

## Тема 7. "Интерфейсы и диалоги"

Введение1

1. Классификация интерфейсов1

1.1. Пакетная технология.2

1.2. Технология командной строки.2

1.3. Графический интерфейс3

1.3.1. Простой графический интерфейс.4

1.3.2. WIMP - интерфейс5

1.4. Речевая технология6

1.5. Биометрическая технология ("Мимический интерфейс").6

1.6. Семантический (Общественный) интерфейс.6

2. Диалог человек - ЭВМ7

2.1 Режимы работы с ЭВМ7

2.2 Правила общения10

2.3. Типы диалога13

Вопросы для самопроверки20

Вопросы реализации21

Пример программы шага диалога и структуры файла описания.21

### Введение

Как любое техническое устройство, компьютер обменивается информацией с человеком посредством набора определенных правил, обязательных как для машины, так и для человека. Эти правила в компьютерной литературе называются интерфейсом. Интерфейс может быть понятным и непонятным, дружественным и нет. К нему подходят многие прилагательные. Но в одном он постоянен: он есть, и никуда от него не денешься.

**Интерфейс, по определению - это правила взаимодействия операционной системы с пользователями, с другими ЭВМ и др.. От интерфейса зависит технология общения человека с компьютером.**

Как правило для каждого интерфейса разрабатывается набор средств и методов его реализации и логической организации с учетом требований задачи и, очень часто, с учетом психологии человека. В случае "общения" человека с ЭВМ эти средства и методы обычно называются диалогами.

### 1. Классификация интерфейсов

Как уже указывалось выше, интерфейс - это, прежде всего, набор правил. Как любые правила, их можно обобщить, собрать в "кодексы", сгруппировать по общему признаку. Таким образом возможно определить понятие "вид интерфейса" как объединение по схожести способов взаимодействия человека и компьютеров. Вкратце можно предложить следующую схематическую классификацию различных интерфейсов общения человека и компьютера.

Основными современными видами интерфейсов являются:

**1) Командный интерфейс.** Командный интерфейс называется так по тому, что в этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает

результат человеку. Командный интерфейс обычно реализован в виде пакетной технологии и технологии командной строки.

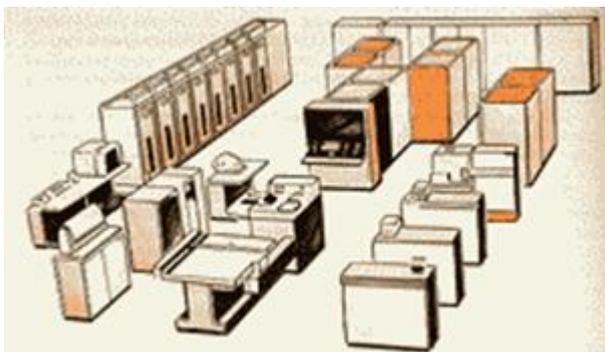
**2) WIMP - интерфейс (Window - окно, Image - образ, Menu - меню, Pointer - указатель).** Характерной особенностью этого вида интерфейса является то, что диалог с пользователем ведется не с помощью команд, а с помощью графических образов - меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "опосредственно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP - интерфейс.

**3) SILK - интерфейс (Speech - речь, Image - образ, Language - язык, Knowledge - знание).** Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения. В рамках этого интерфейса идет обычный "разговор" человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд он также преобразует в понятную человеку форму. Этот вид интерфейса наиболее требователен к аппаратным ресурсам компьютера, и поэтому его применяют в основном для военных целей.

### 1.1. Пакетная технология.

Исторически этот вид технологии появился первым. Она существовала уже на первых ЭВМ. Идея ее проста: на вход компьютера подается последовательность символов, в которых по определенным правилам указывается последовательность запущенных на выполнение программ. После выполнения очередной программы запускается следующая и т.д. Машина по определенным правилам находит для себя команды и данные. В качестве этой последовательности может выступать, например, перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки. Машина также выдает свои сообщения на перфоратор, алфавитно-цифровое печатающее устройство (АЦПУ), ленту пишущей машинки.

Такая машина представляет собой "черный ящик", в который постоянно подается информация и которая также постоянно "информирует" мир о своем состоянии (см. рисунок) Человек здесь имеет малое влияние на работу машины - он может лишь приостановить работу машины, сменить управляющую программу и вновь запустить ЭВМ. Впоследствии, когда машины стали помощнее и могли обслуживать сразу нескольких пользователей, вечное ожидание пользователей типа: "Я послал данные машине. Жду, что она ответит. И ответит ли вообще?" - стало, мягко говоря, надоедать. К тому же вычислительные центры, вслед за газетами, стали вторым крупным "производителем" макулатуры. Поэтому с появлением алфавитно-цифровых дисплеев началась эра по-настоящему пользовательской технологии - командной строки.

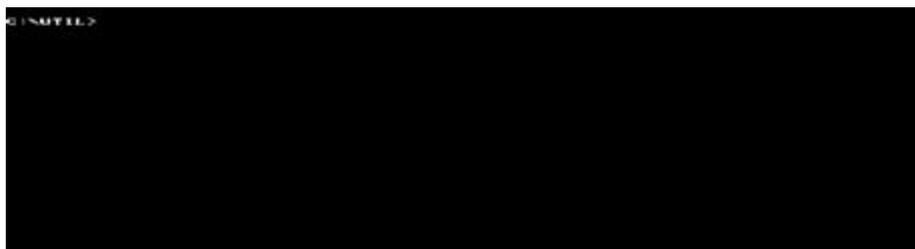


Вид большой ЭВМ серии ЕС ЭВМ.

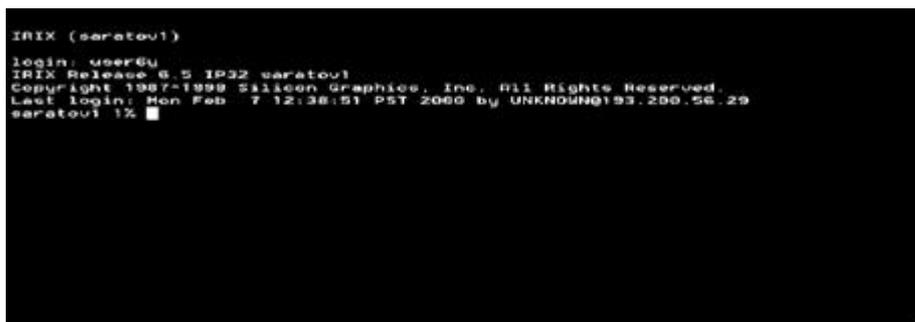
### 1.2. Технология командной строки.

При этой технологии в качестве единственного способа ввода информации от человека к компьютеру служит клавиатура, а компьютер выводит информацию человеку с помощью алфавитно-цифрового дисплея (монитора). Эту комбинацию (монитор + клавиатура) стали называть терминалом, или консолью.

Команды набираются в командной строке. Командная строка представляет собой символ приглашения и мигающий прямоугольник - курсор (см. рисунок 3.) При нажатии клавиши на месте курсора появляются символы, а сам курсор смещается вправо. Это очень похоже на набор команды на пишущей машинке. Однако, в отличие от нее, буквы отображаются на дисплее, а не на бумаге, и неправильно набранный символ можно стереть. Команда заканчивается нажатием клавиши Enter (или Return.) После этого осуществляется переход в начало следующей строки. Именно с этой позиции компьютер выдает на монитор результаты своей работы. Затем процесс повторяется.



а)



б)

Рис.3. Приглашения командной строки в различных операционных системах: а) *MS-DOS*, б) *IRIX*.

Технология командной строки уже работала на монохромных алфавитно-цифровых дисплеях. Поскольку вводить позволялось только буквы, цифры и знаки препинания, то технические характеристики дисплея были не существенны. В качестве монитора можно было использовать телевизионный приемник и даже трубку осциллографа.

