

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.В.ДВ.09.01 Диалоговые средства АСОИ

Направление подготовки (специальность) 09.03.01 Информатика и вычислительная техника

Профиль образовательной программы Автоматизированные системы обработки информации и управления

Форма обучения очная

1. КОНСПЕКТ ЛЕКЦИЙ

1.1 Лекция №1 (2 часа).

Тема: «Субъекты организаций как пользователи АСОИиУ»

1.1.1 Вопросы лекции:

1. Основные определения
2. Виды структурной организации АСОИУ
3. Классификация АСОИУ

1.1.2 Краткое содержание вопросов:

Основные определения

В настоящее время в теории и практике разработки и использования систем автоматизированной обработки информации обозначено много направлений:

1. отраслевые автоматизированные системы управления (АСУ в торговле, АСУ статистики, АСУ образовательных учреждений, АСУ финансовых учреждений и пр.),
2. автоматизированные системы управления территориями или регионами (например, АСУ-город),
3. автоматизированные системы управления предприятиями (АСУП),
4. автоматизированные системы управления производственными процессами (АСУПП),
5. автоматизированные системы управления технологическими процессами (АСУТП),
6. автоматизированные информационные системы (АИС) – под АИС в настоящее время понимаются системы, связанные с процессами обработки больших объемов информации специальных видов:
 - фактографической информации (справочные, библиографические и прочие системы),
 - информации, представленной в тестовых документах (документальные системы),
 - картографической информации,
 - баз знаний (экспертные системы) и прочих специальных видов данных.

Границы между перечисленными видами автоматизированных систем размыты, так как эти системы обладают общими составляющими и свойствами как с точки зрения их разработки, так и с точки зрения их эксплуатации.

Первые три вида автоматизированных систем являются системами организационного управления, в то время как остальные системы решают вопросы управления процессами производства изделий (АСУПП, АСУТП) и процессами обработки больших объемов информации (АИС). Как правило, АСУПП, АСУТП и АИС входят в системы организационного управления в качестве их составных частей. В дальнейшем мы основной упор будем делать на системы организационного управления.

В настоящее время средства автоматизации настолько органично вплетаются в процессы управления на объектах, в которых они используются, что составляют вместе с объектом управления единое и неразрывное целое. Иными словами, АСОИУ – это автоматизированное подразделение предприятия (если автоматизируется некоторое подразделение предприятия), автоматизированное предприятие в целом (если автоматизируется большинство функций на предприятии), автоматизированный регион или отрасль (если автоматизируются функции управления регионом или отраслью). Такие

автоматизированные системы обработки информации и управления мы будем называть организационными АСОИУ или просто *организационными системами*.

В процессе проектирования *организационная система* на современном этапе рассматривается как некоторая человеческая система, то есть такая система, в которую в качестве ее элементов, кроме всего прочего, входят люди, выполняющие определенные ее функции, решающие в связи с этим определенные задачи и составляющие собою совокупный или, как еще говорят, соборный интеллект системы. К организационным системам применяются те же положения теории управления, которые используются в технических системах, однако в данном случае есть и свои особенности.

Виды структурной организации АСОИУ

АСОИУ, как правило, являются иерархическими системами, то есть имеют определенную иерархическую структуру. Например, если в качестве АСОИУ выступает автоматизированное предприятие, то в нем имеются различные отделы и подразделения, в которых обязательно решаются определенные управленческие задачи. Если в качестве АСОИУ выступает холдинг, объединяющий несколько организаций, то эти организации являются элементами иерархической структуры холдинга и т. п. Очевидно, что каждый элемент иерархической структуры АСОИУ как-то связан с другими ее элементами. Эти связи могут быть разных типов. Одним из типов этих связей является подчиненность. Причем подчиненность элемента другому элементу структуры, расположенному на верхнем уровне иерархии, может иметь разную силу в зависимости от количества отдельных функций ПФУ, которые выполняет данный элемент в процессе своей деятельности.

Структуры, у которых каждый элемент нижележащего уровня подчинен только одному элементу, расположенному на более верхнем уровне, называются *структурами типа дерева*. Такие структуры являются структурами с *сильными связями*. Если элементы нижележащего уровня подчиняются не одному, а нескольким элементам вышележащего уровня, то такие структуры являются структурами со *слабыми связями*. На практике важное место занимают многоуровневые иерархические структуры

Классификация АСОИУ.

Классификация АСОИУ осуществляется по разным признакам в зависимости от решаемой задачи. Классификации всегда относительны. Например, по виду управляемого объекта АСОИУ подразделяются на технические (АСУПП, АСУТП), информационные (АИС) и организационные. Однако в технических АСОИУ при ближайшем рассмотрении решается много и организационных вопросов, так как в процессе управления такими АСОИУ (в особенности АСУПП) непосредственное участие принимает человек. Кроме того, в них могут использоваться и ручные операции, которые либо трудно, либо не имеет смысла автоматизировать. Много организационных вопросов присуще и АИС, кроме того, в любой организационной АСОИУ часто возникает необходимость в обработке больших объемов информации, например, при создании и сопровождении нормативно-справочной документации или при обработке входящей и исходящей корреспонденции.

1. *Открытые и закрытые системы.* АСОИУ является открытой системой, если она обменивается информацией с внешней средой. В противном случае АСОИУ является закрытой системой. Конечно, на самом деле абсолютно закрытых АСОИУ не существует. Любая система в той или иной мере в своей работе пользуется информацией извне, например, для организации обратных связей в технических системах с целью обеспечения

устойчивости. Вопрос заключается в характере использования информации извне. Если система способна сохранять свой высокий уровень и даже развиваться, заменяя в случае необходимости свои цели управления, совершенствуя свою структуру и свою продукцию в ответ на меняющуюся внешнюю обстановку, то такая система, безусловно, является открытой.

2. *Хорошо организованные системы.* АСОИУ является хорошо организованной системой, если при ее создании удалось определить все элементы системы и их взаимосвязи между собой, а также взаимосвязи с целями системы в виде детерминированных аналитических, алгоритмических и графических зависимостей, то есть удалось определить детерминированную модель объекта управления и его внешней среды. Как правило, это удается сделать только в технических АСОИУ. Однако любое моделирование систем осуществляется с определенной точностью. При описании поведения сложного объекта в виде какой-то схемы или системы уравнений приходится выделять существенные и не учитывать несущественные компоненты этого объекта. Например, показания датчиков при управлении некоторым технологическим процессом верны с определенной точностью, с определенной точностью осуществляется обработка управляющих воздействий в системе и прочее.

В хорошо организованных системах задачи выбора целей и определения средств их достижения не разделяются. Поведение объекта управления и внешней среды могут быть описаны выражениями, связывающими цель со средствами в виде целевой функции, которая может быть представлена сложным уравнением, формулой, системой уравнений или неравенств. Представление объекта управления в виде хорошо организованной системы применяется в тех случаях, когда можно предложить детерминированное описание (модель) объекта и экспериментально доказать правомерность применения этой модели к данному реальному объекту. Попытки создать АСОИУ для предприятия или организации в виде хорошо организованной системы практически безрезультатны в силу большой сложности процессов в организационных системах и неопределенности, в которой приходится управлять такими системами.

3. *Плохо организованные (диффузные) системы.* АСОИУ является плохо организованной (диффузной) системой, когда не ставится задача детерминированно определить все компоненты и их связи с целями системы. Объект управления и внешняя среда в этом случае описываются закономерностями, которые выявляются на основе исследования не всего объекта, а путем статистического изучения представительных выборок компонентов, характеризующих этот объект. Например, при решении задачи определения числа автобусов, необходимых на заданном маршруте, на всех остановках маршрута в разное время суток в течение нескольких дней и в разное время года собирается информация о количестве пассажиров, ждущих автобус данного маршрута. После статистической обработки собранной информации получают различные усредненные оценки численности пассажиров на данном маршруте, а также вероятности достоверности полученных оценок.

Моделирование объектов управления в виде диффузных систем находит широкое применение при определении пропускной способности систем разного рода, при определении численности штатов в обслуживающих (ремонтных) подразделениях предприятия или в обслуживающих население учреждениях (в магазинах, на почте, в банках), при исследовании документальных потоков информации и прочее.

4. *Самоорганизующиеся системы.* Построение АСОИУ в виде самоорганизующихся систем позволяет решать задачи управления в обстановке большой неопределенности.

Класс самоорганизующихся или развивающихся АСОИУ характеризуется рядом особенностей. *Особенности самоорганизующихся АСОИУ* обусловлены наличием в

системе активных элементов (то есть, людей), которые, с одной стороны, полезны для существования системы в целом, так как обеспечивают приспособляемость системы к изменяющейся внешней среде, а с другой стороны, затрудняют управление системой, так как имеют свои собственные цели, которые могут не совпадать с целями системы в целом. Эти особенности заключаются в следующем:

а) *нестационарность (изменчивость, нестабильность) отдельных параметров и стохастичность поведения;*

б) *уникальность и непредсказуемость поведения системы* в конкретных условиях, благодаря наличию в системе активных элементов, обладающих свободой воли;

в) *способность адаптироваться к изменяющимся условиям среды и помехам*, что является полезным свойством, однако, система способна адаптироваться и по отношению к управляющим воздействиям, что весьма затрудняет управление;

г) *способность противостоять разрушительным тенденциям*, которые характерны для закрытых систем, не имеющих активных элементов;

д) *способность вырабатывать новые варианты поведения и изменять структуру системы*, сохраняя при этом ее целостность и основные свойства;

е) *способность и стремление к целобразованию* в отличие от закрытых (технических) систем, которым цели задаются извне.

1.2 Лекция №2 (2 часа).

Тема: «Классификации пользователей АСОИиУ»

1.2.1 Вопросы лекции:

1. Категории пользователей

1.2.2 Краткое содержание вопросов

Категории пользователей.

Классификации пользователей интерактивных систем. Типы операторской деятельности. Этапы деятельности человека-оператора. Типы поведения человека-оператора. Типы моделей, используемых оператором.

В специальной литературе, посвященной данному вопросу, приводится не-сколько вариантов классификации пользователей интерактивных систем. Но существующие подходы различаются в деталях, сохраняя единые принци-пы такой классификации.

В основу подхода положены факторы, влияющие на отношение пользователя к интерактивной системе и на методы его работы с нею.

- человеческие потребности означают, в частности, потребность быть по-нятым партнером по диалогу;

- навыки пользователя состоят из моторных навыков (относящихся прежде всего к работе с клавиатурой и мышью), лингвистических навыков, навыков в общении и навыков в решении задач;

- свойства личности - это, например, творческие способности, подвер-женность ошибкам, способность к обучению, терпеливость, устойчивость к стрессу и т.д.;

- уровень компьютерной грамотности. Здесь обычно различают програм-мирующих и не программирующих пользователей, а среди последних, в свою очередь, выделяют три категории:

- подготовленные пользователи, решающие творческие задачи, ана-литики и исследователи (то есть пользователи, последовательность дей-ствий которых сложно формализовать);

- подготовленные пользователи, выполняющие рутинные операции,- операторы (то есть пользователи, последовательность действий которых является достаточно устойчивой);

- случайные (или «наивные») пользователи, обладающие минимальным уровнем компьютерной грамотности.

Подготовка в прикладной областипользователя влияет на использование языка (например, профессиональной терминологии) и применяемые методы решения задач.

Причина пользования системойможет быть:

- обязательной - как неотъемлемая часть работы;

- необязательной - как дополнительная составляющая профессиональ-ной деятельности;

- обязательной с точки зрения личных потребностей (например, когда определенную информацию можно получить только с помощью компь-ютера);

- необязательной в частной жизни (например, в качестве развлечения).

Отношение к системе и ожидания от работы с ней определяются уровнем компьютерной грамотности и причиной ее использования; оно может быть:

- нейтральным: пользователь воспринимает компьютер только как ра-бочий инструмент;

- положительным: пользователь охотно использует компьютер, так как надеется получить от него пользу (или удовольствие);

- негативным: пользователь предпочел бы не пользоваться компь-ютером (возможно, из-за субъективной нелюбви или недоверия к тех-нике).

Целями пользователя могут быть:

- решить определенную задачу с помощью компьютера (в том числе и развлекательного характера);

- научиться работать с системой.

Ограничения по времени независимо от характеристик системы (в час-тности, ее быстродействия), в этом случае пользователь может быть вынужден приспособливаться к ограничениям по времени, исходящим от задачи или контек-ста работы.

В зависимости от уровня иерархии управления, на котором на-ходится человек-оператор, и типа системы управления (или обработ-ки информации) можно выделить следующие типы операторской дея-тельности:

оператор - технолог, непосредственно включенный в технологи-ческий процесс и выполняющий стандартные процедуры управления;

- оператор - манипулятор, реализующий процедуры управления, со-стоящие из механических воздействий;

- оператор - наблюдатель, следящий за состоянием процесса и его отклонениями в системах реального времени;

- оператор - исследователь, действия которого не регламентиро-ваны имеющимися процедурами и базируются на понятийном мышлении (программист);

- оператор - руководитель, выполняющий функции организационного и директивного характера;

- оператор - проектировщик, включенный в процесс машинного про-ектирования в составе САПР.

Организация взаимодействия технической (или вычи-слительной) системы с операторами различных категорий имеет свои специфические особенности. Однако существует ряд общих положе-ний, рассматриваемых далее и свойственных всем (или большинству) типам операторов (например, общая модель мышления и принятия ре-шений человеком, его психофизиологические характеристики и др.).

Деятельность человека-оператора.

В процессе решения задач выделяется 5 этапов, характерных практически для всех категорий операторов:

обнаружение - восприятие оператором группы признаков, ини-цирующих решение задачи;

- классификация - распознавание ситуации и определение состо-яния контролируемого процесса и технической системы;

- планирование - принятие решения и разработка последователь-ности действий, которые предположительно приведут к решению за-дачи;

- исполнение - реализация планов с учетом имеющихся ресурсов;

- отслеживание - проверка исполнения действий и их эффектив-ности.

