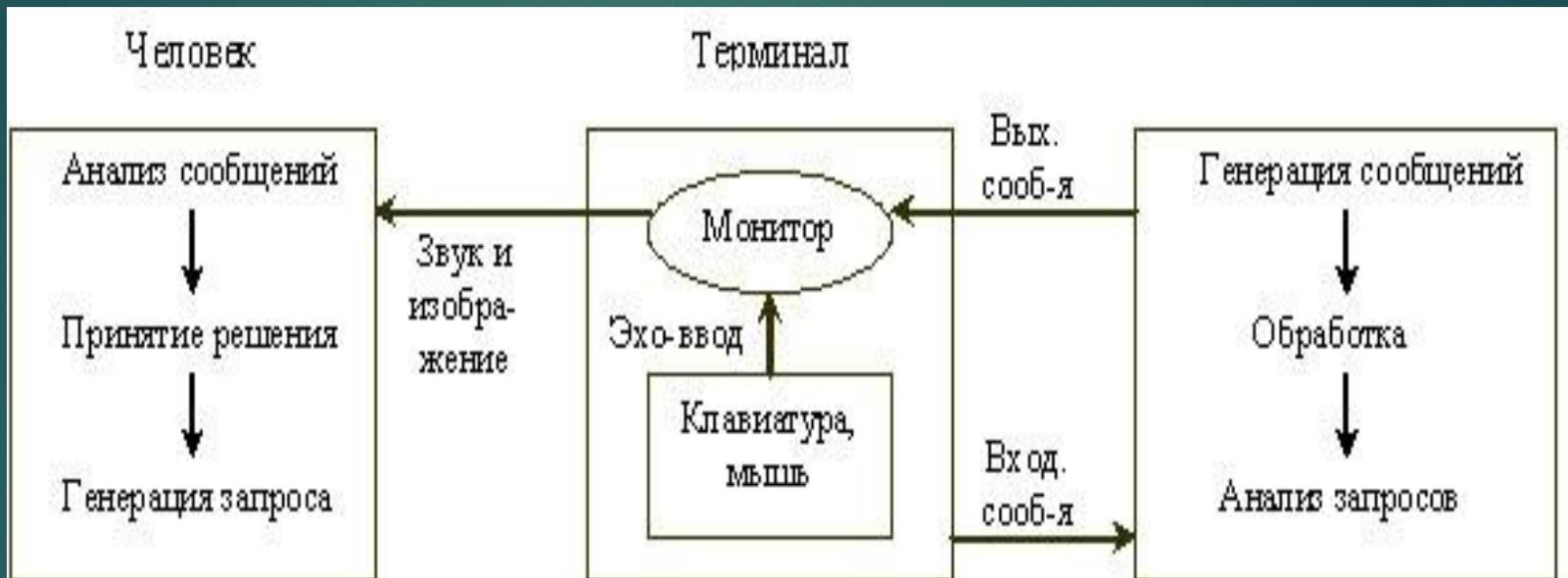


Пользовательский интерфейс

ОРЫНДАҒАНДАР :ӘДІЛХАН Н
ӘБДІҚҰЛОВ Б
ТЕКСЕРГЕН: ЖҰМАГАЛИЕВ Б.И

Пользовательский интерфейс - это совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги. Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи. Каждый диалог состоит из отдельных процессов ввода / вывода, которые физически обеспечивают связь пользователя и компьютера. Обмен информацией осуществляется передачей сообщения.



2. Виды интерфейсов

Интерфейс - это, прежде всего, набор правил. Как любые правила, их можно обобщить, собрать в "кодекс", сгруппировать по общему признаку. Таким образом, мы пришли к понятию "вид интерфейса" как объединение по схожести способов взаимодействия человека и компьютеров. Вкратце можно предложить следующую схематическую классификацию различных интерфейсов общения человека и компьютера.

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 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.

