


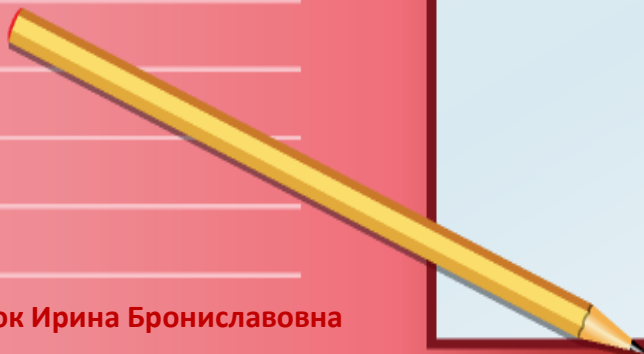


Эпиграф



Нет такой программы или интерфейса, которые бы отвечали всем потребностям пользователя во все времена.


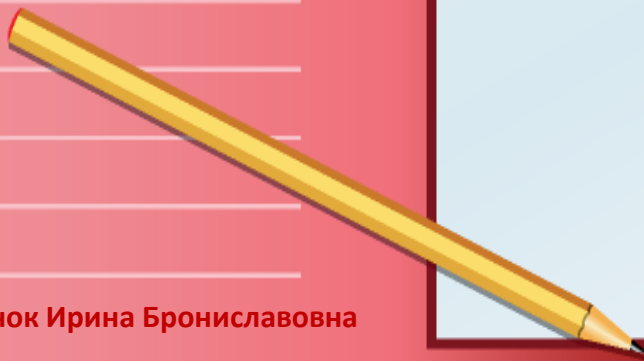
Как проектировщик не зацикливайтесь на одном типе и стиле интерфейса. Держите свой разум открытым для нового. Помните, пользователи вольны менять свое решение в любой момент!






Глава 9

«Графический пользовательский интерфейс»



Тема 24. Основные особенности разработки графического пользовательского интерфейса.

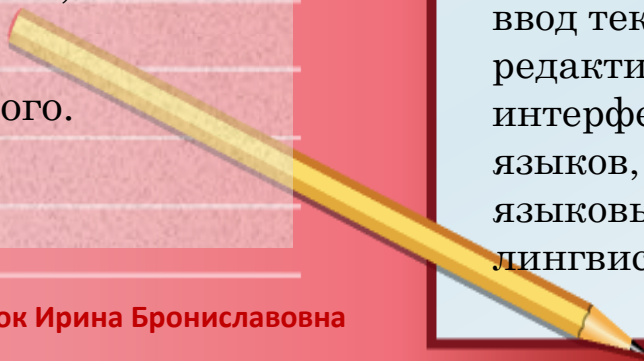
1. Три модели графического пользовательского интерфейса.
 2. Система международных стандартов GUI .
 3. Некоторые принципы проектирования GUI.
 4. Обучающие материалы.
 5. Модные тенденции в GUI-дизайне.
- 




Модели GUI

Первая модель

Концептуальная модель может быть детализирована введением пяти уровней взаимодействия пользователей с информационными системами:

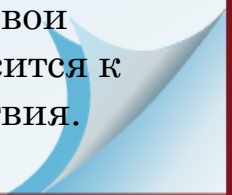
- ☐ физического;
 - ☐ концептуального;
 - ☐ лингвистического;
 - ☐ визуального;
 - ☐ функционального.
- 



Физический уровень определяет состав технических средств и общетехнические требования к ним, например, расположение клавиш на клавиатуре, характеристики устройств автоматического ввода информации, характеристики мониторов. Физический уровень взаимодействия соответствует нижнему уровню интерфейса пользователя в концептуальной модели ИС.

Концептуальный уровень определяет способ отображения состояния среды, с которой взаимодействует пользователь. Он базируется на некотором представлении объектов среды через объекты интерфейса, связываемые между собой в программной среде.

Лингвистический уровень определяет все, что связано с обработкой текстовой информации: сообщения программной среды, ввод текстовых команд пользователя, редактирование текста и т.д. Возможности интерфейса в использовании различных языков, а также то, как он использует свои языковые возможности, — все это относится к лингвистическому уровню взаимодействия.



Концептуальная модель

Первая модель

Концептуальный, лингвистический и визуальный уровни относятся к двум уровням программных средств среды (уровню операционных систем и уровню программных средств общего назначения).

Визуальный уровень конкретизирует отображение концептуальных объектов. Способ отображения концептуальных объектов, их взаиморасположение, возможности пользователя в управлении этим отображением, удобство восприятия и т.п.

Функциональный уровень рассматривает взаимодействие пользователя с системой при решении прикладных задач, то есть типы воздействия пользователя на систему через функции пользовательского интерфейса и способы представления результатов в ответ на эти воздействия.

Функциональный уровень относится к верхнему уровню концептуальной модели, на котором представлены прикладные программы ИС.

Модели GUI

Вторая модель

Модель, реализующая графические пользовательские интерфейсы, представляет собой многоуровневую совокупность компонентов, в которой верхний уровень занимают прикладные программы, а нижние уровни — операционная система и процессор. В дополнении к прикладным программам, операционной системе и процессору, эта модель между ними имеет **пять базовых уровней**.

- ☐ Объектная модель.
- ☐ Прикладные программные интерфейсы (API).
- ☐ Ядро графического пользовательского интерфейса.
- ☐ Оконная система.
- ☐ Модели изображения.

Объектная модель располагается на высшем уровне GUI, но может быть не единственной. Объектная модель отражает реакции и взаимодействие приложений с внешней средой и между собой. Она определяет характер реализуемых прикладных функций, их использование, управление и построение объектов.

Прикладные программные интерфейсы (API) выполняют программно-языковые функции для связи пользовательских приложений с GUI. Программист может специфицировать функции (окна, меню, процессы отображения, иконы), которые необходимы для соответствующих прикладных программ. API включает средства, используемые разработчиками программ при создании GUI для конкретных приложений.

Ядро графического пользовательского интерфейса сосредоточивает экранные функции и элементы.