Задачи, решаемые оператором, в свою очередь, делятся на 3 категории:

- регулирование процесса, предусматривающее незначительное динамическое выравнивание и корректировку отдельных параметров процесса;

- изменение режима, ведущее к принципиальному изменению про-цесса и предусматривающее переходные процессы, большое количест-во операций и сложные алгоритмы действий оператора;

деятельность в аномальных ситуациях, проявляющихся в выходе параметров процесса за установленные границы.

Типы поведения человека-оператора.

В зависимости от сложности и формализованности задач, выделяется 3 уровня поведения оператора:

- поведение, основанное на навыках - действия оператора рефлексивны и оттренированы, от него не требуется вникать в суть происходящих процессов, он должен лишь реагировать заранее определенным способом на возникновение некоторых ситуаций;
- поведение, основанное на правилах - деятельность оператора связана с использованием правил управления, сведенных в инструкции по эксплуатации системы;
- поведение, основанное на знаниях - действия оператора требуют понимания протекающих процессов и обработки информации, являющейся не содержащейся в текущих данных или в заранее подготовленных инструкциях.

Типы моделей, используемых оператором

В процессе решения задач оператор может использовать 5 типов моделей, формализующих как его действия, так и описание контролируемых процессов:

- физическую, содержащую математическое описание процессов, протекающих в системе;
- функциональную, описывающую основные подсистемы, оборудование, его функции и взаимосвязи;
- экономическую, связывающую величины стоимости основных операций управления;
- процедуральную, описывающую правила управления установкой;
- когнитивную, отражающую интуитивную модель процесса, формируемую оператором.

1.3 Лекция №3 (2 часа).

Тема: «Концепции разработки АСОИиУ с точки зрения учета роли КП в системе»

1.3.1 Вопросы лекции:

1. Управление процессорами
2. Управление памятью
3. Управление устройствами

1.3.2 Краткое содержание вопросов:

Управление процессорами

- Основные задачи управления процессором сводятся к решению двух взаимосвязанных проблем:
 - Создание условий, при которых каждый процесс и приложение получают достаточную часть рабочего времени процессора, чтобы обеспечивалось их нормальное функционирование
 - Использование столько циклов процессора, сколько возможно для нормальной работы.
- Основной единицей программного обеспечения, с которой операционная система

работает при планировании работы процессора, является либо процесс, либо поток, в зависимости от операционной системы.

Было бы заманчиво рассматривать процесс как приложение, однако такой подход дает неполную картину того, какая устанавливается взаимосвязь процессов с операционной системой и аппаратными средствами. Видимое пользователем приложение (текстовый редактор, электронная таблица или игра) действительно является процессом, однако это приложение может инициировать запуск некоторых других процессов для решения таких задач, как связь с другими устройствами или компьютерами. Имеется также большое число процессов, которые протекают, не проявляя себя. Например, в Windows XP и UNIX могут быть десятки фоновых процессов, предназначенных для управления сетью, памятью и дисками, проверки на наличие вирусов и т.д.

Таким образом, процесс – это программа, выполняющая определенное действие, и которой можно управлять – силами пользователя, с помощью других приложений или с помощью операционной системы.

Операционная система осуществляет контроль и планирует выполнение центральным процессором процессов, а не приложений. В однозадачной системе планирование выполнения простое. Операционная система разрешает приложению запуститься, временно приостанавливая его выполнение на достаточно длительное время лишь в случае необходимости обслуживания прерываний и пользовательского ввода данных. Прерывания – специальные сигналы, отправляемые на центральный процессор аппаратными средствами или программами. Это похоже на то, как если бы во время оживленного собрания какая-то часть компьютера вдруг подняла руку, требуя к себе внимания центрального процессора. Иногда операционная система устанавливает приоритеты процессов таким образом, что прерывания маскируются, то есть операционная система игнорирует прерывания от некоторых источников, чтобы определенная операция была завершена как можно скорее. Существуют некоторые прерывания (например, вызванные состоянием ошибки или проблемами с памятью), которые настолько важны, что их нельзя игнорировать. Эти немаскируемые прерывания (non-maskable interrupts, NMIs) требуют немедленного решения проблемы, несмотря на то, что должны выполняться другие задачи.

Учитывая, что прерывания создают определенные сложности при выполнении процессов даже в однозадачной системе, функционирование операционной системы становится намного более сложным в многозадачной системе. В последнем случае операционная система должна организовать выполнение приложений таким образом, чтобы создавалось впечатление, что определенные события происходят одновременно. Это сложно осуществить, поскольку центральный процессор в каждый момент времени может делать только одну операцию. Современные многоядерные процессоры и многопроцессорные компьютеры могут выполнять по несколько операций одновременно, однако каждое ядро процессора, как и прежде, в каждый момент времени может делать только одну операцию.

Чтобы создавалось впечатление, что множество событий происходит одновременно, операционная система должна осуществлять переключение между разными процессами тысячи раз в секунду. Это делается следующим образом:

- Процесс занимает определенную часть оперативной памяти. Кроме того, он использует регистры, стеки и очереди в центральном процессоре, а также в пространстве памяти операционной системы.

- Допустим, имеется два многозадачных процесса. Операционная система выделяет на каждую программу по определенному количеству исполнительных циклов.

- После прохождения этого количества циклов операционная система делает копии всех регистров, стеков и очередей, использовавшихся в процессах, и отмечает место, на котором наступила пауза выполнения процесса.

– Затем производится загрузка всех регистров, стеков и очередей, используемых вторым процессом, и этому процессу разрешается прохождение определенного количества циклов центрального компьютера.

– По завершении этих циклов делаются копии всех регистров, стеков и очередей, использовавшихся второй программой, и производится загрузка первой программы.

Управление памятью

Под памятью (*memory*) в данном случае подразумевается оперативная (основная) память компьютера. В однопрограммных операционных системах основная память разделяется на две части. Одна часть для операционной системы (резидентный монитор, ядро), а вторая – для выполняющейся в текущий момент времени программы. В многопрограммных ОС "пользовательская" часть памяти – важнейший ресурс вычислительной системы – должна быть распределена для размещения нескольких процессов, в том числе процессов ОС. Эта задача распределения выполняется операционной системой динамически специальной подсистемой управления памятью (*memory management*). Эффективное управление памятью жизненно важно для многозадачных систем. Если в памяти будет находиться небольшое число процессов, то значительную часть времени процессы будут находиться в состоянии ожидания ввода-вывода и загрузка процессора будет низкой.

В ранних ОС управление памятью сводилось просто к загрузке программы и ее данных из некоторого внешнего накопителя (перфоленты, магнитной ленты или магнитного диска) в ОЗУ. При этом память разделялась между программой и ОС.

С появлением мультипрограммирования задачи ОС, связанные с распределением имеющейся памяти между несколькими одновременно выполняющимися программами, существенно усложнились.

Функциями ОС по управлению памятью в мультипрограммных системах являются:

- отслеживание (учет) свободной и занятой памяти;
- первоначальное и *динамическое выделение памяти* процессам приложений и самой операционной системе и освобождение памяти по завершении процессов;
- настройка адресов программы на конкретную область физической памяти;
- полное или частичное вытеснение кодов и данных процессов из ОП на диск, когда размеры ОП недостаточны для размещения всех процессов, и возвращение их в ОП;
- защита памяти, выделенной процессу, от возможных вмешательств со стороны других процессов;
- *дефрагментация памяти*.

Управление устройствами

Рассмотрим схематично принцип управления внешними устройствами, а затем перейдем к изучению соответствующих структур и функций WRK.

Для пользовательских приложений *операционная система* представляет устройства в виде файлов. Такое *представление* позволяет единообразно работать с разными устройствами, используя одинаковые функции, не задумываясь о деталях реализации доступа к устройствам.

Файл (file) – совокупность данных, имеющих имя и допускающих *операции* чтения-записи. Типичная последовательность работы с файлом: *открытие файла*, выполнение команд чтения-записи, *заккрытие файла*.

При открытии файла создается файловый *объект* типа FILE_OBJECT, который связан с объектом, представляющим конкретное устройство (DEVICE_OBJECT). В объекте-устройстве содержится *информация* о драйвере, который управляет этим устройством. *Драйвер* в системе описывается объектом типа DRIVER_OBJECT. Объекты DRIVER_OBJECT создаются при загрузке в систему нового драйвера. Затем *объект* DRIVER_OBJECT может создать несколько объектов DEVICE_OBJECT – по количеству управляемых драйвером устройств ([рис.15.1](#)).

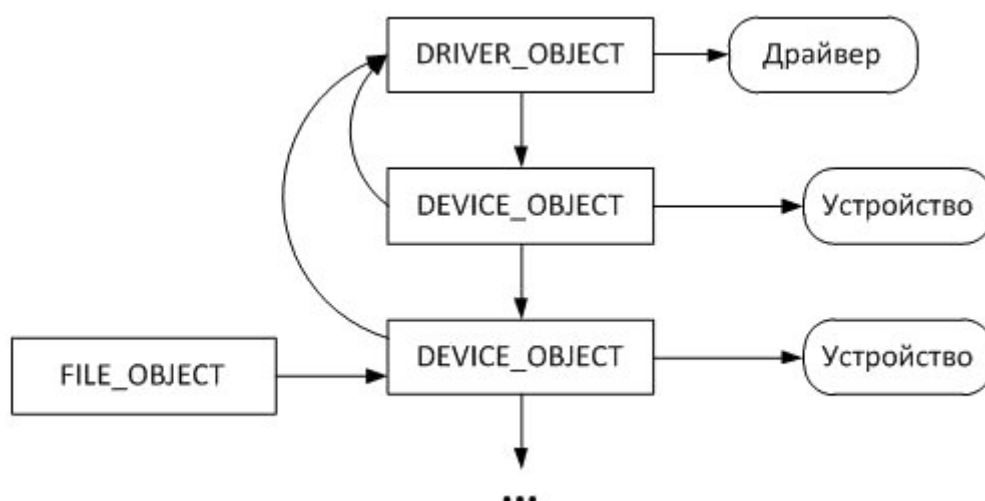


Рис. 15.1. Объекты для управления вводом-выводом

Как видно из рис.15.1, в объекте DRIVER_OBJECT содержится указатель на список объектов-устройств, а в каждом из этих объектов хранится ссылка на управляющий драйвер. Таким образом, имея информацию об объекте DRIVER_OBJECT, можно найти все устройства, которыми он управляет и, наоборот, по объекту DEVICE_OBJECT легко определяется драйвер устройства.

Приложение, которому необходимо произвести некоторую операцию с устройством (файлом), вызывает соответствующую WinAPI функцию (CreateFile, ReadFile, WriteFile и др.), которая, в свою очередь, обращается к функции диспетчера ввода-вывода.

Операция, которая запрашивается приложением, представляется в системе объектом типа IRP (I/O Request Packet – пакет запроса на ввод/вывод). В этом объекте

хранится информация о типе операции ввода/вывода (создание, чтение, запись и т. п.), а также необходимые параметры для данной операции. Пакет IRP передается диспетчером ввода-вывода в очередь IRP потока, который запросил операцию ввода-вывода, после чего вызывается соответствующий драйвер, непосредственно выполняющий запрошенную операцию.

Структуры данных для ввода-вывода

Драйвер в системе описывается объектом типа DRIVER_OBJECT (*файл* base\ntos\inc\io.h, строка 1603), имеющим следующие основные поля:

- Type – поле, определяющее тип структуры подсистемы ввода-вывода. Значения этого поля могут быть следующими –IO_TYPE_DRIVER, IO_TYPE_FILE, IO_TYPE_DEVICE, IO_TYPE_IRP и др. (см. файл base\ntos\inc\io.h, строка 25);
- Size – размер объекта в байтах;
- DeviceObject – ссылка на первый объект DEVICE_OBJECT в списке устройств, управляемых данным драйвером (см. [рис.15.1](#)). Следующие устройства в списке можно определять по полю NextDevice объекта DEVICE_OBJECT;
- Flags – флаги, определяющие тип драйвера (см. файл base\ntos\inc\io.h, строка 1530);
- DriverName – имя драйвера в системе;
- HardwareDatabase – путь в реестре к информации о драйвере;
- DriverStart, DriverSize, DriverSection – информация о расположении драйвера в памяти;
- DriverInit – адрес процедуры DriverEntry (точка входа в драйвер), отвечающей за инициализацию драйвера;
- DriverUnload – адрес процедуры выгрузки драйвера;
- MajorFunction – массив адресов процедур, каждая из которых отвечает за определенную операцию с устройством. Максимальное количество таких процедур равно константе IRP_MJ_MAXIMUM_FUNCTION+ 1 = 28 (файл base\ntos\inc\io.h, строка 80), которая определяет также количество кодов IRP (см. далее).

Устройства представлены объектами типа DEVICE_OBJECT, который включает следующие главные поля (*файл* base\ntos\inc\io.h, строка 1397):

- Type, Size – совпадают по назначению с полями типа DRIVER_OBJECT;
- ReferenceCount – счетчик количества открытых дескрипторов для устройства. Позволяет отслеживать, используется кем-либо устройство или нет;
- DriverObject – ссылка на драйвер, который управляет устройством;
- NextDevice – указатель на следующее устройство в списке устройств для данного драйвера;
- Flags, Characteristics – поля, уточняющие характеристики устройства;
- DeviceType – тип устройства; возможные типы перечислены в файле public\sdk\inc\devioctl.h (строка 26);
- SecurityDescriptor – дескриптор безопасности, сопоставленный с устройством (см. лекцию 9 "Безопасность в Windows").

