

Тема 2.7

Принципы проектирования пользовательского интерфейса

Содержание

1. Виды интерфейсов
2. Классификация интерфейсов пользователя
3. Психологическая основа пользовательского интерфейса
4. Этапы проектирования графического пользовательского интерфейса
5. Основные компоненты графического пользовательского интерфейса и рекомендации по их применению
6. Свойства качественного программного интерфейса

Взаимодействие пользователя (оператора) с вычислительной машиной является важным звеном вычислительного процесса при решении различных прикладных задач как научного, так и производственного плана.

В современных рыночных отношениях создание различного ПО

- информационных сайтов различных организаций и предприятий,
- управления производственными процессами,
- учета выпускаемой продукции и ее реализации,
- автоматизации бухгалтерского учета,
- электронного документооборота
- и даже при такой задаче, как сортировка электронной почты секретарем,

требует разработки удобного для пользователя взаимодействия с ЭВМ.

1. Виды интерфейсов

Основные понятия

В общем случае **интерфейс** (interface) – это

- совокупность логических и физических принципов взаимодействия компонентов технических средств вычислительной системы (ВС), т.е. совокупность правил, алгоритмов и временных соглашений по обмену данными между компонентами ВС (**логический интерфейс**),
- а также совокупность физических, механических и функциональных характеристик средств подключения, реализующих такое взаимодействие (**физический интерфейс**).

Основные понятия

Интерфейсом нередко называют также технические и программные средства, реализующие сопряжение (взаимодействие) между устройствами и узлами ВС.

Интерфейс распространяется на все логические и физические средства взаимодействия вычислительной системы с внешней средой.

Например,

- с операционной системой,
- с оператором
- и т.п.

Виды интерфейсов

Интерфейсы различают по таким характеристикам, как

- структура связей,
- способ подключения и передачи данных,
- принципы управления и синхронизации.

Виды интерфейсов

1. *Внутримашинный интерфейс* – система связи и средств сопряжения узлов и блоков ЭВМ между собой. Внутримашинный интерфейс представляет собой совокупность

- ✓ электрических линий связи (проводов),
- ✓ схем сопряжения с компонентами компьютера,
- ✓ протоколов (алгоритмов) передачи и преобразования сигналов.

Различают два варианта организации внутримашинного интерфейса:

- *многосвязный интерфейс*, при котором каждый блок ПК связан с другими блоками своими локальными проводами;
- *односвязный интерфейс*, в результате которого все блоки ПК связаны друг с другом через общую или системную шину.

Виды интерфейсов

2. *Внешний интерфейс* – система связи системного блока с периферийными устройствами ЭВМ или с другими ЭВМ

Здесь можно выделить также несколько типов внешнего интерфейса:

- *интерфейс периферийных устройств*, подключаемых с помощью шин ввода-вывода
 - ✓ ISA, EISA, VLB, PCI, AGP, USB IEEE 1384 SCSI и др.;
- *сетевой интерфейс*, типа
 - ✓ одноранговой сети
 - ✓ или сети клиент-серверс топологиями типа
 - ✓ звезда,
 - ✓ кольцевая
 - ✓ или шинная.

Виды интерфейсов

3. Интерфейс «человек-машина» или интерфейс «человек-компьютер» или пользовательский интерфейс – это способ, которым оператор выполняет какую-либо задачу с помощью каких-либо средств (какой-либо программы), а именно совершаемые им действия и то, что он получает в ответ.

Интерфейс является ориентированным на человека, если он отвечает нуждам человека и учитывает его слабости.

Пользовательский интерфейс делится на две части:

- машинная часть интерфейса;
- человеческая часть интерфейса.

Виды интерфейсов

Машинная часть интерфейса – часть интерфейса, реализованная в машине (аппаратно-программной ее части) с использованием возможностей вычислительной техники.

Человеческая часть интерфейса – это часть интерфейса, реализуемая человеком с учетом его возможностей, слабостей, привычек, способности к обучению и других факторов.

Наиболее распространенные интерфейсы определены государственными и международными стандартами.

В дальнейшем изложении будет рассматриваться только интерфейс пользователя.