2.1 Командный интерфейс

Пакетная технология. Исторически этот вид технологии появился первым. Она существовала уже на релейных машинах Зюса и Цюзе (Германия, 1937 год). 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 выступить, например, перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки (типа CONSUL). Машина также выдает свои сообщения на перфоратор, алфавитно-цифровое печатающее устройство (АЦПУ), ленту пишущей машинки. Такая машина представляет собой "черный ящик" (точнее "белый шкаф"), в который постоянно подается информация и которая также постоянно "информирует" мир о своем состоянии (см. рисунок 1) Человек здесь имеет малое влияние на работу машины - он может лишь приостановить работу машины, сменить программу и вновь запустить ЭВМ. Впоследствии, когда машины стали мощнее 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 настоящего пользовательской технологии - командной строки.

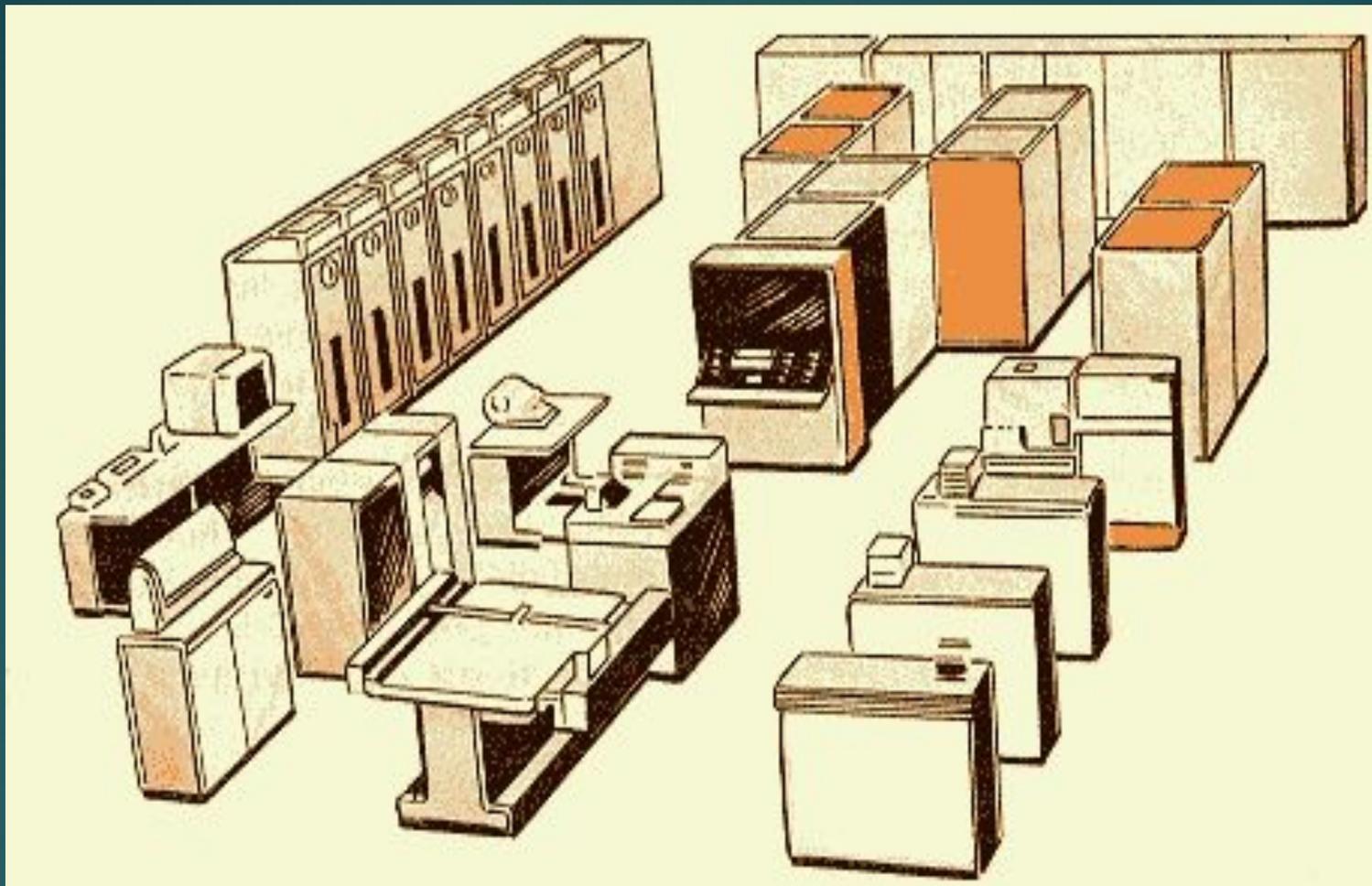


Рис.2. Вид большой ЭВМ серии ЕС ЭВМ

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. 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. 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 многоплатформенной операционной системой CP / M.

