




Эпиграф




*Средний человек
испытывает
большие страдания
от муки новой
идеи.
Адмирал Уильям С. Симс*



Глава 5

«Навигация и
другие аспекты
человекоориен-
тированных
интерфейсов»



Тема 14. Интуитивные и естественные интерфейсы

1. Интуитивные интерфейсы.
2. Навигация.
3. Масштабируемые
интерфейсы.
4. Применение ZoomWorld.

Интуитивные интерфейсы

Одним из самых хвалебных терминов, используемых в отношении интерфейсов, является слово «**интуитивный**».

При ближайшем рассмотрении это понятие исчезает так же бесследно, как шарик в наперстках, и заменяется более простым термином «**знакомый**».

Многие требования, предъявляемые к интерфейсам, предполагают, что конечный продукт должен быть *интуитивным*, или *естественным*.

Когда пользователи говорят, что какой-то интерфейс является «интуитивным», они имеют в виду, что он работает так же, как и какой-то другой метод или программа, с которыми они знакомы.

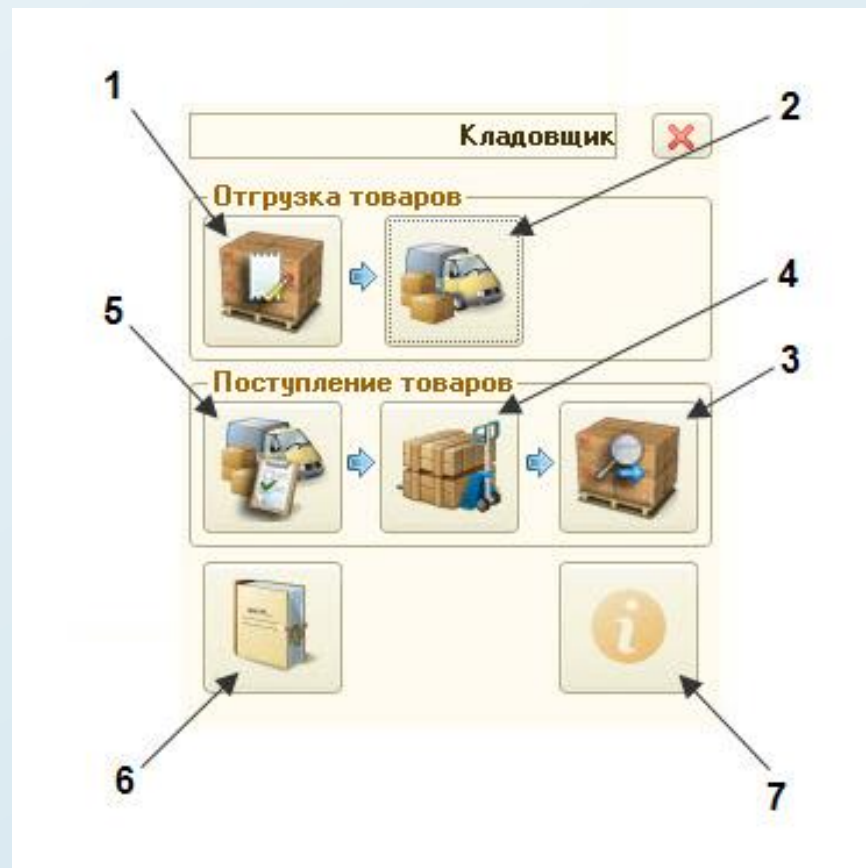
Интерфейс является «естественным», если он работает так, что пользователю совсем не требуется объяснять, как им пользоваться.

Существующие сегодня системы навигации, не являются удачными и абсолютно не подходят для целей обработки терабайт информации, которые мы вынуждены просматривать. Однако люди и животные имеют тысячелетнюю практику перемещения внутри сложных сред и за это время сумели приобрести некоторые полезные навыки.

Эти способности, сложившиеся в течение многих эпох, могут быть привлечены для наших целей — для создания того, что можно назвать *масштабируемым интерфейсом*

Интуитивные интерфейсы (Пример)

1. Сбор готовых заказов в специальную зону.
2. Контроль отгрузки заказов.
3. Оприходывание товара на склад в зону приемки.
4. Размещение товаров на складе с указанием конкретного адреса размещения.
5. Контроль качества продукции.
6. Инвентаризация.
7. Просмотр информации о внутреннем штрих-коде.



Интуитивные интерфейсы



Часто говорится, что использование мыши и есть пример интуитивности и естественности.

Пример: фильм из научно-фантастического сериала «Star Trek» (использование мыши в качестве микрофона).

Пример: Мышь с шариком и колесиком, плюс кнопки по обеим сторонам.



Вывод, мышь не является чем-то очевидным для каждого, кто пытается ею воспользоваться.

Научиться пользоваться мышью очень легко, но это не значит, что использование мыши является интуитивным или естественным.

Что это?

Дизайнеры
Ichimura Shigenori и
Miyazawa Tetsu

Trypoxy Phone



feelings like "pinching", "squeezing" and so on. You can operate it without an effort. You can control "Trypoxy" freely, using your five fingers, just like experienced musical players play their instruments.



Навигация

Как известно из древних легенд и мифов, человек всегда очень плохо ориентировался в лабиринтах.



(Турена, Франция, площадь 4 гектара)
Если бы мы могли легко в них разбираться, они не использовались бы для игр и головоломок.

Пользователи плохо запоминаем длинные последовательности поворотов, — именно поэтому *лабиринты являются хорошими головоломками*, и наши современные навигационные схемы, используемые как в компьютерах, так и в Интернете, так часто приводят пользователей в замешательство.

Многие жалобы на современные системы связаны с их навигацией.

Разработаны частичные решения, — такие как *Избранное (Favorites)* в браузерах.

Пользователи могут хорошо запоминать положение ориентиров и позиционные подсказки, — эти черты развились в людях с эволюцией, и они могут быть использованы в разработке интерфейсов.

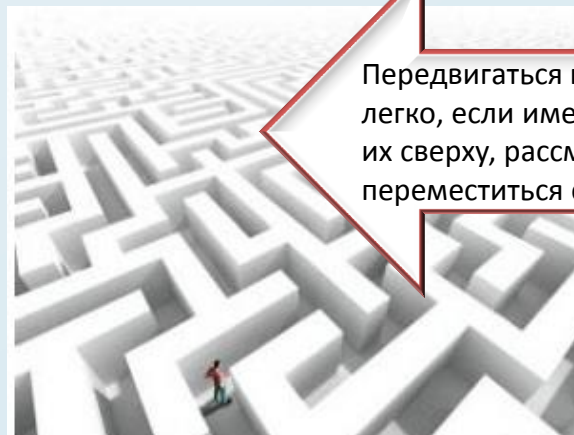
Навигация

Противоположностью лабиринтов является ситуация, в которой вы можете *видеть цель и путь к ней*, который позволяет сохранить чувство ориентации во время перемещения и при необходимости вернуться назад.



Изысканным решением является *принцип масштабируемого интерфейса ПМИ (zooming interface paradigm, ZIP)*.

Данный принцип во многих случаях дает возможность решить *проблемы навигации*, а также проблему *ограниченного экранного пространства*, которая возникает при использовании любой существующей системы отображения.



Передвигаться по лабиринтам можно легко, если имеется возможность увидеть их сверху, рассмотреть их план и переместиться сразу туда, куда нужно.

Принцип масштабируемого интерфейса как раз и дает такую возможность при решении многих задач, выполняемых на компьютере.

Хотя **ПМИ** не может подойти для всех случаев, остановимся на его положительных аспектах для того, чтобы показать, что существуют альтернативы, которые более эффективны по сравнению с пользовательским графическим интерфейсам, основанным на применении рабочего стола.

Zooming User Interface или Zoomable User Interface

Масштабируемый интерфейс пользователя — графический интерфейс пользователя, где рабочее пространство представляет собой большую или неограниченную плоскость, на которой расположены основные элементы, свойства и содержимое которых становятся доступны по мере их «приближения» путём увеличения. Дальнейшее приближение содержимого делает доступным более глубокие уровни.

