

А.И. Калиновский

# Юзабилити:

как сделать сайт удобным





**А.И. Калиновский**

# **Юзабилити:**

**как сделать сайт удобным**



**МИНСК ООО «НОВОЕ ЗНАНИЕ» 2005**

---

УДК 004.92  
ББК 32.973.26-018.2  
К17

**Калиновский А.И.**

**К17 Юзабилити: как сделать сайт удобным / А.И. Калиновский. — Мн. : Новое знание, 2005. — 220 с. : ил.  
ISBN 985-475-130-9.**

Книга представляет собой практическое пособие по созданию удобного интерфейса. Впервые подробно рассмотрено понятие «юзабилити» и его основные аспекты. На примере известных отечественных и зарубежных порталов, интернет-магазинов показано, как улучшить интерфейс пользователя, а следовательно, повысить популярность Web-ресурса.

Адресована как новичкам, которые только приступают к созданию сайта, так и опытным Web-дизайнерам..

**УДК 004.92  
ББК 32.973.26-018.2**

**ISBN 985-475-130-9**

© Калиновский А.И., 2005  
© Оформление. ООО «Новое знание», 2005

# Оглавление

Введение .....	7
<b>Глава 1. Что такое интерфейс .....</b>	<b>8</b>
1.1. Общие сведения .....	8
1.2. Интерфейс в жизни .....	10
1.3. Почему качественный интерфейс обязателен .....	10
1.4. Человекоориентированный интерфейс .....	13
Резюме .....	14
<b>Глава 2. Пользователь — компьютер .....</b>	<b>15</b>
2.1. Структура взаимодействия «пользователь — компьютер» .....	15
2.2. Привычки .....	17
2.3. Опыт против разнообразия .....	20
2.4. Ограничения физические и умственные .....	21
2.5. Фокус внимания .....	24
2.6. Кратковременная память .....	27
2.7. Монотонность .....	28
2.8. Точность данных .....	29
2.9. Время протекания процессов .....	30
2.10. Что такое идеальный интерфейс .....	31
Резюме .....	33
<b>Глава 3. Интерфейс в программном обеспечении .....</b>	<b>35</b>
3.1. Виды программного обеспечения .....	35
3.2. Независимость интерфейса от других компонентов приложения .....	37
3.3. Проектирование программного обеспечения и интерфейса ....	39
Резюме .....	44
<b>Глава 4. Этапы создания интерфейса .....</b>	<b>45</b>
Резюме .....	49
<b>Глава 5. Компонировка Web-страниц .....</b>	<b>50</b>
5.1. Смысловые уровни Web-сайта .....	50
5.2. Основные способы верстки и их особенности .....	51
5.2.1. Текстовая верстка .....	51
5.2.2. Табличная верстка .....	52
5.2.3. Верстка при помощи фреймов .....	53
5.2.4. Флэш .....	55
5.3. Оформление главных страниц .....	56
5.4. Виды верстки .....	57
5.5. Понятие модульной сетки .....	59
5.6. Основные элементы компоновки страниц .....	60
5.6.1. Навигационные блоки .....	61
5.6.2. Информационные блоки .....	65



5.6.3. Сервисные блоки .....	67
5.6.4. Рекламные блоки .....	68
5.7. Пример компоновки .....	70
Резюме .....	71
<b>Глава 6. Элементы интерфейса .....</b>	<b>73</b>
6.1. Обзор аппаратного обеспечения .....	73
6.1.1. Монитор .....	73
6.1.2. Мышь .....	74
6.1.3. Клавиатура .....	75
6.1.4. Скорость соединения и мощность компьютера .....	76
6.2. Элементы взаимодействия системы с пользователем .....	77
6.2.1. Окно браузера .....	77
6.2.2. Формы .....	80
6.2.3. Апплеты .....	80
6.2.4. Диалоговые окна .....	82
6.3. Элементы навигации .....	83
6.4. Поиск .....	98
6.5. Элементы содержания .....	100
6.5.1. Текст .....	100
6.5.2. Особенности восприятия текста .....	105
6.5.3. Поля ввода .....	107
6.5.4. Кнопки .....	109
6.5.5. Гиперссылки .....	109
6.5.6. Пиктограммы .....	111
6.5.7. Списки .....	114
6.6. Элементы оформления .....	116
Резюме .....	117
<b>Глава 7. Юзабилити в теории и на практике .....</b>	<b>119</b>
7.1. Определение юзабилити .....	119
7.2. Удобство работы с информацией в Web .....	122
7.2.1. По левую руку от светофора .....	122
7.2.2. Визуальное выделение важной информации .....	123
7.2.3. «Географическое» выделение важной информации .....	124
7.2.4. Выбор заголовка .....	125
7.2.5. Контрастность текста и шрифт .....	127
7.2.6. Кодировка .....	129
7.2.7. Грамматические ошибки .....	129
7.2.8. Как вы лодку назовете... ..	130
7.3. Удобство работы с навигацией .....	131
7.3.1. Размеры элементов. Школа юного снайпера .....	131
7.3.2. Откуда я пришел? .....	133
7.3.3. Сокращенная и полная форма .....	134
7.3.4. Не отвлекайте меня от работы! .....	135

7.4. Комфортность .....	136
7.4.1. Комфортность в жизни и в Web .....	136
7.4.2. Разрешение монитора .....	136
7.4.3. Браузеры .....	137
7.4.4. Всплывающие окна .....	138
7.4.5. Полосы прокрутки .....	140
7.4.6. Забытые герои .....	141
7.4.7. Считаем шаги .....	142
7.5. Обучаемость .....	145
7.5.1. Стимулы и предпосылки .....	145
7.5.2. Метафора .....	146
7.5.3. Идиомы .....	147
7.5.4. Очевидность .....	147
7.5.5. Привычность решений (роль стандартов) .....	147
7.6. Запоминаемость .....	148
7.6.1. Автоматизм, привычка и простота .....	148
7.6.2. Работа над ошибками .....	150
7.6.3. Право выбора всегда усложняет жизнь .....	152
7.6.4. Sapienti sat .....	152
7.7. Предсказуемость .....	153
7.7.1. Законы жанра .....	153
7.7.2. Цена ошибки .....	155
7.7.3. Соответствие бизнес-процессу .....	156
7.8. Тестирование .....	157
7.8.1. Общий подход .....	157
7.8.2. Кто тестирует .....	159
7.8.3. Что тестируется .....	159
7.9. Разбор уровней юзабилити .....	160
Резюме .....	167
<b>Глава 8. Нестандартизованные стандарты .....</b>	<b>169</b>
8.1. Как появляется стандарт .....	169
8.2. Стандартизованный интерфейс .....	171
8.3. Стандарт: шаблон или копия .....	172
Резюме .....	175
<b>Глава 9. Справка и помощь на Web-сайте .....</b>	<b>176</b>
9.1. Справочная система .....	176
9.2. Сообщения об ошибках .....	178
9.3. Контекстная помощь .....	179
9.4. Всплывающая подсказка .....	180
9.5. Статусная строка .....	182
9.6. Дополнительная поддержка .....	184
Резюме .....	185

<b>Глава 10. Errare humanum est</b> .....	186
10.1. Кто виноват? .....	186
10.2. Типы ошибок .....	186
10.3. Суть предотвращения ошибок .....	187
10.4. Отношение пользователей к сообщениям об ошибке .....	187
10.5. Ошибки ввода данных и их предотвращение .....	189
10.6. Ввод неверных данных и грамматические ошибки .....	193
10.7. Ошибки системы .....	193
10.8. Ошибки незнания предметной области и их предотвращение .....	194
10.9. Ошибки точности движений и их предотвращение .....	195
10.10. Инерционные ошибки и их предотвращение .....	196
10.11. Устранение побочных эффектов .....	197
10.12. Обработка ошибок и обратная связь .....	199
10.13. Правильное сообщение об ошибке .....	202
Резюме .....	202
<b>Приложение</b> .....	204
1. Ozon.ru сменил лицо .....	204
2. Десять тысяч чек-боксов .....	206
3. Юзабилити форума на books.ru .....	208
4. Порочность нестандартных решений .....	210
5. Трехминутный анализ .....	212
6. Поиск на Onliner.by .....	214
7. «А где финики? Финики-то где?..» .....	215
8. Так делать нельзя .....	216
9. Читаемость текста .....	216
10. «Попрыгунчики» .....	218



## Введение

Обычно в начале книги принято говорить о необходимости и важности той информации, которая будет изложена, сослаться на огромный опыт автора, рассказывать то, как ему «открылась истина», и выражать благодарность родственникам и друзьям. Здесь ничего этого не будет.

Я буду учиться вместе с читателем. Буду исследовать некоторые вопросы «по горячим следам» и порой становиться обладателем тех знаний, которые описываю, буквально за день до того, как они появятся на страницах книги (особенно когда дело будет касаться примеров). Я обладаю набором знаний и опыта, не намного большим, чем читатели этой книги, просто систематизация информации порой важна так же, как и сама информация. Основной целью книги я ставлю именно систематизацию знаний о том, что такое интерфейс, удобство использования, и о том, как сделать ваш сайт комфортным, запоминаемым и легким для восприятия. У вас наверняка когда-нибудь были любимые домашние тапочки или куртка, сшитая «как на вас». И вы, уверен, вспоминаете эти вещи теплыми словами, когда надеваете узкие туфли или новомодный пиджак. Так пусть и посетители вашего сайта, число которых станет множиться с каждым днем, будут также вспоминать ваш сайт, попадая в темные закоулки Всемирной паутины, где о них никто и никогда так не заботился. И даже если вы не услышите личных слов благодарности, вас будут радовать цифры — статистика посещаемости, уровень продаж, падение рейтинга конкурентов и т.д.

# Что такое интерфейс

## 1.1. Общие сведения

Всякая система, используемая человеком в своих нуждах, должна обладать механизмами, обеспечивающими взаимодействие с ней.

Если речь идет о бытовой технике, то такими механизмами являются кнопки на панели управления или дистанционном пульте. Например, у меня на кухне есть таймер, который приводится в действие металлической пружиной. Для его установки необходимо нажать на ручку в середине циферблата, повернуть, установив стрелку на нужное количество минут, и отпустить. В механических предметах это те части, которые предназначены для подгонки, закрепления, настройки узлов механизмов. Если мы говорим об интернете, то взаимодействие пользователя с сервером будут обеспечивать операционная система, браузер и компоненты HTML-страниц, отображаемые в окне браузера.

***Интерфейс*** — часть некоторой системы, предназначенная для обеспечения взаимодействия с ней человека либо другой системы. Зачастую эффективность использования всех функций системы и эффективность работы самой системы определяется в большой степени тем, как построен ее интерфейс. Для примера можно рассмотреть работу современного телевизора. В нем есть масса функций, вроде телетекста, сохранения текущих настроек, автоматического сканирования диапазона каналов и т.д., но если способы взаимодействия человека и телевизора сложны и непродуманны, то обмен данными между человеком и системой (телевизором) будет неэффективен. Придется тратить на настройку много времени.

От способов построения интерфейса также зависит правильность понимания отдаваемых системе команд, что позволяет не навредить системе или используемым ею компонентам и работать эффективно.

В этой книге пойдет речь о более узком применении понятия «интерфейс» — графический интерфейс пользователя (GUI —

Graphic User Interface) Web-приложений, к которым относятся Web-сайты, интернет-магазины, поисковые порталы, корпоративные автоматизированные системы и т.д.

Поскольку Web-интерфейс ограничен в возможностях и обладает рядом особенностей, то он постоянно требует пристального внимания разработчика.

Во-первых, простота, а порой и скудность Web-интерфейса диктуется спецификацией языка разметки HTML. Компонент, который не обеспечивается стандартным кодом HTML, может быть создан только искусственным образом. Например, невозможно создать динамическую древовидную структуру, потому что в спецификации нет никакого тега `<tree>` или чего-то подобного. Однако «склеить» динамическое «дерево» можно, например, с помощью элементов многоуровневого списка, JavaScript и свойств «видимый/невидимый».

Во-вторых, способы обмена информацией с пользователем строятся зачастую по общепринятым правилам, а не с учетом особенностей взаимодействия человек — система.

В-третьих, свой отпечаток накладывает специфика Всемирной паутины, ведь человек при малейших неудобствах в достижении своих целей может навсегда уйти «от вас» к другому партнеру только потому, что предлагаемый способ взаимодействия с вашей системой ему неудобен.

Конечно же, проблемы в пользовательском интерфейсе Web-сайта не приведут к таким катастрофическим последствиям, как неполадки в системе управления автомобилем, но они создадут усталость, напряженность и недовольство вашего посетителя, что может сказаться на вашем рейтинге и популярности вашего ресурса.

Интерфейс Web-сайта — это так называемые фронтальные компоненты системы, построенные при помощи языка разметки HTML и протокола передачи данных HTTP, которые позволяют пользователю выполнять необходимые задачи — получать информацию, общаться с другими людьми, развлекаться и т.д. Человек не считает нули и единицы и не вводит их. Он взаимодействует с системой через интерфейс пользователя, который переводит команды на язык, понятный системе.



## 1.2. Интерфейс в жизни

Представьте себе стиральную машину. Для управления ею разработчики сделали переднюю панель с ручками и кнопками. Если мы хотим заставить машину работать определенным образом, то достаточно покрутить ручку или нажать кнопку, т.е. воздействовать на машину посредством ее интерфейса. Чтобы понять, как работает стиральная машина, необходима хорошая инструкция либо опыт работы с бытовой техникой.

Мои познания в области интерфейсов программных систем помогли мне догадаться, как работает моя новая стиральная машина. Ручки управления одновременно являются кнопками (на моей машине их три); если на них нажать, они сравниваются с уровнем панели. Одной из ручек-кнопок устанавливается программа стирки и автоматически соответствующие ей скорость вращения барабана и температура воды. С помощью двух других осуществляется индивидуальный выбор этих параметров. В каком положении должны быть ручки-кнопки для индивидуальных настроек?

Я рассуждал так:

- нажатое состояние ручек больше похоже на неактивное, чем на активное, потому что, когда они нажаты, я не вижу надписей на них, т.е. своих установок;
- внешне панель управления не выглядит так, как будто необходимо фиксировать нажатием кнопки выбранный режим после ввода данных.

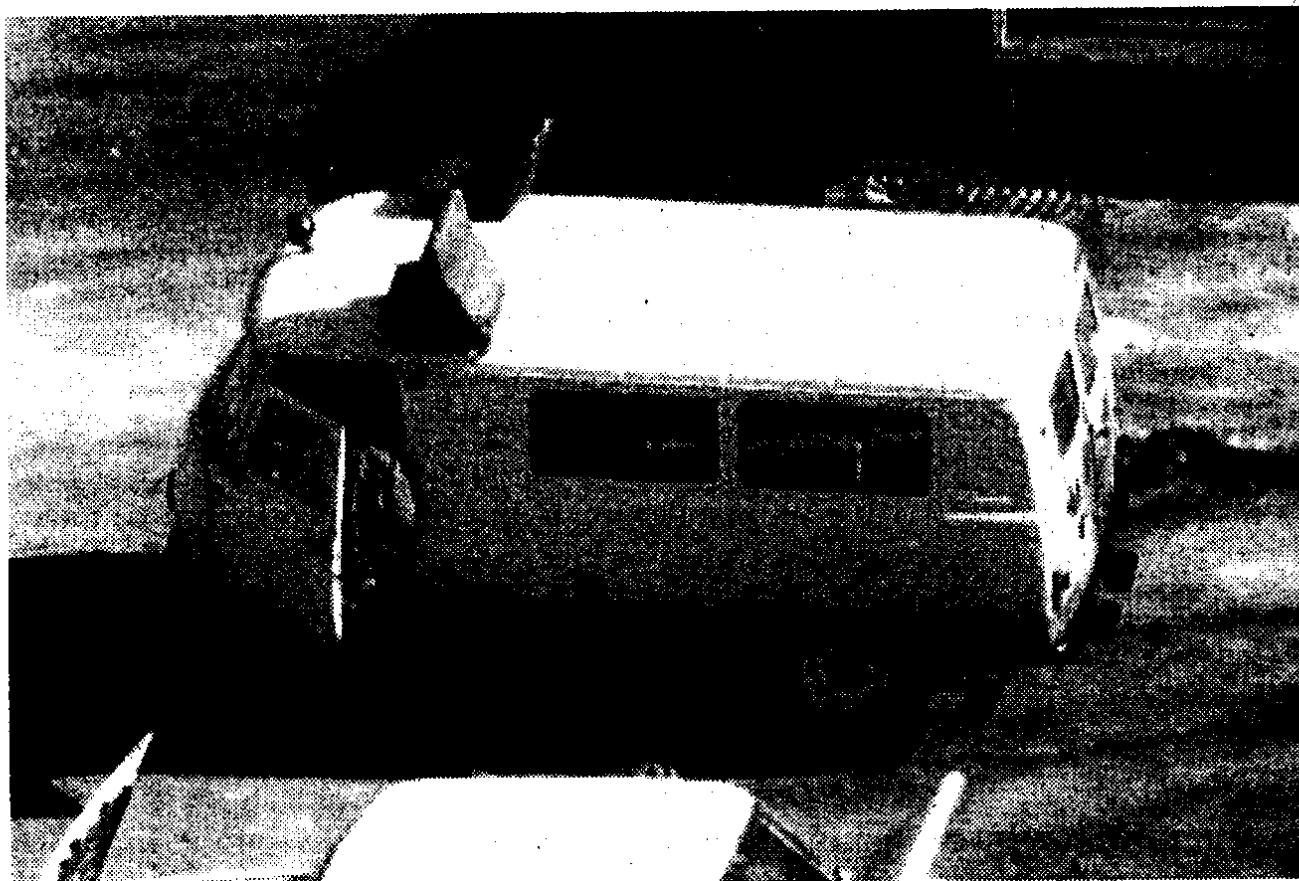
Позже выяснилось, что я оказался прав в своих предположениях о способе работы стиральной машины.

## 1.3. Почему качественный интерфейс обязателен

По совершенно неизвестной мне причине очень многие разработчики считают, что интерфейс далеко не самое главное для Web-сайта. Однако они сами также раздражаются, встретив нелепые решения в реальной жизни (рис. 1.1)...

Свежее молоко можно купить в любом продуктовом магазине, особенно в большом городе. Информацию, размещенную на ва-

шем сайте, можно получить на сотнях подобных Web-страниц конкурентов. Когда захотите схитрить и заманить пользователя в непролазные джунгли плохо проработанного интерфейса, посмотрите на цифры в результатах поиска по запросу, по которому посетители могут попасть на ваш сайт. 100? 500? 1000? Они тоже туда смотрят.



*Рис. 1.1.* Это не фотомонтаж, но сиденья, расположенные на крыше машины, вы не встретите в повседневной жизни. В автомобилестроении никто не позволит себе такой вольности, однако в Web это почему-то считается возможным. Посмотрите, как этот «пользователь» ищет то, что по своей прихоти дизайнер решил расположить совершенно в ином месте

Когда вы купили в магазине вкусное свежее молоко один раз, потом еще один, в третий раз вы пойдете туда, даже если есть магазин поближе. Потому что вы не уверены, купите ли хороший товар в незнакомом месте, зато знаете точно, что в «вашем» магазине все всегда свежее. Привыкание к хорошему, удобному интерфейсу как привыкание к качественному товару посетителя магазина. Единожды польщенные высококачественным сайтом,

радушным отношением к пользователям и удобством работы, они будут вам преданны и благодарны.

Как передается информация о качественном товаре? Представьте себе двух соседа: «А я, Петровна, хожу только в магазин «Молоко», что на углу возле гастронома, там всегда все самое свежее, несколько раз в день привозят. И продавцы все такие приветливые, приятно посмотреть...» Независимо от того, какую продукцию вы поставляете — молоко или Web-сайты — весьма выгодно, когда о вас говорят как о поставщиках только качественной продукции. Это условно бесплатная реклама, повышение вашего «off-line» рейтинга, привлечение клиентов.

Когда вы просите продать вам пакет молока, продавец может недовольно швырнуть перед вами на прилавок первое попавшееся (которое, к примеру, хуже продается, чтобы от него поскорее избавиться), а может начать расспрашивать, какое молоко вы предпочитаете, какого производителя, с биодобавками или без, подороже или подешевле. В первом случае вы решите, что люди здесь работают, лишь бы заработать денег и отвязаться от вас, а во втором — что в вас заинтересованы и хотят сделать так, чтобы вы получили именно то, что хотели купить и остались довольны приобретением. Мнение заказчика, для которого вы разрабатываете интерфейс, или посетителя, который использует вашу систему, намного улучшится, когда он увидит, как серьезно вы подходите к проблеме качества и удобства использования системы. А вы, в свою очередь, сэкономите на сопровождении, т.к. не придется затрачивать огромные усилия на расплетение клубка плохих решений в вашей системе. Заказчик потом все равно потребует изменений, если вы не учли его предварительных требований и пожеланий. А если работа и консультации с потенциальными пользователями велись на ранних стадиях проекта, то недочетов будет намного меньше. Вы просто-напросто снизите затраты на разработку программ.

В конце концов, уже без ассоциаций с молочным магазином, если вы создаете систему для себя и своих сотрудников и клиентов, уделите внимание интерфейсу. Качественному интерфейсу пользователи быстро обучаются, а следовательно, уменьшаются затраты на обучение и поддержку, увеличивается отдача и скорость работы.



## 1.4. Человекоориентированный интерфейс

Под *человекоориентированным интерфейсом* я понимаю интерфейс, который учитывает особенности человеческой психологии, физические ограничения человека и его сознания во время работы. К факторам, о которых обязательно необходимо помнить во время разработки интерфейса, относятся и способность человека отвлекаться, и неточность его движений, и восприятие того, что он видит, и физическое напряжение во время работы. Все эти факторы будут рассмотрены далее в книге, иногда они будут указаны при рассмотрении того или иного решения, иногда будут только подразумеваться, так или иначе именно их учет и является основой комфортной и быстрой работы пользователей.

Под *человекоориентированным подходом* к построению интерфейса я понимаю учет как общечеловеческих факторов, например, средней скорости чтения, точности попадания в мелкие детали, восприятия цвета, так и факторов, критичных для систем, используемых какой-то определенной группой людей, например, людей с ослабленным зрением, нарушением восприятия цвета и т.д. Естественно, невозможно подстроить интерфейс под каждого пользователя индивидуально — кто-то хуже обучается, кто-то лучше, у кого-то усталость за компьютером наступает после часа работы, у кого-то через сутки. Однако всегда есть некоторые бесспорные общие требования, например, необходимость использования контрастного с фоном текста или пиктограмм, с размером большим, чем два на два пикселя.

В чем особенности человекоориентированного интерфейса? Предложу сравнение для более глубокого понимания. Если проанализировать содержание Web-страницы и найти нужную строку вы поручаете компьютеру, т.е. некоторой программе, то он имеет в своем распоряжении гораздо большие возможности, чем человек, глядящий на эту страницу. Нажимая сочетание клавиш Ctrl + F, человек поручает операцию поиска компьютеру, которому все равно, какого размера шрифт используется для отображения искомой строки, какие цвета на странице, даже если это белые буквы на белом фоне. Человек сам по себе такими возможностями не обладает, поэтому при разработке интерфейса для человека

важно учитывать особенности человеческого восприятия и совсем другую логику компьютера. Самым простым примером человекоориентированного подхода ко взаимодействию человека и компьютера может служить придуманная когда-то давно идея использовать «человекочитаемые» доменные имена сайтов вместо IP-адресов. Среднестатистическому человеку в силу особенностей памяти труднее запомнить сочетание 129.42.16.99, чем вполне понятное доменное имя `ibm.com`.

## Резюме

1. Интерфейс — часть некоторой системы, предназначенная для обеспечения взаимодействия с ней человека либо другой системы.
2. Интерфейс Web-сайта — это так называемые фронтальные компоненты системы, построенные при помощи языка разметки HTML и протокола передачи данных HTTP, которые позволяют пользователю выполнять необходимые задачи — получать информацию, общаться с другими людьми, развлекаться и т.д.
3. Качество интерфейса также существенно, как и качество других потребительских товаров и услуг.
4. Человекоориентированный интерфейс — это интерфейс, который учитывает особенности человеческой психологии, умственные и физические ограничения человека.

# Пользователь — компьютер

## 2.1. Структура взаимодействия «пользователь — компьютер»

Человек непоследователен, он отвлекается, не может продуктивно работать долгое время и воспринимает только адаптированную для него информацию. Человек не может в уме пересчитывать большие числа, не может помнить наизусть весь текст «Преступления и наказания» Достоевского, Библию или «Балладу о верткой пуле» Стивена Кинга. Человек ближе к обеду хочет есть, выходит на перекур, пьет чай и обсуждает футбольные матчи с коллегами. Он дышит, думает, чувствует, наблюдает, приспособливается...

А еще он считает, что компьютер должен читать его мысли, быстро и точно выполнять команды, и очень расстраивается, когда этого не происходит (рис. 2.1).

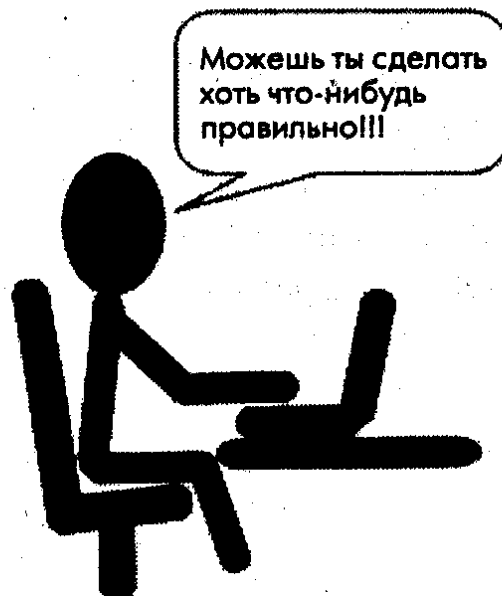


Рис. 2.1. К сожалению, не знаю автора этой карикатуры, которую я когда-то видел

Компьютер оперирует только двумя сущностями — ноль и единица. Он работает, пока его не выключат или пока не прекратится подача электричества. Он не может стать менее бдительным

через сутки непрерывной работы, не отвлекается и не устает. Он педантичен, послушен и мстителен. Он мыслит жестко и прямолинейно. Он ждет команды от человека и беспрекословно выполняет ее, даже если эта команда может навредить ему. Он редко бывает предупредительным и воспитанным.

Давайте рассмотрим, как в общем случае выглядит взаимодействие пользователя с компьютером и, в частности, с какой-либо программной системой.

### 1. Пользователь формирует цель.

Приходилось ли вам когда-нибудь слышать фразы сидящего за компьютером человека: «Почему этот компьютер не так работает! Я хочу, чтобы он просто посчитал мне сумму, а он выдает какие-то графики»? Я такое слышал не один раз. Система не удовлетворила потребности пользователя, и он недоволен. Система всегда должна удовлетворять его потребности. Это ее прямое назначение. Когда пользователь хочет найти телефон какой-нибудь компании в интернет-справочнике, это является его основной целью. Интерфейс должен быть направлен на обеспечение подобных нужд пользователя.


Когда пользователь видит цель, все остальное он теряет из вида, при этом он скорее станет тыкать курсором куда ни попадя, вместо того чтобы остановиться и подумать.

### 2. Пользователь решает, как достигнуть цели.

Выбор способа достижения цели зависит от предсказуемости системы, от знаний и опыта пользователя, от предвидения следующего шага, от степени участия системы в работе пользователя — подсказывает ли она ему следующий шаг и способ достижения цели или пассивно отвечает на запросы. Пока пользователь обдумывает способ совершения следующего шага, в системе ничего не происходит. Именно пользователь должен решить, что делать дальше и как достигнуть цели.

Первый постулат о принятии решений гласит: человек не любит принимать решения. Второй постулат: при работе с любой системой приходится принимать решения. Третий постулат: человек любит, когда ему преподносят готовое решение или информацию для него «на блюдечке с голубой каемочкой».

Часто вместо обдуманного решения пользователи предпочитают наугад щелкать мышью. Самое малое, что можно сделать во избежание подобных ситуаций, — это постоянно «говорить» пользователю о том, что он движется в верном направлении и указывать на возможные дальнейшие шаги.

 «Доктор, моя психика перегружена. Я скоро сойду с ума. Вы представляете, я работаю на апельсиновой фабрике расфасовщиком и отбираю на одну конвейерную линию апельсины покрупнее, а на другую — помельче. Доктор, я так устал — я весь день принимаю решения. Я так больше не могу...»

3. Пользователь ищет в системе элемент, подходящий для достижения своей цели.

Может использоваться не только один элемент — гиперссылка, кнопка, текст, но и несколько, например, тройка поле ввода текста — выбор из списка — кнопка.

4. Пользователь активизирует выбранный элемент.

Он щелкает по найденной ссылке (ставит «птицу» в чек-боксе, нажимает кнопку, прокручивает страницу — в зависимости от предыдущего пункта), чтобы перейти непосредственно к результату действия.

5. Пользователь получает от системы результат действия.

Получив результат, он в некоторой степени удовлетворяется тем, что ему вернули не сообщение об ошибке или проблеме на сервере, а именно результат его запроса к системе.

6. Пользователь анализирует результат.

Анализируя результат, он решает, подходит ли ему этот результат, достигнута ли его цель, придется ли все повторять или можно переходить к следующей цели.

## 2.2. Привычки

Интерфейс призван формировать у пользователя привычки. Привычки формируются независимо от того, думает ли об этом разработчик. Если одно и то же действие повторяется несколько

раз подряд, оно становится привычным. Привычки могут возникать на умственном и на физическом уровне, постепенно переводя сознательные действия в бессознательные.

Иногда я не могу вспомнить номер телефона, по которому раньше часто звонил. Ну не могу, и все. Тогда я просто подхожу к телефону и набираю этот номер, не задумываясь (или представляю, как набирал бы его). Я слежу за движением своих пальцев, потому что они «помнят» то, что мозг уже отложил «на долгосрочное хранение». Точно таким же образом, на основании привычек, я зачастую работаю на компьютере. Когда я ищу какое-то слово внутри документа при помощи файлового менеджера FAR, я использую функциональную клавишу F7. Каково же бывает мое удивление, когда, нажав на ту же привычную клавишу F7 внутри текстового редактора MS Word, я получаю совершенно не нужную мне в данный момент проверку правописания.

С другой стороны, способность формировать привычки, если ее использовать правильно, может приносить положительные результаты. Типичный пример тому — достаточно унифицированный интерфейс большинства приложений MS Windows. Будучи еще начинающим пользователем, я несказанно радовался тому, что, не затратив ни капли усилий, я был знаком с половиной интерфейса MS Excel только потому, что некоторое время успел поработать с MS Word. Однако не хочется кривить душой и говорить, что формирование привычек возможно только при условии давно известного и «выученного наизусть» интерфейса. Нет. Для того чтобы интерфейс формировал привычки, он должен соответствовать следующим требованиям:

1. Интерфейс должен быть достаточно простым. Заметьте, простым, а не примитивным. Примитивно устроены современные соковыжималки. У моей соковыжималки фирмы Brown всего одна ручка: поворачиваешь ее по часовой стрелке — прибор включается, против часовой — выключается. А интерфейс должен быть именно простым, состоящим из достаточно очевидных элементарных действий, так или иначе ведущих пользователя к цели. Например, в стиральной машине нельзя использовать метод «одного выключателя», как в соковыжималке, но в то же время многие домохозяйки утверждают, что научить обращаться со стиральной машиной можно даже обезьяну.



2. Интерфейс должен достаточно часто использоваться для формирования привычек, ведь иначе можно попросту разучиться работе с ним. Многократно повторяемые действия ведут к автоматизму и созданию привычки, а перерыв в совершении этих действий ведет к некоторой потере контроля над ситуацией. Недаром пользователи, пусть даже опытные, по вопросам настройки компьютера обращаются к системным администраторам. Если не заниматься подобной настройкой очень часто, то некоторые тонкости забываются.

Привычки высвобождают внимание. Человек с высвобожденным вниманием устает меньше и меньше напрягается. Однако, с другой стороны, иногда игнорируются опасные ситуации, ведь человек не обдумывает каждое действие, а выполняет его «на автопилоте». Типичным примером может служить уже ставший классическим диалог-предупреждение об удалении файла. Опытный пользователь ПРИВЫК, что операционная система ПОСТОЯННО что-то спрашивает. И практически постоянно ей требуется лишь подтверждение. «Вы действительно хотите удалить этот важный для операционной системы каталог?» — «Ну что ты еще хочешь? Да, хочу...» — «Каталог Windows успешно удален. Перезагрузите компьютер, чтобы изменения вступили в силу». — «Как? Я же хотел папку \TMP... Что же я наделал?!».

Однажды отец сказал мне, что его компьютер сошел с ума и «никак не хотел выключаться». Я пошел посмотреть на это чудо и, к моему стыду, с первого раза не увидел подвоха. Дело в том, что кто-то, работая на этом компьютере (на нем была установлена операционная система MS Windows 98), переключил радио-кнопку с Выключить компьютер на Перезагрузить компьютер. Привычка нажимать один и тот же набор кнопок, не задумываясь о том, что видит система, — вот самый настоящий ритуал. В этом случае после привычного набора действий цель не была достигнута. Здесь мало что можно сделать без ущерба для пользователя. Действие по переключению выключения компьютера на перезагрузку вполне легальное. Оно осуществляется, как правило, сознательным усилием пользователя. Предупреждение о том, что компьютер будет перезагружен, а не выключен, — это «антигуманно». Система и так всегда слишком много спрашивает. Самое большое, что можно сделать — либо восстанавливать первоначальное состояние.

чальное положение флажка выключения компьютера, либо посоветовать пользователю быть внимательней. Одно я знаю точно: лучше не отклоняться от тех способов работы, к которым пользователи успели привыкнуть, потому что внимательнее они никогда не станут.

Есть еще одно следствие формирования привычек — это так называемые предопределенные действия, т.е. я буду применять известный мне подход в случае новой, неизвестной задачи. Если привычные решения не действуют, только тогда я начну думать по-другому. В зависимости от того, как использовать предопределенные действия, это свойство может играть положительную и отрицательную роль. Приведу пример предопределенного действия. Программист, если его утюг перестал работать, первым делом попытается выключить и снова включить его (перезагрузить) просто потому, что с компьютерами этот прием часто срабатывает. Это предопределенное действие «если что-то не работает — перезагрузи это». Если разработчики утюга знают, что их продукцию будут использовать программисты, и предусмотрят функцию перезагрузки утюга, то они улучшат интерфейс утюга. Если они не учтут такие предопределенные действия, то тем самым сделают интерфейс утюга менее удобным.

### 2.3. Опыт против разнообразия

Студенту первого курса медицинского института гораздо проще определить у больного аппендицит, чем студенту пятого курса. Это с первого взгляда абсурдное заявление, тем не менее, очень верно. Старшекурсник знает, что кроме аппендицита подобными симптомами могут обладать, условно говоря, еще с десятков заболеваний, а первокурснику это неизвестно.

При том количестве программ, которые существуют в мире, и различиях в способах работы с этими программами, к великому сожалению, разработчики интерфейсов со временем заведут неопытных пользователей в тупик.

Неопытный пользователь скорее всего найдет единственный выход из старой игрушки в MS-DOS, в которой нет кнопки Quit: действуя по принципу «семь бед — один Reset», он перезагрузит

компьютер. Тут все вытекающие проблемы: несохраненный файл, во время работы над которым пользователь решил развлечься, потерянный ценный адрес интернет-ресурса, «подзатыльник» (Reset) операционной системе и т.д.

Опытному пользователю не придется прибегать к столь радикальным мерам, потому что ему известны магические заклинания и шаманские сочетания клавиш: сначала Esc, потом Esc три раза, потом Esc раз десять, потом Ctrl + C, потом Ctrl + Break, потом Ctrl + Alt + Del и т.д.

Знание различных способов решения одной проблемы и есть опыт. Если бы всегда и везде использовалось только одно сочетание, например, Ctrl + C, такой интерфейс опирался бы на привычный жест для пользователя, а не на его опыт.

К сожалению, в большинстве случаев невозможно выбрать, какой из всего разнообразия жестов наиболее привычный и, следовательно, правильный. Поэтому «думающему» разработчику интерфейса при выборе набора средств, предоставляемых пользователю, остается лишь полагаться на собственную интуицию и стараться хоть как-то ознакомить с приемами работы неопытных пользователей. Для этого необходимо проанализировать программные продукты, знакомые пользователю того уровня, который, как вы полагаете, будет самым активным. Например, пользователи вашего сайта относятся к тем, кто в основном работает в операционной системе Windows и редакторах MS Word и MS Excel. Следовательно, какое-то решение, схожее с привычными для них интерфейсами этих программ, будет ими воспринято очень легко. Иначе говоря, при выборе элементов интерфейса необходимо ориентироваться на опыт большинства пользователей, а не на свой опыт или, тем более, знания эксперта.

## 2.4. Ограничения физические и умственные

Существует огромный контингент людей, в том числе и интернет-пользователей, которые не подпадают под определение «молодой человек, опытный и внимательный, с уникальной памятью, совершенной реакцией, орлиным зрением и последними достижениями техники в аппаратном обеспечении компьютера».

Как уже говорилось ранее, человек — существо, весьма далекое от совершенства. Пользователи могут обладать плохой памятью (либо памятью, которую они не планировали напрягать для запоминания таких неважных вещей, как особенности интерфейса какой-то системы). Следовательно, необходимо стараться сформировать привычки либо использовать predetermined действия на основе уже существующих общих принципов взаимодействия с программной системой.

Пользователи могут быть холериками (беспокойными, вечно спешащими, нетерпеливыми людьми), а могут, наоборот, оказаться медлительными и нерасторопными флегматиками.

Пользователи могут обладать худшими физическими способностями (с трудом попадать в мелкие детали), ограничениями в восприятии цвета (видеть желтый шрифт на оранжевом фоне как сплошное серое поле), не очень хорошим зрением, которое с возрастом становится все хуже (таким людям нужны крупные шрифты, контрастные цветовые схемы, крупные детали). Если это не учитывать во время разработки, то такой интерфейс нельзя назвать человекоориентированным.

У любого человека есть предельная черта сложности, определяющая максимальную сложность системы, которую он способен воспринять. После того как система пересекает эту черту, человек может продолжать работать, но он либо остается на своем уровне, либо медленно и постепенно отодвигает черту. Своеобразная защитная реакция при работе с большими сложными системами — ограничение используемых функций системы и выполнение «ритуалов». Такие действия выполняются не осознанно, а как магическое движение, значение которого для пользователя загадка, но кто-то сказал, что так надо делать.

Человек не может одновременно обрабатывать несколько задач — внимание может быть сосредоточено только на чем-то одном (наверное, самое правильное слово «задача»). Сознание — не многозадачно.

☑ Есть небольшое уточнение. Некоторые задачи могут выполняться одновременно при условии, что одна из них доминирует, а остальные проходят в «фоновом режиме», почти бессознательно. Печатая этот раздел, я постоянно думал о телефонном звонке, который сделал прежде, чем сел за компьютер. И постоянно слышал звук телеви-

зора за спиной. Но я не смог бы одновременно «думать» о слабостях человеческой сути, выгодах вкладов в национальной валюте, падении евро или сложности текстов песен Аланис Мориссетт, которая поет у меня в наушниках.

Человек постоянно отвлекается. Если бы можно было сказать «постояннейше», я бы так и сказал. Десять тысяч вещей отвлекают его от работы. Он думает о пище, включенном уюте и коллеге, который закурил у него в кабинете. Он встает открыть окно, чтобы проветрить помещение, выходит перекусить, играет в пасьянс, если босс отвернулся и т.д.

А если все-таки он работает, то ведет себя часто очень нетерпеливо. Если не предупредить его, что система будет несколько секунд принимать введенные данные, пользователь может нажать на кнопку Подтвердить раза три. Против такого недуга в интерфейсе есть несколько средств: корректно обрабатывать двойные нажатия на кнопки нетерпеливыми пользователями или предупреждать их о том, что нажимать на кнопку два раза нельзя.

Следующий недостаток человечества — инерционность. Передвигая курсор, человек не может остановить руку точно там, где это ему надо, он не может не нажать на клавишу Ввод после того, как несколько раз подряд быстро нажимал на нее в предыдущих формах. Инерционность присуща не только движениям, но и сознанию. Типичным примером инерционности сознания может служить наверняка известная вам детская «подколка», когда человека спрашивают «Какого цвета помидор?» — «Красного». — «Какого цвета кровь?» — «Красного». — «Какого цвета пожарная машина?» — «Красного». — «На какой цвет надо переходить улицу?» — «НА КРАСНЫЙ...»

В отношении интерфейса какой-либо программы можно рассмотреть пример с расположением кнопок. Допустим, в одной операционной системе сначала идет кнопка ОК и после Cancel (наиболее частый вариант), а в другой операционной системе все наоборот. Пользователь, привыкший к одному порядку, будет много ошибаться, нажимая по инерции на кнопку, находящуюся «не на своем месте».

Как видите, врагов у хорошего интерфейса очень много. Это постоянная битва, и побеждает в ней сильнейший. Но сильнейший не из пары человек — компьютер, а из пары ваши конкуренты и вы.

## 2.5. Фокус внимания

**Фокус внимания** человека применительно к компьютерным системам — это место на экране, куда направлен его взгляд и где он сознательно сосредоточен. Цель качественного, хорошо проработанного интерфейса, с одной стороны, удерживать фокус внимания пользователя именно в том месте, где он должен совершить следующее действие, а с другой — не заставлять его перед совершением действия долго размышлять над необходимостью или последствиями. На фокус внимания влияет большое количество факторов, поэтому управлять им достаточно сложно.

Общий принцип управления и удержания фокуса внимания примерно таков: необходимо привлечь внимание человека к какому-либо элементу, при помощи этого элемента заставить его переместить фокус внимания к следующему элементу, при помощи следующего — к другому и так далее, пока цель пользователя или системы не будет достигнута. Таким образом, если вы не «позволяете» человеку отвлекаться и он все время сосредоточен на своей задаче, это выгодно и ему, и вам. Внимание — одно из основных понятий психологии, которое определяет успех обучения, продуктивность работы и т.д.

Человеку необходимо время, для того чтобы переключиться на новую задачу или вернуться к старой. Редко происходит так, что пользователь сразу продуктивно приступает к продолжению работы, если он, например, выходил на перекур или говорил по телефону. Обычно между «на чем я остановился» и «ах да, я писал о...» проходит порядка семи-восьми секунд. Для того чтобы уменьшить этот период и облегчить посетителю возврат к выполняемой задаче, необходимо, чтобы система находилась в том «положении», в котором ее оставили, и сообщала о действии, которое намеревался совершить пользователь.

Как я уже говорил, человек обладает просто фантастической способностью отвлекаться. Это почти не зависит от важности и срочности того, что он делает. Если «по пути» пользователь натывается на что-либо интересное, он готов бросить все.



☑ Весьма показательным примером того, насколько способны отвлекаться пользователи во время работы, служит один случай. Когда я просматривал статистику посещений своего персонального сайта, любезно предоставляемую сервисом [spylog.ru](http://spylog.ru), я наткнулся на такой факт. Некто искал на поисковом портале Rambler информацию о мужских деловых костюмах, затем он наткнулся на ссылку на мой сайт и ушел на него. Судя по отчетам, этот некто провел за чтением не так уж и мало времени, а ведь искал он совсем другое.

Повторюсь, фокус внимания бывает только один, человек осознанно может думать только об одной задаче, а все остальные в его голове протекают бессознательно. Если бы это было не так, вы никогда в жизни не слышали бы фраз «подожди, а то я мысль потеряю», «сейчас, секундочку». Человек бы просто дописывал фразу или не терял мысль, если бы мог иметь несколько фокусов внимания. Кто со мной не согласен, пусть прочитает следующее примечание, декламируя вслух любое из стихотворений (хоть бы и «Наша Таня громко плачет»), а потом попробует сказать, о чем он читал.

☑ Я никогда не мог одновременно различать буквы, передаваемые при помощи азбуки Морзе, и составлять их в слова. Если я слышал более одного слова, то мог понять смысл сообщения только после того, как записал все по буквам. Для выполнения подобной задачи необходимо бессознательно воспринимать отдельные буквы. В таком случае я могу складывать их в слова, не задумываясь над каждой буквой по отдельности.

Способность терять фокус внимания так же, как крепость сна, различная у разных пользователей. Один человек легко отвлекается, стоит ему только услышать или увидеть около себя что-то постороннее (звук за окном, заходящую на посадку муху, услышать жужжание диска внутри CD-ROM). Другие, наоборот, настолько глубоко погружаются в задачу, что «вырвать» их оттуда почти невозможно. Одна моя знакомая, работая за компьютером, «уходила в себя» настолько, что начать разговаривать с ней можно было, лишь громко позвав ее или тронув за плечо. После этого она еще секунд пять смотрела на меня, пытаясь понять, «кто она и что здесь делает».

- ☑ Существует один достаточно старый анекдот про программиста, написавшего программу для себя лично, которая перед выключением компьютера в конце рабочего дня сообщала: «Тебя зовут Вася. Ты программист, тебе 28 лет. Сегодня 24 мая 2004 года, понедельник. Ты сейчас идешь домой. По пути домой ты должен купить молоко».

Для восстановления фокуса внимания пользователю необходимо сообщить достаточно большое количество информации: где он находится, с чем работает, что уже сделал, на каком шаге остановился и какой элемент текущий с точки зрения системы.

- ☑ Существует хороший прием определения фокуса внимания пользователя: в процессе работы за компьютером надо приостановить его и попросить описать форму, которую он видел минуту назад. Он назовет те поля, на которых останавливался фокус внимания и которые попадали в кратковременную память. Если важный элемент формы оказался «невидимым» для пользователя, подумайте над его расположением в форме или способом выделения.

Подсказки, контекстная помощь, глагольное название кнопок, наименование форм и заголовки разделов — все это сообщит пользователю, что должно или может быть выполнено в данный момент.

Существуют четыре способа заострить внимание на текущем объекте в системе:

- *подсветка*

Система подсвечивает объект, который для нее текущий. Например, в браузере Internet Explorer последних версий панель инструментов представляет собой серые пиктограммы, которые становятся цветными при наведении на них курсора. В Web частным случаем подсветки является изменение цвета или вида гиперссылки при наведении на нее мыши.

- *указание*

Можно указать какую-либо область экрана или объект, если это поддерживается программой. Например, может быть указана конкретная строка таблицы после «клика» на ней мышью. Указание — это вариант «долговременной» подсветки.

- *выделение*

Выделение подразумевает, что пользователь каким-либо образом обозначает для системы объект, над которым нужно про-

известить некоторое действие. Выделить объект можно, установив флажок в чек-боксе или нажав радио-кнопку. Можно выделить часть текста, изображение или какую-либо еще часть содержимого страницы. Например, при выборе варианта дизайна бесплатной гостевой книги на сайте [narod.ru](http://narod.ru) из девяти представленных пользователь выделяет нужный ему при помощи радио-кнопки и нажимает кнопку для перехода к следующему действию.

- **активация**

Активировать — значит сделать некоторый элемент интерфейса или объект системы доступным для использования или преобразования. Например, выпадающее меню в некоторых случаях может быть использовано только после активации управляющего элемента, т.е. наведения на него курсора. Текущий раздел меню, как правило, идентифицируется при помощи подсветки.

## 2.6. Кратковременная память

Вся информация, воспринимаемая пользователем при работе с системой, хранится в кратковременной памяти. Кратковременная память имеет весьма ограниченный объем и является по сути не отдельным свойством человеческого мозга, а некоторой составляющей долговременной памяти.

Представьте себе коробку без боковых стенок, в которой один за другим лежат кирпичи. Если с одной стороны попытаться засунуть в эту коробку еще один кирпич, то он туда, безусловно, влезет, но при этом с противоположной стороны вывалится другой, оказавшийся лишним. Таким образом, в коробке всегда может находиться ограниченное число кирпичей.

Так и человеческая память всегда сохраняет текущую информацию, важную в данный момент. Как только потребность в ней исчезает, на ее место приходит новая информация, которая «вытесняет» лишнее из области памяти, ответственной за обработку текущего момента. Названия полей формы поиска постепенно стираются из памяти посетителя, когда он переходит к чтению только что найденной статьи.

Для того чтобы пользователь запомнил, как работать с системой, ему необходимо либо постоянно «загружать» в кратковременную память одинаковые фрагменты (повторять), либо сознательно запоминать возможные способы работы с системой (обучаться), либо получать постоянные консультации по ходу работы (подсказки и помощь). Так постепенно получаемая информация переходит из кратковременной памяти в долговременную.

Быстрой обработке информации способствует такой интерфейс, который не требует серьезного обучения, обеспечивает поддержку пользователя, понятность и предсказуемость всей системы.

Существует еще один фактор, на который мы не в силах повлиять, — это особенности памяти конкретного человека, т.е. хорошая у него память или плохая, какая память у него более развита (зрительная, слуховая, осязательная), какой объем его кратковременной памяти и т.д.

## 2.7. Монотонность

**Монотонность** — это предоставление одного способа достижения одной цели. Предрасположенность человека к такому свойству интерфейса, как монотонность, в некоторой степени обусловлена человеческой натурой. Интерфейс, которому присуща монотонность, упрощает и ускоряет работу. Посудите сами, когда новичку объясняют, что такое удаление и как его выполнить, то ему куда легче понять, что удалить любой объект можно нажатием на клавишу Delete (кстати, это не совсем так).

Но вы окончательно его запутаете и заставите сомневаться всякий раз, а иногда и отказываться от удаления, если скажете: «Удалять можно клавишей Delete, тогда, если это файл, он переместится в Корзину, а если это текст или объект в графическом редакторе, то он просто исчезнет. Такое удаление можно будет отменить. Чтобы удалять «мимо» Корзины, надо нажать сочетание клавиш Shift + Delete. Можно еще пользоваться клавишей F8, если вы находитесь в файловом менеджере вроде Far или Norton Commander. А есть еще такое понятие, как «вырезать», это можно сделать...»


Подобный «букет» методов — результат многолетних стараний огромного количества разработчиков программных систем. С каж-

дым новым способом удаления интерфейс усложняется, особенно для новичков или тех, кто редко использует такую программную систему. Опытный пользователь непроизвольно делает интерфейс любой программы монотонным через предпочтение каких-либо основных способов из всего предлагаемого спектра.

Монотонный интерфейс ускоряет работу, потому что нет необходимости принимать решение о дальнейшем шаге. Кроме того, при монотонном интерфейсе меньше сбивается фокус внимания.

## 2.8. Точность данных

Человек склонен округлять. Это свойство позволяет ускорить работу и более наглядно отобразить информацию. Если я спрошу у вас, чему равно число  $\pi$ , большая часть мне ответит 3,14. Рыбак редко говорит о том, что поймал рыбу длиной 40 сантиметров и весом три килограмма семьсот пятьдесят граммов. Он разведет руки в сторону и скажет: вот такого леща поймал, килограммов на пять. Кроме того, высокая точность для «анализа ситуации» нужна редко.

 Говоря о точности данных, я понимаю разумные пределы округления и ситуации, когда это необходимо делать. Лично я не хотел бы увидеть в результатах поиска: «Примерно пятьдесят сайтов из где-то трех-четырёх сотен».

Существуют два способа представления информации с приблизительной точностью — *округление* и *визуализация*. В первом случае необходимо знать степень округления результата, которая устроит пользователя, во втором — способ и размер визуализации. К примеру, занятый объем жесткого диска можно изобразить в виде диаграммы или полосы, часть которой закрашена другим цветом (рис. 2.2).

Там, где точность данных не важна с математической или статистической точки зрения, желательно выдавать пользователю округленные данные, т.к. их легче воспринимать. Визуализированная информация воспринимается на порядок быстрее цифровой, пусть даже и округленной.

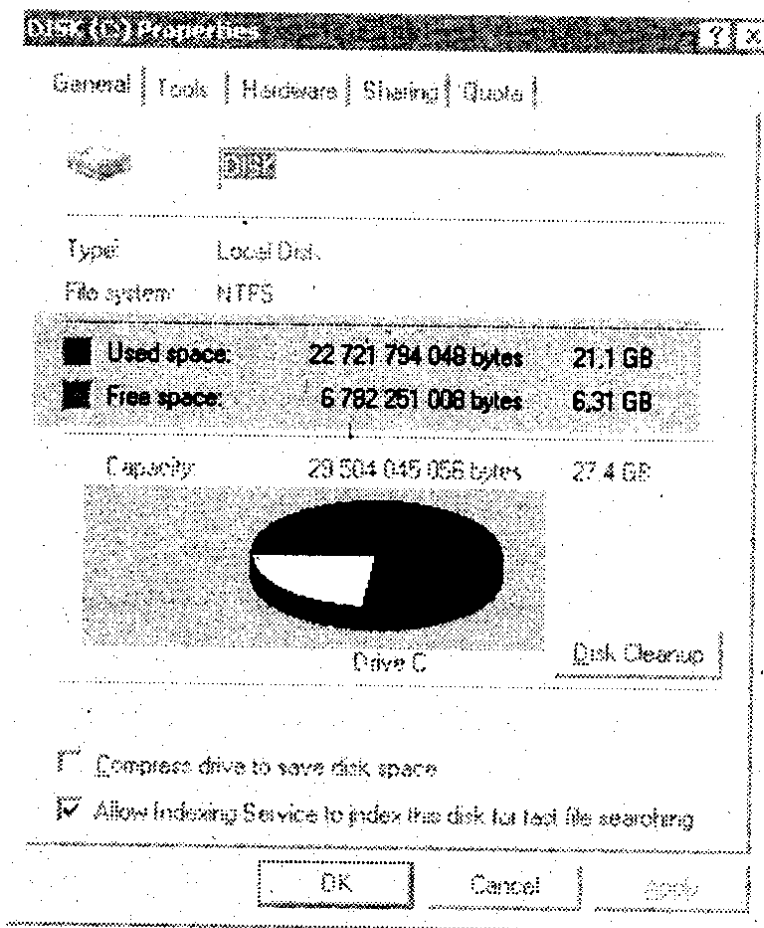


Рис. 2.2. Два способа показать пользователю, сколько свободно места на диске. В верхней части точная информация, в нижней — наглядная

## 2.9. Время протекания процессов

Способ визуализации может быть применен не только для наглядного отображения информации, но и для отвлечения пользователя от долговременного ожидания или для отражения состояния системы. Например, инсталляционные программы часто снабжают пояснениями, анонсами, пресс-релизами и рекламой устанавливаемого продукта. За чтением всего этого время протекает быстрее.

То, что система не «зависла», а просто «задумалась», может показывать индикатор протекания процесса. Когда пользователь вместо пустой страницы видит часы со вращающейся стрелкой и надпись Подождите, он становится более терпеливым, потому что система что-то делает, его команды системе не ушли «в никуда», а адекватно восприняты.



Таким образом, желательно, чтобы интерфейс отвлекал человека во время длительных пауз в работе системы, хотя в базе данных в это время все может просто «кипеть».

## 2.10. Что такое идеальный интерфейс

Чем больше интерфейс отличается от того, к чему пользователь привык в реальной жизни, тем труднее ему работать с таким интерфейсом. Поэтому мне кажется целесообразным рассмотреть понятие идеального интерфейса и, по меньшей мере, к нему стремиться (рис. 2.3). *Идеальный интерфейс* — такой интерфейс, который максимально приближен к тем условиям и к той среде, в которой человек привык жить и работать.

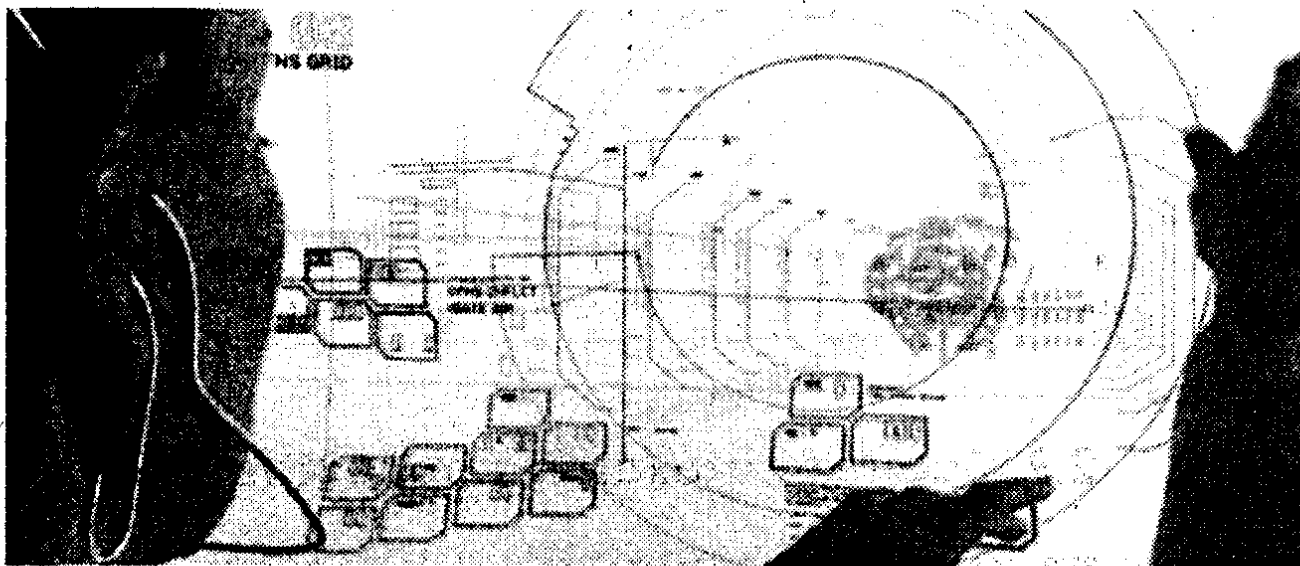


Рис. 2.3. Пример интерфейса, достаточно близкого к тому, что я понимаю под идеальным интерфейсом. Этот кадр из фильма «Матрица-2». Оператор управляет «парковкой» космического корабля при помощи 3-d проекции реальных элементов

Первый пример стремления к идеальному интерфейсу.

Представьте себе опытного водителя с двадцатилетним стажем, который сел за компьютер и впервые включил игру «Need For Speed». Не думаю, что он пройдет трассу быстрее и качественнее, чем его сын-подросток, даже не имеющий водительских прав, но гоняющий на виртуальных трассах вот уже несколько месяцев. Интерфейс компьютерной игры слишком отдален от

того, что этому человеку приходится делать в реальной жизни, и никакая метафора ему не поможет. Однако современные технологии позволяют использовать вместо клавиатуры и мыши приставку, например, руль с педалями. Даже если наш водитель не сразу приспособится к виртуальным приборам управления машиной, он будет чувствовать себя комфортнее и увереннее и уже скоро будет обгонять сына-виртуоза.

**Второй пример стремления к идеальному интерфейсу.**

Представьте себе дизайнера, рисующего на компьютере нечто в технике карандашного рисунка. Мышь — инструмент непослушный и крайне неточный для такой работы. Движения дизайнера скованны и неуклюжи, он не чувствует линий. Но вот приходит его друг — дизайнер номер два. Он видит мучения своего коллеги и предлагает испытать свое «световое перо» — такой электронный карандаш, движения которого по специальному планшету отображаются на экране компьютера. Это достижение техники впечатляет первого дизайнера, он несколько недоверчиво пробует его в деле и очень быстро заканчивает рисунок. Не прошло и пятнадцати минут против тех нескольких часов, которые ушли на мучения с мышью.

**Третий пример стремления к идеальному интерфейсу.**

Диспетчеру отдела водоснабжения сообщили из аварийного наряда о том, что на улице Свободы прорвало трубу и теперь проспект Независимости остался без воды. Он знает, что рядом есть второй источник водоснабжения, так что, если перекрыть прорвавшуюся трубу и включить подачу воды по соседней линии, положение можно спасти.



Я прошу специалистов по водоснабжению, если таковые будут читать эту книгу, простить мне этот пример, потому что он наверняка не соответствует реальной жизни. Он абстрактен, и я мог бы с таким же успехом говорить о газоснабжении или подаче тепла.

Диспетчер долго пытается отыскать это место на электронной карте в компьютере, орудуя увеличением и уменьшением масштаба, наконец, находит его и отдает команду наряду переключить воду на соседнюю линию. Но вот руководитель наряда снова звонит уже в панике, говоря, что теперь без воды еще

и площадь Победителей. Диспетчер возвращается к компьютеру, смотрит на экран и понимает, что из-за слабой разрешающей способности монитора он спутал Центральную и Восточную линии. Он снова отдает команду наряду и через пять минут узнает о новых проблемах.

А теперь представьте того же диспетчера у большой электронной карты-макета на стене, которая управляется специальными сенсорными датчиками. На карте мигает красный сигнал о неполадках на улице Свободы и желтый об отсутствии воды на проспекте Независимости. Диспетчер манипуляциями руки последовательно увеличивает карту до нужного размера, а затем прямо на этой карте нажимает пальцем на изображение прорвавшейся трубы. Красная лампочка гаснет. Затем он также легко и быстро переключает подачу воды на другую линию и видит, что погасла и желтая лампочка. После этого он отдает четкие указания наряду, все неполадки устраняются.

Наверное, только ленивые разработчики программного обеспечения не стремятся к идеальному интерфейсу. Нужно всегда искать возможность хоть немного приблизить интерфейс системы к идеальному. Даже малюсенький шаг в этом направлении приводит к удовлетворению пользователей, повышению продуктивности и скорости их работы.

✓ В стремлении к идеальному интерфейсу разработчики аппаратного обеспечения всегда идут на шаг впереди разработчиков программного обеспечения. Например, недавно я видел ноутбук, который можно использовать как лист бумаги. Монитор в нем поворачивается, закрывая клавиатуру, при этом сам ноутбук становится почти плоским. Запустив специальную программу, на мониторе при помощи светового пера, похожего на обыкновенную шариковую ручку, можно писать, как в блокноте. Записи тут же сохраняются в электронном виде, как если бы это был отсканированный лист из обычного блокнота.

## Резюме

1. Работа человека с какой-либо программной системой происходит таким образом: пользователь формирует цель; решает, как ее достигнуть; ищет в системе элемент, подходящий для

- достижения данной цели; активизирует этот элемент; получает от системы результат действия; анализирует результат.
2. Интерфейс должен быть достаточно простым, но не примитивным.
  3. Многократно повторяемые действия ведут к автоматизму и формированию привычки. Привычки высвобождают внимание. Последствия привычек — так называемые предопределенные действия. Свойство интерфейса формировать привычки может приносить весьма положительные результаты и упрощать работу.
  4. Основные проблемы человека при работе с программной системой: отвлекаемость, однозадачность, нетерпеливость, инерционность, ограниченные физические возможности.
  5. Фокус внимания человека применительно к компьютерным системам — некоторое место на экране, куда направлен его взгляд и где он сознательно сосредоточен. Фокус внимания может быть только один.
  6. Существуют четыре способа идентификации текущего объекта в системе: подсветка, указание, выделение, активация.
  7. Монотонность — предоставление только одного способа достижения одной цели. Интерфейс, которому присуща монотонность, упрощает и ускоряет работу.
  8. Человек склонен округлять. Существуют два способа передачи неточной информации — округление и визуализация. Способ визуализации может быть применен для наглядного отображения информации, а также для отвлечения пользователя от ожидания длительного процесса или для отражения состояния системы.
  9. Идеальный интерфейс максимально приближен к тем условиям и к той среде, в которой человек привык жить и работать.

# Интерфейс в программном обеспечении

«Вы либо ноль, либо единица... Любой мальчишка из гаража может лишить нас работы...». Это фраза из фильма «Антимонополия» (Antitrust) компании Metro — Goldwen — Mayer. В этом фильме активно проводится идея о том, что любой гениальный программист, который изобрел новый способ связи, протокол или гениальное решение для электронной коммерции e-Business, способен «перебежать дорогу» крупной компании по производству программного обеспечения. Идея фильма ошибочна уже в том, что «мальчишка из гаража» всего лишь программист. А за программы без документации, внедрения и сопровождения больших денег никто никогда не платил. Платят за технологию, за сопровождение, за полноценную интегрируемую и масштабируемую систему, но не за циклы и переменные. Программист «без роду, без племени» очень редко способен создать полноценный программный продукт, даже если потратит на него несколько лет. Человек чаще всего может быть лишь кем-то одним: или программистом, или проектировщиком, или дизайнером, или аналитиком. Чтобы сочетать в себе все эти способности, нужно быть гением.