Пакеты запроса на ввод-вывод описываются типом IRP (*I/O Request Packet*), состоящим из двух частей – заголовка фиксированной длины (тело IRP) и одного или нескольких блоков

стека. В заголовке описывается *информация*, общая для запроса. Каждый блок стека содержит данные об одной *операции* ввода-вывода.

Заголовок включает следующие основные поля:

- Type, Size – поля, по назначению аналогичные соответствующим полям типов DRIVER_OBJECT и DEVICE_OBJECT;
- IoStatus – статус операции при завершении;
- RequestorMode – режим, в котором работает поток, инициировавший операцию ввода-вывода, – пользовательский или режим ядра;
- StackCount – количество блоков стека;
- Tail.Overlay.Thread – указатель на структуру ETHREAD потока, запросившего операцию ввода-вывода;
- Tail.Overlay.CurrentStackLocation – указатель на блок стека (IRP Stack Location), который описывается структурой IO_STACK_LOCATION.

Структура блока стека IO_STACK_LOCATION описана в файле base\ntos\inc\io.h, строка 2303) и имеет следующие главные поля:

- MajorFunction – номер основной функции, определяющий запрошенную операцию ввода-вывода и совпадающий с номером функции драйвера в массиве MajorFunction (структура DRIVER_OBJECT, см. выше), которую нужно вызвать для выполнения запрошенной операции. Как уже отмечалось, всего кодов 28 (IRP_MJ_MAXIMUM_FUNCTION + 1), они описаны в файле base\ntos\inc\io.h (строки 51–79);
- DeviceObject – указатель на структуру DEVICE_OBJECT, определяющую устройство для данной операции ввода-вывода;
- FileObject – указатель на структуру FILE_OBJECT (файл base\ntos\inc\io.h, строка 1763), которая ассоциирована со структурой DEVICE_OBJECT.

1.4 Лекция №4 (2 часа).

Тема: «Структурные свойства диалога»

1.4.1 Вопросы лекции:

1. Функции управления
2. Критерии оценки АСОИ

1.4.2 Краткое содержание вопросов:

Функции управления

Объект управления – это часть системы, которой управляют, а управляющий орган (субъект управления) – это та часть системы, которая осуществляет управление.

Внешняя среда – это окружающая обстановка, в которой система существует. Модель объекта управления и модель внешней среды – это совокупность представлений о поведении объекта управления и внешней среды, которыми пользуется управляющий орган системы. В организационных системах этим о деличастично представляются в виде формальных описаний (например, в виде совокупности математических формул или в виде алгоритмических описаний), частично – в виде обычного текста или в виде

интуитивных представлений персонала системы на сознательном и бессознательном уровнях, которые, тем не менее, используются в процессе управления.

Вектор целей управления – иерархически упорядоченная совокупность целей, которых система стремится достичь в процессе своей деятельности. Отдельные составляющие этого вектора называются частными целями. Упорядоченность частных целей в векторе целей отражает их приоритет. Порядок, обратный приоритетному, – это порядок вынужденного отказа от достижения частных целей в случае невозможности достижения их полной совокупности. Каждая из частных целей оценивается параметром, определяющим степень достижения этой цели.

Например, в качестве цели может выступать стремление получить максимально возможный доход фирмы за определенный период времени, тогда в качестве параметра для оценки степени достижения этой цели выступает величина этого дохода. В общем случае параметр для оценки степени достижения цели может представлять собой более сложную информационную конструкцию, нежели простое число. Например, для оценки качества производимого фирмой товара может выступать удовлетворенность покупателей этим товаром. Оценка степени удовлетворенности покупателей товаром, как правило, выливается в отдельную достаточно сложную задачу.

Необходимо заметить, что приоритет частных целей в векторе целей является существенным фактором, который кардинально влияет на деятельность системы. Например, если получение максимума прибыли фирмы выбрать в качестве цели с наивысшим приоритетом (прибыль не смотря ни на что), то это приведет к иному характеру деятельности фирмы, может быть даже где-то к криминальному ее характеру. Выбор же в качестве цели с максимальным приоритетом получение максимума прибыли при соблюдении законности деятельности фирмы приведет совершенно к другому характеру деятельности.

Вектор контрольных параметров – иерархически упорядоченная совокупность параметров, предназначенных для оценки степени достижения частных целей вектора целей системы. Порядок значений параметров в этом векторе отражает порядок соответствующих частных целей в векторе целей. Значения контрольных параметров, воспринимаемые субъектом управления, описывают ее реальное поведение. Но поскольку восприятие реального состояния системы не идеально, то всегда присутствует некоторая ошибка во пределинии истинного состояния, которая может быть как допустимой, так и недопустимой для целей управления системой в данном конкретном случае. Контрольные параметры часто называются так же критериями управления.

Вектор ошибки управления – это расхождение между желаемыми и реальными значениями в векторе контрольных параметров. Вектор ошибки является основой для формирования оценки качества управления в системе.

Вектор управляющих воздействий (параметров) – это совокупность параметров, характеризующих управляющие воздействия на объект управления, а так же на внешнюю среду системы. Значения управляющих параметров в системе меняются непосредственно по произволу субъекта управления. Например, процент отчислений на заработную плату работникам торгового отдела от выручки проданного товара в торговой фирме или величина премий работникам фирмы за прошедший месяц, квартал или год. Список управляющих параметров системы может быть иерархически упорядоченным (управление

в нормальных условиях, управление в опасных обстоятельствах, управление в аварийной обстановке и т.п.).

В модели объекта управления системы наряду с указанными выше контрольными параметрами и параметрами вектора управляющих воздействий могут присутствовать и ряд других параметров, характеризующих состояние системы. Они называются свободными параметрами. Состав свободных параметров системы так же не постоянен и может меняться в процессе управления в результате изменений в системе представлений об объекте управления и внешней среды. Значения свободных параметров лежат в объективно или субъективно обусловленных множествах значений и могут зависеть от поведения внешней среды. Совокупность параметров всех указанных типов образуют вектор состояния системы.

В отличие от технических систем в организационных системах, как правило, не удается так регламентировать управление системой, чтобы добиться постоянства состава всех векторов, перечисленных выше. Это происходит из-за сложности объекта управления и его взаимодействия с внешней средой, а так же из-за объективно существующей неопределенности в поведении элементов системы и внешней среды. По этой причине состав всех перечисленных векторов, в том числе и вектора состояния системы, в процессе управления может меняться.

Устойчивость системы по предсказуемости—это свойство системы, которое заключается в том, что ее поведение под воздействием внешней среды, внутренних изменений и управления в определенной мере предсказуемо. Если система не обладает устойчивостью по предсказуемости, то, очевидно, управление ею невозможно. Например, не в полне опытный шофер, въехав на скользкий участок дороги и передвигаясь по нему, может резко затормозить, ожидая при этом реакции автомобиля как приезде по сухому асфальту. Автомобиль при этом теряет управляемость, его заносит в сторону, и он переворачивается. На лицо неустойчивость системы по предсказуемости в изменившихся условиях внешней среды. Шофер подсознательно ожидал (предсказывал) совсем другое поведение автомобиля, нежели то, которое получилось. Таким образом, устойчивость системы по предсказуемости определяется адекватностью заложенной в нее модели объекта управления и внешней среды (то есть, представлений об их поведении) реальному их поведению.

Аналогичная ситуация возникает и при управлении различными предприятиями и организациями, когда резко изменяется внешняя политическая и экономическая обстановка. В нашей стране в результате смены политического режима и галопирующей инфляции все экономические субъекты попали и в условия, когда прежние, привычные представления оказались неадекватными действительности. Руководство большинства предприятий и организаций страны несумело во время перестроить свои представления и стереотипы поведения в новой экономической и политической обстановке. В результате появилась неустойчивость поведения всех экономических систем по предсказуемости и как следствие—банкротство и развал многих предприятий и организаций.

Традиционно в технических системах устойчивость определяется как способность системы возвращаться в исходный режим существования по всем или по части параметров после того, как внешнее возмущающее воздействие на систему исчезло. Устойчивость по предсказуемости более общее понятие. Действительно, если система устойчива в классическом смысле, то это означает, что заложенная в нее модель объекта управления и внешней среды в достаточной мере адекватна действительности, так как

управление на ее основе возвращает систему в прежнее состояние после исчезновения возмущающего воздействия. Следовательно, такая система устойчива и по предсказуемости.

Полная функция управления (ПФУ) системы—это деятельность, охватывающая все необходимые вопросы по управлению системой. В отличие от технических систем, которые функционируют и должны функционировать в регламентированных, заранее оговоренных при их разработке условиях, организационные системы напротив существуют во внешней среде, которая может изменяться настолько, что эти изменения порой могут привести к необходимости кардинальной перестройки управления системой.

Например, в какое-то время руководство некоторой фирмы, занимающейся производством колбасных изделий, узнает, что основной ее конкурент установил у себя новую технологическую линию, которая обеспечивает значительное повышение производительности производства основных сортов колбасы при снижении потерь сырья и при повышенном качестве продукции. Это, конечно, приведет к снижению себе стоимости продукции и как следствие откроет перед конкурентом широкие возможности завоевания тех секторов рынка, на которые сориентирована данная фирма. В ответ на этот новый фактор внешней среды руководство фирмы решает организовать специальную группу, которая должна изучить проблему и выработать предложения по необходимой реорганизации производства и сбыта продукции.

1.5 Лекция №5 (2 часа).

Тема: «Поддержка КП в диалоге»

1.5.1 Вопросы лекции:

1.Выбор структуры организации

1.5.2 Краткое содержание вопросов

Организационное проектирование

Важнейший начальный этап создания любой организации или подразделения.

В процессе организационного проектирования общепринято выделять три стадии: предпроектная подготовка, техническое проектирование, внедрение организационного проекта.

Предпроектная подготовка

Предпроектная подготовка включает три этапа: организационную подготовку, проведение предпроектного обследования системы и разработку технического задания.

Техническое проектирование

Технический проект (ТП) системы организации производства, труда и управления предприятия разрабатывается на основании утвержденного технического задания. На этапе технического проектирования вырабатываются общие решения о построении

(совершенствовании или преобразовании) системы организации производства, труда и управления.

Внедрение организационного проекта

Внедрение организационного проекта проводится на основании материалов по реализации рабочего проекта организации производства, содержащихся в комплекте рабочей документации, в два этапа:

- подготовка к внедрению;
- внедрение.

1.6 Лекция №6 (2 часа).

Тема: «Структура диалога»

1.6.1. Вопросы лекции:

1. Мотивации, как функция управления.
2. Необходимость контроля. Этапы контроля.

1.6.2 Краткое содержание вопросов:

Мотивация, как функция управления.

Итак, информация о будущей системе собрана. Теперь, пользуясь этой информацией, необходимо создать общую структуру системы («вид с высоты птичьего полета»), т.е. выделить отдельные функциональные блоки и определить, как именно эти блоки связываются между собой. Под отдельным функциональным блоком будем понимать функцию/группу функций, связанных по назначению или области применения в случае программы и группу функций/фрагментов информационного наполнения в случае сайта.

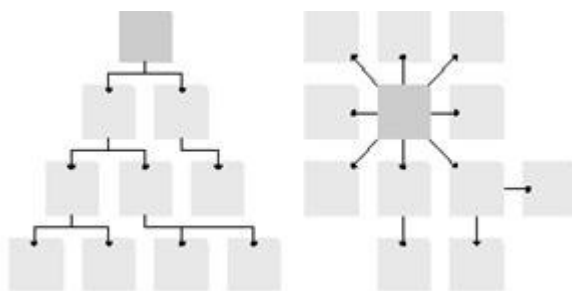


Рис.1. Типичная структура сайта (слева) и типичная структура программы. Если сайт обычно разветвлен, то в смысле, что функции обычно размещаются в отдельных экранах, то программы обычно имеют только один изменяющийся экран, в котором вызываются почти все функции.

Проектирование общей структуры состоит из двух параллельно происходящих процессов: выделения независимых блоков и определения связи между ними. Если проектируется сайт, в завершении необходимо также создать схему навигации.

Рис. 1. Типичная структура сайта (слева) и типичная структура программы. Если сайты обычно разветвлены, в том смысле, что функции обычно размещаются в отдельных экранах, то программы обычно имеют только один изменяющийся экран, в котором и вызываются почти все функции.

Проектирование общей структуры состоит из двух параллельно происходящих процессов: выделения независимых блоков и определения связи между ними. Если проектируется сайт, в завершении необходимо также создать схему навигации.

Выделение независимых блоков

Для этой работы трудно дать какие-либо конкретные рекомендации, поскольку очень многое зависит от проектируемой системы. Тем не менее, можно с уверенностью рекомендовать избегать помещения в один блок более трех функций, поскольку каждый блок в результирующей системе будет заключен в отдельный экран или группу управляющих элементов. Перегружать же интерфейс опасно.

Результатом этой работы должен быть список блоков с необходимыми пояснениями. В качестве примера рассмотрим гипотетическую программу ввода данных. От пользователя требуется выбрать из списка клиента (или добавить в список нового) и указать, какие именно товары клиент заказал (товары в список тоже можно добавлять). Несколько клиентов постоянно что-то заказывают, так что заставлять пользователя каждый раз искать в списке такого клиента неправильно. При этом блоки разделяются следующим образом:

Основной экран	Навигация между функциями системы
Создание нового заказа	
Добавление существующего товара в заказ	А так же простой поиск товара в списке

Сложный поиск товара	
Добавление нового товара в список	
Добавление существующего клиента в заказ	А так же простой поиск клиента в списке
Добавление нового клиента в список	
Выбор постоянного клиента	
Обработка заказа	Печать и его переход в папку Исполняемые

Определение смысловой связи между блоками

Существует три основных вида связи между блоками. Это *логическая* связь, связь *по представлению пользователей* и *процессуальная* связь. Логическая связь определяет взаимодействие между фрагментами системы с точки зрения разработчика (суперпользователя). Пользователи имеют свое мнение о системе, и это мнение тоже является важным видом связи. Наконец, процессуальная связь описывает пусть не вполне логичное, но естественное для имеющегося процесса взаимодействие: например, логика напрямую не командует людям сначала приготовить обед, а потом съесть его, но обычно получается именно так. Все три типа взаимосвязи должны быть заранее предусмотрены при конструировании системы. Разберем это подробнее.