Улучшенная навигация: ZoomWorld

Пример
масштабируемого
интерфейса, который мы
рассмотрим, называется
ZoomWorld (дословно —
масштабируемый мир).

Пример:

[http://zoomquilt2.madmin
dworx.com/zoomquilt2.swf](http://zoomquilt2.madmin
dworx.com/zoomquilt2.swf)



Его идея заключается в том, что пользователь имеет доступ к безграничной плоскости информации с неограниченной степенью разрешения.

Эта плоскость является масштабируемой средой ZoomWorld. Все, к чему вы можете обратиться, находится где-нибудь на плоскости этой среды — будь это ваш компьютер, локальная сеть, к которой ваш компьютер подключен, или сеть сетей (как, например, Интернет).

Вся метафора этой среды заключается в том, что вы можете находиться на определенной «высоте» над ней, поднимаясь «вверх» для увеличения масштаба и спускаясь «вниз» — для уменьшения.

Навигация по системе осуществляется как с помощью изменения высоты, так и с помощью поиска.

Улучшенная навигация: ZoomWorld

Масштабируемое пространство обладает большой гибкостью с точки зрения компоновки содержания.

Система ZoomWorld может служить в качестве поискового интерфейса для баз данных в предприятиях и объединениях, государственных организациях, школах, каталогах научных данных и т.д.

Трехмерные объекты представлены здесь в виде двумерных проекций или других преобразований на дисплее, однако могут быть созданы и **трехмерные интерфейсы ZoomWorld.**

Этот метод может заменить (или улучшить) иерархические каталоги. Важным дополнением к этому может быть достаточно быстрый текстовый поиск (например, функция LEAP), который может использоваться в тех случаях, когда требуется быстро найти какой-то текстовый элемент.

Можно также использовать небольшое количество различных геометрических ориентиров (пиктограмм).

Например, с помощью большого красного креста можно обозначать область в которой можно получить экстренную помощь.



Применение ZoomWorld

Типичное применение
ZoomWorld в проекте,
разработанном для
компании **Apricus**.

*Палаты изображены в виде
нумерованных
прямоугольников с именами
находящихся в них пациентов.*

Необходимо было компьютеризировать большой (площадью приблизительно один квадратный метр) медицинскую карту, использующуюся в отделениях интенсивной терапии (intensive care unit, ICU).

Все перепробованные до этого методы работали медленнее в сравнении с использованием физической карты и требовали больших усилий на изучение. Также нельзя было применять множество мониторов для отображения всей карты целиком.

ZoomWorld смог не только вместить карту — этот интерфейс позволил автоматизировать всю базу данных отделения интенсивной терапии и стал использоваться в качестве базы данных для всего предприятия, и все это без существенных дополнительных затрат на разработку.

Использование нового интерфейса позволило расширить модель бизнеса компании и область ее применения без разработки более сложного интерфейса или внедрения каких-либо дополнительных средств, хотя, конечно, потребовало базы данных большего объема.

Применение ZoomWorld

Пример применения ZoomWorld может быть портал Всемирной сети. Из этого портала мы можем перемещаться в любую область Всемирной сети.

Интерфейс типа ZoomWorld сам по себе является превосходным браузером.

В масштабируемой среде, понятия «открытия» или «закрытия» документов или приложений уже перестают быть необходимыми.

Принцип масштабируемого интерфейса позволяет также смягчить другую редко упоминаемую проблему разработки интерфейсов.

В имеющихся интерфейсах мы достигли какой-то стильной лаконичности в попытках подобрать одно идеальное слово для обозначения элемента меню и вынуждены использовать непонятные шифровки, чтобы поместить на экране как можно больше надписей.

При использовании масштабируемого интерфейса экранное пространство становится более доступным, и с ним мы можем ***позволить большую ясность без экономии пикселей.***

Конечно, даже без учета принципа масштабируемого интерфейса мы всегда должны стремиться больше к ясности, чем к краткости в формулировках надписей.

Вывод

Идея
масштабируемой
среды принадлежит
Имэньюэлу Нойку

Вывод, который можно сделать :
*масштабируемый интерфейс может
заменить браузер, метафору рабочего
стола и традиционную операционную
систему.*

При использовании такого интерфейса
приложения как таковые исчезают.

В сочетании с методами, масштабируемая
среда **ZoomWorld** позволяет в целом
упростить использование компьютерных
систем.

*Масштабируемая среда, построенная с
учетом принципов когнетики, может
полностью соответствовать требованиям,
налагаемым когнитивными
возможностями человека, что упрощает
процесс разработки, изучения и
использования в сравнении с современными
методами создания программного
обеспечения.*

Вывод

Существующие инструменты для разработки интерфейсов не позволяют создавать масштабируемые среды.

Отдельно о
VISUAL STUDIO 2010 PREMIUM

Для этих целей программирование должно проводиться на таких языках, как Perl, C, C++ или Smalltalk.

Модель, имитирующая такую среду, может быть создана с помощью Macromedia Flash.

Однако наши старые привычки в разработке являются одной из трудностей с точки зрения сохранения качества интерфейсов такого рода.

Единственным способом удержать интерфейс на верном пути является тщательное изучение и внимание к основам его разработки с учетом когнитивности.

Что касается аппаратного оборудования, то для работы в масштабируемой среде мышь может иметь три кнопки: две кнопки, обозначенные сверху как **Выделить** и **Активизировать**, и третью кнопку, обозначенную сбоку как **Масштабировать**, с помощью поворота которой (по часовой стрелке или против нее, аналогично направлению резьбы в стандартном шурупе) можно было бы увеличить или уменьшить масштаб.



Глава 5

«Навигация и другие аспекты человекоориен- тированных интерфейсов»



Тема 15. Пиктограммы

1. Пиктограммы.
2. Концепции использования пиктограмм.
3. Разработка пиктограмм пользовательского интерфейса.

Пиктограммы

Компания **Apple Computer**, известный лидер в области разработки интерфейсов, сообщает нам, что

«пиктограммы могут существенным образом увеличить ясность и усилить привлекательность приложения».

Пиктограммы (icons), эти знакомые всем маленькие картинки, служащие для обозначения кнопок и других объектов, являются неотъемлемым признаком современных интерфейсов.



Всякий раз, когда требуется добавить объяснение или надпись, попытайтесь вместо текста использовать пиктограмму.

Использование пиктограмм позволяет намного упростить процесс перевода программ на другие языки.

В более поздних версиях подход к применению пиктограмм не был столь догматичным, однако созданный их использованием вред уже нельзя исправить.

Пиктограммы

Пиктограммы делают интерфейс более привлекательным в визуальном отношении и, при определенных условиях, могут способствовать большей понятности.

Однако со временем стали понятны и **недостатки пиктограмм**.

Например, как в операционной системе Macintosh, так и в Windows сейчас уже используются средства для объяснения значения пиктограмм.

Если вы наводите курсор на какую-то пиктограмму, появляется небольшое окно с текстом, в котором дается ее описание.

Возникает очевидный вопрос: «Почему вместо пиктограмм сразу не использовать текст?»

В самом деле почему бы и нет? Ведь, по сути дела, *вместо того чтобы объяснять, пиктограммы зачастую сами требуют для себя объяснений.*

Использование пиктограмм вместо слов вполне подходит для того, чтобы скрыть или зашифровать какую-то информацию от посторонних глаз.

Проблему пиктограмм можно рассматривать как проблему ограниченной видимости.

Примеры пиктограмм

В Высшей Академической школе графического дизайна при МХУПИ регулярно проходят смотры работ студентов.

В студенты-второкурсники в качестве курсовой работы представляли системы визуальных коммуникаций для разных объектов.