### 3.1. Виды программного обеспечения

Фредерик Брукс — именитый участник крупных западных проектов по созданию сложных программных систем — выделяет несколько разновидностей программного обеспечения.

**Программа.** Самый простой вариант программного обеспечения. Чаще всего это просто код, написанный на каком-либо языке программирования и состоящий только из операторов, циклов, рекурсий, переменных и т.п. Иногда могут быть комментарии в тексте программы или даже простенький файл с описанием работы и настройки программы. Переменные допустимо называть «МояПеременная» или «loop\_for\_get\_a\_number\_of\_user\_input». Программа может работать под управлением операционной

системы и ее версии, которая установлена на компьютере программиста, и совершенно невменяемо вести себя на другом компьютере. Такая программа без исходных файлов — это только исполняемый модуль типа `exe`, `class`. Ключи для запуска программы и ее опции знает только разработчик. Если такая программа имеет визуальное представление, то, как правило, у пользователя оно ничего, кроме раздражения, не вызывает.

**Программный продукт.** Иметь дело с программным продуктом гораздо приятнее. Разработчик пишет тексты программ таким образом, чтобы разобраться в их работе мог не только он один, но и другой программист, знающий соответствующий язык программирования. Программа содержит четкие подробные комментарии, все версии ее исходного текста содержатся в специальной системе управления версиями. Оформлена подробная документация, в том числе и руководство пользователя, программиста. Программа тщательно протестирована, имеет набор контрольных примеров и может быть инсталлирована в различных операционных системах.

**Программный комплекс.** Представляет собой определенный набор интегрируемых программных продуктов, собранных в достаточно сложную систему с согласованными программными интерфейсами и обменивающимися взаимно понятными данными.

**Системный программный продукт.** Это совокупность различных программных продуктов и программных комплексов, со всеми их артефактами<sup>1</sup> и отличиями от обычной программы.

Программное обеспечение создается в первую очередь для людей, а в этом случае «первой скрипкой» выступает интерфейс. Значит, и с этой точки зрения различия между программой и системным программным продуктом очень велики. Поскольку в большинстве случаев крупные системы создаются не для пары-тройки человек, а большого количества людей, то особое внимание должно уделяться отношению пользователей к системе. Чем больше положительных эмоций вызывает работа с системой, тем выше ее шансы стать востребованной. Кроме различий в документировании, сложности, качестве исходных кодов и пе-

<sup>1</sup> Артефакт — результат какой-либо деятельности. Артефактом может быть программа, исходный текст, документ, даже CD-ROM с записанной информацией.



реносимости, программа и системный программный продукт отличаются отношением системы к пользователям. Ведь если интернет-сайтом пользуются сотни людей, то именно они диктуют свои требования, от них зависит прибыльность этого проекта, популярность. Для пользователей самым важным является то, насколько им легко и удобно работать с программой, и если такая система игнорирует предпочтения, желания и слабости пользователей, то она обречена на провал.

Однако невозможно с ходу узнать, что же «нужно» пользователям. Иногда для этого проводят широкие маркетинговые исследования, специальные тесты, изучают привычки и предпочтения пользователей. Тратятся большие средства не на создание нового кода и программных продуктов, а на изменение вроде бы только внешнего вида существующих программ с добавлением или устранением каких-нибудь функций. А результат? Результат в том, что предпочтение отдается именно наиболее удобным программам, наиболее приспособленным для работы, ненапрягающим. И от того, сколько внимания уделяется будущим пользователям и графическому интерфейсу для них, насколько система удовлетворяет их информационные потребности, зависит ее рентабельность. Тщательно спроектированный и проработанный интерфейс позволяет работать без посторонней помощи, без системы справки, без подсказок службы поддержки.

### **3.2. Независимость интерфейса от других компонентов приложения**

Если создание серьезного программного продукта требует многих усилий и ресурсов, то логично было бы предположить, что все части такой сложной системы очень тесно связаны между собой. Зависит ли интерфейс пользователя от способа построения и организации других компонентов Web-приложений? На самом деле, практически нет. Это важно потому, что именно эта особенность помогает разработчикам и специалистам по интерфейсам применять свои знания и анализировать интерфейс программ независимо от того, на каком языке они написаны и в каком окружении работают. Чтобы убедиться в этом, необходимо немного теории.

Большая часть профессионально создаваемых программных продуктов, предназначенных для работы непосредственно с пользователями, построена на базе модели MVC. *Модель MVC* (model-view-controller) — способ построения приложения с помощью трех основных компонентов: 1) *модель* (model) — уровень хранения данных (она же объектная модель, база данных); 2) *представление* (view) — уровень внешнего вида данных, отображаемых пользователю; 3) *контроллер* (controller) — уровень компонентов, реализующих логику приложения.

Система, например интернет-магазин, в соответствии с моделью MVC должна быть построена так:

1. В базе данных хранится вся информация о пользователях магазина, заказах, товарах, статусах и т.п. Эта информация пассивна, если ее отделить от остальных частей системы, то в принципе с ней можно делать любые возможные операции. На этом уровне известно, какими данными обладает система.
2. На уровне представления, т.е. на фронтальном уровне, система выдает пользователю результат работы контроллера и собирает данные для следующих действий. Она показывает результаты поиска, пользователь вводит свои данные, выбирает виды доставки, оплаты, кладет товар в Корзину, чтобы потом вернуться к нему. На этом уровне известно, как выглядят для пользователя результаты работы системы. Именно к представлению и относится интерфейс.
3. На уровне контроллера, т.е. на логическом уровне, описаны законы, по которым работает этот интернет-магазин. Здесь определяется, кто может делать заказ, какова последовательность заказа, как реализуются права доступа конкретного пользователя, как искать товар в базе данных, на основании каких статусов товара выдавать информацию о нем пользователю и т.п. На этом уровне известно, как система обрабатывает существующие и получаемые извне данные.

Данные уровни, как правило, разделены по своему расположению. Существует компьютер пользователя и сервер, который обрабатывает всю информацию. Сервер на основании набора данных и инструкций возвращает HTML-страницу (в большинстве случаев эта страница уже не зависит от языка программирова-

ния, на котором написаны серверные компоненты Web-сайта). Пользователь совершает какое-нибудь действие на странице, и та снова обращается к серверу. Что бы система ни вытворяла на сервере, пользователю возвращается очередная сгенерированная HTML-страница и так далее. Таким образом, результат для пользователя будет одинаковым и при использовании скриптового языка ASP в сочетании с СУБД MS SQL Server, и при использовании языка PHP с MySQL.

Однако далеко не все системы, построенные для работы в Web, организованы по модели MVC. В некоторых случаях разделение на уровни может быть условным. Например, существует некоторая база данных, хранящая объекты и определяющая объектную модель, набор программ-скриптов, часть которых отвечает за представление, т.е. генерирует HTML-код, а другая — за бизнес-логику, т.е. определяет правила работы системы с пользователем.

Частным случаем являются статические HTML-страницы, не изменяющиеся во время работы пользователей. На таком принципе построено огромное количество персональных сайтов на бесплатных хостинг-серверах. На статической HTML-странице находятся и объекты, и бизнес-логика, и представление.

Таким образом, независимо от способа организации сайта в конечном итоге пользователи работают с одним и тем же: визуальные компоненты HTML-страниц, браузер, операционная система. Это весьма важное и положительное обстоятельство. Благодаря ему «развязаны руки» и у разработчика интерфейса, и у программиста, они достаточно независимы друг от друга.

### **3.3. Проектирование программного обеспечения и интерфейса**

Предположим, что вы решили создать некоторый «Электронный телефонный справочник». Сначала о его интерфейсе ничего не известно, есть только самые общие требования. Для того чтобы справочник был действительно полезен и удобен, необходимо точно знать, какого рода справочники предпочитают пользователи, на каком принципе он будет построен, сколько и каких пользователей будут к нему обращаться, кто будет поддерживать его в актуальном состоянии и т.п.

В начале создания программного продукта необходимо обдумать и сформулировать требования к графическому интерфейсу.

Программный продукт должен предназначаться некоторому гипотетическому пользователю. Например, вспомните кота или скрепку — «помощников» в последних версиях MS Word. Можно предположить, что компания Microsoft при разработке интерфейса этой программы исходила из того, что работать с ней будут малоопытные, начинающие пользователи.

Интерфейс, в первую очередь, должен работать на пользователя и позволять делать именно то, что требуется от программы. Потребность пользователя должна быть удовлетворена при помощи тех средств, которые ему предоставляются. Если это программа для сканирования, то она должна позволять максимально просто и корректно осуществлять этот процесс.

Поскольку пользователи Web обладают разными компьютерами, разрешениями монитора, мышами, принтерами, то интерфейс должен обеспечивать удобное и правильное взаимодействие со всем этим оборудованием. То же самое касается и программного обеспечения, в частности браузеров.

Большинство выпускаемых товаров обладают настолько большим набором функций, что иногда становится весьма сложно их использовать. Взять, например, мобильные телефоны. Представьте себе, что вы купили телефон самой последней модели, с цветным дисплеем, тысячей функций, встроенным фотоаппаратом и видеокамерой, соединенный с интернетом и т.д. (рис. 3.1).

Что можно безболезненно удалить из такого телефона, чтобы он не лишился своего прямого предназначения? Все, что угодно, только не звонок, не микрофон, не динамик и не возможность связаться с другим абонентом, набрав его номер. Прямое предназначение телефона — быть телефоном, а уже потом фотоаппаратом, записной книжкой, будильником и всем остальным. Если вы создаете текстовый редактор, вы можете сколь угодно долго отыскивать иконки покрасивее, но вы никак не можете убрать из интерфейса возможность сохранять файлы.

Создаваемая система должна быть четкой, структурированной, понятной и предсказуемой. Даже если мы снабдим «Электронный телефонный справочник» сопутствующими функциями, мы

обязаны гарантировать, что он по-прежнему остается программной системой, которая, используя в качестве основного объекта «абонент», позволяет создавать, искать, редактировать и удалять информацию о нем. Без этих функций она теряет смысл. Кроме того, описанный набор действий должен быть первостепенным для системы, потому что пользователь пришел с конкретной целью, и система обязана обеспечить выполнение этой цели.



Рис. 3.1. Мобильный телефон Sony Ericsson S700c.

Технологии шагают вперед

Та часть требований, которая есть «телефон» в телефоне, которую никаким образом нельзя исключить, без которой система теряет свой смысл, обеспечивает концептуальную целостность системы. Концепция системы неразрывно связана с предметной областью<sup>1</sup>. А интерфейс системы служит, в свою очередь, для поддержания предметной области приложения в полном объеме.

Концептуальная целостность системы будет нарушена, если мы поддержим в «Электронном телефонном справочнике» возможность сканировать и сохранять документы. Может, вам и удастся встроить такую функцию, но это нарушит понимание самой системы и ее интерфейса пользователем. Поддержание подобных

<sup>1</sup> Предметная область — область функций и задач, покрываемых разрабатываемой системой.

функций если и выгодно для кого-нибудь, то не для системы и не для вас. Примером грубейшего нарушения концептуальной целостности может быть идея вмонтировать тостер в телевизор. Это может быть удобно для тех, кто любит поесть за хорошим фильмом румяные хлебцы, но телевизор и все его составляющие должны улучшать изображение и звук, а не качество еды.

Из того, что имеет отношение к интерфейсу, на стадии проектирования известна часть предметной области, базовые технологии, назначение системы, потенциальные пользователи. Однако для начала работы достаточно уже и этих сведений. Необходимо исследовать аппаратное и программное обеспечение, которое будет составлять будущую систему, изучить особенности технологий, которые планируется применять, предпочтения потенциальных пользователей, исследовать продукты и интерфейсные решения конкурентов. Если программный продукт создается не с нуля, а является проектом по развитию существующей системы, то необходимо выделить наиболее удачные интерфейсные решения.

На стадии проектирования подробно описываются все принципы работы системы, ее подсистем, моделей, определяется перечень функций, который будет поддерживаться вашим программным продуктом, создается документация. *Основными критериями для интерфейса* являются частота обращения к системе, длительность одного цикла работы, средний уровень предполагаемого пользователя. Проектируется внешний вид системы и ее стилевое оформление.

Прототипы системы согласовываются (заказчик должен увидеть, что это именно то, чего он хотел, а программист подтвердить, что задуманный макет можно воплотить в программном коде без каких-либо серьезных проблем), затем тестируются, «обкатываются» на пользователях, определяется основная терминология. Постепенно переходят к реализации, т.е. к созданию работающего программного продукта.

На стадии реализации пишется программный код, все макеты и прототипы превращаются в работающие решения. Поскольку полноценная реализация программного продукта возможна не сразу, то на первых этапах могут использоваться заглушки, пустышки. При наладке работы системы корректируются проектные решения, оказавшиеся ошибочными или нереализуемыми.

Часто реализацию стремятся сделать как можно более легкой (особенно в том, что касается интерфейса), чтобы едва ли не дизайнер смог писать полноценные системы или их части. Именно по этой причине начали изобретать визуальные среды разработки интерфейсов, визуальные редакторы и т.п. Наверняка вам не раз приходилось видеть: «Самая простая и легкая в настройке система управления контентом сайта!!!» Эти объявления рассчитаны именно на те компании, которые хотят, чтобы один Web-дизайнер мог работать и как программист, и как специалист по интерфейсам, и как HTML-кодер. На самом деле так не бывает, но многие в это свято верят.

Во время программной реализации системы, особенно если она не новая, почти всегда происходит ее рефакторинг. *Рефакторинг* — изменение программных решений без принципиального изменения основных функций и внешнего вида (если, конечно, речь идет не о рефакторинге интерфейсных решений). Простейшим примером может служить перевод сайта на язык XML. Представьте себе, что ранее сайт был написан на чистом HTML, а затем он так разросся, что поддерживать его стало трудно, и было принято решение использовать технологию XSL/XML для генерации HTML-страниц. Внешний вид сайта останется прежним, но изменится способ создания и передачи браузеру пользователя информации. Возможно и обратное явление: сама логика работы сайта останется прежней, но после многочисленных нареканий со стороны пользователей, дизайнеров и специалистов по юзабилити интерфейс системы необходимо будет переделать.

Приступать к тестированию необходимо с самых ранних стадий создания программного продукта. Это же касается и тестирования приложений. Начинайте пораньше и избежите осложнений.

Первое тестирование — это примитивные действия контролирующего характера, но постепенно они будут становиться все более важными. Чтобы проверить какую-то часть программного продукта, не надо ждать полной реализации. Могут быть еще не написаны те части системы, которые, например, выдают данные для работы какой-либо формы. Для проверки в таком случае используются заглушки и подставные данные. Часть системы, более или менее проверенная и отлаженная, превращается в *протестированный модуль*. Постепенно такие модули охватывают всю

систему, и тогда она тестируется целиком при помощи тех же способов, которые применялись для тестирования каждого модуля по отдельности. По результатам каждого тестирования оформляется список обнаруженных дефектов, делаются отметки об их «судьбе» (исправлены, оставлены, переведены в разряд особенностей) и расставляются приоритеты в исправлении ошибок.

Если программисту были переданы требования к интерфейсу и удобству использования, то необходимо обязательно проверить их выполнение: везде ли есть всплывающие подсказки, проставлены ли размеры картинок, правильно ли работают нестандартные идеи, поддержана ли переносимость системы (например, в разных браузерах) и т.п. Некоторое отношение к интерфейсу пользователя имеют и другие виды тестирования: тестирование производительности, нагрузочное и стрессовое тестирование, тестирование надежности и запаса прочности при работе в реальных условиях, проверка степени безопасности и уязвимости программного продукта, легкости развертывания и размещения, настройки и обновления, анализ сетевого трафика в условиях конкретной сети и т.д. В рамках данной книги мы подробно рассмотрим тестирование юзабилити.

## Резюме

1. Существуют разновидности программного обеспечения: программа, программный продукт, программный комплекс, системный программный продукт.
2. Создание качественного программного продукта предполагает детальную и тщательную проработку интерфейса.
3. Интерфейс Web-приложения практически не зависит от используемых технологий, способа организации компонентов приложения и способов хранения данных.



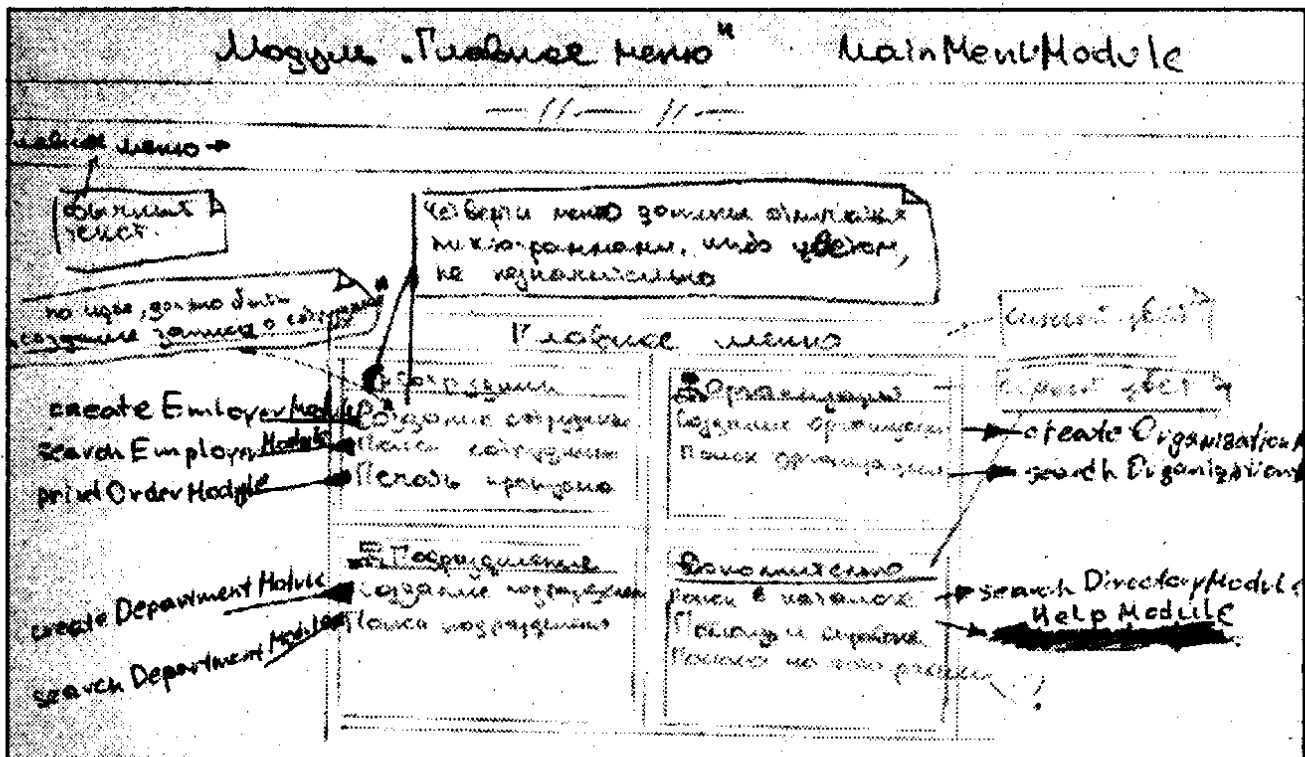
## Этапы создания интерфейса

Интерфейс программного продукта проходит четыре этапа создания: бумажный прототип, электронный прототип, макет, собственно интерфейс.

**Бумажный прототип?! Это так долго. Я на компьютере сделаю все гораздо быстрее!**

## Неизвестный проектировщик

**Бумажный прототип** — нарисованное от руки внешнее представление программного продукта (рис. 4.1). От бумажного прототипа до конечного продукта еще очень далеко. Существует миф о том, что некоторым людям проще нарисовать или написать что-то при помощи компьютера, чем от руки. На самом деле человек, рисующий макет некоторой формы при помощи мыши, клавиатуры и графического редактора, должен проделать массу сопутствующих действий как для начала работы, так и по ее завершении. Надо включить компьютер, найти и запустить подходящий



**Рис. 4.1. Бумажный прототип интерфейса с пояснениями и пометками**

редактор, сохранить результаты, распечатать на принтере и т.п. Кроме того, пользователь бессознательно начинает выравнивать и украшать того, что он сделал. Исключением являются способы ввода информации при помощи «светового пера», поскольку они наиболее соответствуют работе с карандашом и бумагой.

Вы можете рисовать кусками, схематично, можете стирать часть рисунка в процессе обсуждения и дополнять его, и для этого совсем не обязательно усаживать сотрудников за компьютер или таскать с собой ноутбук.

К сожалению, некоторые разработчики даже крупных проектов игнорируют все способы создания прототипов. Они не делают макеты форм в графическом или ином редакторе, а сразу программируют их и зашивают в коде. Это напоминает старый анекдот про программиста, который, чтобы посчитать дважды два, стал писать программу на ассемблере.

Бумажный прототип поможет также при выборе шаблонных решений для компонентов вашего сайта. Например, вы хотите использовать ограниченный набор визуальных компонентов, сформированный в виде некоторой библиотеки. Когда вы проведете анализ нарисованных форм, то вы один раз напишете HTML-представление компонента и будете несколько раз его использовать. В то же время, если вы не сделаете такой анализ, даже при построении простого макета вам придется выполнять много пустой работы.

Кроме того, восприятие изображений на бумаге отличается от восприятия того, что на экране компьютера, следовательно, оценка может быть необъективной. Лично мне, для того чтобы продуктивно читать и понимать документацию, необходимо распечатать ее на принтере.

С помощью бумажного прототипа проводится первое тестирование интерфейса. Что же можно почерпнуть из такого тестирования? Можно узнать об ошибочных решениях, принятых в ходе проектирования интерфейса, проверить как следует надписи на кнопках и очевидность предназначения элементов навигации, выяснить, каких элементов недостает, а какие являются лишними, выделить композиционные части с точки зрения пользователя. Но бумажный интерфейс не может служить материалом для динамического тестирования.

**Электронный прототип** — аккуратно оформленная версия бумажного прототипа в графическом или текстовом редакторе, например, MS Visio, Corel Draw, MS Word (рис. 4.2).

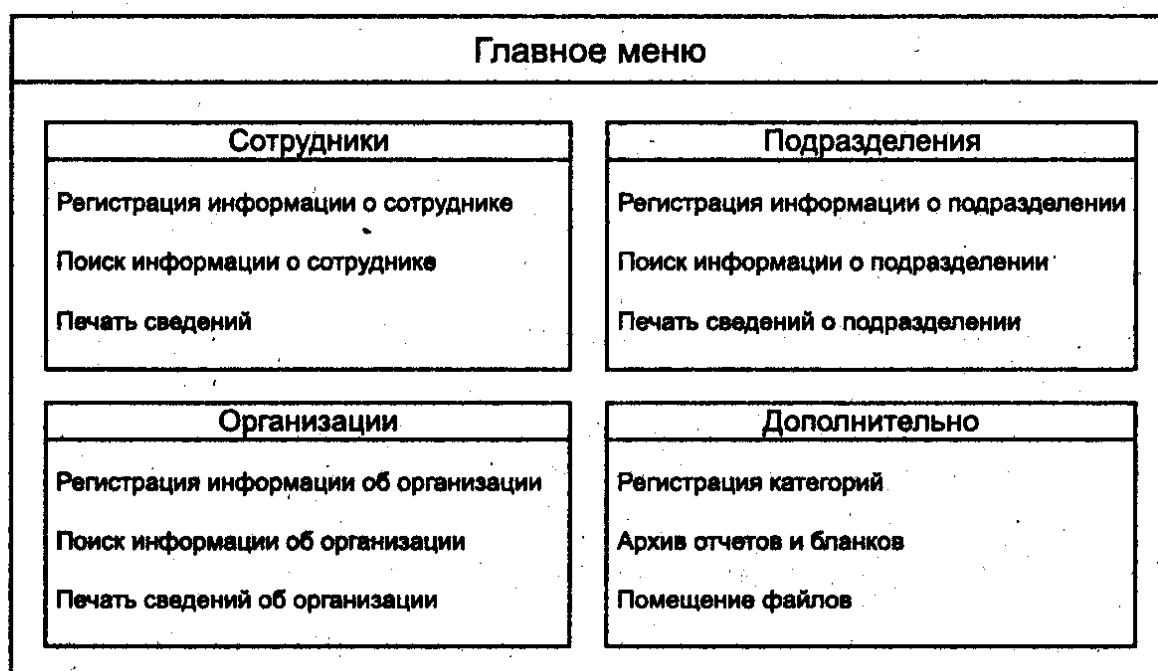


Рис. 4.2. Электронный прототип интерфейса. Пояснения и пометки уже вынесены и на данном рисунке не приводятся

Целями создания электронного прототипа являются упорядочивание информации, полученной на этапе анализа бумажного прототипа, улучшение его внешнего вида. Именно в виде электронного прототипа описание интерфейса фигурирует в проектной документации.

Третий этап — **макет интерфейса**. Этот интерфейс выглядит как реально работающая система, но по сути таковой не является. Это эмуляция работы (рис. 4.3). Вместо объектов базы данных подставляются заглушки, например HTML-форма с уже заполненными полями. Кнопки и гиперссылки ведут от одной статической страницы к другой. Цель создания такого макета — проверка интерфейса в условиях, приближенных к реальным.

Четвертый этап — **собственно интерфейс** рабочей системы (рис. 4.4). Это уже никакой не макет и не прототип, это конечная цель. Если при проектировании и анализе интерфейса все было сделано верно и вовремя, то этот интерфейс уже конечный, он не дорабатывается и не исправляется.

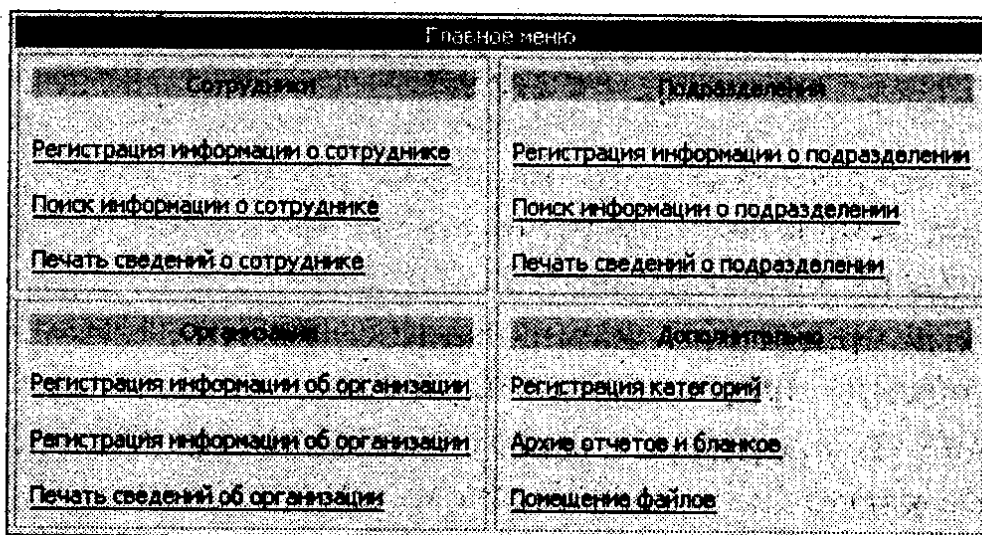


Рис. 4.3. HTML-макет формы главного меню

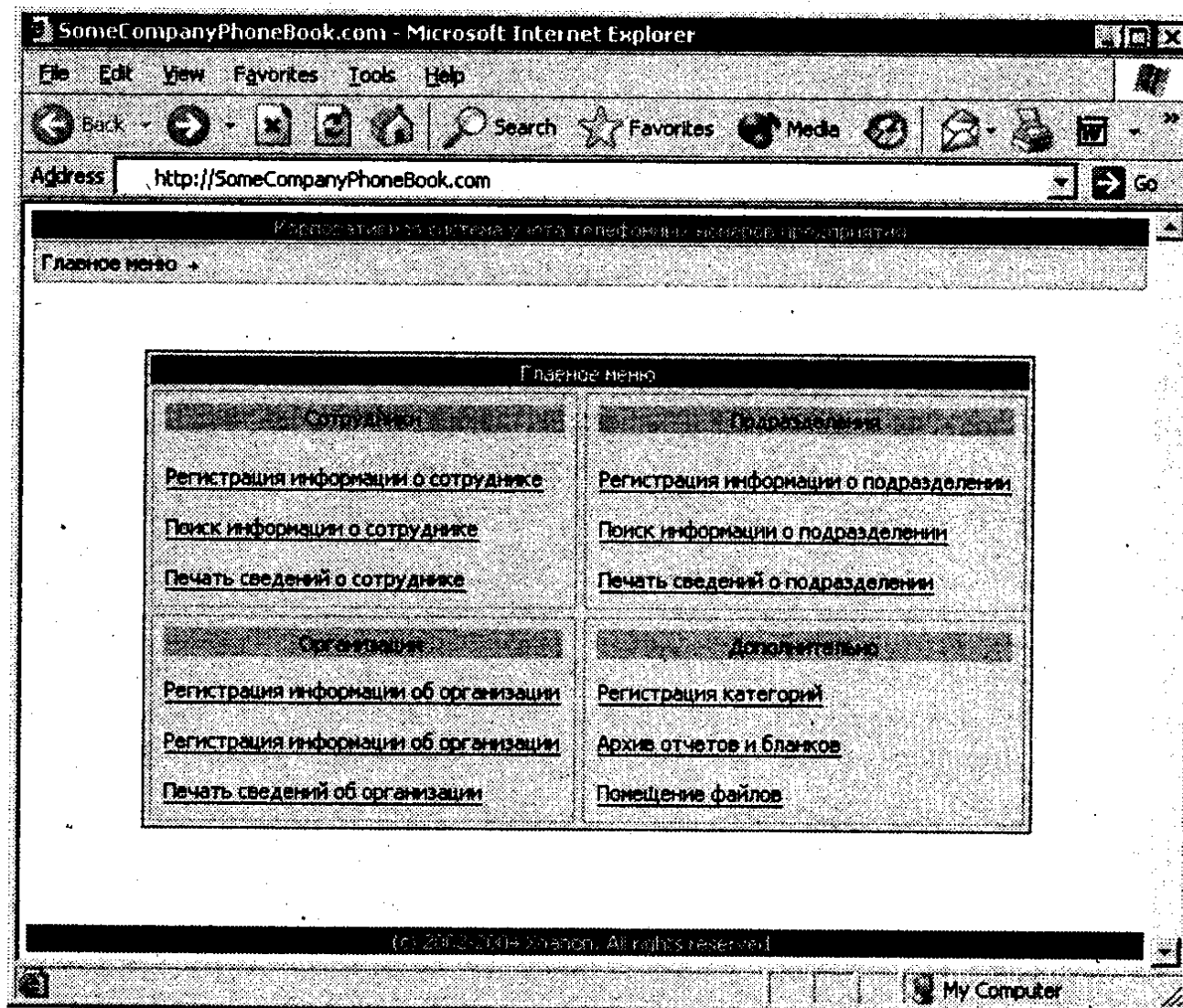


Рис. 4.4. Конечный результат — главное меню как элемент интерфейса рабочей системы

## Резюме

1. Интерфейс программного продукта проходит четыре этапа создания: бумажный прототип, электронный прототип, макет интерфейса и собственно интерфейс.
2. Бумажный прототип — нарисованное от руки первое внешнее представление программного продукта.
3. Электронный прототип — версия бумажного прототипа, выполненная в каком-либо графическом или текстовом редакторе.
4. Макет интерфейса — интерфейс, который выглядит как реально работающая система. Это эмуляция работы.
5. Собственно интерфейс — конечная цель разработки интерфейса.

## Глава 5

# Компоновка Web-страниц

## 5.1. Смысловые уровни Web-сайта

Существуют пять смысловых уровней организации Web-сайта.

### 1. Внешний вид Web-сайта

Это самый верхний, презентационный уровень. Он представляет собой совокупность шрифтов, таблиц, изображений, цветов, результатов поиска — всего, что видит пользователь. На этом уровне определено, как выглядит для посетителя содержимое сайта.

### 2. Уровень компоновки

Это схематичное разделение страницы на части. Здесь нет информации о внешнем представлении — только о смысле конкретной части страницы, например, блок поиска, блок новостей, блок авторизации. На этом уровне определено, как должны располагаться части страницы.

### 3. Уровень структуры

Этот уровень уже не имеет визуального представления, связанного со страницами сайта. На нем определено, как подчинены друг другу разделы, какова иерархия построения информации, как осуществляются переходы между разделами. Если предыдущие уровни визуальные, то уровень структуры логический.

### 4. Предметный уровень

Это абстрактный уровень, который определяет те функции и возможности, которые могут быть предоставлены пользователю.

### 5. Уровень идеи

Он называется уровнем лишь условно и определяет основное направление сайта, его суть: интернет-магазин или поисковый портал, персональный сайт или корпоративный ресурс.

Как видите, интерфейс представлен двумя верхними уровнями, а логика его построения определяется уровнем структуры.

## 5.2. Основные способы верстки и их особенности

**Верстка** — это размещение в поле страницы, предоставляемом браузером, подготовленной информации либо создание заготовки для последующего заполнения. Верстка Web-сайта является неотъемлемым этапом его создания и осуществляется с помощью языка HTML.

☑ Я прекрасно понимаю, что многие читатели воспримут эту информацию как «ликбез», но я просто обязан ее рассмотреть. Я никого не утруждаю подробностями — просто описываю основные разновидности верстки и оцениваю их с точки зрения пользователя. Если вы уверены, что знаете это лучше меня, смело пропускайте этот раздел.

### 5.2.1. Текстовая верстка

**Текстовая верстка** — верстка без какого бы то ни было размещения содержимого, это лишь само содержимое (рис. 5.1).

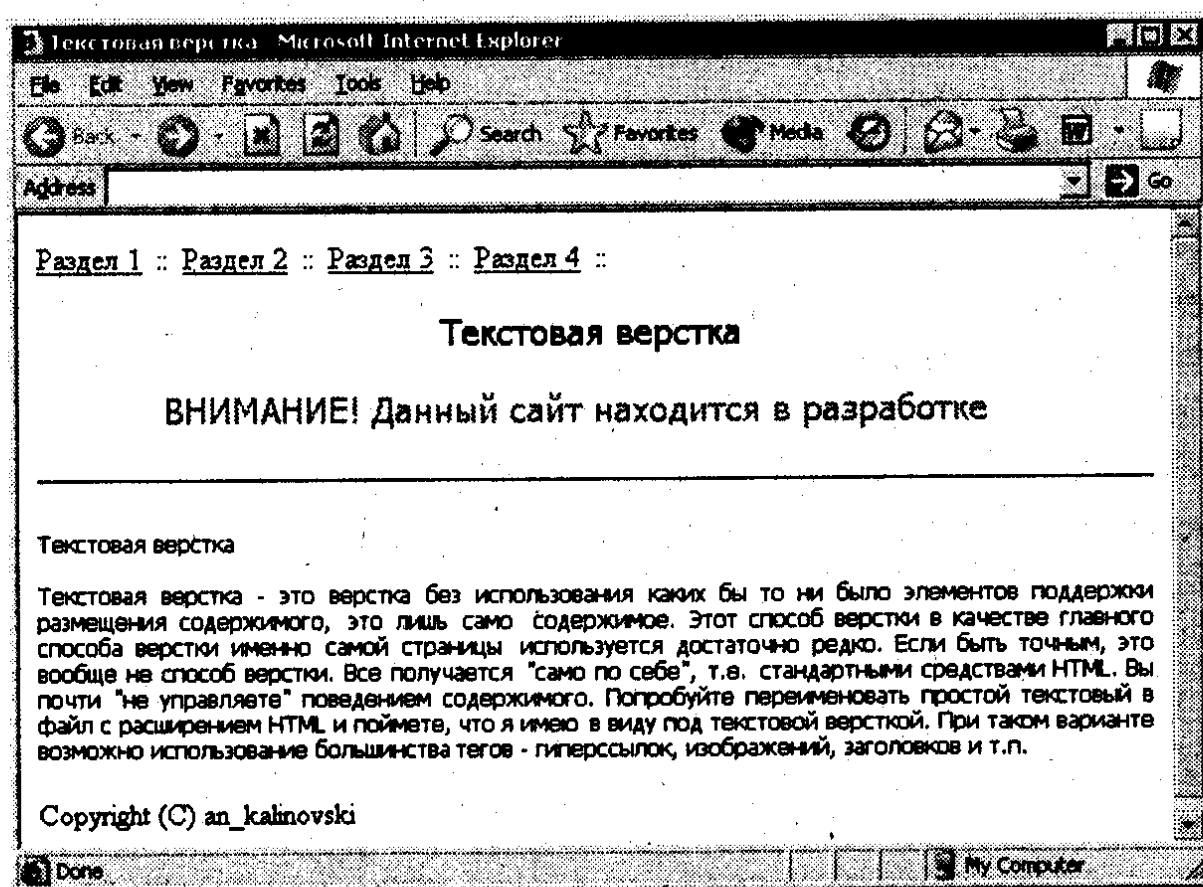


Рис. 5.1. Текстовая верстка. Аскетично, просто, никакого «дизайна ради дизайна»

В качестве главного этот способ верстки используется достаточно редко. Если быть точным, это не совсем верстка. Все получается «само по себе», т.е. стандартными средствами HTML. Вы почти не управляете поведением содержимого. Попробуйте переименовать простой текстовый файл в файл с расширением HTML и поймете, что я имею в виду. При таком варианте возможно использование большинства тегов — гиперссылок, абзацев, изображений, заголовков и т.п. С точки зрения интерфейса и пользователя этот способ хорош тем, что созданные таким образом страницы (особенно, если речь идет не только о текстовой верстке, но и текстовой версии сайта) загружаются едва ли не мгновенно. При чтении таких страниц пользователь редко отвлекается, а отыскать какой-либо элемент гораздо проще, чем на сайтах со сложной структурой.

### 5.2.2. Табличная верстка

*Табличная верстка* основана на использовании таблицы, в ячейках которой располагаются элементы страницы (рис. 5.2).

Страницы с табличной версткой «растягиваются» на всю длину содержимого. При большой длине, как правило, становятся недоступными элементы навигации, которые могут располагаться или в самом верху страницы, или в самом низу. Пользователю это не очень удобно: чтобы перейти дальше или чтобы определить, где он сейчас находится, ему надо возвращаться к верхней части страницы. При «перечитывании» содержимого страницы или переходе к другой с экрана все исчезает и приходится ждать, пока соответствующая страница вновь не отобразится.

Этот способ верстки наиболее распространен во Всемирной паутине. Наряду с текстовой версткой, он отлично подходит для компоновки обособленных частей страницы. Что я имею в виду? Представьте себе, что ваша страница разделена на три части: заголовков, основное содержимое и «подножие» с информацией о вас. Верхний и нижний блок могут быть сверстаны при помощи текстовой верстки, сверху картинка, снизу ваши реквизиты. Если блок основного содержания достаточно сложный и простого форматирования недостаточно, то в таком случае он может быть сделан при помощи таблицы.



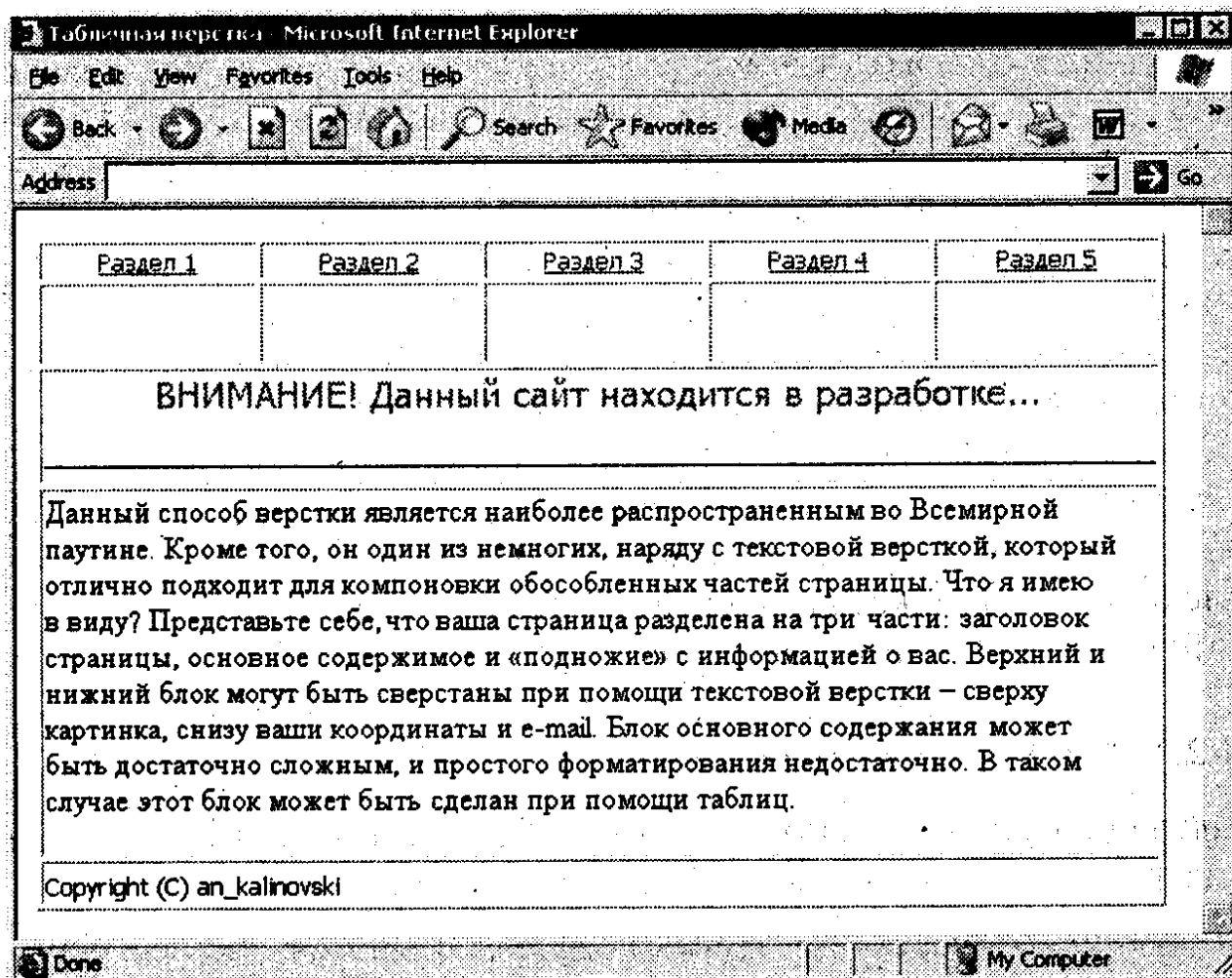


Рис. 5.2. Табличная верстка. Содержимое страницы выглядит более структурированным, доступны многие варианты оформления

### 5.2.3. Верстка при помощи фреймов

**Фрейм** (англ. *frame рамка*) — это HTML-тег. Основной смысл верстки при помощи фреймов — разделение страницы на определенное количество рамок, в которые загружаются отдельные страницы. Таким образом, главная страница содержит всего 10—20 строчек описания рамок, их размеров и того, какая страница в какую из рамок должна загрузиться (рис. 5.3).

Главное достоинство фреймов с точки зрения пользователя в том, что новая страница загружается в какую-то из рамок, а остальные рамки при этом не исчезают и не перезагружаются. Прокручивать надо не саму страницу, а только содержимое конкретной рамки. Кроме того, если расположить в одной из рамок навигацию по странице, то она будет доступна в любой момент.

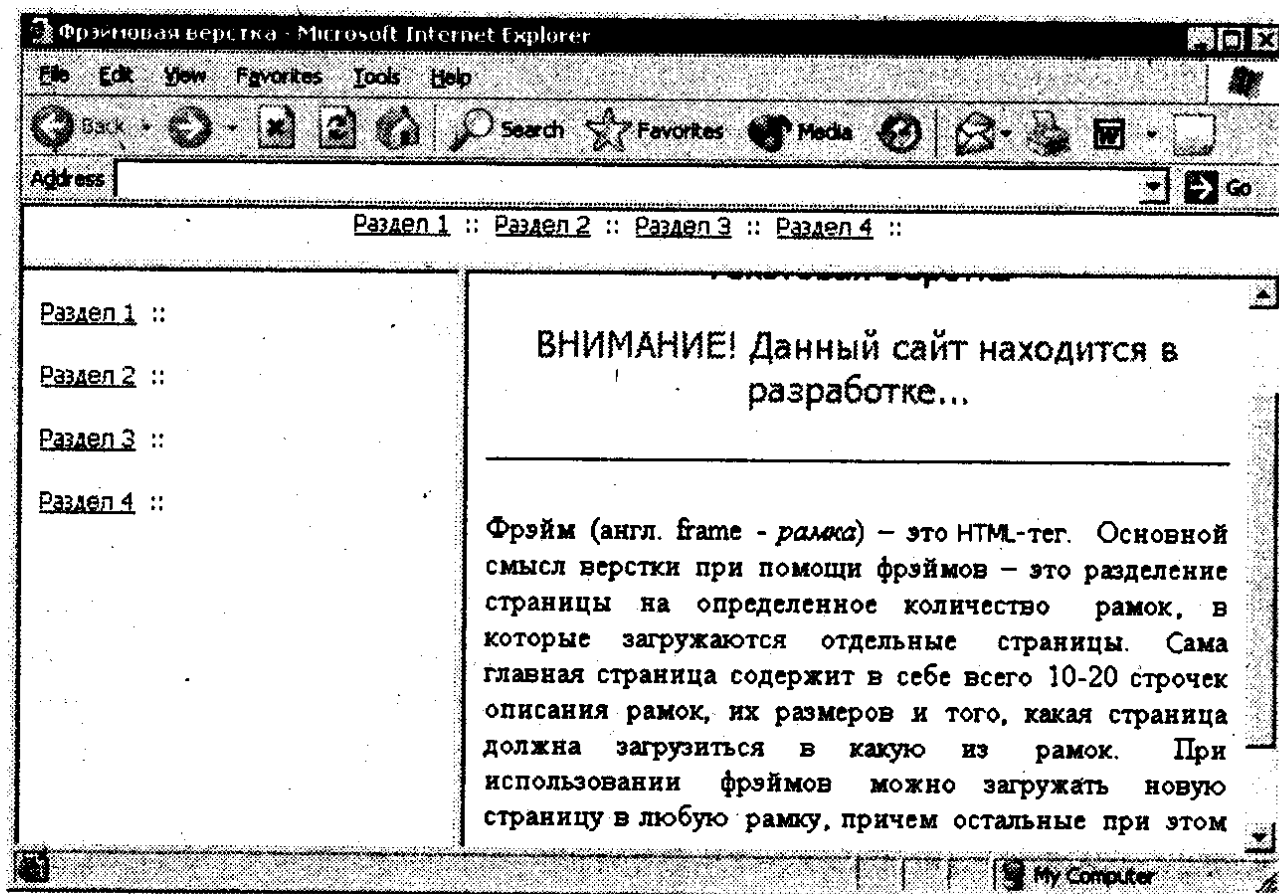


Рис. 5.3. Фреймовая верстка. Три достаточно обособленные рамки, их размеры можно изменять. При прокручивании основного содержимого навигация не исчезает

У фреймов есть ряд существенных недостатков. Поскольку главная страница сайта — всего лишь описание рамок, то она без специальных усилий разработчика не будет проиндексирована в обычных поисковых системах, которые отыскивают гиперссылки и далее идут по ним. Если пользователь попал не на первую страницу сайта, а он вполне может попасть на следующие, то он окажется в капкане, если вы не разместите на этой странице элементы навигации по сайту. Ему больше некуда будет идти, кроме пары-тройки ссылок, которые могут оказаться на странице. Опытного пользователя это не испугает — он просто вручную подправит адрес в адресной строке, а вот новичка поставит в тупик. Поскольку сайт работает через главную страницу, описывающую рамки, то невозможно оставить в Избранном адрес заинтересовавшей страницы.

Что касается разработчика, то фреймы для него чаще всего — это постоянная головная боль. Переходы между отдельными

фреймами сложны, нужно аккуратно передавать информацию между ними, следить, чтобы информация действительно попала именно туда, куда предназначалась.

Часто фреймы используют для составления системы справки и помощи, потому что они наиболее подходят под концепцию организации справочной системы.

#### 5.2.4. Флэш

**Флэш** — верстка при помощи технологии, разработанной компанией Macromedia. В основном эта технология используется для создания шикарных мультимедийных презентаций, хотя иногда таким образом строят едва ли не весь сайт (рис. 5.4). Все мигает, прыгает, дребезжит и вращается. Выглядит флэш, по меткому выражению одного из отечественных специалистов, «стрелочки — тррррррр, буквочки — фрррррррр, кнопочки — пшшшш».

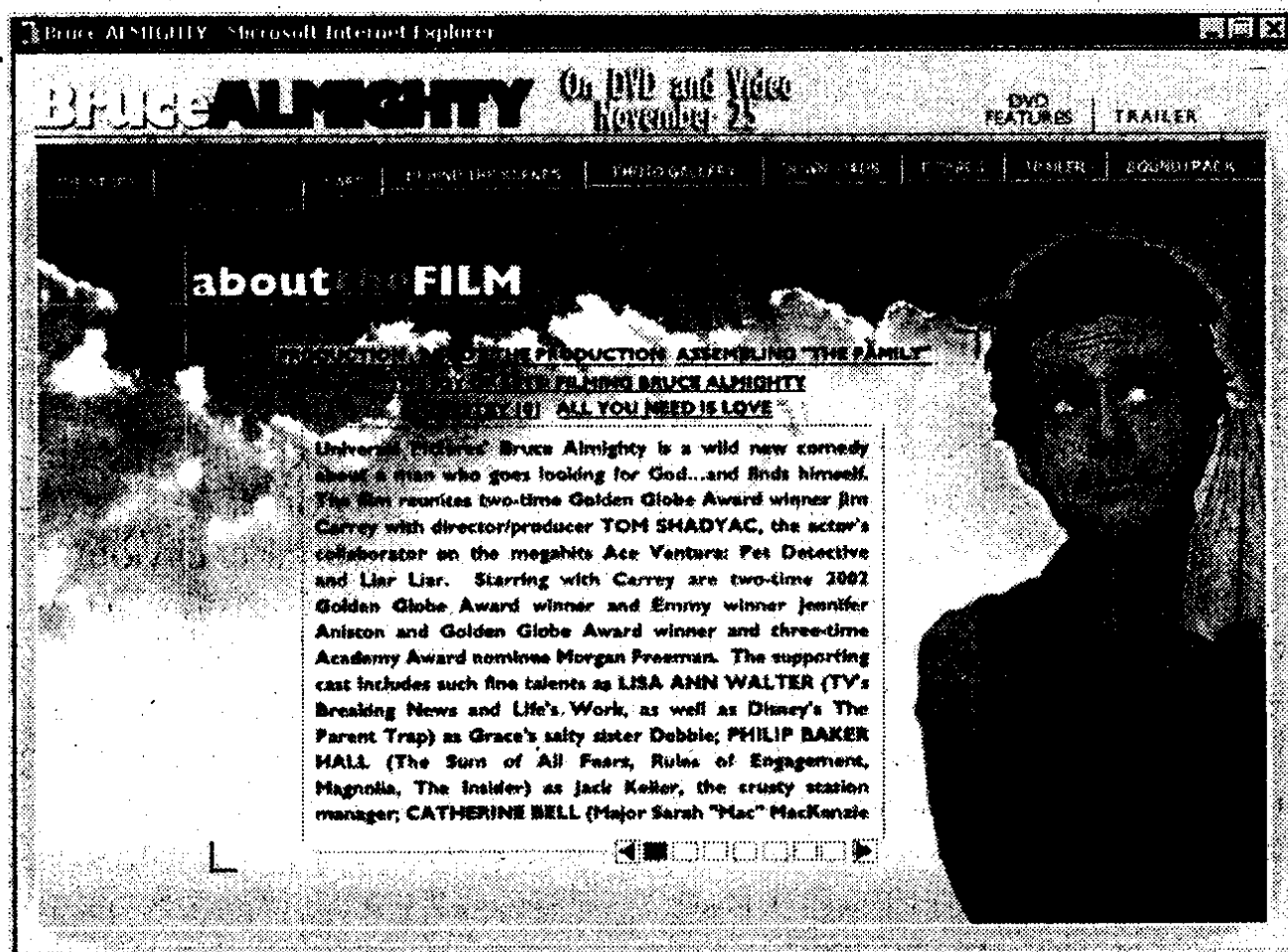


Рис. 5.4. Информация на сайте о фильме «Брюс Всемогущий» сверстана при помощи флэш

На самом деле верстка происходит вне HTML-страницы — в специальном редакторе. На страницу загружается программный модуль, который почти не имеет отношения к Web-интерфейсу. Он просто отображается внутри браузера. К тому же отображение на странице сверстанного при помощи флэш содержимого требует специального программного компонента, установленного на компьютере пользователя. Для пользователя очень неприятно и неприятно, что во флэш не работают такие нужные кнопки Вперед/Назад, Поиск, нельзя копировать текстовое содержимое, изображения и т.д.

Я считаю использование данной технологии для основной верстки страницы неприемлемым.

### 5.3. Оформление главных страниц

Главная (начальная) страница сайта считается «лицом» компании, которой этот сайт принадлежит, поэтому часто оформление главной страницы сайта и его вторичных страниц различаются. Из всего, что я когда-либо видел, для первой страницы наиболее часто используются следующие приемы дизайна и верстки:

- статическая страница-заставка с краткой информацией или слоганом и ссылкой для перехода к основному содержимому;
- флэш-заставка;
- страница со списком главных разделов, оформленных нестандартным образом, например, в виде часов или карты;
- заставка-игра, где надо ухватить за хвост какую-нибудь белку или дернуть за вымя корову, чтобы попасть к основному содержимому.

Как правило, в основном содержимом сайта никаких таких изысков не наблюдается — все привычно и стандартно.

Типичным примером оформления главной страницы является «Дом-страница Артемия Лебедева». На ней расположена только ссылка для входа, изображение в виде дома-стрелки и копирайт (рис. 5.5).

При любом выбранном варианте оформления следует помнить, что на первом плане должна быть информация, а не дизайнерский креатив.

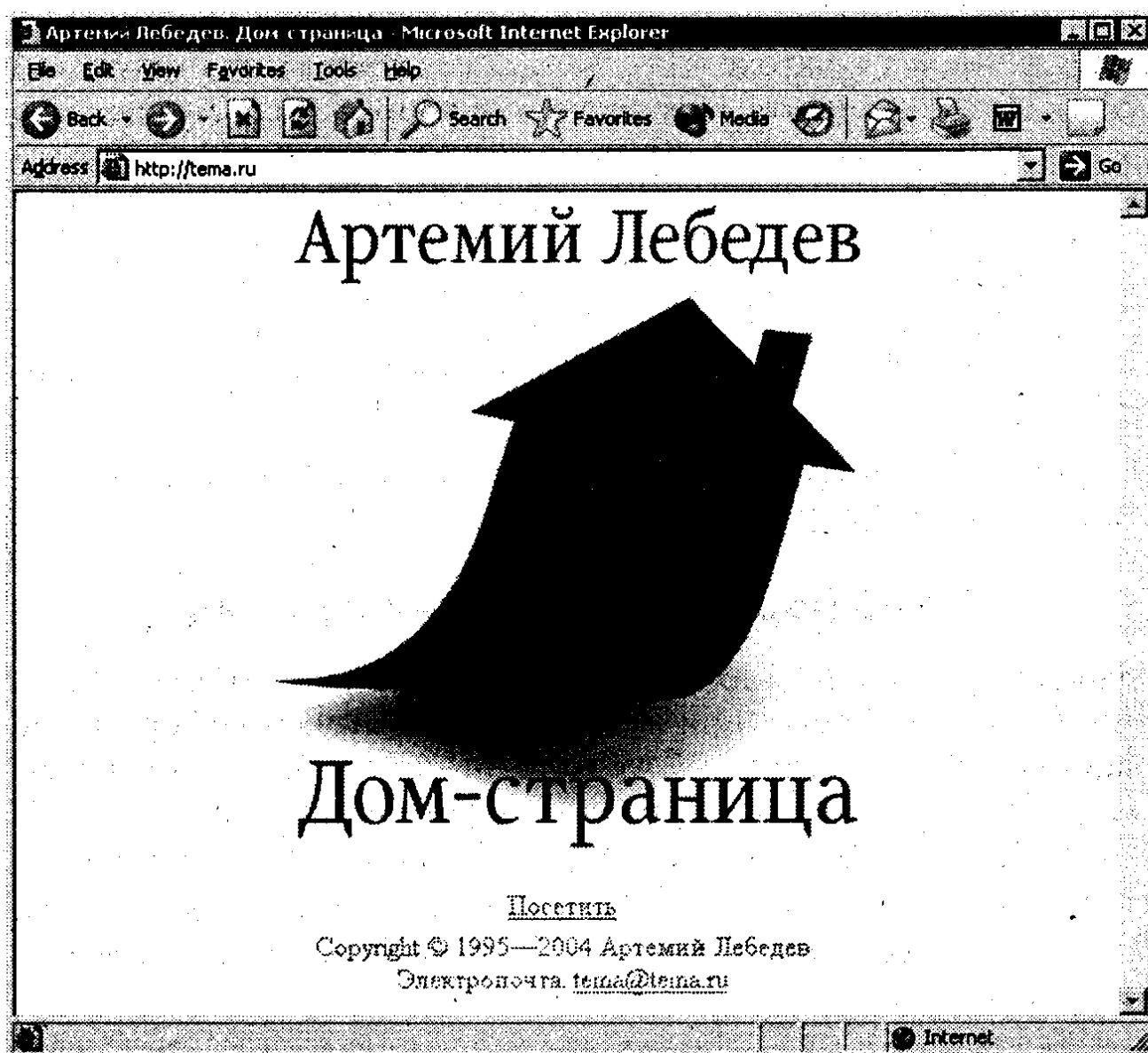


Рис. 5.5. Начальная страница сайта знаменитого российского дизайнера Артемия Лебедева

## 5.4. Виды верстки

В языке HTML существуют два способа задавать размеры элементов страницы: в относительных величинах, например, в процентах от предоставленного элементу пространства, и в абсолютных, например, в пикселях.

- ✓ На самом деле пиксель — величина тоже в некоторой степени относительная. Текст, написанный шрифтом 12 п, будет разной величины при различных разрешениях монитора.

Если размер (ширина/длина) элемента задается в процентах, например, 100 %, то такой элемент занимает всю предоставленную ему площадь и будет расширяться или сужаться до тех пор, пока позволяет обрамляющий элемент. Такую верстку (или дизайн) называют **«резиновой»**.

Если размер элемента задается в абсолютных величинах (в пикселях), то в большинстве случаев его изменить нельзя. Такую верстку называют **жесткой**, или **фиксированной**. Основное среднестатистическое разрешение монитора у пользователей интернета считается равным 1024×768, поэтому размеры элементов определяют исходя из этого разрешения. В действительности их размер чуть меньше, т.к. учитывается ширина полосы прокрутки и элементов окна браузера.

**Смешанный** способ верстки позволяет содержимому растягиваться на всю длину и ширину окна браузера, но конечный размер должен быть не меньше определенного предела (800×600). Иногда в одну из ячеек таблицы с основным содержимым помещают изображение шириной около 800 пикселей, чтобы меньше этого размера сайт «не скручивался».

Теперь подробнее о плюсах и минусах этих способов с точки зрения пользователя.

**Плюсы жесткой верстки:**

- если композиция сайта проработана очень тонко и малейшее нарушение положения элементов может ее разрушить, то при жесткой верстке этого не произойдет;
- страница одинаково приемлемо выглядит при всех разрешениях монитора;
- содержимое страницы ведет себя гораздо более предсказуемо.

**Минусы жесткой верстки:**

- при разрешениях, меньших, чем то, под которое адаптирована верстка, необходимо горизонтальное прокручивание страницы. При больших разрешениях возникают неэстетичные пустоты;
- если разрешение экрана больше того, что жестко задано, то внутри оставшегося пустого места содержимое нужно как-то выравнивать. Если содержимое выровнено влево или вправо по краю страницы, то это не совсем красиво (это лично мое мне-

ние), а если содержимое выровнено по центру страницы, то невозможно использовать элементы с абсолютным позиционированием (точной привязкой к каким-то координатам внутри окна). Часто именно так делают выпадающее меню, и в этом случае интерфейс ограничен в своих возможных элементах.

Плюсы «резиновой» верстки:

- страницы гибко подстраиваются практически под любое разрешение монитора;
- пустого места не остается, следовательно, поместится больше информации, не будет лишних пустот;
- всегда возможно использование элементов с абсолютным позиционированием (при условии, что они будут выровнены влево).

Минусы «резиновой» верстки:

- поведение содержимого в принципе непредсказуемо, текст и картинки переносятся как получится и где получится, подстраиваясь под размеры окна;
- некоторые элементы не всегда корректно ведут себя (точнее, зависят от многих факторов), если они заданы в процентах.

Смешанный способ верстки, сочетая в себе достоинства обоих предыдущих способов, устраняет многие их недостатки. Я считаю его лучшим из трех.

## 5.5. Понятие модульной сетки

**Модульная сетка** (шаблон, трафарет) — воображаемая сетка, образованная, как правило, горизонтальными и вертикальными линиями, к которым «привязываются» основные элементы страницы (рис. 5.6).

Это понятие заимствовано из издательского дела (впрочем, как и верстка), но актуальное и для создания Web-страниц. Именно модульная сетка позволяет определить основные крупные блоки сайта. Даже если о ней не задумываться, она получается как бы сама по себе. По сложности модульной сетки можно в некоторой степени оценить сложность самой верстки сайта и его компоновочных блоков.

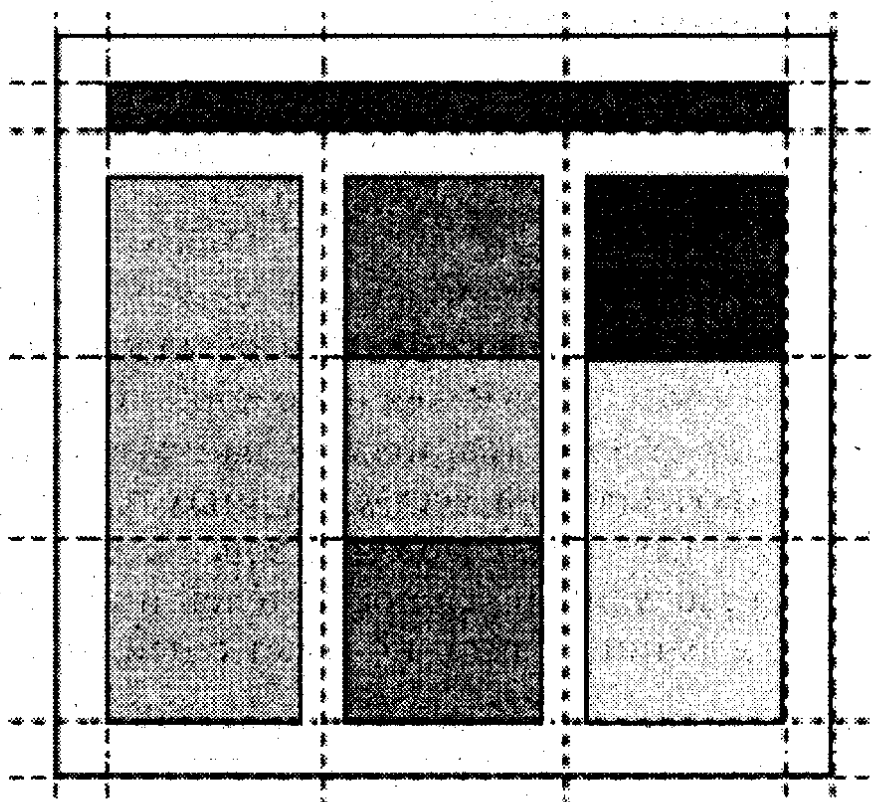


Рис. 5.6. Модульная сетка. Все размеры пропорциональны друг другу

Содержимое страницы визуально разделено на отдельные модули. Модульная сетка отражает второй уровень структуры сайта — уровень компоновки страниц. Могу предложить вам небольшое развлечение: попытайтесь мысленно или на бумаге прорисовать модульную сетку некоторого количества известных вам сайтов, и вы поймете, во-первых, что это такое, во-вторых, увидите, какие компоновочные решения используют чаще всего.