2. Классификация интерфейсов пользователя

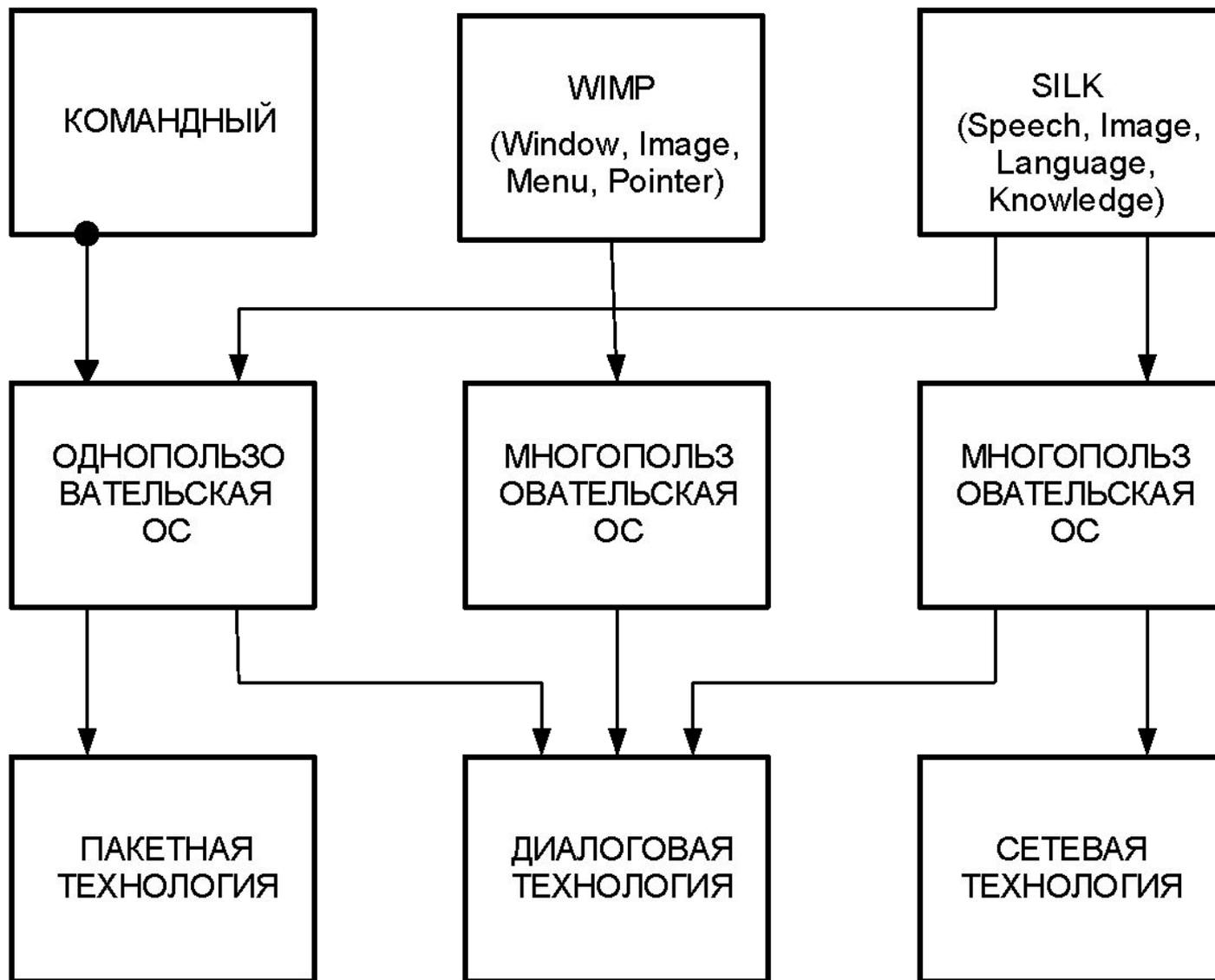
Интерфейсы пользователя

Различают три вида интерфейсов пользователя:

- командный,
- WIMP
- и SILK – интерфейсы.

Взаимодействие перечисленных интерфейсов с операционными системами и технологиями показано на рисунке далее.

Интерфейсы пользователя



Командный интерфейс

1. Командный интерфейс, при котором взаимодействие человека с компьютером осуществляется путем подачи компьютеру команд, которые он выполняет и выдает результат пользователю.

Командный интерфейс может быть реализован в виде

- пакетной технологии
- и технологии командной строки.