Обе эти технологии реализуются в виде командного интерфейса - машине подаются на вход команды, а она как бы "отвечает" на них.

Преобладающим видом файлов при работе с командным интерфейсом стали текстовые файлы - их и только их можно было создать при помощи клавиатуры. На время наиболее широкого использования интерфейса командной строки приходится появление операционной системы UNIX и появление первых восьмиразрядных персональных компьютеров с много платформенной операционной системой CP/M.

### 1.3. Графический интерфейс

## Как и когда появился графический интерфейс?

Его идея зародилась в середине 70-х годов, когда в исследовательском центре Хероха Palo Alto Research Center (PARC) была разработана концепция визуального интерфейса. Предпосылкой графического интерфейса явилось уменьшение времени реакции компьютера на команду, увеличение объема оперативной памяти, а также развитие технической базы компьютеров. Аппаратным основанием концепции, конечно же, явилось появление алфавитно-цифровых дисплеев на компьютерах, причем на этих дисплеях уже имелись такие эффекты, как "мерцание" символов, инверсия цвета (смена начертания белых символов на черном фоне обратным, то есть черных символов на белом фоне), подчеркивание символов. Эти эффекты распространились не на весь экран, а только на один или более символов. Следующим шагом явилось создание цветного дисплея, позволяющего выводить, вместе с этими эффектами, символы в 16 цветах на фоне с палитрой (то есть цветовым набором) из 8 цветов. После появления графических дисплеев, с возможностью вывода любых графических изображений в виде множества точек на экране различного цвета, фантазии в использовании экрана вообще не стало границ! Первая система с графическим интерфейсом 8010 Star Information System группы PARC, таким образом, появилась за четыре месяца до выхода в свет первого компьютера фирмы IBM в 1981 году. Первоначально визуальный интерфейс использовался только в программах. Постепенно он стал переходить и на операционные системы, используемых сначала на компьютерах Atari и Apple Macintosh, а затем и на IBM -- совместимых компьютерах.

С более раннего времени, и под влиянием также и этих концепций, проходил процесс по унификации в использовании клавиатуры и мыши прикладными программами. Слияние этих двух тенденций и привело к созданию того пользовательского интерфейса, с помощью которого, при минимальных затратах времени и средств на переучивание персонала, можно работать с любыми программным продуктом. Описание этого интерфейса, общего для всех приложений и операционных систем, и посвящена данная часть.

Графический интерфейс пользователя за время своего развития прошел две стадии. Об эволюции графического интерфейса с 1974 по настоящее время будет рассказано ниже.

### **1.3.1. Простой графический интерфейс.**

На первом этапе графический интерфейс очень походил на технологию командной строки. Отличия от технологии командной строки заключались в следующем.

а) При отображении символов допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием. Благодаря этому повысилась выразительность изображения.

б) В зависимости от конкретной реализации графического интерфейса курсор может представляться не только мерцающим прямоугольником, но и некоторой областью, охватывающей несколько символов и даже часть экрана. Эта выделенная область отличается от других, невыделенных частей (обычно цветом).

в) Нажатие клавиши Enter не всегда приводит к выполнению команды и переходу к следующей строке. Реакция на нажатие любой клавиши во многом зависит от того, в какой части экрана находился курсор.

д) Кроме клавиши Enter, на клавиатуре все чаще стали использоваться "серые" клавиши управления курсором.

е) Уже в этой редакции графического интерфейса стали использоваться манипуляторы (типа мыши, трекбола и т.п. - см. рисунок 4.) Они позволяли быстро выделять нужную часть экрана и перемещать курсор.



Рис. 4. Манипуляторы

Подводя итоги, можно привести следующие отличительные особенности этого интерфейса.

- 1) Выделение областей экрана.
- 2) Переопределение клавиш клавиатуры в зависимости от контекста.
- 3) Использование манипуляторов и серых клавиш клавиатуры для управления курсором.
- 4) Широкое использование цветных мониторов.

Появление этого типа интерфейса совпадает с широким распространением операционной системы MS-DOS. Именно она внедрила этот интерфейс в массы, благодаря чему 80-е годы прошли под знаком совершенствования этого типа интерфейса, улучшения характеристик отображения символов и других параметров монитора.

Типичным примером использования этого вида интерфейса является файловая оболочка Norton Commander и текстовый редактор Multi-Edit. А текстовые редакторы Лексикон, ChiWriter и текстовый процессор Microsoft Word for Dos являются примером, как этот интерфейс превзошел сам себя.

### 1.3.2. WIMP - интерфейс

Вторым этапом в развитии графического интерфейса стал "чистый" интерфейс WIMP, Этот подвид интерфейса характеризуется следующими особенностями.

1. Вся работа с программами, файлами и документами происходит в окнах - определенных очерченных рамкой частях экрана.
2. Все программы, файлы, документы, устройства и другие объекты представляются в виде значков - иконок. При открытии иконки превращаются в окна.
3. Все действия с объектами осуществляются с помощью меню. Хотя меню появилось на первом этапе становления графического интерфейса, оно не имело в нем главенствующего значения, а служило лишь дополнением к командной строке. В чистом WIMP - интерфейсе меню становится основным элементом управления.
4. Широкое использование манипуляторов для указания на объекты. Манипулятор перестает быть просто игрушкой - дополнением к клавиатуре, а становится основным элементом управления. С помощью манипулятора **УКАЗЫВАЮТ** на любую область экрана, окна или иконки, **ВЫДЕЛЯЮТ** ее, а уже потом через меню или с использованием других технологий осуществляют управление ими.

Следует отметить, что WIMP требует для своей реализации цветной растровый дисплей с высоким разрешением и манипулятор. Также программы, ориентированные на этот вид интерфейса, предъявляют повышенные требования к производительности компьютера, объему его

памяти, пропускной способности шины и т.п. Однако этот вид интерфейса наиболее прост в усвоении и интуитивно понятен. Поэтому сейчас WIMP - интерфейс стал стандартом де-факто.

Ярким примером программ с графическим интерфейсом является операционная система Microsoft Windows.

#### 1.4. Речевая технология

С середины 90-х годов, после появления недорогих звуковых карт и широкого распространения технологий распознавания речи, появился так называемый "речевая технология" SILK - интерфейса. При этой технологии команды подаются голосом путем произнесения специальных зарезервированных слов - команд. Основными такими командами (по правилам системы "Горыныч") являются:

- - "Проснись" - включение голосового интерфейса.
- "Отдыхай" - выключение речевого интерфейса.
- "Открыть" - переход в режим вызова той или иной программы.

Имя программы называется в следующем слове.

- - "Буду диктовать" - переход из режима команд в режим набора текста голосом.
- "Режим команд" - возврат в режим подачи команд голосом.
- и некоторые другие.

Слова должны выговариваться четко, в одном темпе. Между словами обязательна пауза. Из-за незрелости алгоритма распознавания речи такие системы требуют индивидуальной предварительной настройки на каждого конкретного пользователя.

"Речевая" технология является простейшей реализацией SILK - интерфейса.

#### 1.5. Биометрическая технология ("Мимический интерфейс".)

Эта технология возникла в конце 90-х годов XX века и на текущий момент еще разрабатывается. Для управления компьютером используется выражение лица человека, направление его взгляда, размер зрачка и другие признаки. *Для идентификации пользователя используется рисунок радужной оболочки его глаз, отпечатки пальцев и другая уникальная информация.* Изображения считываются с цифровой видеокамеры, а затем с помощью специальных программ распознавания образов из этого изображения выделяются команды. Эта технология, по-видимому, займет свое место в программных продуктах и приложениях, где важно точно идентифицировать пользователя компьютера.

#### 1.6. Семантический (Общественный) интерфейс.

Этот вид интерфейса возник в конце 70-х годов XX века, с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса - он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейс. Основная его отличительная черта - это отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По своей сути это трудно называть интерфейсом - это уже моделирование "общения" человека с компьютером.