## 5.6. Основные элементы компоновки страниц

Теперь нам предстоит довольно кропотливая работа — рассмотреть основные компоновочные блоки, используемые в построении Web-страниц. Блоки следует понимать как некоторую абстракцию, а не как конкретный текст с кнопками и полями.

По назначению все блоки условно делятся на навигационные, информационные, сервисные, рекламные и дизайнерские.

Следует заметить, что различные блоки могут включать друг друга. Например, в один блок «Основное содержание» могут входить несколько блоков «Раздел», блок «Анонс», блок «Фотографии».



### 5.6.1. Навигационные блоки

- *Блок «На главную страницу»*

Данный блок представляет собой гиперссылку, оформленную в виде логотипа либо названия компании, которой принадлежит сайт (рис. 5.7). Он чаще всего ведет пользователя на главную страницу сайта. Если сайт является структурной единицей набора ресурсов, то этот блок может вести и на головной сайт, при этом должно быть соответствующее пояснение. Блок располагается в левой верхней части страниц и не занимает много места. Он должен быть повторен на всех страницах сайта, но на небольших персональных сайтах может и вовсе отсутствовать. Он предназначен для заблудившихся посетителей и тех, кто пришел не на начальную страницу сайта, но хочет с ней ознакомиться.

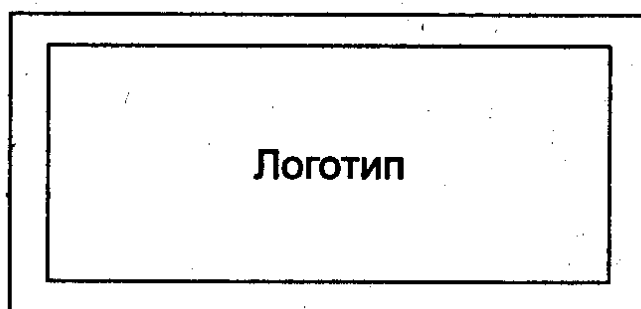


Рис. 5.7. Блок «На главную страницу» может включать логотип, название или адрес и даже слоган

- *Блоки поиска и быстрого перехода*

Несмотря на то что эти функции разные, их можно объединить в один блок из-за их внешней схожести и некоторой схожести по сути. Оба варианта представляют собой поле ввода (редактируемое или нет) с кнопкой выполнения введенного запроса (рис. 5.8). Иногда при помощи JavaScript в блоке быстрого перехода убирают кнопку, и тогда посетитель попадает на другую страницу сразу же после выбора из выпадающего списка. Однако такое решение не всегда очевидно для пользователя и может вызвать проблемы совместимости в различных браузерах. Наиболее часто блоки поиска и быстрого перехода располагаются в верхней правой части страницы. Они важны для навигации, поэтому «прятать» их от посетителя не рекомендуется, особенно если на страницах много информации.

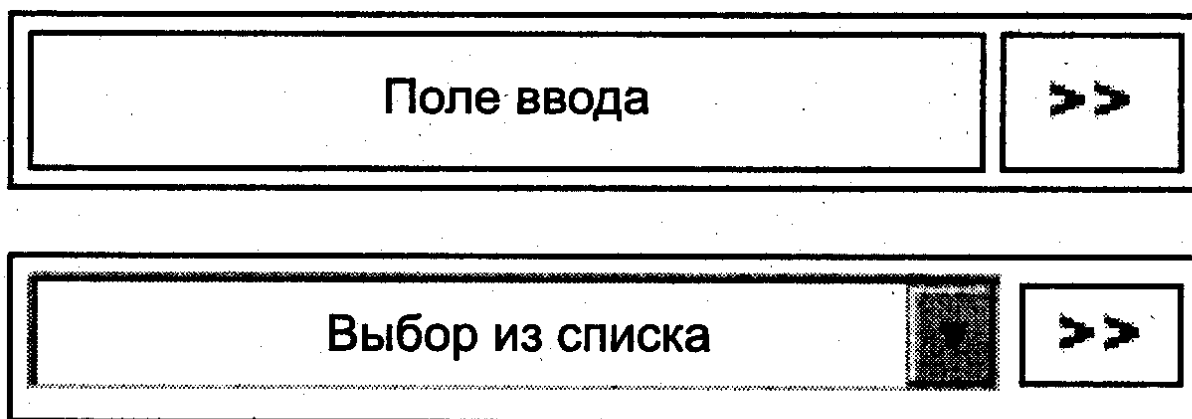


Рис. 5.8. Эти два блока действительно очень похожи. Для посетителей они порой настолько очевидны, что некоторые разработчики даже не подписывают их

- *Блок горизонтального меню*

Один из самых главных блоков на странице. Под блоком горизонтального меню понимается список гиперссылок, ведущих к основным разделам сайта (рис. 5.9). Гиперссылки располагаются на одной горизонтальной линии и могут быть оформлены и как обычный текст (меню), и как текст в виде изображения, и как символ (домик, корзина, конверт), и как вкладки. Они могут быть любого цвета и размера.

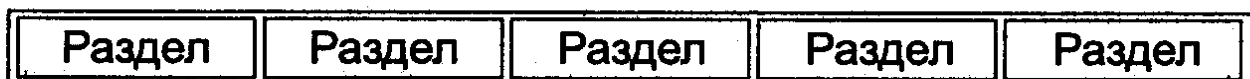


Рис. 5.9. Список разделов сайта обобщенно выглядит именно так

Горизонтальное меню может быть двухуровневым (рис. 5.10). В верхней части находятся разделы первого уровня, при вызове которых появляется начальная страница данного раздела, а также меню второго уровня со списком подразделов.

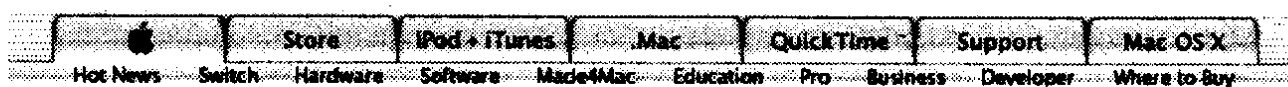


Рис. 5.10. Горизонтальное двухуровневое меню с графическими и текстовыми элементами на apple.com

- *Блок вертикального меню*

Блок вертикального меню по смыслу соответствует горизонтальному. В нем ссылки на разделы расположены одна под одной (рис. 5.11). В силу привычности данный блок располагают по левому краю страницы, но он может находиться и справа, и с обеих сторон одновременно. Этот блок может быть как статичным, так и с выпадающим меню или раскрывающейся древовидной структурой.

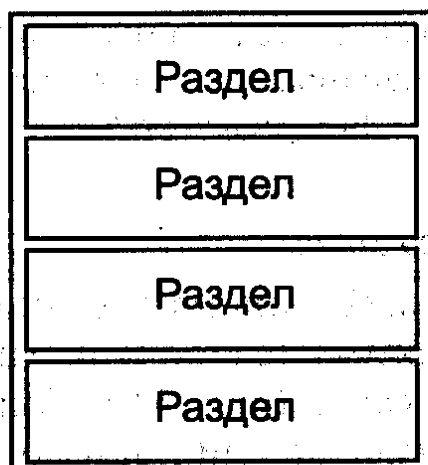


Рис. 5.11. Вертикальное меню. Чуть более «нахальный» родственник горизонтального: занимает больше места на странице и больше бросается в глаза

- *Блок вторичной навигации*

Визуально представляет собой усеченный вариант горизонтального или вертикального меню (подробно о вторичной навигации рассказывается в главе 6 «Элементы интерфейса»). С точки зрения информационного наполнения сайта этот блок не является главным и чаще всего содержит сведения о компании-владельце сайта.

- *Блок навигации по выборке*

Этот блок необходим, когда мы работаем с выборкой некоторых объектов (изображения, ссылки, результаты поиска), которую невозможно отобразить целиком. Для перемещения между частями выборки используется специальная навигация. Для удобства пользователя должен быть визуально выделен текущий фрагмент, должны отображаться соседние элементы. Если не хватает места для отображения всех порций, то следует использовать стрелки или надписи Следующая/Предыдущая (рис. 5.12).



Рис. 5.12. Этот блок применим в большинстве случаев: от ознакомления с результатами поиска в поисковом портале до путешествия по виртуальным фотогалереям

- **Блок авторизации**

Этот блок, наверное, самый предсказуемый и понятный (рис. 5.13). Он располагается на тех сайтах, где пользователь должен идентифицировать себя, чтобы получить доступ к определенной информации или чтобы система могла распознать его и основывать свою работу на ранее введенных данных (например, для очередного заказа в интернет-магазине).

Рис. 5.13. Все просто и знакомо. Я только исключил уже привычно-ироничное «Забыли пароль?» и «Вы впервые у нас? Давайте зарегистрируемся»

- **Блок «Подвал» (текстовые гиперссылки)**

«Подвал» используется для текстовых гиперссылок на основные разделы сайта (рис. 5.14). Он должен быстро загружаться и быть доступным в момент, когда фокус внимания находится внизу страницы. Достаточно удобно: закончил работу с этой информацией и готов перейти к следующей. Часто под «подвалом» понимают все, что находится внизу страницы, а не только список основных разделов.

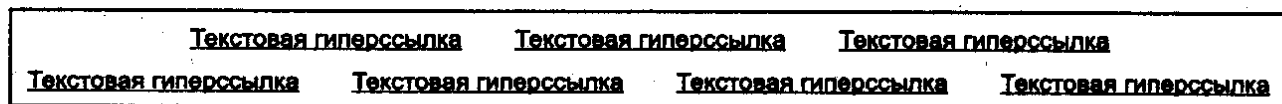


Рис. 5.14. «Подвал» с текстовыми гиперссылками

- *Блок навигационной строки*

Этот блок — последовательность гиперссылок, определяющая путь посетителя к текущей странице. Навигационная строка показывает, где был пользователь, и позволяет быстро вернуться на один и более шагов назад (рис. 5.15). Если список элементов становится слишком длинным, то можно отображать только несколько первых и последних ссылок, а промежуточные заменить на многоточие.

Уровень 1 → Уровень 1-2 → Уровень 1-2-2 → Уровень текущий

Рис. 5.15. Иногда навигационную строку называют «ссылки-цепочки»

## 5.6.2. Информационные блоки

- *Блок содержания*

Блок содержания состоит из заголовка и одного или нескольких блоков с основным содержанием страницы (рис 5.16).

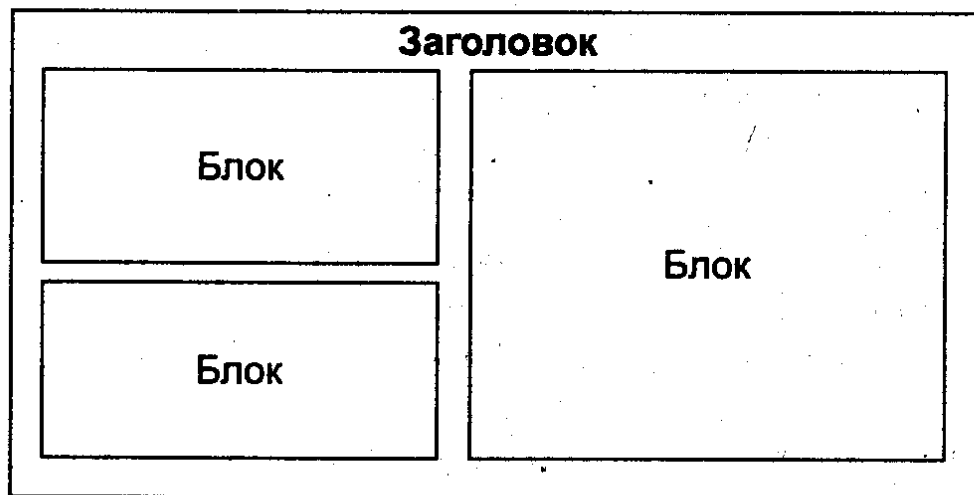


Рис. 5.16. К столь абстрактному изображению и подписать нечего. Это основное содержимое страницы, и все

- *Блок текущей информации*

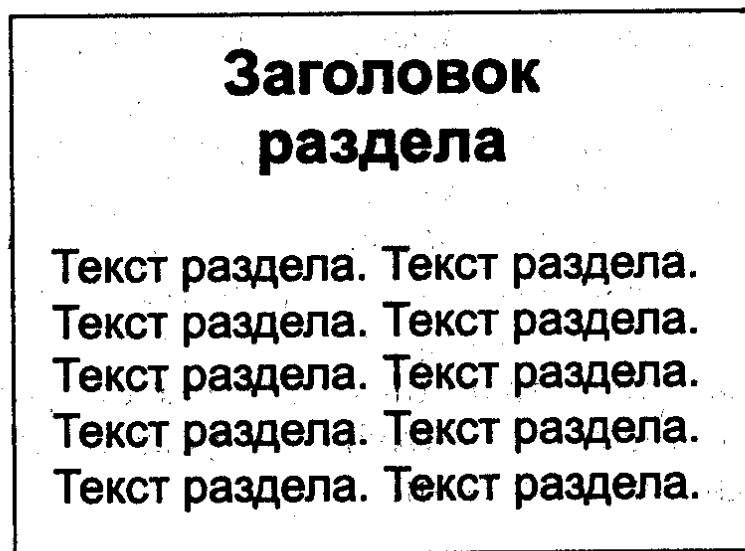
Этот блок используется при необходимости сообщить посетителю какие-то краткие сведения, полезные для его работы. Как правило, такие блоки маленьких размеров и располагаются относительно произвольно, часто внутри других блоков (рис. 5.17).

Сегодня пятница, 7 мая 2004 года : 14:46

*Рис. 5.17.* Я и так помню, что сегодня пятница, 7 мая 2004 года, зато курсы валют или количество дней до Дня рождения Пушкина знаю не всегда

- **Блок «Раздел»**

Данный блок достаточно обособленный и содержит информацию, несколько отличающуюся от основного содержимого страницы. Это может быть анонс, новость, краткое описание услуг компании, опрос и т.д. (рис. 5.18). В блоке должно быть название и содержание раздела, а может еще и гиперссылка, позволяющая перейти к полному содержанию, например, Смотреть новость полностью. Содержимым блока может быть и текст, и набор гиперссылок, и изображение.



*Рис. 5.18.* Блок «Раздел»

- **Блок изображений (галерея)**

Это набор изображений, совмещенный с блоком навигации по выборке, иногда может быть только одно изображение (рис. 5.19).

Изображения не всегда загружаются достаточно быстро. Однако по непонятной мне причине на многих сайтах этот блок состоит из одного большого изображения, вместо нескольких маленьких, которые можно просмотреть в большем размере дополни-



Рис. 5.19. Медленно загружающиеся картинки иногда вызывают досаду

тельным шагом. Я предполагаю, что, во-первых, разработчики ленятся делать картинки двух размеров, а во-вторых, ленятся делать два вида блока — для множества картинок и для одной. Результат — раздражение посетителя. Если я не хочу внимательно рассматривать каждую картинку из шести представленных, я вынужден ждать загрузки страницы с полномасштабным изображением шесть раз, вместо того чтобы один раз дождаться загрузки шести маленьких изображений.


### 5.6.3. Сервисные блоки

- Блок «Выбор языка»

Этот блок необходим, если ваш сайт поддерживает несколько языков. Часто его располагают рядом со вторичной навигацией (см. § 6.3 Элементы навигации), а наименование языка пишут на нем же самом (рис. 5.20).



Рис. 5.20. Do you speak English?

 По такому же принципу построен блок изменения кодировки текста: быстрый переход к нужной кодировке осуществляется при помощи гиперссылки, а не средствами браузера.

- *«Пустой блок»*

Этот блок представляет собой пустое место между другими и может служить для отделения одного блока от другого, для рекламного баннера. Чаще всего он появляется сам по себе в процессе верстки (например, пустое место под горизонтальным меню, когда меню уже закончилось, а основное содержание страницы располагается ниже).

- *Блок «Версия для печати»*

Наверное, наименьший по размерам блок, он инициирует вызов текущей страницы и оптимизирует ее для отправки на принтер средствами браузера. Как правило, он располагается вверху или внизу основного содержимого страницы и совмещен с пиктограммой принтера (рис. 5.21).

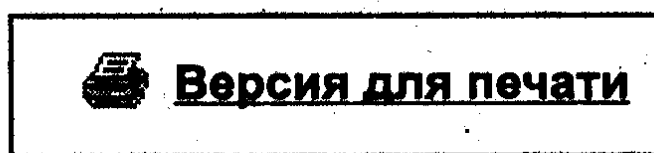


Рис. 5.21. Долой картинки и украшения. Вот уж где выполняется главный постулат интернета — информация превыше всего

#### 5.6.4. Рекламные блоки

- *Блок «Название и слоган»*

Этот блок содержит название компании или самого сайта и, возможно, слоган вроде «Уважаем классику, ценим новое». Этот блок оформляется чаще всего крупным шрифтом, иногда на фоне рисунка или вместе с логотипом компании (рис. 5.22). Назначение его, думаю, понятно — это громкое заявление компании о себе.

- *Блок «Копирайт»*

Блок описания авторских прав компании-владельца сайта, иногда со ссылкой на его создателя, например «Дизайн SuperWeb-Studio».



Как правило, этот блок — одна-две строчки в нижней части страницы (рис. 5.23).

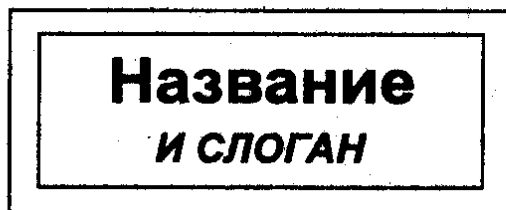


Рис. 5.22. Существуют целые компании по придумыванию лозунгов и названий. А это их «хлеб»

Copyright (c) Somebody Inc. 1987-2004 All rights reserved

Рис. 5.23. Все кому-то принадлежит. И все любят об этом напоминать

- *Блок «Дизайнерское изображение»*

Изображение, созданное дизайнером для украшения сайта и не являющееся одним из основных элементов содержания, может быть использовано и в качестве фона для иных блоков, например, для блока «Название и слоган» или блока текущей информации.

- *Собственно рекламный блок*

Несмотря на то что такой блок напоминает «Раздел», он может выглядеть как угодно: тут все зависит от фантазии разработчика. Это может быть и мерцающий баннер на половину экрана, и маленькая гиперссылка-«завлекалка», и имитация кнопки, например, «Рейтингуется SpyLog» или «Участник Rambler Top 100». Реклама — она и есть реклама — всегда стремится к разнообразию (рис. 5.24).

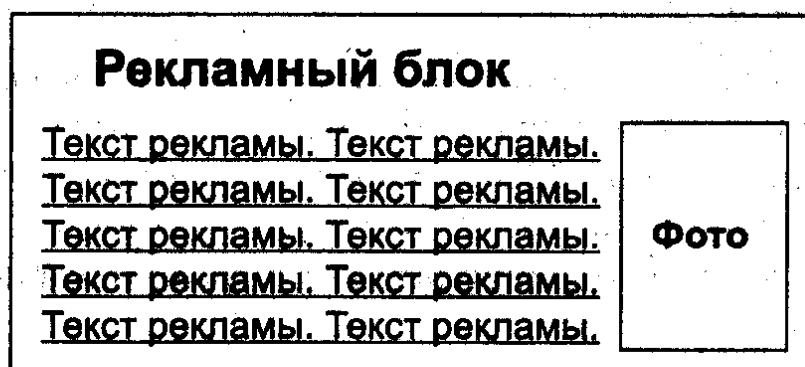


Рис. 5.24. А теперь реклама. Вы читаете самую лучшую книгу по построению Web-интерфейса и улучшению удобства использования сайтов. Обязательно расскажите о ней друзьям

## 5.7. Пример компоновки

Вариантов компоновки блоков великое множество, что позволяет создавать уникальные и запоминающиеся Web-сайты. Прежде всего следует решить, к какой цели вы стремитесь: к выражению собственных творческих способностей или к популярному и удобному информационному ресурсу. Рассмотрим в качестве реального примера сайт знаменитой компании IBM (рис. 5.25).

Данная страница сделана при помощи жесткой верстки и оптимизирована под разрешение 800×600. Разработчики выбрали жесткий дизайн для главной страницы, очевидно, потому, что содержимому просто некуда растягиваться — вся композиция разрушится. Добавить же еще один блок с каким бы то ни было содержимым нельзя — при таком разрешении он «исчезнет».

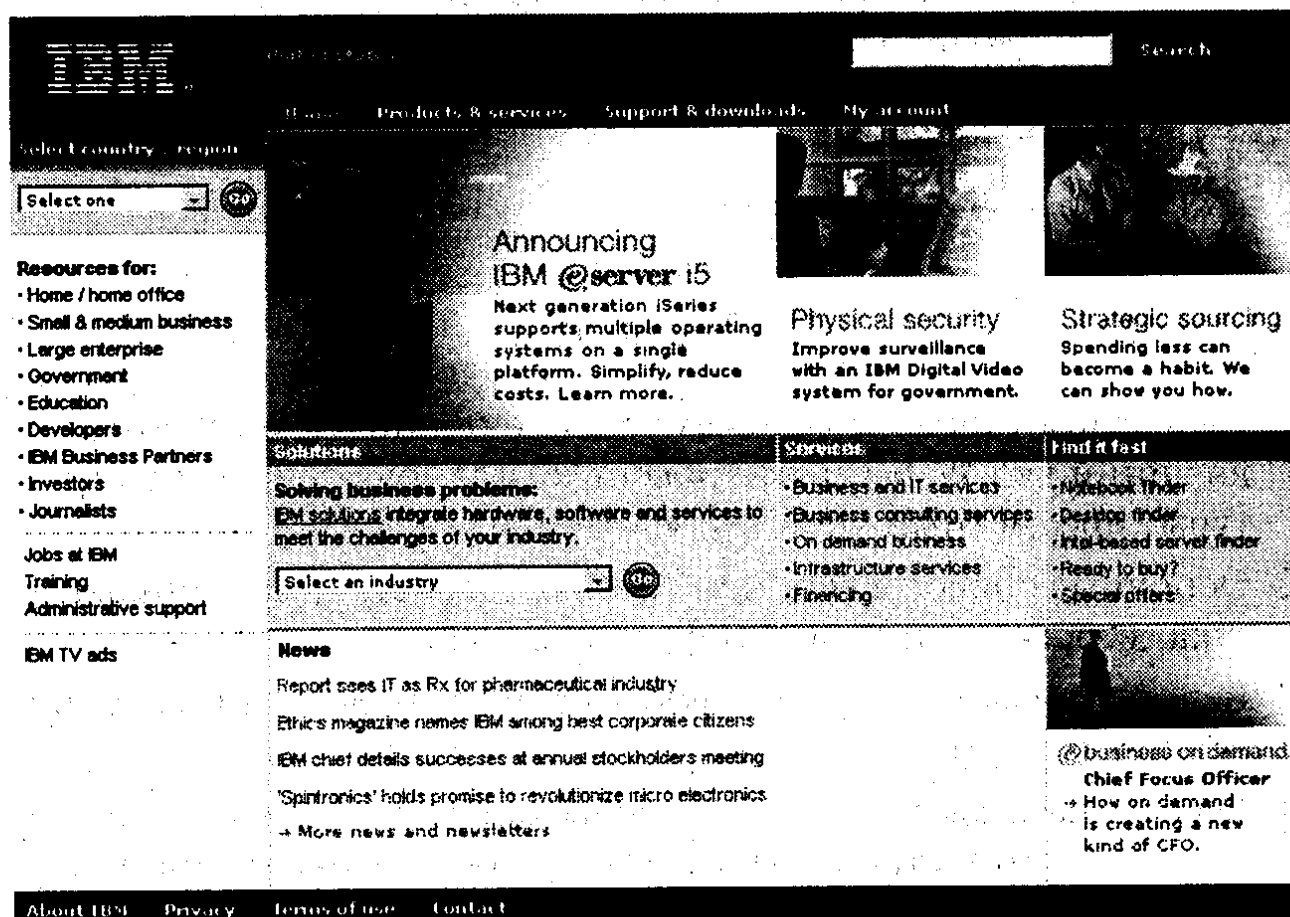


Рис. 5.25. Главная страница сайта компании IBM

Ниже изображена эта же страница, но только с точки зрения компоновочных блоков (рис. 5.26).

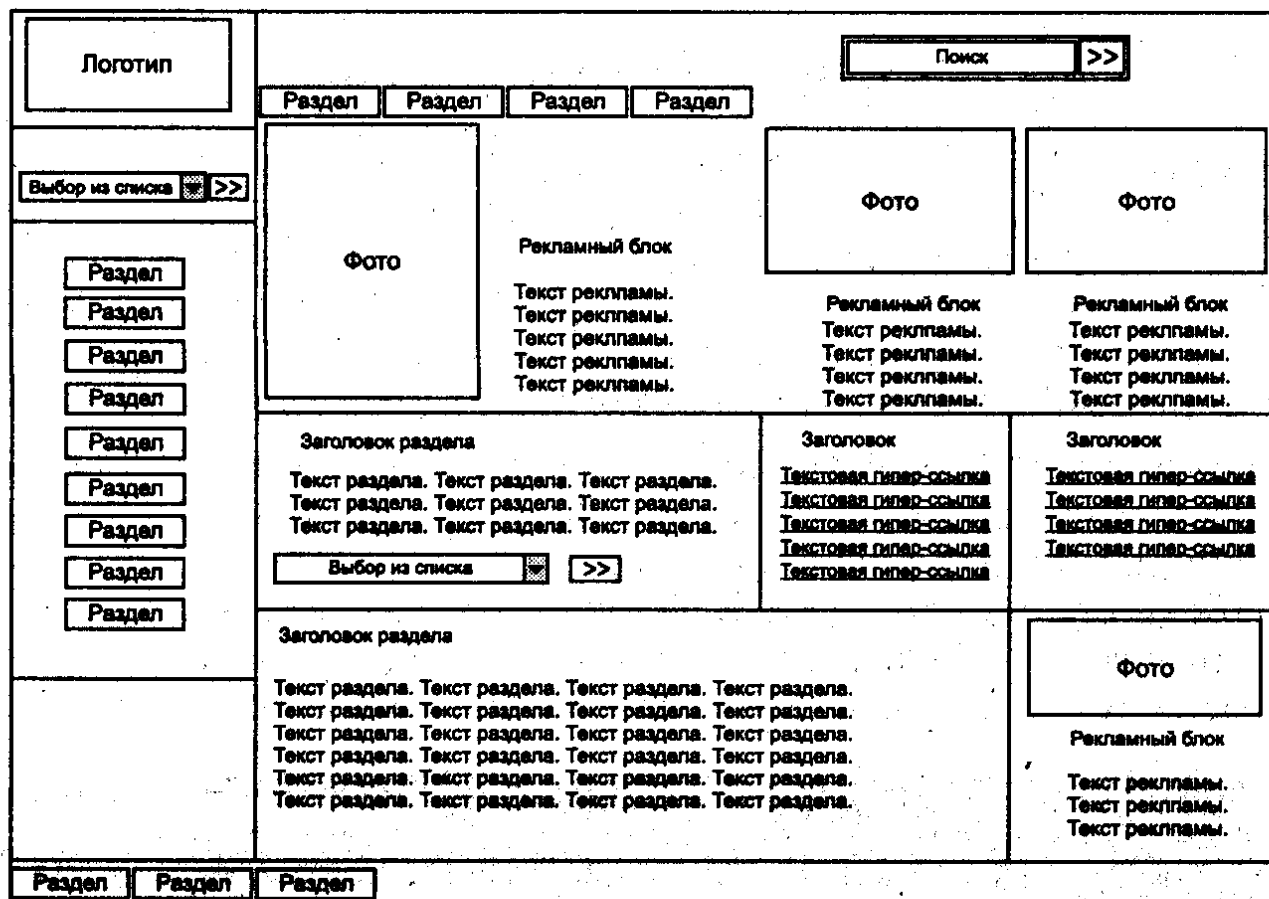


Рис. 5.26. А вот мы с вами разобрали IBM по полочкам...

## Резюме

1. Существуют пять смысловых уровней организации Web-сайта: внешний вид Web-сайта, уровень компоновки, уровень структуры, предметный уровень, уровень идеи.
2. Основные способы верстки: текстовая верстка, табличная верстка, верстка при помощи фреймов, флэш.
3. Верстку называют «резиновой», если размер (ширина/длина) основных элементов страницы задается в процентах.
4. Верстку называют жесткой (фиксированной), если размер основных элементов страницы задается в абсолютных величинах.
5. По назначению элементы страницы делятся на блоки: навигационные, информационные, сервисные, рекламные, дизайнерские.
6. К навигационным блокам относятся следующие блоки: «На главную страницу», поиска и быстрого перехода, блок гори-

горизонтального меню, блок вертикального меню, блок вторичной навигации, блок навигации по выборке, авторизации, «Подвал» (текстовые гиперссылки), блок навигационной строки.

7. К информационным блокам относятся блоки: содержания, текущей информации, «Раздел», изображений (галерея).
8. К сервисным блокам относятся блоки: «Выбор языка», «Пустой блок», «Версия для печати».
9. К рекламным блокам относятся блоки: «Название и слоган», «Копирайт», «Дизайнерское изображение», собственно рекламный блок.

# Элементы интерфейса

## 6.1. Обзор аппаратного обеспечения

К элементам интерфейса необходимо относить все, что используется для отдачи системе команд и получения от нее результата. Конечно же, этот список шире, чем просто кнопки, чек-боксы и поля ввода.

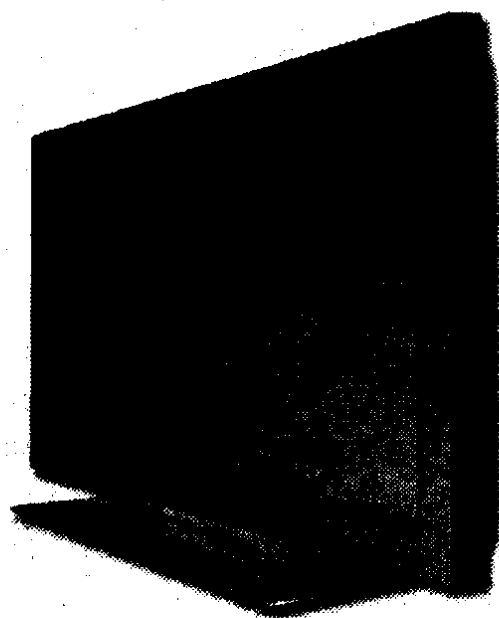
К основным элементам аппаратного обеспечения относят монитор, мышь, клавиатуру. Операционная система и браузер находятся уже ближе к интерфейсу самого Web-сайта и также вносят свою специфику. На аппаратное и программное обеспечение пользователей все же можно влиять, хоть это и не совсем очевидно. Использованию различных мониторов и разрешений экрана можно противопоставить «подгонку» содержимого вашего сайта под наиболее часто встречающиеся параметры. Для владельцев «непослушных» и достаточно древних на данный момент мышей можно предложить использование более крупных деталей и дополнительных элементов навигации. Различия в браузерах нивелируются различными таблицами стилей или обработкой JavaScript, а иногда и попросту фразой «Это лучше смотреть в IE 6.0».

### 6.1.1. Монитор

Монитор может быть цветным или черно-белым (хотя в современном мире второе — уже почти раритет), построен с использованием лучевой трубки или на жидких кристаллах — все это влияет на работу пользователя, его усталость и работоспособность. Безусловно, более комфортно будет себя чувствовать человек, сидящий за цветным плоским монитором с разрешением 1024×768, чем пользователь «антикварного» компьютера с выпуклыми черно-белыми 640×480 с пыльным защитным экраном (рис. 6.1).

Размеры мониторов колеблются в пределах от 14 до 21 дюйма. Навряд ли пользователь за 21-дюймовым монитором будет устанавливать 1024×768, наверняка разрешение будет побольше. Для

такого посетителя сайт с размерами, жестко заданными для 800×600, будет выглядеть как маленькое диалоговое окно в середине экрана. Цвета вашего сайта у всех пользователей также будут разными в зависимости от настроек яркости, контраста, цветности.



*Рис. 6.1. Плазменный монитор фирмы NEC*

### 6.1.2. Мышь

«Послушная» мышь тоже большое дело. Если ваша мышь-старушка с двумя неуверенными в себе кнопками передает системе двойной щелчок, как тройной, один «клик» сделать вообще почти невозможно, а внутри она грязная и не слушает вашу руку, то о каком удобстве использования программы может идти речь.

Что только не вытворяют с мышью (рис. 6.2). Ранее в отечественном компьютеростроении использовался большой, как утюг, и тяжелый «Манипулятор — Колобок». За ним появилась почти современная мышь. Две кнопки. Три кнопки. Потом для замены полос прокрутки на средней кнопке добавилось еще и колесико (скроллер). Крутишь его, не сдвигая мышь с места, и страница перемещается. Следующий шаг прогресса — кнопки горизонтального прокручивания. Далее появились мыши с «горячими» кнопками, например, аналогами бегунка в браузерах, дополнительными эргономичными и ортопедическими элементами.



Рис. 6.2. Новое поколение выбирает...

Со временем разработчики аппаратного обеспечения сведут на нет некоторые проблемы, с которыми борются специалисты по юзабилити, например, проблему горизонтального прокручивания или необходимости тянуться к меню браузера. А пока этого не произошло, все эти факторы необходимо учитывать при проектировании интерфейса.

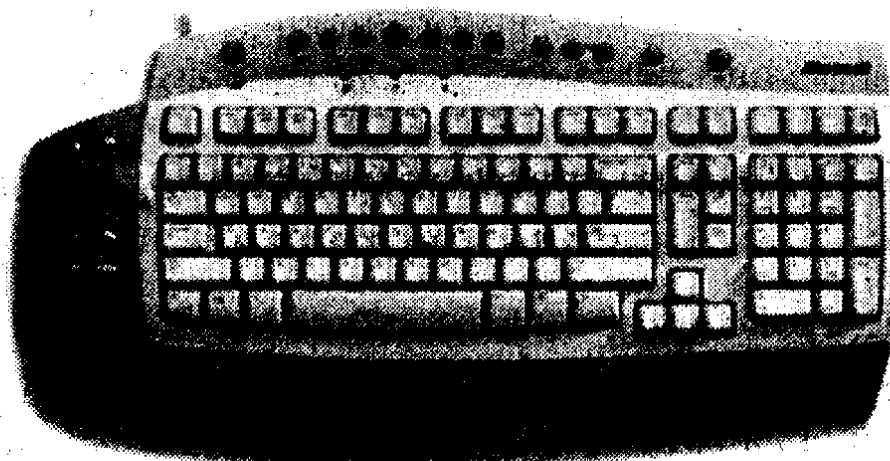
- ☑ Можно привести следующий пример влияния аппаратного обеспечения на программное. Если на мыши есть колесо прокрутки, то в MS Internet Explorer, удерживая клавишу Ctrl, можно изменять размеры шрифта в браузере, вместо того чтобы путешествовать по меню (Вид — Размер шрифта — Средний). Первым способом я переключаю размер шрифта в течение доли секунды, вторым — секунды за четыре.

### 6.1.3. Клавиатура

Вид и особенности клавиатуры влияют на количество ошибок пользователя при вводе данных. Их будет больше, если нажатие на клавиши сопряжено с большим усилием или, наоборот, клавиатура слишком чувствительна. Влияет она и на удобство при работе. Недаром изобретают все новые «примочки» для клавиатуры — поддержка для рук пользователя, «горячие» клавиши для управления звуком или запуском браузера, для «усыпления» компьютера, изменяют привычную плоскую форму, делят на две части (рис. 6.3).

Клавиатура влияет не только на удобство и скорость работы, но и «отражается» на здоровье пользователя, на степени его усталости.

Она также прошла весьма серьезный путь от практически плоских пластов с огромными клавишами до разделенной на две части эргономической клавиатуры с дополнительными клавишами, управляющими едва ли не половиной операционной системы.



*Рис. 6.3.* То, что вы видите, — клавиатура. Несколько отличается от привычной, согласитесь. Почему-то многие специалисты по интерфейсу забывают про стремительное развитие аппаратного обеспечения

Клавиатура находится «дальше» от мыши, когда пользователь работает с обычным сайтом, на котором нет необходимости вводить информацию. В таком случае он не захочет отрываться и переносить руку на клавиши, и наоборот, если он работает с клавиатурой, он постарается воспользоваться клавишами со стрелками вместо колеса прокрутки на сколь угодно «навороченной» мыши.

#### **6.1.4. Скорость соединения и мощность компьютера**

Скорость соединения в первую очередь влияет на скорость загрузки страниц и файлов из интернета. Нельзя рассчитывать на то, что у всех пользователей страницы грузятся так, как с локального диска, — недаром столько сказок сложено об оптимизации содержимого сайта и графики. Если ваше соединение по оптоволокну, то вы почти не будете успевать заметить пустой экран в момент, когда загружается следующая страница, но если у вас медлительный грустный модем, то вы успеете за это время поужинать.



Вывод: если сайт не оптимизирован под отечественные телефонные линии, некоторые посетители могут не дожидаться загрузки вашей флэш-заставки.

Мощность компьютера не очень сильно влияет на Web-интерфейс, т.к. пока всю нагрузку несут преимущественно серверы, а не клиентские машины. Параметры мощности в некоторой степени влияют на то, какая операционная система может быть установлена, и как следствие, на браузер и его версию.

## 6.2. Элементы взаимодействия системы с пользователем

### 6.2.1. Окно браузера

Все Web-страницы, с которыми работают пользователи, отображаются внутри окна браузера. В зависимости от компании-изготовителя и версии браузера существуют некоторые различия, однако общий принцип одинаков. Браузер может быть однооконным и многооконным. Например, в браузере Internet Explorer на каждую открытую страницу по одному окну приложения. В некоторых браузерах (Netscape Navigator, Opera) новые страницы могут открываться и в новом окне, и внутри текущего в окне-вкладке.

С очень небольшими различиями любое окно браузера обладает набором собственных элементов (рис. 6.4).

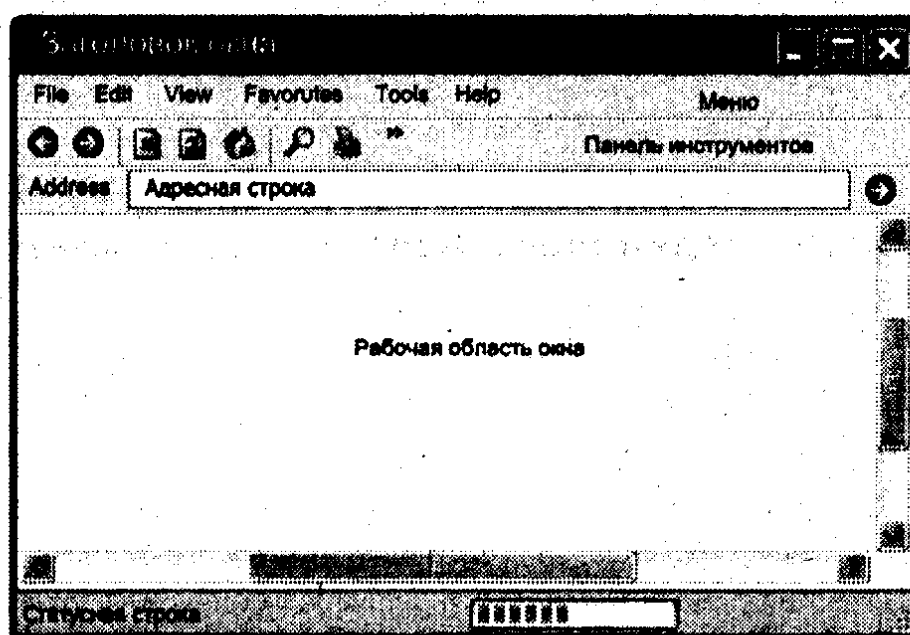


Рис. 6.4. Общий вид окна браузера MS Internet Explorer

Как и в большинстве других приложений, окна браузеров имеют меню с набором команд (я исключаю из рассмотрения так называемые текстовые браузеры). При помощи этих команд пользователь может влиять на страницу и на некоторые настройки операционной системы, устанавливать размер шрифта и кодировку, копировать содержимое, смотреть исходный код, настраивать способ соединения с интернетом, параметры безопасности и т.д.

- *Заголовок*

Каждое окно браузера имеет заголовок. Это полоска в самой верхней части окна, в которой отображается информация о странице (если, конечно, разработчик потрудился написать ее в теге `<title>`). Заголовок окна виден не только в верхней части окна, но и в панели задач (в MS Windows), по нему пользователи часто смотрят, что за страница у них открыта. Поэтому краткий и четкий заголовок — вещь весьма немаловажная.

- *Панель инструментов*

Управление некоторыми элементами браузера и страниц также может осуществляться при помощи панели инструментов, которая в любом браузере представляет собой вынесенные на верхний уровень команды в виде текста или пиктограмм (или того и другого вместе). Основное назначение панели инструментов — предоставлять пользователю быстрый доступ к некоторым функциям браузера, например, к истории посещенных страниц, обновлению страницы, печати текущей страницы и т.д.

Кроме всего этого, окно содержит рабочую область, т.е. место, где располагается Web-страница.

Перечисленные элементы окна браузера практически не влияют на работу с самими страницами и являются не очень значимыми с точки зрения интерфейса Web-приложений. Рассмотрим теперь элементы окна, особо важные для работы пользователя с конкретным сайтом.

- *Строка состояния*

Это узкая полоска внизу окна, которая обычно забита всякой ерундой — информацией от операционной системы, сообщениями от разработчика Web-сайта, что XML — это круто или что вам непременно нужно создать собственный баннер и т.д. Это все неважно. Главное достоинство строки состояния в том, что

она показывает (пусть даже не всегда правильно) степень загрузки страницы, а при работе с уже загруженными страницами при наведении на любую гиперссылку на Web-странице курсора мыши отображает значение этой гиперссылки. Тем самым строка состояния часто уберегает пользователей от захода на рекламные или не интересующие их страницы, которые нечистый на руку Web-мастер выдает за ценную информацию.

- *Адресная строка*

В адресной строке отображается URL (Universal Resource Locator), иначе говоря, путь к текущей странице. Внешнее представление этого адреса также влияет на работу пользователя с информацией. Например, если в нем есть не только вопросительные знаки, проценты и символы Unicode после основного доменного имени, а еще и имена каталогов, то он может подсказать посетителю, какова иерархия сайта. Кроме того, аккуратный скромный URL приятен с эстетической точки зрения и улучшает общее впечатление о вашем ресурсе.

- *Полоса прокрутки*

С ее помощью (или с помощью специального колесика мыши) пользователь пролистывает содержимое, которое не помещается на экране целиком. Полоса прокрутки — не очень удобный инструмент, но в чем-то все-таки полезный. В современных приложениях длина ползунка на полосе прокрутки соотносится в обратной пропорции с длиной документа, поэтому пользователи иногда посматривают на нее. Бывает, что на странице появляется несколько полос прокрутки, и тогда пользователь чувствует себя пилотом сложного вертолетного тренажера.

- *Курсор*

Его форма влияет на точность попадания в элементы Web-страницы и может быть разнообразной. Например, в MS Windows по умолчанию курсор выполнен в виде стрелки. При наведении на любую гиперссылку он изменяет свой вид на «руку». Если главную страницу сайта «особо креативный» дизайнер выполнил в виде изображения, в недрах которого прячется ссылка для входа, то обнаружить ее удастся, исследуя местность до тех пор, пока не появится курсор-«рука».

На этом элементы интерфейса браузера заканчиваются. Переходим к элементам самих Web-страниц.

### 6.2.2. Формы

Под *формой* понимают и внешнее представление определенной страницы (например, форма регистрации пользователя), и часть внутреннего содержимого страницы (HTML-тег). Формы используются для получения информации от пользователя и ее отправки на сервер (или обработки при помощи JavaScript на клиентском рабочем месте). Формой является все, что заключено в специальный HTML-тег `<form>`. Форма содержит поля ввода (выбор из списка, текстовое поле, чек-боксы и радио-кнопки и т.д.). Границы формы, т.е. начало и конец тега `<form>`, пользователь на странице не видит.

В общем случае работа с формой выглядит так: пользователь заполняет какие-либо поля ввода, нажимает на кнопку, а введенная информация одной длинной строкой отправляется на сервер, где будет обработана. Формы используются для принятия данных от пользователя и их отображения.

### 6.2.3. Апплеты

*Апплеты* — это внешние программные сущности, встраиваемые в любое место страницы, так же как и флэш-файлы (рис. 6.5). Апплет создается при помощи языка Java и представляет собой программный код, скомпилированный в файл с расширением `.class`. На HTML-страницу апплет загружается при помощи специального тега `<applet>`. Для нормальной работы апплета необходима поддержка языка Java, что осуществлено практически во всех современных браузерах.



Язык Java является весьма развитым языком, позволяющим осуществлять логику взаимодействия системы с пользователем практически любой сложности.

Апплеты, как правило, загружаются медленнее обычных страниц, но зато могут предоставлять большую и сложную функциональность за счет использования высокоуровневого языка программирования Java. По внешнему виду чаще всего они отличаются от привычного Web-интерфейса и напоминают обычные приложения операционных систем. Несмотря на то что иногда пользователи не замечают различий между обычной HTML-страницей и стра-

ницей с апплетом, необходимость специального программного обеспечения на компьютере пользователя не позволяет широко применять технологию апплетов в Web. Это связано с тем, что не во всех операционных системах и не во всех браузерах существует поддержка Java. Посетителю приходится предлагать то, что для него выглядит как «Скачайте нечто непонятное с сайта какой-то компании Sun Microsystems» (о которой он, возможно, вообще никогда не слышал). К тому же, это «нечто» — файл далеко не маленького размера.

Достоинством технологии апплетов является повышенная безопасность в работе компонентов и их устойчивость к несанкционированному доступу, что обеспечивается конструкцией языка Java.

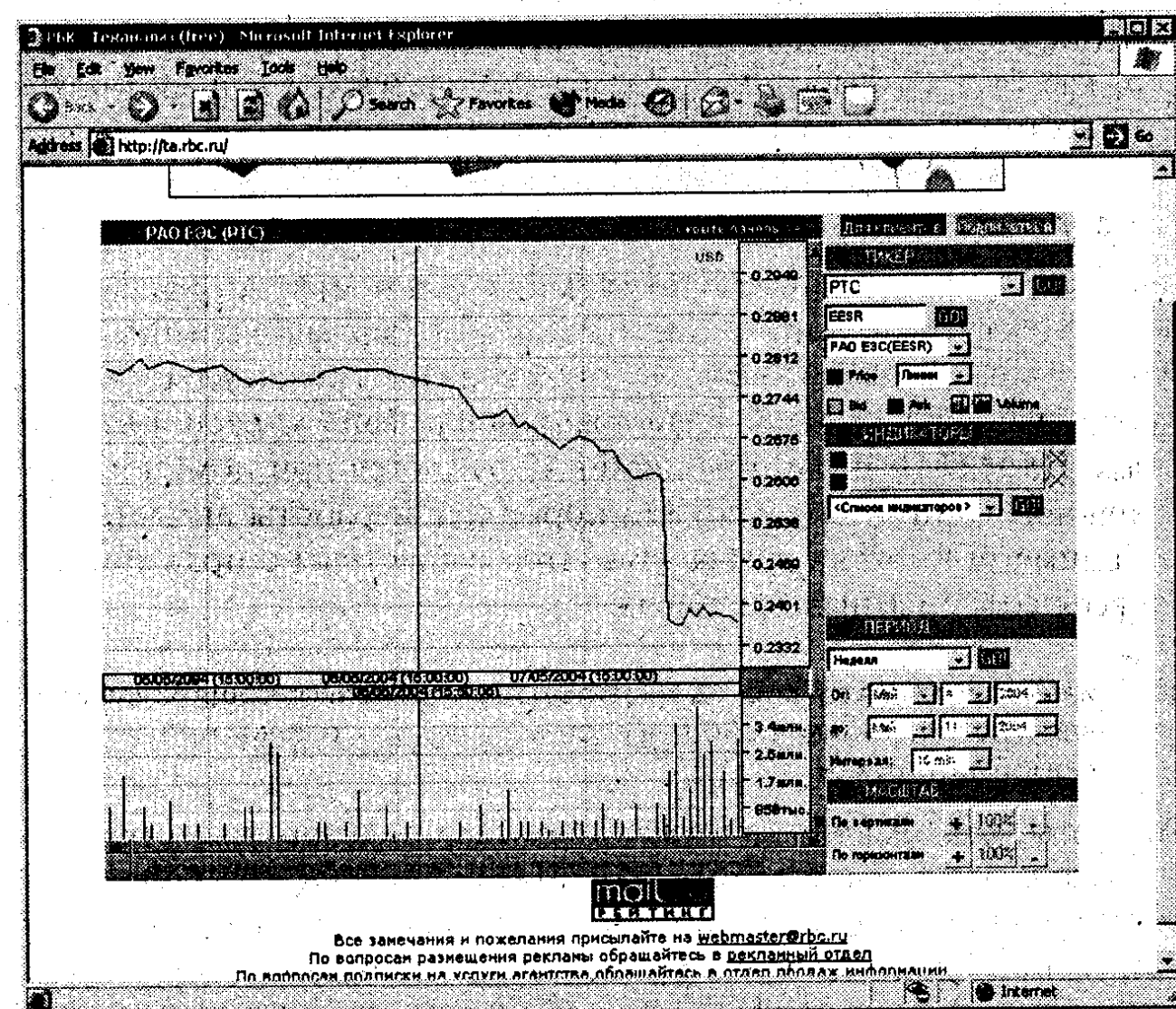


Рис. 6.5. Апплет для интерактивного построения финансовых и статистических отчетов на сайте РосБизнесКонсалтинга

#### 6.2.4. Диалоговые окна

В программном обеспечении под *диалоговым окном* (диалогом) понимают сообщение системы пользователю о некотором событии с запросом на дальнейшие действия. Примером диалогового окна может служить окно отправки на печать текста или сообщение о том, что свободного места на жестком диске нет и файл сохранить невозможно.

Первым делом пользователь сталкивается с необходимостью настройки и подключения браузера ко Всемирной паутине. Изменение определенных требований и настроек нам неподвластно равно как и управление автоматической подстановкой пароля адреса, вопросами браузера о способе его поведения (в этом смысле наиболее любопытным мне кажется норвежская Opera). Не управляем мы изначально и предупреждениями браузера о том, что он собрался разослать информацию, введенную вами о ужас, в «незашифрованном» виде по всему интернету.

Некоторые диалоги мы в силах предотвратить, лишь изменив используемые технологии. Например, на сайтах могут использоваться Active X-компоненты (к счастью, не очень часто), о которых браузер сразу же предупреждает, т.к. они могут представлять серьезную угрозу безопасности пользовательского компьютера. Если разработчику таких сайтов отказаться от использования технологии Active X, по крайней мере в интернете, пользователям наверняка станет легче жить.

Использование стандартных диалоговых окон браузера, вызываемых при помощи языка JavaScript, также мне кажется нежелательным. Во-первых, нарушается концептуальная целостность и единообразие внешнего вида интерфейса обычных приложений и Web-приложений. А во-вторых, они сильно отвлекают.

Существуют еще более навязчивые окна, которые даже нельзя с чистой совестью назвать диалогами. Это так называемые окна отладки сценариев, т.е., по большому счету, сообщения об ошибках JavaScript. Если ошибок много, работать с содержимым становится практически невозможно. Когда вы закроете пятнадцатое сообщение о том, что «addItem не является объектом», вы закроете и саму страницу. И правильно сделаете. Поэтому пользователи часто отключают отладку сценариев в настройках браузера. Как славно, что это можно сделать.

## 6.3. Элементы навигации

Представьте себе, что вы оказались одни в лесу без компаса в облачный день. Лес сухой, мха на деревьях нет, а ветки густо растут со всех сторон и торчат везде одинаково. Вы забрались далеко от шоссе, и шума машин вам не слышно. Жуткая картина. Все «бойскаутские» приемы не работают, и если вы не обладаете сильнейшей интуицией и вы не ясновидящий, то вам будет очень трудно выбраться. У вас отсутствуют средства навигации, вы не знаете, где вы и куда идти.

Когда речь идет не о дикой природе, а о Web-сайтах, то заблудиться, к счастью, вы можете только виртуально. Однако при плохо организованной навигации пользователю очень трудно передвигаться внутри сайта.

Что такое навигация Web-сайта? Навигация — это именно тот инструмент (набор инструментов), с помощью которого пользователь перемещается по Web-сайту. В простонародье «это все, на что можно нажать, чтобы куда-нибудь перейти». Однако это не совсем так: в интерфейсе различных программных систем существуют одни и те же элементы, которые служат для разных целей. Например, изображения могут быть предназначены и для украшения сайта, и для перехода в какой-нибудь раздел, подчеркнутый текст («клинический случай» для таких элементов) может использоваться и для выделения важной части содержания, и как обозначение гиперссылки. Рассматривая элементы навигации, мы будем говорить об их сути, а не о способе реализации с точки зрения HTML или JavaScript.

Итак, **навигация** — это набор способов перехода между страницами и внутри них, обеспечиваемый в основном элементами содержимого страниц, а также средствами браузера, операционной системы и аппаратного обеспечения.

В абсолютном большинстве случаев Web-сайты разбиты на разделы. На домашней страничке увлеченного любителя их может быть два-три, на поисковом мега-портале — тысячи. Несмотря на такие различия, основные способы устройства навигации могут быть схожими.

Одним из элементов навигации является *ссылка на главную страницу* сайта (рис. 6.6). Согласно некоторым исследованиям, этот элемент встречается практически на 100 % сайтов.



Рис. 6.6. Переход на главную страницу на поисковом портале Google возможен с любой страницы

Теперь об основных разделах сайта. Для того чтобы показать посетителю список основных разделов, можно использовать *меню в верхней части страницы и боковое меню*.

Факторы, которые определяют выбор того или иного вида меню

- верхнее меню занимает мало места на странице;
- боковое меню «откусывает» большой кусок страницы, а при наличии длинных страниц на несколько экранов это драгоценное место вообще пустует;
- в верхнем меню ограничена длина наименований разделов, иначе придется переносить их на следующую строку, а тут уж никакой аккуратности не жди;
- в боковом меню можно поместить гораздо более распространенные наименования разделов.

Иногда два способа расположения меню совмещают, однако в таком случае могут возникнуть проблемы с удобством их использования.

Нежелательно располагать часть ссылок на основные разделы сайта в боковом меню, а другую в верхнем, даже если вы считаете, что по смыслу эти части совершенно обособлены и одни из них более важны, другие — менее. Посетитель должен четко знать



куда смотреть в поисках нужного раздела, иначе у него рассеивается фокус внимания и он будет искать ссылку гораздо дольше.

Также нежелательно дублировать ссылки в верхнем и в боковом меню, как это часто делают, т.к. нарушается принцип монотонности интерфейса (рис. 6.7).

☑ Одна моя знакомая всегда предлагала по три-четыре способа сделать какое-нибудь действие в программе, в которой она была специалистом. Ее никто не понимал. Все хотели конкретное указание: нажми сюда, выбери это, установи такую «птицу»... А она говорила: «Ты можешь сделать так, а можешь вот так, ну а если захочешь через меню, то надо сделать вот так, это почти как так, но не так...»

Если дать пользователю три-четыре варианта, то он либо достаточно долгое время никуда не уйдет со страницы, потому что не будет знать, как это сделать, либо уйдет прочь с вашего сайта, либо будет пробовать каждый вариант и смотреть, куда его «выбросит».

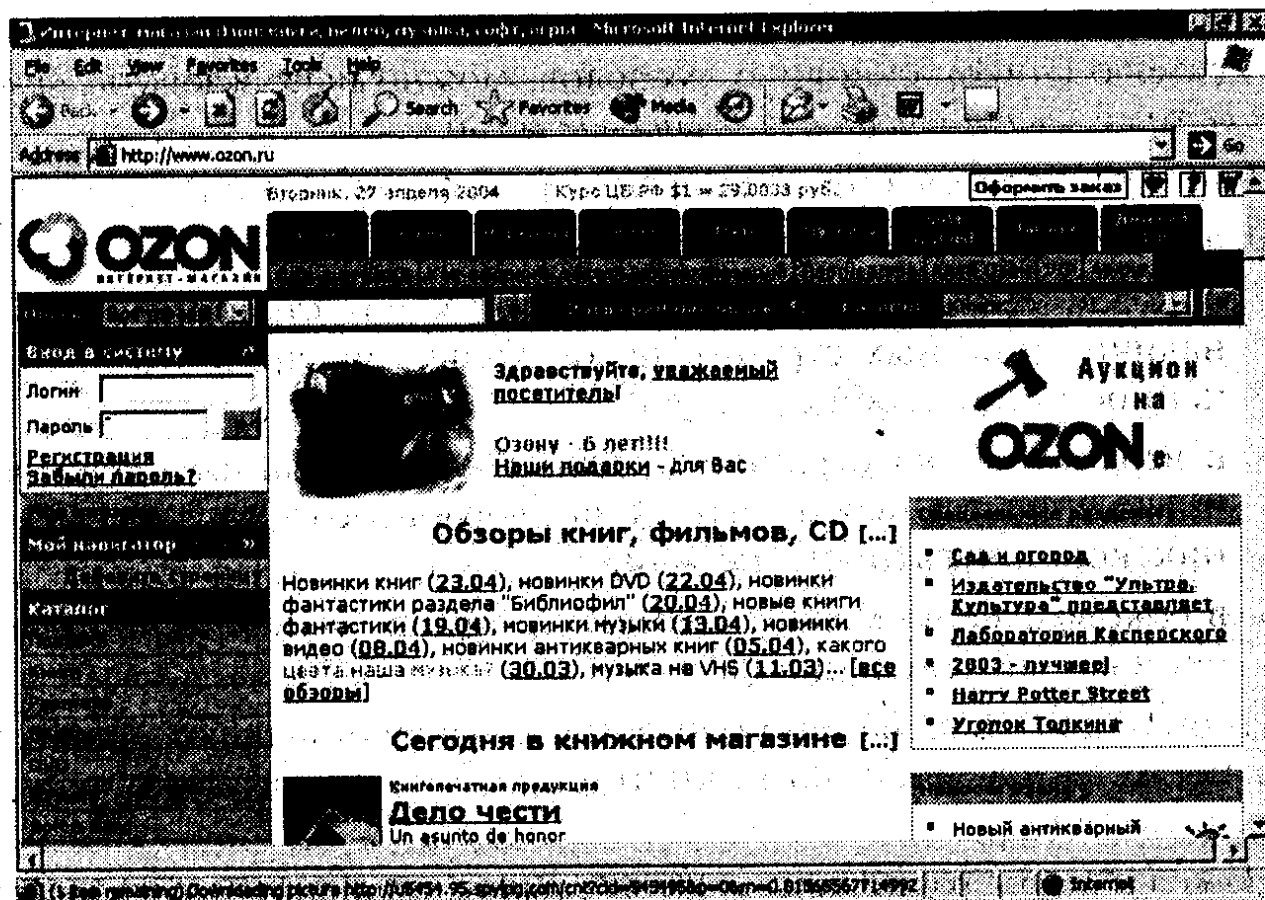


Рис. 6.7. Переход к разделу Книги встречается на одной странице три раза (данная картинка, к счастью, уже несколько устарела — разработчики ozon.ru решили многие проблемы интерфейса)

Нет четких сведений о том, какое именно меню пользователи считают основным. Если ваш посетитель уверен, что список основных разделов всегда размещен сверху, а вы считаете, что это не так, то вы замедлите работу посетителя и он станет больше отвлекаться. «Так, посмотрим, какие услуги оказывает эта компания... О компании... Поддержка... Карта сайта... Не понял, где?... А... У них же все в левом меню...»

Также следует решить вопрос о том, каков должен быть баланс горизонтальных и вертикальных уровней иерархии, какие уровни необходимо показывать в списке основных разделов. Слишком общие наименования разделов могут стать загадкой для посетителя, а чересчур много основных разделов могут запутать и затруднить поиск действительно важной информации. Естественно, где-то между «мало» информации и «много» есть золотая середина. В случае, если иерархия многоуровневая и разделов сайта много, пользователю легче найти нужный раздел на странице, чем искать его в подразделах.

Существуют и дополнительные способы перемещения внутри сайта, например вторичная навигация, подвал, быстрый переход и др.

- *Вторичная навигация*

Не является основным способом перемещения внутри сайта, однако имеет достаточно большое значение с точки зрения маркетинга (рис. 6.8). В конце концов, редко кто создает сайты без обратной связи. Люди, как правило, хотят найти единомышленников, клиентов, партнеров, покупателей. Для того чтобы посетитель сайта мог получить необходимую информацию о вас, используется вторичная навигация — разделы О компании, Контакты, Наши координаты, Служба поддержки. Вторичная навигация, если она представлена на сайте, должна быть доступна на каждой странице.

- *«Подвал»*

Это нижняя часть страницы, где располагают список основных разделов сайта и вторичную навигацию в виде обыкновенных, четких, хорошо читаемых текстовых гиперссылок, иногда включают и описание авторских прав. Назначение подвала может быть разным, однако, в первую очередь, он предназначен для обеспечения перехода в другой раздел, когда посетитель находится внизу страницы. После прочтения интересующей инфор-

мации пользователь готов двигаться далее — вот и надо дать ему эту возможность, когда его фокус внимания находится именно внизу страницы.

• Быстрый переход

Подобная функция часто применяется в технике, например, быстрая запись в видеомаягнитофоне или прямое воспроизведение в музыкальных центрах на несколько компакт-дисков.



Рис. 6.8. Вторичная навигация на сайте компании IBM расположена в левом нижнем углу, поэтому она не отвлекает пользователя от основных разделов, но обнаруживается легко

В Web-интерфейсе этот способ навигации чаще всего реализуют при помощи выпадающего списка с набором разделов и страниц сайта, к которым пользователь может перейти независимо от того, на каком уровне иерархии он находится (рис. 6.9). Если того требует специфика вашего сайта, быстрый переход может переместить посетителя с главной страницы сразу на пять-шесть уровней вниз.

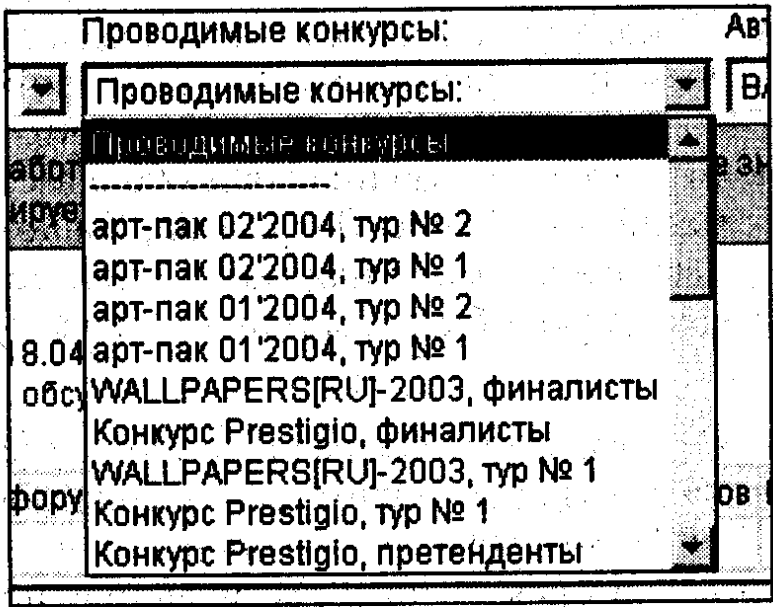


Рис. 6.9. Быстрый переход к прошедшим конкурсам графики на сайте дизайнеров wallpapers.ru

- *Главное меню* (рис. 6.10)

Этот элемент предназначен для перехода к нужному разделу. Он является достаточно экзотическим для интернета, хотя довольно часто используется для интранет-приложений. По существу это представленные пользователям варианты дальнейших действий.