В настоящее время

- ✓ пакетная технология практически не используется,
- ✓ а технология командной строки можно встретить в виде резервного способа общения человека с компьютером.

Командный интерфейс

Пакетная технология

Исторически этот вид технологии появился первым на электромеханических вычислительных машинах К.Цюзе, Г.Айкина, а затем на электронных вычислительных машинах Эккерта и Моучли, на отечественных ЭВМ Лебедева, Брусенцова, на ЭВМ IBM-360, на ЕС ЭВМ и так далее.

Идея его проста и состоит в том, что на вход компьютера подается

- ✓ последовательность программ, набитых, например, на перфокартах,
- ✓ и последовательность символов, определяющих порядок выполнения этих программ.

Человек здесь имеет малое влияние на работу машины. Он может лишь приостановить работу машины, сменить программу и снова запустить ЭВМ.

Командный интерфейс

Технология командной строки

Идея.

- В качестве способа ввода информации оператором в ЭВМ служит клавиатура, а компьютер выводит информацию человеку с помощью алфавитно-цифрового дисплея (монитора). Комбинацию монитор-клавиатура стали называть терминалом или **консолью**.
- Команды набираются в командной строке, представляющей собой символ приглашения и мигающий курсор, при этом набранные символы можно стирать и редактировать. По нажатию клавиши «Enter» («Ввод») ЭВМ принимает команду и начинает ее выполнять.
- После перехода в начало следующей строки компьютер выдает на монитор результаты своей работы. Наиболее распространенным командный интерфейс был в операционной системе MS DOS.

WIMP-интерфейс

2. WIMP– интерфейс (Window, Image, Menu, Pointer)
или **ООМУ–интерфейс**(Окно, Образ, Меню, Указатель)

Характерной чертой этого интерфейса является то, что диалог пользователя с компьютером ведется не с помощью командной строки, а с помощью окон, графических образов меню, курсора и других элементов. Хотя в этом интерфейсе подаются команды машине, но это делается через графические образы.

Идея графического интерфейса зародилась в середине 70-х годов в исследовательском центре фирмы Xerox Palo Alto Research Center (PARC).

WIMP-интерфейс

Предпосылкой графического интерфейса явилось

- ✓ уменьшение времени реакции компьютера на команду,
- ✓ увеличение объема оперативной памяти,
- ✓ а также развитие элементной базы, технических характеристик ЭВМ и в частности мониторов.

После появления графических дисплеев с возможностью вывода любых графических изображений различного цвета графический интерфейс стал неотъемлемой частью всех компьютеров.

Постепенно проходил процесс унификации в использовании клавиатуры и мыши прикладными программами.

WIMP-интерфейс

Слияние этих двух тенденций привело к созданию такого пользовательского интерфейса, с помощью которого при минимальных затратах времени и средств на переучивание персонала можно работать с любыми программными приложениями

Этот вид интерфейса реализован в виде двух уровней:

- простой графический интерфейс;
- полный WIMP–интерфейс.

WIMP-интерфейс

Простой графический интерфейс, который на первом этапе очень походил на технологию командной строки со следующими отличиями:

- при отображении символов с целью повышения выразительности изображения допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием;
- курсор мог быть представлен некоторой областью, выделенной цветом и охватывающей несколько символов и даже часть экрана;
- реакция на нажатие любой клавиши во многом стала зависеть от того, в какой части находится курсор.
- кроме часто используемых клавиш управлением курсором стали использоваться манипуляторы типа мыши, трекбола и т.п., которые позволяли быстро выделять нужную область экрана и перемещать курсор;
- широкое использование цветных мониторов.

WIMP-интерфейс

Появление простого графического интерфейса совпадает с широким распространением операционной системы MS DOS.

Типичным примером его использования являются

- ✓ файловая оболочка Norton Commander
- ✓ и текстовые редакторы MaltEdit, ChiWriter, Microsoft Word для DOS, Лексикон и др.

WIMP-интерфейс

Полный WIMP-интерфейс, явился вторым этапом развития графического интерфейса, который характеризуется следующими особенностями:

- вся работа с программами, файлами и документами происходит в окнах;
- программы, файлы, документы, устройства и другие объекты представляются в виде значков (иконок), которые при открытии превращаются в окна;
- все действия с объектами осуществляются с помощью меню, которое становится основным элементом управления;
- манипулятор выступает в качестве главного средства управления.