Логическая связь. С установлением логической связи между модулями обычно проблем не возникает. Важно только помнить, что полученные связи очень существенно влияют на навигацию в пределах системы (особенно, когда система многооконная). Поэтому, чтобы не перегружать интерфейс, стоит избегать как слишком уж отдельных блоков (их трудно найти), так и блоков, связанных с большим количеством других. По опыту, для одного блока оптимальным числом связей является число три.

Связь по представлению пользователей. В информационных системах, когда необходимо гарантировать, что пользователь найдет всю нужную ему информацию, необходимо устанавливать связи между блоками, основываясь не только на точке зрения разработчика, но основываясь на представлениях пользователей. Дело в том, что чуть ли не единственный распространенный способ поиска, а именно поиск по классификации признаков, работает только в том случае, когда пользователи согласны с принципами этой классификации. Большинство же понятий однозначно классифицированы быть не могут из-за наличия слишком большого количества значимых признаков. Также проблема состоит в том, что реальный классификационный признак может отличаться от широко распространенного. Например, нужно как-то классифицировать съедобные растения. Помидор, который почти все считают овощем, на самом деле ягода. Не менее тяжело признать ягодой арбуз. Это значит, что классификация, приемлемая для ботаника, не будет работать для всех остальных, причем обратное не менее справедливо.

В то же время, существует очень простой способ классификации, с явной ненаучностью сочетающий не менее явную практическую пользу. Способ этот называется карточной сортировкой, при этом его название полностью совпадает с его сущностью. Все понятия, которые требуется классифицировать, пишутся на бумажных карточках из расчета «одно понятие – одна карточка». После чего группе пользователей из целевой аудитории предлагается эти карточки рассортировать (при этом каждый субъект получает свой набор карточек). Получившиеся кучки из карточек нужно разобрать на составляющие и свести результаты от разных субъектов в один способ классификации. Ничего более работоспособного до сих пор человечеством не придумано. В то же время этот способ имеет определенные недостатки: во-первых, трудно заполучить на несколько часов представителей целевой аудитории, а во-вторых, при малом количестве субъектов результаты могут быть сомнительны (как минимум, нужно 4-5 человек).

Процессуальная связь. Установление качественной процессуальной связи обычно довольно трудная задача, поскольку единственным источником информации является наблюдение за пользователями. В то же время установление такой связи дело

исключительно полезное. Зачем, например, рисовать на экране сложную систему навигации, если точно известно, к какому блоку пользователь перейдет дальше? В этом смысле зачастую оправдано навязывать пользователю какую-либо процессуальную связь, жертвуя удобством, зато выигрывая в скорости обучения (поскольку пользователю приходится думать меньше). Жестко заданная связь позволяет также уменьшить количество ошибок, поскольку от пользователя при ней не требуется спрашивать себя «не забыл ли я чего?». Замечательным примером жестко заданной процессуальной связи является устройство мастеров (wizards), при котором пользователя заставляют нажимать кнопку Далее.

Результат

В конце этого этапа должна получиться примерно такая схема (Рисунок 2).

Существует любопытная закономерность: чем эстетически привлекательней выглядит схема (без учета цветового кодирования и веселеньких шрифтов), тем она эффективней. Всегда надо стараться сделать схему возможно более стройной и ясной.

Рисовать такие схемы очень удобно в MS Visio или подобной ей системе. Результат, в особенности сложно выглядящий и напечатанный на листе бумаги большого формата, очень хорошо смотрится на стене и животворяще действует на заказчиков и руководство.

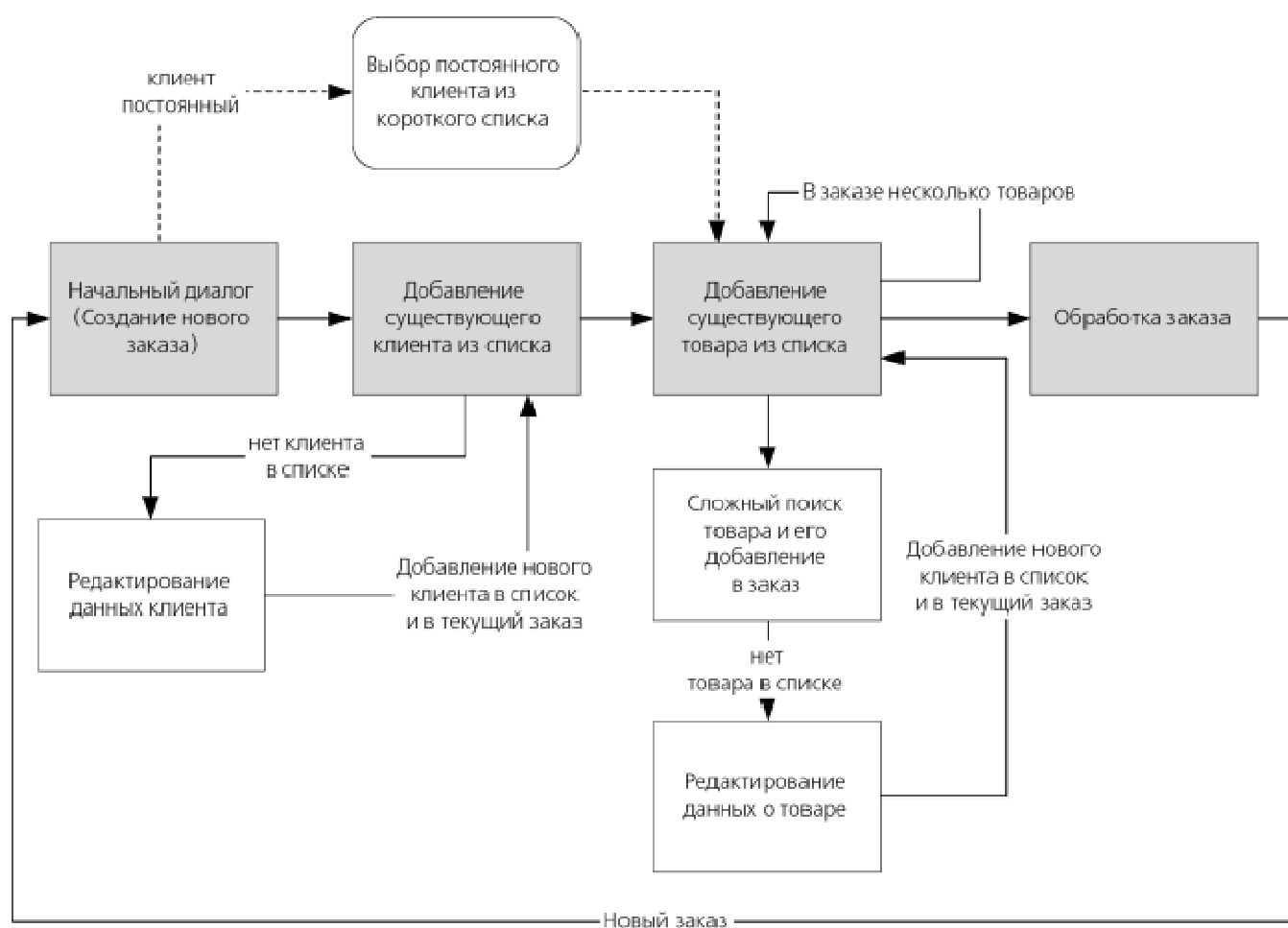


Рис. 2. Пример общей схемы. Прямоугольник обозначает отдельный экран, прямоугольник со скругленными углами – область экрана, пунктирная линия – альтернативное действие. Обратите внимание, что в этой схеме интерфейс заставляет пользователя выполнять задачу в сугубо определенной последовательности.

.Необходимость контроля. Этапы контроля.

Проверка схемы по сценарию.

Последней задачей перед построением прототипа является проверка внутренней логики системы. Дело в том, что всегда существует вероятность того, что вы что-то забыли или спланировали неправильно. Как уже было сказано, исправить эти ошибки лучше всего до построения прототипа (даже первой его версии). Конечно, многие структурные ошибки нельзя найти никакими методами, кроме длительного логического анализа. С другой стороны, практика показывает, что почти все найденные ошибки будут существенными. Так что лишняя проверка не повредит.

Для финальной проверки схемы вам пригодятся разработанные вами пользовательские сценарии. Не глядя на схему, необходимо подробно описать, как все вымышленные пользователи будут взаимодействовать с системой, не пропуская ни одного элемента управления. После чего сверить полученный текст со схемой.

Тут возможно три варианта развития событий: либо вы обнаружите, что вы что-то забыли задокументировать в схеме, либо обнаружите, что свеженаписанный рассказ значительно лучше схемы, вероятнее же всего, что и то и другое произойдет одновременно. На четвертый вариант, а именно на полное отсутствие проблем, рассчитывать, как правило, не стоит.

Весьма эффективным средством оценки получающегося интерфейса является его экспертная оценка. Часто оказывается, что сравнительно дорогое тестирование показывает то, что было бы легко видно постороннему, тем более вооруженному опытом и квалификацией, взгляду. Хотя экспертная оценка не может быть полноценной заменой тестирования, она обладает одним существенным преимуществом – для её проведения не требуется прототип. Это значит, что эксперт может быть приглашен на ранних стадиях работы, когда польза от обнаружения ошибок максимальна.

Для проведения экспертной оценки нужно знать следующее:

- Разные люди обнаруживают разные ошибки. Это значит, что метод работает лучше, когда количество экспертов больше единицы.
- Лучше привлекать несколько экспертов не одновременно, но последовательно.
- Чем больше информации о проектируемой системе будет предоставлено эксперту, тем более сложные проблемы он сможет выявить.
- Нельзя требовать от эксперта работы по весу. В большинстве случаев результатом его работы будут одна или две страницы текста (поскольку описание одной проблемы требует обычно всего двух или трех предложений). Если от эксперта будет

требоваться объемный результат работы, он включит в него много несущественных подробностей.

1.7 Лекция №7 (2 часа).

Тема: «Интерпретация естественного языка (ЕЯ)»

1.7.1 Вопросы лекции:

1. Значение обработки естественного языка.
2. Определить основные проблемы обработки естественного языка.
3. Технологии анализа естественного языка

1.7.2 Краткое содержание вопросов:

Первое устройство для распознавания речи появилось в 1952 году, оно могло распознавать произнесённые человеком цифры.^[1] В 1962 году на ярмарке компьютерных технологий в Нью-Йорке было представлено устройство IBM Shoebox.

В 1963 году в США были презентованы разработанные инженерами корпорации «Сперри» миниатюрные распознающие устройства с волоконно-оптическим запоминающим устройством под названием «Септрон» (Sceptron, но произносится ['septra:n] без «к»),^[2] выполняющие ту или иную последовательность действий на произнесённые человеком-оператором определённые фразы. «Септроны» годились для применения в сфере фиксированной (проводной) связи для автоматизации набора номеров голосом и автоматической записи надиктовываемого текста телетайпом, могли применяться в военной сфере (для голосового управления сложными образцами военной техники), авиации (для создания «умной авионики», реагирующей на команды пилота и членов экипажа), автоматизированных системах управления и др.^{[2][3][4]} В 1983 году был презентован интерактивный комплекс «умной авионики» для ударных вертолётов «Апач», распознающий команды и запросы пилота, преобразующий их в сигналы управления на бортовое оборудование и односложно отвечающий ему голосом относительно возможности реализации поставленной им задачи^[5].

Коммерческие программы по распознаванию речи появились в начале девяностых годов. Обычно их используют люди, которые из-за травмы руки не в состоянии набирать большое количество текста. Эти программы (например, Dragon NaturallySpeaking (англ.)русск., VoiceNavigator (англ.)русск.) переводят голос пользователя в текст, таким образом, разгружая его руки. Надёжность перевода у таких программ не очень высока, но с годами она постепенно улучшается.

Увеличение вычислительных мощностей мобильных устройств позволило и для них создать программы с функцией распознавания речи. Среди таких программ стоит отметить приложение Microsoft Voice Command, которое позволяет работать со многими приложениями при помощи голоса. Например, можно включить воспроизведение музыки в плеере или создать новый документ.

Все большую популярность применение распознавания речи находит в различных сферах бизнеса, например, врач в поликлинике может проговаривать диагнозы, которые тут же будут внесены в электронную карточку. Или другой пример. Наверняка каждый хоть раз в жизни мечтал с помощью голоса выключить свет или открыть окно. В последнее время в телефонных интерактивных приложениях все чаще стали использоваться системы автоматического распознавания и синтеза речи. В этом случае общение с голосовым порталом становится более естественным, так как выбор в нём может быть осуществлен не только с помощью тонового набора, но и с помощью голосовых команд. При этом

системы распознавания являются независимыми от дикторов, то есть распознают голос любого человека.

Следующим шагом технологий распознавания речи можно считать развитие так называемых интерфейсов безмолвного доступа (silent speech interfaces, SSI). Эти системы обработки речи базируются на получении и обработке речевых сигналов на ранней стадии артикулирования. Данный этап развития распознавания речи вызван двумя существенными недостатками современных систем распознавания: чрезмерная чувствительность к шумам, а также необходимость четкой и ясной речи при обращении к системе распознавания. Подход, основанный на SSI, заключается в том, чтобы использовать новые сенсоры, не подверженные влиянию шумов в качестве дополнения к обработанным акустическим сигналам.

1.8 Лекция №8-9 (4 часа).