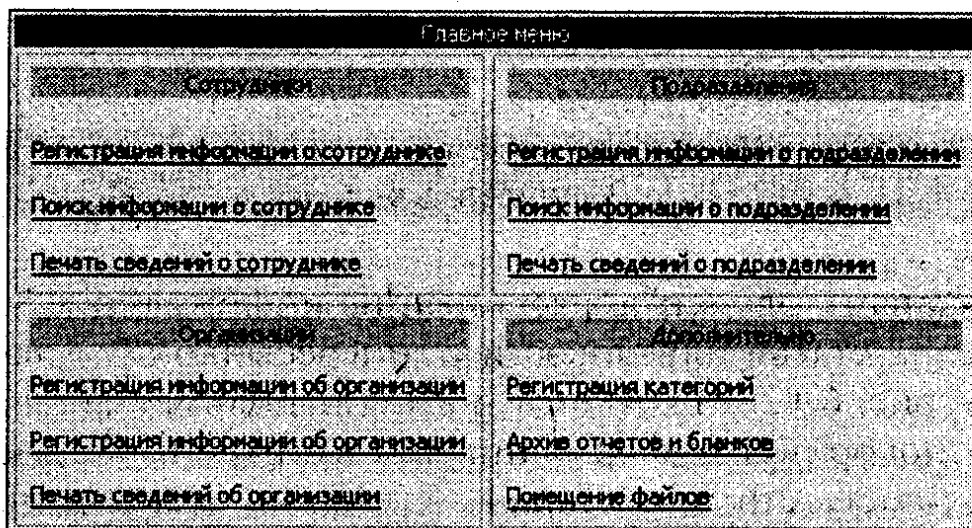


Рис. 6.10. Элемент навигации «Главное меню»

Как правило, сам по себе этот навигационный элемент является несостоятельным. Его можно использовать только в совокупности с другими способами навигации. Автономно он может быть использован только один раз на верхнем уровне сайта.

Если список разделов велик, а показывать его нежелательно, либо хочется вывести побольше новостей, но не загромождать место на странице, то можно использовать элемент Больше/Подробнее (в зависимости от контекста) или, говоря научно, элемент расширения информации. В некоторых источниках его называют «завлекалочка». Звучит, согласитесь, не очень серьезно. Если сравнить этот элемент со зданием, этажами и офисами, то он бы выглядел, как дверь с надписью «А тут есть еще кабинеты».

Если необходимо расширить список отображаемых разделов сайта, то следует использовать гиперссылки Еще, Больше или Все разделы (рис. 6.11). В этом случае пользователь сознательно идет на увеличение количества ссылок на странице и знает почти наверняка, что длина страницы увеличится, но поиск нужного раздела того стоит.

## → Software

- Application & Integration Servers
- Operating Systems
- Application Development
- » **See All**

Рис. 6.11. Вместо десятка ссылок на сайте корпорации Sun Microsystems в разделе Программное обеспечение приведено всего четыре, одна из которых Посмотреть все

Другой вариант элемента расширения информации — вывод части текста, статьи или новости с гиперссылкой Подробнее или Читать все (рис. 6.12). Часто этот подход используется для увеличения разнообразия информации, для устранения эффекта загроможденности страницы и для облегчения восприятия.

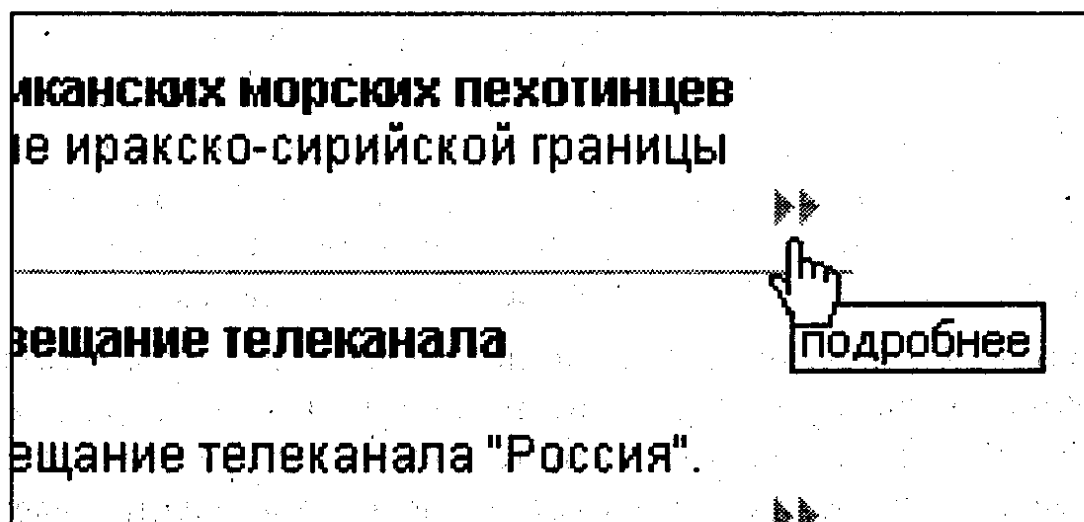


Рис. 6.12. Элемент расширения информации в новостях на gazeta.ru

Средства перехода к подразделам сайта можно организовать разными способами.

- *Выпадающее меню*

Этот способ широко используется в интерфейсах обычных приложений — графических и текстовых редакторов, браузеров, операционных систем. Список подразделов появляется из на-

именования основного раздела при наведении на него курсор или щелчке мыши и исчезает по такому же принципу (рис. 6.13). Меню имеет ограниченную ширину, поэтому названия пунктов должны быть краткими и понятными.

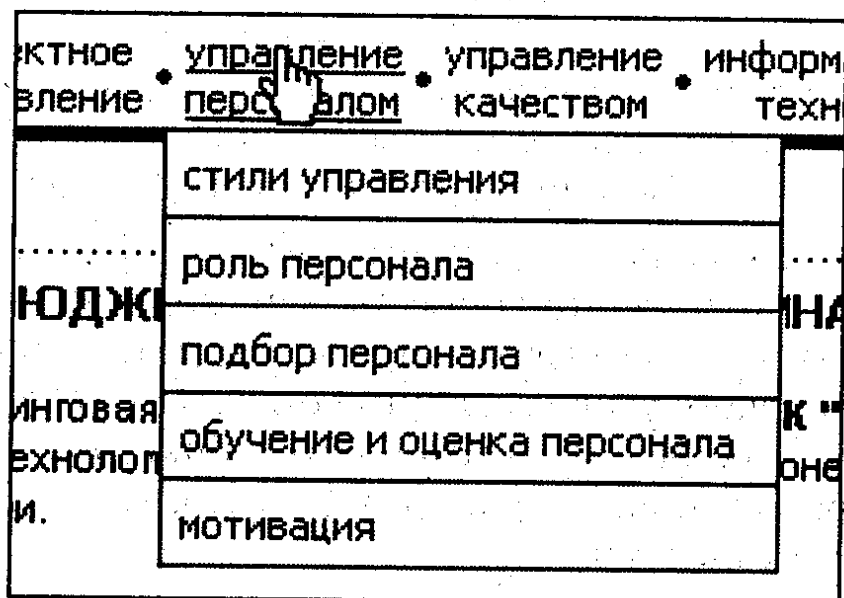


Рис. 6.13. Классический пример выпадающего меню на iteam.ru

Выпадающее меню чаще всего находится в верхней части страницы, поэтому требуется время, чтобы подвести к нему курсор. Кроме этого, скорость реакции выпадающего меню в Web-интерфейсе может быть низкой, посетитель может не понять, что название раздела есть начало выпадающего меню, т.е. что такой элемент навигации вообще присутствует на странице. Для сообщения о том, что данный элемент является корневым в выпадающем меню, должны быть использованы дополнительные средства, например, пиктограмма с вертикальной или горизонтальной стрелкой перед (после) названия раздела меню. Достоинство выпадающего меню в том, что оно занимает намного меньше места, чем весь список его элементов, расположенных непосредственно на странице. Выпадающее меню, подобно быстрому переходу, сокращает количество шагов до нужного уровня иерархии.

#### • Дерево

Это визуально отображенная иерархическая структура, т.е. структура с одним или несколькими корневыми элементами, каждый

из которых содержит в себе некоторое количество более мелких. Дерево обладает лишь двумя типами элементов, условно их можно назвать «узел» и «лист». В древовидных структурах эти типы могут быть расширены и содержать в себе множество объектов разного вида как среди узлов, так и среди листьев. Узел может содержать более мелкие узлы и листья либо быть пустым. Лист не содержит вложенных элементов, это крайний минимальный элемент дерева. С точки зрения файловой системы лист подобен файлу, внутри которого не может быть ни папок, ни иных файлов.

Примером древовидной структуры может служить любая организация. Например, внутри организации (корневого узла) выделяются подразделения (узлы), внутри подразделений — отделы (узлы более низкого уровня), внутри отделов — сотрудники (листья). В то же время могут быть и сотрудники, находящиеся на уровне подразделения, например, менеджер, управляющий тремя смежными отделами.

Практически все современные операционные системы поддерживают эту абстракцию для отображения файловой структуры (рис. 6.14).

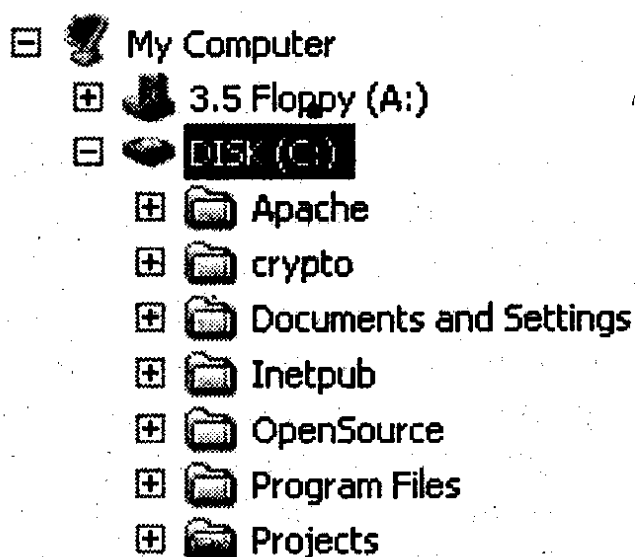


Рис. 6.14. Древовидная организация материала

Узлы могут быть в закрытом и в раскрытом состоянии, что показывается различными способами (например, открытая и закрытая книга, знаки «+» и «-», стрелки вправо (влево) либо вниз). У листьев же таких состояний нет вообще.

Частным случаем древовидной структуры можно считать многоуровневый список (рис. 6.15).

- First list item
- Second list item
  - First list item
  - Second list item
  - Third list item
- Third list item

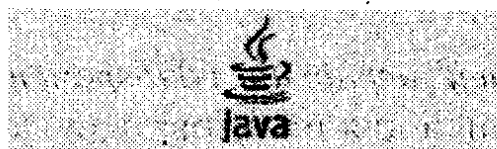
Рис. 6.15. Многоуровневый список — стандартный элемент языка разметки HTML

✓ Как выпадающее меню, так и дерево являются нетипичными для Web-интерфейса, потому что их реализация не всегда тривиальна. Во-первых, в силу различий в отображении одной и той же страницы разными браузерами (динамическая древовидная структура, прекрасно отображаемая в MS Internet Explorer, может совершенно «рассыпаться» в Netscape Navigator или Opera). Во-вторых, в силу относительной сложности в реализации. В-третьих, как правило, подобные элементы, а в особенности древовидные структуры, очень медленно перечитываются, и приходится подолгу ждать раскрытия узла дерева.

#### • Навигационная строка

Это набор ссылок, отражающий все пройденные пользователем уровни сайта (рис. 6.16). Гиперссылки ведут на ключевые в иерархии страницы сайта и разделяются символами «:», «|», «/», «\» и т.п.

developers.sun.com



Developers Home > Developer Forums > Java Technology Forums >

Рис. 6.16. Навигационная строка на сайте Sun Microsystems

Последний уровень иерархии в навигационной строке — текущий для пользователя — отображается простым текстом. Уровни



могут начинаться с первого — главной страницы сайта — и вести до последнего, не имеющего вложений. Такой способ построения навигации позволяет пользователю вернуться на несколько шагов, вплоть до самого начала. Кроме того, можно увидеть, какова иерархия сайта или как организован бизнес-процесс, в контексте которого находится пользователь. Естественно, такой способ навигации целесообразно использовать, когда вложенность уровней достаточно большая и сложная.

☑ Кстати говоря, существует известная, но часто остающаяся незамеченной аналогия со структурой подкаталогов сайта, которая отображается в адресной строке браузера. Опытные пользователи давно научились перемещаться по содержимому, удаляя часть адреса из адресной строки. Например, адрес некоторой страницы может выглядеть так: `http://mysite.net/category/interface/usability/errors/index.php`. Пользователь, видя структуру сайта и зная, что ему необходима информация об элементах интерфейса, может убрать из адреса определенную часть и по адресу `http://mysite.net/category/interface/` сразу перейти в раздел Интерфейс.

#### • Каталог

Каталог обычно организуется на сайтах с большим количеством информации. Это могут быть и просто ссылки, и статьи, и документация. Каталог сайта в большой степени соответствует каталогу в файловой системе: есть один или несколько самых верхних уровней (корневые каталоги), в каждом из них могут располагаться подуровни (подкаталоги) и элементы каталога, например, ссылки (файлы). Попадая в подраздел, вы видите его структуру, элементы, а также можете вернуться на предыдущий уровень. Существует несколько основных методов организации каталогов.

*Метод отображения двух уровней иерархии одновременно.* Это очень распространенный и удобный для пользователя метод. Он основан на сокращении количества шагов при поиске информации и расширенном отображении структуры. Его суть в том, что два уровня иерархии показываются совместно: с наименованием раздела частично или полностью располагаются наименования подразделов, которые служат и элементом навигации, и подсказкой о том, что содержится в данном разделе (рис. 6.17).

**Arts**Movies, Music, Television,...**Home**Consumers, Homeowners, Family,...**Business**Industries, Finance, Jobs,...**Kids and Teens**Computers, Entertainment, School,...**Computers**Hardware, Internet, Software,...**News**Media, Newspapers, Current Events,...

Рис. 6.17. Второй и третий уровень иерархии каталога на google.com

**Метод последовательного уточнения.** Существует в самых разных вариантах. Отличие его от предыдущих методов в том, что он предназначен не для улучшения удобства отображения информации, а для ведения пользователя к цели сквозь структуру сайта. Он может быть применен как к уточнению внешнего вида и «начинки» Web-сайта, к поисковым операциям пользователя, так и к способу навигации по сайту. Этот метод напоминает игру «горячо — холодно», в которую часто играют в детстве. Подобная игра с пользователем при поиске или просто навигации по сайту позволяет, во-первых, достаточно уверенно передвигаться по информации, а во-вторых, пользователи непроизвольно знакомятся с тем, как организована структура вашего сайта, и с каждым разом работают быстрее и качественнее (рис. 6.18).

## **Каталог / Компьютеры и связь / Интернет**

- Услуги доступа и связи (2257)
- Электронная почта (50)
- Отправка сообщений (60)
- Каталоги (339)
- Поисковые системы (57)
- Сайты, адаптированные для КПК (34)
- Универсальное (40)

См. также: Софт для интернета, Вирусы и антивирусы

Постепенное приближение к цели поиска с уведомлением о количестве ресурсов

Метод последовательного уточнения работает примерно так: человек ищет информацию о друге Винни-Пуха Пятачке на детском сайте. Он выбирает сначала раздел Современные сказки, попадая внутрь, выбирает Винни-Пух и все, все, все..., а затем Герои и Пятачок.

А теперь о дополнительных удобствах. Вспомните, как устроена книга по технической дисциплине. У такой книги всегда есть две вещи, с помощью которых читатель «путешествует» по ней, — развернутое содержание и предметный указатель. Во Всемирной паутине есть аналогичные элементы — карта и индекс сайта.

Хотя оба элемента навигации упрощают работу посетителей, они почему-то используются не очень часто, едва ли не игнорируются многими пользователями. Одной из причин является то, что разработчик «стесняется» этих элементов, не делает их «видимыми», считая, что все равно слишком малое количество посетителей обратятся к ним.

Данные два элемента относятся ко вторичной навигации и при правильной организации намного ускоряют работу с информацией.

Карта и индекс сайта особо важны для работы индексирующих роботов, поскольку охватывают все наполнение сайта. При их наличии весь сайт будет качественно проиндексирован.

- *Карта сайта*

Как правило, она представляет собой развернутый список всех разделов и вложенных подразделов с необходимой глубиной (рис. 6.19). Вложенность может быть два, три, четыре уровня. Количество этих уровней определяется тем, насколько применима данная карта. Представьте себе величину и глубину карты для каталогов любого поискового портала или для интернет-магазина с тысячей товаров и услуг. Это немыслимый объем информации. Никто не будет делать карту для таких крупных систем — слишком велик объем информации. В другом случае, если на сайте всего три раздела, а сам он состоит из пяти—семи страниц, тоже нецелесообразно создавать карту, поскольку она будет аналогична основному меню. Карта, как правило, используется на сайтах средней сложности. Она может быть создана и как обычные текстовые ссылки, и как сложный графический объект. Лично мне кажется предпочтительным использование обычных текстовых гиперссылок.

## Карта сайта

### Регистрация

- Выбор программы
- WM Keeper Classic
  - Рекомендации по подключению WM Keeper Classic через Proxy-сервер
  - Подробное описание интерфейса WM Keeper Classic
  - История WM Keeper Classic
  - Описание WM Keeper Phone Plugin
  - Downloads
- WM Keeper Light
  - демонстрация экспорта сертификата
  - демонстрация импорта сертификата
  - о цифровых подписях и цифровых сертификатах
  - пример настроек браузера для использования SSL
- Вопросы и ответы

### О системе

- Описание
- Технология
- О сервисах
- Соглашения
  - Соглашение о трансфере имущественных прав
  - Соглашение о борьбе с незаконной торговлей

*Рис. 6.19. Карта сайта в виде развернутой структуры из текстовых гиперссылок на webmoney.ru*

### • Индекс сайта

Представляет собой ключевые слова и фразы, позволяющие быстро найти необходимую информацию. Это как бы набор характерных запросов с уже подготовленными ссылками на лучшие страницы по данному вопросу (рис. 6.20). Для создания индекса необходимо очень хорошо представлять, что именно будут искать посетители и какая информация является наиболее важной.

Индекс оформляется в виде набора ссылок, отсортированных в алфавитном порядке. Иногда для удобства вверху сайта располагают сам алфавит для быстрого перехода к нужной букве. Индекс сайта создавать и поддерживать гораздо тяжелее, чем карту.

### • Якоря и ссылки Наверх

Они используются для перемещения по одной и той же странице. Якорь представляет собой некоторое место на странице, с кото-

A B C D E F G H I J M N P R S T U W

---

**A** Arrays & Intelligent Storage Server Options

Back to Top

---

**B** Blade Servers  
Boards  
Board Platforms  
Boot Disk Solutions

Back to Top

---

**C** Chip Sets & Logic

Рис. 6.20. Индекс сайта на sun.com

рым связывается гиперссылка. При нажатии на такую ссылку страница прокручивается к соответствующему якорю, и место, где он закреплен, располагается в верхней части рабочей области. Гиперссылка **Наверх** (top) перемещает пользователя в самое начало страницы (рис. 6.21). Эти два элемента чаще всего используются тогда, когда страница слишком длинная и легче не заставлять пользователя прокручивать ее, а использовать для этого браузер.

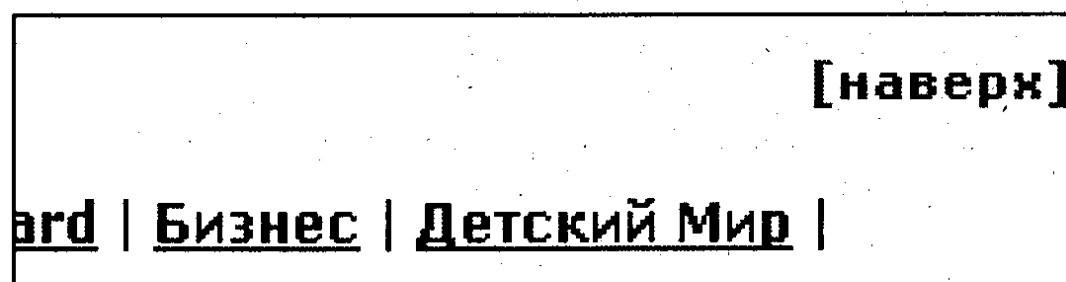


Рис. 6.21. Ссылка **Наверх** на ozon.ru оформлена как текст

Альтернативой ссылке **Наверх** может быть дублирование навигации (рис. 6.22). Этот вариант подобен подвалу, однако дублируются не ссылки на основные разделы, а элементы навигации — кнопки, исполняемые гиперссылки (Отправить форму, Удалить заказ и т.п.).

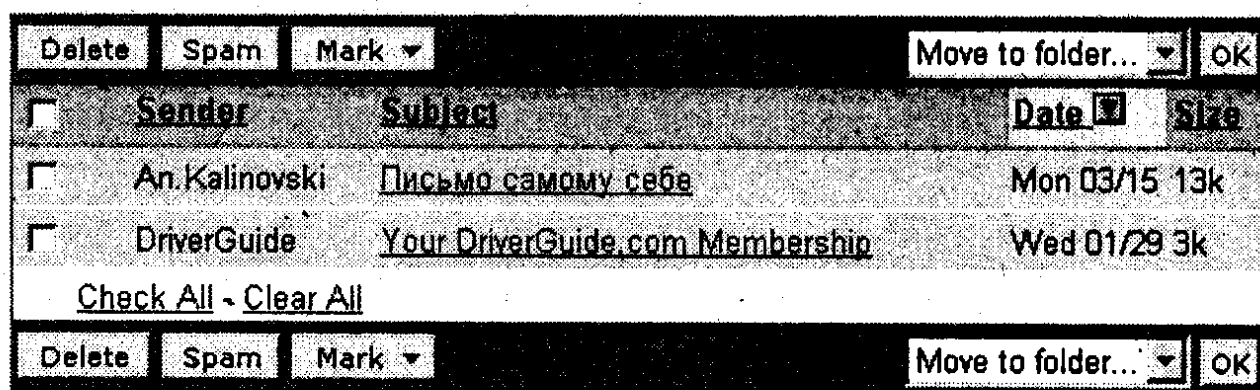


Рис. 6.22. Дублирование навигации — исключение из правила о том, что нельзя предлагать два элемента, ведущие к одному и тому же действию системы

## 6.4. Поиск

Многие посетители даже и не подумают разбираться в навигации и том, как устроена информация на сложном и объемном сайте, если разработчики обеспечили их функцией Поиск. Нередко они обращаются к ней почти сразу же после появления на вашем сайте, потому что они привыкли к проблемам с навигацией и не желают зря возиться. С другой стороны, если модель навигации не совсем понятна и запутанна, то единственное, что может удержать посетителей на сайте, — это также функция Поиск. Особенно она актуальна на сайтах, содержащих большое количество информации.

Важность поискового аппарата поясню на примере. При помощи поисковых серверов я ишу статью об использовании во Всемирной паутине платформы «.NET». Когда я по одной из ссылок попадаю на страницу, где, по мнению поискового робота, содержится сочетание «Microsoft.NET in WWW», я вижу огромный список статей с аннотациями к ним. Найти нужную мне статью глазами среди такого изобилия информации просто невозможно. Тогда я использую поиск средствами браузера и моментально нахожу нужную строчку. Аналогичная ситуация и с поиском по сайту, осуществляемым с самой Web-страницы.

Как должна выглядеть форма поиска (рис. 6.23)?

Поиск по сайту в большинстве случаев осуществляется в несколько шагов и состоит из следующих элементов.



Рис. 6.23. Форма поиска на сайте компании Sun Microsystems

#### Первый шаг. Начальный поиск:

- поле для ввода запроса;
- список Область поиска, например, разделы сайта, в которых следует искать. Этот элемент встречается нечасто;
- кнопка (гиперссылка), инициирующая сам поиск (обязательно срабатывающая при нажатии на клавишу Enter);
- ссылка на описание работы поиска и специальных символов для построения сложных запросов. На обычных Web-сайтах встречается редко.

#### Второй шаг. Уточнение результатов поиска:

- вызов Расширенного поиска;
- поиск среди результатов.

Форма поиска располагается, как правило, в верхней части страницы и состоит из текстового поля и кнопки. Это сочетание стало для многих пользователей настолько привычным, что, даже не идентифицируя эту форму, вы не собьете посетителя с толку. Но, конечно же, форму необходимо подписать.

Форму поиска предпочтительнее сразу выводить на страницу, а не помещать ссылку на нее, — это сокращает количество шагов. Знаменитый специалист по Web-дизайну Якоб Нильсен (Jakob Nielsen) уверяет, что, когда он заменил на своем сайте <http://www.useit.com> гиперссылку Поиск на саму форму поиска, использование данной функции возросло на 80%.

Пользователи интернета ленивы и разбалованы, поэтому они не будут педантично менять свой запрос в поле поиска, чтобы добиться нужного результата. Кроме того, не все обучены созданию правильных поисковых запросов. Таким образом, не достигнув нужного результата, они просто уйдут на другой сайт.

Поэтому после создания функции Поиск необходимо провести ее тестирование.

**В о - п е р в ы х**, надо выяснить, какие запросы наиболее часто будут вводить пользователи. Для этого можно воспользоваться информацией с поисковых систем, которую, как правило, анализируют для «раскрутки» сайта, для добавления в список ключевых слов.

**В о - в т о р ы х**, проанализировать, как именно посетители будут работать с вашим сайтом, т.е. протестировать запросы вашей целевой аудитории. Для этого необходимо привлечь несколько пользователей, ознакомить в общих чертах с содержанием сайта и попросить найти какую-либо информацию, воспользовавшись функцией Поиск. Перед началом анализа нужно выяснить, что именно хочет найти пользователь. Это поможет вам проверить, совпадут ли его ожидания с тем, что ему вернет ваш сайт. Во время такого тестирования необходимо следить за тем, какие запросы задают пользователи, насколько они сложны (одно слово, два и более), использует ли кто-либо специальные разделители («\*», «+», «OR»), как пользователи реагируют на результаты, выбирают ли они тот результат, который соответствует их запросу. Стоит уточнить, почему пользователь остановился именно на каком-то определенном результате. Думаю, суть вам ясна.

Если ваш сайт — большая серьезная разработка, вам придется фиксировать все запросы посетителей (например, в специальном файле) для последующего анализа и улучшения результатов поиска. Чем более четко и качественно работает поиск по сайту, тем крепче вы удерживаете посетителя. Это очень важно.

Отдельный разговор об оформлении результатов поиска и удобстве их использования. В кратких результатах поиска уже должны присутствовать и быть выделены слова из запроса. Это помогает понять контекст использования этих слов и быстро проанализировать результаты поиска.

## **6.5. Элементы содержания**

### **6.5.1. Текст**

Текст занимает 60–65 % Всемирной паутины. Способ подачи этой информации неоспоримо важен, однако большое значение имеет и заинтересованность посетителя в той информации, ко-




тору он ищет. Человек согласится читать все что угодно, если у него есть к этому стимул, пусть это статья про его соседей в городском новостном портале или информация о лечении редкой болезни его любимого хомячка. Ресурсов, которые могут не обращать внимания на то, как они ведут себя по отношению к пользователям, весьма немного. Главный двигатель прогресса в Web — это конкуренция. И удержать пользователя на своем сайте разработчику ой как непросто. Представьте себе любой крупный рынок с рядом лотков, продающих одно и то же. Кто из продавцов вас заинтересует своим предложением, тот и продаст его, хотя товар у всех практически одинаковый. Можно привести и другое обоснование вашего выбора. Вспомните ситуацию, когда вы оказались в городе без часов и вам нужно было узнать время. Разве вы спросите первого попавшегося человека? Лично я не раз ловил себя на мысли, что в такие моменты я спрашиваю только тех людей, которые подсознательно внушают мне доверие. «Этот злой какой-то... Эта расфуфыренная дамочка вообще не смотрит никуда, кроме своего маникюра... У этого, видно, и часов-то нет... А вот... Извините, не подскажете, который час?..» Доверие посетителей надо заслужить.

Подход к оформлению текста может быть различным. Существуют коммерческие сайты, электронные журналы и газеты, агентства новостей, Web-студии, поисковые и информационные порталы — это *первая группа сайтов*. Тексты таких сайтов всегда более структурированы и отличаются официальным стилем. Вольности в подаче материала допускаются крайне редко. Соответственно, и отношение у посетителей к текстам такое же — официоз и есть официоз, а если на таких сайтах ошибки или разговорный язык, то такому источнику просто перестают доверять.

Ко *второй группе* относятся развлекательные сайты, ресурсы, оказывающие помощь в проблемах, связанных с чем бы то ни было, форумы по настройке утюгов фирмы «Три простыни», «тусовка» и персональные сайты. Тексты на таких сайтах подобны разговорам в обычной жизни. Здесь может попадаться и ненормативная лексика, и жаргон, и намеренные грамматические ошибки — «такое знчт случаецца». Пользователи относятся к таким текстам снисходительно.

**Текст** — всякая завершенная последовательность предложений, слов, знаков, находящаяся на странице (статья, новость, рекламный блок, высказывания пользователей по какому-либо вопросу и т.д.). В большинстве случаев текст обладает заголовком, разделен на абзацы, которые, в свою очередь, состоят из предложений.

 Любая часть текста может быть оформлена не только в виде обычного статического текста, но и в виде гиперссылок.

Вот как выглядит структура любого обособленного текстового содержимого:

- Текстовое содержимое
  - Заголовок
  - Аннотация
  - Текст
    - Абзац
      - Предложение
        - Слово
          - Буква

Теперь пройдем последовательно по всем этим уровням.

Содержание **заголовка** зависит от целей, которые вы преследуете. Если вы хотите завлечь пользователя, то для таких целей часто используют «громкие» и интригующие заголовки. Например, «Подросток проглотил бензопилу» или «Квартиры теперь стоят втрое меньше». И не важно, что проглотил он малюсенькую бензопилу от игрушки своего младшего брата, а квартиры — в доме, который через два года снесут. Посетитель наверняка «клюнет». Хотя такие методы являются несколько нечестными по отношению к посетителям, они часто используются. Они утрированно демонстрируют то, как устроено внимание: люди часто хватают информацию кусками и, в первую очередь, заголовки. Отношение к заголовкам не меняется, даже если речь идет об информационных порталах. В случае, когда пользователь ищет информацию и завлекать его не надо, нужно удержать его внимание,

показав, что у вас есть то, что он ищет. Заголовок должен быть кратким, четким и ясным, чтобы пользователи понимали, с чем они столкнулись. Если ваш заголовок, например, «Создание качественных Web-страниц при помощи визуальных редакторов», а посетителю совершенно неинтересны визуальные редакторы, то зачем вам его удерживать, ведь он не из вашей целевой аудитории и больше к вам не вернется? Зачем обманывать его и тратить свое время на придумывание завлекающих элементов?

При составлении заголовка надо помнить:

- ваш заголовок будут читать все посетители — и опытные, и новички, — поэтому нежелательно использовать в заголовке профессиональный жаргон, понятный только узким специалистам;
- заголовок должен быть кратким и ясным;
- заголовок должен отражать самую суть содержимого. Учтите, если вы на скорую руку придумали «какой-нибудь» заголовок, то вы потеряете больше, чем если бы потратили на его придумывание целый день;
- в заголовок должны быть включены только самые важные слова, ничего лишнего, никакой «воды»;
- ключевое слово (фраза) должно быть первым.

Рассмотрим в качестве примера приведенный выше заголовок «Создание качественных Web-страниц при помощи визуальных редакторов». Он неплох, как мне кажется. Посмотрим, что с ним можно сделать. Представлю, что я посетитель. Меня лично совершенно не интересуют визуальные редакторы, т.к. я создаю «качественные» Web-страницы при помощи простого текстового редактора. Но я могу и не дочитать этот довольно длинный заголовок до конца и сразу начну знакомиться со статьей, а потом обижусь на то, что меня «обманули». Либо, наоборот, я мечтаю о создании своего сайта, но не знаю, как это сделать. В обоих случаях ключевыми словами являются «визуальные редакторы», поэтому необходимо их перенести в заголовке в начало.

«Визуальные редакторы: создание качественных Web-страниц». Уже лучше. Обратите внимание, что сам по себе заголовок еще и укоротился.

Далее «качественные Web-страницы». Те посетители, которые попали на ваш сайт не случайно, свято верят в то, что они всегда создают качественные страницы. Зачем говорить им об этом еще раз? В то же время, если опустить это слово и оставить только «Визуальные редакторы: создание Web-страниц», то такой заголовок покажется слишком общим, по принципу «все обо всем», в нем не будет изюминки. Я, конечно же, не претендую на самого лучшего составителя заголовков, но на своем сайте, над статьей о работе Web-мастеров в визуальных редакторах, я бы написал: «Визуальные редакторы: особенности создания Web-страниц».

**Аннотация** — это краткое изложение либо описание того, что человеку предстоит прочитать. Она должна давать представление, какие именно знания получит читатель при ознакомлении с информацией. Аннотация используется не очень часто, потому что не совсем соответствует той сути, которую придают Web большинство людей. Зачастую сама информация организована так, что в аннотации нет нужды. Простой пример: на сайте с международными новостями расположен список статей в виде гиперссылок-заголовков. Щелкаешь по заголовку — попадаешь в краткое содержание статьи. Щелкаешь по ссылке читать всю статью — раскрывается полное содержание.

Иногда аннотация используется для общего описания содержимого, например, для описания книги в интернет-магазине берется реальная аннотация из этой книги. Это самое удачное ее применение.

**Абзац** — это некоторая логически завершенная часть текста. Даже начинающих писателей в книгах и на курсах писательского мастерства (я читал об этом не раз) учат не делать абзацы длинными. Такие абзацы во много раз тяжелее воспринимаются, кроме того, читать текст «по диагонали» уже сложнее. Не стоит лишать посетителей этой возможности.

**Предложение** — логически завершенная часть абзаца (некоторая мысль). Длинные предложения воспринимать сложнее. При чтении предложения формируется мысль, и к моменту, когда память уже не может удержать эту мысль в воздухе и должна ее воспринять целиком или потерять — в зависимости от важности, —

внезапно появляется какой-нибудь деепричастный оборот, несколько прилагательных, которые, казалось бы, нужны именно в этой части, и посетитель уже совершенно потерялся в дебрях вашего предложения и не может вспомнить ничего, ведь вы не потрудились помочь ему в этом. Не хотите ли перечитать предыдущее предложение еще раз?

✓ Привожу исправленный вариант: «При чтении предложения мысль постепенно формируется, и наступает момент, когда память уже не может удержать эту мысль в воздухе. Она должна воспринять ее целиком или потерять. Если в этом месте внезапно появляется какой-нибудь деепричастный оборот или несколько, казалось бы, нужных прилагательных, то посетитель совершенно потеряется в дебрях вашего предложения. Он не может вспомнить ничего из прочитанного».

Слова, которые вы используете в предложении, зависят от вашего лексического запаса, образования, воспитания, привычек и темы. Науке составлять слова в предложения порой учат несколько лет, поэтому я не стану учить вас сейчас в рамках данной книги. Одно могу четко сказать — жаргон и просторечия, а также намеренные ошибки, «которые случаются даже при типа частных базарах про то, как забахать клевый сайтец», воспринимаются с большим трудом и отталкивают посетителей. Это правила плохого тона. Но многие почему-то считают, что урок по созданию, например, флэш-анимации, будет более понятен и доступен неопытному читателю, если он прочтет «зафигачить» вместо «вставить».

✓ К большей части текстовой информации в интернете можно применить те же знания и принципы, что и в издательском деле и полиграфии — о количестве шрифтов на странице, о читаемости текста, о дизайне. Если эти знания вам доступны — применяйте их.

### 6.5.2. Особенности восприятия текста

Восприятие текста как элемента содержания Web-страницы имеет свои особенности.

Во-первых, с экрана читать текст сложнее, чем с бумаги. Все сказки о том, что скоро мониторы станут большими, качествен-

ными и тогда читать с экрана будет также легко, я не воспринимаю всерьез. Чтобы перейти к предыдущей странице книги, мне достаточно только перевернуть ее — ни полос прокрутки, ни мышей, ни клавиатур — просто движение пальцем. Бумага не требует настройки контрастности и яркости, на ней нет панели инструментов и окон приложений. В конце концов, с книгой или листом бумаги я могу вальяжно развалиться в кресле, где читать мне удобнее.

Во-вторых, пользователи часто читают «по диагонали», чтобы уловить суть и решить, это ли им надо.

В-третьих, они не всегда успевают остановиться на подробностях. Осбопеотси всорпиятия члеовкеом тескта вобоше всеьма сртанны. Елси вы попытаетесь прчоитать зотт тескт мледенно, взомжоно, вы нчиего и не пйомете. Но, не напягаясь и чтая бсытро, вы легко овсоите исукссвто ткаого четния. Члеовек вопрсиниамет пркатисечки тоьлко певрую и полсендюю бувку, а не все слвоо.

☑ Особенности восприятия человеком текста вообще весьма странные. Если вы попытаетесь прочитать этот текст медленно, возможно, вы ничего и не поймете. Но, не напрягаясь и читая быстро, вы легко освоите искусство такого чтения. Человек воспринимает практически только первую и последнюю букву, а не все слово.

В-четвертых, размер шрифта (это касается и электронной книги, и обычной) влияет на восприятие информации.

В-пятых, то, насколько внимательно посетители читают текст, зависит от их заинтересованности в информации и ее эксклюзивности.

В завершение разговора о тексте хочу добавить, что Web отличается своей динамичностью. Его главная сущность — гиперссылка. Ей нужно поставить памятник, ведь без нее Всемирная паутина превратилась бы в плохо организованную Всемирную свалку информации. По этой причине из всего текста посетители сначала выхватывают взглядом гиперссылки. Им всем хочется динамики, хочется срочно куда-нибудь перейти. Они вечно спешат, спешат. Только настоящие «гуру» могут остановить их обычным текстом на своем сайте. Я желаю вам оказаться среди этих «гуру».

### 6.5.3. Поля ввода

**Поля ввода** — это компоненты HTML-страниц, посредством которых пользователь передает информацию системе. Они могут быть совершенно непохожи друг на друга, как, например, чек-бокс непохож на поле Выбор из списка.

К полям ввода относятся следующие элементы:

- текстовое поле ввода (однострочное (рис. 6.24) и многострочное (рис. 6.25));
- поле Выбор из списка (однострочное (рис. 6.26) и многострочное (рис. 6.27));
- чек-боксы (рис. 6.28) и радио-кнопки (рис. 6.29);
- нехарактерные для Web-интерфейса элементы (например, ползунок).

## Однострочное текстовое поле

Рис. 6.24. Текстовое поле ввода для одной строки

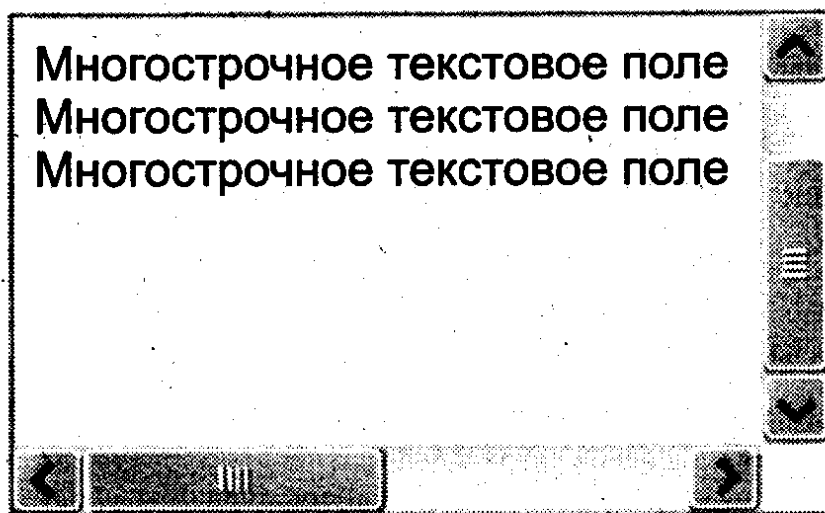


Рис. 6.25. Текстовое поле ввода для большого количества строк

## Выпадающий список



Рис. 6.26. Элемент Выбор из списка. Он же элемент одиночного выбора

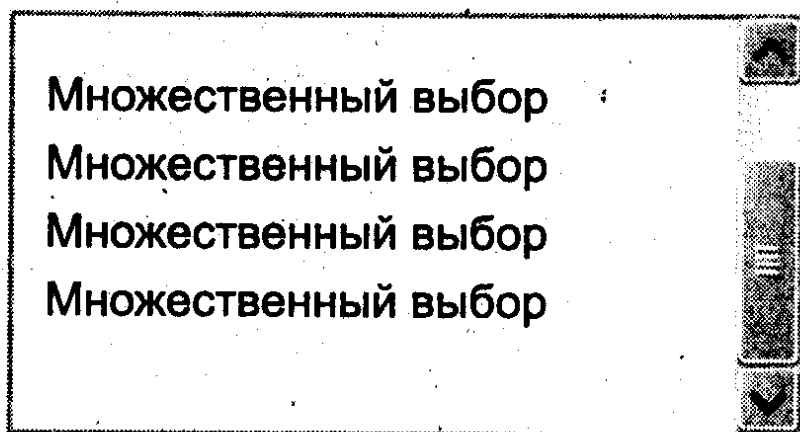


Рис. 6.27. Элемент множественного выбора



Рис. 6.28. Чек-бокс

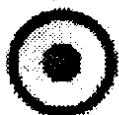


Рис. 6.29. Радио-кнопка

При выборе того или иного поля ввода и способа его оформления обратите внимание на следующие особенности.

Во-первых, длина и ширина полей ввода должны быть соразмерны предполагаемым единицам ввода. Странно выглядит поле для ввода месяца, которое растянуто на всю ширину страницы.

Во-вторых, если поле редактируемое, нельзя менять его вид таким образом, что его недоступность перестанет быть очевидной.

В-третьих, подписи к полям ввода могут располагаться несколькими способами. Некоторые специалисты считают, что подписи к редко используемым полям ввода можно располагать перед ними (такие надписи читаются перед тем, как взгляд упадет на поле ввода), а к часто используемым — над ними (чтобы не засорять кратковременную память пользователя уже известной ему информацией). Еще один способ — расположить подпись внутри элемента, что позволяет сэкономить пространство



экрана. При этом она выводится в самом поле ввода точно так же, как и текст, который нужно вводить. Необходимо только отслеживать фокус ввода, чтобы при его установке убиралась подпись. Такое нестандартное решение неплохо работает в интернете. Если на экран нужно вывести много информации, а экранное пространство очень жалко, стоит воспользоваться этим способом.

Не рекомендую использовать поля ввода, которые нельзя сделать стандартным способом в HTML. Они привычны для обычного программного обеспечения, но сложны и неустойчивы в Web.

#### 6.5.4. Кнопки

**Кнопки** — элементы немедленного действия, т.е. воздействие на них (нажатие или щелчок мыши) вызывает динамическую реакцию системы — переход на другую страницу, вызов нового окна, вызов программы и т.д. Это отличает их от обычных полей ввода, поскольку при вводе информации в какое-либо поле ввода ничего подобного не происходит.

Надписи на кнопках должны объяснять, что произойдет при нажатии на них. Пользователь как бы отдает приказ системе. Представьте себе джинна, который спрашивает: «Что ты хочешь, о Великий пользователь?» — «Отправить форму по e-mail». — «Я отправил форму, что ты хочешь теперь?» — «Перейти к следующему шагу». — «Исполняю...»

Глагольные формы (настроить, отменить) в надписях позволяют быстрее восстановить фокус внимания и легче понимать суть действия. Кроме того, надпись должна отражать не только действие, но и объект, если он не очевиден (Настроить параметры поиска, Удалить абонента). Исключение составляют общепринятые надписи вроде Помощь и ОК (рис. 6.30, 6.31).

#### 6.5.5. Гиперссылки

По отношению к текущей странице гиперссылки могут быть внутренними (якоря, ссылки наверх) и внешними (ссылки навигации и действия).

Гиперссылка — основной элемент управления перемещением пользователя в Web, поэтому к ней предъявляется ряд требований. Ссылка должна быть «видима»: если какой-то элемент

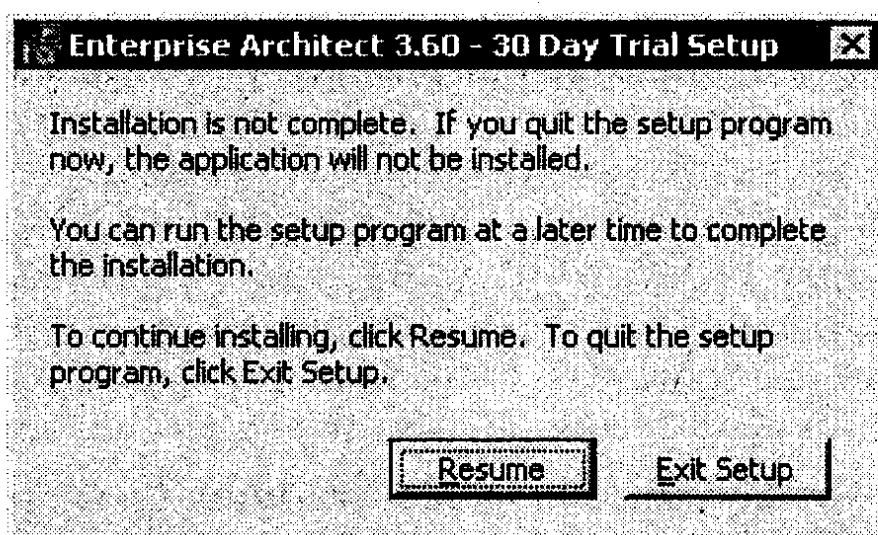


Рис. 6.30. Удачные надписи Возобновить и Выйти из программы установки позволяют вообще не читать информацию в диалоговом окне — все понятно и так

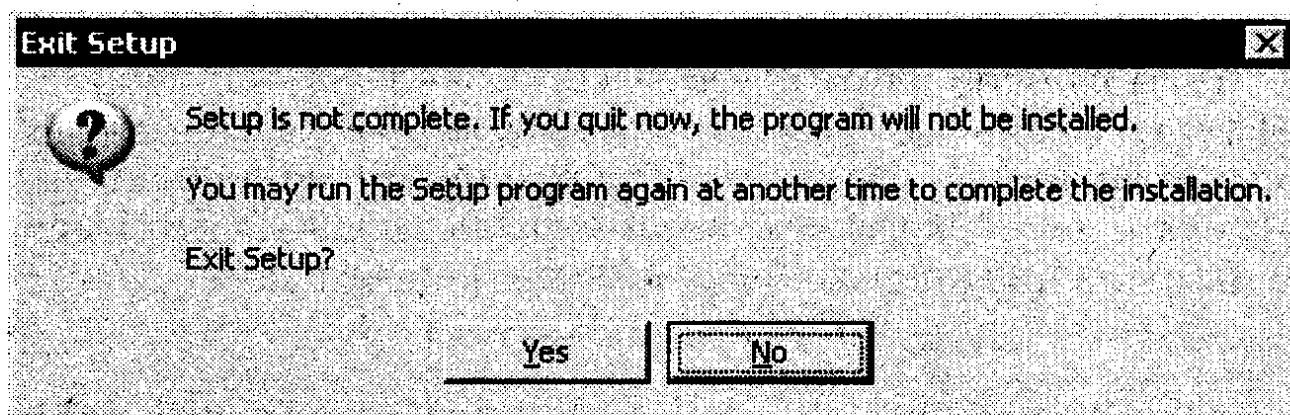


Рис. 6.31. Не очень удачные надписи на кнопках — Да и Нет. Для того чтобы понять назначение кнопок, необходимо внимательно прочитать всю информацию в диалоговом окне

страницы связан с гиперссылкой, это должно явным образом следовать из его внешнего вида. Одним из главных средств, обеспечивающих «видимость» гиперссылки, является подчеркивание. Это настолько неоспоримое условие принадлежности текста гиперссылке, что лично я любой подчеркнутый текст воспринимаю только таким образом и автоматически подвожу к такому тексту курсор. Текст в Web может быть подчеркнут, только если он является гиперссылкой.

По умолчанию ссылки в Web всегда синие. Если сделать их иного цвета, «видимость» их снижается. Оттенки, близкие к синему,

могут быть трактованы как синий, поэтому в крайнем случае для гиперссылок можно использовать цвета от зеленого до темно-синего.

Посещенные ссылки по умолчанию лиловые, хотя, в зависимости от настроек монитора, они имеют оттенки от грязно-розового до фиолетового. Тут та же история, что и с цветом активных ссылок: во-первых, обязательно задавать изменение цвета посещенных ссылок, а во-вторых, этот цвет должен находиться «поблизости» от лилового.

Не заставляйте посетителя гадать и вспоминать, где он уже был, а где еще нет.

Иногда роль гиперссылок играют другие элементы, например, вызов нового окна происходит таким образом:

```
<SPAN class=«span_href» onclick="window.open('next.html',  
'Вызов', 'height=300, width=300')">Вызов в новом окне </SPAN>
```

Если вы считаете, что гиперссылку использовать не стоит, т.е. эффект, к которому вы стремитесь, возможен только при использовании иных элементов, необходимо при помощи стилей, цветом и подчеркиванием симитировать гиперссылку. Кроме того, не забудьте про задание внешнего вида курсора:

```
.span_href {CURSOR: hand;}
```

Иначе никто не поймет, что это гиперссылка и по ней можно «кликнуть».

Еще одним важным фактором является выделение каким-либо образом текста гиперссылки при наведении на нее курсора (так называемый эффект *hover*). Пусть ссылка меняет цвет, становится не подчеркнутой или меняется фон элемента, внутри которого находится ссылка. Это дает понять, какая именно гиперссылка сработает при «клике», особенно если они расположены близко друг от друга.

### 6.5.6. Пиктограммы

Самая большая проблема пиктограмм в том, что их недостатки часто перекрывают достоинства. Как часто случается, используют с благими намерениями, а оборачивается сплошными проблемами. Использование пиктограмм не всегда оправдано.

Во-первых, их делают мелкими — в Web место на вес золота — следовательно, мало раздолья для фантазии, в иконки сложнее попасть, необходимы более точные движения и тому подобные проблемы мелких деталей. А если сделать их большими, то они станут занимать больше места, чем текст, т.е. теряют часть своих достоинств.

Во-вторых, порой непонятно, что они значат (а вообще, что это изображено — буква G, змея или пожарный гидрант), не говоря уж о понятности действия.

В-третьих (специфично для Web), пиктограммы могут исчезать, если у пользователя отключена графика или страница плохо загрузилась. В таком случае пользователь лишается возможности хоть как-то управлять ситуацией.

В-четвертых, из-за малого размера трудно поместить в пиктограмму что-нибудь четкое и понятное, поэтому большинство пиктограмм пользователи вынуждены просто зазубривать, «набивая себе шишки» в работе с ними.

В-пятых, текст редко бывает одинаковым, а вот пиктограммы часто похожи между собой. Если они находятся рядом, быстро работающий пользователь может «промахнуться» только из-за того, что рядом есть еще одна, выполненная в такой же цветовой гамме.

Теперь о плюсах.

- Пиктограммы хороши для наиболее значимых элементов, особенно в совокупности с текстовыми обозначениями, например, для пунктов меню. Чтобы выделить среди остальных какой-то пункт меню, можно оформить его полужирным шрифтом или шрифтом большего размера, а можно вставить перед ним пиктограмму.
- При использовании пиктограмм увеличивается скорость поиска взглядом нужного элемента (при условии, что пиктограммы хорошо выполнены и грамотно скомпонованы).
- Пиктограммы «украшают» страницу, т.е. делают ее не такой скучной, как обычный текст.
- Существует набор стандартных пиктограмм, которые понятны почти каждому пользователю, например, изображение дома (на главную страницу), дискеты (сохранить), конверта (напи-

сать электронное письмо). Однако количество таких пиктограмм крайне мало.

- Система с пиктограммами легче переводится на другие языки, т.к. изображения переводить не надо (за исключением изображений с текстом).

Надо отметить, что стилизованная пиктограмма (как в детстве «точка, точка, запятая — вышла рожица кривая...») воспринимается проще, чем обремененная мелкими деталями.

Для выделения нужной пиктограммы из всех остальных можно использовать приглушенные цвета, которые становятся ярче при наведении на них курсора. Решение с серыми неактивными пиктограммами, как в Internet Explorer 5.0, несколько ошибочно: они сильно похожи друг на друга по цвету, поэтому трудно различимы.

Контекст работы с пиктограммой также важен для точного понимания ее значения. Например, в любой операционной системе, где используются пиктограммы, изображение дискеты значит «сохранить куда-то», а в интернете — «сохранить откуда-то». Поэтому при создании пиктограммы имеет значение не только то, что на ней нарисовано, но и в каком значении.

Некоторые пиктограммы практически невозможно сделать понятными. Например, я решил поставить эксперимент с созданием пиктограммы, обозначающей фильтр. Когда я показал вымученную картинку 24×24 пикселя, один человек сказал мне, что это мясорубка, второй — что это глубокая тарелка, а третий спросил, почему я так странно нарисовал перевернутый зонтик (рис. 6.32).



Рис. 6.32. Пиктограмма «фильтр». Непонятно даже то, что это фильтр, не говоря уже о функции данной пиктограммы

С эстетической точки зрения важно поддерживать единую стилистику пиктограмм. Например, плавные, скругленные контуры или «плоские» пиктограммы, пиктограммы с одним фоновым элементом или стилизованные «под хохлому» (рис. 6.33).

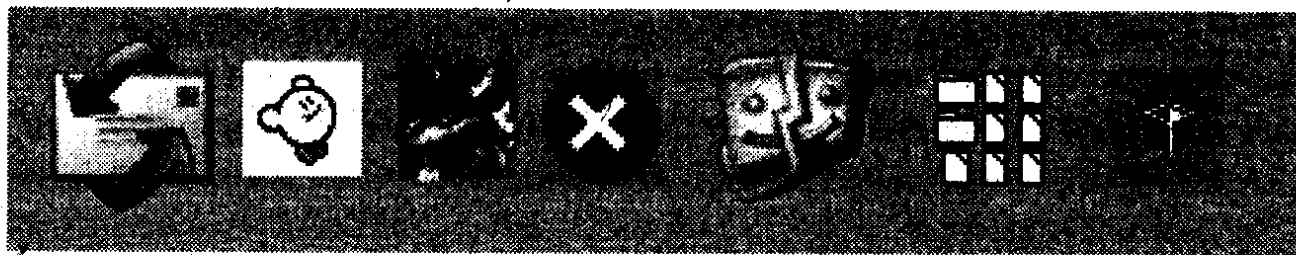


Рис. 6.33. Мир пиктограмм разнообразен и пестр

А теперь самый главный закон в создании пиктограмм — они не должны находиться рядом, выглядеть одинаково или быть оформлены в одной цветовой гамме (рис. 6.34).

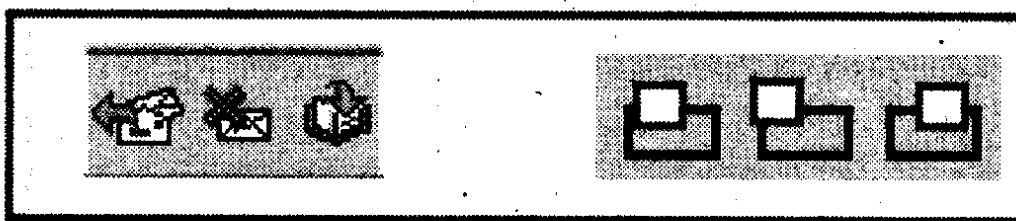


Рис. 6.34. «Одинаковые» пиктограммы (слева) и то, как их видит пользователь (справа)

Без комментариев (рис. 6.35).

### 6.5.7. Списки

**Список** — это элемент, представляющий собой развернутую древовидную структуру, обычно используется для перечисления однотипных структурных единиц (рис. 6.36).

Внешний вид списка может быть весьма разнообразным: списки могут быть сортированные и несортированные, элементы списка могут быть обозначены буквами или цифрами, стандартными значками (круг, квадрат), изображениями (пиктограммы папок).

Списки могут использоваться как способ подачи информации, и тогда они чаще всего оформлены в виде обычного текста, а могут быть и своеобразным элементом навигации, тогда элементы содержат текст с гиперссылками.

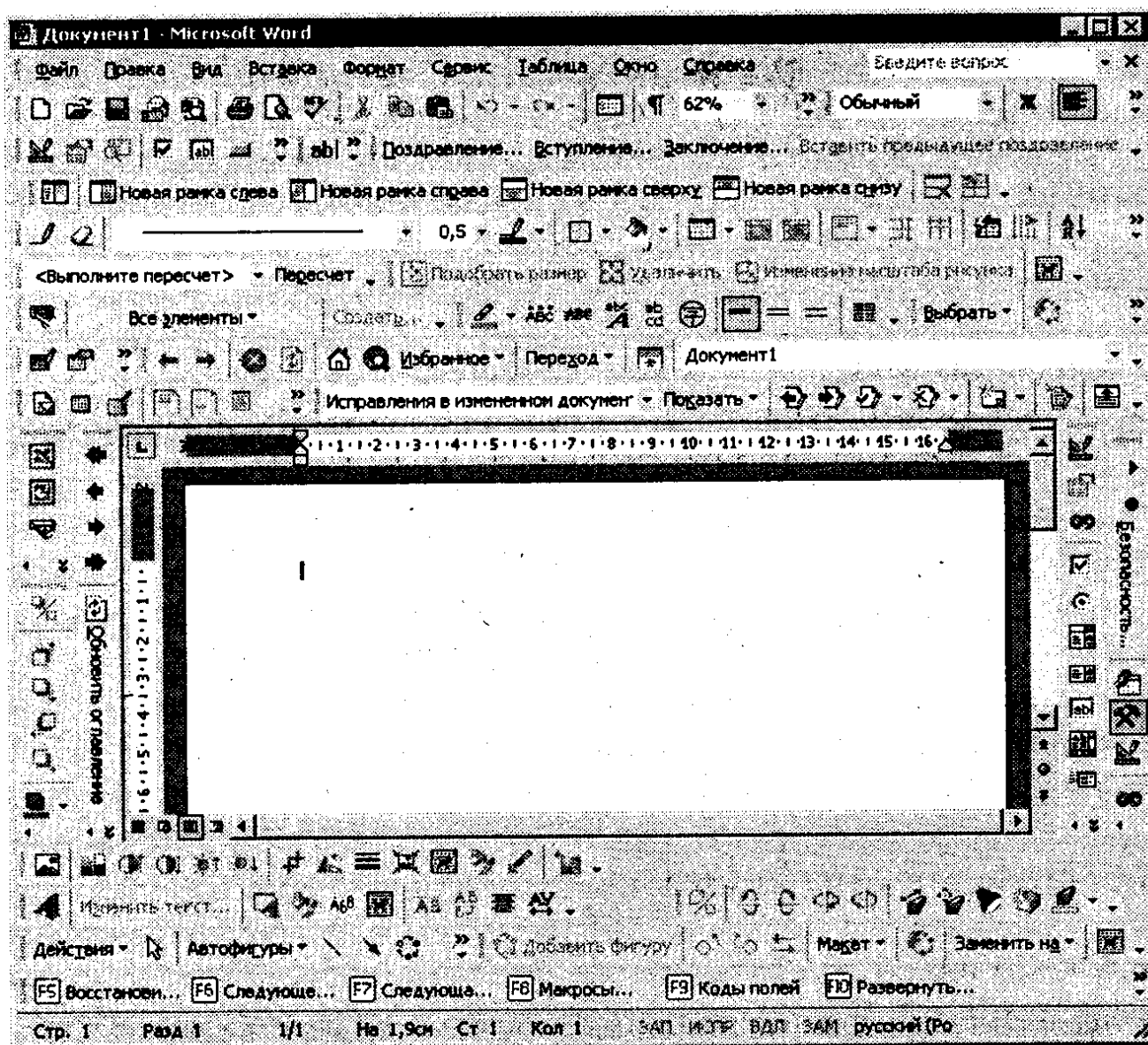


Рис. 6.35. Кризис жанра. Это, конечно же, шутка. Но в каждой шутке есть доля шутки, а все остальное правда

- First list item
- Second list item
  - First list item
  - Second list item
  - Third list item
- Third list item

Рис. 6.36. Вот так выглядит простой список

## 6.6. Элементы оформления

К элементам оформления относятся элементы, которые образуют самую верхнюю часть Web-сайта — его внешний вид. Это не сама информация, а лишь ее представление. Я не хотел в этой книге говорить ни слова о дизайне, потому что это разрушило бы ее концепцию. Поэтому речь идет только об элементах оформления, а не о пропорциях, светотенях и особенностях Web-дизайна.

Первым из таких элементов является *цвет*. Имеется в виду любой цвет, используемый на странице: цвет шрифта, цвет фона, цвет границ. Возможное количество цветов в композиции и способы их сочетания подчиняются законам дизайна и графики. Цветовые свойства элементов определяются важностью, ролью на странице, размерами и т.д. Цвет влияет не только на психологический комфорт посетителя, но и на усталость глаз, и на легкость восприятия информации.

Вторым элементом является *фон*. Фон может быть выполнен каким-то одним цветом, а может быть оформлен при помощи накладываемого изображения. Основная проблема с изображением в том, что его нельзя делать четким и контрастным, потому что тогда «потеряется» основная информация и ее будет сложнее воспринимать. Кроме того, если использовать в качестве фона не какую-либо картинку, а абстрактный рисунок или градиент, то незнание законов дизайна и композиции может породить графический хаос.


Однако если использовать в качестве фона рисунок из файла, то можно сделать и полезные вещи. Например, изображение, загружаемое в ячейку таблицы (с указанием не «размножить» рисунок фона) не в явном виде (`<img>`), а как фон, избавит вас от досадных недоразумений с недозагрузкой картинок: пока изображение не будет готово полностью, пользователь будет видеть лишь фон определенного цвета.

*Границы* имеют практически все элементы, для которых можно задать фон и цвет. Границы характеризуются цветом, толщиной, способом начертания (двойная, прерывистая, сплошная и т.д.). Их внешний вид может быть задан для каждой из сторон эле-



мента. По умолчанию все границы имеют некоторый визуальный объем. Весьма интересно, насколько легко это получается. Если нарисуете прямоугольник какого-либо цвета и сделаете его верхнюю и правую границы чуть светлее, а нижнюю и левую чуть темнее, то вы получите выпуклый прямоугольник, который будет походить на кнопку.

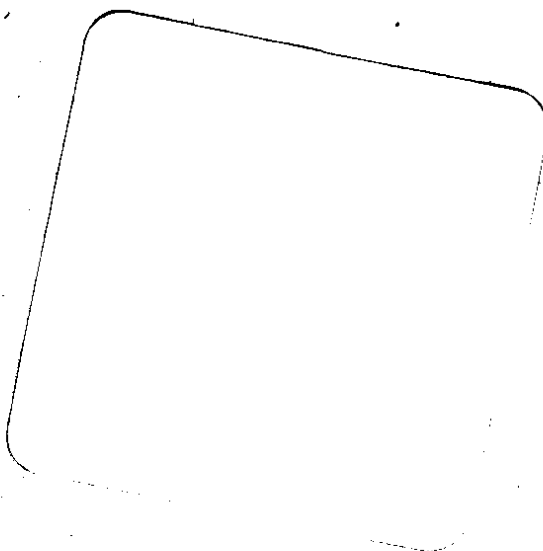
Одним из основных элементов оформления является *шрифт*. Он обладает такими свойствами, как размер (90 %, +3, 12 pt), начертание (обычный, полужирный, подчеркнутый), гарнитура (Times, Arial, Tahoma, Verdana и т.д.), цвет (синий, красный, черный).

