postulates of NLP. These axioms actuate innovating thinking (creative ability) and let see the problem in different aspects through the special filter of perception. The third depicts term “predicate”. Using predicates let us better communicate, so it’s very important for the process of study, and to improve in-group interaction and as a consequent to increase the result of command work in engineer practice.

НЛП (нейролингвистическое программирование) — это современный раздел психологии, исследующий область коммуникаций как межличностного, так и группового плана. В НЛП выделены различные модели стратегий успешной деятельности в области науки, техники, бизнеса, образования, психотерапии и др.

Для практического использования НЛП предоставляет набор моделей, умений и технологий, направленных на то, чтобы мыслить и действовать эффективно в предложенных обстоятельствах.

В НЛП широко используются лингвистические структуры в качестве межличностного кода. Для успешного освоения навыков мета моделирования необходима определенная база знаний, с помощью которой возможно быстро ориентироваться в понятиях НЛП, инструментах и моделях.

Целью проделанной работы было создание проблемно-ориентированной базы данных для поддержания интеллектуальной деятельности с использованием НЛП. Её структура включает три основных раздела. Первый содержит основные термины, необходимые для полноценного

использования методик теории НЛП для усвоения технических дисциплин, а также для дальнейшего изучения и применения подходов НЛП в творческой инженерной деятельности.

Во втором разделе приводятся пресуппозиции или аксиоматический аппарат НЛП. Приведенные здесь утверждения стимулируют творческое мышление, позволяя увидеть решаемую проблему в различных аспектах, через определенный фильтр восприятия.

Третий раздел посвящен такому понятию как сенсорноопределенные слова или предикаты. Этот термин применяют для обозначения слов (прилагательных, наречий, глаголов), имеющих определенную сенсорную окраску. Наиболее часто используются части речи визуального, аудиального и кинестетического характера, реже — обонятельного и вкусового. Например, приведем следующие фразы: “Рассмотрим следующий вопрос...”, “Обратимся к следующему вопросу ...”, “Затронем следующий вопрос...”. Смысл выражений не меняется, однако, предикаты используются разные. По сути дела, язык, которым пользуется человек, указывает на его предпочитаемую репрезентативную систему, т.е. на то каким образом он думает: либо он мыслит картинками, либо звуками, либо ощущениями. По этой причине, между людьми, использующими разные репрезентативные системы, а, как следствие, и разные предикаты, может возникать непонимание, а зачастую и споры по поводу значения этих слов. Интересно отметить, что в научной литературе авторы уходят от “личностного” языка и, в основном, используют нейтральные слова, не имеющие определенной сенсорной окраски, как например: “понимать”, “думать”, “объяснять”, “процесс”, “изучать” и т.д.

Первоначально, эти разработки основывались на наблюдениях в психологических группах и в дальнейшем были ориентированы на психотерапию. Однако их применение оказывается важным и необходимым также и в учебном процессе, и в инженерной практике. Метамодел — это уникальный способ передачи знаний и творческого опыта.

Использование определенных предикатов позволяет осуществлять хорошую коммуникацию, которая в свою очередь является одним из основных условий учебного процесса. Учитывая особенности языка собеседника, подстраиваясь к его предикатам и используя определенный способ мышления, можно достигнуть взаимопонимания, полноценного общения, при этом исключая появление каких-либо недоразумений в дальнейшем. Владея основами этой методики, преподаватель имеет возможность излагать материал в максимально доступной форме. Учитывая новейшие технические достижения, учебный процесс можно моделировать, ориентируясь на ту или иную, а чаще всего на все сразу репрезентативные системы. Такой подход можно реализовать предусмотрев раздаточный материал к курсу лекций в виде схем, таблиц, рисунков (рассчитано на визуалов); акцентируя ключевые фразы, еще раз проговаривая основные положения (рассчитано на аудиальную репрезентативную систему); используя учебные программы, фильмы, демонстрационные образцы, позволяющие, грубо говоря, “потрогать” или “осязать” теорию (рассчитано на кинестетиков). Интересные возможности представляет анализ работ студентов. Можно проследить определенную зависимость между темой проекта студента и его предпочитаемой репрезентативной системой. Анализ письменных работ может дать массу полезных для преподавателя сведений о студентах и позволит на этой основе полнее раскрыть их технические таланты. Результаты такого исследования помогут в дальнейшем ориентироваться при выборе того или иного направления исследовательских работ.

Что касается инженерной практики, то применение и умение определять предикаты в этой области, в первую очередь, необходимо для полноценной коммуникации. Также, учитывая репрезентативные системы и образ мышления сотрудников, можно более эффективно формировать команды для решения тех или иных задач. При этом улучшится взаимодействие внутри группы и, как следствие, результат совместной работы будет качественно выше.

Таким образом разработки НЛП находят широкое применение в творческой инженерной деятельности. Необходимость в структурировании: основных понятий,

исходных терминов, широко известных и апробированных моделей НЛП, стала причиной создания данной программы.

Эта программа рассчитана на студентов и преподавателей, желающих овладеть стратегиями, моделями и методами НЛП для передачи технического опыта, инженерных знаний и развития творческого мышления. В ее основе лежит древовидная структура, т.е. переход от крупных разделов к более мелким, а также перекрестная система ссылок между элементами базы данных, что позволяет организовать быстрый поиск необходимой информации.

Программа содержит две основные составляющие. Первая – сама база данных, реализованная средствами языков HTML и JavaScript, т.е. открытая широкому кругу пользователей через INTERNET, и вторая (являющаяся Windows – приложением написанным на DELPHI) предназначенная для редактирования и обновления базы данных пользователями не владеющими выше описанными языками. Такая организация программы приведет к всестороннему пополнению и обновлению существующей базы новыми технологиями современного инженерного образования.

Описанная структура программы существенно облегчит понимание базовых положений НЛП, а как следствие, будет служить основой к освоению, разработке и исследованию новых моделей образования в технических ВУЗах, что позволит развить и усилить творческий аспект в дальнейшей инженерной деятельности студентов.

Литература

1. Дж. О'Коннор, Дж. Сеймор “Введение в нейролингвистическое программирование”, Библиотека А. Миллера, 1998.