WIMP-интерфейс

Следует отметить, что WIMP-интерфейс требует для своей реализации повышенного требования

- ✓ к производительности компьютера,
- ✓ объему его оперативной памяти,
- ✓ качественного растрового цветного дисплея,
- ✓ программного обеспечения, ориентированного на этот вид интерфейса.

В настоящее время WIMP-интерфейс стал фактическим стандартом интерфейса ПО, а операционная система Microsoft Windows стала ярким его представителем.

SILK-интерфейс

3. SILK-интерфейс (Speech, Image, Language, Knowledge) или **РОЯЗ-интерфейс** (Речь, Образ, Язык, Знания)

Этот интерфейс 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.
- Результаты выполнения команд он также преобразует в понятную человеку форму.

Этот вид интерфейса требует больших аппаратных затрат, поэтому находится в стадии разработки и совершенствования и используется пока только в военных целях, в системах искусственного интеллекта.

SILK-интерфейс

SILK- интерфейс для общения человека с машиной использует:

- речевую технологию;
- биометрическую технологию (мимический интерфейс);
- семантический (общественный) интерфейс.

SILK-интерфейс

Речевая технология появилась в середине 90-х годов после появления недорогих звуковых карт и широкого распространения технологий распознавания речи.

При этой технологии команды подаются голосом путем произнесения специальных стандартных слов (команд), которые должны выговариваться четко, в одном темпе с обязательными паузами между словами.

Учитывая, что алгоритмы распознавания речи недостаточно развиты, требуется индивидуальная предварительная настройка компьютерной системы на конкретного пользователя.

Это простейшая реализация SILK-интерфейса.

SILK-интерфейс

Биометрическая технология («Мимический интерфейс») возникла в конце 90-х годов и в настоящее время находится в стадии разработки.

Для управления компьютером используется выражение лица, направление взгляда, размер зрачка и другие признаки человека.

Для идентификации пользователя используется

- ✓ рисунок радужной оболочки его глаз,
- ✓ отпечатки пальцев
- ✓ и другая уникальная информация,

которая считывается с цифровой камеры, а затем с помощью программы распознавания образов из этого изображения выделяются команды.

SILK-интерфейс

Семантический интерфейс возник еще в конце 70-х годов XX века, с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса, так как он включает в себя

- ✓ и интерфейс командной строки,
- ✓ и графический,
- ✓ и речевой,
- ✓ и мимический интерфейсы.

Основной его особенностью является отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По сути – это моделирование общения человека с компьютером.

В настоящее время используется для военных целей. Такой интерфейс крайне необходим в обстановке ведения воздушного боя.

Графический интерфейс

В настоящее время наиболее актуально изучение технологии WIMP-интерфейса, так как он наиболее часто используемый для всех типов компьютеров.

Основное внимание будет уделяться программной части интерфейса пользователя, в которой важным звеном является человеческая часть.

Далее в лекциях под графическим интерфейсом пользователя будет подразумеваться именно WIMP-интерфейс.

Т.о. определим общее понятие графического интерфейса пользователя.

Графическим пользовательским интерфейсом (ГИП)

(***Graphical User Interface – GUI***) называется

пользовательский интерфейс, основанный на визуализации объектов, с которыми взаимодействует пользователь в процессе работы.

3. Психологическая основа пользовательского интерфейса

-
- ✓ Наши компьютеры и сотовые телефоны оснащены самыми последними моделями чипов и другой электронной начинкой. Современные операционные системы способны радовать глаз великолепными цветными заставками и стремительными трехмерными эффектами.
 - ✓ Но когда Вы начинаете пользоваться этой системой, выясняется, что в некоторых случаях она ограничивает вас своим непредсказуемым поведением. Из тысячи команд, предусмотренных в системе, вам не удастся найти ту, которая нужна в данный момент, а простые стандартные процедуры выполняются бесконечно долго.

Разрешение этого противоречия лежит в разработке удачного интерфейса пользователя.

В основе разработки хороших интерфейсов лежат принципы, которые на сегодня не являются общеизвестными.