С середины 90-х годов XX века редко встречаются публикации, относящихся к семантическому интерфейсу. Похоже, что в связи с важным военным значением этих разработок (например, для автономного ведения современного боя машинами - роботами, для "семантической" криптографии) эти направления были засекречены. Информация, что эти

исследования продолжаются, иногда появляется в периодической печати (обычно в разделах компьютерных новостей).

## 2. Диалог человек - ЭВМ

### 2.1 Режимы работы с ЭВМ

Взаимодействие человека с ЭВМ не сразу приняло диалоговые формы, наиболее распространенные в настоящее время. Исторически диалоговому режиму общения с вычислительной машиной предшествовали **системы пакетного**, или закрытого, типа.

**Пакетный режим** характеризуется тем, что задачи готовятся автономно от ЭВМ и собираются в *пакет* — массив входных заданий. Задания вводятся в машину оператором ЭВМ по одному в порядке очереди в пакете. Последовательность выполнения программ и их отдельных шагов задается системой таким образом, чтобы эффективность использования центральных, наиболее дорогих, устройств ЭВМ была максимальной. При работе в пакетном режиме программист связан с вычислительным процессом косвенно, а промежуток времени между подачей задачи на машину и получением результата составляет обычно часы или даже дни.

В качестве носителей информации обычно используются перфокарты, реже — перфоленты. В последнее время в качестве носителей информации при работе в пакетном режиме применяются магнитные ленты или гибкие магнитные диски. Часто используются электронные письма

*При пакетной обработке достигается максимальная эффективность использования ЭВМ. Такой режим обслуживания минимизирует нерациональную загрузку средств, вычислительной техники.*

Широкое распространение вычислительных машин снизило актуальность максимальной загрузки ЭВМ и выдвинуло на первый план повышение эффективности работы пользователя, главным образом непрофессионального. С решением этой задачи связано появление и развитие **диалогового режима работы** на ЭВМ.

**Диалоговый режим** общения с вычислительной машиной предполагает наличие диалоговой системы, с помощью которой пользователь получает непосредственный доступ к ЭВМ. Как правило, он делит машинные ресурсы (центральный процессор, память, внешние устройства, программное обеспечение ЭВМ) с другими пользователями или программными процессами, численность и состав которых могут меняться более или менее случайным образом. В процессе обработки задачи пользователь имеет возможность непосредственно и оперативно контролировать работу программы, менять решения и разнообразить действия при общении с ЭВМ.

Возможности диалоговой системы и удобство работы с ней во многом зависят от *гибкости пользования*, определяемой числом различных процессов, которые могут выполняться с помощью системы.

Организация работы с машиной в диалоговом режиме предполагает общение с ней на языках, приближенных к естественному, со скоростью, близкой к темпу диалога с человеком, причем в некоторых случаях на уровне операционной сложности, адаптируемой в процессе общения к опыту и квалификации пользователя.

**Операционная сложность** означает сложность действий (операций), которые должен выполнить пользователь для решения поставленной задачи.

Для работы *начинающих пользователей* достаточно иметь системы с незначительной гибкостью пользования и обязательно с минимально допустимой операционной сложностью.

**Сравнение диалогового и пакетного режимов** не дает явных преимуществ ни одному из них безотносительно типа решаемых на ЭВМ задач и квалификации пользователей, однако в ряде случаев диалоговый режим имеет неоспоримые преимущества.

Исторически более ранний пакетный режим ориентирован, в первую очередь, на повышение загрузки центральных устройств ЭВМ. Действительно, при использовании этого режима процессор загружен выполнением прикладных («полезных») программ примерно на 40—50% больше, чем в диалоговом режиме. Общее время отладки прикладных программ меньше (иногда в несколько раз) в диалоговом режиме, однако суммарные затраты машинного времени в этом режиме в 4—6 раз больше, чем в пакетном.

Задача практического освоения пакетного режима плохо разбивается на мелкие составляющие. Поэтому при его использовании многие начинающие бывают настолько обескуражены требуемым объемом начальных знаний и первыми результатами, что прекращают работу на ЭВМ.

***Диалоговый режим лучше приспособлен к работе начинающих и непрофессиональных пользователей, чем пакетный.***

Время освоения ЭВМ в диалоговом режиме, как правило, меньше, чем в пакетном. Диалог повышает энтузиазм и активность обучающихся, в том числе учащихся со средними способностями, что особенно важно при широком внедрении вычислительной техники. Хорошие результаты достигаются при его использовании и среди профессиональных пользователей.

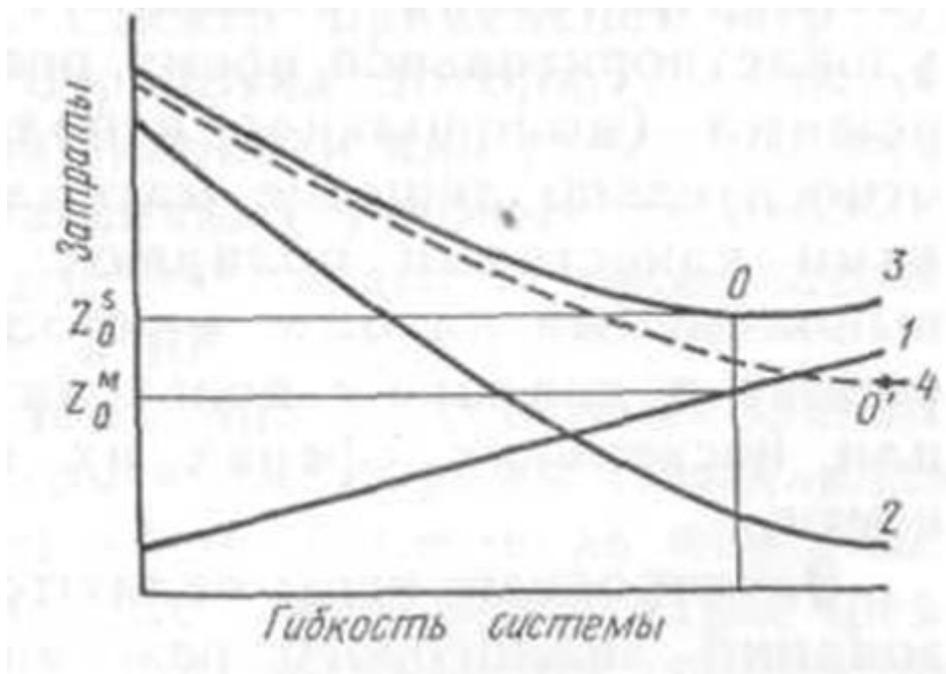
Диалоговый режим нивелирует индивидуальные различия в ошибках и времени решения задач в системе «человек — ЭВМ» по сравнению с пакетным режимом.

Присущие рассмотренным режимам положительные качества объединяются в современных вычислительных системах, когда вместе с пользователями, работающими в диалоговом режиме, на вычислительной машине выполняется одна или несколько «фоновых» пакетных задач.

**Экономический аспект** затрат ресурсов следует оценивать с точки зрения как экономии времени пользователя, так и эффективности работы ЭВМ.

Увеличение машинных ресурсов может оказаться необходимым не только для повышения эффективности деятельности пользователя, но и для улучшения технико-экономических показателей системы «человек — ЭВМ».

Рассмотрим, как меняются затраты на эту систему с развитием функциональных характеристик машинной составляющей (рис. 1).



Кривая 1 соответствует суммарным затратам на разработку программной и аппаратной частей системы и ее эксплуатацию;

кривая 2 — непосредственным затратам пользователя на общение с ЭВМ для получения требуемого результата;

кривая 3 соответствует полным затратам на разработку и эксплуатацию человеко-машинной системы.

Минимум этих затрат приходится на точку 0. Поэтому при затратах на машинную часть системы, меньших  $Z_0^M$ , общие затраты на систему будут превышать минимальное значение  $Z_0^S$ . Таким образом, дополнительные затраты на создание удобных аппаратных и программных средств, ориентированных на ведение активного диалога, могут снизить общие затраты в системе «человек — ЭВМ».