В таких системах, как OpenLook и Motif — это полное средство обеспечения пользовательского интерфейса. В системе Microsoft Windows в ядре содержится только часть функций GUI.

OpenLook — спецификация графического интерфейса пользователя для рабочих станций UNIX.

Модели, реализующие GUI

Вторая модель

Реализации представленной этой многоуровневой модели графических пользовательских интерфейсов весьма разнообразны и, в той или иной степени, ориентированы на аппаратные и операционные платформы.

GKS (Graphical Kernel System)

— базовая графическая система.

PHIGS (Programmer's Hierarchical Interactive Graphics System) — иерархическая интерактивная графическая система программиста.

Оконная система выделяется в некоторых случаях для повышения гибкости развития всего GUI. Так, например, X Windows первоначально создавалась только как оконная система, а не весь графический пользовательский интерфейс. Однако системы функционально быстро развиваются, и оконные системы сливаются с ядром GUI.

Модели изображения формируются для различных пользовательских интерфейсов с помощью графических средств, в том числе X Windows.

Для унификации интерфейсов могут применяться международные стандарты **GKS (ISO 7942)**, **PHIGS (ISO 9592)** и другие.

Однако общие тенденции состоят в стремлении обеспечить их, по возможности, широкую мобильность между платформами с учетом применения международных стандартов и некоторых стандартов де-факто. При этом важнейшей задачей остается сохранение всей совокупности функций, доступных пользователю, унификация технологии и процедур его взаимодействия с приложениями при любых платформах.

Модели GUI

Третья модель

Данная модель отражает общий методологический подход к реализации интерфейсов пользователя и выбору объектов стандартизации.

Все компоненты пользовательского интерфейса должны поддерживаться операционными системами станций клиентов и серверов.

Эта модель интерфейсов пользователя подразумевает разделение функциональных компонентов приложений на клиентские и серверные части независимо от какой-либо определенной архитектуры среды распределенной обработки данных.

В распределенных информационных системах с архитектурой клиент-сервер пользователи непосредственно взаимодействуют с клиентской частью системы, управляющей запуском и режимами работы прикладных программ. Серверные части прикладных программ обеспечивают доступ к данным и вычислительным ресурсам серверов. Система предоставляет пользователю формы документов и наборы процедур, которые он может выполнять. Эти функции обеспечиваются прикладными программами. Для реализации такого взаимодействия компоненты клиентской части среды, относящиеся к группе функций пользовательского интерфейса, *обеспечивают средства работы с документами (текстами), механизмы управления окнами, готовые примитивы символов (алфавитно-цифровые и графические примитивы)* для формирования нужных объектов, непосредственное манипулирование объектами на экране. Наконец, физический уровень взаимодействия пользователя с системой обеспечивают устройства ввода-вывода информации, входящие в состав автоматизированных рабочих мест.

Таким образом, средства, поддерживающие функции интерфейсов пользователя, размещаются на трех уровнях ИС:

- ☐ уровне прикладных программ,
- ☐ уровне программных средств среды,
- ☐ уровне технических средств среды.

Система международных стандартов

Построение пирамиды руководящих принципов корпоративного стиля продукта



Аббревиатура — любое сокращённое слово или словосочетание.

Акрóним — аббревиатура, образованная из начальных букв, частей слов или словосочетаний, произносимая как единое слово, а не побуквенно. Пример: «ГУМ» как гум, а не гэ-у-эм.

Мнемоника — совокупность специальных приёмов и способов, облегчающих запоминание нужной информации.

Проблемы, которые необходимо решить при проектировании международных интерфейсов.

- ☐ Аббревиатуры
- ☐ Акронимы
- ☐ Выделение мнемоник
- ☐ Грамматика персонализации
- ☐ Двухнаправленные языки (например, японский)
- ☐ Длина текста
- ☐ Заглавные буквы
- ☐ Заголовки колонок
- ☐ Иконки
- ☐ Использование символов
- ☐ Использование цвета
- ☐ Клавиши быстрого доступа
- ☐ Комбинации клавиш
- ☐ Назначение функциональных клавиш
- ☐ Описатели полей ввода
- ☐ Описательный текст
- ☐ Первые буквы для навигации
- ☐ Символы с кодировкой в два байта (DBCS, Unicode)
- ☐ Сортировка информации
- ☐ Формат адреса
- ☐ Формат времени
- ☐ Формат даты
- ☐ Формат денежных единиц
- ☐ Формат размера бумаги
- ☐ Формат размеров
- ☐ Формат телефонных номеров
- ☐ Формат чисел
- ☐ Юмор

Система международных стандартов GUI

Стандарты GUI
регламентируют
*процедуры управления
диалогом* на основе
механизма права доступа и
ряда правил упорядочения
примитивов услуг.

Для согласования параметров образуется ассоциация — соединение между прикладными объектами.

Стандарты предусматривают операции двух типов:

- ☐ синхронные
- ☐ асинхронные.

Синхронные обеспечивают попеременный, взаимоисключающий обмен со стороны различных объектов.

Асинхронные — назначают определенным объектам право доступа на все время их существования.

Семантика диалогов пользователей определяется в стандартах в терминах манипуляции над абстрактными объектами и типов самих объектов, к которым имеют доступ оба пользователя. Они совместно используют концептуальную область взаимодействия.

Для описания операций сервиса GUI используются абстрактные объекты:

- ☐ концептуальная область взаимодействия
- ☐ интерпретатор команд.

Концептуальная область взаимодействия содержит память данных:

- ☐ одного или нескольких дисплейных объектов;
- ☐ управления, сигнализации и статуса;
- ☐ управления доступом;
- ☐ описаний дисплейных объектов, дополнительных устройств, объектов управления и т.п.

Система международных стандартов GUI

Стандарт ISO 9040

Стандарт **ISO 9040** абстрактным образом определяет внешние услуги базового класса виртуального терминала (BT) внутри прикладного уровня ВОО (взаимосвязи открытых систем) путем задания модели взаимодействия между пользователями этих услуг и связанных с ними событий, в зависимости от параметров, содержащихся в каждом примитиве и взаимосвязи последовательностей примитивов.