Несмотря на то, что

- ✓ интерфейсы непрерывно совершенствовались в течение двух десятилетий,
- ✓ опубликованы руководства по созданию интерфейсов
- ✓ и созданы средства их разработки

проблема совершенствования интерфейсов пользователя с учетом более глубоких познаний менталитета и психологии пользователя является актуальной.

Тем более что проблема разработки однопользовательских интерфейсов еще не решена, а если индивидуальное взаимодействие с некоторой системой не проходит для пользователя легко и комфортно, то в результате этот недостаток будет негативно отражаться на качестве работы всей системы, независимо от того насколько она хороша в других своих проявлениях.

Интерфейс пользователя связан с такими науками, как

- психология
- и эргономика.

Эргономика – это наука о принципах взаимодействия системы «человек-машина».

Психология – наука о поведении человека в различных ситуациях, его менталитете, интеллектуальных способностях, привычках, реакции на внешние и внутренние раздражители.

Хороший стиль проектирования заключается отнюдь не только в механическом повторении правил.

Если разработчик хочет создать нечто большее, чем сомнительный, кое-как сделанный интерфейс, ему придется получить некоторые теоретические знания, касающиеся человеческой психологии взаимодействия с машиной.

4. Этапы проектирования графического пользовательского интерфейса

1. Анализ целей пользователей и особенностей выполняемых ими задач.

Здесь определяются

- ✓ основные компоненты или объекты, с которыми взаимодействует пользователь,
- ✓ характерные особенности объектов каждого типа
- ✓ и их операции.

2. Способы взаимодействия пользователя с объектами различных типов.

Здесь определяется

- ✓ форма визуального представления объектов,
- ✓ образ которой в зависимости от ситуации может изменяться.

3. Компоновка и пространственное размещение на экране визуальных элементов интерфейса.

Здесь осуществляется

- ✓ выбор цвета, размера и других атрибутов элементов интерфейса,
- ✓ а также выбор средств и методов привлечения внимания пользователя к наиболее важной информации, отображаемой на экране.

Кроме того, здесь продумывается удобный доступ пользователя к средствам помощи.

5. Основные компоненты графического пользовательского интерфейса и рекомендации по их применению

Основные компоненты ГИП

Графические пользовательские интерфейсы (GUI) в настоящее время поддерживаются практически всеми ОС

- ✓ Windows,
- ✓ Linux,
- ✓ Apple Macintosh,
- ✓ OS/2 и т.д.).

В рамках этих ОС для создания интерфейсов разработаны наборы стандартных компонентов взаимодействия с пользователем. Эти наборы не идентичны, как и основные приемы работы с интерфейсами различных ОС.

Основные компоненты ГИП

Как уже отмечалось GUI большинства современных программ строятся по технологии WIMP:

- W - Windows (окна),
- I - Icons (пиктограммы),
- M - Mouse (мышь),
- P - Pop-up (всплывающие или выпадающие меню).

Основными элементами GUI являются:

- окна,
- пиктограммы,
- компоненты ввода-вывода и мышь, которую используют в качестве указывающего устройства и устройства прямого манипулирования объектами на экране.

Основные компоненты ГИП

1. *Окна*

Окно – обычно прямоугольная, ограниченная рамкой область физического экрана. Окно может менять размеры и местоположение в пределах экрана.

Все окна можно разделить на 5 категорий:

- 1) основные окна (окна приложений);
- 2) дочерние или подчиненные окна;
- 3) окна диалога;
- 4) информационные окна;
- 5) окна меню.

Основные компоненты ГИП

Окно приложения обычно содержит:

- ✓ рамку, ограничивающую рабочую область окна,
- ✓ строку заголовка с кнопкой системного меню и кнопками выбора представления окна и выхода,
- ✓ строку меню,
- ✓ пиктографическое меню (панель инструментов),
- ✓ горизонтальные и вертикальные прокрутки
- ✓ и строку состояния.

Дочернее окно используют в многодокументных программных интерфейсах (MDI), предполагающих, что ПО должно работать с несколькими документами одновременно. В отличие от окна приложения дочернее окно не содержит меню.

Основные компоненты ГИП

2. *Пиктограммы*

Пиктограмма представляет собой небольшое окно с графическим изображением, отражающим содержимое буфера, с которым она связана.

Различают:

- 1) программные пиктограммы;
- 2) пиктограммы дочерних окон;
- 3) пиктограммы панели инструментов;
- 4) пиктограммы объектов.