 Все свойства элементов, рассматриваемых в данном разделе, не могут быть описаны в одном месте в одно время — они слишком разнообразны. Я лишь перечислил их, а особое внимание каждому конкретному элементу уделю в дальнейшем повествовании.

## Резюме

1. Навигация — это набор способов перехода между страницами и внутри них, обеспечиваемый в основном элементами содержимого страниц, а также средствами браузера, операционной системы и аппаратного обеспечения.
2. Необходимо обеспечивать посетителей функцией Поиск. Особенно актуален поиск на сайтах, содержащих большое количество информации.
3. Форму поиска предпочтительнее сразу выводить на странице, а не помещать ссылку на нее.
4. Структура любого обособленного текстового содержимого имеет вид:
  - Текстовое содержимое
    - Заголовок
    - Аннотация
    - Текст
      - Абзац
      - Предложение
      - Слово
      - Буква

5. Глагольные формы в надписях на кнопках позволяют пользователю быстрее восстановить фокус внимания и легче понять суть действия, которое произойдет.
6. Гиперссылка должна быть «видима», т.е. если какой-то элемент страницы связан с гиперссылкой, это должно явным образом следовать из его внешнего вида. По умолчанию ссылки в Web всегда синие и подчеркнутые. Посещенные ссылки по умолчанию лиловые. Нежелательно использовать в Web просто подчеркнутый текст, т.к. это вызывает ассоциацию с гиперссылкой.
7. Плюсы пиктограмм: они хороши для наиболее значимых элементов, увеличивается скорость поиска взглядом нужного элемента, стандартные пиктограммы понятны почти каждому пользователю, система с использованием пиктограмм легче переводится на другие языки.
8. Минусы пиктограмм: они должны быть мелкими, порой они непонятны, пиктограммы могут исчезать, если у пользователя отключена графика или страница плохо загрузилась, из-за малого размера трудно поместить в пиктограмму что-нибудь четкое и понятное, пиктограммы часто похожи.



# Юзабилити в теории и на практике

*Vestra salus — nostra salus*<sup>1</sup>

Все фотографии с моей свадьбы были сделаны и собраны в одну пачку, когда после этого события прошло несколько месяцев. Моя жена, наконец-таки, купила такой альбом, который хотела. Я принялся вставлять фотографии, но Таня меня остановила, отобрала все снимки и стала раскладывать их на полу. На мой вопрос, зачем это делать, она ответила: «Чтобы все снимки в альбоме были расположены парами, т.е. вертикальные фотографии с вертикальными, а горизонтальные с горизонтальными. Человеку, который будет смотреть альбом, не надо будет вертеть его в разные стороны. Он увидит первый снимок, повернет, если это надо, и сможет посмотреть второй кадр, не меняя положения альбома». В этом случае способ использования альбома (интерфейс) не изменился: остались все те же страницы, которые необходимо переворачивать вручную, зато удобство использования (так называемое юзабилити) улучшилось.

## 7.1. Определение юзабилити

Что же такое юзабилити (usability)?

Если буквально переводить с английского языка это слово, то получится что-то вроде «способность к использованию» или «удобство употребления». В общем случае *юзабилити* — степень удобства и простоты использования некоторого продукта для его потребителя, а также способы обеспечения этого удобства. Помимо слова «юзабилити» существует более известный в русском языке термин «эргономика».


В определении юзабилити я употребил слово «продукт», что говорит о широкой применимости понятия. Юзабилити является очень важным критерием не только для Web-сайтов, но и для различных программ (графических и текстовых редакторов, кли-

---

<sup>1</sup> Ваше благо — наше благо (лат.).

ентских оболочек СУБД и т.д.). Кроме того, из приведенного примера с фотографиями ясно, что юзабилити — свойство, присущее и вполне бытовым вещам.

Сталкивались ли вы с ситуацией, когда, запуская на своем компьютере новую программу, вы часами возились с ней, пытаетесь сделать элементарные действия, но в другой программе подобного плана вы разбирались в течение нескольких минут? А как быстро вы установите чужой видеомаягнитофон на запись 16 канала через 2 часа 30 минут?

 Юзабилити невозможно измерить количественно. Нельзя сказать, что юзабилити некоторого поискового портала равно 6,9 сайто-часов или 18 человеко-страниц. Можно лишь говорить о том, что работать с одной поисковой системой удобнее, чем с другой.

Конкретный смысл понятия «юзабилити» определяется в контексте:

- если дело касается внешнего вида некоторого Web-сайта, то юзабилити определяют специфичные для Web законы (цвета ссылок, размеры кнопок и пиктограмм и т.д.);
- если речь идет о юзабилити программного продукта, например, графического редактора, то становятся более важными способы выделения объектов, количество «горячих» клавиш и «скрытых» команд, очевидность взаимодействия с элементами, перегруженность интерфейса лишней информацией и т.д.;
- если мы говорим о «начинке» компьютера, то на удобство его использования при настройке и сборке влияют способы крепления деталей, количество переключателей и перемычек на платах и т.п. К слову, когда мне самому понадобилось поменять оперативную память в компьютере, чего раньше никогда не делал, я почти десять минут пытался понять, как же она крепится.

Однако вернемся к программным продуктам и Web-сайтам.

Основной целью любого программного продукта с точки зрения интерфейса и удобства использования является, на мой взгляд, следующее: пользователь должен работать эффективно, рационально, получать удовлетворение от работы и максимально быстро достигать поставленных целей. Иначе говоря, входными

данными для юзабилити должны служить цели пользователя, а выходными — эффективность его работы.

В качестве примера рассмотрим книжный интернет-магазин. Представьте себе, что вы его покупатель и желаете приобрести некий «Справочник».

Вы обладаете входными параметрами (запросами):

- вы никогда ничего не покупали в интернет-магазинах;
- у вас ограниченный трафик и странички загружаются медленно;
- за каждую лишнюю минуту в интернете вы платите из своего кармана;
- вам очень нужен «Справочник».

Высокий уровень юзабилити должен обеспечить следующие выходные параметры (результаты):

- относительно легко разобраться с тем, как сделать покупку;
- купить именно тот «Справочник», который нужен;
- не заходить на те странички, которые не содержат нужную вам информацию;
- сделать все быстро, чтобы сэкономить свои деньги.

Теперь представьте себя по другую сторону, т.е. вы продаете «Справочник».

В интернете может быть много магазинов, подобных вашему, и пользователи об этом знают. Малейшие неудобства при работе, раздражающее внешнее оформление, отсутствие или неполнота той информации, которую вы «обещали», ошибки в работе — эти и другие факторы могут заставить пользователя незамедлительно покинуть вас.

Это явная потеря клиентов и партнеров, денег, престижа, авторитета — кому что важнее.

Чтобы этого не происходило, вы обязаны гарантировать пользователям следующее:

- начав знакомство с вашим программным продуктом с нуля, пользователь должен быстро обучиться работе с ним;
- пользователь должен как можно меньше экспериментировать, чтобы выполнить свою работу, не посещать ненужные ему страницы и не выискивать полчаса кнопку или ссылку;

- сделав ошибку, пользователь не должен снова заполнять большую и сложную форму, на которую он уже убил четверть часа. Для этого нужно на понятном ему языке предупреждать об ошибке либо «не дать» ее совершить;
- пользователь должен убедиться в том, что сделал все правильно, и получить то, что ожидал. Для этого может быть выдано сообщение, что его действия завершены и все выполнено верно;
- если у пользователя снова возникла необходимость заказать книги именно у вас (а при должном уровне юзабилити и удовольствии от посещения вашего интернет-магазина так и должно быть), он должен работать так же эффективно, как при оформлении заказа в прошлый раз, когда он уже наловчился. Для этого интерфейс вашего сайта должен быть запоминаемым;
- наконец, поработав с вашим интернет-магазином, пользователь не должен в психозе разбить клавиатуру или пойти нервно курить из-за того, что у него ничего не получилось, а он убил массу своего драгоценного времени. Он должен откинуться на спинку стула, потереть руки или сладко потянуться, получив удовлетворение от выполненной работы.

Человек устроен так, что на все хорошее, что с ним происходит, он обращает гораздо меньше внимания, чем на плохое. Это относится и к работе с Web-сайтом. Интернет-ресурсы могут коренным образом отличаться, но, если с ними удобно работать, эти отличия словно исчезают и самого процесса достижения цели мы не замечаем. В то же время неудачные интерфейсные решения въедаются в сознание, как ржавчина в металл, и стереть их очень трудно. Вы не обязательно должны получать массу благодарственных писем за хороший дизайн и высокий уровень юзабилити, но за противоположное, будьте уверены, вы натерпитесь. Вас завалят недовольными отзывами, и ваша репутация будет безнадежно испорчена.

## **7.2. Удобство работы с информацией в Web**

### **7.2.1. По левую руку от светофора**

Если вам приходится искать магазин или офис по описанию, то происходит это примерно так: вы начинаете поиски с какого-то

места, например, автобусной остановки, и относительно быстро направляетесь к ближайшему объекту, который описывал вам знакомый, например, к перекрестку. Вы переходите дорогу и несколько замедляете шаг в поисках «высокого кирпичного дома по левую руку от светофора». Тут выясняется, что понятие «по левую руку» у вас с вашим знакомым расходится, потому что вы видите край какого-то кирпичного дома, но только с правой стороны. Неуверенно вы направляетесь к нему, но, как только в поле зрения появляется лестница, которая по описанию должна быть сразу за этим домом, вы снова ускоряете шаг — вы видите следующую цель и знаете, что она приведет именно туда, куда нужно. По пути вы не обращаете внимания на магазин с вывеской «Вам сюда», хотя именно его вы и ищете. Подойдя к лестнице, вы приостанавливаетесь, быстро находите указатель и следуете в его направлении. Через несколько минут вы уже у магазина. Вам помогло то, что у вас была возможность легко принять решение о следующем шаге, в отличие от людей, которые ничего не знали о светофоре, кирпичном доме, лестнице и указателе.

### **7.2.2. Визуальное выделение важной информации**

Не всякая информация нужна пользователю, поэтому выделять следует только важную для принятия решения информацию. Вы наверняка знаете, что требуется для совершения следующего шага и как осуществляется переход к нему — нажатие на кнопку, выбор из списка, перенос по принципу «drag and drop» какого-либо элемента. Важно сосредоточить пользователя на этой детали, сделать ее «видимой». Представьте себе, насколько легче было бы найти магазин «Вам сюда», если бы на черно-белом ландшафте выделялись оранжево-коричневый кирпичный дом, лестница с синими перилами и бирюзовые буквы на указателе. Иногда подобное выделение важной информации, точнее особенности восприятия ее человеком, используют в целях сокрытия иных сведений. Некоторое время в моем городе были расставлены рекламные щиты одного из операторов сотовой связи со словами «БЕСПЛАТНЫЕ РАЗГОВОРЫ С ЛЮБИМЫМИ номерами до конца апреля» (рис. 7.1). После этот прием был использован еще не один раз.



Рис. 7.1. Выделение важной информации. Фраза «Бесплатные разговоры с любимыми» особо выделена, хотя является единым целым со словами «номера до конца апреля»

Человек, который читает эти плакаты, получает толчок к действию или к размышлению на тему «Какие они классные парни — предоставляют бесплатные разговоры с любимыми». Он может не дочитать слова «до конца апреля», потому что эта информация не выделена.

### 7.2.3. «Географическое» выделение важной информации

Выделение информации может осуществляться не только на визуальном уровне (при помощи цвета, размера, шрифта), но и на композиционном уровне (рис. 7.2).

✓ Начиная с версии 3.0, я всегда пользовался англоязычной версией редактора Adobe Photoshop. Однажды я столкнулся с его русификацией у знакомого, который попросил помочь с восстановлением старой фотографии. Я не понимал значения и половины слов меню, потому что перевод совершенно очевидных для меня английских терминов оказался не на высоте. Однако, обладая хорошей зрительной памятью, я смог выбирать нужные пункты меню, не читая их названий, а просто вспоминая, где они располагались. Конечно же, я иногда ошибался, но работал все-таки достаточно продуктивно.

Вот постулаты композиционного выделения информации:

- человек воспринимает информацию слева направо и сверху вниз. Поэтому компонент Web-страницы, на который вы хотите обратить внимание, будет быстрее замечен, если он расположен слева вверху;



- крупные блоки ищутся быстрее мелких, а если они еще и выделены, например, оттенком цвета или заметной картинкой, то перейти к ним и начать работать можно гораздо быстрее.

Пользователь смотрит на страницу с композиционными блоками примерно так: мне нужен тот прямоугольник желтого цвета, который расположен слева внизу.

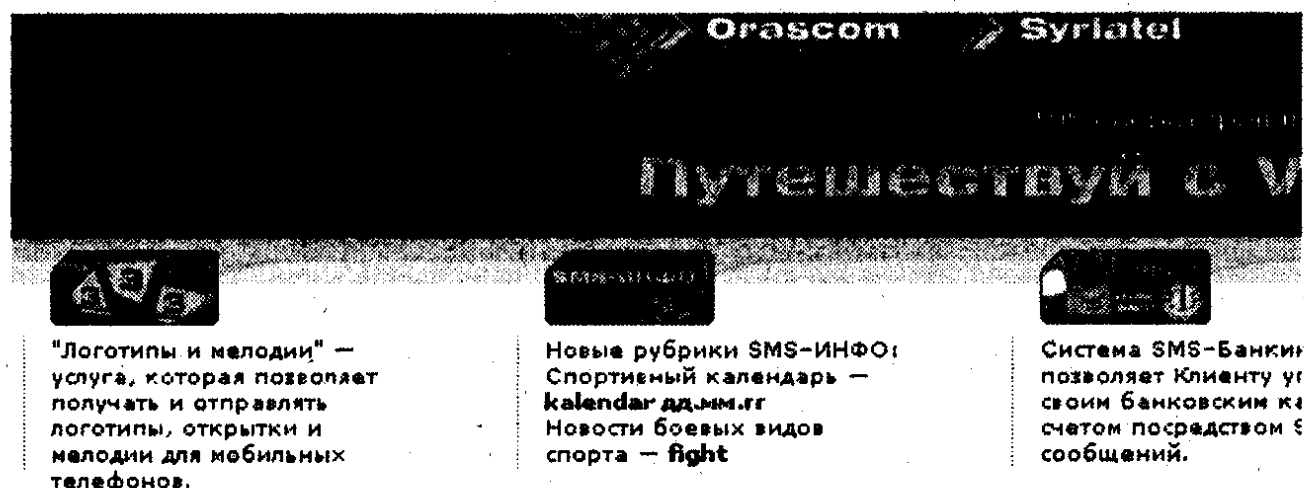


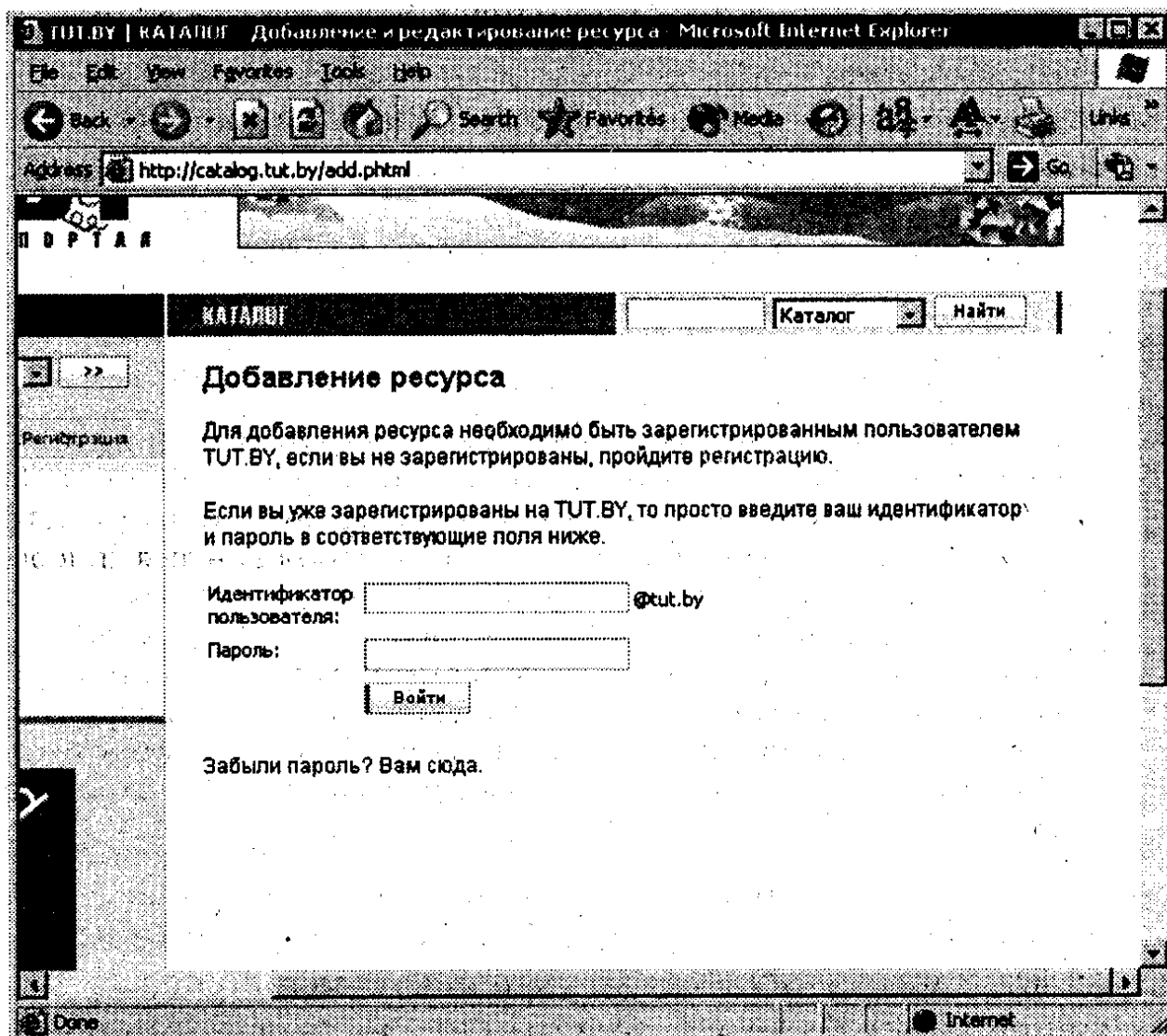
Рис. 7.2. Композиционные блоки хорошо отделены и обозначены отличающимися картинками

Перемещение мыши часто происходит бессознательно, к тому же зрительные образы «работают» быстрее, чем осознанный поиск перехода к следующему шагу. Посетитель попадает мышью в нужную область экрана почти сразу, и туда же перемещается его фокус внимания. Это позволяет не отвлекаться на внимательные поиски нужного элемента.

- ☑ Эти два постулата удачно совмещаются. Например, вы создали композиционные блоки, которые содержат набор ссылок. Некоторые из этих ссылок являются наиболее важными и будут использоваться чаще других. В таком случае, если вы используете полужирный шрифт для их отображения или сопроводите их пиктограммами, обнаружить их будет намного легче.

#### 7.2.4. Выбор заголовка

Мы уже упоминали о том, какую роль в обеспечении удобства использования Web-сайта играют четкие, и ясные заголовки (рис. 7.3).



**Рис. 7.3.** Четкий и ясный заголовок. На сайте он еще выделен бордовым цветом

Человек никогда не начинает сразу вчитываться в текст на странице. Он сначала пробегает глазами саму страницу, выхватывая лишь крупные или «видимые» куски содержимого — композиционные блоки, заголовки, отвлекающие элементы. Они позволяют быстро понять содержание, что можно здесь сделать и нужна ли ему эта страница.

Представьте себе грибника в лесу. Сам лес — это как бы ваш сайт, грибы — информация, за которой пришел пользователь. Опытный грибник, только попав в лес, не будет искать подберезовики. Он сначала найдет место, где они «должны» расти: открытую полянку, не сырую, не в хвойной части леса, достаточно освещенную солнцем... Так же и пользователи: прежде чем на-

чать читать котировки акций, они найдут заголовок «Котировки акций».

Важно отметить, что при использовании для заголовков стандартного тега `<h1>` можно улучшить не только юзабилити самого Web-сайта, но и результаты поиска в поисковых системах. Когда индексирующие роботы поисковых серверов натываются на заголовки `<h1>`, они используют его содержимое в качестве заголовка результата поиска.

☑ Раньше я как-то не задумывался о юзабилити результатов поиска на поисковых серверах. Однако недавно как-то ради интереса я задал запрос на [rambler.ru](http://rambler.ru) на отображение всех страниц моего личного сайта, проиндексированных этим сервером. Мне вернулся список, в котором заголовок каждого результата был одинаковым. Нечто вроде «Персональный сайт Андрея Калиновского». Удручающая картина, надо заметить. Еще больше я расстроился, когда посмотрел таким же образом страницы сайта одной знакомой мне фирмы, и там все было красиво и четко. Если страница была об «Автоматизации бизнес-процессов», то и в заголовке результата поиска было то же самое.

Естественно, пользователю куда проще и понятнее «кликнуть» по конкретной ссылке «Предоставление услуг бесплатного хостинга», чем зайти на «Web-сайт крутейшего мегапортала». Эти ссылки приведут на одну и ту же страницу, но первый заголовок внимание посетителя не пропустит, а мимо второго проскочит и даже не задержится. Фразы типа «10 000 баксов в месяц — это реально!» меня лично всегда отпугивают, я думаю, не только меня одного.

Итак, необходимую пользователю информацию, по нашему мнению, мы выделили. Теперь на странице можно более или менее ориентироваться. Далее поговорим о способе представления этой информации.

### 7.2.5. Контрастность текста и шрифт

Известно, что удобнее всего читать текст, который контрастирует с фоном, например, темный шрифт на белом фоне (либо на едва заметных оттенках других цветов). Возможен и обратный вариант, например, светло-серый текст на темно-синем фоне, но он менее привычен, отвлекает, замедляет чтение и, по моему

личному мнению, претенциозней «черного по белому». Если контраст достаточно высок, глаза меньше устают, читаемость текста улучшается и пользователь чувствует себя более комфортно, чем если бы ему приходилось вглядываться в каждое слово (рис. 7.4).

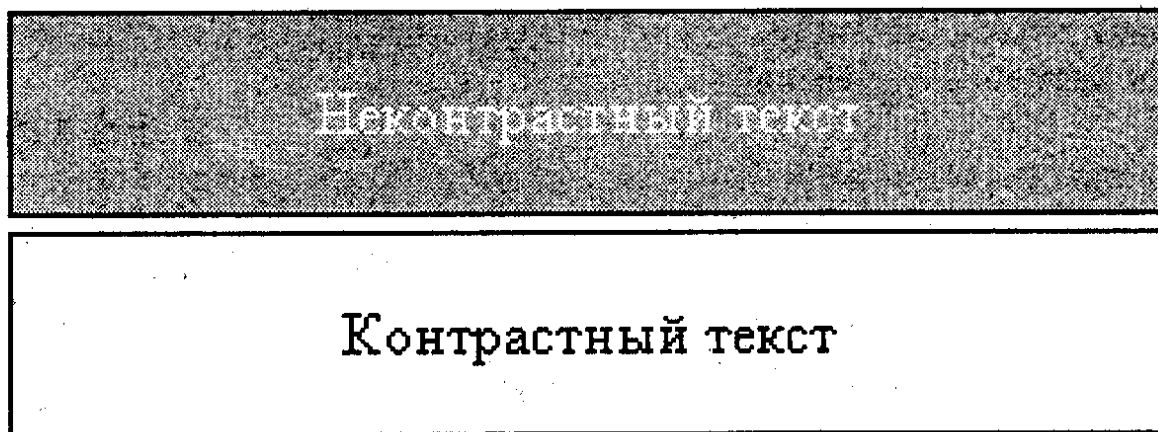


Рис. 7.4. Текст верхнего блока гораздо труднее прочесть из-за малой контрастности текста и фона

Задавались ли вы когда-нибудь вопросом, почему книги не печатают изящным шрифтом с завитушками? А почему вы не видите в магазинах поздравительных открыток на испанском языке? Правильно. Такого рода информацию трудно либо вовсе невозможно воспринимать. Говоря о том, что допустимо использовать не более трех шрифтов, я лишь повторяю мнение большинства специалистов, работающих не только в Web. Это правило применяется и при верстке обычных газет, и в большинстве книг. Желательно использовать привычные шрифты (Arial, Times, Courier, Verdana...). И дело даже не в том, что ????? ?????? ?????? ?? ????????, т.е., простите, у пользователя таких шрифтов может не оказаться (в конце концов, шрифты можно хранить на сервере и загружать в случае необходимости). Стандартные шрифты хорошо читаемы, просты для восприятия и привычны. Они как незнакомец, который чем-то похож на вашего друга: вы еще не знаете его, а уже расположены к нему. Есть еще масса других правил насчет того, что шрифт с засечками читается проще, **ТЕКСТ, НАПИСАННЫЙ ЗАГЛАВНЫМИ БУКВАМИ, ЧИТАЕТСЯ МЕДЛЕННЕЕ**, а *курсив* воспринимается как *рукописный*. Впрочем, по этим вопросам есть масса литературы.

### 7.2.6. Кодировка

Что касается кодировки, то эти вопросы решаются «малой кровью». Чаще всего правильное автоматическое распознавание браузером кодировки текста обеспечивается мета-тегом в заголовке страницы:

```
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">.
```

Достаточно прописать это на странице, и пользователю уже не придется вручную настраивать кодировку и перечитывать страницу снова (рис. 7.5).

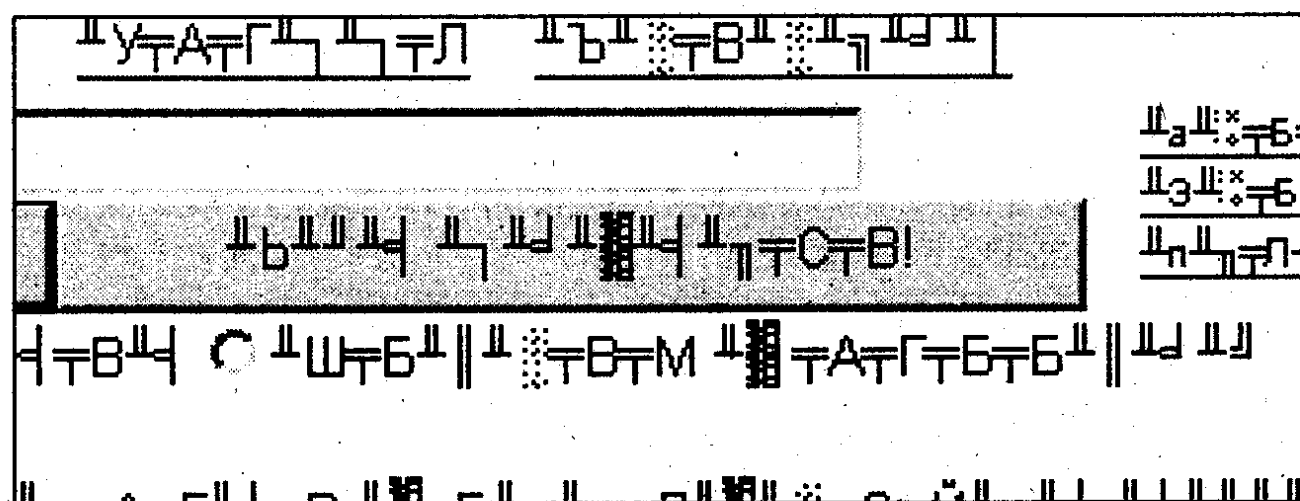


Рис. 7.5. Страницу можно прочесть, только переключив кодировку средствами браузера

### 7.2.7. Грамматические ошибки

Ничто так не раздражает, не отвлекает и не вызывает неприязни во время чтения, как грамматические ошибки в тексте. Я уже говорил, что хорошее не замечается, в то время как плохое помнят долго. Прежде чем выдавать посетителю информацию, проверяйте ее на наличие стилистических и грамматических ошибок. Это проявление элементарного уважения к пользователю и демонстрация хорошего стиля. Ведь вы не хотели бы работать в операционной системе, например, Microsoft Windows, где в меню «Пуск» содержатся пункты «Праграммы», «Спрафка» и «Завиршенее сианса» (рис. 7.6). Смешным очепяткам место на развлекательном сайте в соответствующем разделе; а не в сведениях о компании или новостях...

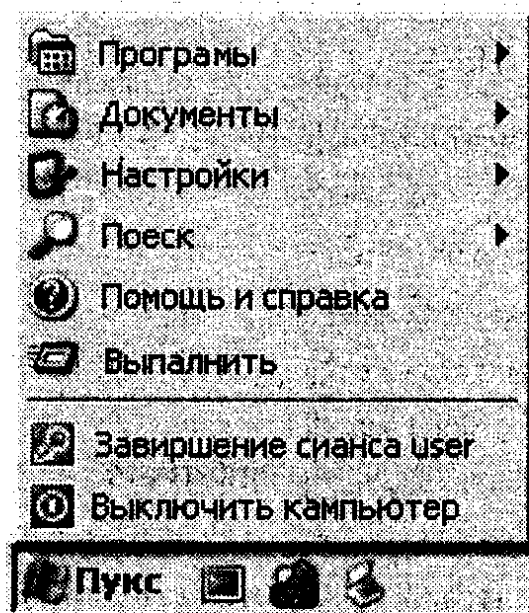


Рис. 7.6. Господа из Microsoft не имеют к надписям на этом изображении никакого отношения. Это фотомонтаж. Надеюсь, достаточно показательный

Посетитель нашел информацию и смог легко ее воспринять. Теперь необходимо позаботиться о том, чтобы было ясно, что эта информация в себе несет, т.е. о ее содержании.

### 7.2.8. Как вы лодку назовете

Разгадка ребусов не является целью посетителей большинства сайтов, если это, конечно же, не ресурсы, посвященные головоломкам. Чем яснее надписи, значения пиктограмм, термины, тем быстрее пользователь поймет, надо ли ему «кликать» мышью на них и куда они его приведут (рис. 7.7). Что может означать действие, описанное как Выбрать документ, когда пользователь уже находится в форме просмотра документа? Обработать? Запомнить? Просмотреть подробнее? Будет ли фраза Проиндексировать сайт понятна новичку? Или он решит, что это какое-то страшное действие, которое сотрет весь интернет с лица земли? Надпись Связь в верхнем меню сайта интернет-провайдера ведет к описанию услуг связи или к координатам головного офиса компании? Никто не должен задумываться над подобными вопросами в ходе работы.

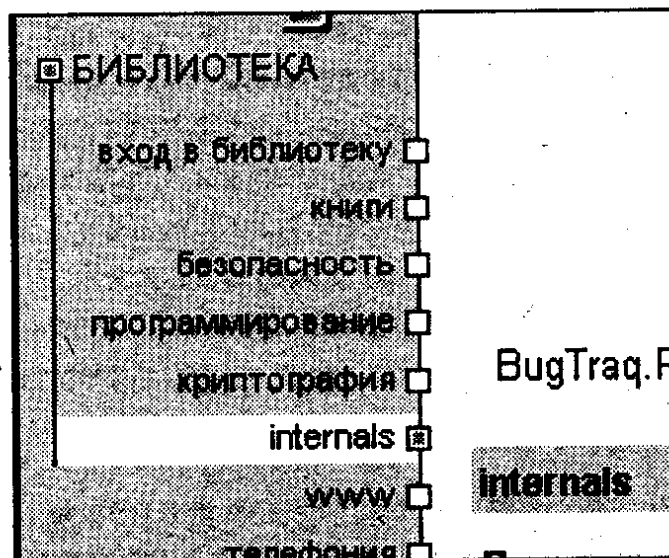


Рис. 7.7. Совершенно непонятный пункт меню Internals, суть которого я не понял, даже зайдя в него по ссылке. Ясность не вносит и перевод этого слова с английского — «внутренние», «внутренности»

## 7.3. Удобство работы с навигацией

### 7.3.1. Размеры элементов. Школа юного снайпера

Одним из основополагающих законов разработки интерфейса пользователя является закон Фитса<sup>1</sup>. Для ускорения работы элемент интерфейса не должен быть маленьким и находиться в отдаленных уголках экрана. В мелкие детали трудно попасть, однако это не значит, что этот элемент надо делать огромным (рис. 7.8).

Введите текст:

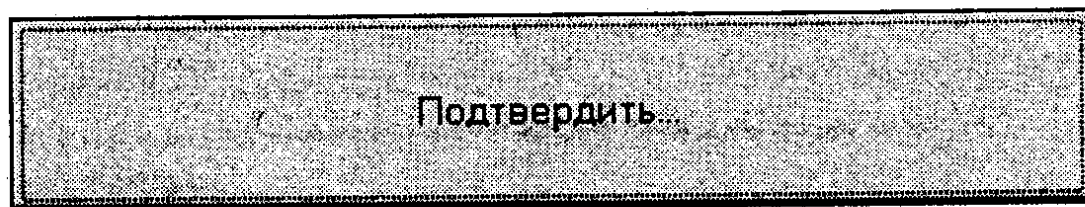


Рис. 7.8. Неразумное увеличение элементов интерфейса и выглядит нелепо, и места занимает много

<sup>1</sup> Время достижения цели обратно пропорционально размеру цели и расстоянию до нее.

Место на экране слишком дорого, особенно если нужно поместить «море информации». Проблемы с использованием мелких деталей появляются из-за физиологических ограничений и точности аппаратного обеспечения. Редкий пользователь сможет попасть курсором в точку размером  $2 \times 2$  пикселя. Такие проблемы особенно актуальны для графических редакторов, где рядом могут находиться совершенно разные объекты. Порой выделить нужный из них удастся только с пятой-шестой попытки (рис. 7.9). Нельзя делать элементы интерфейса слишком мелкими и близкими друг к другу — это первое правило.



Рис. 7.9. Я ни разу, НИ РАЗУ за многие годы моего знакомства с текстовым редактором Microsoft Word не смог ухватить этот ползунок с первой попытки

Второе гласит о том, что пользователь может не достигнуть своей цели только потому, что не будет попадать в нужное место экрана. Иначе говоря, если в навигации сайта использовать элемент расширения информации не в виде ссылки Подробнее..., а просто в виде многоточия «...», то число пользователей, полностью прочитавших такую новость, уменьшится на порядок. Кто-то просто не заметит многоточия, а кто-то не захочет даже пытаться проверить, гиперссылка ли это.

Для обеспечения удобной работы с мелкими деталями в некоторых случаях можно их искусственно увеличивать. Вы можете по каким бы то ни было соображениям (эстетическим, дизайнерским) решить, что деталь должна остаться мелкой (например, элемент расширения информации в виде обычной стрелки). В таком случае сделайте его не картинкой размером  $3 \times 3$  пикселя, к которой прикреплена гиперссылка, а той же стрелкой  $3 \times 3$  пикселя, только вокруг добавьте пустое поле (рис. 7.10). Тогда, целясь в картинку, пользователь может и не попасть точно в цель, но наверняка попадет в поле, где также есть гиперссылка.



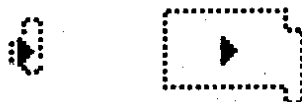


Рис. 7.10. Два варианта одной и той же гиперссылки показаны в момент, когда на них установлен фокус в HTML-форме. И куда, по-вашему, легче попасть?

### 7.3.2. Откуда я пришел?

Предоставление списка уже посещенных разделов, результатов поиска, хранение некоторой предварительной информации о пользователе — все эти приемы также ускоряют работу. Можно провести аналогию с обычными приложениями, в которых используется пункт меню Recent files (последние файлы). В нем содержится список файлов, к которым обращались в последнее время. Если необходимо вернуться к работе над файлом, не нужно выискивать его в файловой системе, а достаточно вызвать его из соответствующего пункта меню (рис. 7.11). Таким же образом

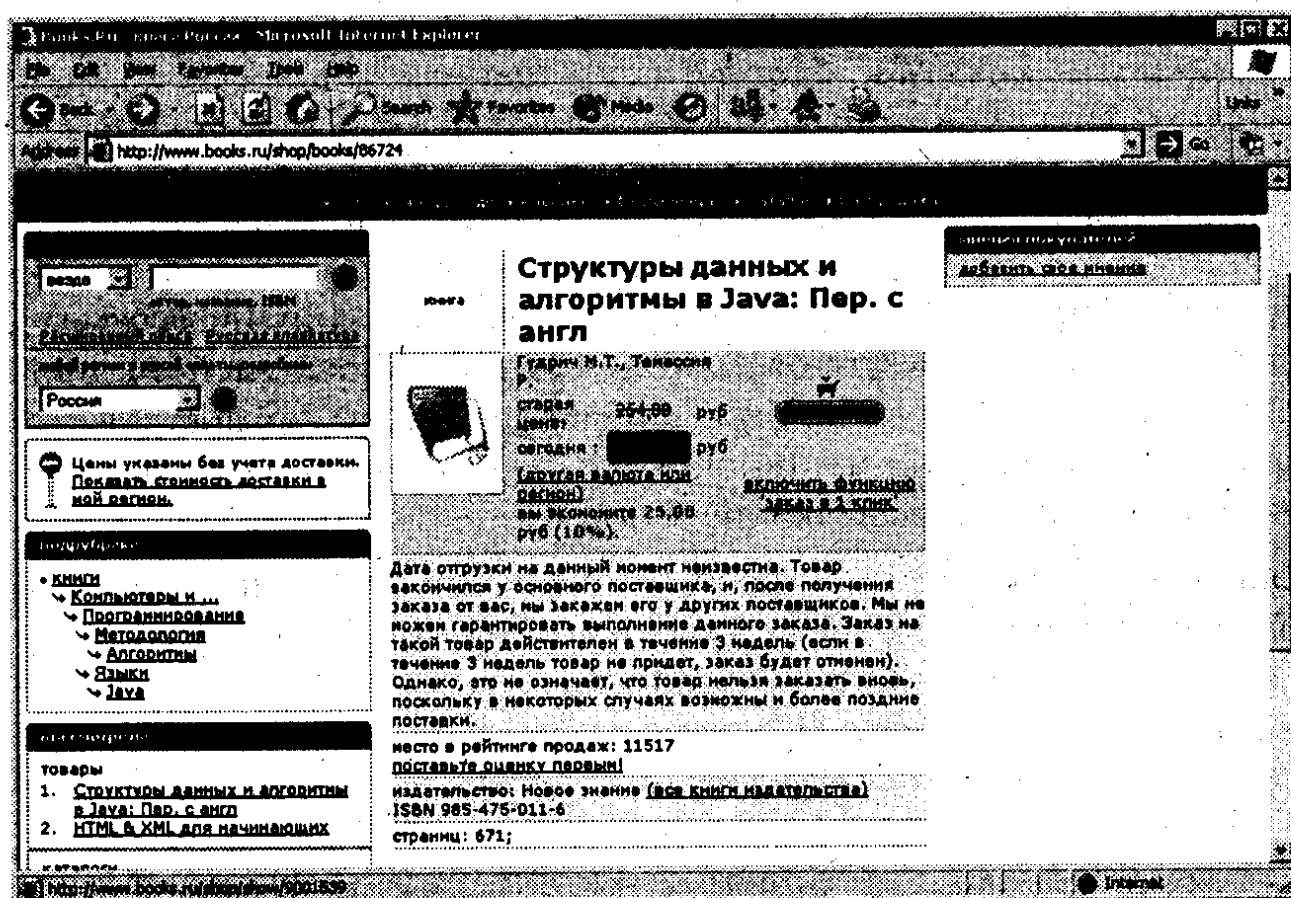


Рис. 7.11. В левой части страницы на books.ru расположены список подразделов текущего раздела, схожие разделы и список просмотренных ранее товаров

работает и список уже посещенных разделов. Если я искал в интернет-магазине книгу и нашел практически то, что хотел, но решил «походить по базару, поискать дешевле», то в случае фиксирования результатов поиска пусть даже в рамках одного сеанса работы я могу вернуться к описанию этой книги. Гиперссылка Вы искали ранее избавляет от необходимости повторять свой поиск, всякий раз ожидая, пока система сгенерирует результаты запросов, которые я уже сделал десять минут назад.

### 7.3.3. Сокращенная и полная форма

Для использования следующего приема ускорения работы пользователей необходимо не только знание основ построения интерфейса, но и знание предметной области системы. **Предметная область** — та часть функций и понятий, которая переносится в программную систему из процесса выполнения какой-то задачи в реальной жизни. Предметная область диктует, что наиболее значимо для работы пользователей, именно эта информация должна быть предоставлена прежде всего. Для отображения данных используются две формы: сокращенная и полная. **Сокращенная форма** экономит место на экране, занимаемое информацией, ускоряет восприятие и упрощает работу. В сокращенной форме должны быть только самые важные сведения (рис. 7.12).

**2. HTML & XML для начинающих**  
 Моррисон М.  
 цена 127,00 руб (другая валюта или регион)

Склад в Москве [-]. Ожидаемое поступление: 03.05.2004; планируемая отправка: 04.05.2004  
 Склад в С.-Петербурге [-]. Ожидаемое поступление: 29.04.2004; планируемая отправка: 30.04.2004

оценка покупателей ★★★★★ (голосов: 1)

Практическое пособие. Даже те из вас, кто не помышляет о собственном Web-узле, вероятно, уже поняли, что в будущем им не... далее

#### HTML & XML для начинающих



Моррисон

цена 127,00 руб (другая валюта или регион)

Склад в Москве [-]. Ожидаемое поступление: 03.05.2004; планируемая отправка: 04.05.2004  
 Склад в С.-Петербурге [-]. Ожидаемое поступление: 29.04.2004; планируемая отправка: 30.04.2004

место в рейтинге продаж: 594  
 оценка покупателей ★★★★★ (голосов: 1)

издательство: Эком (все книги издательства)  
 серия: Для начинающих  
 дата выхода: октябрь 2002  
 ISBN 5-7163-0100-2  
 страниц: 384  
 обложка: мягкая  
 иллюстрированное издание

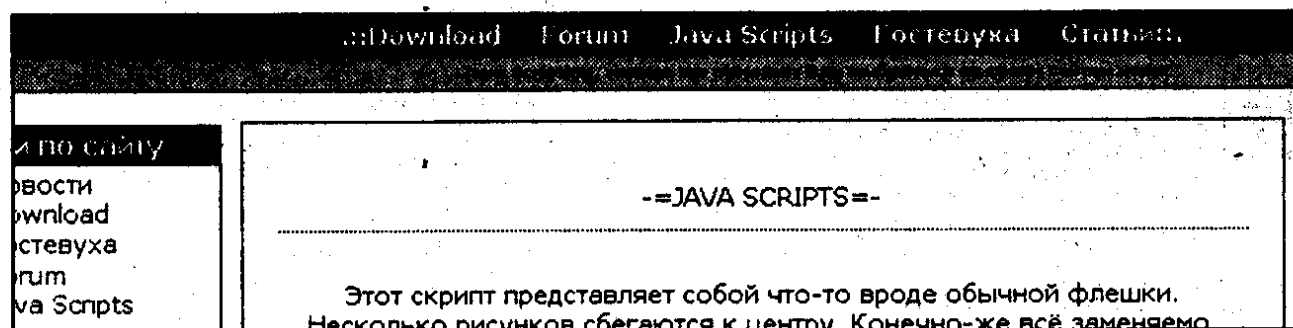
Рис. 7.12. Сокращенная и полная форма отображения информации о книге на books.ru

**Полная форма** — развернутое описание объекта и его свойств, она содержит все доступные сведения о нем.

Сокращенная форма хороша в том случае, если набор полей объекта, которые видит пользователь в результатах поиска, удовлетворяет его информационные потребности. Если вас интересует в первую очередь название и адрес организации, то использование сокращенной формы, в которой вам покажут фамилию руководителя и номера служебных машин, вашу работу не только не ускорит, но еще и замедлит (особенно, если разработчик не предусмотрит в своей системе быстрый переход от сокращенной формы к полной).

### 7.3.4. Не отвлекайте меня от работы!

Вы пытались когда-нибудь готовиться к экзамену в комнате с работающим телевизором, или, может, вы пробовали работать, когда ваш коллега мечется перед вашим столом, как загнанный зверь? Когда вас отвлекают, сосредоточиться невозможно. Роль такого мечущегося коллеги может исполнять прыгающий по странице текст или бегущая строка в нижней части экрана. На его месте может быть и «о-чудо-javascript-а» — ползущая за вами панель с кнопками или баннером, которая подпрыгивает, а затем останавливается в одном и том же месте экрана, как только вы перестаете крутить колесо прокрутки. Это отвлекающие элементы, их нужно уничтожать, если только они не предназначены для развлечения (рис. 7.13).



**Рис. 7.13.** На сайтах зачастую помещают различные изыски, созданные при помощи JavaScript. Бегущая строка под горизонтальным меню на `lia54.h11.ru` сразу же отвлекла меня, и я не смог работать с информацией, пока не прочел всю надпись. А потом постоянно поглядывал, не изменилась ли она

Такие элементы всегда уводят пользователя от информации, потому что он неосознанно следит взглядом за тем, что движется. О какой сосредоточенности на задаче может идти речь? Долой бегущие строки и меню-попрыгунчики с сайтов, где пользователи занимаются делом! Пользуйтесь этими приемами, если вы намеренно желаете отвлечь пользователя или хотите ему срочно что-то сказать, но не в иных случаях.

## **7.4. Комфортность**

### **7.4.1. Комфортность в жизни и в Web**

Комфортность как элемент юзабилити по своей сути не является управляющей частью интерфейса пользователя. Она носит скорее характер дополнительных удобств.

Представьте себе два загородных коттеджа.

Один из них стоит на площадке, покрытой бетонной плиткой, нет ни сада, ни веранды, ни навеса, чтобы спрятаться от солнца в жаркий день. Холодная вода подается раз в сутки в течение двух-трех часов, а нагревать ее специальным агрегатом, объем которого ограничен. Кровати узкие, с сеткой, провисающей почти до пола. Кухонная плита с газовым баллоном, он тяжелый и большой, менять его положено раз в два месяца. На полах нет ковров, и босиком по ним ходить холодно, особенно зимой.

Второй коттедж в зеленом тенистом саду. Горячая и холодная вода из водопровода круглосуточно. Удобные широкие кровати. Повсюду мягкие пушистые ковры, стоит телевизор, музыкальный центр. На кухню газ подведен централизованно, и перебоев с его подачей не бывает.

Жить можно и там, и там. Только ощущения разные. Первый вариант — школа выживания, второй — номер-«люкс».

### **7.4.2. Разрешение монитора**

До сих пор не существует единого разрешения монитора у всех пользователей. Кроме того, вы не заставите пользователя специально для просмотра вашего сайта переключаться в режим 1024×768, если он привык к 800×600. Все прекрасно знают, что

поддержка нескольких разрешений монитора — занятие нетривиальное, однако подстраиваться придется вам, если вы хотите, чтобы ваши посетители чувствовали себя комфортно.

Необходимо добиться того, чтобы ваш сайт практически одинаково выглядел при различных разрешениях монитора (естественно, речь идет только об основных).

### 7.4.3. Браузеры

Большая часть информации на странице остается доступной как при различных разрешениях монитора, так и во всех популярных браузерах. Однако то, как выглядит эта информация, порой оставляет желать лучшего. Хорошая идея, часто и эффективно используемая, может сыграть злую шутку. Как-то я попал при помощи браузера Mozilla на сайт одной компании, где был просто сражен наповал «висящей» перед основным информационным блоком совершенно непонятной таблицей. Заинтересованный таким поворотом событий, я стал экспериментировать: можно ли все-таки прочитывать этот информационный блок, работают ли гиперссылки на этом куске таблицы. Работать с содержимым я все-таки смог, но в другой раз у меня не будет ни времени, ни желания разгадывать подобные головоломки.

Я скопировал в буфер обмена адрес этой экстравагантной страницы и открыл ее во «всеядном» MS Internet Explorer. Среди прочих достоинств он имеет свойство, нещадно эксплуатируемое любителями нестандартных решений, — этот браузер как будто построен по принципу «отображу все, а что не отображается, то все равно отображу». Например, в спецификации HTML у тега `<table>` не существует атрибутов «height» и «width»:

`<table height=«100» width=«100%»>`, но редко кто пишет страницы, строго следуя спецификации;

`<table style=«height:100%; width:100%»>`.

Internet Explorer отреагирует на обе записи одинаково, потому, что он «всеядный», в нем работают практически любые «изыски».

И вот, нисколько не удивившись, я увидел, что в Internet Explorer все выглядело красиво и стройно, а «неизвестная таблица» на самом деле принадлежала выпадающему меню, построенному каким-то особо хитрым способом.



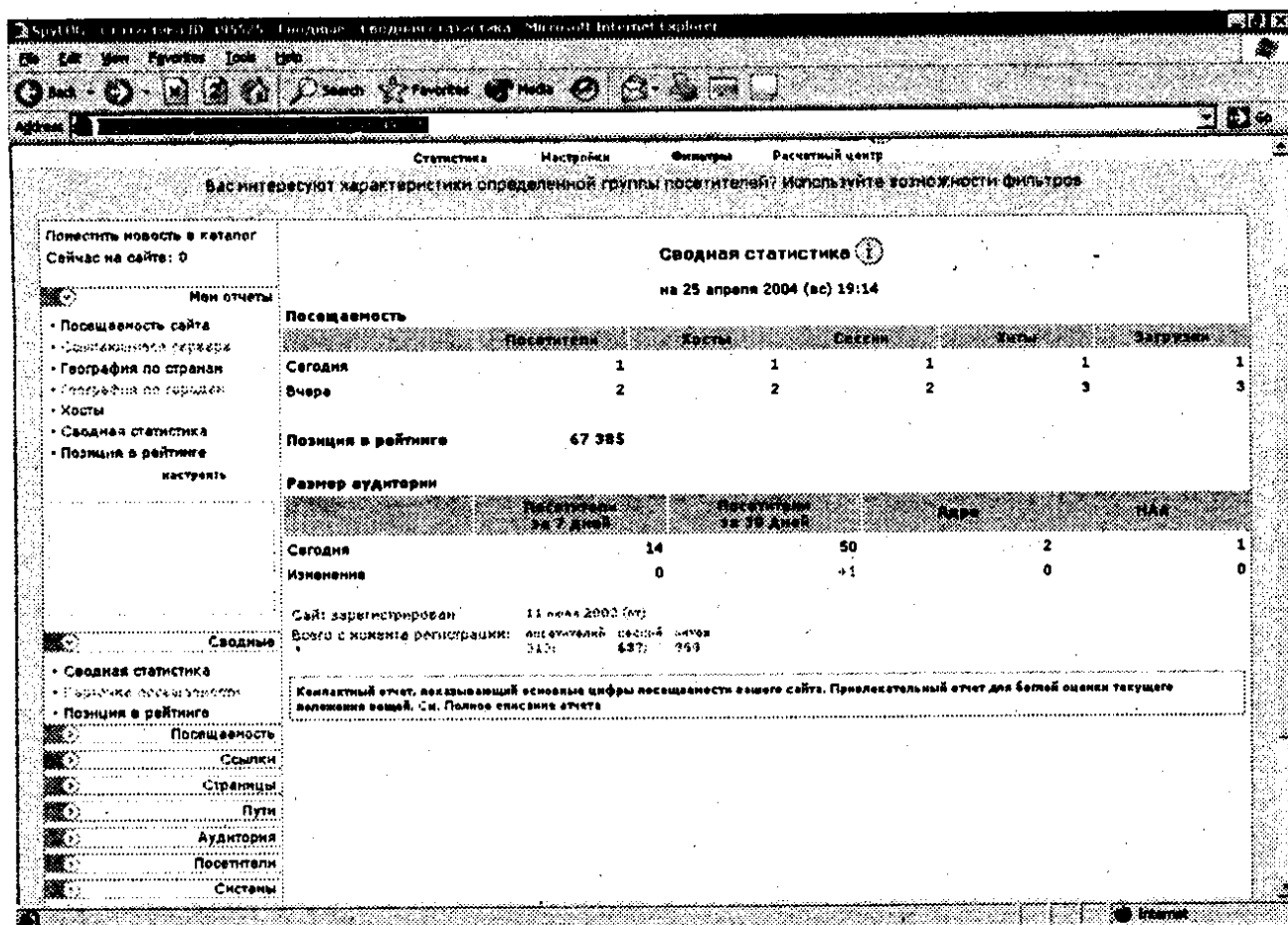


Рис. 7.15. Вот так должна выглядеть страница сводной статистики на SpyLog

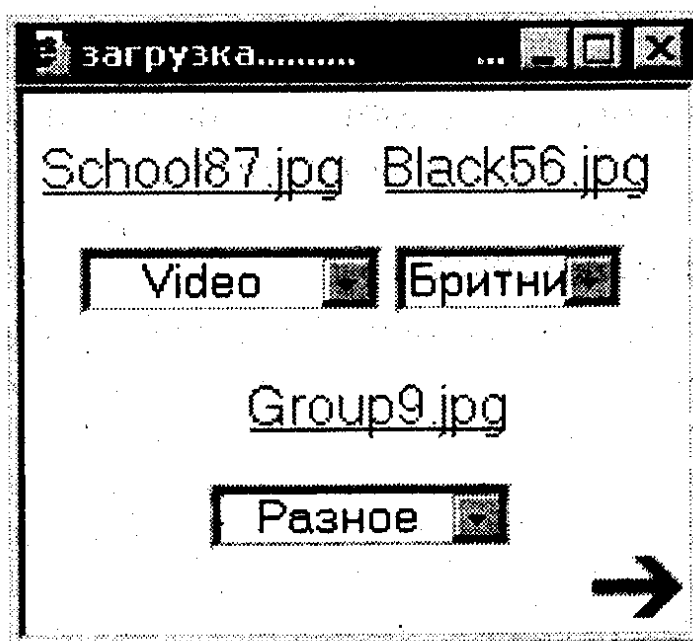


Рис. 7.16. Зайдешь, бывало, почитать анекдоты на <http://fomenko.ru>, а там, словно в сказке, как выпрыгнет, как выскочит...

У многооконной организации Web-интерфейса, особенно если она используется в интернете, а не интранет-сетях, есть ряд недостатков:

- порой трудно разделить ситуации на те, в которых следует новые окна открывать «без позволения» пользователя, и те, в которых нужно дождаться его запроса;
- всплывающие окна часто произвольно «убиваются» опытными пользователями;
- может случиться так, что появившееся окно можно просто не заметить во время работы, если оно открывается на заднем плане;
- иногда и обновление содержимого вспомогательного окна может происходить практически незаметно, по этой причине пользователь может решить, что система не отвечает на его запросы.

#### 7.4.5. Полосы прокрутки

Если приходится работать со страницами на несколько экранов, то независимо от того, дублируете ли вы в нижней части элементы навигации или нет, вы вынуждены долго пролистывать содержимое. Использование элемента Наверх может ухудшить дизайн страниц, а в некоторых случаях все равно не решит проблемы, ведь пользователь должен всегда иметь возможность видеть «контекст» своей работы (навигационную строку или краткое описание текущего действия). В подобных случаях иногда верстку делают таким образом, чтобы основные навигационные элементы не перемещались при прокрутке. Исключением являются элементы, «плавающие» по странице. Меня они невероятно раздражают. Они не прокручиваются вместе с остальным содержанием и отвлекают от работы. Я советую избегать таких элементов-«попрыгунчиков».

Любой предмет, находясь не в нужное время и не в нужном месте, может препятствовать выполнению работы. Содержимое страницы должно прокручиваться только одной полосой прокрутки (скроллером) и только в одном направлении (вверх/вниз) (рис. 7.17).

Ситуация, когда необходима горизонтальная полоса прокрутки, крайне нежелательна. Также следует избегать двух и более вер-



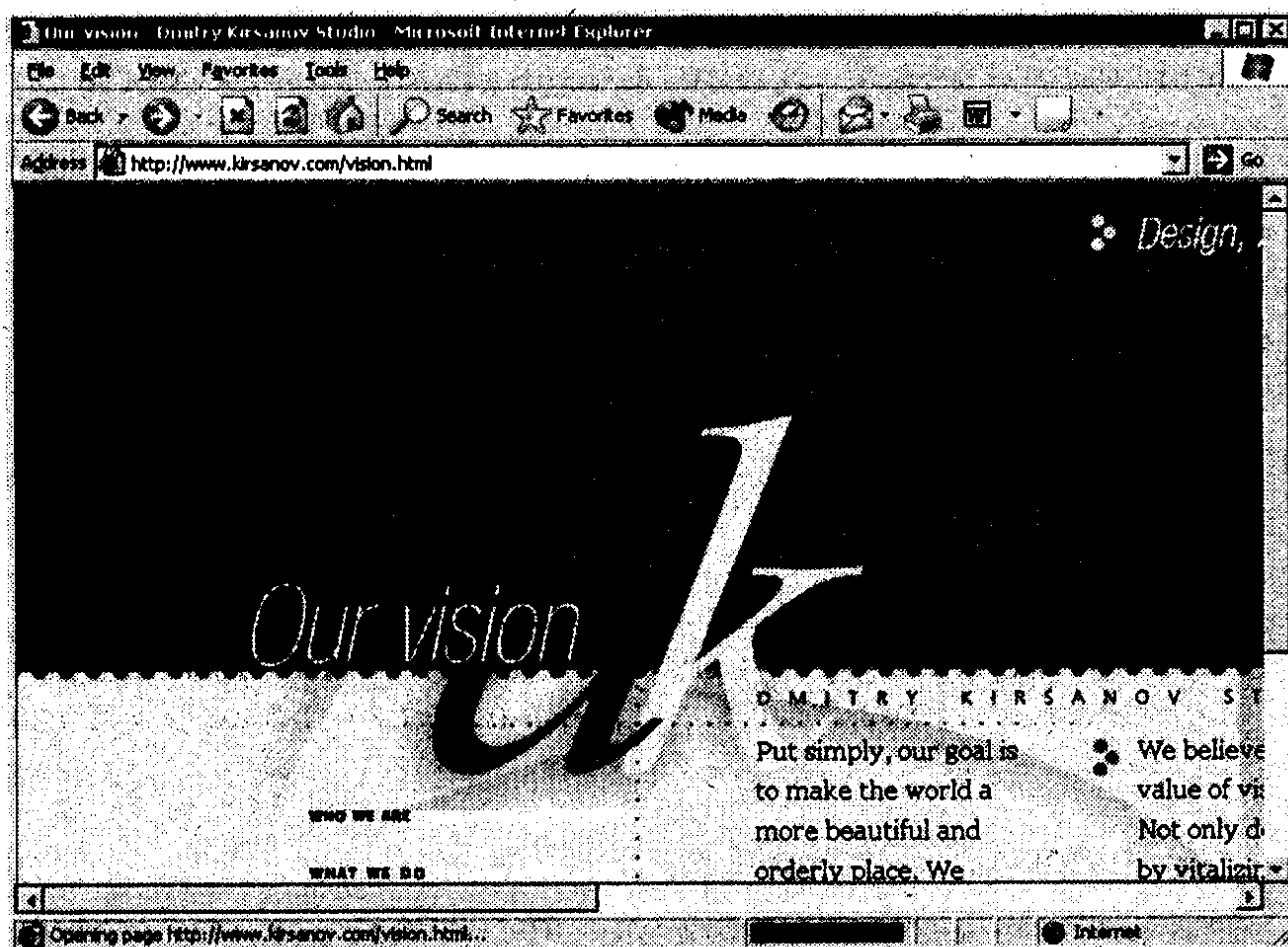


Рис. 7.17. Как вы думаете, удобно ли работать на сайте знаменитого автора книги «Web-дизайн» Кирсанова при наличии двух таких полосок прокрутки? Попробуйте

тикальных. Мне приходилось сталкиваться со страницами, на которых надо орудовать сразу двумя полосами прокрутки. Крайне неудобно, прошу заметить. А об увеличении скорости работы пользователей не может идти и речи. Такие вольности можно позволить себе только там, где речь идет о дизайне, а пользователи не работают.

#### 7.4.6. Забытые герои

Факторов, ускоряющих работу пользователя и упрощающих ее, настолько много, что некоторые можно упустить из виду. Я едва не забыл упомянуть о клавише Ввод. Помните, что очень часто пользователи работают с сайтом или поисковым порталом при помощи этой клавиши. Они печатают запрос, заполняют форму и жмут на Ввод. Одного этого нажатия должно быть достаточно

для продолжения работы с системой. Не заставляйте пользователя тянуться к мыши, а потом еще и искать нужное место для «клика».

Еще один «забытый герой» — шрифт, размер которого установлен в браузере по умолчанию. Некоторые любят мелкие буквы, некоторые — крупные. Ваш чудесный сайт может развалиться на куски, если вы не проверили его поведение при установках шрифта Самый маленький и Самый большой. Как правило, если со страницей все нормально при этих граничных условиях, то в остальных случаях проблемы также не возникают. Здесь есть одно «но». Часто разработчики используют размер шрифта, жестко заданный при помощи стилей. Это не совсем корректно по отношению к пользователям, потому что это типичная «уровнировка». За тысячи посетителей дизайнер решил, как им удобнее читать. Он поставил в одинаковые условия и человека с ослабленным зрением, и пользователя, у которого «зоркий орлиный глаз».

Как видите, наличие хотя бы только этих «забытых героев» говорит о том, что всего не упомнишь. Каким бы ни был ваш опыт, он подскажет вам то, о чем я мог забыть.

#### **7.4.7. Считаю шаги**

Сокращения времени работы можно добиться за счет сокращения количества шагов до поставленной цели. В качестве примера рассмотрим главное меню сетевого телефонного справочника.

Интернет-справочник позволяет искать телефоны как сотрудников и подразделений, так и самих организаций, при этом формы поиска всех этих объектов отличаются. Он позволяет также создавать эти объекты, получать по ним отчеты, получать справку и искать альтернативным способом (например, навигация по каталогу).

Как перейти к поиску организации?

Учитывая небольшое количество объектов системы, можно предложить несколько вариантов.

##### **1. Создать пункт главного меню Поиск (рис. 7.18)**

После выбора этого пункта предлагается уточнить объект одним из двух способов — выбором объекта из списка (рис. 7.19)

Главное меню

Регистрация информации

Поиск информации

Печать сведений

Дополнительно...

Рис. 7.18. Первый шаг — выбор действия Поиск

Поиск

Поиск сотрудника

Поиск подразделения

Поиск организации

Рис. 7.19. Второй шаг — выбор объекта действия Поиск

или набора вкладок (рис. 7.20).

Поиск

Сотрудник	Подразделение	Организация	Дополнительно
Фамилия	<input type="text"/>		
Имя	<input type="text"/>		
Отчество	<input type="text"/>		
<input type="button" value="Искать"/>			

Рис. 7.20. Второй шаг — переход на нужную закладку

## 2. Показать форму со вкладками

По умолчанию целесообразно загружать открытой ту вкладку, которая чаще всего используется для поиска (например, Сотрудник). Это хорошее решение, но лишний уточняющий шаг убран только для тех, кто ищет сотрудников.

## 3. Вынести в главное меню опции Искать сотрудника, Искать подразделение, Искать организации

Человек сразу попадет в нужную ему форму поиска. Третий вариант предпочтительней. Теперь уточним его. Есть соблазн сгруппировать сходные действия, но это ошибочное решение. Если мы выделим композиционные блоки Найти, Регистрировать, Печатать, человек быстро перейдет к блоку поиска, но остановится на выборе объекта: он будет высматривать слово «поиск», которое будет написано три раза. Поэтому пользователя необходимо направлять «от объектов», а не «от действий», в данном случае разделить главное меню на блоки Сотрудники, Подразделения и Организации. Тогда пользователь быстро найдет блок Организации, обнаружит в нем слово «поиск» и «кликнет» на нем. Блоки меню целесообразно расположить исходя из их важности и частоты использования в соответствии с направлением восприятия (слева направо и сверху вниз) (рис. 7.21).