Тема: «Распознавание речи»

1.8.1 Краткое описание проводимого занятия

Классификация систем распознавания речи

Системы распознавания речи классифицируются:

- по размеру словаря (ограниченный набор слов, словарь большого размера);
- по зависимости от диктора (дикторозависимые и дикторонезависимые системы);
- по типу речи (слитная или раздельная речь);
- по назначению (системы диктовки, командные системы);
- по используемому алгоритму (нейронные сети, скрытые Марковские модели, динамическое программирование);
- по типу структурной единицы (фразы, слова, фонемы, дифоны, аллофоны);
- по принципу выделения структурных единиц (распознавание по шаблону, выделение лексических элементов).

Для систем автоматического распознавания речи, помехозащищённость обеспечивается, прежде всего, использованием двух механизмов:

- Использование нескольких, параллельно работающих, способов выделения одних и тех же элементов речевого сигнала на базе анализа акустического сигнала;
- Параллельное независимое использование сегментного (фонемного) и целостного восприятия слов в потоке речи.

Методы и алгоритмы распознавания речи

«... очевидно, что алгоритмы обработки речевого сигнала в модели восприятия речи должны использовать ту же систему понятий и отношений, которой пользуется человек.»

Сегодня системы распознавания речи строятся на основе принципов признания **форм распознавания**. Методы и алгоритмы, которые использовались до сих пор, могут быть разделены на следующие большие классы:

Классификация методов распознавания речи на основе сравнения с эталоном.

- Динамическое программирование — временные динамические алгоритмы (Dynamic Time Warping).

Контекстно-зависимая классификация. При её реализации из потока речи выделяются отдельные лексические элементы — фонемы и аллофоны, которые затем объединяются в слоги и морфемы.

- Методы дискриминантного анализа, основанные на Байесовской дискриминации (Bayesian discrimination);
- Скрытые Марковские модели (Hidden Markov Model);
- Нейронные сети (Neural networks).

Архитектура систем распознавания

Типичная архитектура статистических систем автоматической обработки речи.

- Модуль шумоочистки и отделение полезного сигнала.
- Акустическая модель — позволяет оценить распознавание речевого сегмента с точки зрения схожести на звуковом уровне. Для каждого звука изначально строится сложная статистическая модель, которая описывает произнесение этого звука в речи.
- Языковая модель — позволяют определить наиболее вероятные словесные последовательности. Сложность построения языковой модели во многом зависит от конкретного языка. Так, для английского языка, достаточно использовать статистические модели (так называемые N-граммы). Для высокофлективных языков (языков, в которых существует много форм одного и того же слова), к которым относится и русский, языковые модели, построенные только с использованием статистики, уже не дают такого эффекта — слишком много нужно данных, чтобы достоверно оценить статистические связи между словами. Поэтому применяют гибридные языковые модели, использующие правила русского языка, информацию о части речи и форме слова и классическую статистическую модель.
- Декодер — программный компонент системы распознавания, который совмещает данные, получаемые в ходе распознавания от акустических и языковых моделей, и на основании их объединения, определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания слитной речи.

Этапы распознавания

1. Обработка речи начинается с оценки качества речевого сигнала. На этом этапе определяется уровень помех и искажений.
2. Результат оценки поступает в модуль акустической адаптации, который управляет модулем расчета параметров речи, необходимых для распознавания.
3. В сигнале выделяются участки, содержащие речь, и происходит оценка параметров речи. Происходит выделение фонетических и просодических вероятностных характеристик для синтаксического, семантического и прагматического анализа. (Оценка информации о части речи, форме слова и статистические связи между словами.)
4. Далее параметры речи поступают в основной блок системы распознавания — декодер. Это компонент, который сопоставляет входной речевой поток с информацией, хранящейся в акустических и языковых моделях, и определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания.

Признаки эмоционально окрашенной речи в системах распознавания

Основные понятия, которые характеризуют параметры речи человека, связанные с формой, размерами, динамикой изменения речеобразующего тракта и описывающие эмоциональное состояние человека, можно разделить на четыре группы объективных

признаков, позволяющих различать речевые образцы: спектрально-временные, кепстральные, амплитудно-частотные и признаки нелинейной динамики. Подробнее, каждая группа признаков:

Спектрально-временные признаки

Спектральные признаки:

- Среднее значение спектра анализируемого речевого сигнала;
- Нормализованные средние значения спектра;
- Относительное время пребывания сигнала в полосах спектра;
- Нормализованное время пребывания сигнала в полосах спектра;
- Медианное значение спектра речи в полосах;
- Относительная мощность спектра речи в полосах;
- Вариация огибающих спектра речи;
- Нормализованные величины вариации огибающих спектра речи;
- Коэффициенты кросскорреляции спектральных огибающих между полосами спектра.

Временные признаки:

- Длительность сегмента, фонемы;
- Высота сегмента;
- Коэффициент формы сегмента.

Спектрально-временные признаки характеризуют речевой сигнал в его физико-математической сущности исходя из наличия компонентов трех видов:

1. периодических (тональных) участков звуковой волны;
2. непериодических участков звуковой волны (шумовых, взрывных);
3. участков, не содержащих речевых пауз.

Спектрально-временные признаки позволяют отражать своеобразие формы временного ряда и спектра голосовых импульсов у разных лиц и особенности фильтрующих функций их речевых трактов. Характеризуют особенности речевого потока, связанные с динамикой перестройки артикуляционных органов речи говорящего, и являются интегральными характеристиками речевого потока, отражающими своеобразие взаимосвязи или синхронности движения артикуляционных органов говорящего.

Кепстральные признаки

Мел-частотные кепстральные коэффициенты;

- Коэффициенты линейного предсказания с коррекцией на неравномерность чувствительности человеческого уха;
- Коэффициенты мощности частоты регистрации;
- Коэффициенты спектра линейного предсказания;
- Коэффициенты кепстра линейного предсказания.

Большинство современных автоматических систем распознавания речи сосредотачивают усилия на извлечении частотной характеристики речевого тракта человека, отбрасывая при этом характеристики сигнала возбуждения. Это объяснено тем, что коэффициенты первой модели обеспечивают лучшую делимость звуков. Для отделения сигнала возбуждения от сигнала речевого тракта прибегают к кепстральному анализу.

Амплитудно-частотные признаки

- Интенсивность, амплитуда
- Энергия
- Частота основного тона (ЧОТ)
- Формантные частоты
- Джиттер (jitter) — дрожание частотная модуляция основного тона (шумовой параметр);
- Шиммер (shimmer) — амплитудная модуляция на основном тоне (шумовой параметр);
- Радиальная базисная ядерная функция
- Нелинейный оператор Тигер

Амплитудно-частотные признаки позволяют получать оценки, значения которых могут меняться в зависимости от параметров дискретного преобразования Фурье (вида и ширины окна), а также при незначительных сдвигах окна по выборке. Речевой сигнал акустически представляют собой распространяемые в воздушной среде сложные по своей структуре звуковые колебания, которые характеризуются в отношении их частоты (числа колебаний в секунду), интенсивности (амплитуды колебаний) и длительности. Амплитудно-частотные признаки несут необходимую и достаточную информацию для человека по речевому сигналу при минимальном времени восприятия. Но применение этих признаков не позволяет в полной мере использовать их в качестве инструмента идентификации эмоционально окрашенной речи.

Признаки нелинейной динамики

- Отображение Пуанкаре;
- Рекуррентный график;
- Максимальный характеристический показатель Ляпунова — Эмоциональное состояние человека, которому соответствует определенная геометрия аттрактора (фазовый портрет);^[17]
- Фазовый портрет (аттрактор);
- Размерность Каплана-Йорка — количественная мера эмоционального состояния человека, от «спокойствия» до «гнева» (деформация и последующее смещение спектра речевого сигнала).^[17]

Для группы признаков нелинейной динамики речевой сигнал рассматривается как скалярная величина, наблюдаемая в системе голосового тракта человека. Процесс речеобразования можно считать нелинейным и анализировать его методами нелинейной динамики. Задача нелинейной динамики состоит в нахождении и подробном исследовании базовых математических моделей и реальных систем, которые исходят из наиболее типичных предположений о свойствах отдельных элементов, составляющих систему, и законах взаимодействия между ними. В настоящее время методы нелинейной динамики базируются на фундаментальной математической теории, в основе которой лежит теорема Такенса (англ.)русск., которая подводит строгую математическую основу под идеи нелинейной авторегрессии и доказывает возможность восстановления фазового портрета аттрактора по временному ряду или по одной его координате. (Под аттрактором понимают множество точек или подпространство в фазовом пространстве, к которому приближается фазовая траектория после затухания переходных процессов.) Оценки характеристик сигнала из восстановленных речевых траекторий используются в построении нелинейных детерминированных фазово-пространственных моделей наблюдаемого временного ряда. Выявленные отличия в форме аттракторов можно использовать для диагностических правил и признаков, позволяющих распознать и правильно идентифицировать различные эмоции в эмоционально окрашенном речевом сигнале.

Параметры качества речи по цифровым каналам:

- Слоговая разборчивость речи;
- Фразовая разборчивость речи;
- Качество речи по сравнению с качеством речи эталонного тракта;
- Качество речи в реальных условиях работы.

Основные понятия

Разборчивость речи — относительное количество правильно принятых элементов речи (звуков, слогов, слов, фраз), выраженное в процентах от общего числа переданных элементов.

- Качество речи — параметр, характеризующий субъективную оценку звучания речи в испытываемой системе передачи речи.
- Нормальный темп речи — произнесение речи со скоростью, при которой средняя длительность контрольной фразы равна 2,4 с.
- Ускоренный темп речи — произнесение речи со скоростью, при которой средняя длительность контрольной фразы равна 1,5-1,6 с.
- Узнаваемость голоса говорящего — возможность слушателей отождествлять звучание голоса, с конкретным лицом, известным слушателю ранее.
- Смысловая разборчивость — показатель степени правильного воспроизведения информационного содержания речи.
- Интегральное качество — показатель, характеризующий общее впечатление слушателя от принимаемой речи.

Основным преимуществом голосовых систем объявлялась дружелюбность к пользователю. Речевые команды должны были избавить конечного пользователя от необходимости использования сенсорных и иных методов ввода данных и команд.

- Голосовое управление
- Голосовые команды
- Голосовой ввод текста
- Голосовой поиск

Успешными примерами использования технологии распознавания речи в мобильных приложениях являются: ввод адреса голосом в Яндекс.Навигаторе, голосовой поиск Google Now.

Помимо мобильных устройств, технология распознавания речи находит широкое распространение в различных сферах бизнеса:

- Телефония: автоматизация обработки входящих и исходящих звонков путём создания голосовых систем самообслуживания в частности для: получения справочной информации и консультирования, заказа услуг/товаров, изменения параметров действующих услуг, проведения опросов, анкетирования, сбора информации, информирования и любые другие сценарии;
- Решения "Умный дом": голосовой интерфейс управления системами «Умный дом»;
- Бытовая техника и роботы: голосовой интерфейс электронных роботов; голосовое управление бытовой техникой и т.д;
- Десктопы и ноутбуки: голосовой ввод в компьютерных играх и приложениях;
- Автомобили: голосовое управление в салоне автомобиля — например, навигационной системой;
- Социальные сервисы для людей с ограниченными возможностями.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

2.1 Практическое занятие №1,2 (4 часа).

Тема: «Основные критерии, используемые при разработке пользовательских интерфейсов АСОИиУ»

2.1.1 Задание для работы:

1. Подготовить описание программного продукта – разработанной в ходе преддипломной практики дипломного проектирования программной системы.

2. Не менее, чем двумя из трех предложенных в методических указаниях методов рассчитать технико-экономические показатели (ТЭП) проекта.

3. Выбрав из полученных вариантов наиболее выгодный, с коммерческой точки зрения, определить стоимость (договорную цену) разработки и внедрения программной системы.

2.1.2 Краткое описание проводимого занятия:

АИС «ТиС» - «Автоматизированная информационная система «Торговля и склад», версия 1.1. Поставщик АИС «ТиС» – лаборатория объектно-ориентированного моделирования информационных систем кафедры автоматизации обработки информации ТУСУР. Адрес: 634045, г. Томск, ул. Вершинина, 74, корп. ФЭТТУСУРа, ауд. 405. тел. (3822) 41-44-70. АИС «ТиС» предназначена для информационной поддержки процессов управления складами торговыми представительскими организациями заказчика. АИС «ТиС» соответствует ГОСТ Р ИСО/МЭК 12119-2000 «Информационная технология. Пакеты программ. Требования к качеству и тестирование».

Программно-технические средства:

- программное обеспечение АИС «ТиС» разработано в архитектуре «клиент-сервер», средой разработки является язык программирования Delphi, СУБД – Firebird 1.5, под управлением операционной системы MS Windows 2000/XP; 57

- техническое обеспечение:
– Сервер: CPU Intel Pentium IV 2,6 GHz / 1024 Mb, HDD 160 Gb SATA, сеть 1000 / 100 Mb / s, ОС MS Windows 2000/XP, СУБД – Firebird 1.5;

- Клиент: CPU Intel Celeron 1,3 GHz / 128 Mb, HDD 120 Gb, ОС MS Windows 2000/XP, СУБД – Firebird 1.5.

В комплект поставки входят:

1) оптический носитель (компакт-диск), содержащий дистрибутив системы в загрузочных модулях и инсталляционные пакеты, в том числе:

- клиентская часть, представляющая собой программное обеспечение для работы с базой данных АИС;

- серверная часть, представляющая собой файл данных БД АИС «ТиС»;

- инсталляционные пакеты СУБД Firebird 1.5.1. под Linux и Windows, являющейся клоном Interbase и разрабатываемой как продукт OpenSource. Данная СУБД может использоваться бесплатно. Информацию о СУБД Firebird 1.5.1. можно получить на сайте производителя <http://www.ibphoenix.com>. АИС «ТиС» на стадии комплексных испытаний и опытной эксплуатации протестирован в работе с данной СУБД и является полностью совместимой с ней.