2.2 Графический интерфейс

Как и когда появился графический интерфейс? Его идея зародилась в середине 70-х годов, когда в исследовательском центре Xerox 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.

2.2.1 Простой графический интерфейс

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

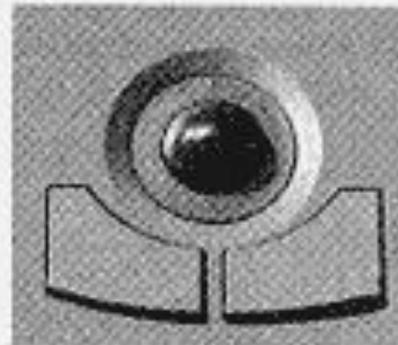
1. При отображении символов допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием. Благодаря этому повысилась выразительность изображения.
2. В зависимости от конкретной реализации графического интерфейса курсор может представляться не только мерцающим прямоугольником, но и некоторой областью, охватывающей несколько символов и даже часть экрана. Эта выделенная область отличается от других, невыделенных частей (обычно цветом).
3. 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.
4. Кроме клавиши Enter, на клавиатуре все чаще стали использоваться "серые" клавиши управления курсором.
5. Уже в этой редакции графического интерфейса стали использоваться манипуляторы (типа мыши, трекбола и т.п. - см. рис.3) Они позволяли быстро выделять нужную часть экрана и перемещать курсор.



Мышь



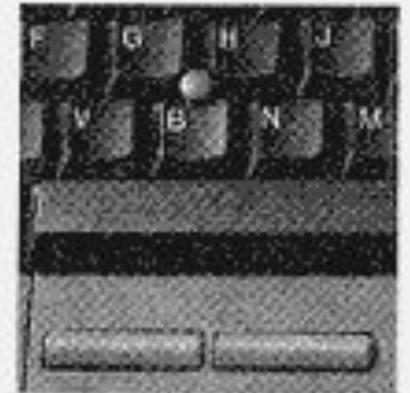
Трекбол (для настольных компьютеров)



Трекбол (в портативном компьютере)



Сенсорная панель



Трекпойнт (между клавишами с буквами G, H и B)

Рис.3. 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 (о файловых оболочках смотри ниже) 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.

2.2.2 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.

2.3 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 неразвитости алгоритма распознавания речи такие системы требуют индивидуальной предварительной настройки на каждого конкретного пользователя.