Услуги обеспечиваются с использованием общего прикладного сервисного элемента, а также услуг уровня представления и могут использоваться пользователем.

Услуги ВТ применяются для реализации интерактивных прикладных процессов, требующих обмена информацией с пользователем и ориентированных на терминал, т.е. для передачи и манипуляции графических образов, состоящих из дискретных графических элементов, организованных в одно-, двух- или трехмерную структуру, которые могут иметь атрибуты, задающие способ их отображения. Стандарт определяет синхронный и асинхронный режим работы, профили и функциональную среду ВТ.

В приложении А определены два рекомендуемых профиля: для асинхронного и синхронного режима работы. В остальных приложениях содержатся некоторые профили ВТ, а также дополнительная информация о стандарте.

Для обеспечения мобильности прикладных программ и данных их взаимодействие с пользователями целесообразно организовывать по принципам и моделям, заложенным в международных стандартах графических систем.


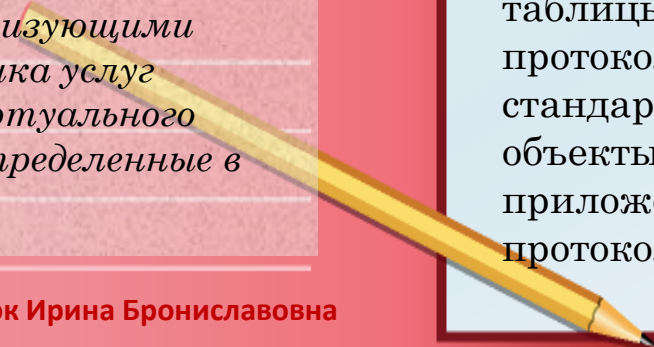


Система международных стандартов GUI

Стандарт ISO 9041

Стандарт ISO 9041

конкретизирует процедуры передачи данных и управляющей информации, ориентированные на соединение, между протокольными автоматами, реализующими функции поставщика услуг базового класса виртуального терминала (BT), определенные в ISO 9040.



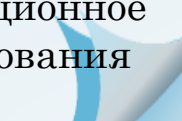
Процедуры определены в терминах: взаимодействий между протокольными автоматами, осуществляемых элементами BT, и взаимодействий протокольного автомата с поставщиком услуг уровня представления путем обмена сервисными примитивами.

Определены две категории систем BT: простые и способные к форматированию. Обе разновидности систем обеспечивают позначное взаимодействие пользователей, однако различаются возможностями удаленной системы обрабатывать данные, вводимые пользователями.

Установлена структура протокольного блока данных, определены три подмножества процедур и способы согласования необходимого подмножества.

Заданы аттестационные требования.

В обязательных приложениях приведены таблицы переходов состояний для протокольного автомата BT и дан список стандартизованных имен, идентифицирующих объекты данного стандарта. Информационное приложение содержит примеры кодирования протокольных блоков данных в ASN1.



Некоторые принципы проектирования GUI

Стандарт ISO 7942

Стандарт ISO 7942

*определяет набор функций для
программирования машинной
графики — Систему графического
ядра (GKS).*

GKS является основой графической системы для приложений, которые создают двумерное машинное изображение на устройствах вывода построчной или растровой графики.

Она поддерживает вывод и интерактивный режим путем обеспечения основных функций вывода и сегментации изображения, позволяет сохранять и динамически модифицировать изображение.

Основой GKS является концепция графической рабочей станции, являющейся абстракцией физического устройства и состоящей из ряда устройств ввода и единственного устройства вывода. Несколько станций могут использоваться одновременно.

Функции управления GKS обеспечивают работу с несколькими логическими рабочими станциями. Для построения изображений в системе используется три системы координат: мировая; нормированная и система координат устройства. Прикладной программист использует мировую систему координат, GKS переводит данные в нормированную систему, а устройство вывода выдает их на отображение в системе координат устройства. Поддержан вывод с изменением атрибутов шести примитивов: ломаная линия; набор маркеров; заполненная область; текст; массив ячеек и обобщенный графический примитив. Стандарт охватывает конструкции и библиотечные вызовы двумерного изображения, в принципе, любого вида. Прикладные программы позволяют адаптировать режимы GKS на рабочую станцию для расширения ее возможностей.


Стандарт включает функции, для хранения и передачи внешнего графического файла. GKS определяет языково-независимое ядро графической системы. Для включения в язык программирования, GKS имеет встроенный языковозависимый уровень.

N 1ST лицензирован комплект аттестационных тестов для GKS и организована сертификация соответствия ему графических систем. Разработано несколько десятков пакетов, реализующих стандарт на различных языках программирования и на различных аппаратных платформах.




Система международных стандартов GUI

Стандарт ISO 8651-1-4
Стандарт ISO 8805



Стандарт **ISO 8651-1-4** из четырех частей формализует внесение в языки программирования Фортран, Паскаль, Ада, Си ядра графической системы (GKS), которое встраивается на языковозависимом уровне, подчиняясь требованиям данного языка.


Стандарт **ISO 8805** определяет набор функций для программирования машинной графики трехмерного ядра графической системы (GKS-3D). GKS-3D является основной графической системой для приложений, создающих трехмерные изображения на устройствах графического вывода.





Система международных стандартов GUI


Стандарт ISO 8806-1-4



Стандарт из четырех частей **ISO 8806-1-4** формализует внесение в языки программирования Фортран, Паскаль, Ада, Си трехмерного ядра графической системы (GKS-3D), которое встраивается на языковом зависимом уровне, подчиняясь требованиям данного языка.

Привязка графического стандарта к языку программирования определяет отображение абстрактных типов данных, используемых в описании стандарта, на реальные типы данных конкретного базового языка.