Основные компоненты ГИП

Программными пиктограммами, которые связаны с соответствующей программой, управляет ОС.

- ✓ Так, можно «свернуть» окно приложения в пиктограмму на панели задач Windows
- ✓ или «развернуть» его обратно «на рабочий стол».

Аналогично многодокументная программная система управляет **пиктограммами дочерних окон**, обеспечивающими доступ к различным документа, одновременно обрабатываемым программной системой.

Пиктограммы панели инструментов обычно дублируют доступ к соответствующим функциям через меню, обеспечивая их быстрый вызов.

Основные компоненты ГИП

Пиктограммы объектов используют для прямого манипулирования этими объектами.

Как правило, все пиктограммы можно перемещать мышью.

Для облегчения работы с пиктограммами обычно использую «всплывающие» подсказки, которые появляются, если пользователь в течение некоторого времени удерживает мышь над пиктограммой панели инструментов.

Основные компоненты ГИП

3. Компоненты ввода-вывода

Как уже упоминалось, в окнах приложения могут размещаться специальные компоненты, используемые для ввода-вывода информации.

Интерфейс практически любого современной ПО включает несколько **меню**:

- основное или «ниспадающее» иерархическое меню,
- пиктографические меню (панели инструментов)
- и контекстные меню для разных ситуаций.

Любое из указанных меню представляет собой компонент ввода-вывода, реализующий диалог с пользователем, используя табличную форму.

Основные компоненты ГИП

Кроме меню в интерфейсе используют и другие компоненты ввода-вывода, к которым относят

- текстовые поля,
- списки,
- выпадающие списки,
- флажки
- и т.д.

I. Композиция и организация

При проектировании визуальных элементов ГИП целесообразно опираться на следующие принципы:

1. Иерархическая организация отображаемой информации

Предусматривает упорядочивание информации с точки зрения важности ее предоставления пользователю.

2. Визуальное выделение наиболее важных элементов

Предусматривает, во-первых, выбор на каждом шаге диалога некоторой основной идеи, наиболее важной для выполнения этого шага, и, во-вторых, соответствующее представление и размещение реализующих эту идею элементов.

3. Сбалансированность структуры экрана

Предполагает,

- ✓ с одной стороны, рациональное использование пространства экрана,
- ✓ а с другой – такое размещение информации, при котором на экране в каждый момент времени представлена только та ее часть, которая действительно необходима для выполнения очередного шага задания пользователя.

4. Интеграция

Предполагает использование единых подходов к визуализации отображаемой информации не только в рамках приложения, но и рабочей среды в целом.

5. Визуальное объединение логически взаимосвязанных элементов

Способствует уяснению пользователем того, как именно представленная на экране информация и элементы управления связаны с выполняемым шагом задания и друг с другом.

6. Обеспечение удобочитаемости и логической согласованности отображаемой информации

Предполагает, что

- ✓ любая информация на экране должна быть выражена в компактной и вместе с тем в доступной форме;
- ✓ кроме того, пользователь должен быть способен уяснить, как она связана с предыдущими и последующими шагами задания.

II. Цвет

При выборе цветовой палитры:

1. Целесообразно в рамках одного приложения использовать ограниченное множество цветов, причем предпочтение следует отдавать приглушенным, пастельным тонам.
2. Разрешите пользователю самому настраивать цветовую гамму, где это возможно.

III. Шрифт

При выборе шрифта:

1. Старайтесь не использовать курсив и рубленый шрифт (Serif),
 - ✓ поскольку они трудны для чтения, особенно при низком разрешении монитора.
2. По возможности используйте стандартный системный шрифт для общих элементов интерфейса, т.к. это
 - ✓ обеспечивает визуальную согласованность между интерфейсом вашего приложения и интерфейсом рабочей среды
 - ✓ и, кроме того, делает ваш интерфейс легче масштабируемым.

IV. "Многомерность" экрана

При изображении многих элементов ГИП полезно использовать

- перспективу,
- подсветку
- и затенения

с целью обеспечения эффекта трехмерного образа.

Это способствует

- ✓ повышению функциональности интерфейса
- ✓ и наглядности обратной связи при работе пользователя с компьютерными аналогами реального мира.

При этом:

- используйте трехмерные эффекты только для изображения интерактивных элементов.