Главное меню			
<b>Сотрудники</b>		<b>Подразделения</b>	
<u>Регистрация информации о сотруднике</u>		<u>Регистрация информации о подразделении</u>	
<u>Поиск информации о сотруднике</u>		<u>Поиск информации о подразделении</u>	
<u>Печать сведений о сотруднике</u>		<u>Печать сведений о подразделении</u>	
<b>Организации</b>		<b>Дополнительно</b>	
<u>Регистрация информации об организации</u>		<u>Регистрация категорий</u>	
<u>Регистрация информации об организации</u>		<u>Архив отчетов и бланков</u>	
<u>Печать сведений об организации</u>		<u>Помещение файлов</u>	

Рис. 7.21. Действия сокращены на один шаг

## 7.5. Обучаемость

### 7.5.1. Стимулы и предпосылки

Человек учится чему-либо только тогда, когда он этого хочет. В вузе он учится, если хочет получить высшее образование. Класть плитку он учится, если хочет обустроить свой дом или самостоятельно сделать ремонт, игре на пианино — если ему нравится музыка и он горит желанием освоить какой-нибудь музыкальный инструмент (или если родители обещали ему купить компьютер, когда он сыграет им «Лунную сонату»). Обучаемость и желание учиться могут быть совершенно разными, но при наличии не очень большого стимула к анализу и изучению системы прельстить пользователя можно только легкостью обучения (можно даже сказать «незаметностью»). Если ему не приходится сильно напрягаться, чтобы понять правила работы и изучить функциональность, он, так уж и быть, согласится поработать с вашим продуктом.

Во всяком программном продукте существует набор функций, о которых пользователь может не узнать, не погружаясь чересчур глубоко. Тем не менее этот набор может облегчить работу с системой. Например, я до определенного момента не знал нескольких приемов работы в графическом редакторе Adobe Photoshop. Так, если во время работы с изображением нажать и удерживать клавишу Пробел, то обычный курсор изменится на инструмент Рука, которым можно приблизить части изображения, скрытые за пределами области экрана. А если после первого щелчка инструментом Ластик нажать клавишу Shift и щелкнуть в другой точке экрана, то будет стерта вся область соответствующей ширины между этими двумя точками. Я не мог узнать этих вещей, потому что сама программа мне этого не подсказала, к справке я не обращался, поскольку мне удалось изучить этот редактор «методом научного тыка» еще с третьей версии, а книг по Photoshop я не читал. Эти приемы мне подсказал более опытный пользователь, и я стал работать эффективнее. Однако незнание тонкостей практически не мешало работать. Вывод напрашивается сам: пока пользователю кто бы то ни было — система или человек —

не расскажут о некоторой функции, она остается недоступной. Между жизненно важными для системы и пользователя функциями и сопутствующими находится тонкая грань, которую необходимо чувствовать. О жизненно важных необходимо сообщать достаточно «громко», а об остальных — ненавязчиво, аккуратно, чтобы не отвлекать пользователей.

Если ваши посетители легко и быстро научатся работать с вашей системой, они навряд ли захотят уйти к другому ресурсу и изучать его с нуля. Я не хочу переходить к графическому редактору Adobe Illustrator, потому что уже знаю схожий Corel Draw. Зачем мне изучать что-то новое, если у меня уже есть подходящий инструмент? Если мне скажут, что в Adobe Illustrator удобнее работать, я отвечу, что мне удобнее работать в том продукте, который я знаю и к которому давно привык, а не в том, который удобнее. Только массивная наглядная агитация заставит меня подумать, прав ли я.

Для того чтобы пользователь понимал назначение того или иного действия, необходимо объяснять, что это за действие, для чего оно предназначено и к чему приведет. Мотивируйте поступки системы и пользователя, и тогда он быстро научится работать с вашим продуктом и будет все понимать, а не совершать «ритуалы». Такие объяснения запоминаются надолго.

### 7.5.2. Метафора

Существует особый подход к построению интерфейса с использованием метафор. **Метафора** — некоторая аналогия с процессами, происходящими в реальной жизни, поддержанная в интерфейсе системы. Метафорой является, например, Корзина в операционных системах, т.е. пользователь понимает, что в корзину обычно выбрасывают ненужные вещи. Если ярлык перетащить в Корзину, то понятно, что его таким образом отправляют в «мусор», удаляют. На самом деле метафоричность таких процессов достаточно ограничена. Даже всем известный рабочий стол — тоже не самая удачная метафора (кстати, почему-то считается, что рабочий стол украшают «обои», а не «скатерть»). Метафора используется не только для обозначения действия, но и для изображений на пиктограммах (дискета, принтер и знак вопроса). Других изображений, однозначно передающих свой смысл при помощи метафоры, я не знаю.

Метафора не всегда удобна, т.к. четкое следование жизненным аналогиям может навредить интерфейсу и сделать его менее удобным и понятным. Отступление от метафоры в некоторых моментах делает ее нечеткой аналогией, которую пользователю надо додумывать. Кроме того, зачастую метафорой считают вовсе не метафору, а так называемые идиомы.

### 7.5.3. Идиома

**Идиома** — часть интерфейса, которая устойчиво ассоциируется с некоторым действием и несет в себе определенную идею. Например, гиперссылка (текст синего цвета с подчеркиванием) сообщает, что, щелкнув на ней мышью, пользователь перейдет на другую страницу либо к другой части содержимого текущей страницы, если это якорь. Пиктограмма с крестиком значит «закрыть», какого бы цвета, толщины, величины этот крестик не был. Полоса с бегунком говорит о том, что с ее помощью можно прокрутить содержимое (это понятно даже в некоторых компьютерных играх, где прокрутка может быть нарисована, например, в виде сложного кузнечного изделия, перемещающегося по стилизованным воротам). Идиома — устойчивое словосочетание, значение которого не обязательно следует из значений составляющих его слов (например, смотреть сквозь пальцы, работать спустя рукава). Аналогичным образом идиома оказалась применима к визуальным элементам.

### 7.5.4. Очевидность

**Очевидность** — качество интерфейса, обеспечивающее корректное взаимодействие пользователя и системы. Она основана на принципе «ручки — чтобы крутить, кнопки — чтобы нажимать». Если вставить в интерфейс краны с синим и красным цветом, то пользователь на основе идиомы из реальной жизни и очевидности будет, не пробуя рукой кран (он не может этого сделать на экране монитора), безошибочно включать нужную воду.

### 7.5.5. Привычность решений (роль стандартов)

В Web существуют условные стандарты, которые определены привычкой пользователей. Являются ли эти стандарты факторами, увеличивающими юзабилити, достаточно серьезный вопрос. Но вот в обучаемости стандарты играют роль своего рода идиом.



Рассмотрим один из примеров. На очень многих Web-сайтах навигационное меню расположено с левой стороны страницы. С точки зрения физиологии человека и закона Фитса человек быстрее перейдет от полосы прокрутки к меню, если оно будет с правой стороны. Однако все привыкли, что если есть боковое меню, то оно слева. При отсутствии альтернативного варианта навигации нарушение этого шаблона вызовет недовольство у пользователя и даже замедлит работу, потому что он по привычке сразу перемещает фокус внимания к гипотетическому меню слева, а потом возвращается туда, где оно действительно есть. Руки работают медленнее мысли, по крайней мере, у большинства. То же касается и логотипа с переходом на главную страницу слева вверх. Почти все уже давно привыкли, что он там. Попробуйте поместить его в другом месте, и пользователь потратит дополнительные секунды, чтобы отыскать его. Никто не захочет запоминать, что у вас на сайте логотип с переходом на домашнюю страницу внизу или меню с основными разделами «выплывает» из какой-то странной пиктограммы в углу страницы.

## **7.6. Запоминаемость**

### **7.6.1. Автоматизм, привычка и простота**

Мне нравятся программные продукты Adobe Systems Inc. Взять, к примеру, Photoshop. Его гениальная особенность в том, что набор инструментов и способов их использования достаточно прост, а эффекты, которые можно получить при помощи элементарных действий, шикарны. Это то, к чему надо стремиться при разработке интерфейса. Любая сколь угодно сложная задача может быть решена путем совершения последовательности простейших действий.

Другой пример — изучение языков программирования. В большинстве современных языков программирования, например, в Java, существует некоторый набор лексики и стандартных приемов, который позволяет работать, думая не о лексических правилах языка, а только об алгоритмах, шаблонах, конструкциях.

Я веду к тому, что пользователь, обладающий информацией о часто используемых простых стандартных решениях, легко по-



нимает, что ему необходимо сделать для решения задачи, как он это может сделать и какими приемами пользоваться.

Обеспечьте пользователей простыми и стандартными (привычными) способами решения сложных проблем, и они намного быстрее разберутся и запомнят, как работает система. Этот подход гораздо эффективнее, чем придумывать мудреные способы решения какой-либо проблемы, тем более что проблемы имеют свойство проявляться во многих местах и ипостасях.

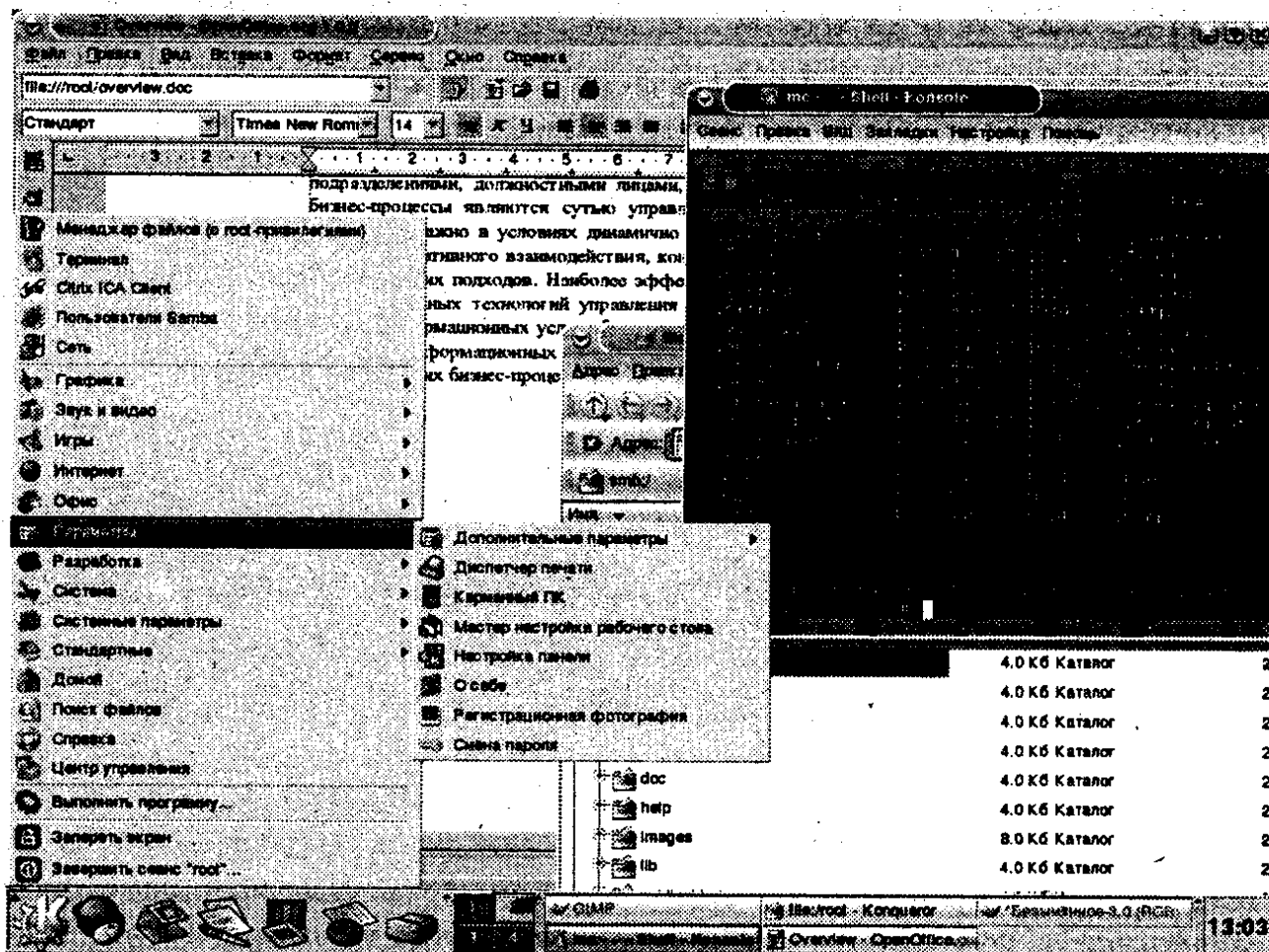
Пользователь всегда к чему-то привыкает. Даже при наличии большого количества средств совершения действия он всегда выбирает один, привычный для него, и если не находит такого в другой программе, страдает. Такие привычки часто возникают из-за формирования «стандартов» (которые вовсе не стандартизованы) из многократно используемых способов подачи того или иного материала.

Сильнейшее, почти маниакальное привыкание к некоторым элементам интерфейса, например, кнопке закрытия окна, можно продемонстрировать на простом примере. Когда я смотрю на картинку с изображением какого-либо диалогового окна, первое, что я делаю, чтобы перейти к следующему действию, щелкаю на кнопке закрытия окна. Оно не закрывается (еще бы, это же картинка, а не настоящее окно). Я снова щелкаю. Оно по-прежнему на месте. Я начинаю раздражаться и только после третьего-четвертого раза понимаю, что попался в ловушку собственных привычек. И уже тогда действительно понимаю, что же происходит и кто неправильно работает.

Посмотрите внимательно на то, как может выглядеть операционная система Red Hat LINUX для рядового пользователя (рис. 7.22).

Если пользователь имеет опыт работы с Windows, он достаточно быстро переучится, т.к. в основном идиомы операционных систем схожи.

Следует сказать, нестандартное решение не всегда неправильное. По утверждению знаменитого специалиста по интерфейсам, автора большинства идей в компьютерах Apple Macintosh Джефа Раскина (Jef Raskin), при возможности построения интерфейса, который будет простым и удобным для пользователей, но потребует некоторого дополнительного обучения, лучше



**Рис. 7.22.** Работать с незнакомым интерфейсом кажется невозможным только с первого взгляда. Не погружаясь в настройки операционной системы и самого компьютера, можно работать так же эффективно, как и со знакомой системой

создать такой интерфейс, чем сложный и громоздкий, но который будет использовать привычные решения и не потребует дополнительного обучения. Я полностью разделяю это мнение. В качестве примера я приведу интерфейс простого редактора универсального языка моделирования UML (рис. 7.23). В некоторых случаях этот редактор гораздо удобнее и проще в работе, чем такие знаменитые «монстры», как MS Visio или Rational Rose.

### 7.6.2. Работа над ошибками

Человек склонен повторять свои действия, мысли, слова, причем точность не зависит от того, сколько времени прошло. В этом я убедился на собственном опыте. При использовании программ, с которыми я работаю крайне редко, я делаю те же ошиб-

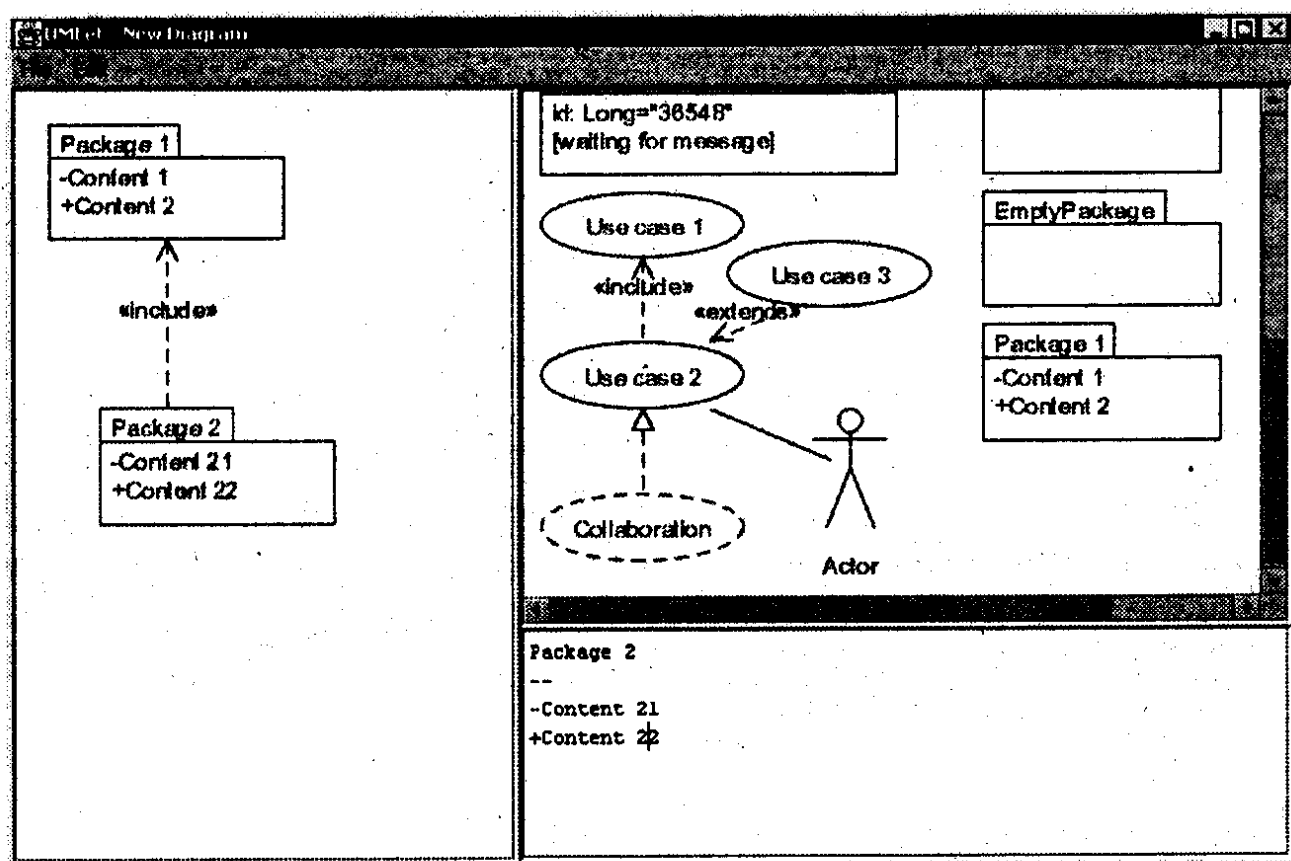


Рис. 7.23. Редактор UMLet. Для создания фигуры необходимо всего лишь выбрать ее шаблон в правой верхней части. Редактировать свойства фигуры можно в обычном текстовом режиме справа внизу, а сама рабочая область — слева

ки, что и ранее. Разница лишь в том, что, исправляя их, вспоминаю, что уже их совершал.

Еще один пример. Когда-то давно, когда компьютеры были еще редким явлением, я печатал тексты на пишущей машинке, потом сканировал их, распознавал и вручную исправлял в электронном варианте огрехи распознавания. Я читал часть предложения, исправлял ошибки, удалял совершенно нечитаемый кусок и вместо него набирал на клавиатуре «из головы» то, что по логике повествования должно было быть написано. И когда я сверялся с печатным оригиналом, всякий раз удивлялся, что, не глядя в бумагу, воспроизводил фразу с точностью до 90–95%. Связано это не с тем, что у меня хорошая память, а с тем, что я думаю приблизительно одинаково. Если в первый раз мною был совершен набор действий, который привел к ошибке, то он будет совершен и второй раз.

Основываться на постулате «единожды сделав — снова повторяю» можно только при наличии одного возможного способа достижения цели, осуществляемого набором относительно простых действий, понятных пользователю.

Не стоит надеяться и на то, что пользователи, сделав ошибку в работе с вашим сайтом, не повторят ее в следующий раз. Ошибки в интерфейсе необходимо отыскивать и исправлять.

### 7.6.3. Право выбора всегда усложняет жизнь

Я уже говорил о том, что предлагать несколько решений одной проблемы — плохой стиль. Когда пользователь вспоминает, как он выполнил то или иное действие в прошлый раз, он мечется между различными способами, а то решение проблемы, которое он помнит, на самом деле состоит из фрагментов двух-трех способов.

Есть такая простенькая компьютерная игра «Кубики». В ней отменить ход можно двумя способами: правой клавишей мыши и соответствующей пиктограммой в графическом меню в верхней части окна. Я понимаю, что способ «правого клика» быстрее и, наверное, удобней (мышь двигать не надо — всего лишь слабое усилие безымянного пальца), но рука сама тянется к верхнему меню. Иногда, когда я успеваю подумать о том, какой способ предпочтительней, я могу заставить себя нажать правую клавишу, но уже проделав большую часть пути к пиктограмме. И самое главное и худшее, что я не помню ни одного раза, когда бы я бессознательно, по привычке, выбрал один из вариантов. Это нарушение правил построения пользовательского интерфейса — интерфейс должен формировать привычки, а не отучать от них.

### 7.6.4. *Sapienti sat*<sup>1</sup>

Следующая рекомендация носит абстрактный характер. Прозрачная и логичная навигационная модель позволяет пользователям легче понимать суть действий и, следовательно, лучше их помнить и быстрее обучаться. За исключением использования ассоциативной памяти (например, цвета радуги, обозначенные первыми буквами фразы «Каждый Охотник Желает Знать, Где

<sup>1</sup> Разумному достаточно (лат.).

Сидит Фазан), вызубрить — хуже, чем разобраться. Если вы бездумно заучили наизусть доказательство теоремы Пифагора, то вы забудете его быстро и бесповоротно. Если вы в нем разобрались и вывели самостоятельно, даже в случае потери части информации вам будет легко ее восстановить при помощи обычной логики и простых расчетов.

Необходимо тщательно продумывать навигационную модель, способ каталогизации и рубрикации, чтобы пользователи понимали, а не запоминали.

## 7.7. Предсказуемость

### 7.7.1. Законы жанра

Во всякой сфере деятельности человека существуют так называемые законы жанра. Если вы большой любитель детективов, то рано или поздно вы сможете предугадывать события. Когда речь идет о Web-сайтах, предсказуемость возможна при наличии некоторой технологии, по которой работает пользователь. Особенно легко предугадать необходимые действия, если они соответствуют аналогу, к которому пользователь привык в реальной жизни.

Конечно же, Web-сайты редко соответствуют примерам из жизни, однако стоит рассмотреть некоторые показательные примеры.

Представьте себе систему с интерфейсом, которая позволяет через компьютер управлять проявкой фотографий. Если с ней работает профессиональный фотограф, который знает весь технологический процесс изготовления снимков, то он безошибочно выберет последовательность действий, даже если вы расположите в произвольном порядке кнопки Закрепить, Проявить, Просушить, Погасить свет и т.д. Однако не стоит злоупотреблять этим. Если человек знает, что пельмени достают из холодильника и забрасывают в воду уже после того, как она закипела, не заставляйте его выкладывать их в пустую кастрюлю, а затем заливать кипятком из чайника.

Предсказуемость во многом обусловлена теми же факторами, что и другие свойства юзабилити. Например, скорость работы пользователя может быть повышена за счет использования стан-

дартных шаблонов, которые увеличивают предсказуемость системы и упрощают принятие решения на следующем шаге. То же самое в некоторой мере относится и к четкой и ясной («прозрачной») навигационной модели.

Для повышения предсказуемости желательно использовать общепринятые понятия и термины (либо каким-то образом их расшифровывать), однозначные, стандартные или хотя бы часто используемые другими сайтами смысловые идеи пиктограмм, гиперссылки делать подчеркнутыми и стандартным для них цветом (синим или его оттенками), нельзя допускать сокращений (если это не Web-сайт для специалистов, которые обязаны знать, что такое ВВП, МВФ, НДС), невнятных формулировок и т.п.

Предсказуемость системы также значит, что пользователь может в любой момент сказать, где он находится, с каким объектом работает, что в данный момент делает система, какие действия можно предпринять далее.

Для построения предсказуемой системы можно рекомендовать такие приемы:

- навигационная пошаговая подсказка. Она применяется для отображения текущего действия пользователя и указывает, на каком шаге находится пользователь, в каком разделе, что за действие сейчас должно быть совершено (Ввод адреса, Результаты поиска, Выберите файл);
- окно ожидания (waiting box). Может быть использовано для отображения того, что система занята обработкой данных. Это некоторое статическое или динамическое изображение (например, анимированный рисунок) с надписью Загрузка (Обработка, Ждите). Во-первых, такой прием покажет пользователю, что система не «зависла» и обрабатывает информацию, во-вторых, время идет «быстрее» и десять секунд задержки покажутся пятью;
- индикатор шагов. Его часто используют в инсталляциях программ, где перечислены все шаги, которые необходимо сделать для завершения действия (например, заказа в интернет-магазине), и выделен текущий шаг. При таком подходе пользователь уже знает, что ему предстоит сделать;
- указание местоположения пользователя. Используется в тех случаях, когда речь идет о некоторой информационной струк-

туре (например, каталог ссылок). Это хорошее подспорье в работе. Представьте хотя бы человека, заблудившегося в лесу, который вдруг набрел на табличку с надписью «Вы — здесь!» и указателем на карте леса;


- всегда выдавать информацию об используемом (редактируемом) объекте системы (заказанной книге или сотруднике, телефон которого вы ищете).

### 7.7.2. Цена ошибки

Начиная работать над данной главой в Microsoft Word, я случайно нажал на пиктограмму отправки документа по e-mail вместо Сохранить. Внутри текстового редактора появилась часть интерфейса MS Outlook.

Как вы думаете, сколько времени у меня заняло размышление над тем, как вернуть все в прежнее положение? Секунд десять на то, чтобы подумать, к чему приведет нажатие на крестик в правом верхнем углу окна. Однако при нажатии на крестик мне предложили закрыть прежний документ. Я хотел еще поработать, честно говоря. Следующие десять секунд я пытался затащить кусок Outlook-а в середину окна, чтобы он превратился в отдельную панель и я смог ее закрыть. Это тоже мне не удалось. Еще десять секунд мне понадобилось для того, чтобы догадаться снова нажать на пиктограмму отправки документа по почте.


В итоге полминуты я потерял из-за того, что промахнулся по нужной пиктограмме и здорово отвлекся на решение появившейся проблемы. Можно ли такой интерфейс назвать предсказуемым? Повысилась ли скорость работы? С одной стороны, система выполнила все правильно и не могла знать, что мне действительно сейчас нужно. С другой — ошибка не должна приводить к столь мучительным последствиям. В таких случаях интерфейс должен обеспечивать быстрый и легкий возврат к предыдущему состоянию.

-  Для тестирования предсказуемости можно применить достаточно простой, но эффективный прием: во время процесса тестирования надо спрашивать пользователя о том, что он собирается сейчас сделать, куда, по его мнению, он попадет и почему система должна поступить именно так.

### 7.7.3. Соответствие бизнес-процессу

Под **бизнес-процессом** понимается некоторая постоянная либо изменяющаяся в незначительной степени совокупность действий, представляющая собой завершенную технологическую цепочку.

Бизнес-процесс — это основа для проектирования программных систем, на него необходимо опираться на протяжении всего цикла разработки. Рассмотрим бизнес-процесс работы с электронным телефонным справочником. Мы не будем описывать всех возможных пользователей (как известно, есть разделение как минимум на рядовых пользователей и администраторов), а представим себе некоторого среднестатистического посетителя, который имеет право делать в справочнике все что угодно (удалять, добавлять и редактировать объекты, получать отчеты, системную информацию и т.п.). Во всех этих действиях должен отражаться исходный бизнес-процесс поиска справочной информации.

 Необходимо понимать, что бизнес-процесс — это не результат. Это то, что существует и без всякого рода автоматизации и программирования. Это действия людей. Другое дело, чем они пользуются для осуществления своих действий. Если вы предлагаете некоторый инструмент в виде программного продукта, то вы не создаете новый бизнес-процесс, а автоматизируете существующий. Земля вертелась независимо от того, знал об этом Галилей или нет.

Вот так выглядит исходный бизнес-процесс, осуществляемый электронным телефонным справочником:

- пользователь обладает некоторыми знаниями об объекте, точные данные о котором он хочет получить (фамилия, номер телефона, адрес, наименование организации и т.д.);
- на основании этих знаний он получает один или несколько результатов запроса;
- полученные результаты необходимо каким-либо образом зафиксировать (запомнить или записать);
- при повторной необходимости пользователь должен снова получить данные, теперь уже из своих записей.

Как видим, в данном случае бизнес-процесс весьма прост. Это и есть самая суть, которая интуитивно понятна практически любому



пользователю. Добавьте в нее функции «Поиск в один клик», «Заказ пиццы на дом» и «Проверить состояние счета в швейцарском банке», и пользователь будет мгновенно сбит с толку, потому что предложенный набор уже не соответствует бизнес-процессу поиска телефона в справочнике. Думаю, вывод вам понятен. Насколько бы сложной ни была ваша система, какой бы большой набор инструментов она ни предлагала пользователю, первое, о чем следует позаботиться — это о соответствии бизнес-процессу.

## 7.8. Тестирование

### 7.8.1. Общий подход

Как проверить юзабилити, если вы сами создавали сайт и знаете все его тонкости и значение каждой ссылки и пиктограммы? Ответ прост — привлечь других. Звоните приятелям, усадите за компьютер свою маму, сестру. Главное — в это время не отходите ни на секунду. Каждое лишнее движение, каждый ошибочный шаг, каждое «фыркание» и нахмуренная бровь — это очень важно. Достаточно мнения трех-четырех пользователей, чтобы с ходу в несколько раз повысить уровень юзабилити. То, что пользователи говорят вам, далеко не столь важно, как то, что они делают. Чтобы начать тестирование юзабилити, не дожидайтесь завершения своего проекта. Чем позже вы поймете, что сделали ошибку, тем тяжелее исправить ее. Сначала покажите просто макет, потом более-менее рабочую систему, а уж затем окончательный вариант. И помните, вы во столько же раз увеличите количество своих клиентов и сторонников, во сколько раз повысите юзабилити.

На уровень юзабилити влияют практически все элементы интерфейса пользователя и их характеристики. Ранее я постарался рассмотреть наиболее важные из них, но мои советы не могут быть панацеей, потому что сколько Web-сайтов, сколько программных продуктов, столько и тонкостей в их работе.

Тестирование удобства использования, как и другие виды тестирования, приносит много пользы, поэтому не стоит им пренебрегать. Тестирование должно начинаться одновременно с вашим проектом. Чем раньше вы начнете проверять свои идеи построения

интерфейса, тем выше шансы избежать фатальных ошибок, исправление которых будет стоить очень дорого. Конечно, многое зависит от того, как вы строите свой Web-сайт. Если вы используете шаблоны проектирования, модель построения приложения MVC, если ваш сайт построен из набора визуальных компонентов, если вы применяете «гибкие» конструкции, то изменение интерфейса, который неожиданно окажется совершенно неудобным для пользователей, обойдется вам не очень дорого. Однако даже в подобном случае вы потеряете гораздо больше времени, чем если бы начали тестирование на ранних стадиях.

Представьте себе, что вы организовали способ заказа вашего товара через интернет по принципу «Мастера» (wizard), т.е. набора последовательных действий, которые должен выполнить пользователь. Вам казалось, что все четко и логично: вот он товар, вот они шаги. Вы выполнили реализацию, приготовились открыть свой виртуальный магазин и решили, что пора бы и протестировать ваш Web-сайт. И вот все, кому вы показываете свое детище, делают (говорят) то, что повергает вас в шок. Они ищут дополнительную информацию в середине процесса заказа, и если прервать его, то все необходимо начинать с первого шага. Они хотят заказать сразу несколько товаров, а не делать почти одинаковые действия три раза. Они хотят получать помощь и подсказки от системы в процессе заказа, а вы не предусмотрели эту возможность. Ваши действия, если вы, конечно, не считаете, что пользователи ничего не понимают и с ними надо бороться, — срочно переделать все, чтобы исправить хотя бы самые откровенные «ляпы» в работе сайта. В итоге вы имеете сотни файлов, которые надо переписать на 60–70 %, картинки, нарисованные под конкретную последовательность действий (очень красиво сделано, столько часов труда убито — жалко будет их выбросить), испорченное настроение и предстоящие бессонные ночи, потому что вы договорились запустить проект уже через две недели.

Каков же может быть иной сценарий развития событий?

Вы собираетесь поместить свой интернет-магазин в сети только через три-четыре месяца. Вы только начали работу. Что у вас имеется на данный момент? Ничего или почти ничего. У вас есть идеи, которые еще не затвердели в скриптах, HTML-коде и картинках. У вас есть несколько листов бумаги, на которых вы на-

бросали основные принципы и схемы. Вы показываете их тем же «пользователям», и они говорят то же, что и в первом случае. Но, думаю, вы и сами хорошо представляете последующие шаги.

### 7.8.2. Кто тестирует

Прежде чем приступить к тестированию юзабилити, следует выделить базовые категории пользователей и определить их цели. Эти категории различаются опытом, целями и особенностями психологии. Например, нет смысла привлекать домохозяйку для тестирования юзабилити сайта с финансовой аналитикой, точно так же как бесполезно просить взрослого человека протестировать сайт для детей. Выделив подобные категории пользователей, поставьте наиболее типичную задачу, например, найти на сайте по продаже бытовой техники сведения об особенностях конкретной модели утюга или микроволновой печи. Если вы создаете интернет-магазин, попросите оформить заказ на какую-нибудь книгу.

Для качественного тестирования юзабилити необходимо привлечь от трех до пяти человек. Такие цифры обусловлены повторяемостью результатов. Как правило, с некоторыми различиями пользователи будут сообщать вам одно и то же (либо вы сами будете это видеть) плюс несколько уникальных замечаний. При количестве пользователей больше пяти вы уже не узнаете почти ничего нового. Например, если вы поместили на сайте большое количество пиктограмм и ни одна из них не понятна, а всплывающие подсказки вы по каким-то причинам упустили, на 99 % могу гарантировать, что каждый из пользователей спросит нечто подобное: «А что означают эти картинки? Тут нигде нет... Обычно такая желтая штука выезжает и на ней написано — это сохранить или удалить или еще что-нибудь...»

Итак, привлечение одного-двух человек создает опасность упустить что-нибудь важное, а больше пяти — очень мало улучшает качество очередного теста.

### 7.8.3. Что тестируется

Помимо общего тестирования юзабилити существуют некоторые более узкие его задачи. Вас может интересовать, например,

только сам процесс заказа товара в интернет-магазине и его понятность. Может появиться необходимость проверить, как трактуют пользователи созданные дизайнером пиктограммы, которые они ранее не видели. Может понадобится проверка «гуманности» подсказок и помощи по работе с Web-сайтом, для чего можно симитировать проблему и попросить пользователя решить ее с помощью указаний в справке. Одним словом, не всегда целесообразно делать глобальное тестирование удобства использования. Особенно важно протестировать таким образом нестандартные решения.

При тестировании важно понять задачу и четко сформулировать ее. Вы не потеряете свое драгоценное время, гоняясь за подставными целями. Четко определите то, что вы хотите выяснить, сделайте соответствующие только этой цели тесты и проанализируйте их результаты. Если все сделано правильно, то вы получите необходимые знания за минимальное время. Четкая постановка задачи — гарантия четкого результата.

## 7.9. Разбор уровней юзабилити

Теперь самое время привести конкретные примеры того, как выглядят «добро и зло» в юзабилити и как они влияют на работу пользователя. Не все, но очень многие критерии удобства использования становятся очевидными даже при поверхностном анализе.

Итак, у меня есть два электронных почтовых ящика. Первый я зарегистрировал достаточно давно на Yahoo!. Этот сервер всегда сильно загружен, размер ящика долгое время был скромненький, всего 4 мегабайта, да и далеко это от наших широт. Словом, позже я решил завести еще один почтовый ящик поближе, на белорусском портале TUT.BY. Скорость хорошая, патриотизм, места сколько угодно... Вот только работать с почтой мне гораздо приятнее на Yahoo!. Я объясню, почему.

При входе в электронный почтовый ящик на TUT.BY я должен вводить одни и те же данные всякий раз, даже если работал с почтой пять минут назад и закрыл браузер (рис. 7.24).

Страница TUT.BY загрузилась в течение одной секунды. После этого я быстро скопировал изображение в буфер обмена, запустил

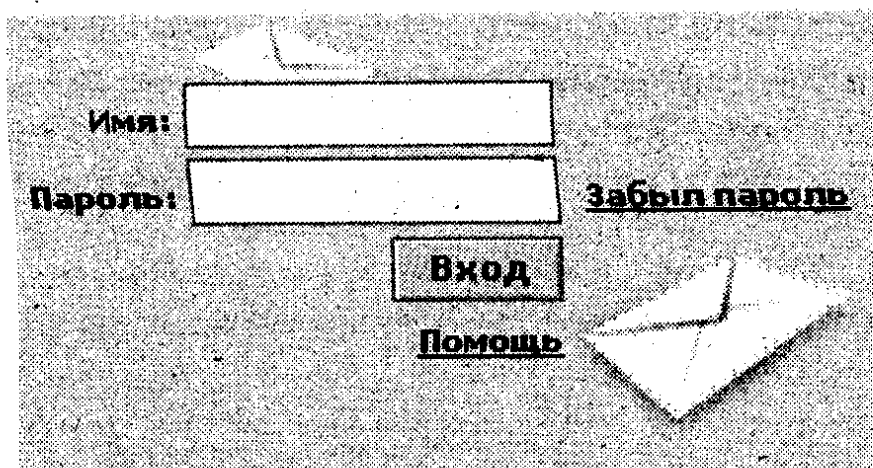


Рис. 7.24. Фрагмент страницы входа в почтовый ящик на TUT.BY

графический редактор, слегка обработал картинку. За это время Yahoo! еще не появился. По прошествии целой минуты страница Yahoo! загрузилась и я попал в свой почтовый ящик, введя только пароль (рис. 7.25). Изредка, наверное, один раз в два-три месяца, меня просят подтвердить свой логин и пароль (рис. 7.26).

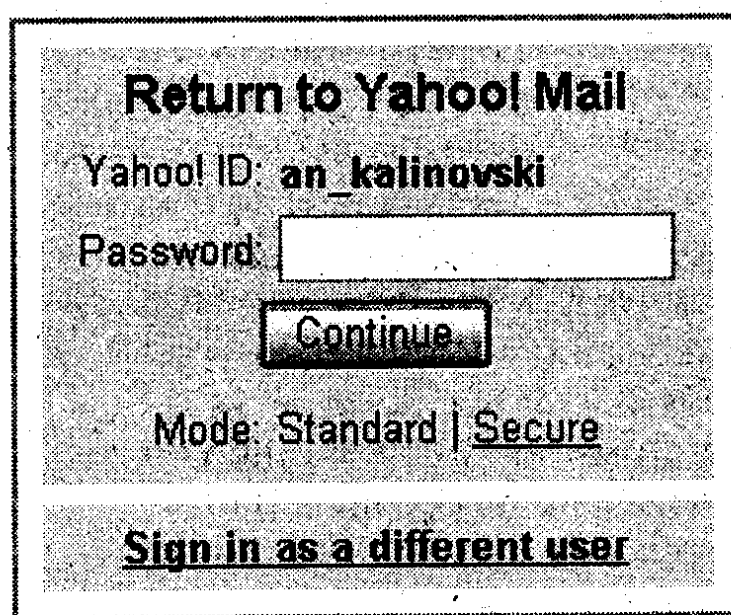


Рис. 7.25. Фрагмент страницы входа в почтовый ящик на yahoo.com.  
Я должен ввести только пароль

При установке опции запоминания данных пользователя я могу входить в свой почтовый ящик, не вводя длинное имя каждый раз (я всегда забываю, оно написано с подчеркиванием или через минус). Лично для меня это очень удобно.

**Existing Yahoo! users**

Enter your ID and password to sign in

Yahoo! ID:

Password:

☒ Remember my ID on this computer

**Sign In**

Mode: Standard | Secure

[Sign-in help](#) [Forgot your password?](#)

Рис. 7.26. Фрагмент страницы входа в почтовый ящик на yahoo.com. Выделена опция Запомнить мои данные при входе с этого компьютера

На главных страницах почтовых ящиков нет различий, критических для юзабилити. Единственное отличие, достойное внимания, — визуализация использованного места в почтовом ящике (рис. 7.27, 7.28).

Размер	Писем	Новых	Непро-
4781Kb	01	1	

Рис. 7.27. Занятое письмами место на TUT.BY отображается с помощью цифр. Я никогда их не вижу

You are using 2% of your 4.0MB limit.

Need more space? [Get more storage.](#)

Рис. 7.28. Занятое письмами место на yahoo.com отображается с помощью визуального индикатора. Для оценки достаточно беглого взгляда

Посмотрев на шкалу, я моментально представляю себе положение вещей (занято «ерунда» мегабайт почтового ящика — можно спокойно работать дальше), в то время как на TUT.BY я никогда не смотрю на цифры. Их труднее воспринимать, они плохо выделяются на серо-голубом фоне. Кроме того, надо сделать хотя бы грубый подсчет, а вести в уме расчеты при работе с почтой я не хочу.

На рисунках 7.29 и 7.30 показано, как отображаются письма на TUT.BY и на Yahoo!.

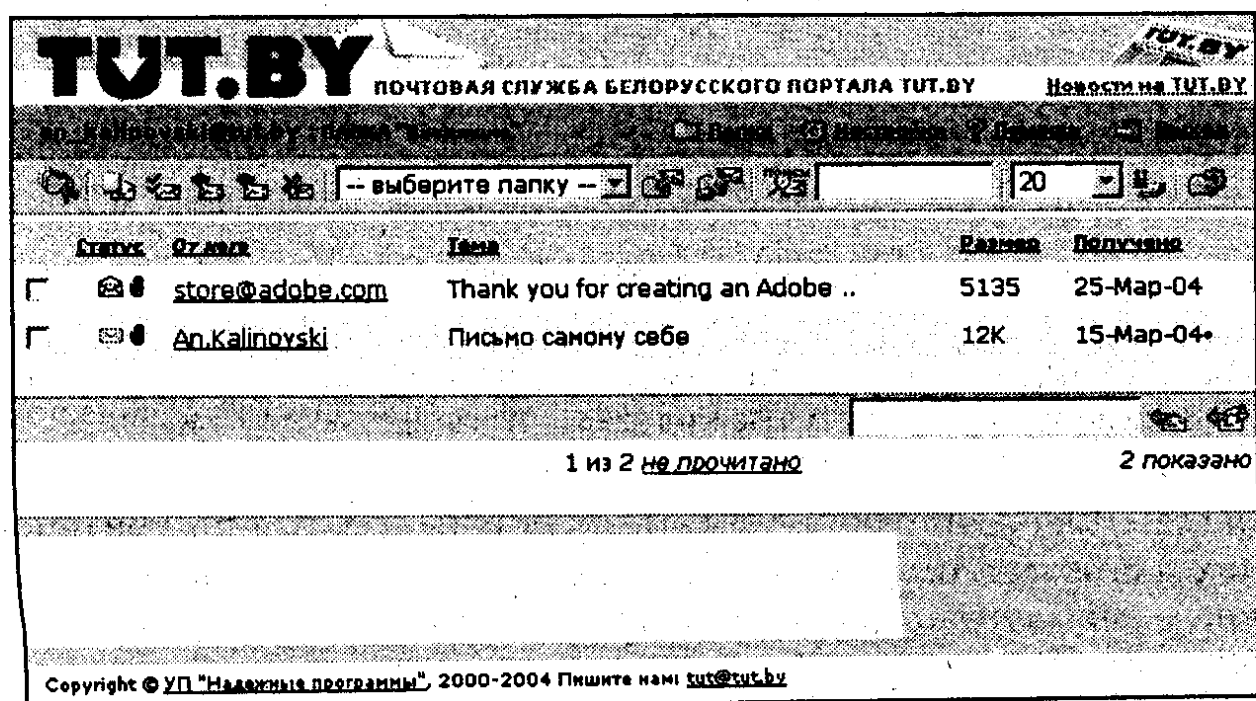


Рис. 7.29. Список писем на TUT.BY

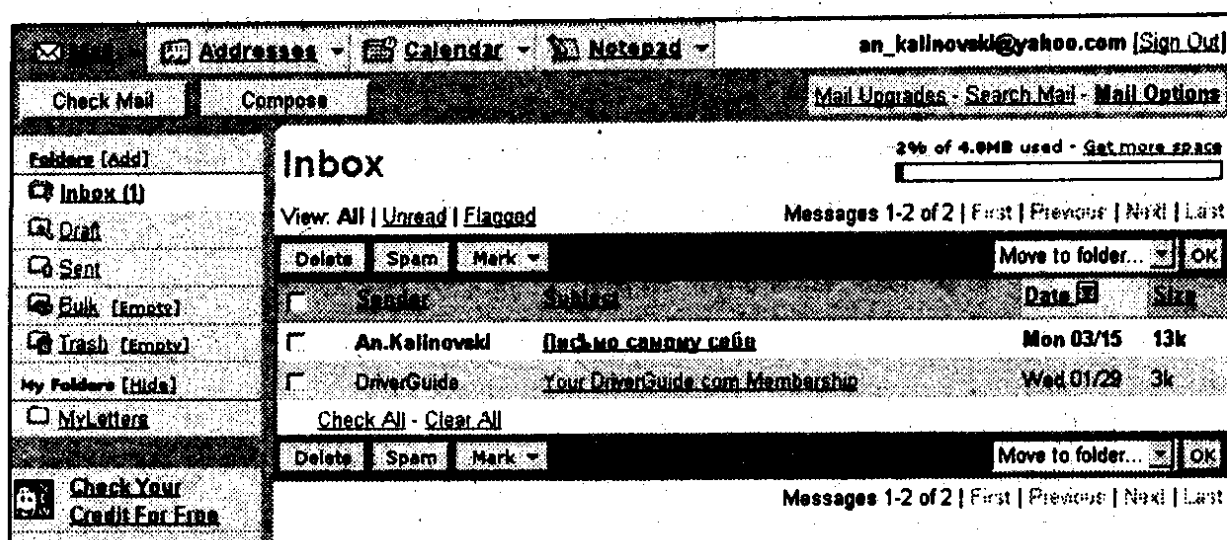


Рис. 7.30. Список писем на yahoo.com

Внимательно рассмотрите эти рисунки. Различия интерфейсов почтовых служб таковы:

1. Элементы навигации на TUT.BY — это пиктограммы, а на Yahoo! — текст (рис. 7.31, 7.32).

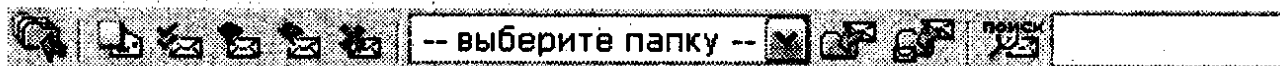


Рис. 7.31. Элементы навигации на TUT.BY



Рис. 7.32. Элементы навигации на yahoo.com

В спину пиктограмм вонзили уже немало ножей, и я, наверное, не буду продолжать эту традицию. Просто скажу, что неудобно:

- форма перегружена графикой;
- я не могу запомнить значения всех пиктограмм;
- пока картинки не загрузятся, я не могу работать дальше;
- в мелкие картинки сложно попасть мышью.

2. На TUT.BY отсутствуют элементы навигации в нижней части списка писем. Конечно, при малом количестве «входящих» дубли кнопок на Yahoo! смотрятся, мягко говоря, странно. Однако это не покажется таковым, если в списке будут сотни старых и десятки новых посланий. Вы выделяете ненужные письма и собираетесь их удалить. Поскольку вы уже внизу списка, вам придется долго листать страницу вверх, чтобы нажать на пиктограмму Удалить (рис. 7.33, 7.34).

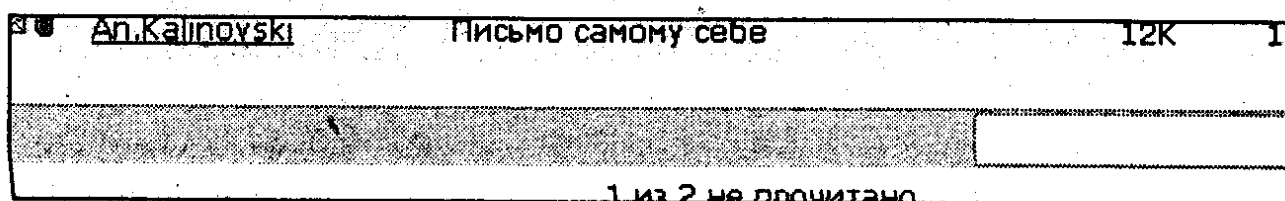


Рис. 7.33. Дубли элементов навигации на TUT.BY отсутствуют



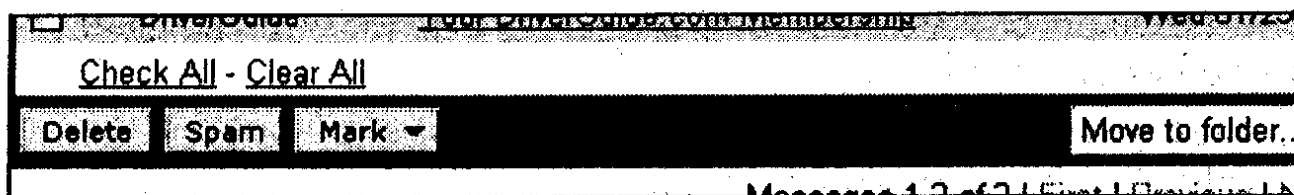


Рис. 7.34. Элементы навигации на yahoo.com продублированы внизу страницы

3. При переходе к списку папок на Yahoo! вы сделаете всего одно движение и щелчок мыши. Кроме того, вы сразу можете видеть состояние других папок и наличие в них новых писем (рис. 7.35).

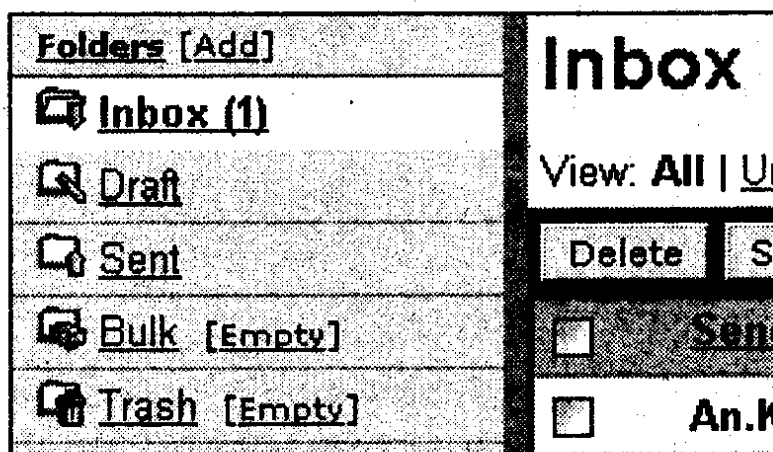


Рис. 7.35. Папки и управление ими на yahoo.com всегда под рукой

Чтобы перейти к папкам на TUT.BY, вам придется сначала нажать на соответствующую гиперссылку (рис. 7.36).

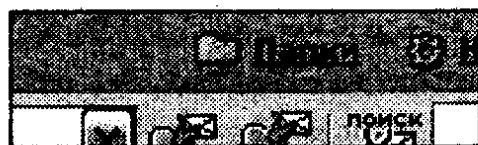


Рис. 7.36. Для перехода к папкам на TUT.BY необходим дополнительный шаг и время на загрузку страницы

Вспомните привычные для вас почтовые программы (Microsoft Outlook, The Bat, Becky) — везде поддержана абстракция папок. На Yahoo! разработчики и дизайнеры не отступали от этого, а на TUT.BY проявили оригинальность. В результате

юзабилити снизилось из-за лишних шагов и отступления от общепринятого шаблона.

4. Следующее отличие, однако, уже не в пользу Yahoo!. Если вы захотите перенаправить кому-либо свою почту, вам придется посылать каждое письмо по отдельности (рис. 7.37). Это вам быстро надоест, и все достоинства Yahoo! уже не будут казаться столь значительными.



Рис. 7.37. На TUT.BY поддержана функция перенаправления порции писем на другой адрес, хотя назначение этого поля и пиктограмм неочевидно. На Yahoo! подобная функция вовсе отсутствует

5. С почтой я работаю таким образом: я читаю письмо, удаляю его и перехожу к следующему. Мне неудобно, что на TUT.BY я не могу так работать: я должен выбирать либо переход к следующему письму при помощи щелчка мыши на пиктограмме, либо удаление, но тогда система автоматически «выбрасывает» меня к списку писем. Опять лишний шаг. Плюс все та же история с отсутствием дублей кнопок, что и в списке писем. На Yahoo! этих проблем нет. Удаляя письмо, я автоматически попадаю в следующее.

Есть и другие особенности, например, способ установки и снятия выделения всех писем в папке.

А теперь несколько мелких «бляк».

Если работать на TUT.BY около часа, то происходит следующее (рис. 7.38).

**Вы отсоединены от сервера! Войдите, пожалуйста, снова.**

Рис. 7.38. TUT.BY заботится о безопасности.  
Сессии быстро «погибают»

Небольшая загадка (рис. 7.39). Попробуйте угадать, что скрывается за крестиками.

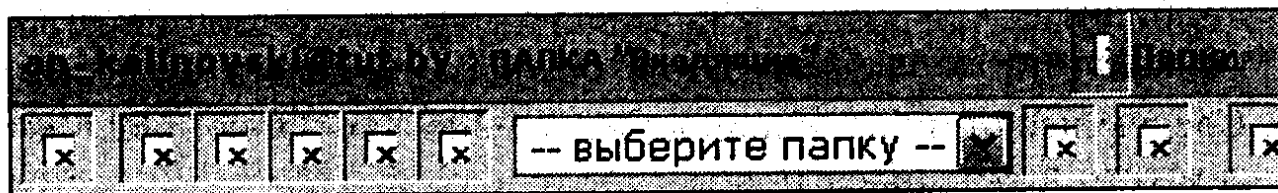


Рис. 7.39. Из-за определенных проблем пиктограммы не загрузились

Иногда пиктограммы не появляются, иногда вместо них крестики. Я приспособился работать быстрее, чем кажется: я просто помню, где находится наиболее нужная мне пиктограмма Удалить (кстати, на разных страницах она в разных местах). Я ставлю курсор в соответствующую область экрана, появляется всплывающая подсказка (благо здесь разработчики не поленились) и курсор в виде руки, и тогда я нажимаю на это «нечто». Но разве это выход?

В этом рЙУШЕНПе (в смысле, в письме) проблемы с кодировкой (рис. 7.40). На Yahoo иногда случается, а вот на TUT.BY я ни разу не видел странных рЙУШЕНПов.



Рис. 7.40. Порой кодировка на Yahoo! «хромает»

- ✍ Описывая одну из форм TUT.BY, я закрыл браузер. Я смог вспомнить назначения только нескольких пиктограмм. Лишенный возможности подвести курсор к пиктограмме и прочитать всплывающую подсказку, я оказался «безоружным» перед фантазией дизайнера, который их придумал.

## Резюме

1. Юзабилити — степень удобства и простоты использования некоего продукта для его потребителя, а также способы обеспечения этого удобства. Начав знакомство с вашим программным продуктом с нуля, пользователь должен быстро обучиться работе с ним.
2. Неудачные интерфейсные решения создают негативное впечатление о вашем ресурсе и надолго запоминаются.


3. Для ускорения работы и повышения предсказуемости интерфейса важную информацию необходимо выделять. Существуют способы выделения информации на визуальном и на композиционном уровнях.
4. Надписи на элементах интерфейса должны быть понятными.
5. Отвлекающие элементы («прыгающий» по странице текст или бегущая строка) уводят пользователя от информации. Пользуйтесь этими приемами только в том случае, если вы намеренно желаете отвлечь пользователя.
6. Нельзя делать элементы интерфейса пользователя слишком мелкими и близкими друг к другу. Наиболее важные элементы интерфейса не следует располагать в отдаленных уголках экрана.
7. Нежелательно использовать горизонтальную полосу прокрутки, а также две и более вертикальных.
8. Предоставление списка посещенных разделов, результатов поиска, хранение некоторой предварительной информации о пользователе ускоряют работу.
9. Сокращенная форма отображения данных экономит место на экране, ускоряет восприятие информации, упрощает работу, однако требует знания предметной области системы.
10. Удобнее всего читать текст, который контрастирует с фоном. Допустимо использовать не более трех шрифтов (желательно стандартных). Необходимо поддерживать правильное распознавание браузером кодировки текста.
11. Важны четкие и ясные заголовки.
12. Грамматические ошибки в тексте недопустимы. Обработка системой ошибок пользователя обеспечивает комфортность в работе.
13. Не стоит предлагать несколько решений одной проблемы.
14. Тестирование юзабилити необходимо начинать практически одновременно с началом работы над проектом. До начала тестирования необходимо выделить базовые категории пользователей вашего сайта. Для результативного тестирования достаточно привлечь три — пять человек.

# Нестандартизованные стандарты

## 8.1. Как появляется стандарт

Если бы во Всемирной паутине не было никаких стандартов, я бы не смог написать эту книгу, ведь пришлось бы описывать тысячи вариантов построения интерфейса.

Стандартное решение в Web носит характер скорее рекомендательный, чем обязывающий. Никто не оштрафует вас, если вы вдруг решите расположить ссылку на начальную страницу сайта в середине экрана. Постоянные клиенты не откажутся покупать ваши товары потому, что вы используете в своем интернет-магазине технологию флэш или java-апплеты.

 Под словом «стандартный» в данном случае понимается не какой-то строгий стандарт, например, рекомендованный консорциумом W3C, а часто используемое решение. Поэтому в данной главе я буду употреблять слово «стандарт» условно.

Как же появляются стандарты в Web, если никакая организация их не разрабатывает, а каждый разработчик Web-сайтов хочет выделяться и не быть похожим на остальных?

Дело в том, что пользователи очень быстро схватывают какое-нибудь более-менее понятное решение и принимают его за стандарт. Причем чем менее опытен пользователь, тем быстрее он формирует для себя понятие стандартного решения. Если ему кто-то показал, как пользоваться поисковым сервером, то он будет уверен, что если в мире и существуют другие поисковые серверы, то все они должны работать именно так. Подтверждение тому я наблюдал не так давно. Один мой знакомый, начинающий пользователь, на старом компьютере у себя на работе любил поиграть в игру «Шарики», которая запускалась из меню Norton Commander клавишей F2. Когда мы оказались вместе за другим компьютером, он спросил, есть ли там «Шарики». Я ответил, что есть, и уточнил, знает ли он, каким образом запускать эту игру. Он без тени сомнения ответил, что нажмет F2, выберет из меню строчку с названием игры и нажмет Ввод.

Все когда-то бывает в первый раз. Кто-то первый придумал функцию поиска на страницах Web-сайта. Кто-то придумал выпадающее меню, кто-то предложил сделать гиперссылки синими... Когда появляется это первое решение, возникает «второй» кто-то. Этот второй смотрит на удачную или просто интересную идею и говорит: «Отлично, здорово придумано, надо это использовать». Хорошая идея становится всеобщим достоянием, а «первопроходец» зачастую забывается. Когда появляется некто «третий», он действует следующим образом: он знает, что ему нужно примерно «что-то такое, чтобы...», изучает примеры на других сайтах и, если видит похожее решение не раз и не два, а в 60–70 % случаев, решает, что так делают все. «Четвертый» анализирует все, что происходило до него, и пишет статью о том, что есть такое вот решение, которое следует применять в определенных случаях. Таким образом, решение становится стандартом без подтверждения или документального оформления.

Стандартными решениями могут быть не только те, к которым привыкли пользователи интернета. Например, вы организуете заказ журналов и газет в библиотеке через некоторый on-line сервис. Для этого вам необходимо создать в форме компонент для ввода списка запрашиваемых номеров. В таком случае вы можете применить привычный для большинства пользователей способ на основании диалога, аналогичный выводу документа на печать в MS Word. При таком варианте вы можете быть уверены, что кто-то до вас объяснял, как следует вводить номера страниц. Это решение можно в какой-то мере называть стандартным, хотя оно пришло и не из Web.

Примерами «нестандартизованных» стандартов могут служить такие интерфейсные решения, как ссылка на главную страницу слева вверху каждой страницы, форма поиска в виде поля ввода запроса с кнопкой, навигационное меню с левой стороны страницы, пиктограмма конверта рядом с ссылкой для отправки электронной почты, гиперссылка Подробнее и т.д.

Я думаю, что найдутся люди, которые не согласятся со мной и приведут в качестве примера официальных стандартов в Web какое-нибудь руководство NASA или Apple. На это я могу ответить, что знать о таких стандартах могут только специалисты.

в области построения интерфейсов, но не многочисленные Web-мастеры и дизайнеры. Широко распространенных стандартов построения Web-приложений еще нет, и появятся они, к сожалению, не скоро.

## 8.2. Стандартизованный интерфейс

Использование большинства привычных для пользователей решений приводит к стандартизации пользовательского интерфейса. Как я уже сказал, это понятие в значительной степени условное. Стандартизованный интерфейс легок для изучения, он не вызывает дискомфорта и напряжения, не заставляет пользователей задумываться о том, как тот или иной элемент должен работать. Чем более привычное решение используется, тем меньше сил тратится на его объяснение, меньше места «съедается» поясняющими надписями и подсказками, возрастает удовлетворенность пользователя работой системы.

С точки зрения разработчика, стандартизация — весьма выгодная идея. Как пример можно привести приложения, используемые во внутренних интранет-сетях. Количество пользователей систем такого рода, во-первых, ограничено, во-вторых, часто одни и те же пользователи работают с различными приложениями одного разработчика (набор приложений Microsoft Office). Стандартизованные решения ускоряют создание новых программных продуктов и повышают возможности их интеграции, что важно для разработчика, и повышают обучаемость пользователей и продуктивность их работы, что важно уже для заказчика. Стандартизованное приложение, в котором легко работать, усиливает привыкание, следовательно, пользователи будут меньше «смотреть» в сторону конкурентов.

Во внешнем мире все эти выгоды блекнут, стоит только подумать о том, что корпоративный сайт знаменитой компании будут называть невыразительным и скучным, потому что он использует только привычные и едва ли не устаревшие идеи. Во Всемирной паутине все хотят выделиться — того требует рынок. «Если ваш сайт не выглядит особенно, то вы и не создаете ничего особенного», — это мнение многих людей. Как можно доверять

студии Web-дизайна создание лица компании в интернете, если та сама создала себе скучный унылый сайт как у всех? В интернете встречаются «по одежке», поэтому никакая уважающая себя Web-студия не станет делать пяти заказчикам одинаково выглядящие сайты, различающиеся только содержимым. Ее просто засмеют.

В Web никто не хочет полной стандартизации и унификации, потому что тогда исчезнут дизайнерские изыски, креатив и «классные фишки» — все станет одинаковым, а ведь фирменный стиль, престиж, отличительные особенности еще никто не отменял.

Использование стандартных решений не во внешнем оформлении сайта, а в его интерфейсе (надеюсь, что отличия этих понятий очевидны) уже не столь опасно, скорее наоборот. Они повышают юзабилити интерфейса, поддерживают предопределенные действия, проверены на большом количестве пользователей, и потому в них можно быть уверенным.

### 8.3. Стандарт: шаблон или копия

В любом стандарте важна не только идея, но и ее исполнение. Идея может быть гениальной, а исполнение ужасным. Приводимый ниже пример уже перестал быть актуальным в настоящий момент, однако я не буду убирать его из книги, ведь это своего рода страницы истории (рис. 8.1).

Так раньше выглядела начальная страница интернет-магазина [ozon.ru](http://ozon.ru). Основная навигация, которая повторяется на каждой странице, была построена из двухуровневого меню, причем без корневого раздела. Выходило так, что с начальным меню, которое я вижу на первой странице, я больше не смогу работать — оно исчезает. Единственный выход — перейти в начало по адресу [www.ozon.ru](http://www.ozon.ru).