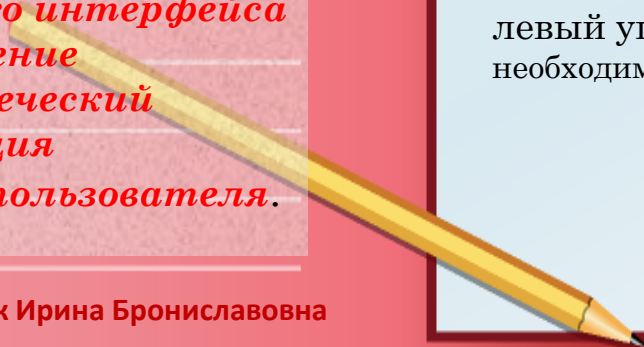
Привязка определяет также представление абстрактных имен функций и списков параметров средствами конкретного языка программирования с учетом всех налагаемых ограничений.





Некоторые принципы проектирования GUI

*При разработке проекта
пользовательского интерфейса
учитывается мнение
художника, человеческий
фактор и интуиция
потенциального пользователя.*



Некоторые основные принципы построения оконных интерфейсов.

1. Необходимо понять, в чем состоит работа пользователя.


Как правило, проектировщики интерфейсов проводят анализ заданий для понимания сути работы пользователя. Имеющийся подход к характеристике прецедентов приблизительно соответствует такому анализу заданий.

2. Нужно сделать так, чтобы клиент чувствовал, что он контролирует взаимодействие.

Интерфейс пользователя должен обеспечивать возможность отмены действий.


3. Желательно предоставить клиенту несколько вариантов завершения каждой операции, связанной с интерфейсом (наподобие закрытия окна или файла), и не обращать особого внимания на его ошибки.

4. Внимание пользователя привлекает верхний левый угол экрана. Поэтому самую важную информацию необходимо размещать именно там.





Некоторые принципы проектирования GUI



5. Нужно учитывать пространственное расположение элементов. Связанные друг с другом компоненты экрана следует размещать рядом, например в одной рамке.


6. Необходимо обращать особое внимание на удобочитаемость и ясность элементов интерфейса (*использовать понятные всем и простые слова!*). Для передачи идей и понятий используйте активный залог.

7. Надо ограничивать количество используемых цветов. Из всего многообразия необходимо остановиться на нескольких.

Чрезмерное множество цветов будет отвлекать пользователя от выполняемых им задач. Кстати, очень неплохо предоставить клиенту возможность самому изменять цвета.


8. Тем, кто подумывает об использовании цвета для выделения содержания, необходимо помнить, что пользователю не всегда легко провести ассоциацию между цветом и содержанием.

Также следует знать, что некоторые пользователи (около 10% взрослых мужчин) плохо различают цвета, поэтому им будет сложно отличить один цвет от другого.





Некоторые принципы проектирования GUI



9. Как и в случае с цветами, необходимо ограничивать число шрифтов.

Желательно избегать курсивов и витиеватых шрифтов. Шрифт под названием "Haettenschweiler" довольно приятен, но не стоит его использовать постоянно.

10. Старайтесь создавать компоненты (типа кнопок и списков) одинакового размера.

При использовании компонентов различных размеров, разнообразии цветов и шрифтов создается мешанина или даже мозаика, которую специалисты в области GUI называют дизайном в стиле "клоунских штанов".

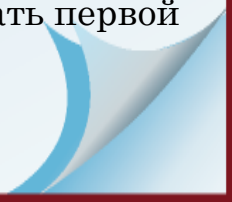
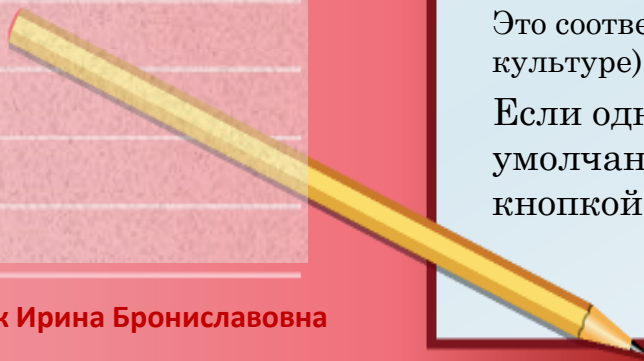
11. Выравнивайте компоненты и поля данных по левому краю.

Это уменьшает нагрузку на глаза при просмотре экрана.

12. Если пользователь после прочтения и обработки определенного блока информации должен щелкать на кнопках, то такие кнопки лучше разместить справа от блока информации или же под этим блоком и, опять таки, справа.

Это соответствует естественной тенденции (присущей нашей культуре) читать слева направо.

Если одна из кнопок является кнопкой по умолчанию, то ее нужно выделить и сделать первой кнопкой в наборе.



Обучающие материалы

Как интегрировать справочную систему или документацию их с интерфейсом?

Что нам нужно и что у нас есть

Количество подсистем справки, нужных для того, чтобы пользователь научился пользоваться системой, довольно невелико, так что все их можно легко разобрать.

Базовая справка объясняет пользователю сущность и назначение системы. Обычно должна сработать только один раз, объясняя пользователю, зачем система нужна. Как правило, не требуется для ПО, зато почти всегда требуется для сайтов.

Обзорная справка рекламирует пользователю функции системы. Также обычно срабатывает один раз. Нужна и ПО и сайтам, и нужна тем более, чем более функциональна система. Поскольку у зрелых систем функциональность обычно очень велика, невозможно добиться того, чтобы пользователи запоминали её за один раз. В этом случае оптимальным вариантом является слежение за действиями пользователя и показ коротких реклам типа «А вы знаете, что...» в случае заранее определенных действий пользователей (примером такого подхода являются помощники в последних версиях MS Office).

Обучающие материалы

Справка предметной области отвечает на вопрос «Как сделать хорошо?». Поскольку от пользователей зачастую нельзя рассчитывать знания предметной области, необходимо снабжать их этим знанием на ходу.

При этом действуют два правила:

- ❑ во-первых, пользователи ненавидят признавать, что они чего-либо не знают, соответственно, подавать это знание надо максимально «небрежным тоном»;
- ❑ во-вторых, наличие такого знания всегда повышает субъективную оценку справочной системы в целом, т.е. приводит к тому, что пользователи чаще обращаются к справочной системе и от этого эффективней учатся.