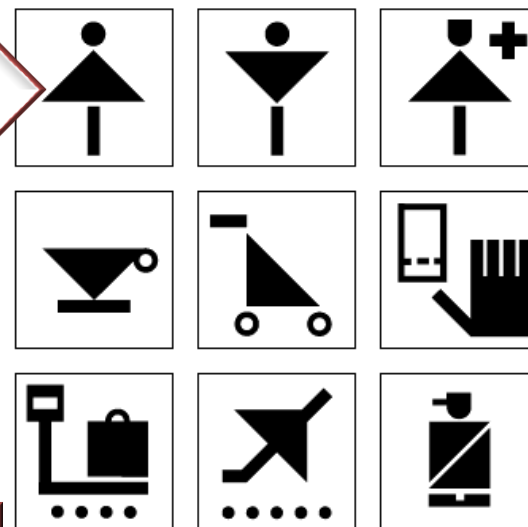
Каждый из студентов разрабатывал набор для того или иного учреждения или заведения. Это были аэропорт, школа, турбаза, универсам, зоопарк, больница, автосервис и т.д.

В качестве обязательных изображений, требовались пиктограммы мужского и женского туалета, а также гардероба или кассы (в зависимости от специфики объекта).

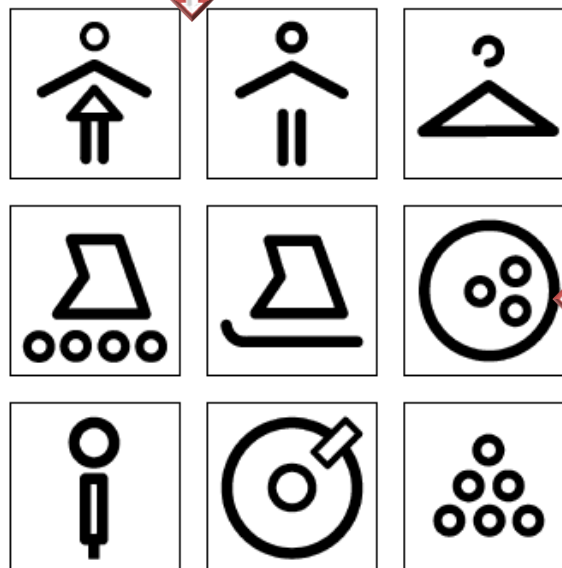
Содержание остальных знаков было произвольным.

"Медсестра в штатском",
"Мерилин Монро",
"Медсестра в униформе".
Шутки шутками, а почему туалет для медсестер отдельный — непонятно.

Аэропорт



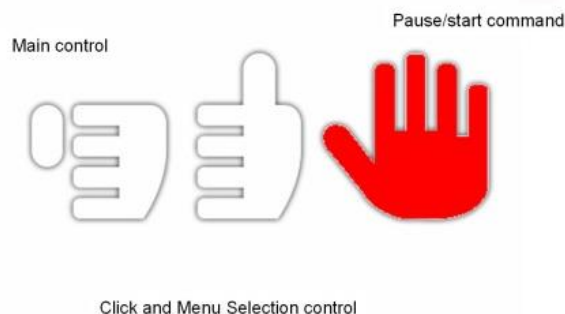
Развлекательный центр



"Танк",
"Танк задавил
лыжника",
"Яичница-
глазунья"

Пиктограммы

В интерфейсе показана пиктограмма, но ее смысл невидим, или же ее изображение может дать неверное сообщение для тех, для кого это изображение незнакомо или кем это изображение может быть истолковано по-другому.



Например, пиктограмма, изображающая ладонь поднятой руки, в Соединенных Штатах означает «стоп», а в Греции — «вот вам экскременты на ваше лицо».

Пиктограммы могут вызвать раздражение не только у специалистов по разработке интерфейсов.

По поводу автомобилей :


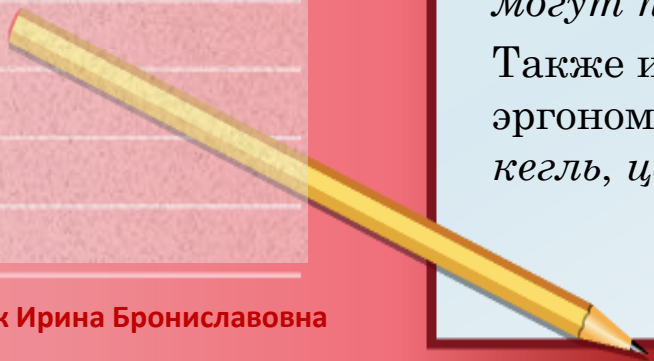
«Чтобы понять назначение ручек настройки [автомагнитолы], нужно доставать инструкцию по использованию, так как никаких надписей (вроде надписи «громкость») на кнопках не имеется» .

Иногда использование пиктограмм обосновывается их независимостью от языка, что является ценным свойством для программного обеспечения с точки зрения расширяющегося международного рынка. Однако следует учитывать, что, как было указано, *смысл пиктограмм зависит от культуры*, в которой они используются.



Пиктограммы

Две концепции в
использовании
пиктограмм.




В предисловии к «Книге о пиктограммах» (The Icon Book) ее автор **Уильям Хортон** (William Horton) говорит: *«Около десяти лет я пользуюсь системами с графическими пользовательскими интерфейсами и предпочел бы щелкать мышью по понятным картинкам, чем вводить техническую тарабарщину в виде команд, — даже если бы я мог запомнить, как правильно эти команды пишутся».*

Однако **Хортон** не говорит о другом варианте, а именно: щелкать мышью можно и по кнопке, обозначенной одним или двумя хорошо подобранными словами.

Мы хорошо умеем визуально отличать одно слово от другого, и, в то же время, слова могут передавать сложный смысл.

Также имеют значение такие эргономические факторы, как *регистр, кегль, цвет* и другие атрибуты шрифта.



Пиктограммы

Мэйхью (Mayhew)

упоминает ряд исследований, посвященных использованию пиктограмм.

К сожалению, в большинстве исследований не проводилось сравнений между пиктограммами и надписями.

Однако из этих и других исследований мы можем сделать вывод, что пиктограммы являются наиболее эффективными, если **количество пиктограмм, которые видно одновременно, не превышает 12**.

Кроме того, необходимо, чтобы пиктограммы:

- ☐ **визуально отличались друг от друга;**
- ☐ **хорошо отражали соответствующее понятие;**
- ☐ **имели разумно большой размер (обычно больший, чем могла бы быть текстовая надпись).**

Разработка объектов пользовательского интерфейса

1. Сайт разработчика:

<http://www.aha-soft.com>

2. Сайт разработчика:

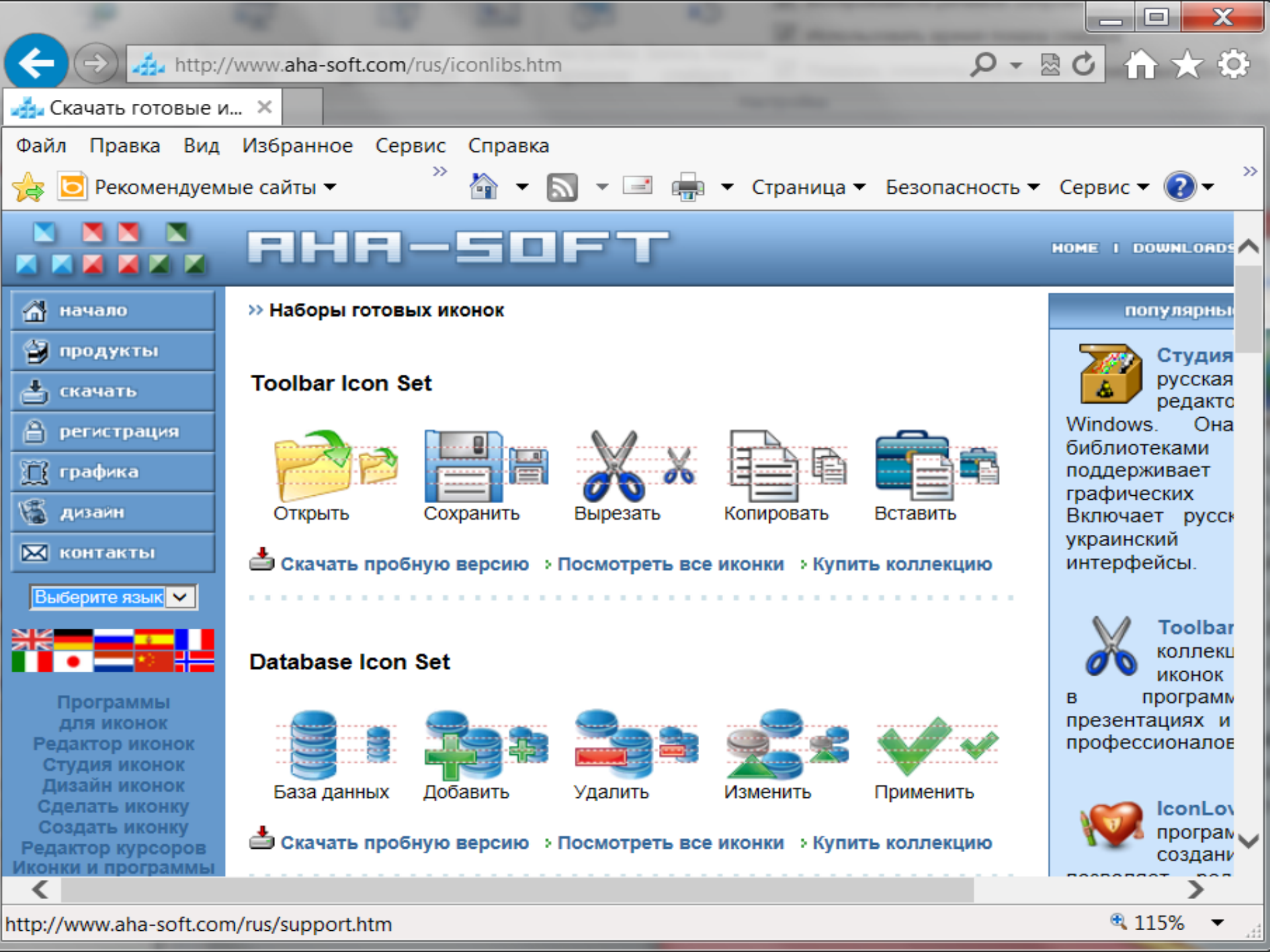
<http://www.iconarchive.com>



Программа для работы с пиктограммами: поиск, создание, редактирование, экспорт/импорт, создание библиотек иконок и т.п.

Поддерживается импорт иконок из файлов форматов ico, ani, cur, wmf, emf, bmp, wbmp, jpg, gif и png, редактирование файлов bmp, jpeg, png и gif (размером до 127x127), извлечение иконок из exe-файлов и библиотек, а также создание собственных коллекций иконок.

Помимо формата, принятого в Windows, поддерживаются иконки mac-формата.



AHA-SOFT

HOME | DOWNLOADS

- начало
- продукты
- скачать
- регистрация
- графика
- дизайн
- контакты

Выберите язык



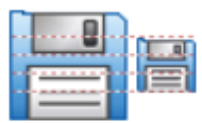
Программы для иконок
Редактор иконок
Студия иконок
Дизайн иконок
Сделать иконку
Создать иконку
Редактор курсоров
Иконки и программы

Наборы готовых иконок

Toolbar Icon Set



Открыть



Сохранить



Вырезать



Копировать



Вставить

Скачать пробную версию | Посмотреть все иконки | Купить коллекцию

Database Icon Set



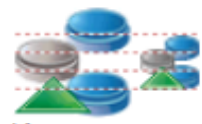
База данных



Добавить



Удалить



Изменить



Применить

Скачать пробную версию | Посмотреть все иконки | Купить коллекцию

популярные



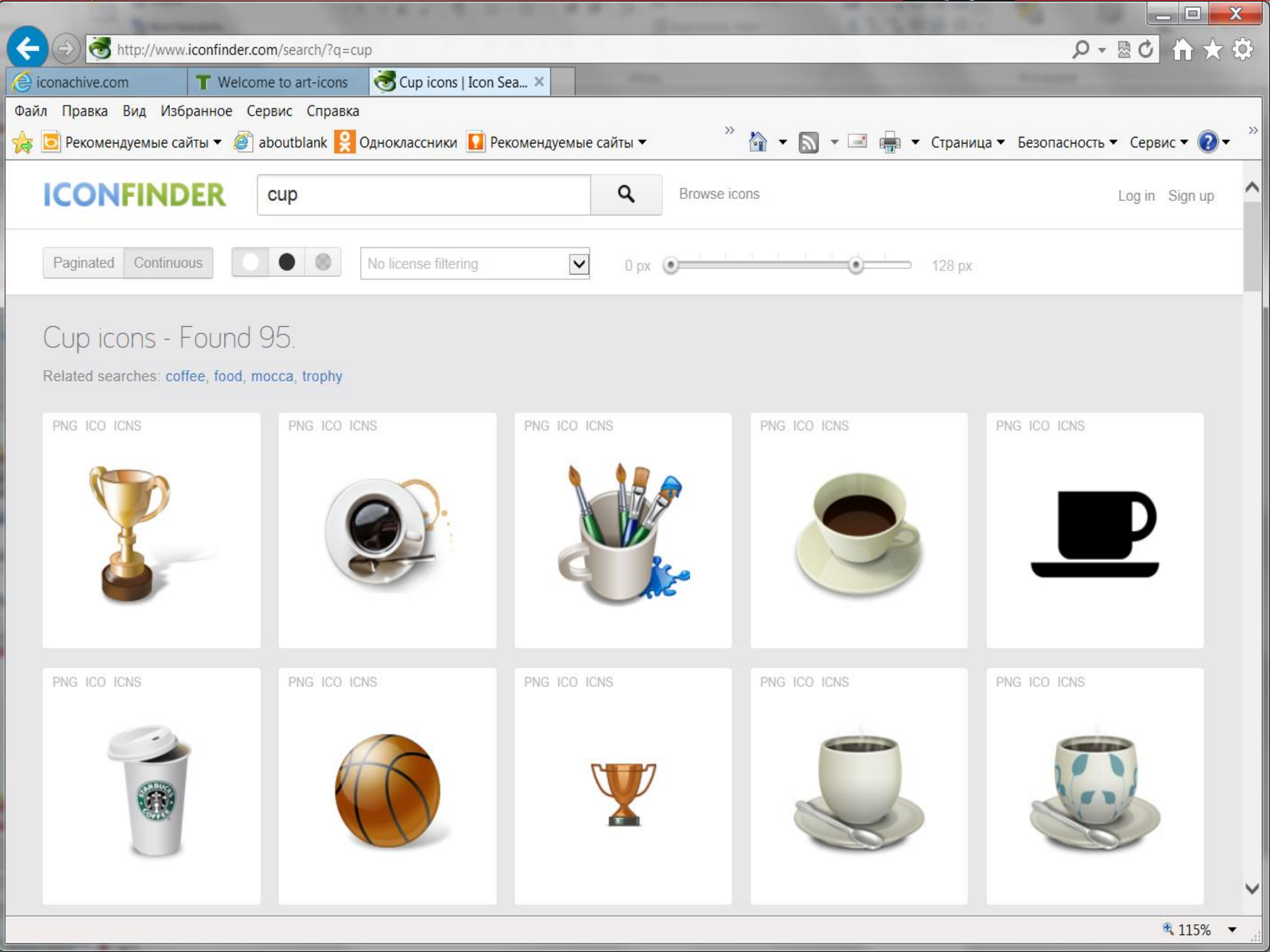
Студия русская редактор Windows. Она библиотеками поддерживает графических Включает русский украинский интерфейсы.



Toolbar коллекция иконок в программ презентациях и профессионалов



IconLove програм создани



web design.

Your purchase gets you over four hundred icons in semi-transparent Windows XP and 8-bit formats, in sizes of 16x16, 20x20, 24x24, 32x32, and 48x48 pixels, and in three states: normal, disabled, and highlighted. You will get the icons in all of the following file formats: Windows Icon (ICO), Bitmap (BMP), GIF, and PNG.

Purchase for \$39.95 only



Perfect Toolbar Icons



Your cart is empty. Click on icons to purchase them.