Сайт копирует некоторые решения знаменитого торгового портала [amazon.com](http://amazon.com) (созвучность названий, то же самое двухуровневое меню, напоминающее вкладки), что само по себе уже не очень хорошо (рис. 8.2).



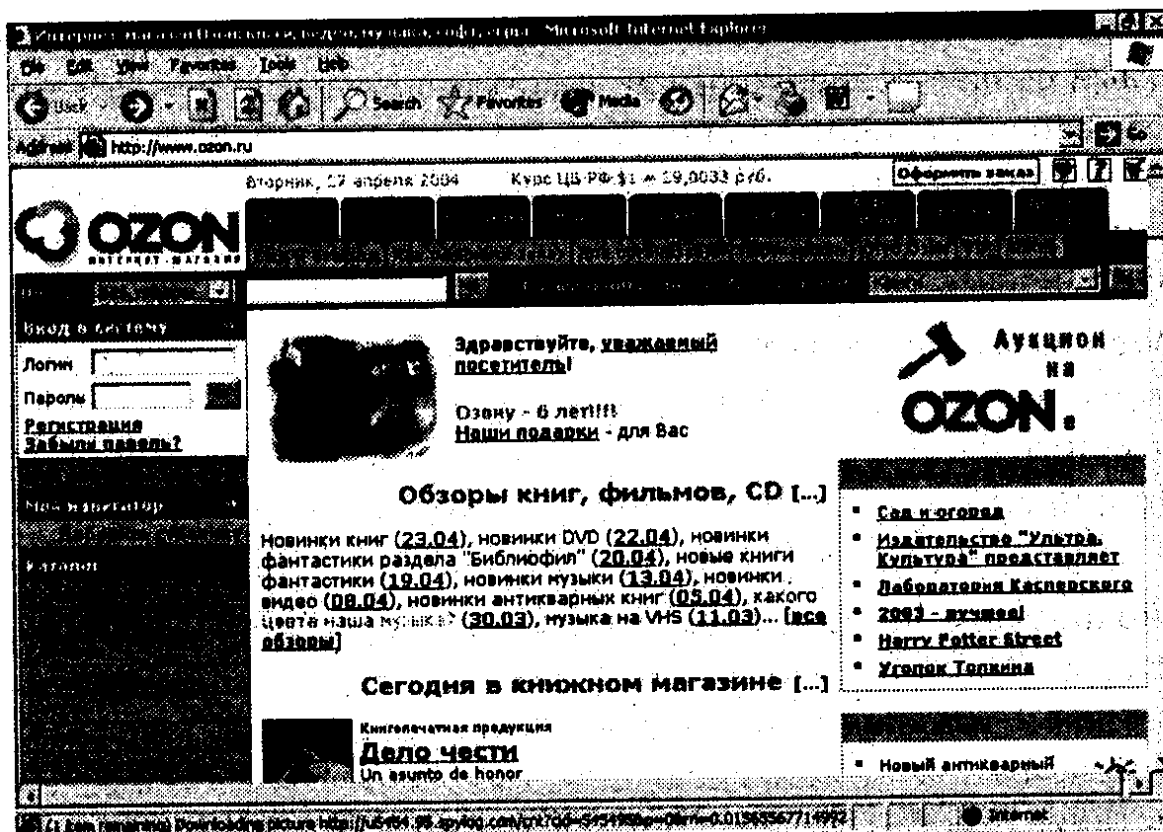


Рис. 8.1. Старый интерфейс ozon.ru

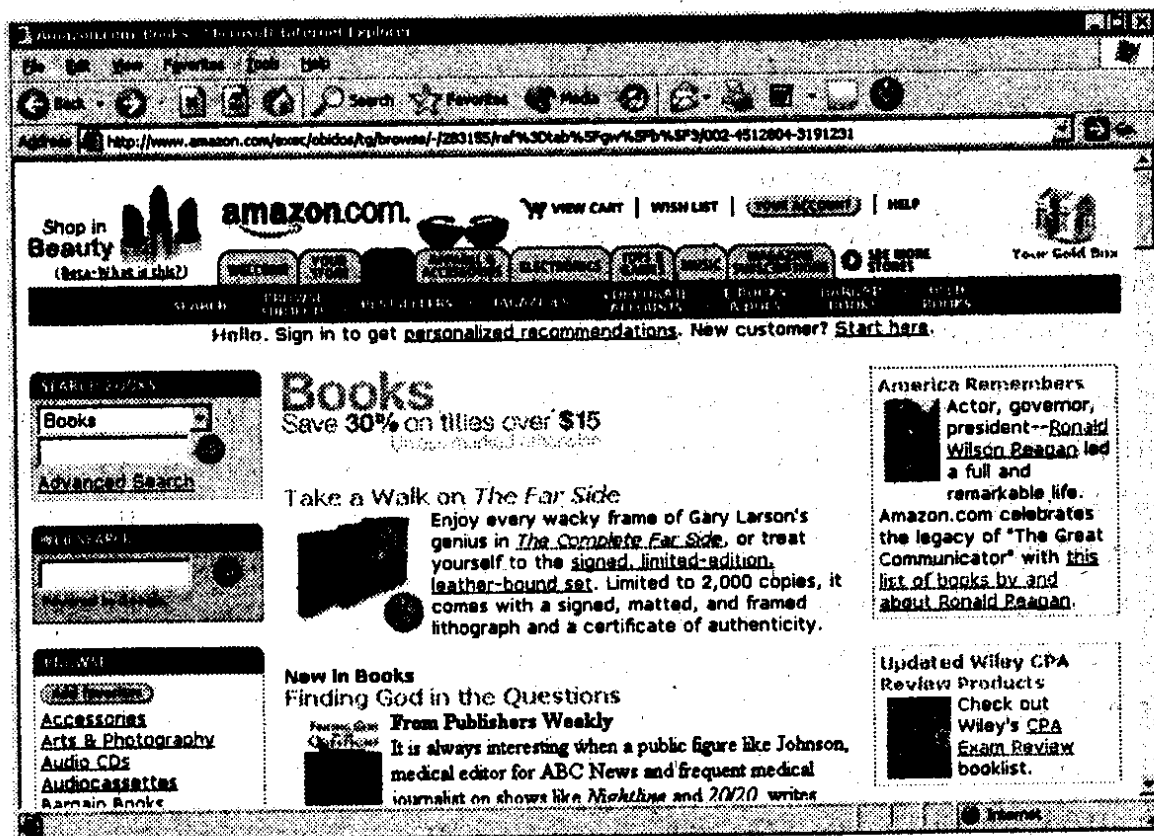


Рис. 8.2. Вот откуда, вероятно, происходило двухуровневое меню, ранее существовавшее на сайте ozon.ru

Однако принцип двухуровневого меню позаимствован не до конца. Я совершенно не понимаю, почему разработчики Озона решили разделить эти уровни цветом и границей. Такое решение заставляет пользователя задумываться, действительно ли это подчиненные части или произвольно меняющиеся уровни. Гораздо правильнее было бы слить эти уровни в единое целое.

Я предполагаю, разработчики посчитали, что пользователю покажется текущим тот пункт меню, который находится под наименованием раздела, например, Books. И решили сделать немного по-своему. Таким образом, была взята хорошая идея, но перенесена несколько небрежно.

- ❑ Разработчики Озона со временем исправили многие недостатки сайта, о которых мне было известно (рис. 8.3). Были устранены очевидные для меня проблемы интерфейса, что лишний раз доказало, что я все-таки имею право писать подобную книгу, и даже если я в чем-то ошибусь, то все-таки донесу до людей какую-то часть ценного знания.

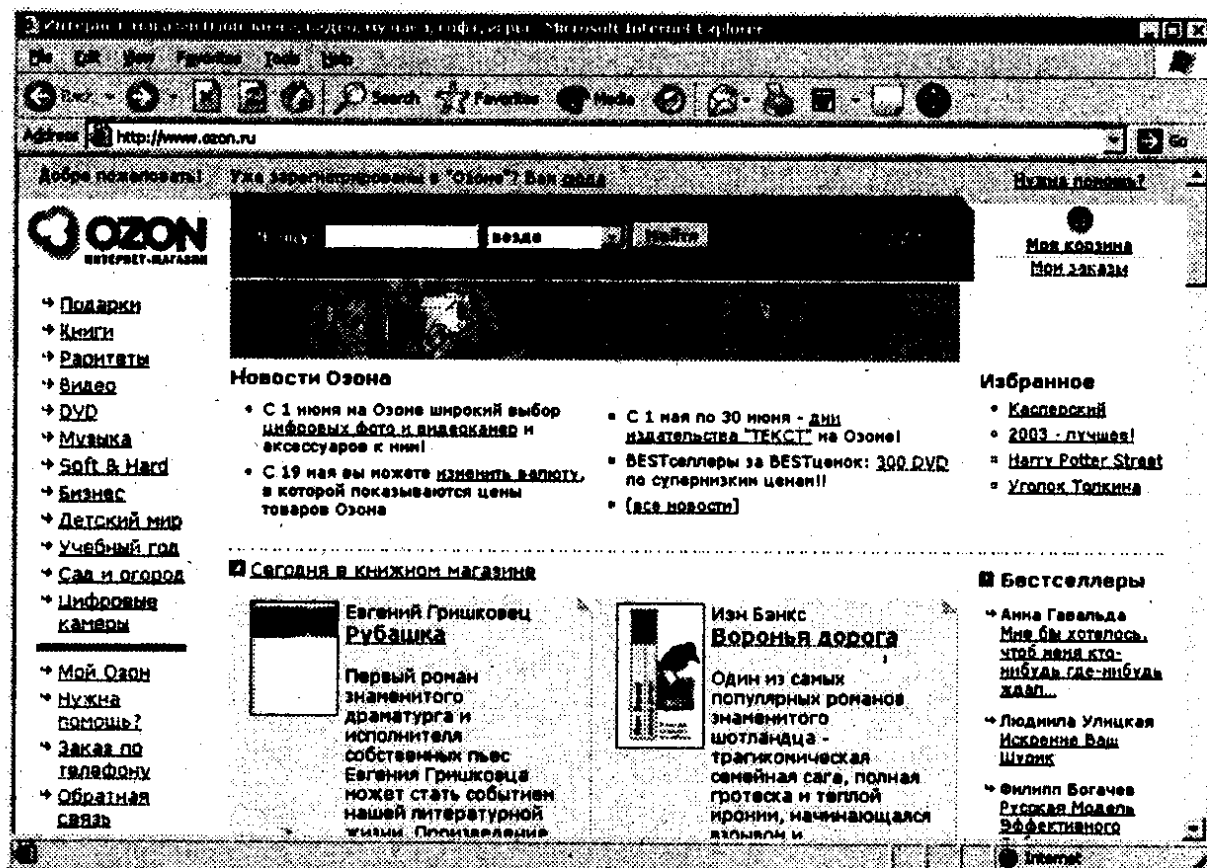


Рис. 8.3. Кто-то здорово постарался над юзабилити сайта ozon.ru.  
Честь ему и хвала

## Резюме

1. Стандартное решение в Web носит рекомендательный, а не обязывающий характер.
2. Часто решение становится стандартом без подтверждения или документального оформления.
3. В любом стандарте важна не только идея, но и ее исполнение.
4. В Web никто не хочет полной стандартизации и унификации, но стандартизованный интерфейс выгоден как для пользователей, так и для разработчиков.

# Справка и помощь на Web-сайте

## 9.1. Справочная система

Вы не скажете, во сколько отходит ближайший поезд на Москву? А где здесь киоск со свежими газетами? Как зовут этого актера? А чтобы включить сушилку для рук, нужно на эту кнопку нажать? Вам нужно написать большой курс обмена, чем там предлагают. Так вы говорите, это вкусная колбаса? Вот смотрите, сюда вставляется пластиковая карточка...

Пользователь — существо гордое. Пока в ловушку не попадет, помощи не попросит. Он не будет читать документацию перед началом работы с системой. Разнообразие порождает нетерпеливость и недоверие. Во Всемирной паутине на человека, который сначала читает, как работать с Web-сайтом, а потом проверяет, есть ли на нем нужная информация, опытные пользователи посмотрят с жалостью. Целесообразность использования справочной системы на сайте определяется его сложностью, объемом, количеством пользователей, никогда ранее не посещавших сайт.

Система справки и помощи в любом ее виде весьма сложна и многогранна. Это не только руководство по работе с системой, которое появляется, если нажать на знак вопроса или гиперссылку Помощь, — это всевозможные элементы «пассивной» помощи. К таким элементам я отношу всплывающие подсказки, сообщения в диалоговых окнах, названия форм, кнопок и пошаговые подсказки, списки посещенных разделов, поиск.

*Справочная система* представляет собой документацию и различные руководства по использованию и настройке программного продукта. В зависимости от способа использования (Web-сайт, интранет-приложение, Web-интерфейс к какой-либо программе или устройству) справочная система программного продукта может включать следующие виды документации:

- пресс-релиз системы;
- документация по установке;
- документация по управлению системой и настройке;


- краткое описание основных функций системы;
- подробное описание всех функций системы;
- примеры способов работы с системой;
- документация по развитию и дополнению системы и др.

Вся система помощи и справки разделена на три большие части: руководство пользователя, руководство администратора и руководство программиста. Все остальные ее элементы — подмножества этих трех частей. Не обязательно все эти части должны быть представлены в системе. Зачастую ограничиваются только руководством пользователя, тем более что в интернете во внешний мир «смотрит» только оно.

Смысл системы справки и помощи независимо от того, для кого она предназначена, всегда одинаков. Это иерархически организованные разделы с информацией, и доступ к ним может быть получен из разных точек. В о - п е р в ы х, пользователь может попасть в систему помощи и справки из сообщения об ошибке. Это наиболее вероятный путь, т.к. в большинстве случаев пользователи обращаются за помощью именно тогда, когда произошла ошибка. В о - в т о р ы х, он может перейти к справочной системе на каком-то конкретном шаге, если та информация, которая видна на странице, недостаточна. В - т р е т ь и х, на каждой странице должна находиться ссылка на систему помощи и справки, для того чтобы пользователь мог перейти на главную страницу этой системы и ознакомиться с ней.

Справочная система должна обязательно иметь три составляющие: собственно справка, поиск в справке и глоссарий (словарь терминов).

Справочная документация может быть представлена в бумажном и электронном виде. За исключением особенностей чтения с экрана и с бумаги, документация в электронном виде имеет существенное преимущество: она может быть сколь угодно велика и без усилий размножена сколько угодно раз. Лучшим решением было бы предоставлять пользователям и бумажную, и электронную документацию. Понятно, что в случае с интернет-магазином это нонсенс, зато если вы пишете Web-сайт для использования *его как некоторого корпоративного портала в своей организации*, то целесообразно сделать доступными обе версии документации.

 Необходимо также предусмотреть трансформацию документов из одного вида в другой. Преобразование электронной документации в бумажную, в особенности on-line документации, происходит путем отправки страницы, отображаемой в браузере, на принтер. Именно для этого и предназначена Версия для печати. С обратным преобразованием сложнее. Для трансформации бумажной документации в электронный вид придется использовать сканер и систему распознавания текста.

Различается также восприятие одной и той же системы справки в on-line и off-line режимах. Когда вы читаете какое-нибудь руководство в интернете, вам приходится ждать загрузки страниц, вы отвлекаетесь и постоянно теряете нить. Читая то же самое справочное руководство с жесткого диска своего компьютера, вы куда лучше усваиваете материал и меньше напрягаетесь. Учитывая это, было бы неплохо позволить пользователю получить документацию именно в виде архива с информацией по системе или справочника в формате pdf, chm или doc.

## 9.2. Сообщения об ошибках

Сообщение об ошибке — первый способ ознакомления пользователя со справочной системой. К сожалению, программы и Web-сайты несовершенны, и часто знакомство с программным продуктом начинается именно с сообщения об ошибке. В таком сообщении может быть сказано о том, что произошло, почему произошло и что теперь делать. Если предположить, что пользователи интернета не читают документацию, то это едва ли не единственный способ рассказать им о работе системы, хотя появление сообщения об ошибке крайне нежелательно. Если вы сможете добиться того, что пользователи вашего сайта не будут допускать ошибок, то им не понадобится справка и помощь. Однако ошибаются даже опытные пользователи, поэтому вам придется продумать варианты сообщений об ошибках.

Одним из таких вариантов является предупреждающее диалоговое окно (рис. 9.1). Оно является как бы сообщением об ошибке еще до ее совершения.

Диалог с пользователем в Web-приложениях может быть организован в новом окне (рис. 9.2) и в основном окне браузера, т.е. основная информация замещается на форму диалога.

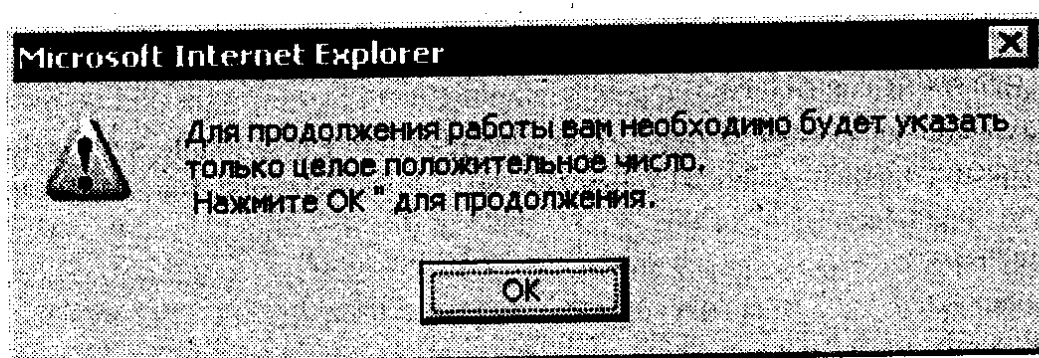


Рис. 9.1. Предупреждающее диалоговое окно

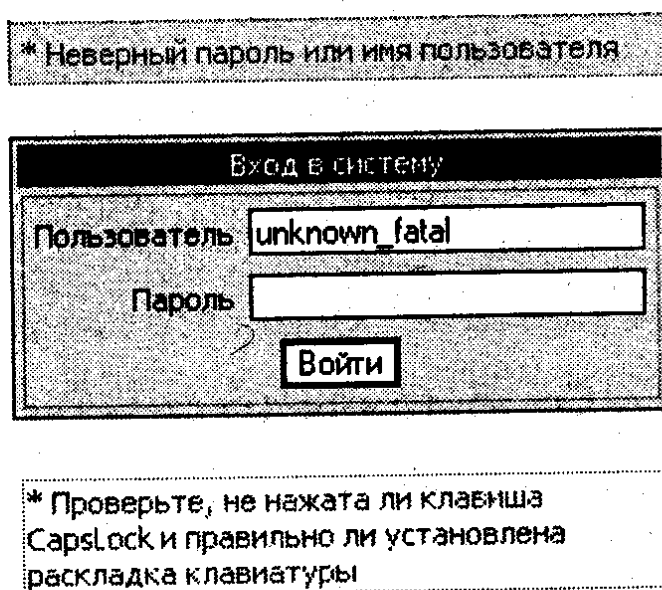


Рис. 9.2. Диалог с пользователем внутри Web-формы

Последний способ может быть реализован как стандартное диалоговое окно, принадлежащее браузеру и вызываемое при помощи JavaScript, и как новое окно приложения с некоторой формой (кнопки, сообщение, вопрос).

### 9.3. Контекстная помощь

**Контекст** — это некоторое окружение, актуальное в данный момент и обладающее законченным смыслом.

**Контекстная помощь** — помощь, оказываемая системой по ходу действия. Если посетитель оформляет заказ, то при помощи контекстной помощи система сообщает о том, как это сделать удобно и правильно, либо о ближайшем действии (рис. 9.3).

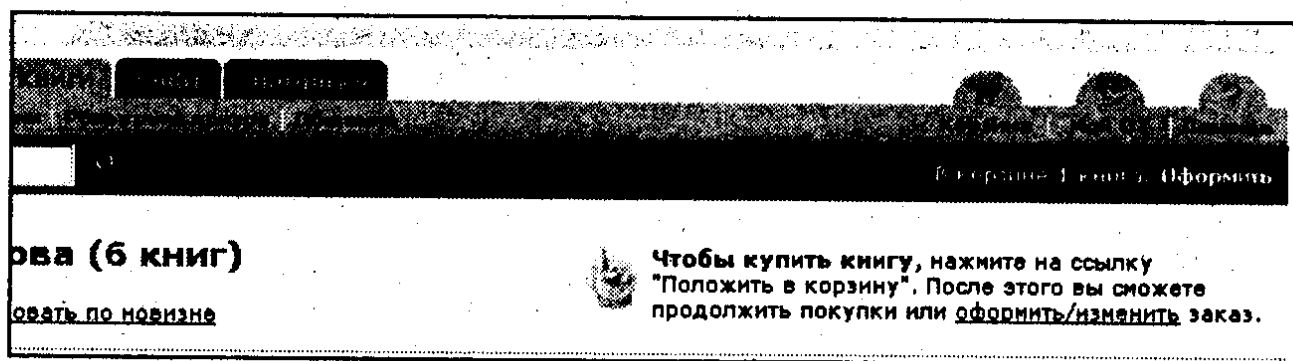


Рис. 9.3. Контекстная помощь на сайте интернет-магазина oz.by

К контекстной помощи относятся пошаговая подсказка и заголовок формы. Они помогают сориентироваться в текущем действии. Кроме того, если посетитель отвлекся, то такая подсказка поможет быстро вернуться к полноценной работе с системой. Контекстная помощь должна быть небольшого размера и работать по принципу «если хотите сделать это — делайте так». Располагать ее следует где-то на открытом месте, чтобы она не путалась и не терялась в гуще информации, и точно в месте следующего действия (рис. 9.4).

Оптимальный объем контекстной помощи подобрать сложно. Поскольку она используется на том же уровне, что и основные функции системы, то она должна быть компактной. С другой стороны, нельзя делать ее слишком краткой, т.к. в этом случае она потеряет свой смысл. Поэтому в каждом случае возможный объем контекстной помощи зависит от потребности в подсказке, ее местоположения на странице и способа реализации.

#### 9.4. Всплывающая подсказка

Всплывающая подсказка в 90 % случаев — тот самый светло-желтый прямоугольник с поясняющим текстом, который появляется при наведении курсора на определенный элемент (рис. 9.5).

Она может быть реализована при помощи атрибута `<title>`, который есть у большинства HTML-тегов. Написать ее разработчику нетрудно, но порой она очень помогает пользователю, особенно если речь идет о пиктограммах и встроенных в текст гиперссылках.



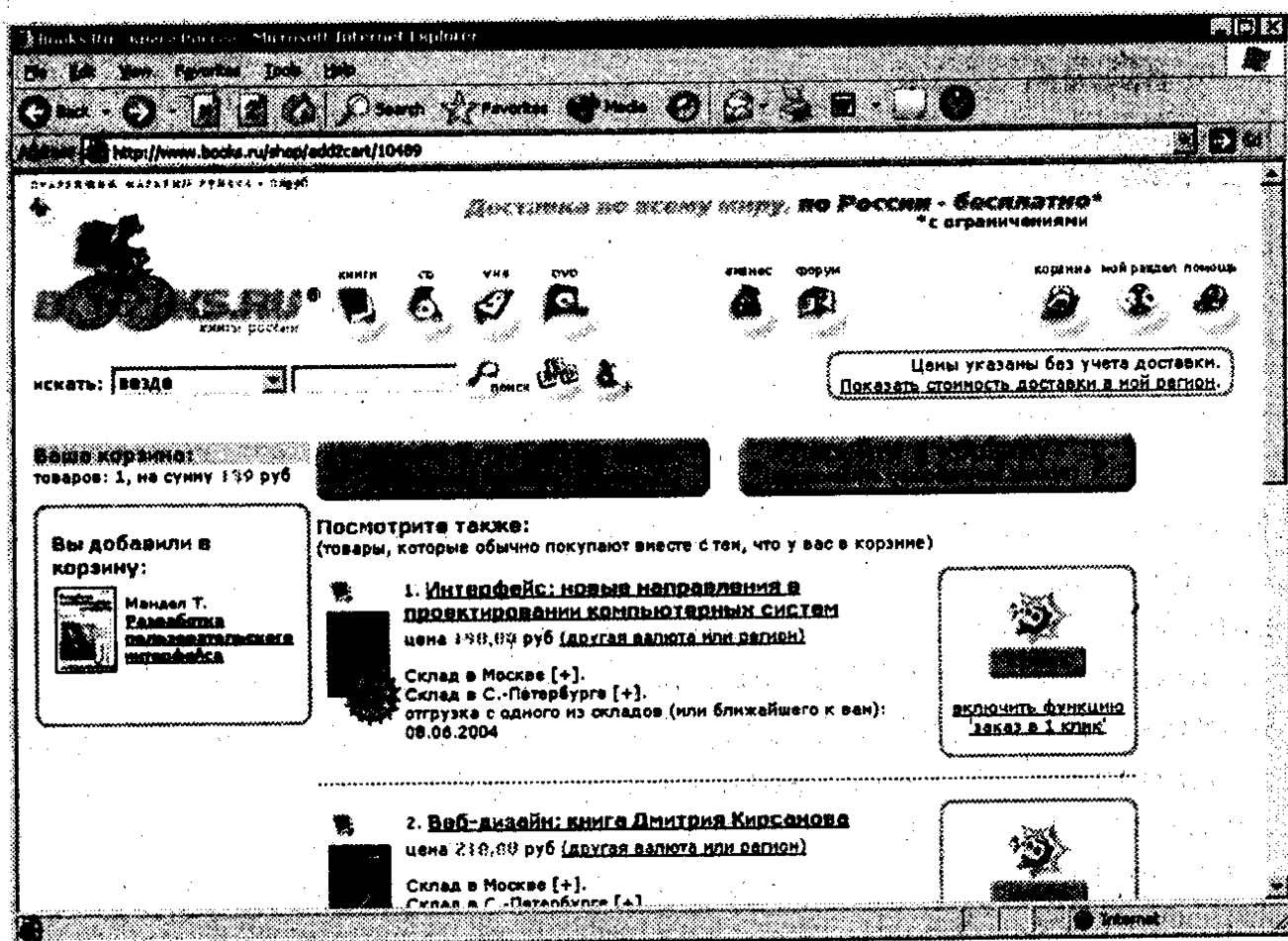


Рис. 9.4. Обратите внимание на две большие кнопки, одна из которых со стрелкой. Очевидно, что она предназначена для дальнейшего действия. Кроме того, сразу появилась информация о том, что есть в моей корзине (слева)

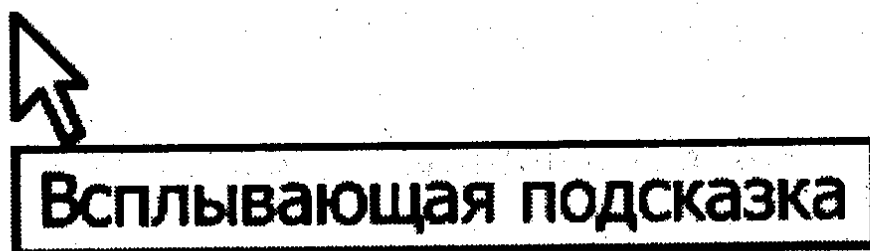


Рис. 9.5. Классическая всплывающая подсказка

Другая разновидность всплывающей подсказки — самостоятельно реализованный элемент, реагирующий на наведение курсора (реже на щелчок мыши). К такой всплывающей подсказке прибегают те, кого не удовлетворяет стилевое оформление стандартного элемента. Иногда во всплывающую подсказку помещают

сноску, а иногда едва ли не таблицу с элементами, необходимыми в данный момент (рис. 9.6). Мне кажется, такое применение всплывающей подсказки разрушает ее концептуальный смысл.

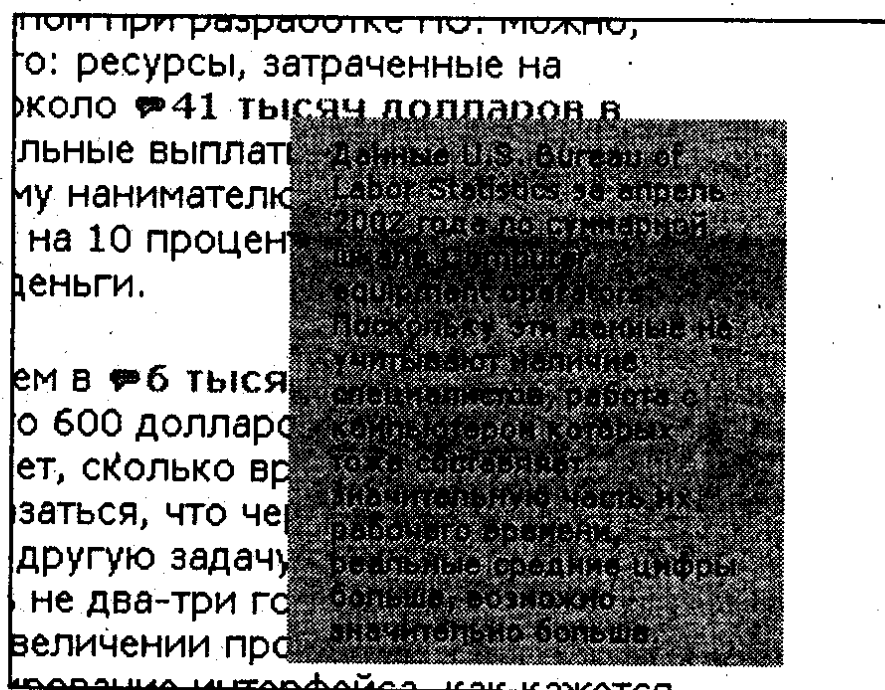


Рис. 9.6. Нестандартная реализация всплывающей подсказки на [ddd.exmachina.ru](http://ddd.exmachina.ru)

Всплывающие подсказки на всех навигационных элементах форм значительно облегчат работу пользователей. Главное — надписи должны четко и конкретно описывать действие, которое произойдет. Использование глагольного наклонения здесь не так критично, как в надписях на кнопках, т.к. пользователь воспринимает надпись на всплывающей подсказке именно как подсказку, а не как приказ системе. Возможны надписи типа «Почта», «Сохранение на диск» и т.п.

Иногда можно встретить «клоны» данного решения, предназначенные для других целей и не относящиеся к системе справки и помощи (ну разве что только к помощи) (рис. 9.7).

## 9.5. Статусная строка

Я считаю, что использование статусной строки как элемента системы справки и помощи в Web-приложениях нежелательно. Во-первых, большинство пользователей в процессе работы туда

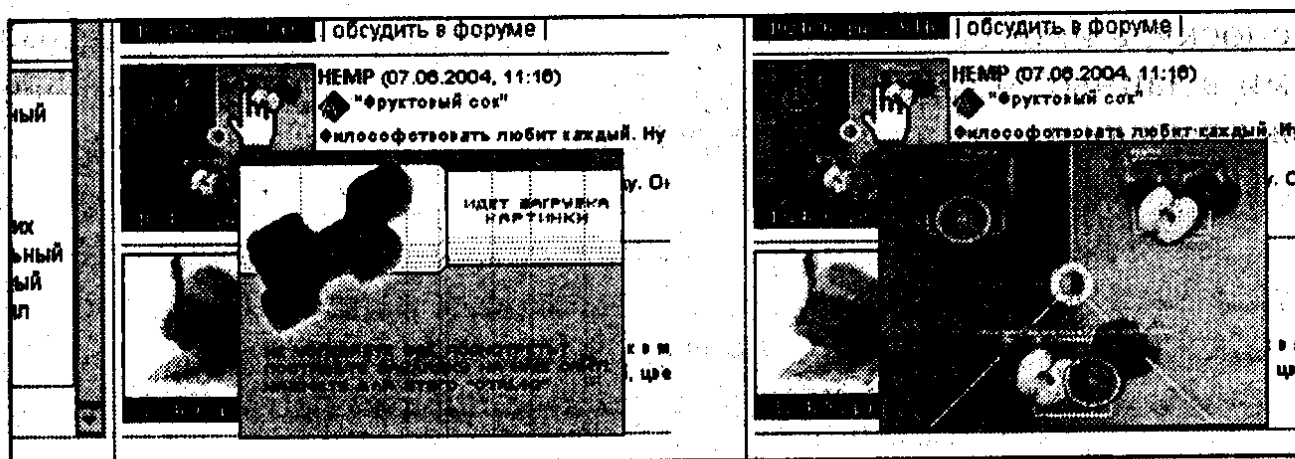


Рис. 9.7. Предварительная загрузка увеличенного изображения на сайте wallpaper.ru, а перед ним надпись со смыслом «подождите, загружается картинка». Это вовсе не подсказка, а только подобная ей идея

не смотрят. Во-вторых, ее надо оставлять свободной для отображения адреса гиперссылки. Опытные пользователи привыкли смотреть «под ноги», т.е. на статусную строку, и решать, идти ли по ссылке, заканчивающейся на `/cgi/?click56784/banner-counter`.

А еще бывают бесполезные статусные строки, которые только отвлекают пользователя (рис. 9.8).

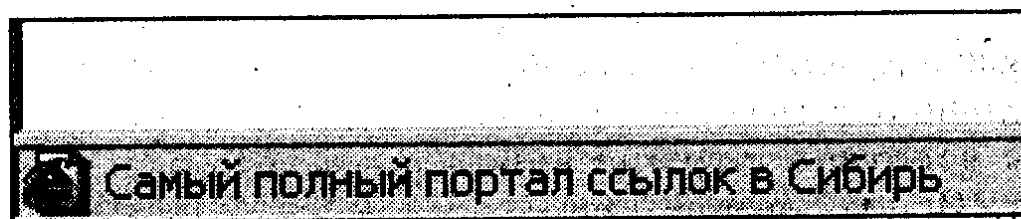


Рис. 9.8. Бесполезная и даже несколько вредная статусная строка

Подсказки в статусной строке хороши в обычных приложениях (рис. 9.9). В Web их мало кто будет замечать. По моему мнению, пользователи Web настолько не привыкли смотреть на статусную строку, а разработчики сайтов так редко пишут в ней что-то полезное, что на данный момент она не играет никакой роли в улучшении юзабилити.



Рис. 9.9. Весьма полезная статусная строка редактора Photoshop

## 9.6. Дополнительная поддержка

Существуют элементы страниц, которые можно отнести только к своего рода «дополнительной поддержке» пользователя. К таким элементам можно причислить список найденных разделов, результаты поиска, запросы (рис. 9.10). В большей части дополнительная поддержка актуальна до завершения работы системы, т.е. в пределах сессии.

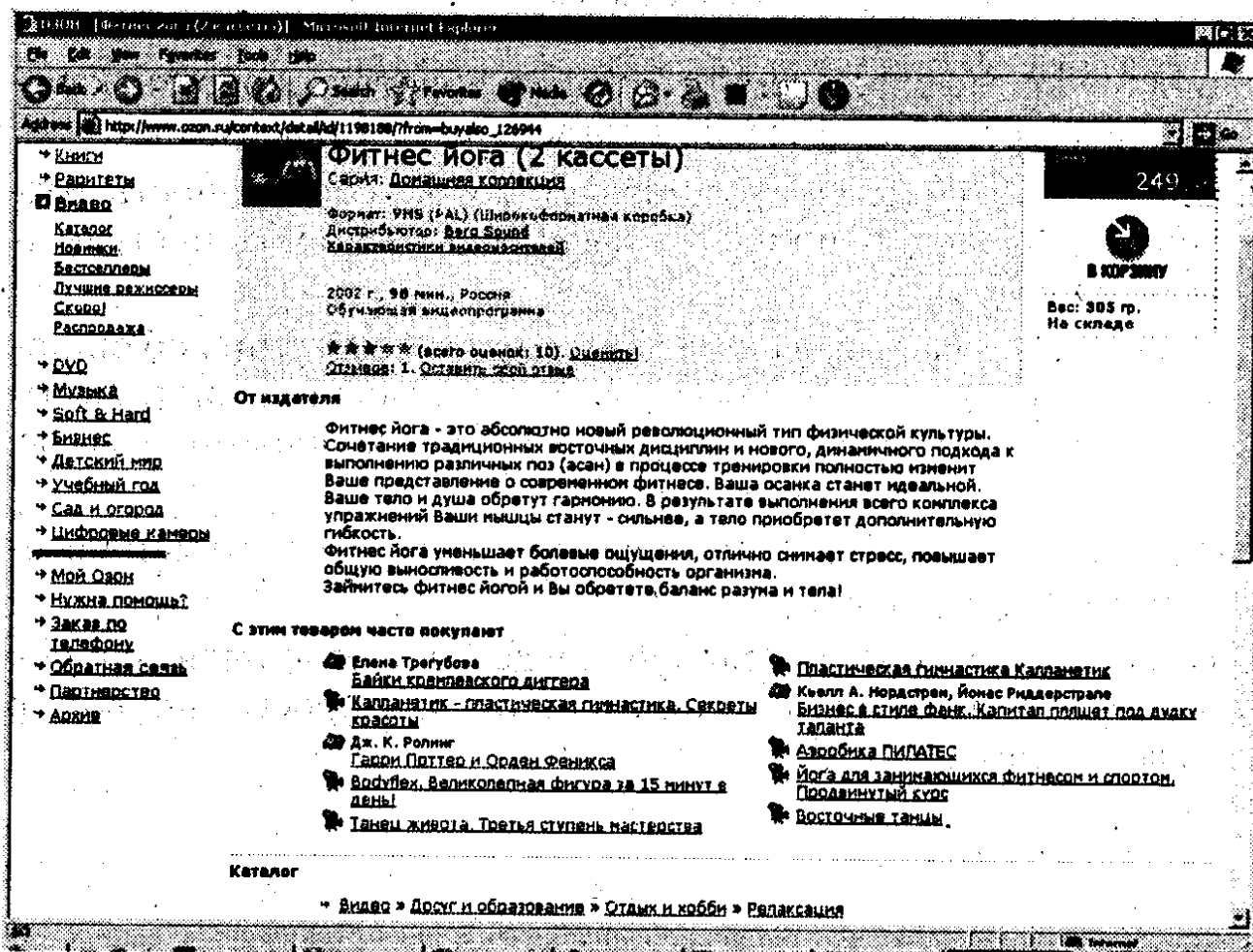


Рис. 9.10. Обратите внимание на информацию о том, что часто покупают с этим товаром в магазине ozon.ru. Ниже приведен путь к данному товару в каталоге. Это и есть дополнительная поддержка

Часто к элементам дополнительной поддержки относят сообщения о проблемах во время работы (в крайнем случае, ссылку Написать администратору о проблемах), хотя, как мне кажется, это помощь не пользователю, а разработчику.

Наконец, «чистопородный» элемент дополнительной поддержки — стандартный встроенный в браузер индикатор хода процесса, или окно ожидания. Этот элемент позволяет пользователю судить о том, что система не «уснула», не «зависла», а реагирует на вводимую информацию и до завершения обработки осталось примерно столько-то времени.

Элементы дополнительной поддержки делают работу пользователя более спокойной и комфортной.

## Резюме

1. Вид и полнота справочной системы влияют на скорость, качество и удобство восприятия пользователем информации.
2. Справочная система представляет собой совокупность различных руководств по использованию и настройке программного продукта.
3. Способы входа в справочную систему: из сообщения об ошибке, переход к конкретному разделу справочной системы на каком-то конкретном шаге, переход к корневому разделу справочной системы.
4. Контекстная помощь — помощь, оказываемая системой посетителю по ходу действия.
5. К элементам дополнительной поддержки относятся: список найденных разделов, результаты поиска, запросы, индикатор хода процесса.

## Errare humanum est<sup>1</sup>

### 10.1. Кто виноват?

Перед началом разговора об ошибках я должен сказать то, что обычно говорю людям, которые утверждают, что компьютер работает неправильно. Неправильно работать может только человек. Компьютер лишь выполняет приказы на уровне нулей и единиц — ни больше ни меньше. Что ему говорят, то он и делает. Если что-то выполняется неверно, значит, ошибка в программе. Если нет ошибок в программе, как гласит один из законов компьютерного мира, значит, необходимо искать ошибки в компиляторе.

Любая программа также работает без ошибок. Она делает именно то, что приказал ей программист. Если он посчитал, что 64 килобайта памяти будет достаточно, то программа никогда не примет самостоятельного решения об изменении этого числа. Если программа утверждает, что  $2 + 2$  равно 5, то она не ошибается. Она работает по тому алгоритму, который задал ей программист. Он заставляет ее утверждать, что эта сумма равна пяти.

Таким образом, существуют только человеческие ошибки. Эти ошибки имеют совершенно разные ипостаси, они присутствуют в проектировании, в использовании системы, в программах, в построении интерфейса. Практически любая ошибка в работе системы так или иначе может быть устранена. Если вам кажется, что устранить ее нельзя, значит, вы просто не знаете способ, как это сделать.

### 10.2. Типы ошибок

Знание типов ошибок позволяет легче их обрабатывать и предотвращать. В основном ошибки делятся на такие категории:

- ошибки ввода данных пользователем;
- ввод неверных данных;
- грамматические ошибки;

---

<sup>1</sup> Человеку свойственно ошибаться (лат.)

- ошибки системы;
- ошибки незнания предметной области;
- ошибки точности движений;
- инерционные ошибки.

Выделенные типы часто пересекаются. Например, ошибки точности движения могут быть практически идентичны инерционным.

### **10.3. Суть предотвращения ошибок**

Качественно спроектированная система должна думать на шаг вперед пользователя и подстраховывать его, а не спасать, когда он уже «тонет». Если представить себе отсутствие механизмов предотвращения ошибок в обычных электрических приборах, то последствия могут выглядеть угрожающе. Ваш «последнее-слово-техники» утюг, который вы случайно поставили на несовместимый с выпуском пара режим, вместо того чтобы включить предупреждающую лампочку и сбросить опасные установки, перегрелся и сгорел, а после этого стал мигать индикатор неисправности с надписью «Утюг выполнил недопустимую операцию и будет расплавлен».

Предотвращение ошибки — устранение возможности ввода неверных данных или совершения неправильных с точки зрения системы действий.

### **10.4. Отношение пользователей к сообщениям об ошибке**

Чтобы правильно составить сообщение об ошибке и лишний раз не нервировать пользователя, вам следует знать его отношение к такого рода сообщениям. Пользователи привыкают к сообщениям об ошибках, особенно при работе в интранет-приложениях и в интернете. Иногда они даже начинают их читать. Всякую систему большинство воспринимает как собеседника. Будет ли этот собеседник воспитанным и предупредительным, зависит от тех, кто разрабатывал систему.

Сообщения об ошибках, по «классификации» некоторых пользователей, бывают страшными, дурацкими и неправильными.

*Страшных* сообщений пользователи пугаются. Они воспринимают ужас ошибки по своей шкале на основании, например, пиктограммы или знакомого слова в сообщении. Мне известен случай, когда процент страха перед системой был уменьшен только за счет замены пиктограммы с внушающим ужас крестиком на красном фоне на более спокойный восклицательный знак в желтом треугольнике.

*Дурацкими* сообщениями они считают те, которых не понимают. «Введите дату отгрузки товара перед регистрацией автоматизируемого бизнес-процесса». «Вы уверены, что идентификатор корректности в допустимых пределах?». «Внимание! Число не является буквой». И так далее... С пользователями надо говорить на их языке.

*Неправильным* пользователи считают сообщение, в котором говорится, что они не правы. А ведь пользователь всегда прав. Если пользователь вводит IP-адрес компьютера в сети, и в этом адресе, по его мнению, есть число больше 255, то он прав до тех пор, пока вы не скажете ему, что это значение может находиться только в пределах от 0 до 255. Если же вы вместо этого говорите, что «невозможно соединиться с указанным хостом», то пользователь считает, что система работает неправильно, а он прав.

☑ Достаточно комичная ситуация произошла однажды с одним знакомым программистом. Он написал программу на Delphi, в которой допустил неточность. Программа выдала сообщение «Непредсказуемая ошибка. Обратитесь к разработчику». Он с непониманием посмотрел на экран, почесал затылок и сказал: «Так я же и есть разработчик, мне даже спросить не у кого, почему она не работает».

Часто пользователи запоминают необходимую, по их мнению, информацию об ошибке или способ ее устранения и в дальнейшем его применяют. В случае проблем они порой достаточно дисциплинированы и обращаются к системе справки и поддержки.



## 10.5. Ошибки ввода данных и их предотвращение

Когда я работал тестировщиком программного обеспечения, моим излюбленным способом поставить систему «на колени» был ввод заведомо неверных и так называемых граничных данных. Например, нескольких букв в поле для ввода даты, запрос на поиск в базе и вывод на экран списка из 10 000 документов, ввод длинной беспробельной строки или символов «<» в HTML-форме. Все эти действия относятся к ошибкам ввода данных и являются далеко не самыми безобидными для деятельности системы. Некоторые из них могли ввести систему в такой «клинч», что приходилось едва ли не перезагружать сервер. Этот же способ можно использовать для анализа ошибок ввода данных. Позже подобные ошибки стали обрабатываться и предотвращаться самой системой. Так, при нахождении 10 000 документов мне предлагалось либо уточнить поисковый запрос для лучшего результата, либо автоматически включался механизм работы с порциями по 100 (50, 20) документов. Символы «<» и «>» просто «проглатывались» и обрабатывались специальным нормализатором ввода, а беспробельные строки разрезались на куски по сто символов. Пользователи, не став более внимательными, совершали меньше ошибок и работали быстрее.

Ошибки ввода данных могут быть предотвращены такими способами:

- уменьшение самостоятельности пользователя при выборе данных (если команда выбирается из предоставляемого системой списка, ошибку допустить намного сложнее, чем в командной строке);
- ограничения в способе ввода данных (предоставление выбора численных значений только из выпадающего списка);
- валидация (проверка корректности) и нормализация (приведение к «безопасному» виду) системой данных перед тем, как принимать их от пользователя (например, проверка наличия знака «@» в адресе электронной почты или проверка длины пароля).

Валидация бывает двух видов: прикладная и системная. **Системная валидация** — проверка корректности информации на уровне типа данных, являются ли введенные данные строкой, числом, временем, адресом электронной почты и т.п. **Прикладная валидация** — проверка диапазона данных конкретного типа (равно, не равно, больше, меньше, от и до). Типы не сильно отличаются: прикладная валидация скорее частный случай системной. Если пользователь должен ввести год своего рождения, то при системной валидации мы проверяем, все ли введенные символы есть цифры, а при прикладной мы предполагаем, что человек не живет больше 150 лет, и ограничиваем ввод данных числом в промежутке от 1854 до 2004.

Простейшими примерами предотвращения ошибок ввода данных могут служить элементы Календарь и Выбор из списка, используемые вместо простых полей для ввода даты. Рассмотрим несколько вариантов на примере ввода в форме даты рождения.

1. *Составной компонент из трех простых полей ввода* (рис. 10.1).

Введите дату Вашего рождения (дд мм гггг)

день  месяц  год

Рис. 10.1. Компонент ввода даты рождения реализован при помощи трех простых полей ввода

Минусы этого варианта:

- компонент выглядит несколько громоздко и занимает много места;
- пользователь может ввести данные, не соответствующие формату;
- пользователь может ошибиться при выборе формата ввода месяца («апрель»);
- ввод данных осуществляется с клавиатуры, поэтому необходимо отвлекаться, чтобы совершить это действие, кроме того, возможны опечатки.

Плюс у такого варианта только один: реализация подобного компонента требует минимальных знаний и усилий.

2. Составной компонент из трех полей «выпадающий список» (рис. 10.2).

Введите дату своего рождения

День	<input type="text" value="01"/>	Месяц	<input type="text" value="сентября"/>	Год	<input type="text" value="1967"/>
------	---------------------------------	-------	---------------------------------------	-----	-----------------------------------

Рис. 10.2. Компонент ввода даты рождения реализован при помощи трех полей «выпадающий список»

Минусы этого варианта:

- компонент также выглядит несколько громоздко и занимает много места;
- листать и просматривать список может быть неудобно из-за большого количества элементов;
- возможно произвольное изменение пользователем введенных данных при использовании колеса прокрутки в то время, как фокус установлен в каком-либо поле «выбор из списка».

Плюсы этого варианта:

- пользователь не может ввести данные, не соответствующие формату (если предусмотреть проверку соответствия месяца и количества дней);
- реализация подобного компонента не требует глубоких знаний.

3. Компонент ввода при помощи выбора даты из календаря (рис. 10.3).

Введите дату Вашего рождения		<input type="text" value="2004.04.07"/>
------------------------------	--	---

Date

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Рис. 10.3. Компонент ввода даты рождения реализован как выбор даты из календаря


### Минусы этого варианта:

- листать и просматривать список может быть неудобно из-за большого количества элементов;
- маловероятно, но все же возможно произвольное изменение введенных данных;
- необходимо догадаться, что пиктограмма с листком календаря не просто украшение;
- некоторые пользователи попытаются ввести дату в недоступное для редактирования поле (при этом установленная по умолчанию дата в поле ввода может вообще поставить пользователя в тупик);
- реализация компонента требует серьезных знаний и дизайнерских навыков.

### Плюсы этого варианта:

- компонент в пассивном состоянии занимает мало места;
- пользователь не может ввести данные, не соответствующие формату (если предусмотрена проверка соответствия месяца и количества дней).

Проанализировав эти варианты, получаем, что худший — первый, а оптимальный — второй. Если рассматривать возможные способы ввода даты рождения только с позиции предотвращения ошибок, то пользователь не совершит ошибку при вводе данных во втором и третьем вариантах. Кроме того, сложные программы наиболее подвержены ошибкам программистов. Скорее всего программист напишет календарь на JavaScript, а какой-либо из браузеров наверняка откажется его корректно отобразить. Наряду с требованиями по предотвращению ошибок необходимо учитывать и юзабилити. С этой точки зрения оптимальным кажется составной компонент из трех полей «выпадающий список».

 Я сократил количество вариантов, устранив те, что явно будут неудобны пользователю. В варианте с календарем я рассматриваю сам принцип, а не способы организации календаря, ведь его тоже можно исполнить по-разному.

## 10.6. Ввод неверных данных и грамматические ошибки

Ввод неверных данных предотвратить, к сожалению, практически невозможно. Если пользователь вводит корректный с точки зрения системы, но неверный адрес электронной почты, никто не скажет ему: «Приятель, у твоего знакомого не такой адрес — письмо не дойдет».

Если пользователь хочет ввести в какой-либо программе, предназначенной для работы в интернете, IP-адрес компьютера, с которым ему надо соединиться, то попытки предотвратить ошибку соединения при неправильном указании IP могут привести к абсурдной ситуации. Представьте себе, что вы звоните другу на мобильный телефон, а оператор соединяет вас с другим человеком, потому что ваш друг «временно недоступен» или внезапно на вашем телефоне перестает нажиматься кнопка вызова абонента после набора номера.

Грамматические ошибки, если они не являются разновидностью ввода неверных данных (например, ввод неправильно написанной фамилии в строке для поиска), не могут быть критическими для работы системы. Предупреждение таких ошибок также фактически невозможно. Данные вводит пользователь, а он может писать на любом языке, доступном в его операционной системе, может намеренно совершать ошибку, если, например, пишет электронное письмо, где цитирует своего годовалого сынишку. Максимум, что можно придумать, — это проверка орфографии в полях ввода формы. Но до такого чуда в Web, видно, еще далеко.

## 10.7. Ошибки системы

Во время работы пользователей происходит огромное количество взаимодействий внутри системы: соединение с сервером, поиск в базах данных, обращение к различным серверным компонентам, к страницам как существующим, так и к несуществующим, ожидание ответа от внешней системы, если она сильно загружена. Во время этих взаимодействий может произойти что угодно,

например, прерваться соединение, «зависнуть» операционная система, истечь время ожидания ответа на запрос и т.п. Не всегда такого рода проблемы можно четко идентифицировать, да и к тому же рядовому посетителю неинтересно, что на сервере, к которому он обратился, отсутствует jdbc-драйвер или изменились настройки файла `htconf`. Поэтому такого рода ошибки обрабатываются следующим образом: пользователю выдается некоторое читаемое сообщение, например, «Из-за проблем на сервере соединиться с указанным ресурсом не удалось. Попробуйте позже...» или «Ошибка обращения к базе данных. Попробуйте обратиться к администратору». На уровне интерфейса данные ошибки предотвратить невозможно.

### **10.8. Ошибки незнания предметной области и их предотвращение**

Данная разновидность ошибок на работу системы в принципе не влияет. Она влияет на корректность данных с точки зрения предметной области системы. Часто правильная последовательность действий, которую необходимо выполнить системе, бывает оставлена, что называется, на совесть пользователя. Таким образом, система не контролирует четкое соответствие бизнес-процессу, а только предоставляет человеку необходимые сервисы по работе с информацией.

Представьте себе систему регистрации участников международного семинара. Сначала необходимо зарегистрировать заявку на участие, потом самого участника, потом аккредитовать его, поселить в гостинице, а через некоторое время выселить и отметить его убытие. В системе может быть не предусмотрен контроль соответствия объектов и операций над ними. Например, может отсутствовать запрет на внесение отметки о выселении, если участник еще не прибыл. В таком случае нерадивый пользователь сможет разместить заявку на участие в гостинице, выдать талоны на питание автомобилю и выселить того, кто еще не приехал.

Ошибки незнания предметной области чаще всего встречаются в интранете, где пользователей едва ли не обязывают учиться работе с системой, либо это выгодно им самим.

Методами предотвращения подобных ошибок являются: обучение пользователей, наличие справочной информации и подсказки. Однако тут есть некоторые проблемы. Пользователи во Всемирной паутине не желают обучаться. Никто не будет записываться на авторизованные курсы Microsoft, чтобы уметь пользоваться услугами, предоставляемыми их сайтом. Постарайтесь проработать интерфейс таким образом, чтобы посетителям не приходилось тщательно изучать систему, прежде чем начать с ней работать.

Справку и помощь также мало кто читает, ведь куда проще обратиться к другому ресурсу, где все понятно и не надо будет напрягаться. Однако все же существует некоторый процент пользователей, которые могут обратиться к справке. Именно для них и предусмотрена эта система.

Подробная и понятная подсказка хороша лишь тогда, когда пользователь работает с системой в первый или во второй раз. Затем она будет только мешать, отвлекать и занимать место. Поверхностная и сумбурная подсказка запутывает и приносит больше вреда, чем пользы. Подсказки целесообразно использовать в том случае, если в системе каждый день меняется состав пользователей, подключается большое количество новых посетителей, которые не работают в ней постоянно.

## **10.9. Ошибки точности движений и их предотвращение**

Как говорилось ранее, человек не вчитывается в надписи и не всматривается в картинки, если он работает с программной системой не первый раз. Он видит нечто похожее на то, что ему надо, и расположенное приблизительно там, где обычно, и нажимает на это. Я сотни раз вместо иконки для запуска нужной мне программы в меню Пуск нажимал на соседнюю. По закону подлости запускался какой-нибудь тяжеловесный «монстр» вроде 3D Max, и приходилось долго ждать, пока я смогу работать дальше. Впоследствии я уже более аккуратно нажимал на кнопку мыши и проверял местоположение курсора. Такая же история часто происходит и на некоторых сайтах, когда я целюсь в одну ссылку, а попадаю по другой. Иногда такие ошибки могут нанести

вред информации, которую вводит пользователь. Например, он отмечает чек-бокс напротив элемента, который хочет удалить, затем переносит фокус внимания на кнопку Удалить, нажимает на нее и снова возвращается взглядом к списку элементов. И он опять видит то, что секунду назад удалил! «А что же я тогда стер?!» Такая ситуация возможна, если человек совершил неточное движение и выделил другой элемент.

Предотвращение ошибок точности движения рассматривалось в главе «Юзабилити в теории и на практике», остается лишь повториться. Необходимо увеличивать мелкие детали, использовать для навигации не только сам текст гиперссылки, а еще и некоторое окружающее пространство, разносить близкие мелкие элементы и увеличивать расстояние между ними. Компонировка страницы должна быть сделана таким образом, чтобы элементы, которые выглядят практически одинаково, не находились рядом.

### **10.10. Инерционные ошибки и их предотвращение**

Инерционные ошибки, как понятно из самого названия, совершаются по инерции, т.е. пользователь выполняет какое-либо действие «разогнавшись» и не успевает вовремя остановиться. Как правило, эти ошибки совершаются опытными пользователями. Такие посетители, во-первых, предугадывают, что сейчас произойдет, во-вторых, не думают над каждым шагом, в-третьих, виртуозно пользуются мышью и клавиатурой. Иногда их руки опережают разум, т.к. все знакомые и часто повторяемые действия переведены в привычку и отработаны до автоматизма.

Типичными примерами такого рода ошибок являются быстрое нажатие на кнопку ОК в диалоге предупреждения, закрытие внезапно появившегося всплывающего окна, которое ассоциируется только с рекламой, нажатие на кнопку Удалить, которая расположена в том месте на экране, где чаще всего находится кнопка Сохранить и др.

Зачастую инерционные ошибки совпадают с ошибками точности движения. Работая с электронной почтой, я вместо удаления письма иногда нажимаю на Добавить отправителя в адресную книгу.



Так получается только потому, что пиктограммы находятся рядом, а я быстро перевожу курсор из нижней части экрана в верхнюю и, не успев остановиться, нажимаю кнопку мыши.

Предотвратить инерционные ошибки можно, используя те же приемы, что и для ошибок точности движения, а также единообразием интерфейса всей системы, например, располагая кнопки ОК и Отменить всегда в одном месте (допустим, ОК слева, Отменить справа и всегда в правой части формы). Тем самым мы вырабатываем у пользователей привычки в работе с нашей системой. Можно использовать однооконный интерфейс, чтобы предупредить закрытие пользователем внезапно появившегося всплывающего окна или не допустить упущения обновившейся информации в уже открытом втором окне.

## 10.11. Устранение побочных эффектов

Под побочным эффектом я понимаю не санкционированное пользователем изменение информации и состояния системы, когда совершаемые системой действия скрыты от пользователя и могут не соответствовать тем указаниям, которые он дает. К побочным эффектам можно также отнести случаи неадекватного понимания действий системы. Приведем примеры такого рода побочных эффектов.

Пользователь работает с некоторым набором объектов по результатам поиска. Найдено 100 объектов, которые показываются порциями по 20. При нажатии на кнопку Выделить все объекты может появиться побочный эффект, если разработчик посчитал, что система должна оперировать со всеми ста объектами, а пользователь выделил и собирается удалить объекты только из текущей порции. Каково же будет его удивление и огорчение, когда он перейдет к следующей порции, а она окажется пустой. И наоборот, пользователь может посчитать, что Выделить все объекты относится ко всем найденным, а в системе после удаления останется большая их часть. В таком случае можно предложить использовать оба способа удаления. Сделайте кнопки Удалить из текущей порции и Удалить все найденные объекты, и эта двойственность системы исчезнет. Кстати, нежелательно употреблять термин «выборка» в названиях или пояснениях, потому что для многих пользователей он может быть непонятен.

Примером непоправимого вреда, наносимого системой из-за побочных эффектов, может быть следующий случай. И хотя он имеет небольшое отношение к Web-интерфейсу, он, тем не менее, весьма примечателен. Это история о том, как я сознательно удалил почти весь свой рабочий каталог с жесткого диска. Я работал в любимом мной файловом менеджере FAR (этот случай не подмочил его репутацию, потому что в большей степени виноват был я). Свободного места на диске оставалось немного, и я решил выяснить, какие каталоги занимают больше всего места. Для этого в FAR достаточно выделить их клавишей Insert и нажать F3. В правой части рабочей панели вместо <folder> (папка) отобразятся размеры выбранного каталога. Начав выделять папки, я обнаружил, что у меня есть еще папка «TMP\_Works», в которой лежат старые ненужные файлы, занимающие много места. Я решил почистить только ее, но снять выделения с остальных забыл, потому что в фокусе моего внимания уже была «TMP\_Works». Для удаления мимо корзины я нажал сочетание клавиш Shift + Del. Естественно, система восприняла это как команду на удаление выделенных каталогов, а не того, на котором стоял курсор (как это воспринимал я). Она спросила, действительно ли я хочу удалить четыре выделенные объекта. Я знал, о чем меня спросят, поэтому, не задумываясь, нажал ОК и стал блаженно смотреть на то, как в списке удаляемых файлов пробегают имена ценнейших документов и графики из моего личного каталога. Когда до меня дошло, что я оказался жертвой собственной невнимательности и побочных эффектов, половина информации была навсегда потеряна. Несмотря на то, что многое было сделано разработчиками интерфейсов, чтобы предотвратить это, — и предупреждение об удалении, и Корзина, и яркая подсветка выделенных каталогов, мой более-менее богатый опыт работы с компьютерами погубил каталоги.

Настройки по умолчанию также могут оказаться опасными для системы или для информации. Однажды я выкачивал из интернета большой файл при помощи программы FlashFXP. Эта программа по какой-то причине постоянно теряла соединение, кроме того, она еще и неожиданно всплывала поверх всех окон после обрыва соединения. Я набирал текст, а когда скачалось

около двадцати мегабайт, внезапно выскочило окошко. Меня спрашивали, перезаписывать ли файл при восстановлении соединения или продолжать запись. Как раз в эту секунду я нажал на клавишу Ввод, считая, что по-прежнему набираю текст. А разработчики FlashFXP по умолчанию установили фокус на кнопку Перезаписать (Overwrite), и все двадцать мегабайт информации были удалены. Вот вам и «человекоориентированный» интерфейс. Я готов был сломать клавиатуру от злости. Естественно, что правильнее было бы не делать таких настроек по умолчанию, а вместо всплывающего окна FlashFXP можно было просто вывести мигающее предупреждение в панели задач.

## 10.12. Обработка ошибок и обратная связь

Как считает большинство экспертов по интерфейсам и юзабилити, всякое сообщение об ошибке является изначально порочным. Однако могут сложиться ситуации, когда выдавать его просто необходимо, т.к. выбрать другой вариант не представляется возможным. Невозможно бывает и запретить сделать ошибку. Что ж, как говорил один из героев Жюль Верна, если этого зрелища нельзя избежать, пусть оно будет, по крайней мере, красивым.

В случае ошибки система обязана гарантировать:

- данные, введенные пользователем, никуда не исчезнут — должна быть возможность использовать их повторно;
- пользователь получит четкое и ясное уведомление о том, что же произошло не так, почему, как эту ситуацию можно исправить;
- пользователь получит информацию о том, насколько серьезна произошедшая ошибка и успешно ли завершилось последнее действие;
- пользователь всегда может обратиться с вопросом к компонентам справки и помощи для более подробного изучения системы.

Рассмотрим наиболее частую ситуацию — ошибки в заполнении формы с обязательными полями (рис. 10.4).

Рис. 10.4. Форма с обязательными для заполнения полями

Если при заполнении формы будут пропущены обязательные для заполнения поля, то дополнительно выводится список полей (рис. 10.5).

Рис. 10.5. Не совсем удачное сообщение об ошибке

Сообщение об ошибке занимает много места и требует от посетителя постоянного перемещения фокуса внимания со списка обязательных полей на форму с полями. Это сообщение можно улучшить, если сообщить, что не заполнены некоторые обязательные поля, а сами поля отметить в форме каким-либо явным

образом (рис. 10.6). В таком случае будет сэкономлено место на экране, а посетитель, читая сообщение, будет скользить взглядом сверху вниз, постепенно заполняя все обязательные поля.

Создание записи о сотруднике

Ошибка! Необходимо заполнить все обязательные поля.  
(не заполненные обязательные поля помечены знаком \*)

Фамилия

Имя \*

Отчество

№ кабинета

Электронная почта \*


поля, обязательные для заполнения отмечены полужирным шрифтом

Рис. 10.6. Улучшенное сообщение об ошибке

При выводе подобного сообщения обязательно выводить заполненную ранее форму. Не один раз мне приходилось трудиться над длинными формами регистрации, и, если обязательные поля были не заполнены или валидация полей не проходила, мне возвращали пустую форму, потеряв куда-то все, что я уже заполнил. Есть обстоятельства, которые нельзя четко отнести ни к предупреждению ошибок, ни к обработке. Одним из них является отсутствие в системе заявленных возможностей. Ленивый разработчик сделает все пункты меню гиперссылками, даже если один из них временно не доступен. Он будет выдавать пользователю сообщения типа «under construction» или «извините, данная страница временно недоступна». Не проще ли не давать возможность совершить это действие, убрать на период «временной недоступности страницы» эту гиперссылку, заменив ее на обычный текст? Наверное, это слишком сложно, потому что я сплошь и рядом вижу подобные ссылки и подобные Web-сайты.