Средства вычислительной техники активно развиваются. Появляются новые технологии производства компонентов ЭВМ и разработки программного обеспечения, внедряются перспективные архитектурные и схемотехнические решения при проектировании и производстве вычислительной техники. Это ведет к существенному росту эффективности производства машинного компонента системы «человек — ЭВМ». В то же время стоимость часа человеческого труда в обществе со временем возрастает. Поэтому результирующая кривая затрат (рис. 1, кривая 3) сместится таким образом, что новый минимум затрат окажется на графике правее и несколько ниже старого (точка 0', кривая 4). Таким образом, доля машинных затрат и гибкость вновь разрабатываемых систем «человек — ЭВМ» также объективно должно возрастать.

В диалоговых системах, реализованных на больших машинах, значительная часть затрат на разработку программного обеспечения приходится на поддержание режима разделения времени, создающего у пользователей иллюзию монопольной работы с ЭВМ. В то же время доля программного обеспечения в стоимости универсальных ЭВМ достигает более 100% при незначительном росте производительности разработки программ (по некоторым источникам она не превышает 3% в год).

С развитием микроэлектроники оказалось возможным объединить положительные качества диалогового режима (непосредственное общение с вычислительной машиной и удовлетворительное время реакции системы) и пакетного режима (монопольное использование ресурсов ЭВМ и относительно дешевое математическое обеспечение). Такими качествами обладают персональные компьютеры, выполняемые на базе микроЭВМ и ориентированные на

работу в диалоге с пользователями в какой-либо одной или нескольких сферах их профессиональной деятельности.

## 2.2 Правила общения

Использование ЭВМ связано с решением разнообразных технических, математических и организационных проблем. Ими с большим или меньшим успехом занимаются специалисты в области вычислительной техники. Значительно менее исследованными остаются *психологические и социальные факторы*.

*В то же время исследованиями установлено, что различия в эффективности деятельности отдельных пользователей перекрывают различия, обусловленные типом и характеристиками используемой вычислительной системы.*

Поэтому в человеко-машинных системах в первую очередь важны не потенциальные возможности аппаратно - программной компоненты по решению прикладных задач, а предоставляемые этими системами сервисные возможности для обеспечения эффективной работы пользователей, в том числе непрофессиональных.

Разрабатываемые диалоговые системы в значительной степени ориентированы на потребности непрофессиональных пользователей. Созданы даже такие системы, которые имитируют привычную рабочую обстановку пользователя: письменный стол, папки с бумагами и т. д. В этих условиях важно умение в полной мере эффективно использовать предоставляемые системой возможности. Кроме чисто профессиональных качеств для достижения положительных результатов при общении с ЭВМ важны и *личные качества пользователя*. В наибольшей степени способствуют успеху, *настойчивость, интравертированность* (обращенность вовнутрь, ориентация на собственные идеи), *внутренняя управляемость пользователя, высокая терпимость к неопределенности* (особенно на начальном этапе работы с машиной), *умение быть точным* (особенно на завершающей стадии работы), *скромность, способность переносить стресс*.

Общению между людьми и между человеком и ЭВМ присущи как отличительные, так и общие черты.

Выяснив, какие правила ведения диалога являются общими а какие — разными, можно значительно быстрее освоить диалоговую систему.

**Под диалогом будем понимать непосредственный соизмеримый по скорости с процессом мышления обмен сообщениями между двумя субъектами для достижения определенной цели, при котором в любой момент времени возможна смена ролей информатора и реципиента (приемника информации).**

Особенности ведения диалога между людьми в некоторых случаях могут быть распространены на диалог в системе «человек — ЭВМ».

Вступая в общение друг с другом, участники диалога преследуют определенные цели:

- пытаются получить ответ на какой-либо вопрос,
- поделиться тем, что их волнует,
- произвести определенное впечатление,
- просто «скоротать» время.

Многие, кроме того, в важных случаях заранее продумывают «сценарий» диалога, с тем чтобы с большим успехом достичь намеченной цели.

Выделим следующие **основные особенности ведения диалога**.

1. *У вступающих в общение есть определенная цель и, возможно, заранее определен «сценарий» диалога.*  
В процессе общения обычно выясняются цели, мнения участников диалога по обсуждаемой проблеме, выявляется общность и отличие взглядов. Как правило, минимизируются отличия в подходах к рассматриваемой проблеме и даже в отношении к ней, в том числе и путем изменения «сценария» диалога.
2. *Участники диалога адаптируются друг к другу.*  
В процессе диалога или на отдельных его этапах каждый из собеседников выполняет определенную разговорную роль: один спрашивает, другой отвечает; один активно пытается склонить другого к своей точке зрения, второй отстаивает свое мнение и т. д.
3. *Происходит распределение разговорных ролей, каждый из участников диалога выполняет определенные обязательства по отношению к теме общения.*  
В процессе ведения диалога разговорные роли могут неоднократно перераспределяться. Они подчиняются нормам поведения и лингвистическим правилам, т. е. принимают определенные формы, которые, в известной степени, оказывают влияние на результат отдельных фрагментов или диалога в целом.
4. *Принятые формы диалога и распределение разговорных ролей ведут к определенной предсказуемости результатов диалога.*  
Известная предсказуемость результатов диалога активизирует его участников на вступление в те общения, которые потенциально ведут к положительному результату и, наоборот, заставляют избегать контактов, которые могут привести к нежелательным последствиям. Усилия, затрачиваемые на вступление в общение и ведение диалога, в определенной степени пропорциональны важности предполагаемых результатов для его участников.
5. *Участники диалога мобилизуют свои усилия, предпочитая вступить в общения с заведомо определенным положительным результатом.*  
При этом зачастую даже ожидаемый положительный результат может быть нивелирован предполагаемыми при общении отрицательными эмоциями, которые могут заставить одного из потенциальных участников диалога отказаться от общения.  
Кроме внутренней логики, присущей сценарию диалога, успех общения во многом зависит от того, в каком настроении вступают в диалог его участники. Поэтому вначале важно выяснить это, добиться доброжелательного отношения к себе и заинтересовать в предполагаемой теме общения второго участника диалога.  
Чрезвычайно важно и самому относиться доброжелательно к собеседнику. Особое значение для каждого из участников диалога имеет оценка его разговорной роли собеседником. Поэтому ведение диалога в положительном эмоциональном тоне, возможно ближе к интересам собеседника, с максимально высокой объективной оценкой его роли с большей вероятностью ведет к достижению желаемого результата.
6. *Люди эмоциональны и преувеличенно чувствительны к недооценке их роли, в значительной мере форма общения оказывается для них важнее содержания.*  
Поведение участников диалога существенно зависит от внешнего вида собеседника и его умения держаться.

Кроме отмеченных существуют, конечно, и другие особенности ведения диалога между людьми. Здесь были отмечены только наиболее существенные для целей организации диалога в системе «человек — ЭВМ».

Пункты 1—5 присущи как диалогу в человеко-машинной системе, так и общению между людьми. Пункт 6 свойствен диалогу в системе «человек — ЭВМ» только в той степени, в которой разработчики диалоговой системы сумели передать свои эмоции вычислительной машине. При этом ведение диалога с ЭВМ регламентируется более строгими правилами и ограничениями, чем диалог между людьми.

**Диалог человек — ЭВМ представляет собой разновидность диалога, где в качестве одного из субъектов выступает человек, в качестве другого — диалоговая система, и обмен интерактивными посланиями ведется в соответствии с заранее определенными языком и формой общения.**

Для достижения максимального эффекта общения с ЭВМ важно не только построить свой диалог в соответствии с инструкцией по работе с системой, но и тщательно подготовиться к выходу на машину и грамотно проанализировать результаты сеанса работы с ЭВМ. Поэтому полезно рассмотреть действия пользователя на этих трех фазах общения с вычислительной машиной вне зависимости от типа используемой диалоговой системы.