>> Database Toolbar Icons



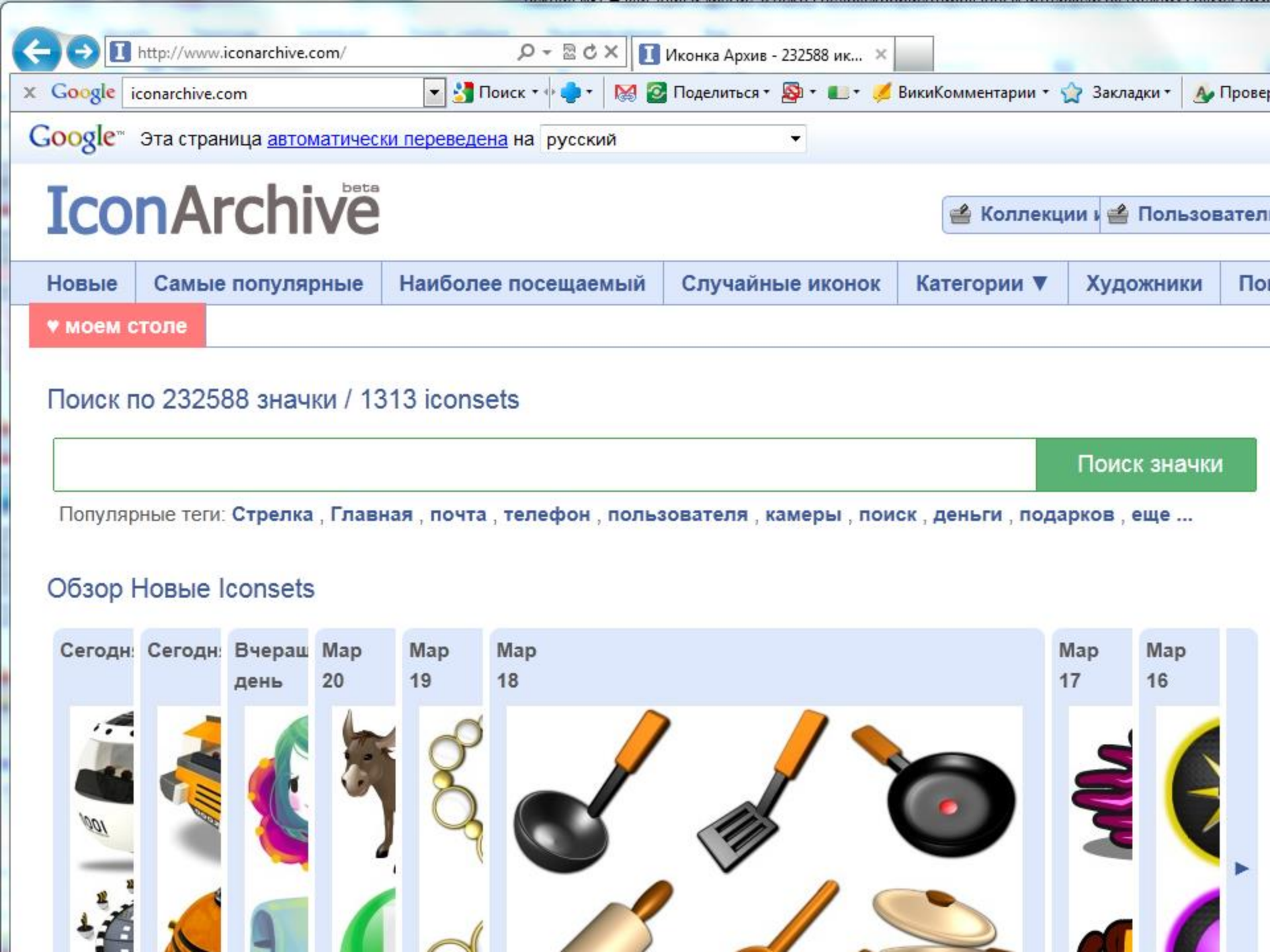
>> People Icons for Vista



>> Multimedia Icons for Vista



> Icon Usage Agreement


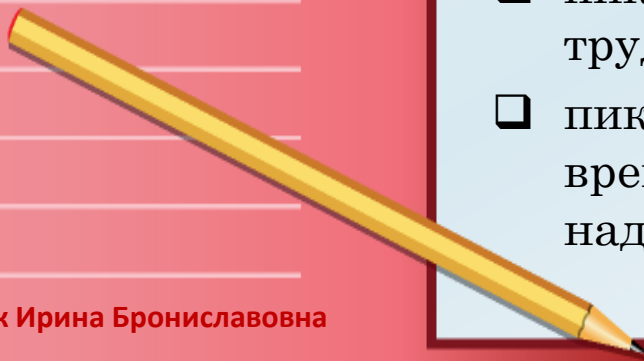




Разработка объектов пользовательского интерфейса

В графических пользовательских интерфейсах мы часто можем встретить *окна с множеством одинаковых пиктограмм, имеющих надписи.*

Пиктограммы имеют небольшой размер, и их может использоваться множество. Существуют десятки разных видов пиктограмм.




В современных компьютерных системах нельзя *достичь* тех *ограниченных условий*, при которых *пиктограммы могут быть эффективными.*

Достоинства:

- ☐ пиктограмма небольшого размера занимает меньше экранного пространства, чем надпись.

Недостатки:

- ☐ чем меньше экранная кнопка, тем больше времени требуется на то, чтобы ею воспользоваться, и тем труднее обнаружить ее на экране.
 - ☐ пиктограмму небольшого размера трудно сделать различимой.
 - ☐ пиктограммы требуют больше времени на создание, чем надписи.
- 

Разработка объектов пользовательского интерфейса

Геральдические символы
английской знати



Тенденция к чрезмерному
использованию графики
стала препятствием для
разработки хороших
интерфейсов.

1. Описание пиктограмм
при помощи слов.

Пример: описываются
геральдические символы
английской знати. Вот пример
геральдического описания щита:
«В зеленом поле золотой на
задних лапах лев от серебряного
поля отгорожен».



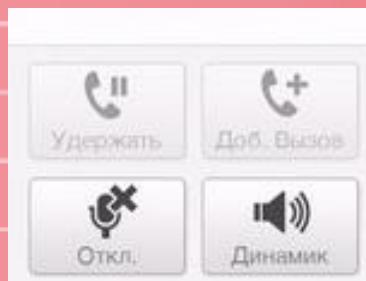
2. Использование цветовых палитр для
выделения пиктограмм.

Хортон отмечает, что «*выделение цветом
может быть бесполезным, если используется
слишком много цветов или если имеется
слишком много пиктограмм одного цвета*».

Правила:

- ☐ *если пиктограммы расположены
близко друг к другу, не следует
использовать более семи цветов;*
- ☐ *если же пиктограммы разбросаны по
всему экрану, то число цветов не
должно превышать четырех.*

Разработка объектов пользовательского интерфейса



Пример:

Разработчики интерфейса для одной из телекоммуникационных программ должны были предусмотреть в нем отображение следующих состояний: **набор номера, звонок, занятость линии, отсутствие сигнала в линии, чрезмерный шум на линии, попытка соединения, успешное соединение.**

После нескольких месяцев подбора для этой цели пиктограмм и их тестирования разработчики пришли к выводу, что невозможно создать такой набор пиктограмм, который мог бы однозначно передать пользователю требуемую информацию.

Вопрос: «Как можно графически отобразить звук сигнала «занято»?», после чего было решено использовать список соответствующих слов вместо пиктограмм.

Результаты тестирования, подтвердили эффективность этого решения.

Вывод

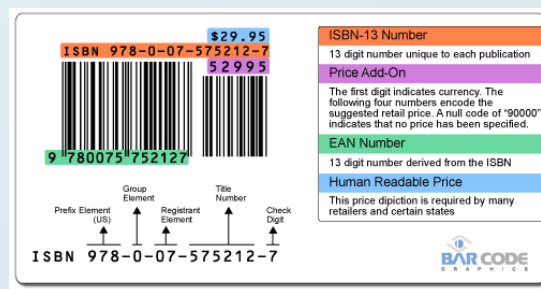
Библиографические пиктограммы
(<http://www.scran.ac.uk/iconstd/>)



SCRAN является благотворительной для обучения в режиме онлайн ресурсной базой с более чем 360000 авторских изображений и средств массовой информации из музеев, галерей, архивов и СМИ.