Процедурная справка отвечает на вопрос «Как это сделать?». В идеале она должна быть максимально более доступна, поскольку если пользователь не найдет нужную информацию быстро, он перестанет искать и так и не научится пользоваться функцией (возможно, никогда).

Обучающие материалы


Система должна индентифицировать все свои состояния.

Контекстная справка отвечает на вопросы «Что это делает?» и «Зачем это нужно?». Как правило, наибольший интерес в ПО представляет первый вопрос, поскольку уже по названию элемента должно быть понятно его назначение (в противном случае его лучше вообще выкинуть), а в Интернете – второй (из-за невозможности предугадать, что именно будет на следующей странице).


Поскольку пользователи обращаются к контекстной справке во время выполнения какого-либо действия, она ни в коем случае не должна прерывать это действие (чтобы не ломать контекст действий), её облик должен быть максимально сдержанным, а объем информации в ней – минимальным.

Справка состояния отвечает на вопрос «Что происходит в настоящий момент?». Поскольку она требуется именно что в настоящий момент, она не может быть вынесена из интерфейса. В целом это самая неproblemатичная для разработчиков система справки.

Сообщения об ошибках. Об этом несколько позже.



Среды передачи информации





Так как наша цель состоит только в определении интеграции справочных систем с интерфейсом, разумно будет свести получившийся список типов обучающих материалов, которые нам нужны, со средами передачи этих материалов, которые у нас имеются.

Среды передачи, имеющиеся в нашем распоряжении:


Бумажная книга. На одном листе может быть сконденсировано очень много материала, легко позволяет читателю получить большой объем материала за один сеанс, наилучшим образом работает при последовательном чтении. Сравнительно плохой поиск нужных сведений. Объем практически всегда лимитирован.

Справочная карта. Отдельная краткая бумажная документация, демонстрирующая основные способы взаимодействия с системой (quick reference card). Будучи реализована на едином листе бумаги, позволяет пользователю повесить её перед собой. Хороша как средство обучения продвинутым способам взаимодействия с системой и устройству навигации в системе.





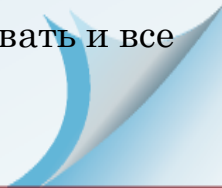
Среды передачи информации



Структурированная электронная документация. Плохо предназначена для чтения больших объемов материала, зато обеспечивает легкий поиск и не имеет лимита объема. Занимает большой объем пространства экрана. Плохо подходит для показа крупных изображений, зато в неё могут быть легко интегрированы видео и звук.

Фрагменты пространства интерфейса, показывающие справочную информацию. Занимают пространство экрана, но пространство ограниченное. Отвлекают внимание, как минимум один раз воспринимаются всеми пользователями. Как правило, неспособны передавать большой объем информации.

Всплывающие подсказки. Хорошо справляются с ответом на вопросы «Что это такое» и «Зачем это нужно», при условии, что объем ответов сравнительно невелик. Поскольку вызываются пользователями вручную, в обычном режиме не занимают пространства экрана и не отвлекают внимания пользователей. С другой стороны, очень легко вызывают отвыкание – после первого же случая неудовлетворения пользователя подсказкой, пользователь перестает вызывать и все остальные подсказки.



Виды справочных систем

А теперь те виды справочных систем, за которые ответственен дизайнер интерфейсов, разберем более детально.

Базовая и обзорная справки

Эти системы справки обычно реализуются в бумажной документации. Это хорошо, но, вообще говоря, можно сделать и лучше, поскольку в последнее время появилась возможность интегрировать в справочную систему видео при помощи либо Macromedia Flash, либо Shockwave.

Нет сомнений, что реклама, поданная не просто в виде текста с картинками, но в виде анимации, способна как повысить желание её просмотреть, так и повысить субъективное удовлетворение пользователей от системы.

На этом уровне заниматься этим должен не дизайнер интерфейсов, а отдельный графический дизайнер.

Как правило, работа эта не очень сложна (поскольку, фактически, всем содержимым этой анимации является показ сменяющих друг друга скриншотов). Сложно создать более-менее хороший сценарий (его лучше отдать профессиональному писателю).

Дизайнер интерфейса в этом случае должен только составить список функций, которые нужно рекламировать.



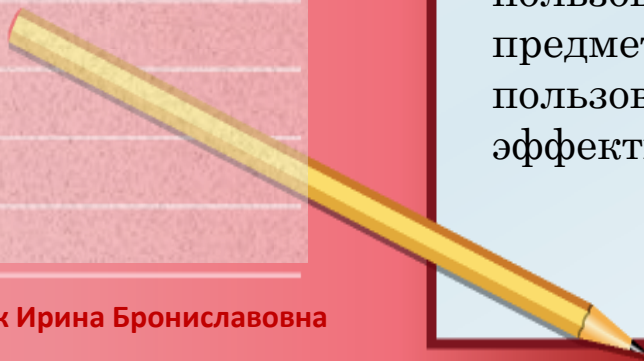
Виды справочных систем



Справка предметной области

Справка предметной области также реализуется обычно в бумажной документации, найти аргументы против такого положения вещей достаточно затруднительно. Однако, как минимум, часть её можно подавать пользователям в интерфейсе вместе с выдержками из обзорной справки (лучше динамически, а la «Помощник» из MS Office).

По-моему, справка предметной области является самой важной подсистемой справки. Достаточно упертый или «компьютерно-грамотный» пользователь сможет воспользоваться системой, лишенной всех справочных систем, более того, такой пользователь сможет даже *научиться* пользоваться такой системой. Но без знания предметной области он никогда не сможет пользоваться системой правильно и эффективно.