V. Пространственное размещение визуальных элементов

- При размещении визуальных элементов старайтесь группировать взаимосвязанные элементы управления ГИП. Для этого используйте
 - ✓ либо специальный элемент – группирующий блок (в Delphi – GroupBox, TPanel),
 - ✓ либо просто размещение элементов на соответствующем расстоянии друг от друга.

- При пространственном размещении элементов управления:
 - ✓ рекомендуется оставлять между элементами в группе промежуток, равный по крайней мере 4 дискретам, а расстояние до края окна и между группами должно быть не менее 7 дискрет;
 - ✓ основные кнопки управления вторичного окна целесообразно группировать в верхнем правом углу окна или расположить в виде линейки вдоль нижнего края окна;
 - ✓ при работе с набором кнопок управления лучше установить кнопку *Отменить* в конце набора, но перед кнопкой *Справка*.
 - ✓ помнить, что область действия кнопок или других элементов управления одной группы, должна распространяться только на эту группу.

- Старайтесь выравнивать взаимосвязанные элементы управления ГИП.

Различают три *способа выравнивания*:

- ✓ вертикальное (по левому или правому краю выравниваемых элементов);
- ✓ горизонтальное (по верхней строке или по верхнему краю элемента);
- ✓ смежное выравнивание (когда элементы смыкаются краями, например, в сгруппированные кнопки в панели инструментов).

VI. Визуализация выполняемых операций

Является одним из способов предоставления пользователю обратной связи с приложением.

Позволяет пользователю не только лучше уяснить сущность выполняемой операции, но и обеспечивает своевременную и корректную его реакцию в случае ошибочных действий.

Целесообразно использовать следующие *виды визуализации*:

- ✓ визуализация операций выбора, предусматривающая изменение изображения выбираемого элемента или области;
- ✓ визуализация операций пересылки, предусматривающая перемещение образа пересылаемого объекта одновременно с перемещением указателя;
- ✓ анимация (например, анимация операции копирования или удаления).

6. Свойства качественного программного интерфейса

Свойства интерфейса

Для того, чтобы программный интерфейс был качественным (эффективным), он должен обладать следующими свойствами:

1. Естественность интерфейса,

- т.е. его способность выдавать сообщения и результаты, которые не требуют дополнительных пояснений.

2. Согласованность интерфейса,

- т.е. его способность предоставлять пользователям возможность
 - ✓ переносить имеющиеся знания на новые задания,
 - ✓ осваивать новые аспекты быстрее,
 - ✓ и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на уяснение различий в использовании тех или иных элементов управления, команд и т.д.

Свойства интерфейса

Согласованность рассматривают в трех аспектах:

- 1) согласованность в пределах приложения
 - ✓ т.е. одна и та же команда должна выполнять одни и те же функции, где бы она не встретилась, причем одним и тем же образом;
- 2) согласованность в пределах рабочей среды
 - ✓ т.е. приложение должно "опираться" на знания и навыки пользователей, которые он получил ранее при работе в среде ОС;
- 3) согласованность в использовании названий
 - ✓ т.е. поведение каждого объекта интерфейса должно соответствовать тому названию, которое ему присвоено.

Свойства интерфейса

3. Дружественность интерфейса,

- т.е. его способность предотвращать ситуации, которые, вероятно, закончатся ошибками, вследствие неправильного ввода команды или данных пользователем.

4. Обратимость интерфейса,

- т.е. его способность каждое действие пользователей сопровождать визуальным или звуковым подтверждением того, что приложение восприняло команду.

5. Простота интерфейса,

- т.е. легкость в его использовании, изучении и в предоставлении доступа ко всему перечню функциональных возможностей, предусмотренных данным приложением.

Свойства интерфейса

6. Гибкость (адаптивность) интерфейса,

- т.е. его способность учитывать уровень подготовки и производительность труда пользователя.

7. Эстетическая привлекательность,

- т.е. его способность приложения обеспечить формирование на экране такой среды, которая
 - ✓ не только содействовала бы пониманию пользователем представленной информации,
 - ✓ но и позволяла бы сосредоточиться на наиболее важных ее аспектах.