Самыми понятными из них, наверное, являются те, на которых есть надписи Menu и AND, а также, особенно если вы профессионально занимаетесь книгами, ISBN.



Таким образом, как это ни удивительно, пиктограммы нарушают принцип видимости, поскольку именно смысл пиктограмм остается невидимым.


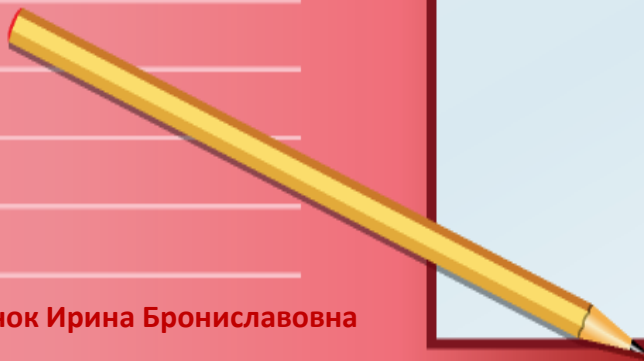
Используйте пиктограммы только в тех случаях, когда исследования показывают, что их применение может быть более эффективным.

В других случаях слова лучше.




Глава 5

«Навигация и другие аспекты человекоориентированных интерфейсов»



Тема 16. Способы и средства помощи в человекоориентированных интерфейсах

1. Особенности человекоориентированного интерфейса.
 2. Средства помощи в человекоориентированных интерфейсах.
 3. Операции вырезать и вставить в человекоориентированных интерфейсах.
 4. Сообщения пользователю в человекоориентированных интерфейсах.
 5. Упрощение входа в систему.
 6. Автоповтор и другие приемы работы с клавиатурой.
- 

Человекоориентированный интерфейс

Интерфейс с рабочим столом имеет довольно низкую эффективность, так как, находясь на его уровне, вы не можете выполнять свои задачи.

В человекоориентированном интерфейсе, в котором нет ни рабочего стола, ни приложений, пользователь может в любой момент работать с необходимым ему содержимым.

Мы можем отказаться от использования файлов и файловых имен, оставив только одно пространство — **пространство содержимого**.

В такой среде не требуется открывать или закрывать документы — вы просто уменьшаете масштаб системы до их размера и начинаете работать.



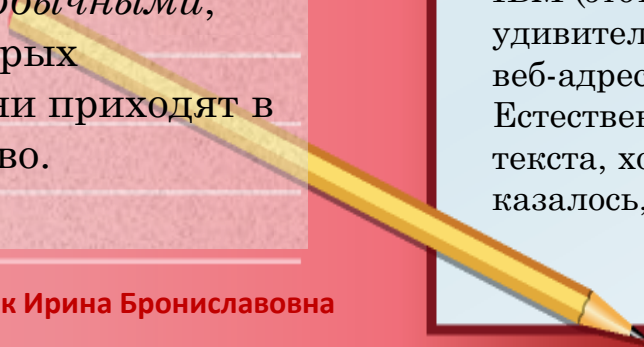
Вам не требуется открывать приложения — вы просто копируете чистый документ (или что-то еще).


Игру не требуется запускать, вы просто масштабируетесь до нее (причем, если игра многопользовательская, это можно сделать прямо по ее ходу).



Системы SwyftWare и Canon Cat

Выводы:

-  **Начинающие пользователи** считают, что системы, в которых нет рабочего стола или файловых имен, очень просты в использовании.
 -  **Опытным пользователям** современных систем эти принципы построения интерфейсов кажутся настолько необычными, что без некоторых объяснений они приходят в замешательство.
- 




Результаты исследований, на основе которых были разработаны системы **SwyftWare** и **Canon Cat**.

Пример: Рассмотрение компанией IBM компьютера Canon Cat на предмет возможного интереса к его интерфейсу.

Для выполнения оценки интерфейса компания IBM выбрала двух своих бывших сотрудников (чтобы избежать заключения соглашения о неразглашении полученной информации), которые были довольно опытными пользователями персональных компьютеров.

Цель исследования: смогут ли они быстро изучить интерфейс без помощи руководств, онлайн-овой системы подсказок или пособий.

Результат: у них ничего не получилось. Они пытались запустить текстовый процессор с помощью целого набора команд, начиная от косых черт из языка управления задачами с использованием перфокарт IBM (этот безобразный стиль разделителей, как ни удивительно, сохранился до сих пор и встречается в веб-адресах) и заканчивая командами IBM DOS. Естественно, это приводило лишь к набору и удалению текста, хотя они и делали свои попытки, как им казалось, на «уровне систем».



Самообучающийся интерфейс

***Интерфейс должен
быть по возможности
самообучающимся.***

Это не означает, что интерфейс должен быть интуитивным, а означает, что пользователь может легко найти понятные объяснения или инструкции в тех случаях, когда он в них нуждается.

Для упрощения процесса изучения при первой активизации продукта на экране должен отображаться экран с текстом инструктивного содержания.

Учебник и полное руководство по использованию должны быть частью интерфейса, доступной в любой момент.

Справочные экраны должны быть просто частью содержания, и для их использования не должны применяться какие-то особые методы или средства. Для поиска необходимой информации должно быть достаточно лишь использовать функцию LEAP по любому ключевому слову или же несколько раз нажать на клавишу <LEAP Again>.

Также можно сначала просто увеличить масштаб справочной области, а затем той ее части, которая вам необходима.

Справочные системы в пользовательских графических интерфейсах являются дополнением к основной части интерфейса, внедряются отдельно и работают по своим собственным правилам, что создает дополнительные трудности как для пользователей, так и для разработчиков.

Особенности человекоориен- тированного интерфейса

Допустим, вы пытаетесь
что-то ввести в
заблокированный текст

С помощью клавиши
<LEAP> или
непосредственно мыши вы
можете переместить курсор
в заблокированный текст.
Если теперь вы
попытайтесь ввести какой-
то текст, как должна
реагировать система?

Более или менее традиционным методом реагирования в данном случае является подача звукового сигнала и, возможно, мелькание экрана или панели меню для того, чтобы показать, что вы делаете что-то неправильно; после чего иногда может появиться диалоговое окно с сообщением, что совершается недопустимое действие.

Однако такая реакция не подходит для человекоориентированного интерфейса, поскольку выдаваемое сообщение необходимо закрывать, и любые введенные символы теряются, что приводит к нарушению аналога **Первого закона робототехники Азимова**, суть которого сводится к тому, *что все содержание пользователя является священным.*

См. <http://www.outzone.ru/post/3041>
(30 законов робототехники Азимова)

Особенности человекоориен- тированного интерфейса

Вырезать и вставить

В обычных интерфейсах также имеется и другая проблема, связанная с методом «вырезать и вставить» (cut-and-paste).

При использовании этой функции многие пользователи сталкивались с ситуацией, когда какая-то часть работы пропадала после случайного повторения операции вырезания до того, как первый вырезанный отрывок был вставлен.

Когда текст удален, он не должен исчезать в небытии, и особенно он не должен перемещаться в невидимый буфер вырезания.

Одно из решений может заключаться в том, чтобы помещать вырезанный текст в конце какого-то документа, в котором собираются удаленные элементы.

Такой документ может быть самым обычным текстовым документом. Важным здесь является то, что с удаляемым текстом не происходит ничего мистического, и для обнаружения документа, содержащего удаленный текст, не требуется использовать какие-то специальные команды или методы.

Для этой цели такой документ может начинаться с обычного текста, например: «Этот документ содержит удаленный текст». Пользователь может ввести любую фразу в качестве целевого объекта поиска в этом документе, и, конечно, любая фраза из удаленного текста также может быть использована для выполнения поиска, как будто она и не является удаленной.

Особенности человекоориен- тированного интерфейса

Любой человекоориентированный метод удаления:

- ☐ не отличается в работе от других команд;
- ☐ ничего не подвергает риску при удалении или перемещении текста;
- ☐ не использует особый буфер или другое скрытое место «системного уровня», в которое текст перемещается;
- ☐ удаляет отдельные символы таким же образом, как и отрывки, состоящие из множества символов;
- ☐ допускает отмену или повторение действия.