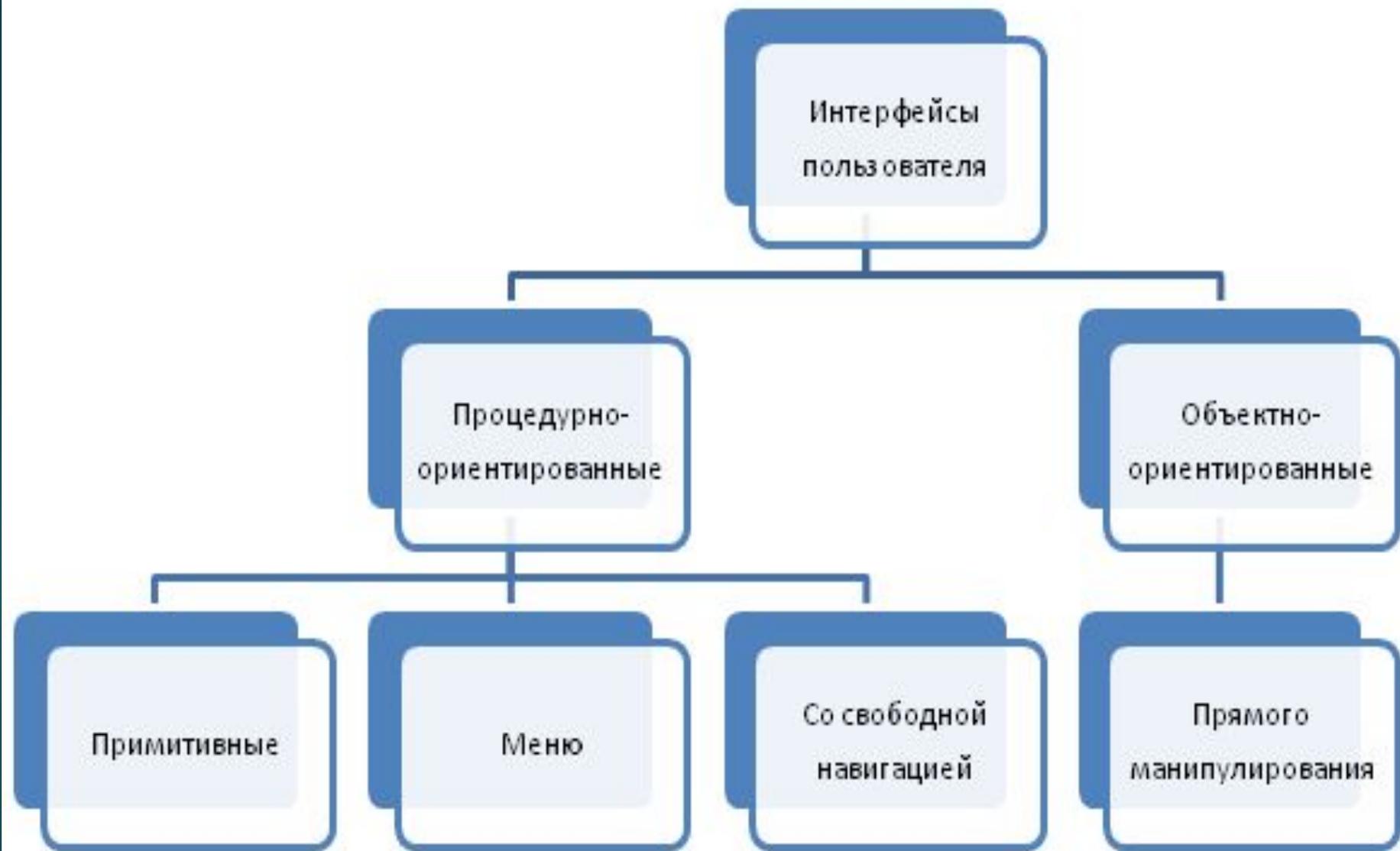
"Речевая" технология является простейшей реализацией SILK - интерфейса

2.4 Биометрическая технология

Эта технология возникла в конце 90-х годов XX века и на момент написания книги 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.

2.5 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 публикации, относящиеся к семантическому интерфейсу, уже не встречались. 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).



Процедурно-ориентированный интерфейс использует традиционную модель взаимодействия с пользователем, основанную на понятиях "процедура" и "операция". В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствие данных и следствием выполнения которых является получение желаемого результата.

Объектно-ориентированные интерфейсы используют модель взаимодействия с пользователем, ориентированную на манипулирование объектами предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов. Задача пользователя формулируется как целенаправленное изменение некоторого объекта. Объект понимается в широком смысле слова - модель БД, системы и т.д. Объектно-ориентированный интерфейс предполагает, что взаимодействие с пользователем осуществляется посредством выбора и перемещения пиктограмм соответствующей объектно-ориентированной области. Различают однодокументные (SDI) и многодокументные (MDI) интерфейсы.



Процедурно-ориентированные интерфейсы:

- 1) Обеспечивают пользователю функции, необходимые для выполнения задач;
- 2) Акцент делается на задачи;
- 3) Пиктограммы представляют приложения, окна или операции;
- 4) Содержание папок и справочников отражается с помощью таблицы-списка.

Объектно-ориентированные интерфейсы:

- 1) Обеспечивает пользователю возможность взаимодействия с объектами;
- 2) Акцент делается на входные данные и результаты;
- 3) Пиктограммы представляют объекты;
- 4) Папки и справочники являются визуальными контейнерами объектов.