Перед **работой** с системой надо прежде всего выяснить ее название, функциональные возможности, режим работы диалоговой системы, **поддерживаемый** вычислительным центром (ВЦ). Полезно также выяснить, к кому обращаться в случае возникновения организационных, технических и других затруднений, а также принятый порядок **консультаций**. При изучении описания диалоговой системы надо всегда помнить: для решения какой (или каких) прикладной задачи предполагается **использовать** данную систему. Если изучаемая диалоговая система для этих целей непригодна, следует выяснить, какие еще системы поддерживаются. До освоения диалоговой системы лучше ее не критиковать, чтобы не попасть в неудобное положение из-за плохого знания объекта критики. При освоении системы особое внимание надо уделить уточнению своих действий в тупиковых ситуациях.

Во время работы с системой пользователь должен помнить, что она защищена от ошибочных действий, которые могут ее разрушить или причинить пользователям большой ущерб. Если система все-таки будет разрушена — значит она просто неудачно спроектирована, и в следующих версиях разработчики должны учесть эту недоработку.

В процессе работы может возникнуть ситуация, когда пользователю не ясно, что от него требуется, каковы его дальнейшие действия. В большинстве диалоговых систем реализованы функции помощи, обратившись к которым пользователь может преодолеть свои затруднения.

При работе с системой следует обращаться к машине за помощью. Важно помнить, что машина помогает пользователю только в рутинной части работы, оставляя ему творческую часть работы и принятие решений. Кроме того, у вычислительной машины отсутствуют эмоции, поэтому многократные повторы и «непонятливость» пользователя, как правило, не отражаются на состоянии диалога с ЭВМ.

Диалоговая система может работать в темпе, превышающем потребности пользователя. В этом случае следует работать в своем темпе, просматривая все сообщения системы и не «гонясь» за ЭВМ.

Новые сообщения, выдаваемые системой, следует внимательно изучить, выделив те из них, которые влияют на дальнейший ход диалога. При неоднократной неудаче полезно переформулировать запрос к ЭВМ, подойти к проблеме с другой стороны. В случае неудачной попытки выполнить функционально сложный запрос следует разбить его на более элементарные действия, что дает, как правило, хороший результат.

Понимание логики построения машинного диалога поможет пользователю, получив удачный вариант взаимодействия в одном режиме и проведя аналогию, с большим успехом достичь результата в похожей ситуации другого режима работы системы.

При отсутствии новых идей по отладке программы следует прекратить бесплодные попытки, чтобы не испортить полученные за сеанс предварительные результаты. Не следует пытаться решить все проблемы за экраном: опыт показывает, что это, как правило, малоэффективно. Наиболее правильные решения пользователь находит обычно за столом.

В конце сеанса необходимо получить распечатки, файлы протоколов и т.п., которые могут понадобиться для анализа результатов работы за экраном дисплея. Это могут быть: текст последней версии программы, результаты тестирования и т. д.

После окончания работы с системой необходимо проанализировать результаты, если этого не было сделано во время сеанса, на порядок полученных величин. Ошибочные результаты машинного расчета в большинстве случаев возникают из-за ошибок пользователя. Крайне редко они появляются или из-за машинного сбоя или из-за выхода ЭВМ из строя.

Необходимо также анализировать весь ход диалога с ЭВМ, особенно в случае отрицательного результата, найти свои ошибки, наметить пути их устранения: более детально ознакомиться с документацией по системе, проконсультироваться со специалистами, разработать новую стратегию ведения диалога и т. д. После реализации намеченных мероприятий следует подготовить материал к следующему сеансу, а также план работы за дисплеем.

Итак, резюмируя правила ведения диалога, можно посоветовать выполнение следующих **рекомендаций как при работе с ЭВМ, так и в общении с людьми:**

- *знайте, что вы хотите;*
- *умейте слушать собеседника;*
- *пользуйтесь помощью при затруднениях;*
- *обучайтесь в процессе диалога;*
- *перестраивайтесь в процессе диалога при неудачной попытке установить контакт;*
- *анализируйте результаты общения.*

### 2.3. Типы диалога

В прикладной диалоговой системе можно выделить два функциональных компонента:

- *собственно прикладную систему, с которой работает пользователь;*
- *диалоговый компонент, управляющий диалогом между пользователем и прикладной системой.*

Примерами прикладных систем являются все системы, с которыми работают пользователи ЭВМ в сфере своих профессиональных интересов, — различного рода автоматизированные системы управления (АСУ), системы автоматизированного проектирования (САПР), пакеты прикладных программ (ППП) и др.

Эти системы характеризуются тем, что диалог ведется в терминах проблемной области и на выходе пользователь получает конечный результат, являющийся целью его общения с вычислительной машиной. Некоторые системы могут работать не только в диалоговом, но и в пакетном режиме. Семантика конечного результата при этом не зависит от используемого режима работы.

Диалоговый компонент является *надстройкой над прикладной системой* и помогает пользователю уточнить постановку проблемной задачи, предоставляя для этого соответствующие лингвистические формы и набор сервисных функций. Диалог в этом случае ведется в терминах диалоговой системы и называется *метадиалогом*.

Используемая терминология практически не зависит от проблемной области. Наличие в прикладной системе развитого метадиалога дает возможность пользователю, знакомому только с проблемной областью, получать требуемые результаты без предварительного детального изучения диалоговой системы. Поэтому доступность и распространенность использования прикладной системы во многом зависят от качества построения диалоговой надстройки.

С учетом наличия двух компонентов в диалоговой системе на рис. 2 представлена обобщенная схема алгоритма простейшего диалога с машиной. При этом однократное прохождение по одной из ветвей схемы алгоритма от блока «Начало» до блока «Конец» или до возврата к началу схемы алгоритма будем называть *шагом диалога или транзакцией*. Шаг диалога характеризуется интерактивным взаимодействием человека с ЭВМ, т. е. в данном случае в ответ на ввод сообщения пользователя после его обработки диалоговой системой выдается выходное сообщение, отражающее состояние системы или диалога.

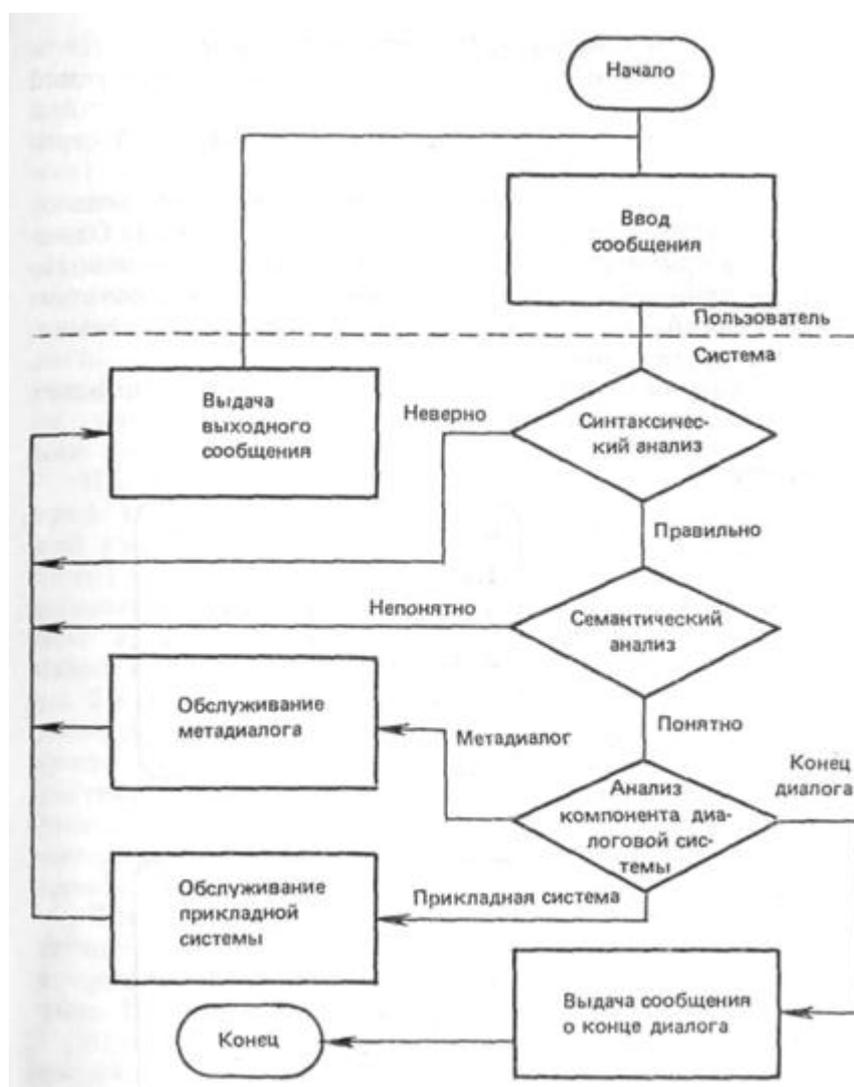


Рис. 2

Для более быстрого освоения диалоговой системы пользователю необходимо знать *основные типы диалога* и формы его реализации на вычислительной машине.