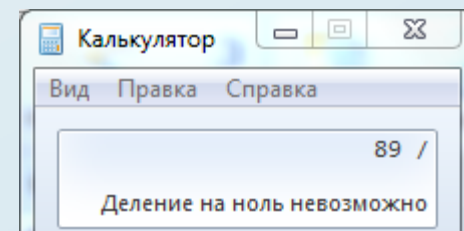
Особенности человекоориен- тированного интерфейса

*Когда вы пытаетесь
написать текст
сообщения об ошибке,
пожалуйста,
остановитесь и
переделайте интерфейс
таким образом, чтобы
условие, при котором
это сообщение об
ошибке вызывается, не
возникало.*

Другими словами, сообщение об ошибке сигнализирует именно об ошибке, но о той, которая имеется обычно в структуре системы или интерфейса, а не совершается со стороны пользователя.

Например, при разработке одного пакета для выполнения арифметических вычислений, на первый взгляд, казалось, что не существует способа избежать выдачи сообщения об ошибке в тех случаях, когда пользователь пытается выполнить деление на 0.

Однако более удачным методом является выдача значения, названного «неопределимо».



В арифметическом стандарте № 754 Института инженеров по электронике и электротехнике (**IEEE (Institute for Electrical and Electronic Engineers)**) для этой цели используется выражение NaN, что означает «не является числом» (not a number). Следует отметить, что арифметические операции, применяемые к *неопределимым* значениям, являются вполне определенными (например, *неопределимо* + 3 = *неопределимо*), а полученные в таких вычислениях результаты являются более полезными и информативными, чем просто остановка вычислений.

Сообщения: учебный пример

Одним из интересных решений, полученных в результате устранения набора сообщений об ошибках, является набор методов, разработанных доктором **Джеймсом Уинтером** (James Winter) для сохранения и загрузки информации из внешней памяти в компьютере Canon Cat.

Еще один пример, иллюстрирующий следующий принцип: ***не предоставляйте пользователю тех средств управления, которые должны работать всегда или никогда.***

Например, пользователю может быть дано предупреждение о том, что он пытается закрыть не сохраненный файл. Изначально имелось две команды для сохранения и открытия данных, каждая из которых подается с помощью специальной клавиши.

Доктор Уинтер показал, что требуется только одна команда и что такой интерфейс будет более безопасным.

Метод был использован в промышленно выпускаемой модели и оказался вполне успешным и популярным. В частности, были устранены многие ошибки, которые совершаются при сохранении и загрузке информации и зачастую приводят к потере данных.

Идея Уинтера

Идея Уинтера заключалась в следующем. Использовалась только одна команда под названием DISK. При ее выполнении производился анализ ситуации, и автоматически совершалось необходимое действие. Для того чтобы показать, как работает эта система, **Уинтер** составил простую таблицу, в которой отразил, какие действия будут выполнены системой при различных условиях применения команды DISK.

Состояние диска	Состояние памяти		
	Не измененное	Измененное	Пустая память
Не измененное	нет действия	сохранение	нет действия
Измененное	загрузка	выдача предупреждения	загрузка
Пустой диск	сохранение (дублирование)	сохранение	нет действия

Если вы загрузили в память компьютера рабочую среду с диска и не изменили ее, состояние памяти остается «не измененным». Если этот диск находится в машине, и вы нажали на клавишу <DISK>, больше ничего делать не нужно — содержание памяти автоматически сравнивается с содержанием диска. Если же диск был извлечен и заменен на другой («измененное» состояние диска), система стирала память и производила загрузку с нового диска, т.к. системе известно, что копия текущей рабочей среды имеется на дискете и поэтому не будет потеряна. В третьем случае дискета может быть чистой (пустой, не использованной), что означает, вероятно, желание пользователя сделать копию рабочей среды, что системой и выполняется. Форматировать дискеты пользователю не требуется, поскольку эта операция выполняется автоматически при необходимости. Если система совершает ошибку в каком-либо из этих случаев, данным не причиняется ущерба.

Упрощение входа в систему

При входе в большинство систем пользователи выполняют больше работы, чем это необходимо.

Сначала требуется сообщить, кто вы, для чего вы указываете свой «идентификатор», «онлайновое имя» или «системное имя», и затем вводите свой пароль.

Как предполагается, с помощью имени вы сообщаете системе о том, кто вы, а пароль позволяет предотвратить несанкционированный доступ к вашей учетной записи.

На самом деле вы дважды сообщаете системе о том, кто вы. *По логике все, что от вас должно требоваться в этом случае, — набрать пароль.* И это не снизит безопасность вашей системы.

Утверждение, что ввод двух разных цепочек символов увеличивает надежность, ошибочно.

Если онлайновое имя состоит из j символов, а пароль — из k символов, то пользователь для входа в систему должен ввести $j+k$ символов, и только k символов из этой суммы являются неизвестными для потенциального нарушителя вашей безопасности. Если пароль был выбран случайным образом (что является самым лучшим методом) из набора в q символов, вероятность угадать учетную запись пользователя с первой попытки равна $1/q^k$.

Удлинение пароля даже на один символ и отказ от использования имени уменьшают вероятность угадывания пароля на величину q и избавляют пользователя от необходимости вводить лишние $j-1$ символов, к тому же в двух полях (вместо одного).

Убрав поле для имени и увеличив минимальную длину пароля на один символ, мы получаем большую степень безопасности, используем меньше экранного пространства и достигаем большей простоты использования. Мы не теряем ничего.

Упрощение входа в систему

Для некоторых программ имеет смысл использовать менее банальные методы обеспечения безопасности (как, например, *использование «отпечатков голоса»* (voiceprinting) или *пальцев*, или других неизменяемых физических характеристик пользователя), хотя в этом случае *вы не сможете сказать своему доверенному лицу, как войти под вашей учетной записью.*

Здесь возникает следующий вопрос: как обеспечить уникальность символьных паролей? Как избежать случаев, когда два или более пользователей выберут одинаковый пароль?

Можно предоставить системе самой назначать пароли. Однако это приведет к тому, что генерируемые системой пароли будут плохо запоминаемыми (как, например, *2534-788834-003PR7* или *ty6*>fj`d%d*).

Упрощение входа в систему

Существует много способов создания запоминающихся паролей, и вы всегда можете дать пользователю возможность выбора из пяти или шести таких способов.

Например, вы можете предложить компьютеру случайно выбрать два прилагательных и одно существительное из большого словаря и предоставить вам такой список:

- ☐ эксклюзивный уродливый тюлень;
- ☐ вкусный человекоподобный оракул;
- ☐ старая свободная папайя;
- ☐ цветущий маленький лабиринт;
- ☐ скверная репообразная история.

из которого пользователь сможет сам выбрать наиболее ему понравившийся вариант.

В английском словаре возможно около двух триллионов подобных комбинаций.

Даже пробуя по миллиону комбинаций в день, вы будете подбирать пароль более 25 лет. Такая безопасность достаточна.

Автоповтор и другие приемы работы с клавиатурой

Первая коммерчески успешная пишущая машинка была изобретена в сентябре **1867** года американцем **Кристофером Шоулзом**

Профессор
Университета
Вашингтона Август
Дворак нашёл и
множество
недостатков,
присущих QWERTY
раскладке.



Он вычислил, что более половины символов набирается в верхней строке, что вынуждает печатающего смещать пальцы от ряда, над которым они располагаются в исходной позиции. Кроме того, большая часть символов набирается левой, как правило, не доминирующей рукой, а 30% печати приходится на нижнюю строку, наиболее «медленную» и сложную для правильного позиционирования.

Раскладка **Дворака**

В настоящее время, рядом с вашим компьютером имеется обычная буквенно-цифровая клавиатура. Множество попыток реформирования клавиатуры (например, раскладка **Дворака**) оказались бессильными для того, чтобы преодолеть инерцию многих миллионов людей, которые научены вслепую набирать на клавиатуре с раскладкой **QWERTY**.