3. Методы и средства разработки пользовательского интерфейса

Интерфейс имеет важное значение для любой программной системы и является неотъемлемой ее составляющей, ориентированной, прежде всего, на конечного пользователя. Именно через интерфейс пользователь судит о прикладной программе в целом; более того, часто решение об использовании прикладной программы пользователь принимает по тому, насколько ему удобен и понятен пользовательский интерфейс. Вместе с тем, трудоемкость проектирования и разработки интерфейса достаточно велика. По оценкам [специалистов](#) в среднем она составляет более половины времени реализации проекта. Актуальным является снижение затрат на разработку и сопровождение программных систем или разработка эффективного программного инструментария.

Одним из путей снижения затрат на разработку и сопровождение программных систем является наличие в инструментарии средств четвертого поколения, позволяющих на высоком уровне описать (специфицировать) создаваемое программное средство и далее по спецификации автоматически сгенерировать исполнимый код.

В литературе не существует единой общепринятой классификации средств для разработки пользовательского интерфейса. Так, программное обеспечение для разработки пользовательского интерфейса можно разделить на две основные группы - инструментарий для разработки пользовательского интерфейса (toolkits) и высокоуровневые средства разработки интерфейса (higher-level development tools). Инструментарий для разработки пользовательского интерфейса, как правило, включает в себя библиотеку примитивов компонентов интерфейса (меню, кнопки, полосы прокрутки и др.) и предназначен для использования программистами. Высокоуровневые средства разработки интерфейса могут быть использованы непрограммистами и снабжены языком, который позволяет специфицировать функции ввода-вывода, а также определять, используя технику непосредственного манипулирования, интерфейсные элементы. К таким средствам относятся построители [диалога](#) (interface builders) и СУПИ - [системы управления](#) пользовательским интерфейсом (User Interface Management Systems - UIMS). Помимо СУПИ, некоторые авторы используют такие [термины](#), как User Interface Development Systems (UIDS) - системы разработки пользовательского интерфейса, User Interface Design Environment (UIDE) - среда разработки пользовательского интерфейса и др.

Специализированные средства для разработки интерфейса позволяют упростить разработку пользовательского интерфейса, предлагая разработчику специфицировать компоненты пользовательского интерфейса с использованием языков спецификаций. Можно выделить несколько основных способов спецификации интерфейса:

1. Языковой, когда применяются специальные языки для задания синтаксиса интерфейса (декларативные, объектно-ориентированные, языки событий и др.).
 2. Графическая спецификация связана с определением интерфейса, как правило, средствами визуального программирования, программированием демонстраций и по примерам. Подобный способ поддерживает ограниченный класс интерфейсов.
 3. Спецификация интерфейса, основанная на объектно-ориентированном подходе, связана с принципом, называемым непосредственное манипулирование. Основное его свойство - взаимодействие пользователя с индивидуальными объектами, а не со всей системой как единым целым. Типичными компонентами, используемыми для манипуляций с объектами и управляющими функциями, являются обработчики, меню, зоны диалога, кнопки различного вида.
 4. Спецификация интерфейса по спецификации прикладной задачи. Здесь интерфейс создается автоматически по спецификации семантики прикладной задачи. Однако сложность описания интерфейса затрудняет возможности скорого появления систем, реализующих данный подход.
- Основной концепцией СУПИ является отделение разработки пользовательского интерфейса от остального приложения. В настоящее время идея отдельного проектирования интерфейса и приложения либо закреплена в определении СУПИ либо является основным его свойством.
- В состав СУПИ определен как набор инструментов этапа разработки и периода исполнения. Инструменты этапа разработки оперируют с моделями интерфейса для построения их проектов. Они могут разделяться на две группы: интерактивные инструменты, например редакторы моделей, и автоматические инструменты, например генератор форм. Инструменты периода исполнения используют модель интерфейса для поддержки деятельности пользователя, например, для сбора и анализа используемых данных.
- Функциями СУПИ является содействие и облегчение разработки и сопровождения пользовательского интерфейса, а также управление взаимодействием между пользователем и прикладной программой. Таким образом, в настоящее время существует большое количество инструментальных средств для разработки интерфейса, поддерживающих различные методы его реализации.