Наиболее известны в человеко-машинной среде следующие типы диалога:

- — **меню;**
- — **вопросы, требующие ответа ДА/НЕТ;**
- — **шаблон;**
- — **простой запрос;**
- — **команда;**
- — **взаимодействие на естественном языке.**

**Меню** — наиболее распространенный тип диалога. При обращении система явно или косвенно инициирует диалог. Таким образом, шаг диалога начинается с выдачи системой входного сообщения. После выдачи пользователем выходного сообщения и обработки его системой шаг диалога завершается.

Возможно несколько форм машинной реализации данного типа диалога. Во всех случаях в качестве входного сообщения на экране дисплея высвечивается подмножество функций системы, реализация которых возможна в текущем состоянии диалога. Такое входное сообщение будем называть **меню**.

Пользователь может выбрать требуемую функцию одним из следующих основных способов:

- а) набором требуемой директивы или ее аббревиатуры;
- б) набором номера выбранной директивы;
- в) подведением курсора в строку экрана с требуемой директивой;
- г) нажатием фотоселектора в соответствующей строке экрана.

Наиболее экономичным по затратам времени пользователя является способ г), затем способы б) и в). Однако иногда применяется способ а), так как при его использовании происходит подтверждение директивы и возможность случайного набора соседней директивы меньше, чем при других способах.

Графически диалог можно представить в виде **графа**. Его узлы соответствуют подмножеству функций системы (входные сообщения), а дуги, направленные вниз, — возможным альтернативам пользователя (выходные сообщения). При этом число дуг равно числу предлагаемых системой функций. Дуга вместе с вершиной, из которой она исходит, соответствует шагу диалога. Верхний узел (корень) дерева соответствует начальному сообщению системы.

На рис. 3 приведен граф диалога меню в виде дерева. Корень дерева (верхний узел, соответствующий начальному сообщению системы) и исходящие из него дуги соответствуют шагу диалогов; предлагаемые в меню функции — дугам дерева в направлении слева направо.

Движение вниз по дереву графа меню в направлении прикладной системы называется «погружением».

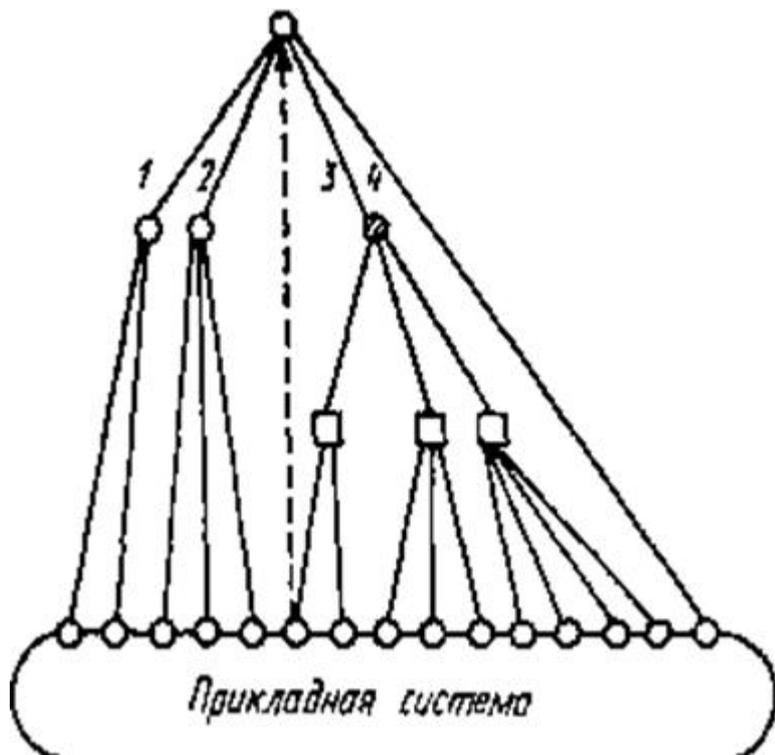


Рис. 3.

Для выполнения функции, соответствующей соседним ветвям дерева графа, при метадиалоге или после работы с прикладной системой используется механизм «всплытия». Его можно реализовать двумя способами:

а) последовательным движением вверх по дугам графа диалога до узла, непосредственно содержащего заданную функцию или содержащего ее в узлах, в которые из него можно перейти, используя «погружение»;

б) непосредственным переходом в корень дерева графа диалога с последующим погружением.

Графически второй способ соответствует наличию в каждом узле ориентированной к вершине дерева дуги графа. Одна из таких дуг изображена на рис. 2.3 штриховой линией.

Механизм «всплытия» может быть представлен пользователю одним из двух способов:

а) его описание содержится только в документации на систему (в этом случае из любой вершины «всплытие» происходит, как правило, единообразно);

б) кроме документации одна или несколько альтернатив каждого меню системы описывает механизм «всплытия».

**Вопросы, требующие ответа ДА/НЕТ**, являются частным случаем диалога типа «меню», когда в шаге диалога в качестве альтернатив пользователю предлагаются два ответа: ДА и НЕТ.

В диалоговой системе возможны разные реализации ответа пользователя:

- а) ответ дается с помощью первых букв слов ДА и НЕТ;
- б) в качестве ответа используются какие-нибудь символы, соответствующие ДА и НЕТ (например, 1 и 0);

- в) ответ набирается полностью (ДА или НЕТ);
- г) одна из альтернатив (ДА или НЕТ) предполагается по умолчанию в комбинации с одной из трех описанных реализаций.

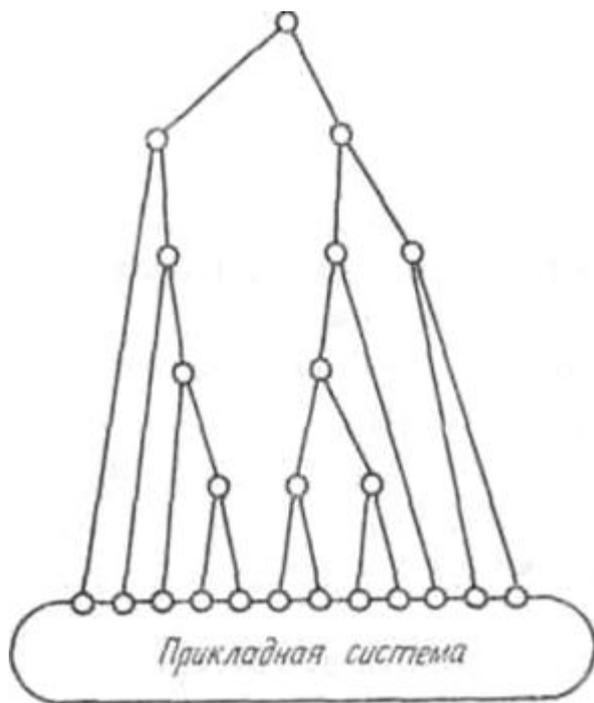


Рис 4.