Виды справочных систем

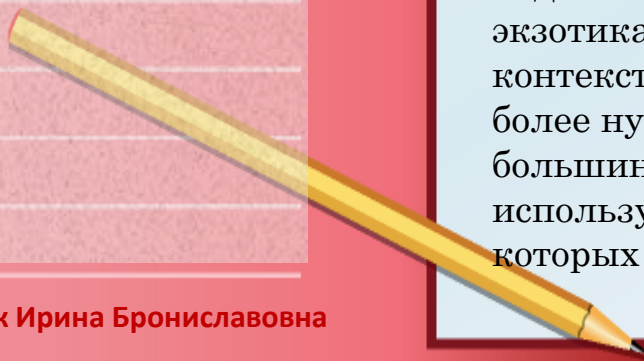


Процедурная справка

Лучшим местом для процедурной справки является выделенная справочная система. В неё, собственно говоря, она чаще всего и помещается. Вызывает, однако, сожаление тот факт, что разработчики чаще всего не привязывают темы справки к интерфейсу: когда пользователям непонятно, как выполнить нужное им действие, им приходится искать в справочной системе нужную тему. Это неправильно, тем более что технических проблем в этом нет.

Контекстная справка

Для контекстной справки заслуженно используют всплывающие подсказки (ToolTip) и, в последнее время, пузыри. Это очень правильное решение, с которым невозможно поспорить. Огорчает только практически полное отсутствие этого типа справки в интернете. Если разработчики ПО уже привыкли писать ко всем объектам и элементам управления подсказки, то для веб-дизайнеров это пока экзотика. Интересно при этом, что в интернете контекстная справочная система, как правило, более нужна, нежели в ПО – просто потому, что большинство сайтов являются однократно используемыми системами, пользователями которых являются изначально необученные люди.






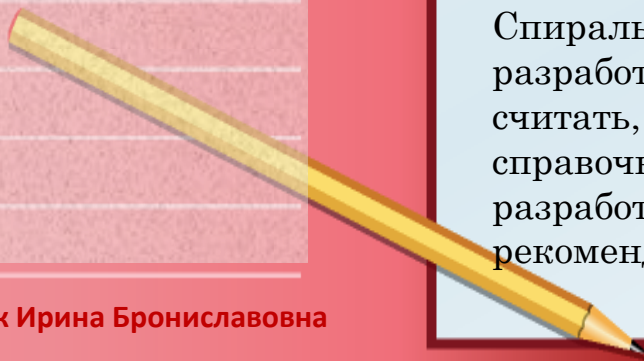
Спиральность

В отличие от художественной литературы, справочные системы не предназначены для того, чтобы приносить удовольствие.

Таким образом, следует всемерно сокращать объем справочной системы, чтобы тем самым сократить длительность неудовольствия.

Однако при малом объеме справочной системы возрастает риск того, что пользователи не найдут в ней ответы на свои вопросы.

Куда ни кинь – всюду клин.



Есть, однако, исключительно эффективный метод решения этой проблемы: так называемые *спиральные тексты*.

Идея заключается в следующем. При возникновении вопроса пользователь получает только чрезвычайно сжатый, но ограниченный ответ (1-3 предложения).

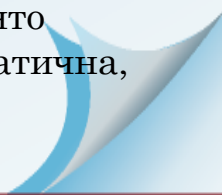
Если ответ достаточен, пользователь волен вернуться к выполнению текущей задачи, тем самым длительность доступа к справочной системе (и неудовольствие) оказывается минимальной.

Если ответ не удовлетворяет пользователя, пользователь может запросить более полный, но и более объемный ответ.

Если и этот ответ недостаточен (что случается, разумеется, весьма редко), пользователь может обратиться к ещё более подробному ответу.

Таким образом, при использовании этого метода, пользователи получают именно тот объем справочной системы, который им нужен.

Спиральность текста считается нормой при разработке документаций. Есть веские основания считать, что она необходима вообще в любой справочной системе. Учитывая тот факт, что разработка спирали в справке не проблематична, рекомендуется делать её во всех случаях.



Модные тенденции в GUI-дизайне

В середине девяностых с развитием настольных издательских систем и в связи с проникновением графического интерфейса в массы (вместе с Windows) широкое распространение начали получать *нестандартные графические интерфейсы*.

Одним из апологетов данного направления выступал ныне почти забытый гуру GUI-дизайна **Кай Краузе**, работавший в мелкой компании HSC, которая потом слилась с Fractal Design (изготовитель первых версий Painter) была переименована в Metatools, затем в Metacreations, а потом и вовсе растасканная на части Corel, Procreate и другими компаниями.



Первый опыт в проектировании *нестандартных интерфейсов* был по-настоящему дерзко реализован в плагинах для фотошопа, наиболее известный из которых — Kai's Power Tools...

<http://img518.imageshack.us/i/kpt6mattr6.jpg/>

Модные тенденции в GUI-дизайне

Затем последовали не менее легендарные **Kai's PowerGoo**, **Soap**, **Convolver** и по-настоящему монстроидальные **Bryce**, **Poser** и **Carrara**.

Успех ярких, необычных, но профессионально выполненных решений Краузе породил целую плеяду хороших и не очень попыток выйти за рамки того скучного набора интерфейсных элементов, которые предлагали Windows, MacOS и Unix.

В основном необычным интерфейсом отличались плагины для Photoshop, AfterEffects и Painter. Хотя ни одна из известных программ не смогла приблизиться к тому уровню, который задал в своих приложениях Кай Краузе.

Его неожиданные формы *непрямоугольных окон*, *металлические кнопки* и *полупрозрачные панели* дюжину лет назад вызывали дикий восторг у любого непосвященного. Многие ставили у себя KPT или Bryce лишь только для эстетического удовольствия подвигать какую-нибудь полупрозрачную линзу с ползунками поверх картинки. Однако, со временем продукция MetaCreations становилась всё более эргономичной, все эти графические изыски сглаживались и упрощались, хотя элементы ОС также не использовались.

В итоге, когда компания перестала существовать, а программы были раскуплены, сам Кай Краузе куда-то запропал, **к началу нового тысячелетия угас интерес к нестандартным GUI.**

Модные тенденции в GUI-дизайне

Интересные решения можно было увидеть лишь в двух секторах программного обеспечения:

- ☐ музыкальные редакторы,
- ☐ программы трехмерного моделирования.

Особенно бурно развивалась индустрия VST-плагинов для музыкального софта.