~	!	@	#	\$	%	^	&	*	()	-	+	←
	1	2	3	4	5	6	7	8	9	0		=	Backspace
Tab	Q	W	F	P	G	J	L	U	Y	:	{	}	
										;	[]	\
←	A	R	S	T	D	H	N	E	I	O	"	'	Enter
Backspace													↵
Shift		Z	X	C	V	B	K	M	<	>	?	Shift	
↵									,	.	/	↵	
Ctrl	Win	Alt									Alt Gr	Win	Menu
												Key	Ctrl

Латинская раскладка клавиатуры, созданная в 2006 году **Шаем Коулманом** (Shai Coleman). Является альтернативой латинским раскладкам **QWERTY** и **Дворака**. Третья по популярности латинская раскладка клавиатуры в мире

Автоповтор и другие приемы работы с клавиатурой

Все, что мы можем сделать в качестве разработчиков интерфейсов, — это ходить где-нибудь с краю и *делать небольшие улучшения, не требующие серьезного переучивания.*

Для того чтобы начался автоматический повтор, в большинстве клавиатур требуется удерживать клавишу в нажатом положении в течение 500 мс. *Это является примером фиксированной задержки.*

Однако имеет смысл не использовать в интерфейсах фиксированный интервал задержки. Любая фиксированная задержка может оказаться как слишком большой, так и слишком малой, в зависимости от пользователя и обстоятельств.

Например, у пользователей интерфейса Macintosh следующая ситуация вызывает особенное раздражение: чтобы изменить имя файла после открытия тома или папки, вам требуется щелкнуть по имени и подождать полсекунды до тех пор, пока не появится специальное обрамление или изменение цвета, указывающее на переход в режим редактирования. Это было предусмотрено для того, чтобы пользователь мог выбрать имя файла одним нажатием на кнопку мыши без риска случайного его изменения. После перехода в предварительный режим редактирования вы должны щелкнуть по имени еще раз, чтобы перевести систему в состояние редактирования.

Автоповтор и другие приемы работы с клавиатурой

Джон Бумгарнер (John Bumgarner), работавший в компании Information Appliance, предложил хорошее решение проблемы автоповтора. Он заметил, что в большинстве фонетических языков одна буква почти никогда не встречается три раза подряд. Он также заметил, что автоповтор редко используется, если букву требуется повторить менее пяти раз (в этом случае пользователь просто нажимает на клавишу необходимое число раз).

При использовании метода Бумгарнера автоповтор начинается, если клавиша удерживается более 100 мс после третьего подряд нажатия на клавишу.

Другими словами, чтобы получить строку, состоящую из знаков равенства, требуется нажать следующую последовательность:

==↓

После этого клавишу со знаком равенства следует удерживать в нажатом положении до тех пор, пока не появится необходимое количество символов, и затем отпустить.

Автоповтор и другие приемы работы с клавиатурой

Многократное нажатие на одну и ту же клавишу выполнить быстрее, чем набирать разные символы, и GOMS-анализ показывает, что задержка перед началом автоповтора падает:

- ❑ с 700 мс при обычных методах
- ❑ до 400 мс в предложенном методе.

Метод автоповтора **Бумгарнера** достаточно прост в использовании и, как показали тестирования, такой автоповтор никогда не запускается случайно (даже если ваша кошка сядет на клавиатуру).

Отрицательной стороной, которая свойственна также и стандартному методу автоповтора, является то, что автоповтор работает как невидимая функция, которая нигде в системе не обозначена.

Автоповтор и другие приемы работы с клавиатурой

Хорошо разработанные компьютеры и информационные устройства снабжаются **аккордными клавиатурами** (chord keyboards), чтобы в программном обеспечении можно было предусматривать распознавание одновременного нажатия нескольких клавиш.

Старые и более примитивные компьютеры имели клавиатуры, в которых только некоторые специальные клавиши (например, <Shift>) можно было одновременно нажимать вместе с другими клавишами.

Аккордные клавиатуры позволяют решить ряд сложных интерфейсных проблем. Например, рассмотрим наложение символов. Требуется логически ясный метод создания двух символов в одном месте. Например, для того чтобы сделать знак доллара с помощью наложения буквы s и вертикальной черты (|), должна быть предусмотрена возможность одновременного нажатия клавиш:

s↓ |↓ |↑ s↑


Это не мешало бы совмещенному нажатию на клавиши, которое обычно происходит при большой скорости набора и при котором клавиша, нажатая первой, отпускается только после того, как нажимаются одна или несколько других клавиш. Слово *the* часто набирается не так:

t↓ t↑ h↓ h↑ e↓ e↑


а вот так (приведем один из множества возможных вариантов):

t↓ h↓ e↓ t↑ h↑ e↑

Современные клавиатуры и их программное обеспечение допускают использование таких совмещенных нажатий клавиш, что называется **циклическим буфером** (rollover).



Автоповтор и другие приемы работы с клавиатурой



Большинство клавиатур имеют n -клавишный циклический буфер. Это означает, что система сможет различить n -е количество одновременно нажатых клавиш.

С учетом человеческой анатомии *коэффициент n вряд ли должен превышать 10, хотя с технической точки зрения вообще нет необходимости его ограничивать, если компьютер оснащен аккордной клавиатурой.*

Автоповтор и другие приемы работы с клавиатурой

Кроме того, наложение символов полезно для написания математических и других специальных символов, а также в языках программирования (например, APL).

С учетом общепринятого метода создания наложений с помощью нажатия на одну клавишу во время того, как удерживается другая, *знаки ударения и диакритические символы могут также рассматриваться как налагаемые символы* и вводиться аналогичным образом. Например, в компьютере Macintosh букву *é*, как в слове Dupré, можно набрать с помощью сложной последовательности клавиш:

Option↓ e↓↑↑ e↓ e↑

Обратите внимание на то, что здесь используется модальный метод типа «глагол-существительное», что является нарушением собственных принципов компании Apple. Кроме того, этот метод работает непоследовательно. Если вы используете нижеприведенную последовательность, то будет получена кавычка, за которой стоит буква t, а не буква t со знаком ударения, как можно было бы ожидать:

Option↓ t↓↑↑ t↓ t↑

Если же наложение выполнять с помощью квазирежима, ввод знаков акцента и других диакритических символов упрощается и делается более последовательным:

e↓ '↓ '↑ e↑

Вы нажимаете на букву e и, удерживая ее, нажимаете на символ ударения. То же самое сочетание можно получить и в обратном порядке:

'↓ e↓ e↑ '↑

Логически нет никакой разницы, в каком порядке вы выполняете эту операцию.

Автоповтор и другие приемы работы с клавиатурой

Может возникнуть вопрос: почему вместо наложения символов нельзя просто включить нужные символы в шрифтовые наборы, ведь наши дисплеи являются полностью растровыми?

Действительно, так можно поступить, однако не все захотят тратить время на разработку новых символов и их установку в каждый шрифтовой набор, в котором мы хотим их использовать.

С другой стороны, вряд ли в современном компьютере нельзя получить те же возможности, которые были легко доступны с помощью обычных механических печатных машинок.


Наложение не должно ограничиваться только двумя символами. Любые символы могут налагаться друг на друга, как, например, в следующей последовательности:
Shift↓ s↓ Shift↑ |↓ /↓ /↑ |↑ s↑

Такая последовательность даст в результате знак доллара, перечеркнутый косой чертой.

Функция наложения символов должна ограничиваться скорее только лишь эстетическими соображениями и доступностью для чтения, чем аппаратными или программными соображениями.



Вывод



Если используются n -клавишный циклический буфер и описанные выше методы наложения, то для обратной связи во время набора интерфейс может временно отображать пару налагаемых друг на друга символов в виде смежных символов. Смысл этого заключается в том, что *интерфейс не может отличить одновременное нажатие клавиш при быстром наборе от одновременного нажатия с целью наложения символов друг на друга до тех пор, пока клавиши не отпущены, после чего слияние накладываемых символов происходит автоматически.*

Также требуется радикальная реформа клавиатуры, связанная с удалением клавиши <CapsLock>. Эта клавиша порождает режим.