4. Стандартизация пользовательского интерфейса

В первом подходе оценку производит конечный пользователь (или тестер), суммируя результаты работы с программой в рамках следующих показателей ISO 9241-10-98 Ergonomic requirements for office work with visual display terminals (VDTs). P.11. Guidance on usability specification and measures:

эффективности (effectiveness) - влияния интерфейса на полноту и точность достижения пользователем целевых результатов;

продуктивности (efficiency) или влияния интерфейса на производительность пользователя;

степени (субъективной) удовлетворенности (satisfaction) конечного пользователя этим интерфейсом.

Эффективность является критерием функциональности интерфейса, а степень удовлетворенности и, косвенно, продуктивность - критерием эргономичности. Вводимые здесь меры соответствуют общей прагматической концепции оценки качества по соотношению "цели / затраты".

о втором подходе пытаются установить, каким (руководящим эргономическим) принципам должен удовлетворять пользовательский интерфейс с точки зрения оптимальности человеко-машинного взаимодействия. Развитие этого аналитического подхода было вызвано **потребностями** проектирования и разработки ПО, поскольку позволяет сформулировать руководящие указания по организации и характеристикам оптимального пользовательского интерфейса. Этот подход может быть использован и при оценке качества разработанного пользовательского интерфейса. В этом случае показатель качества оценивается экспертом по степени реализации руководящих принципов или вытекающих из них более конкретных графических и операционных особенностей оптимального "человеко-ориентированного" пользовательского интерфейса.

Стандартизация и проектирование. При проектировании пользовательского интерфейса исходным решением является выбор базовых стандартов типов управляющих средств интерфейса, который должен учитывать специфику соответствующей предметной области. Конкретизация стиля пользовательского интерфейса осуществляется в нормативных документах отраслевого и фирменного уровня. Возможна дальнейшая детализация дизайна интерфейса для определенной группы программных продуктов фирмы-разработчика. При разработке пользовательского интерфейса необходим учет характеристик предполагаемых конечных пользователей разрабатываемого программного средства. Спецификация типа пользовательского интерфейса определяет только его синтактику. Второе направление стандартизации в области проектирования - формирование конкретной системы руководящих эргономических принципов.

Решение об их выборе должно вырабатываться совместно всеми членами команды по проектированию [9]. Эта система должна быть согласована с соответствующим базовым стандартом (или группой стандартов). Для того чтобы стать эффективным инструментом проектирования система руководящих принципов должна быть доведена до уровня конкретных инструкций для программистов. При разработке инструкций учитываются нормативные документы по типу (стилю) интерфейса, а нормативные документы по проектированию пользовательского интерфейса должны войти в профиль стандартов [13] программного проекта и в техническое задание.

Стандарты и качество. Формально стандартизованность пользовательского интерфейса уместно связать с другими инфраструктурными субхарактеристиками качества программного продукта, такими, как соответствие (conformance) (в том числе и соответствие стандартам) и взаимозаменяемость (replaceability) (ГОСТ Р ИСО МЭК 9126-93).

Выбор конкретного средства проектирования (языки быстрой разработки приложений, CASE-средства, конструкторы графических интерфейсов) может привести разработчика к необходимости придерживаться стандарта интерфейса, положенного в его основу. С другой стороны, выбор разработчиком стандарта типа (стиля) пользовательского интерфейса, адекватного предметной области и используемой ОС, потенциально должен обеспечить, хотя бы отчасти, выполнение таких принципов качества пользовательского интерфейса, как естественность и согласованность в пределах рабочей среды [13]. Явный учет синтактики интерфейса облегчает создание однородного по стилю и предсказуемого для пользователя интерфейса. Кроме того, нужно учесть, что при разработке самого стандарта уже учитывались базовые принципы проектирования пользовательского интерфейса.

Вводимые в ISO 9241-11 меры практичности организация-заказчик может использовать до разработки заказной системы в качестве общих рамок для определения требований по практичности, которым должна соответствовать будущая система и по которым будут проводиться приемочные испытания. Таким образом, создается основа для обеспечения полноты, измеримости и сопоставимости этих требований, что может косвенно оказывать позитивное влияние на качество проектируемого программного изделия.

Спасибо за внимание!!!