Здесь и по сей день можно встретить занятные экземпляры графического интерфейса.

Кому-то подобные решения покажутся излишними, а кто-то привык и уже жить без них не может.



Как правило, **элементы управления везде схожие** (кнопки, ползунки, чекбоксы, списки и т.п.), но вот **их оформление и взаимосвязь зачастую различается**.

Модные тенденции в GUI-дизайне

*Что это — новая тенденция в
сфере GUI-дизайна?*

*И следует ли нам ожидать
появление новой масштабной волны
нестандартных интерфейсов? Особенно
в тёмной гамме...*

Хорошо это или плохо?

*Насколько удобно или неудобно
работать с такими приложениями?*

*Сказывается ли это на
устойчивости, производительности
труда и креативности?*

*Или цвет и форма элементов
интерфейса никоим образом не влияет
на рабочий процесс, и GUI программы
может быть каким угодно?*

В последнее время всё чаще стали появляться программы с нестандартным, отличающимся от GUI операционных систем интерфейсом.

Особенно много стало «гламурных» тёмных интерфейсов. Причём такой софт выпускают отнюдь не маленькие компании, штампующие фотошоповские плагины, а вполне серьезные гиганты софт-индустрии.

Adobe представила «тёмные» Lightroom и Bridge.

Apple отметилась Aperture и Logic.

Даже Microsoft не удержалась и представила последнюю версию Windows Media Player в модном нынче «готичном» оттенке.



Задача разработать интерфейс для телевизора

На телевизоре

- 33. Первый канал
Клипы нашей молодости
- 34. Первый игровой
Судьба новой приставки
- 35. A-one
Дело в шляпе
- 36. **Ozon**
Альтернативная музыка
- 37. Ru-tv
Клипы
- 38. MTV
Свидание наугад
- 39. Enaza TV
Игровые новости
- 40. NG HD
Природа нашей страны

На компьютере

IP-TV Player v0.28

39. Здоровое ТВ
46% "Детское здоровье"

40. Драйв
55% Внутри Гран При

41. Усадьба ТВ
86% Миссия: перестановка

42. AXN Sci-Fi
81% Звездные врата - Дети Богов, часть 2

43. Fashion TV
55% ФОКУС НА

44. Discovery Science
66% Требуется сборка. Серия 1

45. Euronews
55% Новости, спорт, экономика, обзор, I

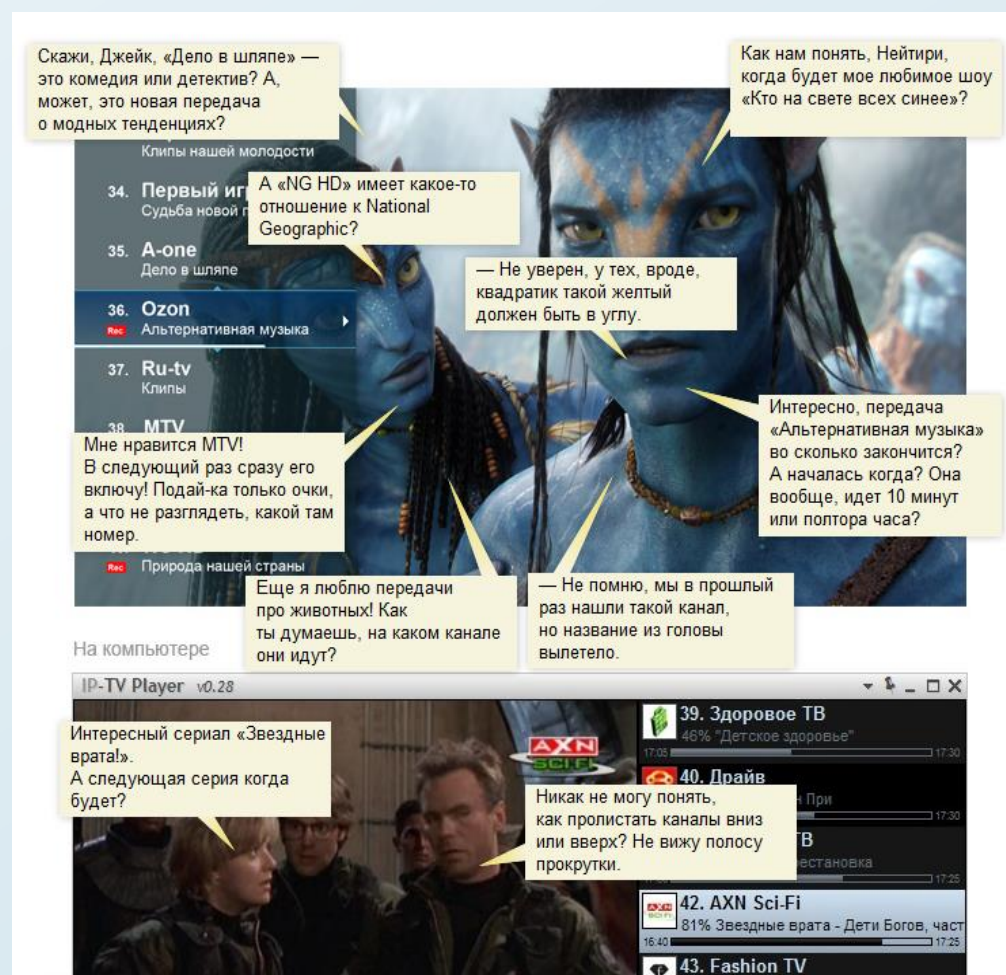
№42 - AXN Sci-Fi. 17:17:46 (84%)
Звездные врата - Дети Богов, часть 2 I /...
16:40 17:25

Канал AXN Sci-Fi

(((Live)))

Задача разработать интерфейс для телевизора

Вот такие вопросы
возникают у тех, кто
решил посмотреть
телевизор, снабженный
подобным интерфейсом.





Задача разработать интерфейс для телевизора

Недостатки:

Зрители с тоской вспоминают старые добрые газеты «с программой», где ответы на эти вопросы, как правило, вполне очевидные.

Здесь же зрители вынуждены переключаться между каналами вслепую.