Графическая интерпретация рассматриваемого типа диалога изображена на рис. 4 — это граф бинарного дерева, из каждого узла которого выходит вниз (к прикладной системе) по две ветви. «Погружение» и «всплытие» в этом типе диалога аналогичны меню. В данном типе диалога для работы с прикладной системой приходится выполнять, как правило, больше шагов по сравнению с диалогом типа «меню». Основным достоинством вопросов, требующих ответа ДА/НЕТ, является простота выполнения каждого отдельного шага диалога.

**Шаблон** — это инициированный системой диалог, на каждом шаге которого система воспринимает только синтаксически ограниченное входное сообщение пользователя в соответствии с заданным форматом. Как и в меню, диалог начинает система. Данный тип диалога называют также параметрической системой. Возможные варианты ответа пользователя ограничиваются форматами, предъявляемыми ему на дисплее. Поэтому гибкость пользования системой с диалогом такого типа относительно невысока, однако достаточно низка ее операционная сложность.

Из нескольких возможных вариантов реализации такого типа диалога назовем два:

- 1) указание системой на экране дисплея формата вводимого пользователем сообщения;
- 2) резервирование места для сообщения пользователя в тексте сообщения системы на экране дисплея (например, с помощью поля повышенной яркости).

Шаблон обычно используется для ввода данных, значения которых или «прозрачны» (например, значение пола при заполнении анкеты или текущая дата), или являются профессиональными терминами и поэтому они известны пользователю. Однако формы подачи этих данных не определяются их значениями. Например, дата 1 июля 1987 года может быть введена в систему в записанном виде и в более компактных форматах: 87/182 (182-й день 1987 года), 87/07/01, 01/07/87 и др. Символ «/» здесь используется в качестве разделителя. Значение пола можно ввести полностью, например: «мужской» сокращенно — «муж» и по первой букве — «М»

Графически диалог типа «шаблон» можно интерпретировать с помощью графа, приведенного на рис. 5.

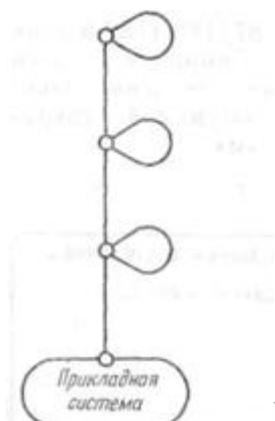


Рис. 5.

Из каждой вершины графа выходит по две дуги. Дуга, направляющаяся к следующему узлу, соответствует семантически правильному сообщению пользователя, а дуга, замыкающаяся на одном узле, — семантически неправильному сообщению пользователя. При неверном сообщении система выдаст предупреждение об ошибке и останется в начале текущего шага диалога.

На рис. 6 проиллюстрирована реализация диалога с помощью двух типов диалога: меню и шаблона. Из рисунка видно, что в данном случае более компактным и удобным для пользователя будет диалог типа «шаблон». Каждому фрагменту сообщения, заданному своей частью формата, соответствует шаг диалога меню.

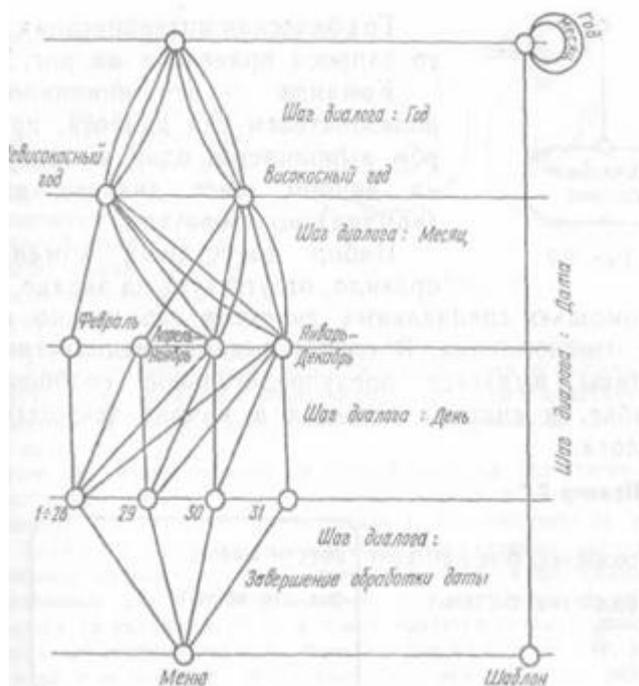


Рис 6.

**Простой запрос** является частным случаем шаблона. Обычно используется в различного рода автоматизированных системах при сборе данных. Пользователю предоставляется возможность вводить массив данных по формату, заданному системой. Весь диалог сводится к одному шагу. Формат может выводиться на экран, а может быть описан только в руководстве по работе с системой.

Простой запрос — это тип диалога, инициируемый системой и состоящий из одного шага, где пользователю предоставляется возможность ввести более одного сообщения по заранее

определенному системой единому формату. Диалог завершается «всплытием», которое выполняется при задании пользователем определенной директивы. Графическая интерпретация простого запроса приведена на рис.7.



Рис. 7.

**Команда** — это инициированный пользователем тип диалога, при котором выполняется одна из допустимых на данном шаге диалога директив (команд) пользователя. Набор допустимых команд, как правило, отсутствует на экране, однако с помощью специальных директив его можно вывести для ознакомления. В случае задания недопустимой директивы выдается предупредительное сообщение об ошибке, и система остается в начале текущего шага диалога.

Графически данный тип запроса может быть представлен с помощью ориентированного графа (рис. 8). Характерным является наличие дуг, замыкающихся на каждом узле графа. Такие дуги соответствуют недопустимым директивам на данном шаге диалога.

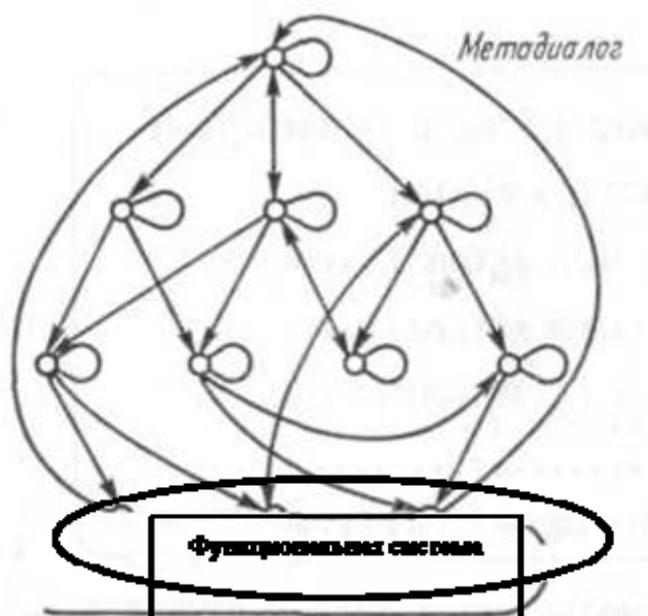


Рис. 8.

**Взаимодействие на естественном языке** возникает по инициативе пользователя. Этот тип диалога налагает наименьшие ограничения на форму ведения общения со стороны пользователя, которому предоставляется возможность свободно выбирать формулировку задачи, используя естественный язык. В связи с неоднозначностью естественных языков со стороны системы возможны вопросы, уточняющие формулировки пользователя и предметную область рассматриваемой проблемы.

Взаимодействием на естественном языке будем называть тип диалога инициированный пользователем и приводящий к решению поставленной им задачи, когда общение со стороны пользователя ведется на естественном языке.