2) техническая документация в составе: – АИС «ТиС». Техническое задание. – АИС «ТиС». Общее описание системы. – АИС «ТиС». Руководство пользователя.

- АИС «ТиС». Руководство программиста. Документация выполнена в соответствии с Государственным стандартом РФ ГОСТ Р ИСО/МЭК 15910—2002. Информационная технология. Процесс создания документации пользователя программного средства. АИС «ТиС» снабжена программой инсталляции, обеспечивающей установку и настройку программного обеспечения в локальной сети (сервер) и на отдельные компьютеры

(рабочие станции). 58 Для установки системы необходимо вставить установочный компакт-диск в привод CD-ROM, самостоятельно запустить файл setup.exe и следовать в дальнейшем руководству по установке (входит в «Руководство пользователя»), включающем так же инструкцию по установке установочных пакетов СУБД Firebird 1.5.1. Предусмотрена техническая поддержка и сопровождение системы на основании соответствующего соглашения Поставщика и Покупателя. Функциональные возможности системы. АИС «ТиС» реализована как распределенная система управления процессами закупки и реализации продукции в составе комплекса взаимосвязанных АРМ («Администратор», «Прием продукции», «Прием заказов», «Прием платежей», «Исполнение заказов»), объединенных единым форматом представления данных, идеологией обработки информации ориентированных на использование баз данных общего пользования, единую техническую базу и операционную среду. Каждый АРМ рассчитан на эксплуатацию конкретным подразделением организации. Для оптимизации процессов сбора и обработки информации, минимизации затрат на ввод и хранение данных, повышения актуальности, достоверности и сопоставимости данных различных информационных систем в АИС «ТиС» реализованы следующие технологические требования: – централизованная база данных, с возможностью подключения к ней с удаленных терминалов посредством сети Интернет;

- однократный ввод данных в систему с возможностью дальнейшего их использования в функционально связанных подсистемах;

- включение в состав хранимой БД информации только тех данных, для которых существуют надежные тракты актуализации;

- своевременная актуализация данных в базе в зависимости от вида хранимой информации. 59 АИС «ТиС» выполняет следующие функции: – внесение, поиск, обработка информации о поставщиках и поставляемых продуктах;

- внесение, поиск, обработка информации о заказчиках и заказах;

- внесение, поиск, обработка информации о платежах;

- проведение товарно-финансовых операций при исполнении заказов;

- администрирование АИС, в том числе ведение словарей, справочников, классификаторов, списков пользователей.

- разграничение доступа к данным функциям АИС на основе системы ролей и привилегий. АИС «ТиС» обеспечивает круглосуточную бесперебойную работу при отсутствии помех со стороны аппаратного обеспечения и операционной системы. Предусмотрены механизмы автоматического резервного копирования данных, возможна настройка периодичности выполнения резервного копирования. Предусмотрена возможность восстановления системы по команде администратора на основе последних результатов резервного копирования. В случае возникновения системного сбоя (отключение питания, неполадки в аппаратном обеспечении, в работе операционной системы) АИС «ТиС» выполняет самостоятельное восстановление работоспособности после перезагрузки системы. Взаимодействие с пользователем организовано посредством графического пользовательского интерфейса в общепринятой форме. Предусмотрены возможности настройки пользовательского интерфейса под требования пользователя: возможна настройка системы меню, внешнего вида приложения. Настройки хранятся на сервере БД и действуют для конкретного пользователя из любой точки доступа к АИС. При работе с системой предполагается, что пользователи знакомы в общих чертах с операционной системой MS Windows 9x/XP и владеют базовыми навыками работы в ней, при этом они должны обладать простейшим опытом работы с окнами (Windows), приемами работы с меню и уметь пользоваться 60 стандартными диалогами, а так же обладать минимальными навыками работы в сети Интернет. Для облегчения работы пользователя в АИС «ТиС» предусмотрена контекстная справочная система, содержащая информацию о порядке работы с конкретными функциями АИС. Эффективность использования системы заключается в сокращении времени выполнения расчетов по

текущим торговым операциям – сокращении времени построения оперативных и регламентных отчетов;

–повышении качества и контроля исполнительской дисциплины сотрудников организации. Время отклика системы на запрос пользователя при условии, что аппаратная конфигурация удовлетворяет рекомендуемым системным требованиям, не превышает –1сек.–при выполнении оперативных внесений в БД;

–10сек.– при построении оперативных отчетов;

–5мин.-при построении регламентных отчетов. Построение регламентных отчетов происходит в фоновом режиме (пользователь может продолжать работу с системой в процессе построения регламентного отчета). Время восстановления работоспособности системы после перезагрузки не превышает 15 минут, при условии, что аппаратная конфигурация удовлетворяет рекомендуемым системным требованиям. ВАИС «ТиС» предусмотрена возможность уведомления администратора по электронной почте о возникновении сбоев в работе системы. Специальные возможности изменения режимов функционирования системы не предусмотрены. Специальные возможности модификации системы не предусмотрены. Возможность перенесения АИС «ТиС» на другую аппаратно-программную платформу не предусмотрена. Надежность, практичность, эффективность, сопровождаемость и мобильность системы соответствует основным положениям ГОСТ28195-89 Оценка качества программных средств. Общие положения.

2.2 Практическое занятие №3,4 (4часа).

Тема: «Показатели, используемые для оценки удобства диалоговой системы для пользователя»

2.2.1 Задание для работы:

1.Научить разрабатывать интерфейс пользователя в соответствии с требованиями ТЗ и ТП.

2.Сформировать умения создавать БД, соответствующую требованиям ТЗИТП при использовании разработанных моделей.

3.Сформировать умения разрабатывать алгоритмы и программировать задачи в соответствии с ТП.

2.2.2 Краткое описание проводимого занятия:

Описание база данных

База данных—представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

В базе данных (рисунок1) создано три таблицы: «Услуга», «Клиент» и «Заказ».

«Услуга» предназначена для хранения информации об услугах предоставляемых «Интернет-центром» и включает в себя следующие поля:

1)уникальный номеруслуги;

2)наименование;

3)стоимость;

4)срок выполнения.

«Клиент» служит для хранения информации о клиентах, которые осуществляют заказана какие-либо услуги в «Интернет - центре» состоит из полей:

1)уникальный номер клиента;

2)фамилия;

3)имя;

4)отчество;

5)мобильный телефон;

6)номер паспорта.

«Заказ»

являетсявторостепенной,ислужитдлясвязитаблиц«Услуга»и«Клиент».Даннаятаблицасостоитизследующихполей:

1)уникальный номер заказа;

2)уникальный номер клиента;

3)уникальный номер услуги.

Описание интерфейса

Интерфейс совокупность возможностей, способов и методов в взаимодействия двух систем (любых, а необязательно являющиеся вычислительными или информационными), устройств или программ для обмена информацией между ними, определённая их характеристиками, характеристиками соединения,сигналов обменаи т.п. В случае, если одна из взаимодействующих систем человек, чаще говорят лишь о второй системе, то есть об интерфейсе той системы, с которой человек взаимодействует.

Первый запуск программы вызывает окно (рисунок4), в котором пользователю доступен перечень услуг для ознакомления и дана возможность на осуществления заказа на какую-либо услугу.

Для выбора необходимой услуги, пользователю достаточно лишь нажать на неё и произойдёт автоматическое выделение выбранной услуги в следствии которой можно нажать на кнопку «Произвести заказ» и ввести необходимые данные для формирования заказа.

Так же при первом запуске, доступны 2 пункта меню «Функции» и «О программе», каждый из которых имеет свои подпункты.

Пункт меню «Функции» предназначен в основном для манипуляции с данными хранящимися в базе данных «Интернет-центр», а так же есть аналогичный пункт кнопке «Произвести заказ» который имеет точно такое же наименование и функцию. Сам же пункт состоит из следующих подпунктов

1)произвести заказ;

2)администратор.

Функция «Администратор» предназначена для персонала «Интернет-центра» и защищена паролем. При нажатии на данную функцию левой кнопкой мыши, произойдет вызов окна (рисунок 6), где потребуется ввести пароль для доступа к функциям «Администратора».

2.3 Практическое занятие №5,6 (4 часа).

Тема: «Основные концепции построения пользовательских интерфейсов в АСОИиУ»

2.3.1 Задание для работы:

1. Произвести Сравнение компьютеров между собой с оценкой их производительности.

2. Анализ подходов к оценке производительности вычислительных систем.

2.3.2 Краткое описание проводимого занятия:

Сравнение компьютеров между собой обычно начинают с оценки их производительности. Это потребовало введения соответствующих единиц измерения производительности и разработки стандартных методов ее оценки.

Методы оценки производительности вычислительных систем должны отвечать определенным требованиям. Прежде всего, они должны быть общепризнанными, максимально полно оценивать вычислительные системы и соответствовать задачам пользователя. Должна быть обеспечена их доступность для независимой и самостоятельной экспертизы.

В основе используемых в настоящее время методов оценки и сравнения производительности вычислительных систем лежит время.

Компьютер, который выполняет определенный объем работы за меньшее время, считается более быстродействующим.

Оценка временных характеристик выполняемых процессором *работ*-далеко нетривиальная задача.

Для пользователя важно время, за которое решается задача. Но практически нет программ, которые могут работать без некоторой операционной системы. В связи с этим возникают дополнительные накладные временные *затраты*. Поэтому при измерении производительности чаще всего используется сумма пользовательского и системного времени центрального процессора.

Дополнительные сложности при сравнении временных характеристик решаемых на разных компьютерах задач обусловлены тем, что системный код при решении задачи на одном компьютере на другом компьютере может стать пользовательским.

В ходе измерений появляются значительные методические ошибки, вызванные нарушением базовой концепции построения измерительных систем, в результате чего "измерительный прибор" становится неотъемлемой частью измеряемой системы.

Возникают серьезные методические трудности сравнения производительности компьютеров с разными операционными системами.

В связи с этим в методиках оценки производительности используют разное время:

- астрономическое;
- время выполнения(*executiontime*);
- время ответа(*responsetime*);
- прошедшее время(*elapsedtime*), представляющее задержку выполнения задания и включающее: время работы процессора, время обращения к жесткому диску, время обращения к ОП, время выполнения операций ввода/вывода, накладные расходы операционной системы.

Для измерения времени, затрачиваемого процессором на выполнение данной программе, используется специальное время, которое называют временем центрального процессора(*CPUtime*). Из этого времени исключено *время ожидания* выполнения ввода-вывода и *время выполнения* других программ. Необходимость применения этого параметра обусловлена тем, что в многопрограммном режиме происходит совмещение во времени процедур, обслуживающих задачи, которые используют разные ресурсы.

Время центрального процессора можно разделить пользовательское и системное.

Все современные ЭВМ работают в общем случае, как *минимум*, в двух задачном режиме, обслуживая запросы операционной системы и запросы задачи пользователя.

В случаях, когда функции операционных систем могут быть сведены до одной простой задачи диспетчеризации заданий, системным временем центрального процессора можно пренебречь. Примером могут служить бортовые системы управления авиационных и космических комплексов, *автоматизированные системы управления технологическими процессами*.

Особенностью процессора как объекта измерения является то, что скорость взаимодействия функциональных устройств процессора чаще всего не зависит от динамических характеристик этих устройств, а задается частотой генератора тактовых импульсов. Поэтому время центрального процессора может быть вычислено двумя способами:

- умножением количества тактов синхронизации необходимых для выполнения данной программы на длительность такта синхронизации,
- либо делением количества тактов синхронизации для данной программы на частоту синхронизации.

Важной характеристикой процессора является среднее количество тактов синхронизации, необходимых для выполнения одной команды (*CPI clock cyclesper instruction*). Этот *параметр* позволяет легко оценить время центрального процессора, необходимое для данной программы, зная количество выполняемых в программе команд.

Производительность центрального процессора определяется тремя параметрами:

- тактовой частотой;
- средним количеством тактов на команду;
- количеством выполняемых команд.

Все эти параметры тесно связаны между собой:

- среднее количество тактов на команду зависит от функциональной организации и системы команд;
- частота синхронизации зависит от уровня технологии производства аппаратных средств и *функциональной организацией* процессора;

- количество выполняемых в программе команд определяется решаемой задачей, архитектурой системы и алгоритмом работы компиляторов.
- Для сравнения двух компьютеров по их производительности необходимо рассматривать все три компонента.

В процессе поиска стандартной метрики для оценки производительности вычислительных систем было выбрано несколько единиц измерения.

Одной из наиболее распространенных средних является *MIPS*(*MillionInstructionPerSecond*). Она показывает количество команд программы, выполняемых в секунду.

Производительность определяется как обратная ко времени выполнения величина, а более быстродействующий компьютер имеет более высокий *MIPS-рейтинг*.

На первый взгляд *метрика* очень проста, но ее использование не позволяет учесть особенности выполняемой программы и архитектуру процессора. Невозможно сравнивать компьютеры, имеющие разные системы команд. Даже на одном компьютере результат меняется при решении разных задач. Серьезные проблемы возникают в связи с использованием сопроцессоров для выполнения команд с плавающей запятой и оптимизирующих компиляторов. Все это не дает возможности выполнять объективное сравнение разных процессоров.

С понятием *MIPS* в свое время была связано еще две метрики, основанные на производительности эталонных компьютеров *VAX11/780* компании *DEC* и одной из моделей *RS/6000* компании *IBM*. *Производительность* этих компьютеров определялась как *1MIPS*. *Производительность* тестируемых систем вычислялась относительно производительности этих компьютеров, которые являются эталонными. Приблизительное соотношение между этими величинами таково: *1IBM MIPS* равен *1.6DEC MIPS*.

Использование эталонного компьютера сопряжено с серьезными трудностями:

- Необходимо постоянно воспроизводить или сохранять эталон;
- Требуется двойное программирование тестовых задач: на тестируемом и эталонном компьютере.