Понять что за передача идет, можно только ее посмотрев, а узнать, что будет по ее окончании, вообще нет никакой возможности.

Достоинства:

От дополнительного уровня информации, который дает логотип телекомпании, отказываться не стоит.

Логотип не только подсказывает тематику канала, но и значительно повышает узнаваемость, ведь его показывают в углу экрана постоянно, его образ гораздо более запоминаемый, чем название.

Прогресс-бар тоже помогает зрителю принять решение, стоит ли переключаться на передачу, которая идет в данный момент. Только, кроме относительного значения, важно и абсолютное. 25% выпуска новостей совсем не то же самое, что 25% «Аватара».



Визуальных элементы управления Studio

Чтобы узнать больше, посетите **DevExpress** в Интернете по адресу:

<http://community.devexpress.com/blogs/>

и

<http://devexpress.com/Products/NET/Controls/WPF>

Чтобы загрузить бесплатную пробную версию **DXperience** и полный спектр визуальных элементов управления Studio, посетить их в Интернете по адресу:

<http://www.devexpress.com/Downloads/NET/>




DevExpress DXperience + eXpressApp Framework v8.3.2 (Source + Installer) — это набор популярных библиотек для .NET от компании Developer Express, Dxperience.

Представленный набор компонентов для Visual Studio представляет собой уже признанный инструментарий создания отчетности для .NET и расширения среды разработки приложений для формирования объектных структур. Включает в себя UI компоненты, такие как XtraGrid, XtraCharts, eXpressPersistentObjects (XPO), компоненты для Web, WPF, Silverlight, платформу для быстрого создания "бизнес"-приложений eXpressApp Framework.

Полностью интегрированные в среды Visual Studio 2005 и 2008, 2010 технологии Developer Express помогают разработчикам больше времени уделять повышению квалификации за счет ускорения процесса разработки и использования командной стратегии для увеличения продуктивности с помощью комплексного набора средств.

DevExpress WinForms products are available individually, or as part of the DXperience WinForms (without source), DXperience Enterprise (with source) and DXperience Universal (with source) subscriptions.

See our Windows Forms products in action:

 XtraGrid™ XtraVerticalGrid™ XtraReports™ XtraNavBar™ XtraBars™ XtraEditors™ XtraScheduler™ XtraPrinting™ XtraRichEdit™ XPO™ XtraCharts™ XtraSpellChecker™ XtraPivotGrid™ XtraGauges™ XtraLayout Control™ XtraWizard™ XtraTreeList™


















Online
Documentation



Training Videos

Multi Editors



Category	Product1	Product2	Product3
 Product Name	Chai	Teatime Chocolate Biscuits	Ipoh Coffee
 Quantity Per Unit	10 boxes x 20 bags	10 boxes x 12 pieces	16 - 500 g tins
 Unit Price	18,00p.	9,20p.	46,00p.
 Units in Stock	39	25	17
 Discontinued	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 Category	 Beverages	 Confections	 Beverages
 Supplier	Exotic Liquids	Specialty Biscuits, Ltd.	Leka Trading
 Contact Name	Shelley Burke	Robb Merchant	Sven Petersen
 Phone	(100)555-4822	(111)555-1222	(120)555-1154
 Last Order	14.12.2001	20.07.2000	07.01.2002
 Relevance			



In this demo editors are dynamically assigned to individual cells. For the cells in the last node that use progress bar in-place editors you can press the '+' or '-' keys to increase or decrease cell values, respectively.

Add New Row (Grouped Mode)
 Alpha Blending
 Banded View Navigation
 Calc Row Height
 Card View
 Column Alignment
 Column Customization
 Column Format
 Control Navigator
 Custom Draw
 Custom Filter Dialog
 Custom Menu
 Custom Tooltips
 Double Click Row
 Drag&Drop External
 Drag&Drop Internal
 Draw Column Objects
 Edit Data In Popup Form
 Editor Show Mode
Editors (External Repository)
 Editors (Internal Repository)
 Error Notification
 Error Notification Advanced
 Export
 Field Auto Height
 Filter Popup Date (updated)
 Filtered View
 Focused Row
 Group Level Style
 Hit Info
 IBindingList
 Incremental Search
 Indicator
 Look And Feel
 Master-Detail
 Multi Editors
 Multi Select
 New Item Row
 Options

Editors (External Repository)

Stop

Trademark Model Category

Accessibility

Внешний вид

Поведение

Accessibility

! This example demonstrates repository is used.

Drag a column header here to group by that column

Main	Performance Attributes		
Trademark	HP	Cyl	
Category	Model	Capacity	# of Gear
Price	Delivery Date	In Stock?	Automatic Transmission
Mercedes-Benz		302	
> Sports	SL500 Roadster	4,966 Liters	
\$83 800,00	24.07.2002	<input checked="" type="checkbox"/>	<input checked="" type="radio"/> Yes <input type="radio"/> No
Mercedes-Benz		342	
> Sports	CLK55 AMG Cabriolet	5,439 Liters	
\$79 645,00	17.12.2001	<input type="checkbox"/>	<input checked="" type="radio"/> Yes <input type="radio"/> No
Mercedes-Benz		189	

XtraGrid Control (gridControl1)

Description:

The InitEditors() method creates the repository items that hold the settings and event handlers for editors of particular types and adds them to the external editor repository (the PersistentRepository component). The repository items stored in it can be used by multiple controls. The grid control is bound to the PersistentRepository via the ExternalRepository property. After the external repository has been assigned to the grid, the repository items are bound to the grid's columns.

Related Member List:

ExternalRepository, PersistentRepository, ColumnEdit

Code

```
private void InitEditors() {
    RepositoryItemPictureEdit itemPictureEdit = new RepositoryItemPict
    RepositoryItemRadioGroup itemRadioGroup = new RepositoryItemRadioG
    RepositoryItemSpinEdit itemSpinEdit = new RepositoryItemSpinEdit()
    RepositoryItemCalcEdit itemCalcEdit = new RepositoryItemCalcEdit()
```

Copy

Close