Правила эффективного интерфейса

Для создания эффективного программного интерфейса необходимо соблюдать следующие правила:

1. Программный интерфейс необходимо проектировать и разрабатывать как отдельный компонент приложения в соответствии с требованиями стандартизации.
2. Необходимо учитывать возможности и особенности аппаратно-программных средств, на базе которых реализуется интерфейс.
3. Процесс разработки программного интерфейса должен носить итерационный характер, его обязательным элементом должно быть согласование результатов с потенциальным пользователем.
4. Средства и методы реализации интерфейса должны обеспечивать возможность его адаптации к потребностям пользователя.

Правила и принципы практичности

Практичность – это то, чему должен удовлетворять любой хороший пользовательский интерфейс.

Практичный интерфейс – это интерфейс удобный в обращении и простой в понимании.

Для создания практичного программного интерфейса необходимо соблюдать следующие правила:

1. Доступность

Система должна быть настолько практичной, чтобы пользователь, имеющий знания в данной предметной области, но никогда ранее не работавший с системой, без помощи инструкций и руководств мог бы сразу начать ею пользоваться.

2. Эффективность

Система не должна никоим образом препятствовать эффективной работе опытных пользователей, долгое время работающих с ней.

Правила и принципы практичности

3. Продвижение вперед

Система должна способствовать непрерывному повышению уровня знаний, умений и навыков и приспособляться к меняющемуся опыту пользователя.

4. Поддержка

Система должна способствовать более простому, быстрому или увлекательному решению задач, стоящих перед пользователями, предоставлять новые возможности.

5. Контекст

Система должна удовлетворять существующим реалиям и текущей среде эксплуатационного контекста, внутри которого она будет разворачиваться и применяться.

Правила и принципы практичности

Для создания практичного интерфейса необходимо руководствоваться следующими принципами:

1. Структурный принцип

- ✓ Организация пользовательского интерфейса должна быть целесообразной, осмысленной и удобной.
- ✓ Должна базироваться на четких, целостных моделях, очевидных и распознаваемых пользователями.
- ✓ Родственные понятия должны быть связаны, независимые – разделены.
- ✓ Непохожие элементы должны дифференцироваться, а похожие – выглядеть похоже.

Т.е. свойства и функции, которые используются совместно, должны располагаться вместе или должны быть четко и ясно взаимосвязаны.

Элементы не связанные между собой должны быть разнесены в интерфейсе.

Правила и принципы практичности

2. Принцип простоты

- ✓ Максимальное упрощение управления наиболее распространенными операциями.
- ✓ Общение с пользователем должно вестись на понятном для него языке.
- ✓ Должны предоставляться ссылки, логичным образом указывающие на более сложные процедуры.

Т.к. упрощение чего-либо одного ведет к усложнению чего-либо другого, то необходимо знание того, какие задачи пользователь выполняет наиболее часто, и какие из них, с точки зрения пользователя, проще.

Именно такие задачи следует упрощать, что бы пользователь мог их быстро решать.

Правила и принципы практичности

3. Принцип видимости

- ✓ Все функции и данные, необходимые для выполнения данной задачи, должны быть видны, чтобы пользователь не отвлекался на дополнительную избыточную информацию.

Т.е. должны быть видны все нужные и важные функции. Интерфейс не должен заваливать пользователя большим количеством возможных вариантов, смущать его избыточной информацией.

Правила и принципы практичности

4. Принцип обратной связи

- ✓ Сообщайте пользователям о действиях системы, ее реакциях, изменениях состояния или ситуации, о важных ошибках и исключениях.
- ✓ Сообщения должны быть четкими, краткими, однозначными и понятными.

Т.е. должен вестись диалог с пользователем, сообщая ему о том, что происходит в системе.

Правила и принципы практичности

5. Принцип толерантности

- ✓ Интерфейс должен быть гибким и толерантным.
- ✓ Ущерб, наносимый ошибками пользователя, должен снижаться за счет возможности отмены и повтора действий и за счет предотвращения появления этих ошибок путем анализа различных форматов ввода и разумной интерпретации любых разумных действий.

Т.е. чем практичнее система, тем сильнее она помогает пользователям избежать ошибок.

Правила и принципы практичности

6. *Принцип повторного использования*

- ✓ Многократное использование внутренних и внешних компонентов и принципов поведения системы, поддерживая устойчивость осмысленно, а не просто за счет избыточности.
- ✓ Это способствует уменьшению объема информации, которую пользователям приходится запоминать.