Научные и технические трудности создания системы, взаимодействующей с пользователем на естественном языке, приводят к реализации квази естественного (похожего на естественный) или проблемно-ориентированного (для общения в одной узкой проблемной области) языка. Поэтому пользователь должен быть готов к тому, что диалоговая система не распознает запрос с первого раза (если распознает его вообще). Развитие диалоговых систем в направлении повышения их интеллектуальности приводит к повышению их гибкости и адаптивности, а следовательно, и большей восприимчивости к запросам на квази естественном языке.

Графическая интерпретация взаимодействия на естественном и квази естественном языках приведена на рис 9, а, б соответственно.

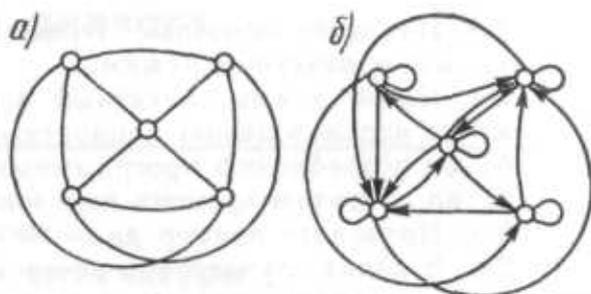


Рис. 9.

Отличительной особенностью графа диалога на естественном языке является наличие парных противоположно направленных дуг графа между любыми парами узлов (на рисунке каждая пара противоположно направленных дуг обозначена одной неориентированной дугой). Граф диалога на квазиестественном языке является ориентированным, характерная его особенность — наличие в каждом узле дуг, замыкающихся на этом же узле, что соответствует недопустимым директивам на данном шаге диалога. В отличие от графа диалога на естественном языке на квазиестественном не все пары узлов соединены противоположно направленными дугами.

**Анализ** описанных типов диалога показывает, что применение только одного из них (за исключением последнего) не позволяет создать достаточно гибкую и эффективную систему.

**Рассмотрим некоторые ограничения** на структуру диалога, накладываемые разными его типами,

- а) вопросы, требующие ответа ДА/НЕТ — мала эффективность диалога, велика избыточность графа диалога при усложнении темы общения, чрезмерное их использование приводит к переутомлению пользователя;
- б) шаблон — эффективен только в случае ввода фрагмента табличных данных, при ограниченном (от двух до четырех) выборе возможных альтернатив и малом числе шагов диалога;
- в) команда — мала эффективность при значительном числе (не меньше пяти) шагов диалога и альтернативных команд на каждом его шаге.

Использование в диалоговой системе нескольких типов диалога повышает гибкость системы и снижает ее операционную сложность. Работая с диалоговой системой, пользователь должен быть готов к тому, что ему придется встретиться с несколькими типами диалога, в том числе и с такими, которые не были описаны в этом разделе.

### Вопросы для самопроверки

1. Какая составляющая вычислительной системы дешевле: программная или аппаратная?

2. В чем принципиальное отличие диалогового режима работы с ЭВМ от пакетного?

### Вопросы реализации

Необходимо разработать сценарий диалога для некоторой функциональной часть программного комплекса.

Функциональная часть программного комплекса в дальнейшем будет представлена модельно. То есть в виде некоторой программы или набора программ, имитирующих реальное выполнение обработки информации (в простейшем случае - вывод сообщения о работе некоторого алгоритма).

Разрабатываемый сценарий диалога должен строится как "диалог типа меню" и содержать не менее двух уровней (не считая вершины диалога). Модель функциональной части должна содержать не менее семи точек входа.

Реализацию программы диалоговой надстройки рекомендуется осуществить на основе файловой - списочной структуры. При этом - информация о шаге диалога кодируется во внешнем файле, а навигация от шага к шагу выполняется по ссылочной информации, содержащейся в файле. Количество файлов соответствует количеству «узлов» графа диалога типа меню. Например, структура файла - шага диалога:

Имя файла (для справок)
количество строк с информацией шага диалога для пользователя
- Вопрос шага диалога
- Вопрос шага диалога
Первый вариант ответа
Второй вариант ответа
(Строка с ИШДП) - .....
ПРИЗНАК_ДЕЙСТВИЯ ИМЯ_ФАЙЛА_ШАГА_ДИАЛОГА или ИМЯ_"ДЕЙСТВИЯ"
ПРИЗНАК_ДЕЙСТВИЯ ИМЯ_ФАЙЛА_ШАГА_ДИАЛОГА или ИМЯ_"ДЕЙСТВИЯ"
.....
.....

### Пример программы шага диалога и структуры файла описания.

Пример программы, реализующий шаг диалога на основе простейшей структуры файла, описывающего шаг диалога приведена ниже.

- Первые строки файла, длина строки не более 80-ти символов, содержат строки вопроса, строки вариантов ответа и запрос на ввод номера выбранного ответа;
- Затем следует строка, содержащая символ # в первой позиции;
- Затем следуют строки, количество которых соответствует количеству возможных ответов. В каждой такой строке содержится:  
в первой позиции символ 0 или символ 1;  
во второй позиции символ «пробел»;  
затем следует «код возврата номера функциональной части» в случае символа «0» в

начале строки или «имя файла содержащего информацию о следующем шаге диалога» в случае символа «1» в начале строки.

Пусть файл test.txt содержит следующие строки:

1. Чем Вы желаете заняться?
2. 1 - пойти в кино
3. 2 - пойти к другу
4. 3 - пойти на пару
5. Введите номер желаемого ответа
6. #
7. 1 test1.txt
8. 1 test2.txt
9. 0 666

Тогда программа, реализующая шаг диалога может быть записана следующим кодом языка программирования C++:

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
// программа "диалог"
int step_d(char *);
// функции "функциональной части"
void fun_1(void);
void fun_2(void);
void fun_3(void);
void fun_4(void);
void fun_5(void);

int main(void)
{
    int kod;
begin:
    kod=step_d("test.txt");
    printf(" kod= %d \n", kod); // для тестирования
    if (kod == 1) { fun_1(); goto begin;};
    if (kod == 2) { fun_2(); goto begin;};
    if (kod == 3) { fun_3(); goto begin;};
    if (kod == 4) { fun_4(); goto begin;};
    if (kod == 5) { fun_5(); goto begin;};
// и т.д. рационально использовать оператор switch
    getch();
    return 0;
};

int step_d(char * mfile)
{
// строки вопроса
// и строки ответа
// до строки с символом #
// строки "ответа" - 0 или 1 затем пробел
// затем код_возврата или имя_файла_шага_меню

    int num_str, rc=-1, num, k;
```

```

char nfile[80];
char buf[80];
char cnum;
FILE *h;
strcpy(nfile, mfile);
beg:
h=fopen(nfile, "r");
if (h == NULL )
{ printf( "Cannot open input file <%s> \n", nfile); return -1; };

// начало шага диалога.....

    clrscr();
// gotoxy(1,1);
    num_str=1;
// задаем вопрос и варианты ответа
next:
    if ( fgets(buf, 80, h) == NULL )
        { printf( "Ошибка в области вопроса строка N %d в файле <%s>
\n", num_str, nfile);
          return -1;
        };
    num_str++;
    if (buf[0] == '#' ) goto ask;
    k=strlen(buf)-1; if (k>=0) buf[k]='\0';
    printf( "%s\n", buf);
    goto next;
ask:
    scanf("%c",&cnum); num=cnum - '0';
    if ( num <= 0 || num >= 10) goto err;
// поиск "нужной" строки "ответа"
    k=0;
next1:
    if ( fgets(buf, 80, h) == NULL ) goto err;
    k++;
    if ( k != num ) goto next1; // не нашли

// нашли анализ типа строки ответа
    if (buf[0] == '0') // ВСЕ - код возврата....
        { fclose(h); sscanf( buf+2, "%d", &rc); return rc; };
    if (buf[0] == '1') // новый узел диалога....
        { fclose(h); sscanf( buf+2, "%s", nfile); goto beg; };

err:
    gotoxy(1,20);
    printf(" Вы ошиблись (%d)....., повторите\n", num);
    getchar();
    getchar();
    fclose(h); goto beg;
};

```