Использование измерения производительности компьютеров в *MIPS* не всегда объективно. Примером могут служить задачи, связанные с научно-техническими расчетами. В них широко применяется арифметика с плавающей точкой. Решение этих задач требует значительных вычислительных ресурсов. Вопрос оценки производительности используемых вычислительных систем всегда стоял очень остро. При этом достигнутые показатели производительности служили и служат показателем уровня разработок вычислительной техники.

Поэтому для подобных приложений *производительность* процессора оценивалась и оценивается в *FLOPS* (*Floating point Operations PerSecond*), который показывает, сколько операций с плавающей запятой выполняется компьютером в секунду.

Современные компьютеры имеют достаточно высокий уровень производительности, поэтому в настоящее время используются *производные* величины от *FLOPS*: *MFLOPS*, *GFOPS*, *TFLOPS*, *PFLOPS*.

Этот показатель в основном характеризует использование компьютеров в научно-технических приложениях. Он предназначен для оценки быстродействия при выполнении вычислительной системой операций плавающей арифметики. В противном случае его использование не всегда корректно.

Этот показатель производительности, также как и *MIPS*, зависит от конкретной выполняемой программы и архитектуры процессора.

Однако у *MFLOPS* есть свои недостатки. Прежде всего, это связано с тем, что отсутствует совместимость операций с плавающей точкой на различных компьютерах.

Другая проблема заключается в том, что *время выполнения* разных операций с плавающей запятой может существенно отличаться, в результате *рейтинг MFLOPS* существенно зависит от состава тестовой смеси. Например, *программа*, состоящая из операций сложения, будет иметь более высокий *рейтинг*, чем *программа*, состоящая из операций умножения, при равном количестве этих операций.

Для оценки показателей производительности в *MIPS* и *MFLOPS* используется очень много различных тестов. Рассмотрим не сколько примеров.

Тестовая смесь *Dhrystone* в настоящее время практически не применяется.

Смесь состоит из ста команд. Из них 53 оператора присваивания, 32 команды управления, 15 вызовов функций. В качестве метрики используется количество *Dhrystone* в секунду. Тест очень короткий[301,302].

"Ливерморские циклы" представляют собой типичный набор фрагментов программ на языке *FORTRAN*. [301,302] В этих программах реализованы разные вычислительные алгоритмы:

- сеточные;
- волновые;
- последовательные.

Их выбор был основан на богатом опыте создания суперкомпьютеров и проведения сложнейших научных инженерных расчетов Ливерморской национальной лаборатории им. Э. Лоуренса (Lawrence Livermore National Laboratory, LLNL) Министерства энергетики США

При тестировании используется либо малый набор (14 циклов), либо большой набор (24 цикла).

Коэффициент распараллеливания применяемых алгоритмов лежит в диапазоне от 0 до 1. Это позволяет использовать "Ливерморские циклы" для оценки производительности вычислительных систем, имеющих различную архитектуру. Тест практически не используется.

Тесты *LINPACK* представляют собой программы решения систем линейных алгебраических уравнений большой размерности. Они написаны на языке программирования *FORTRAN* [301,302].

Применяемые в этих программах алгоритмы широко используются при решении многих практических задач. С этим связана популярность этого теста среди пользователей.

Аргонская национальная лаборатория (*Argonne National Lab*) Министерства энергетики США регулярно публикует результаты тестирования различных вычислительных систем на тестах *LINPACK*.

Тест *LINPACK* используется при составлении рейтинга самых высокопроизводительных компьютеров мира. В основе используемых в *LINPACK*

алгоритмов лежит метод декомпозиции, широко применяемый при высокопроизводительных вычислениях. Достоинством тестов *LINPACK* является их структурированность. Для реализации элементарных операций над векторами, которые включают *умножение* векторов на скаляр, *сложение* векторов, *скалярное произведение* векторов выделяется базовый уровень системы, называемый BLAS (*Basic Linear Algebra Subprograms*).

Исходные данные для тестирования представляются в виде вещественных чисел двойной точности. Полученные результаты выражаются в *MFLOPS*.

Тест *LINPACK* имеет два уровня.

В тесте первого уровня *LINPACKDP* используется исходной матрицы размером 100*100. В тесте второго уровня *LINPACKTPP* исходная матрица имеет размерность 1000*1000.

Первый уровень этого теста нельзя применять в вычислительных системах, позволяющих поместить всю исходную матрицу в *кэш*-памяти.

В этом случае полученные результаты могут существенно превышать реальные возможности системы.

Применение тестов *LINPACKTPP*, *LINPACKTPP* для систем с массовым параллелизмом поводит к неадекватной оценки их производительности

Для оценки производительности таких систем используется тест *LINPACKHPC*(*Highly Parallel Computing*), который обеспечивает полную загрузку вычислительных ресурсов MPP-системы, увеличивая размеры матрицы. Вариант этого теста разработан и для параллельных вычислительных систем.

Тестовый пакета *LINPACK* с двойной точностью широко используется создателями высокопроизводительных вычислительных систем.

Анализ подходов к оценке производительности вычислительных систем позволяет сделать следующие выводы:

Методика измерения производительности компьютеров во многом зависит от выбора единиц ее измерения;

Инструментальные средства измерения производительности вычислительных систем должны быть адаптивными как к их архитектурным особенностям, так и *классам решаемых* с их помощью задач.

2.4 Практическое занятие №7,8 (4часа).

Тема: «Структурные свойства шага диалога»

2.4.1 Задание для работы:

- 1.Подготовить описание структурированного свойства диалога.
- 2.Сделать вывод по полной функции управления в АСОИУ

2.4.2 Краткое описание проводимого занятия:

Объект управления—это часть системы, которой управляют, а *управляющий орган (субъектуправления)* — это та часть системы, которая осуществляет управление.

1. *Внешняя среда*—это окружающая обстановка, в которой система существует. *Модель объекта управления и модель внешней среды* это совокупность представлений о поведении объекта управления и внешней среды, которыми пользуется управляющий орган системы.

2. *Вектор целей управления* —иерархически упорядоченная совокупность целей, которых система стремится достичь в процессе своей деятельности. Отдельные составляющие этого вектора называются частными целями.

3. *Вектор контрольных параметров* —иерархически упорядоченная совокупность параметров, предназначенных для оценки степени достижения частных целей вектора целей системы.

4. *Вектор ошибки управления* —это расхождение между желаемыми и реальными значениями в векторе контрольных параметров. Вектор ошибки является основой для формирования оценки качества управления в системе.

5. *Вектор управляющих воздействий (параметров)* —это совокупность параметров, характеризующих управляющие воздействия на объект управления, а так же на внешнюю среду системы. Значения управляющих параметров в системе меняются непосредственно по произволу субъекта управления.

6. *Устойчивость системы попредсказуемости* — это свойство системы, которое заключается в том, что ее поведение под воздействием внешней среды, внутренних изменений и управления в определенной мере предсказуемо.

7. *Полная функция управления (ПФУ)* системы —это деятельность, охватывающая все необходимые вопросы по управлению системой. В отличие от технических систем, которые функционируют и должны функционировать в регламентированных, заранее оговоренных при их разработке условиях, организационные системы напротив существуют во внешней среде, которая может изменяться настолько, что эти изменения порой могут привести к необходимости кардинальной перестройки управления системой.

Полная функция управления в АСОИУ реализует поэтапный процесс, начиная от формирования вектора целей и заканчивая их осуществлением. Содержательным фрагментом полной функции является *целевая функция управления*, то есть концепция достижения одной из частных целей вектора целей системы на одном из этапов ее существования. Совокупность целевых функций управления, то есть концепций управления по отношению ко всем частным целям, образует совокупную концепцию управления или просто *концепцию управления* системы.

Полная функция управления в АСОИУ в силу много плановости и разнообразия условий, в которой ей приходится существовать, должна включать в себя следующие виды управленческой деятельности.

2.5 Практическое занятие №9,10 (4часа).

Тема: «Принципы построения интеллектуальных диалоговых систем(ИДС))»

2.5.1 Задание для работы:

- 1.Определить значение индивидуальных диалоговых средств
- 2.Ряд условий, необходимых для осуществления процесса диалога

2.5.2 Краткое описание проводимого занятия:

В диалоговых системах речевые акты должны быть оформлены соответствующим образом, поскольку они составляют компонент человеческого общения, а так же являются не менее важным аспектом спонтанного общения. Диалоговые системы –неотъемлемая составляющая интеллектуальных компьютерных систем, служащих для переработки информации. Именно с их помощью осуществляются практически все процессы внутри системы. Основным требованием диалоговых систем является обеспечение более удобной и естественной формы взаимодействия интеллектуальных систем с пользователями. Другими словами – это системы с естественно-языковым интерфейсом.

Среди систем обработки естественного языка обычно выделяются вопросно-ответные системы, диалоговые системы решения задачи системы обработки связных текстов.

В целом, процесс так называемого «общения» человека с компьютерной системой можно назвать диалогом. Но дадим более общее представление этому понятию.

Диалог- это процесс обмена сообщениями между пользователем и компьютером. В диалоге постоянно происходит смена ролей информатора и пользователя, к тому же, смена ролей должна быть достаточно оперативной.

Приведем ряд условий, необходимых для осуществления процесса диалога:

- общая цель пользователя и информатора;
- смена ролей пользователя и компьютера;
- общий язык общения;
- наличие общей базы данных;
- возможность пополнения базы.

Реакция системы на входной текст определяется не только самим текстом, но и возможностями и знаниями системы. В случае, если входной текст не соответствует возможностям и знаниям системы, система, соответственно, не может сообщить ему интересующую его информацию. Чтобы пользователь мог понять причины не соответствия, система должна в виде косвенного ответа объяснить причину своих затруднений.

2.6 Практическое занятие №11,12 (4часа).

Тема: «Понимание естественного языка (ЕЯ)»

2.6.1 Задание для работы:

- 1.Значение обработки естественного языка.
- 2.Определить основные проблемы обработки естественного языка.
- 3.Технологии анализа естественного языка

2.6.2Краткое описание проводимого занятия:

Первое устройство для распознавания речи появилось в 1952 году, оно могло распознавать произнесённые человеком цифры.^[1] В 1962 году на ярмарке компьютерных технологий в Нью-Йорке было представлено устройство IBM Shoebox.

В 1963 году в США были представлены разработанные инженерами корпорации «Сперри» миниатюрные распознающие устройства с волоконно-оптическим запоминающим устройством под названием «Септрон» (Sceptron, но произносится ['septra:n] без «к»),^[2] выполняющие ту или иную последовательность действий на произнесённые человеком-оператором определённые фразы. «Септроны» годились для применения в сфере фиксированной (проводной) связи для автоматизации набора номеров голосом и автоматической записи надиктовываемого текста телетайпом, могли применяться в военной сфере (для голосового управления сложными образцами военной техники), авиации (для создания «умной авионики», реагирующей на команды пилота и членов экипажа), автоматизированных системах управления и др.^{[2][3][4]} В 1983 году был представлен интерактивный комплекс «умной авионики» для ударных вертолётчиков «Апач», распознающий команды и запросы пилота, преобразующий их в сигналы управления на бортовое оборудование и односложно отвечающий ему голосом относительно возможности реализации поставленной им задачи^[5].

Коммерческие программы по распознаванию речи появились в начале девяностых годов. Обычно их используют люди, которые из-за травмы руки не в состоянии набирать большое количество текста. Эти программы (например, Dragon NaturallySpeaking (англ.)русск., VoiceNavigator (англ.)русск.) переводят голос пользователя в текст, таким образом, разгружая его руки. Надёжность перевода у таких программ не очень высока, но с годами она постепенно улучшается.

Увеличение вычислительных мощностей мобильных устройств позволило и для них создать программы с функцией распознавания речи. Среди таких программ стоит отметить приложение Microsoft Voice Command, которое позволяет работать со многими приложениями при помощи голоса. Например, можно включить воспроизведение музыки в плеере или создать новый документ.

Все большую популярность применение распознавания речи находит в различных сферах бизнеса, например, врач в поликлинике может проговаривать диагнозы, которые тут же будут внесены в электронную карточку. Или другой пример. Наверняка каждый хоть раз в жизни мечтал с помощью голоса выключить свет или открыть окно. В последнее время в телефонных интерактивных приложениях все чаще стали использоваться системы автоматического распознавания и синтеза речи. В этом случае общение с голосовым порталом становится более естественным, так как выбор в нём может быть осуществлен не только с помощью тонового набора, но и с помощью голосовых команд. При этом системы распознавания являются независимыми от дикторов, то есть распознают голос любого человека.

Следующим шагом технологий распознавания речи можно считать развитие так называемых интерфейсов безмолвного доступа (silent speech interfaces, SSI). Эти системы обработки речи базируются на получении и обработке речевых сигналов на ранней стадии артикулирования. Данный этап развития распознавания речи вызван двумя существенными недостатками современных систем распознавания: чрезмерная чувствительность к шумам, а также необходимость четкой и ясной речи при обращении к системе распознавания. Подход, основанный на SSI, заключается в том, чтобы использовать новые сенсоры, не подверженные влиянию шумов в качестве дополнения к обработанным акустическим сигналам.

2.7 Практическое занятие №13-14 (4 часов).

Тема: «Поддержка КП в диалоге»

2.7.1 Краткое описание проводимого занятия:

Алгоритм симметричного шифрования требует наличия одного ключа для шифрования и дешифрования сообщений. Такой ключ называется общим секретным, поскольку все пользователи, участвующие в обмене данными, имеют один и тот же ключ.

В настоящее время имеется целый ряд алгоритмов симметричного шифрования. Среди них отметим DES (Data Encryption Standard — стандарт шифрования данных), IDEA (International Data Encryption Algorithm — международный алгоритм шифрования данных) — патентованный алгоритм, применяемый в PGP, и Blowfish — непатентованный алгоритм, применяемый в SSH.

С алгоритмами симметричного шифрования связано понятие стойкости шифра. Стойкость — это мера сопротивления крипто аналитическим атакам. Стойкость алгоритма определяется размерами с используемого ключа. В IDEA применяются 128 разрядные ключи. В алгоритме Blowfish размер ключа конфигурируется от 32 до 448 бит. Чем длиннее ключ, тем более стойкий шифр. В DES используются 56 разрядные ключи, поэтому данный алгоритм считается относительно слабым.

Для повышения стойкости шифра можно применить несколько ключей или выполнить алгоритм шифрования несколько раз подряд. Примером такой реализации является алгоритм TripleDES (встроен в некоторые свободно распространяемые утилиты), где данные сначала шифруются одним ключом, затем дешифруются другими, наконец, повторно шифруются третьим.

Основная проблема, связанная с алгоритмами симметричного шифрования, — необходимость использования секретного ключа. Прежде чем начать зашифрованный диалог, следует убедиться в том, что все участники диалога имеют соответствующий ключ. Этого можно добиться разными способами: выслать ключ по факсу, по почте, прибегнуть к услугам службы курьерской доставки. Но все они достаточно не удобны имеют свои слабые места. Более надежный, хотя и не лишенный недостатков метод — воспользоваться асимметричным шифрованием для кодирования секретного ключа и выслать его по электронной почте.

Асимметричное шифрование

Алгоритм асимметричного шифрования требует использовать один ключ для шифрования данных и другой, но взаимосвязанный с ним ключ — для дешифрования. Один из ключей в такой схеме доступен любому, кто его запрашивает. Такой ключ называется открытым. Другой ключ известен только владельцу и называется личным.

Алгоритмы асимметричного шифрования возникли в связи с необходимостью передавать секретные ключи по незащищенным каналам. Первую систему такого рода разработал Ральф Меркл (Ralph Merkle) в 1974 году. Первым алгоритмом, завоевавшим широкую популярность, был алгоритм Диффи Хеллмана, созданный Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman) в 1976 году. В 1977 году Рон Ривест (Ron Rivest), Ади Шамир (Adi Shamir) и Лен Эйдельман (Len Adleman) разработали схожий алгоритм RSA.

Алгоритмы асимметричного шифрования можно применять по прямому назначению (обеспечение конфиденциальности), а так же для создания цифровых подписей (аутентификация). Но по своей надежности они не соперники алгоритмам симметричного шифрования. В связи с этим асимметричные алгоритмы чаще всего

применяют для шифрования секретных ключей, передаваемых по не защищенным каналам, и для создания цифровых подписей.

2.8 Практическое занятие №15,16,17 (6 часов).

Тема: «Распознавание речи»

2.8.1 Краткое описание проводимого занятия:

Системы распознавания речи классифицируются:

- по размеру словаря (ограниченный набор слов, словарь большого размера);
- по зависимости от диктора (дикторозависимые и дикторонезависимые системы);
- по типу речи (слитная или раздельная речь);
- по назначению (системы диктовки, командные системы);
- по используемому алгоритму (нейронные сети, скрытые Марковские модели, динамическое программирование);
- по типу структурной единицы (фразы, слова, фонемы, дифоны, аллофоны);
- по принципу выделения структурных единиц (распознавание по шаблону, выделение лексических элементов).

Для систем автоматического распознавания речи, помехозащищённость обеспечивается, прежде всего, использованием двух механизмов:

- Использование нескольких, параллельно работающих, способов выделения одних и тех же элементов речевого сигнала на базе анализа акустического сигнала;
- Параллельное независимое использование сегментного (фонемного) и целостного восприятия слов в потоке речи.

Методы и алгоритмы распознавания речи

«... очевидно, что алгоритмы обработки речевого сигнала в модели восприятия речи должны использовать ту же систему понятий и отношений, которой пользуется человек.»

Сегодня системы распознавания речи строятся на основе принципов признания **форм распознавания**. Методы и алгоритмы, которые использовались до сих пор, могут быть разделены на следующие большие классы:

Классификация методов распознавания речи на основе сравнения с эталоном.

- Динамическое программирование — временные динамические алгоритмы (Dynamic Time Warping).

Контекстно-зависимая классификация. При её реализации из потока речи выделяются отдельные лексические элементы — фонемы и аллофоны, которые затем объединяются в слоги и морфемы.

- Методы дискриминантного анализа, основанные на Байесовской дискриминации (Bayesian discrimination);
- Скрытые Марковские модели (Hidden Markov Model);
- Нейронные сети (Neural networks).

Типичная архитектура статистических систем автоматической обработки речи.

- Модуль шумоочистки и отделение полезного сигнала.
- Акустическая модель — позволяет оценить распознавание речевого сегмента с точки зрения схожести на звуковом уровне. Для каждого звука изначально строится сложная статистическая модель, которая описывает произнесение этого звука в речи.
- Языковая модель — позволяют определить наиболее вероятные словесные последовательности. Сложность построения языковой модели во многом зависит от конкретного языка. Так, для английского языка, достаточно использовать статистические модели (так называемые N-граммы). Для высокофлективных языков (языков, в которых существует много форм одного и того же слова), к которым относится и русский, языковые модели, построенные только с использованием статистики, уже не дают такого эффекта — слишком много нужно данных, чтобы достоверно оценить статистические связи между словами. Поэтому применяют гибридные языковые модели, использующие правила русского языка, информацию о части речи и форме слова и классическую статистическую модель.
- Декодер — программный компонент системы распознавания, который совмещает данные, получаемые в ходе распознавания от акустических и языковых моделей, и на основании их объединения, определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания слитной речи.

Этапы распознавания

5. Обработка речи начинается с оценки качества речевого сигнала. На этом этапе определяется уровень помех и искажений.

6. Результат оценки поступает в модуль акустической адаптации, который управляет модулем расчета параметров речи, необходимых для распознавания.

7. В сигнале выделяются участки, содержащие речь, и происходит оценка параметров речи. Происходит выделение фонетических и просодических вероятностных характеристик для синтаксического, семантического и прагматического анализа. (Оценка информации о части речи, форме слова и статистические связи между словами.)

8. Далее параметры речи поступают в основной блок системы распознавания — декодер. Это компонент, который сопоставляет входной речевой поток с информацией, хранящейся в акустических и языковых моделях, и определяет наиболее вероятную последовательность слов, которая и является конечным результатом распознавания.

Основные понятия, которые характеризуют параметры речи человека, связанные с формой, размерами, динамикой изменения речеобразующего тракта и описывающие эмоциональное состояние человека, можно разделить на четыре группы объективных признаков, позволяющих различать речевые образцы: спектрально-временные, кекстральные, амплитудно-частотные и признаки нелинейной динамики. Подробнее, каждая группа признаков:

Спектрально-временные признаки

Спектральные признаки:

- Среднее значение спектра анализируемого речевого сигнала;
- Нормализованные средние значения спектра;
- Относительное время пребывания сигнала в полосах спектра;

- Нормализованное время пребывания сигнала в полосах спектра;
- Медианное значение спектра речи в полосах;
- Относительная мощность спектра речи в полосах;
- Вариация огибающих спектра речи;
- Нормализованные величины вариации огибающих спектра речи;
- Коэффициенты кросскорреляции спектральных огибающих между полосами спектра.

Временные признаки:

- Длительность сегмента, фонемы;
- Высота сегмента;
- Коэффициент формы сегмента.

Спектрально-временные признаки характеризуют речевой сигнал в его физико-математической сущности исходя из наличия компонентов трех видов:

4. периодических (тональных) участков звуковой волны;
5. непериодических участков звуковой волны (шумовых, взрывных);
6. участков, не содержащих речевых пауз.

Спектрально-временные признаки позволяют отражать своеобразие формы временного ряда и спектра голосовых импульсов у разных лиц и особенности фильтрующих функций их речевых трактов. Характеризуют особенности речевого потока, связанные с динамикой перестройки артикуляционных органов речи говорящего, и являются интегральными характеристиками речевого потока, отражающими своеобразие взаимосвязи или синхронности движения артикуляционных органов говорящего.

Кепстральные признаки

Мел-частотные кепстральные коэффициенты;

- Коэффициенты линейного предсказания с коррекцией на неравномерность чувствительности человеческого уха;
- Коэффициенты мощности частоты регистрации;
- Коэффициенты спектра линейного предсказания;
- Коэффициенты кепстра линейного предсказания.

Большинство современных автоматических систем распознавания речи сосредотачивают усилия на извлечении частотной характеристики речевого тракта человека, отбрасывая при этом характеристики сигнала возбуждения. Это объяснено тем, что коэффициенты первой модели обеспечивают лучшую разделимость звуков. Для отделения сигнала возбуждения от сигнала речевого тракта прибегают к кепстральному анализу.

Амплитудно-частотные признаки

- Интенсивность, амплитуда
- Энергия
- Частота основного тона (ЧОТ)
- Формантные частоты
- Джиттер (jitter) — дрожание частотная модуляция основного тона (шумовой параметр);
- Шиммер (shimmer) — амплитудная модуляция на основном тоне (шумовой параметр);

- Радиальная базисная ядерная функция
- Нелинейный оператор Тигер

Амплитудно-частотные признаки позволяют получать оценки, значения которых могут меняться в зависимости от параметров дискретного преобразования Фурье (вида и ширины окна), а также при незначительных сдвигах окна по выборке. Речевой сигнал акустически представляют собой распространяемые в воздушной среде сложные по своей структуре звуковые колебания, которые характеризуются в отношении их частоты (числа колебаний в секунду), интенсивности (амплитуды колебаний) и длительности. Амплитудно-частотные признаки несут необходимую и достаточную информацию для человека по речевому сигналу при минимальном времени восприятия. Но применение этих признаков не позволяет в полной мере использовать их в качестве инструмента идентификации эмоционально окрашенной речи.

Признаки нелинейной динамики

- Отображение Пуанкаре;
- Рекуррентный график;
- Максимальный характеристический показатель Ляпунова — Эмоциональное состояние человека, которому соответствует определенная геометрия аттрактора (фазовый портрет);
- Фазовый портрет (аттрактор);
- Размерность Каплана-Йорка — количественная мера эмоционального состояния человека, от «спокойствия» до «гнева» (деформация и последующее смещение спектра речевого сигнала).

Для группы признаков нелинейной динамики речевой сигнал рассматривается как скалярная величина, наблюдаемая в системе голосового тракта человека. Процесс речеобразования можно считать нелинейным и анализировать его методами нелинейной динамики. Задача нелинейной динамики состоит в нахождении и подробном исследовании базовых математических моделей и реальных систем, которые исходят из наиболее типичных предположений о свойствах отдельных элементов, составляющих систему, и законах взаимодействия между ними. В настоящее время методы нелинейной динамики базируются на фундаментальной математической теории, в основе которой лежит теорема Такенса (англ.)русск., которая подводит строгую математическую основу под идеи нелинейной авторегрессии и доказывает возможность восстановления фазового портрета аттрактора по временному ряду или по одной его координате. (Под аттрактором понимают множество точек или подпространство в фазовом пространстве, к которому приближается фазовая траектория после затухания переходных процессов.) Оценки характеристик сигнала из восстановленных речевых траекторий используются в построении нелинейных детерминированных фазово-пространственных моделей наблюдаемого временного ряда. Выявленные отличия в форме аттракторов можно использовать для диагностических правил и признаков, позволяющих распознать и правильно идентифицировать различные эмоции в эмоционально окрашенном речевом сигнале.

Параметры качества речи по цифровым каналам:

- Слоговая разборчивость речи;
- Фразовая разборчивость речи;
- Качество речи по сравнению с качеством речи эталонного тракта;
- Качество речи в реальных условиях работы.

Разборчивость речи — относительное количество правильно принятых элементов речи (звуков, слогов, слов, фраз), выраженное в процентах от общего числа переданных элементов.

- Качество речи — параметр, характеризующий субъективную оценку звучания речи в испытываемой системе передачи речи.
- Нормальный темп речи — произнесение речи со скоростью, при которой средняя длительность контрольной фразы равна 2,4 с.
- Ускоренный темп речи — произнесение речи со скоростью, при которой средняя длительность контрольной фразы равна 1,5-1,6 с.
- Узнаваемость голоса говорящего — возможность слушателей отождествлять звучание голоса, с конкретным лицом, известным слушателю ранее.
- Смысловая разборчивость — показатель степени правильного воспроизведения информационного содержания речи.
- Интегральное качество — показатель, характеризующий общее впечатление слушателя от принимаемой речи.

Основным преимуществом голосовых систем объявлялась дружелюбность к пользователю. Речевые команды должны были избавить конечного пользователя от необходимости использования сенсорных и иных методов ввода данных и команд.

- Голосовое управление
- Голосовые команды
- Голосовой ввод текста
- Голосовой поиск

Успешными примерами использования технологии распознавания речи в мобильных приложениях являются: ввод адреса голосом в Яндекс.Навигаторе, голосовой поиск Google Now.

Помимо мобильных устройств, технология распознавания речи находит широкое распространение в различных сферах бизнеса:

- Телефония: автоматизация обработки входящих и исходящих звонков путём создания голосовых систем самообслуживания в частности для: получения справочной информации и консультирования, заказа услуг/товаров, изменения параметров действующих услуг, проведения опросов, анкетирования, сбора информации, информирования и любые другие сценарии;
 - Решения "Умный дом": голосовой интерфейс управления системами «Умный дом»;
 - Бытовая техника и роботы: голосовой интерфейс электронных роботов; голосовое управление бытовой техникой и т.д;
 - Десктопы и ноутбуки: голосовой ввод в компьютерных играх и приложениях;
 - Автомобили: голосовое управление в салоне автомобиля — например, навигационной системой;
 - Социальные сервисы для людей с ограниченными возможностями.