### 10.13. Правильное сообщение об ошибке

Правильное описание ошибки должно четко описывать, что вызвало ошибку, предлагать способ решения возникшей проблемы, быть понятным пользователю, возможно, предупреждать его о критичности данной ошибки для информации. А еще оно должно быть вежливым, кратким и ненавязчивым. Как и любая другая важная информация, сообщение об ошибке должно быть «видимым», иначе пользователь совершит ее не один, а три раза.

 Недаром ходят анекдоты про обработку ошибок и предупреждение пользователя в MS Windows.

«Вы пытаетесь закрыть документ. Если вы закроете документ, вы больше не сможете с ним работать. Вам необходимо будет открыть его снова. Закрыть?»

«ОК».

«Вы уверены, что хотите закрыть документ, с которым закончили работать?»

«ОК».

«Закрытие данного документа может привести к нестабильной работе других приложений. По-прежнему настаиваете на закрытии?»

«ОК».

«Ну смотрите, сейчас документ будет закрыт. Мы ни за что не отвечаем».

«OOOKKK!!!!!!»

«Документ закрыт. Чтобы изменения вступили в силу, необходимо перезагрузить компьютер».

### Резюме

1. Ошибки делятся на основные категории: ошибки ввода данных пользователем, ввод неверных данных, грамматические ошибки, ошибки системы, ошибки незнания предметной области, ошибки точности движений, инерционные ошибки.
2. Предотвращение ошибки — устранение возможности ввода неверных данных или совершения неправильных с точки зрения

системы действий. Предотвратить ошибки ввода данных можно, уменьшив самостоятельность пользователя в выборе вводимых данных, ограничив способы ввода данных, валидацией данных.

3. Необходимо выявлять и устранять побочные эффекты, которые также могут привести к ошибкам.
4. В случае ошибки система обязана гарантировать: данные, введенные пользователем, никуда не исчезнут; пользователь получит четкое уведомление о том, что произошло, почему и как эту ситуацию можно исправить; пользователь получит информацию о том, насколько серьезна произошедшая ошибка и успешно ли завершилось последнее действие.
5. На сайте не должно быть гиперссылок, которые ведут к недоступным элементам. Такие ссылки целесообразно заменить на обычный текст.
6. Правильное сообщение об ошибке должно четко описывать, что произошло, предлагать пользователю способ решения возникшей проблемы, быть понятным, «видимым», кратким и ненавязчивым.

# Приложение<sup>1</sup>

## 1. Ozon.ru сменил лицо

Однажды я зашел на сайт ozon.ru и не узнал его: кто-то здорово поработал над интерфейсом этого электронного магазина.

Вот так выглядел старый интерфейс (рис. 11.1).

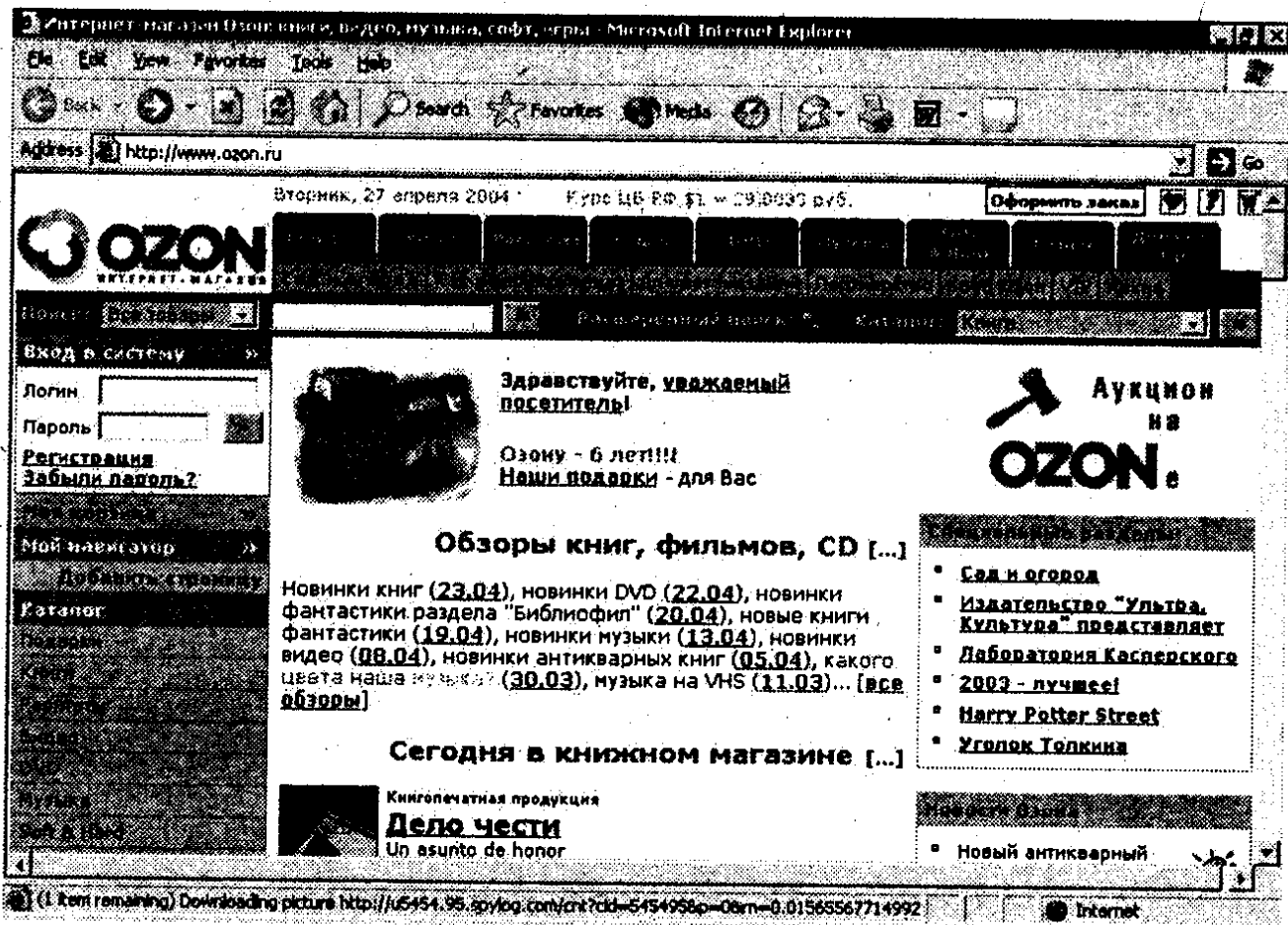


Рис. 11.1. Прежний интерфейс ozon.ru

Каковы были его проблемы и особенности интерфейса?

Ссылка в каталог Книги встречается три раза на странице и еще один раз в «подвале». Это грубое нарушение принципа монотонности интерфейса.

Надпись справа вверху Оформить заказ повторяет ссылку, которая «навешена» на значок корзины совсем рядом (при том, что это уже совершенно привычный для большинства пользователей значок).

<sup>1</sup> В приложении рассмотрены примеры, которые могут вам пригодиться.



Двухуровневое меню неуклюже скопировано со знаменитого amazon.com. Верхний уровень — основные разделы сайта, нижний — его подразделы. Однако визуально не определить, как они подчинены, потому что граница между уровнями не исчезает.

Дизайн кажется «крепким», надежным, словно кирпичный дом, в отличие от нового, несколько «воздушного» варианта. Вот так стал выглядеть интерфейс (рис. 11.2).

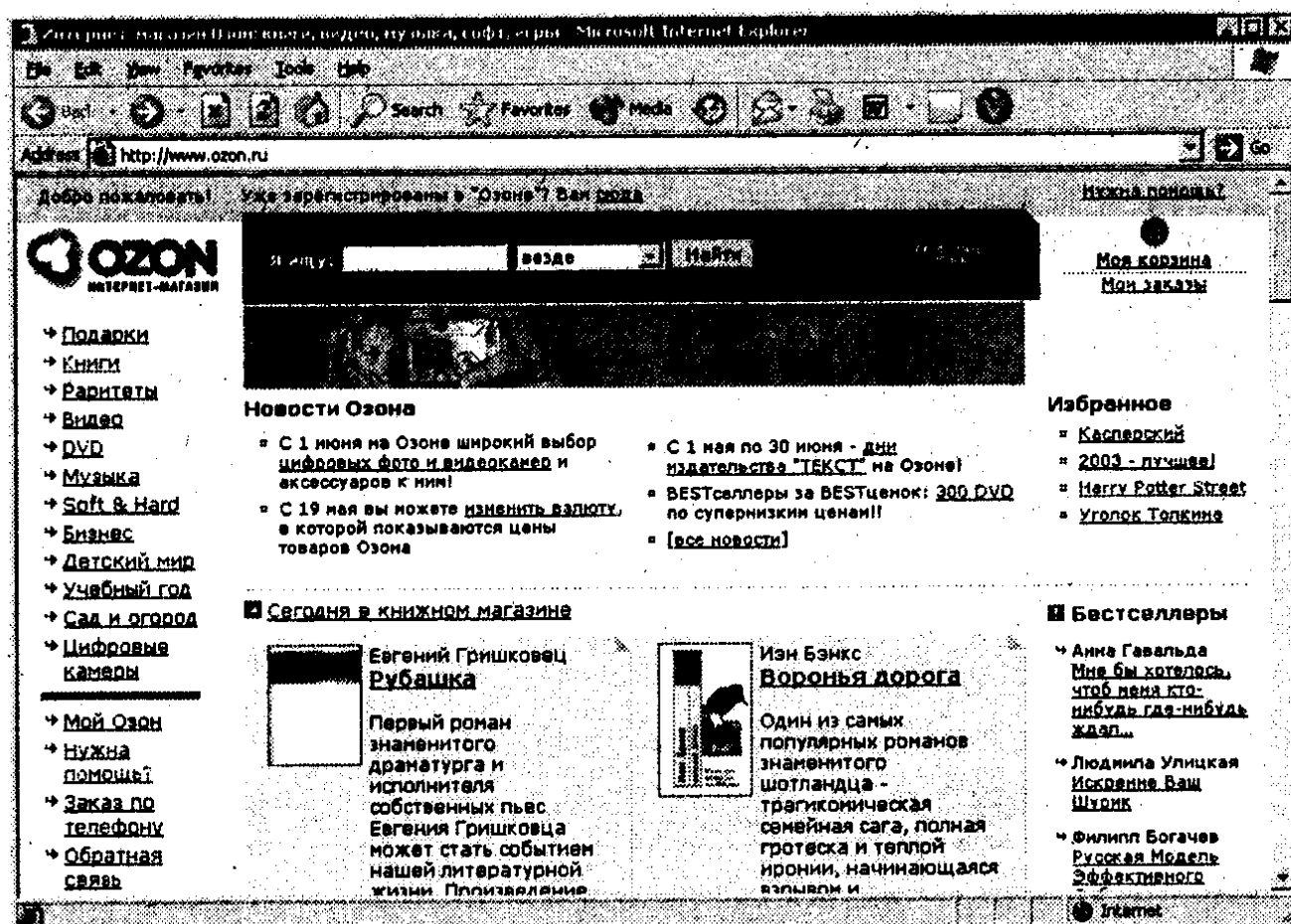


Рис. 11.2. Интерфейс ozon.ru сегодня

Верхнее двухуровневое меню исчезло. Подразделы открываются как древовидная структура в боковом меню. Поиск стал доминирующим элементом на странице в отличие от прежнего интерфейса, где его не было вообще.

Исчезли лишние элементы (блок авторизации, текущая дата и курсы валют).

Пиктограмма корзины с подписью — лучший вариант использования пиктограмм.

## 2. Десять тысяч чек-боксов

Часто в Web-формах можно встретить множество чек-боксов, находящихся рядом. Существует несколько способов, при помощи которых можно управлять большим количеством этих элементов.

**Первый способ.** Не управлять вообще. Пусть пользователь сам отмечает все десять тысяч чек-боксов. За такие вещи следовало бы наказать разработчика и пользователя, который сразу не ушел с такого сайта.

**Второй способ.** Предоставить возможность включать и выключать все чек-боксы одним движением мыши. Тут уже появляется разнообразие решений.

**Прием 1.** Включать все чек-боксы при помощи Java Script, не перегружая форму (страницу). Главное в этом приеме — поддерживать корректную работу в основных браузерах.

**Прием 2.** Включать все чек-боксы при помощи пиктограммы или текстовой гиперссылки. Уже хуже, потому что перезагружается форма и пользователю приходится ждать.

Рассмотрим некоторые варианты управления чек-боксами на конкретных примерах.

### 1. Выбор набора гороскопов (рис. 11.3).

Я недаром выделил на рисунке чек-бкс Отметить все. Он невидим для пользователя, т.к. похож на остальные. Включение/выключение этого чек-бкса вызывает выделение/снятие всего ряда. Это решение хорошее, ведь оно достаточно предсказуемое и очевидное.

Главный минус этого варианта — незаметность управляющего чек-бкса.

### 2. Выбор списка писем в почтовом клиенте на TUT.BY (рис. 11.4).

Чек-бкс Отметить все видим для пользователя, поскольку оформлен в виде понятной пиктограммы (и обладает всеми их недотатками). При нажатии на пиктограмму выделяется весь ряд чек-боксов, при этом сама страница перезагружается. Снять выделения со всего ряда чек-боксов можно только вручную, по крайней мере, я не нашел нужной пиктограммы.

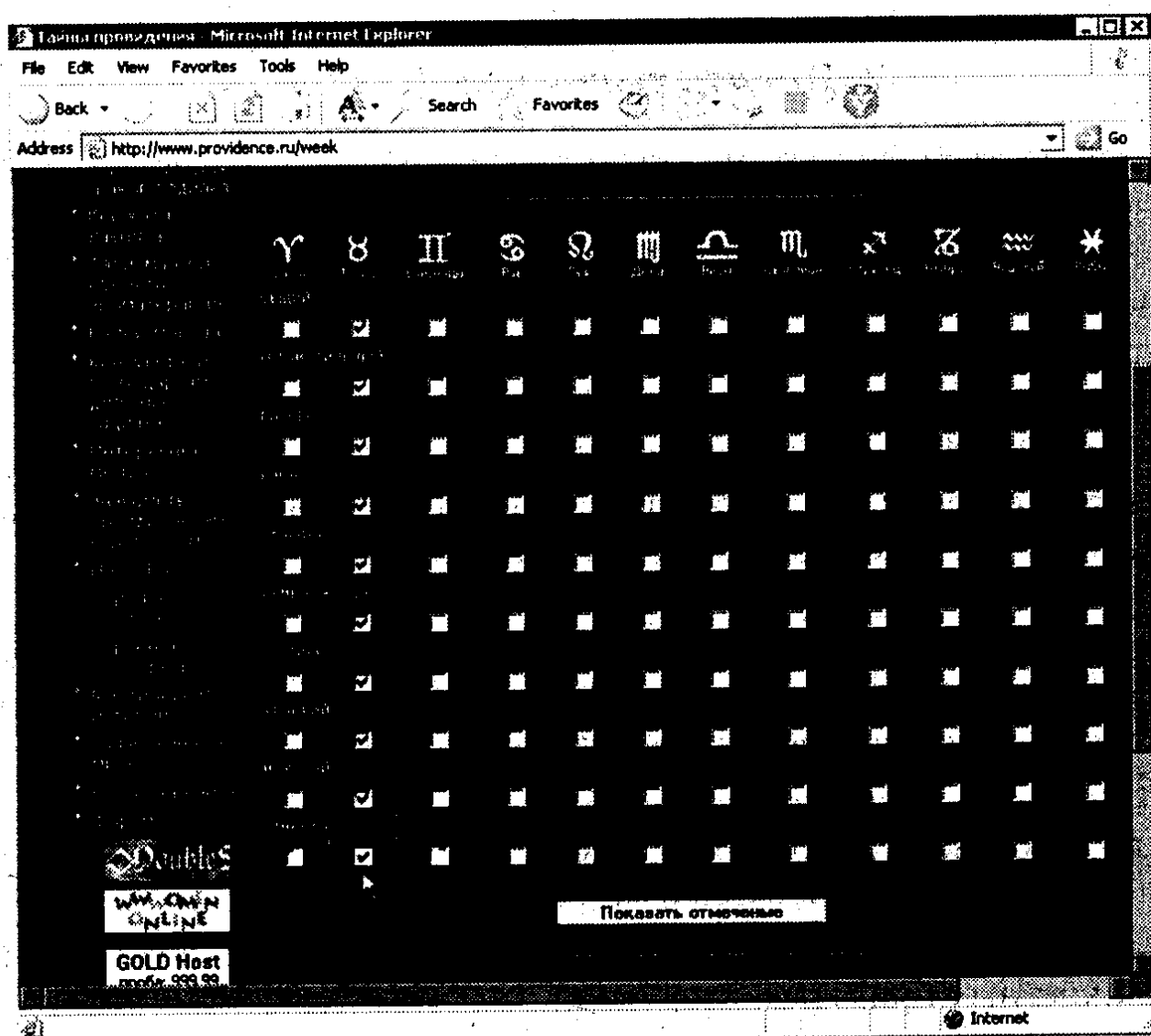


Рис. 11.3. Выделение чек-боксов управляется Java Script

Главные минусы этого варианта — перезагрузка страницы и невозможность снятия выделения автоматически.

### 3. Выбор списка писем в почтовом клиенте на Yahoo! (рис. 11.5).

Данный вариант практически аналогичен списку гороскопов, если не считать того, что чек-боксы более «видимы». Управляющий чек-бокс находится на сером фоне, отличном от фона остальных чек-боксов. Форма не перезагружается, но здесь, как и в списке гороскопов, один и тот же элемент ведет к разным действиям (в зависимости от контекста использования).

Главный минус — незаметность управляющего чек-бокса.

### 4. Выбор разворачиваемых приложений в Sun ONE Application Server (рис. 11.6).

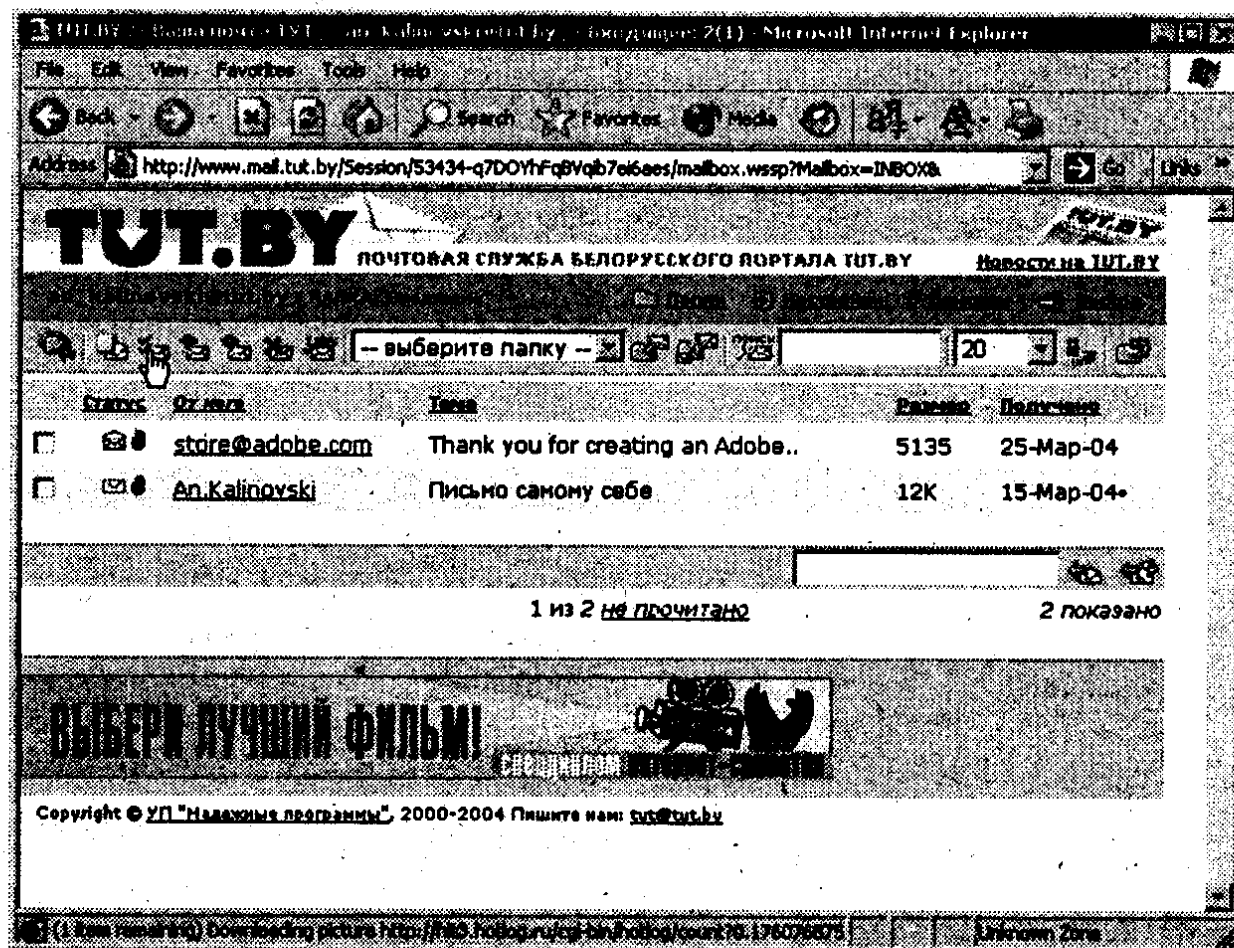


Рис. 11.4. Так можно отметить все чек-боксы на TUT.BY

Выбор/сброс всех приложений происходит при нажатии на одну из двух пиктограмм. Они расположены совсем рядом, мелкие, неочевидные (выглядят как простое украшение столбца таблицы), зато форма не перезагружается.

Главные минусы — назначение пиктограмм неочевидно, и мелкие детали, по которым можно промахнуться.

### 3. Юзабилити форума на books.ru

Юзабилити форума на books.ru оставляет желать лучшего (рис. 11.7).

Во - первых, при переходе по ссылке Форум вы попадаете на страницу, где главной частью всего содержимого оказывается информация о том, что, извините, «выразаться» в форуме нельзя. Во - вторых, посетителю, попавшему на форум первый раз, приходится выискивать, есть ли вообще ссылки на темы форумов. Лично у меня первая мысль была: «А где форум?»

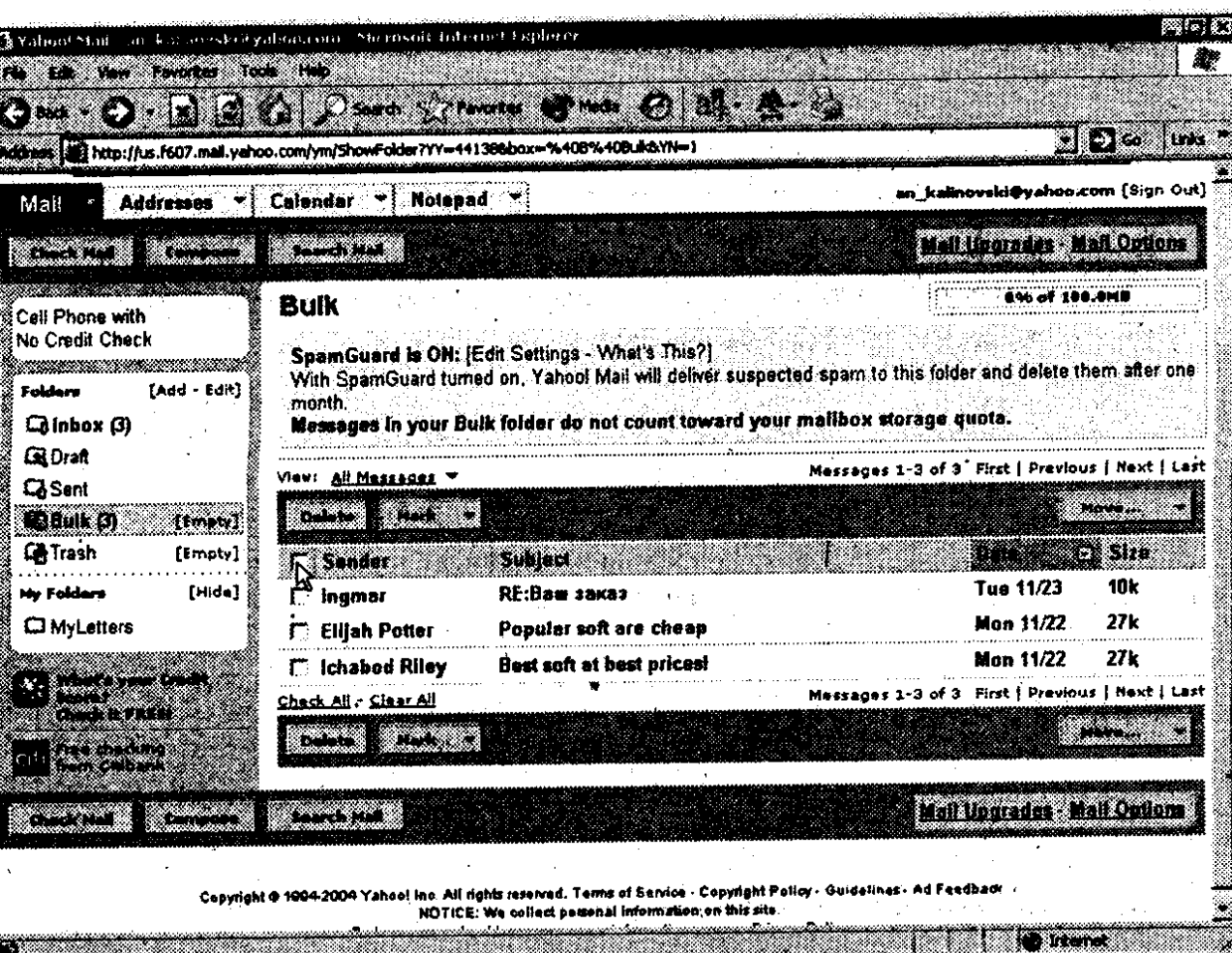


Рис. 11.5. Отметить все письма на yahoo.com

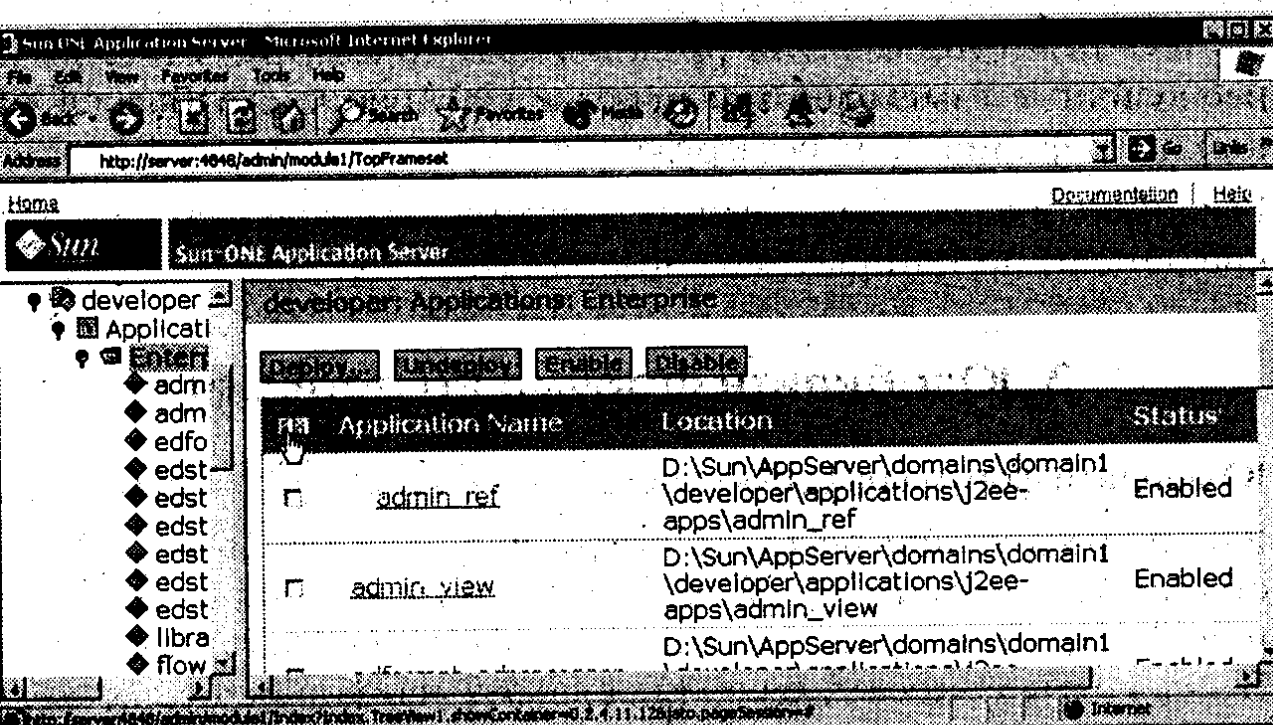


Рис. 11.6. Иконки для отметки чек-боксов на сайте Sun Microsystems

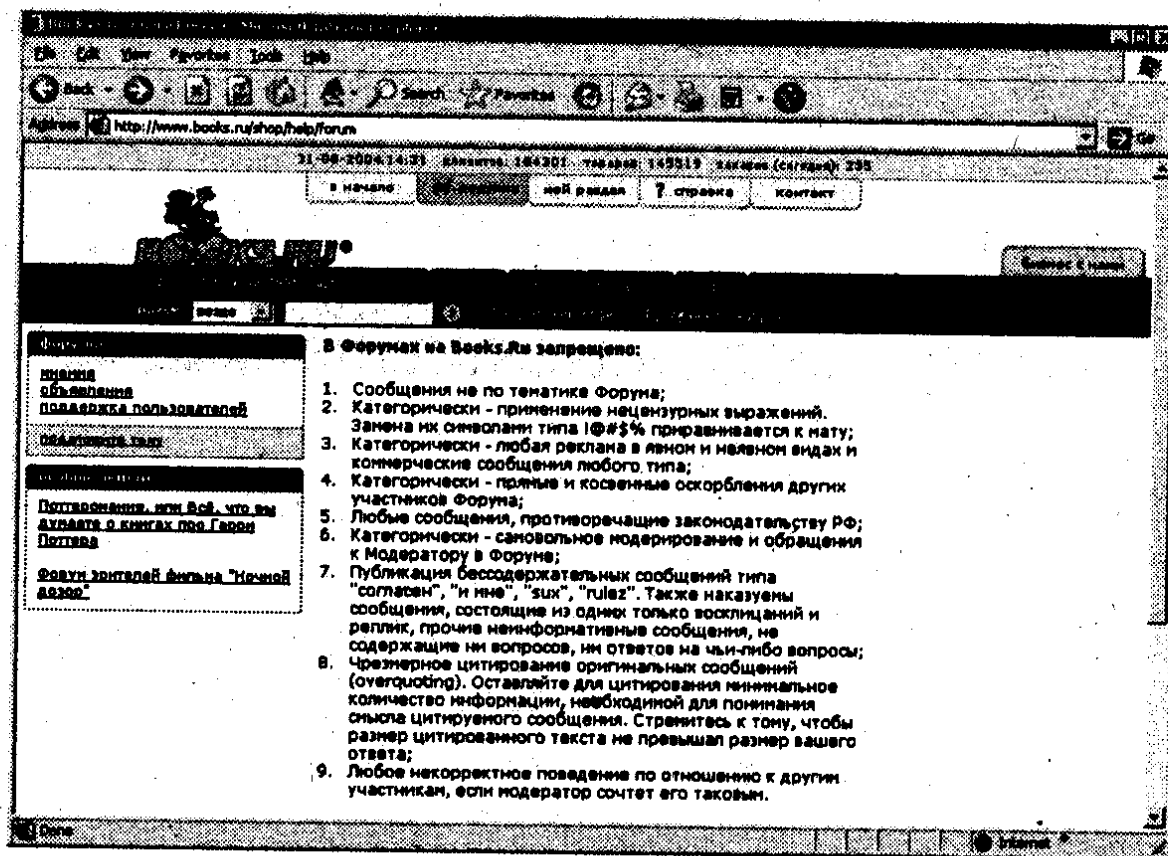


Рис. 11.7. Главная страница форума на books.ru

Немного подумав, можно улучшить юзабилити данной страницы (рис. 11.8).

В-третьих, после входа в сам форум я сразу вижу не темы по товарам, т.к. они представлены мелким шрифтом, а ничего не говорящие высказывания разных людей (рис. 11.9). Весьма неудобно.

Вот один из способов улучшить юзабилити данной страницы и устранить указанные недостатки (11.10).

#### 4. Порочность нестандартных решений

Нестандартные решения порочны. Несмотря на то что стандартов на оформление тех или иных элементов интерфейса пользователя официально не существует, они все-таки есть. Это как народный фольклор или как суслик из серии фильмов «ДМБ» — ты его не видишь, и я не вижу, а он есть.

На рис. 11.11 представлено нестандартное решение, придуманное фантазером-разработчиком сайта.

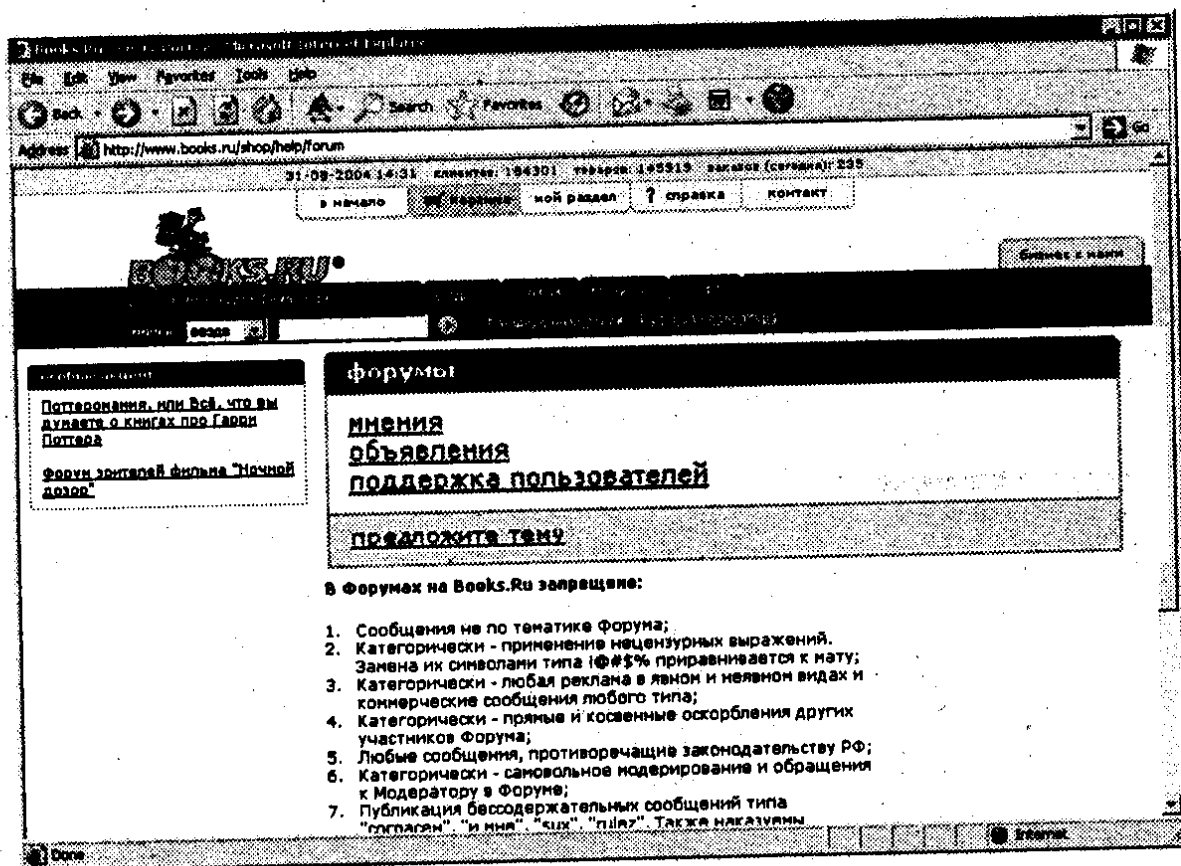


Рис. 11.8. Улучшенная версия главной страницы форумов

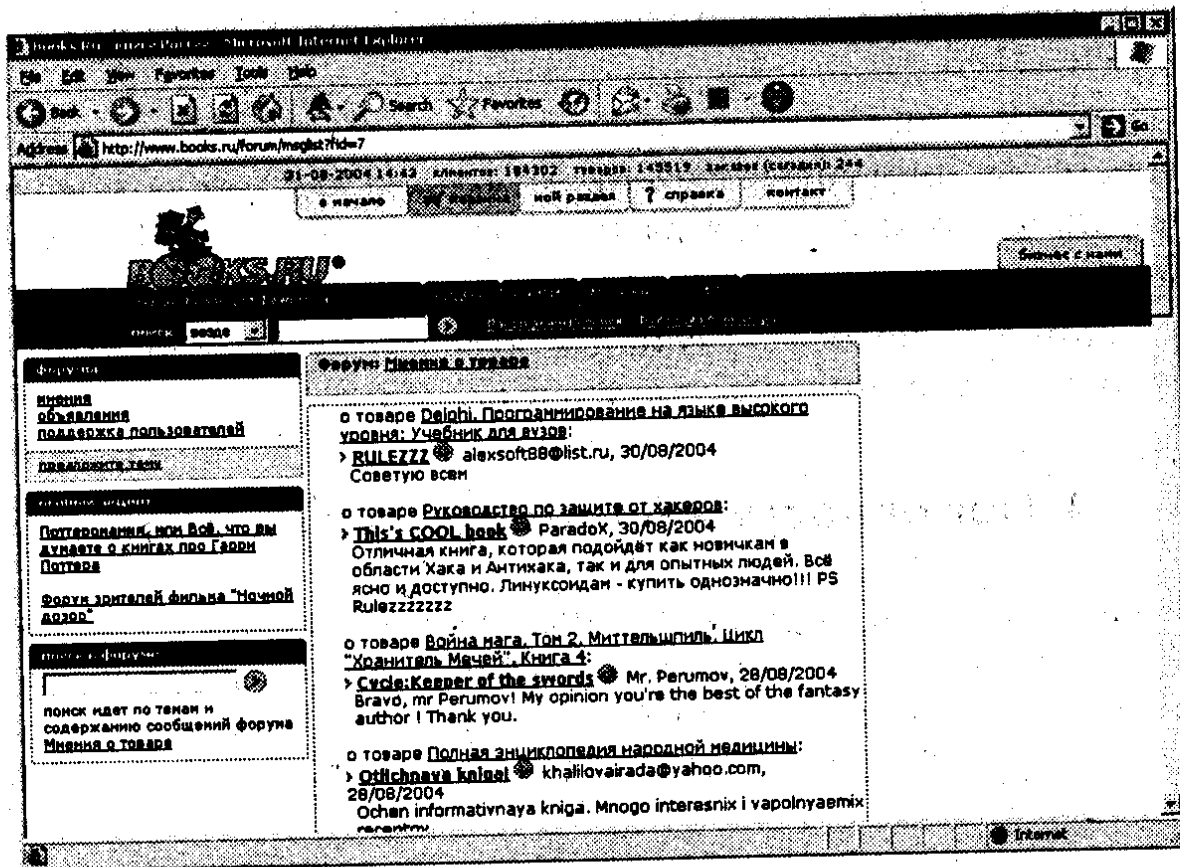


Рис. 11.9. Страница форума «Мнения о товаре»



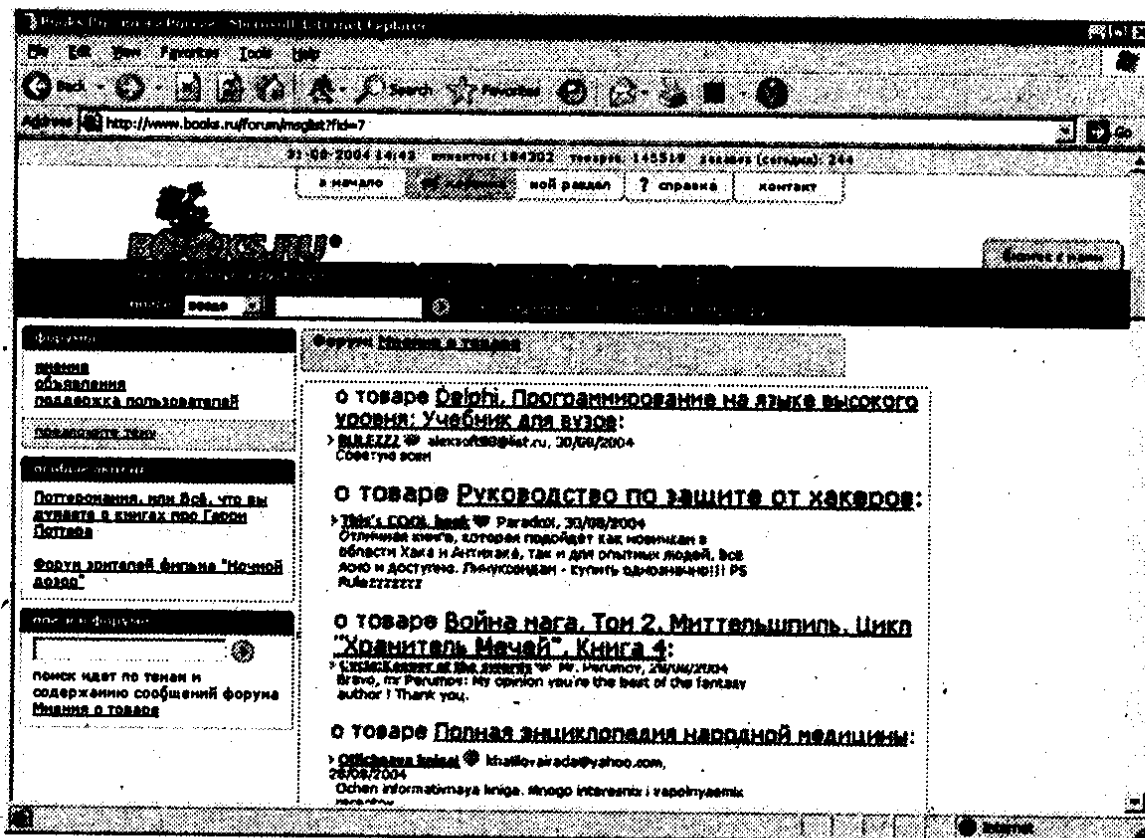


Рис. 11.10. Улучшенная версия страницы форума «Мнения о товаре»

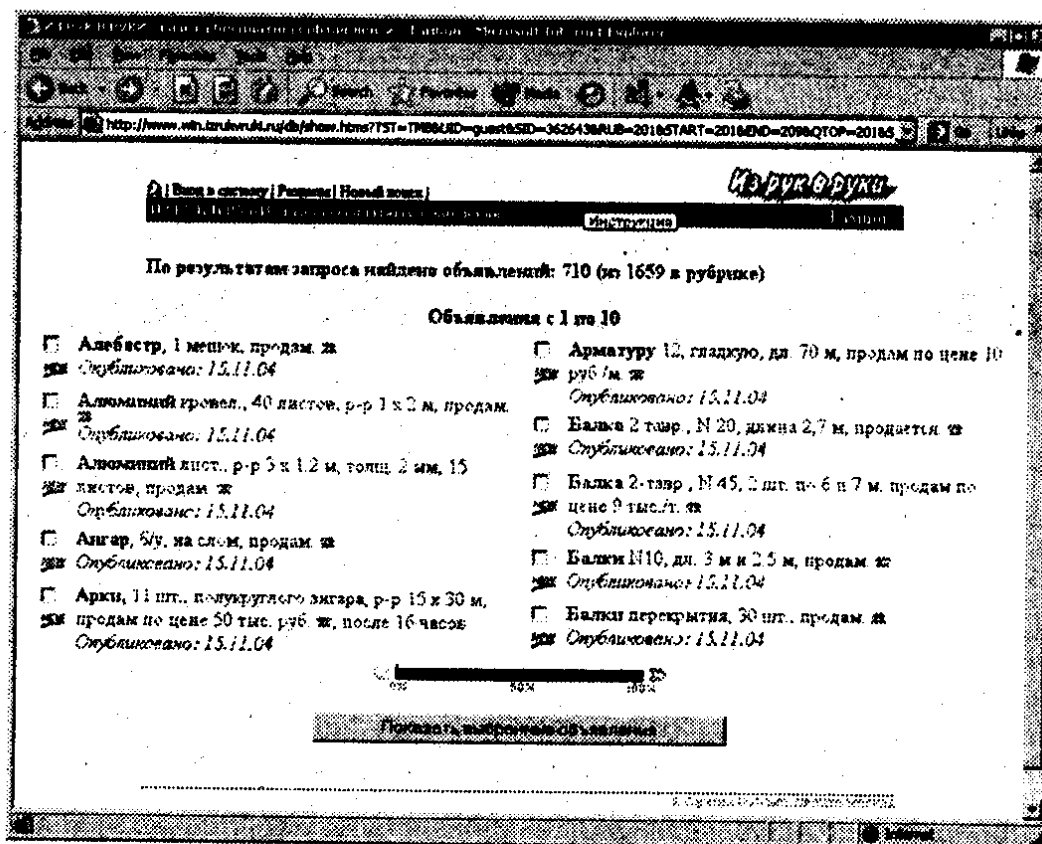


Рис. 11.11. Нестандартный элемент навигации на сайте газеты «Из рук в руки»



Во-первых, я до-о-о-о-о-лго додумывался, как же на странице перейти к следующей порции данных, тем более что в верхней части страницы явно было написано «57 объявлений из 181».

Поначалу я подумал, что элемент внизу страницы — какой-то особый баннер, и только когда не нашел ничего более подходящего, решил с ним поэкспериментировать.

Вот аналогичный элемент навигации по выборке объектов, взятый с [rambler.ru](http://rambler.ru) (рис. 11.12). Я ни на секунду не задумываюсь, как мне перейти к следующей порции.

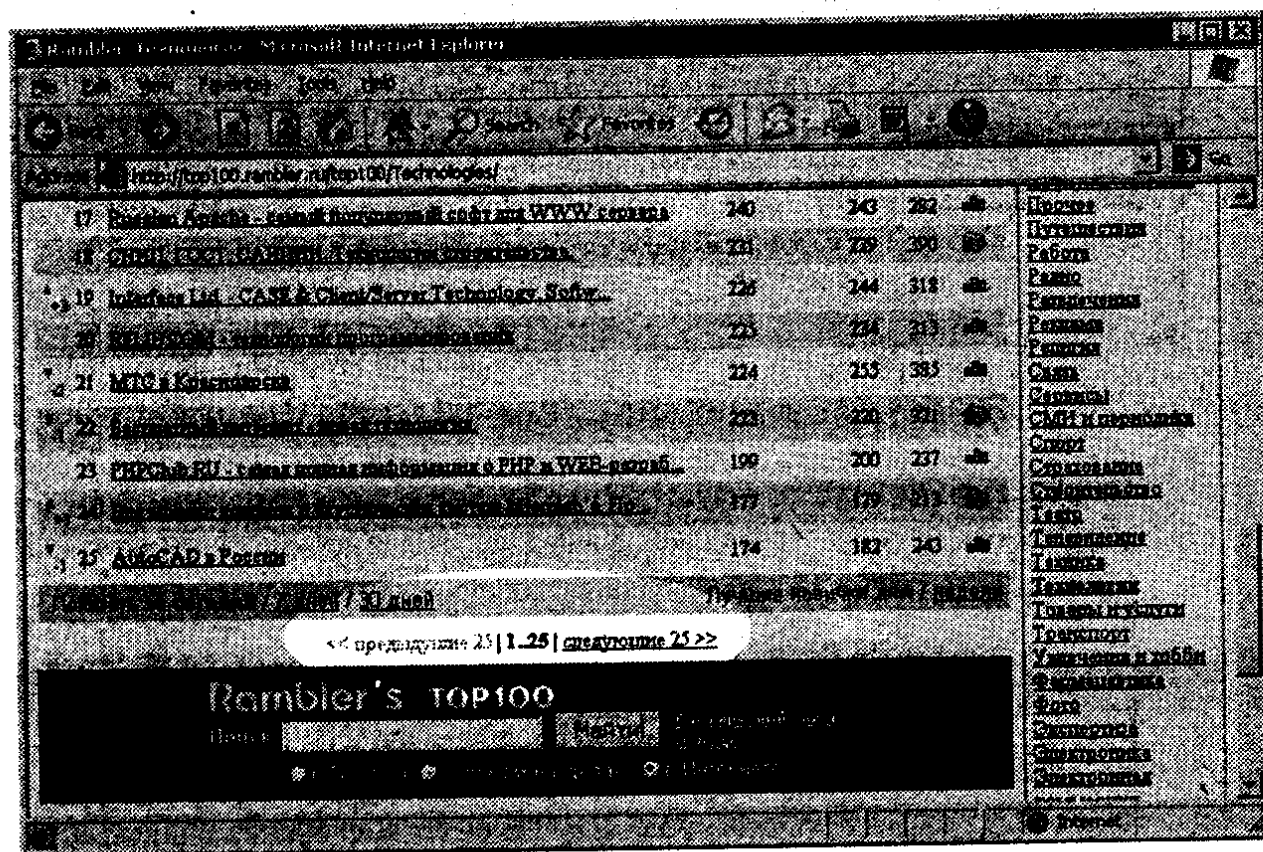


Рис. 11.12. «Стандартный» элемент навигации на сайте [rambler.ru](http://rambler.ru)

## 5. Трехминутный анализ

Как-то я случайно попал на сайт [price.ru](http://price.ru) и удивился его неудобству (рис. 11.13).

Вот недостатки, которые сразу бросаются в глаза:

- Заголовок сайта расположен только в заголовке окна и набран мелким шрифтом («Все цены и фирмы: компьютеры, hi-tech,

офисное оборудование»). Это неудачное решение: посетитель должен сразу понимать, куда он попал. Если он из целевой аудитории, то заголовок удержит его на сайте, в противном случае этот посетитель не будет зря «напрягать» сервер.

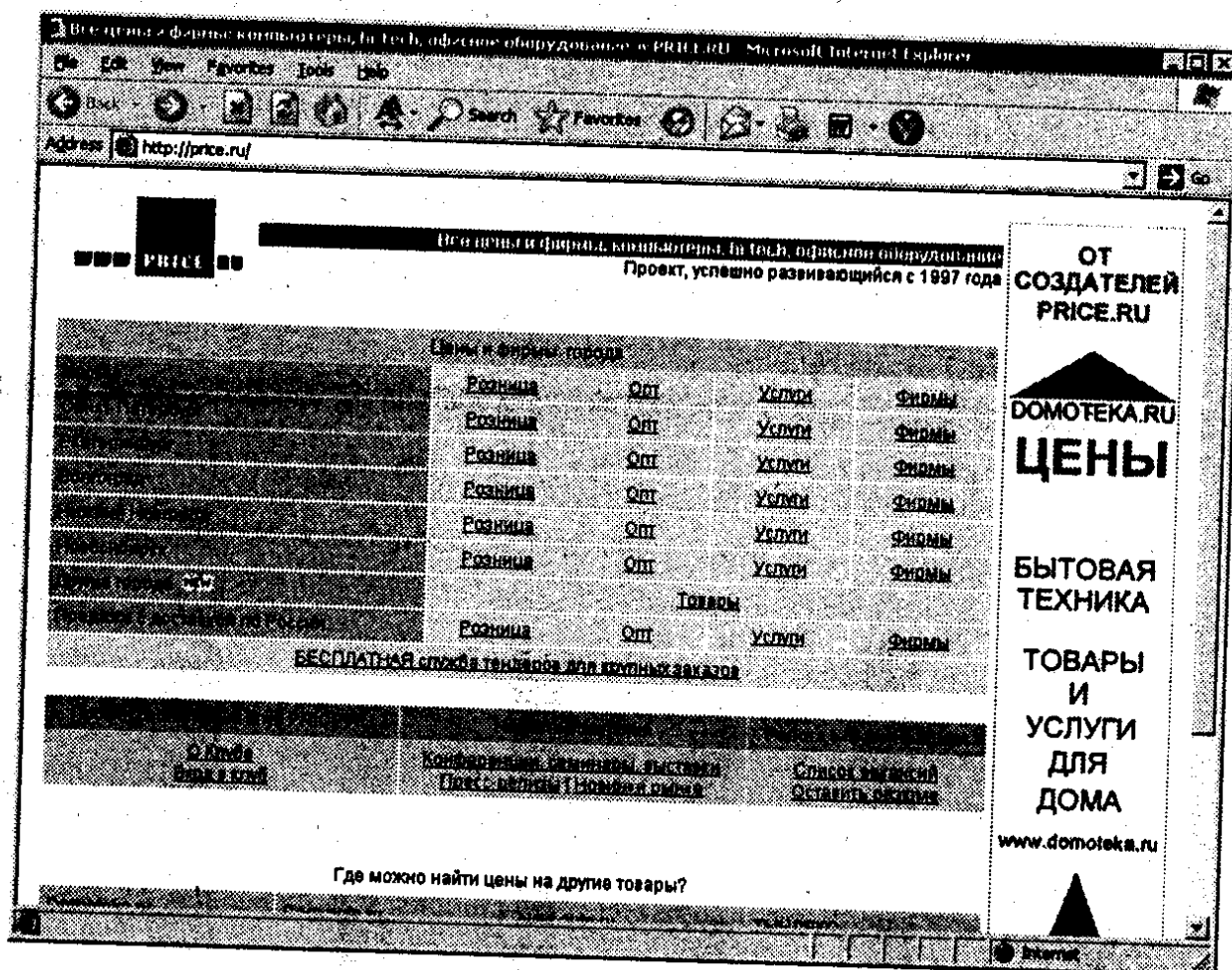


Рис. 11.13. «Хромающее» юзабилити сайта price.ru

- В первую очередь в глаза бросаются ссылки Опт, Розница, а они далеко не главные на странице. Кроме того, они обе ведут на страницу со вкладками, поэтому в некоторой степени нарушен принцип монотонности интерфейса.
- В основном пользователей интересуют либо товары, либо города, но выбирать сразу ни из того, ни из другого они не могут.
- Все разделы и ссылки обезличены. Неужели разработчики не хотят привлечь посетителя в клуб? Он просто не заметит этой ссылки, даже если был бы не против клуба.

Как видите, и трех минут бывает достаточно для беглой оценки юзабилити.

## 6. Поиск на Onliner.by

Когда мне понадобилось собрать информацию о мобильных телефонах для моего знакомого, знающие люди подсказали мне «рыбное» место — белорусский портал Onliner.by (рис. 11.14). Сам сайт мне очень понравился и по информативности, и по удобству, кроме одного большого НО. Я зашел по нужной ссылке и долго пытался понять, как же мне отыскать «раскладушку» с цветным экраном. Наконец, коллега указал чуть выше того места, где я искал нужную форму поиска.

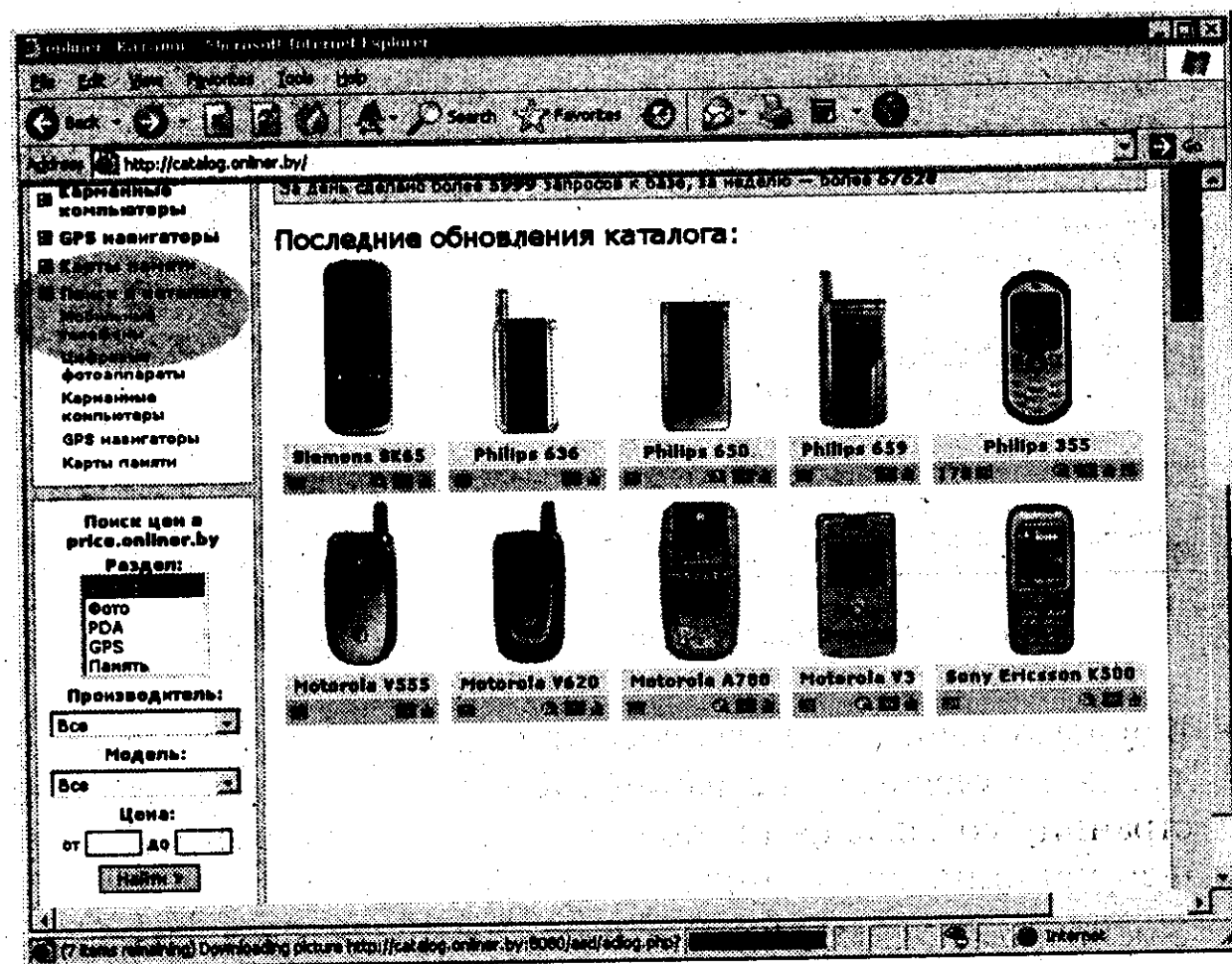


Рис. 11.14. Форма поиска с полями ввода, как вы видите, не позволяет искать так, как хочется. Нужно обнаружить ссылку Поиск в каталоге

Проблема появилась из-за того, что функция поиска была разделена на две разновидности. Одна — в виде формы с сокращенными параметрами, а другая, чуть выше, — в виде ссылки на

форму поиска (она-то и была нужна мне). Ссылка ничем не отличалась от других ссылок в вертикальном меню каталога.

Достаточно бросить беглый взгляд, чтобы понять, о чем я говорю и почему «нужный» поиск невидим.

## 7. «А где финики? Финики-то где?..»

Эта фраза из отечественного мультфильма про Братца Кролика, если не ошибаюсь.

В форме рассылок на mail.ru всегда присутствует мигающий баннер сайта strana.ru (рис. 11.15).

Смотрю, читаю. «Сенсация!!!». Нажимаю на баннер и попадаю всего лишь на главную страницу www.strana.ru. Это плохой прием, худший из возможных. Ссылки должны вести туда, куда обещают. Лично я, как пользователь, больше никогда не нажму на такой баннер — как говорится, обжегшись на молоке, дуют и на воду. Если я захочу почитать strana.ru, я наберу адрес в адресной строке, а я хочу почитать только то, что сказал террорист. Нет адекватных ссылок — нет посетителя.

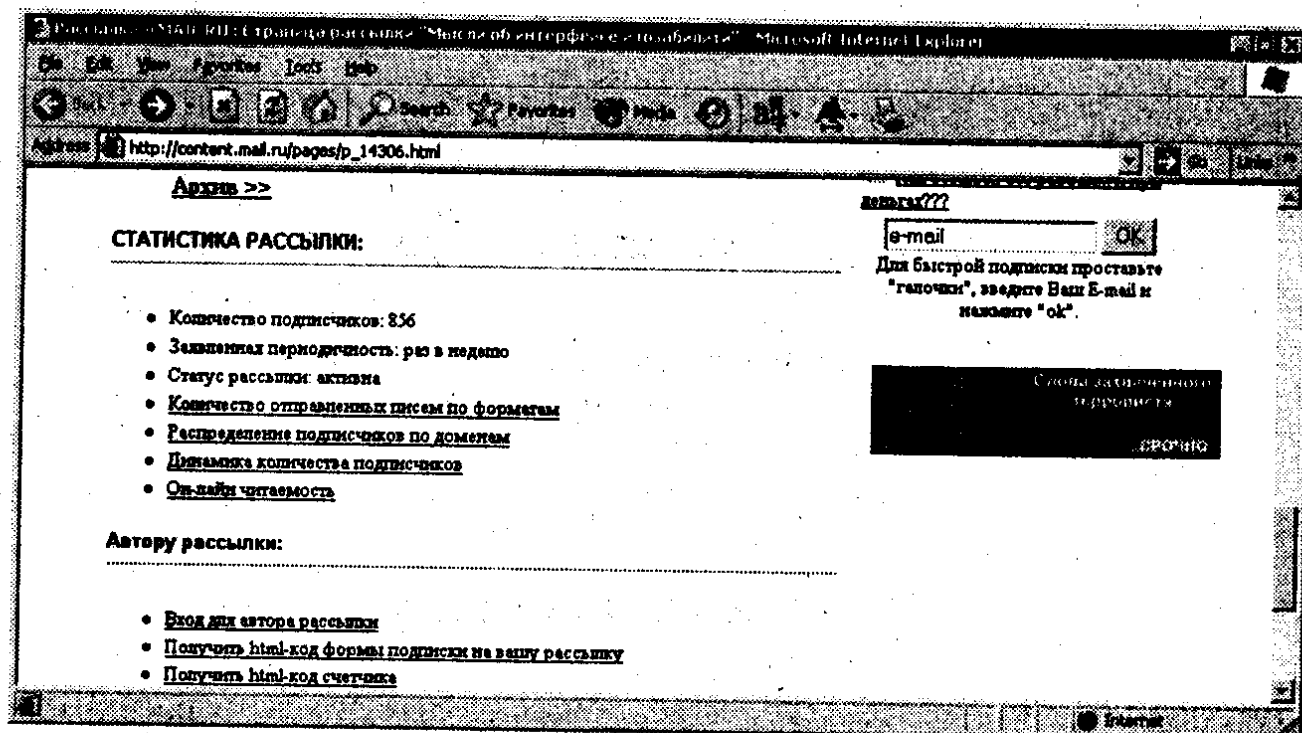


Рис. 11.15. Завлекающий новостной баннер

## 8. Так делать нельзя

Ярчайший пример того, как делать нельзя, на [pashen.kiev.ua](http://pashen.kiev.ua) (рис. 11.16). Форма поиска без кнопки! Вы где-нибудь еще такое видели? Мало того, что нужно догадываться о ее использовании, «оно» еще и не работает. Вот вам маленькая головоломка: как вы будете использовать такую чудную форму?

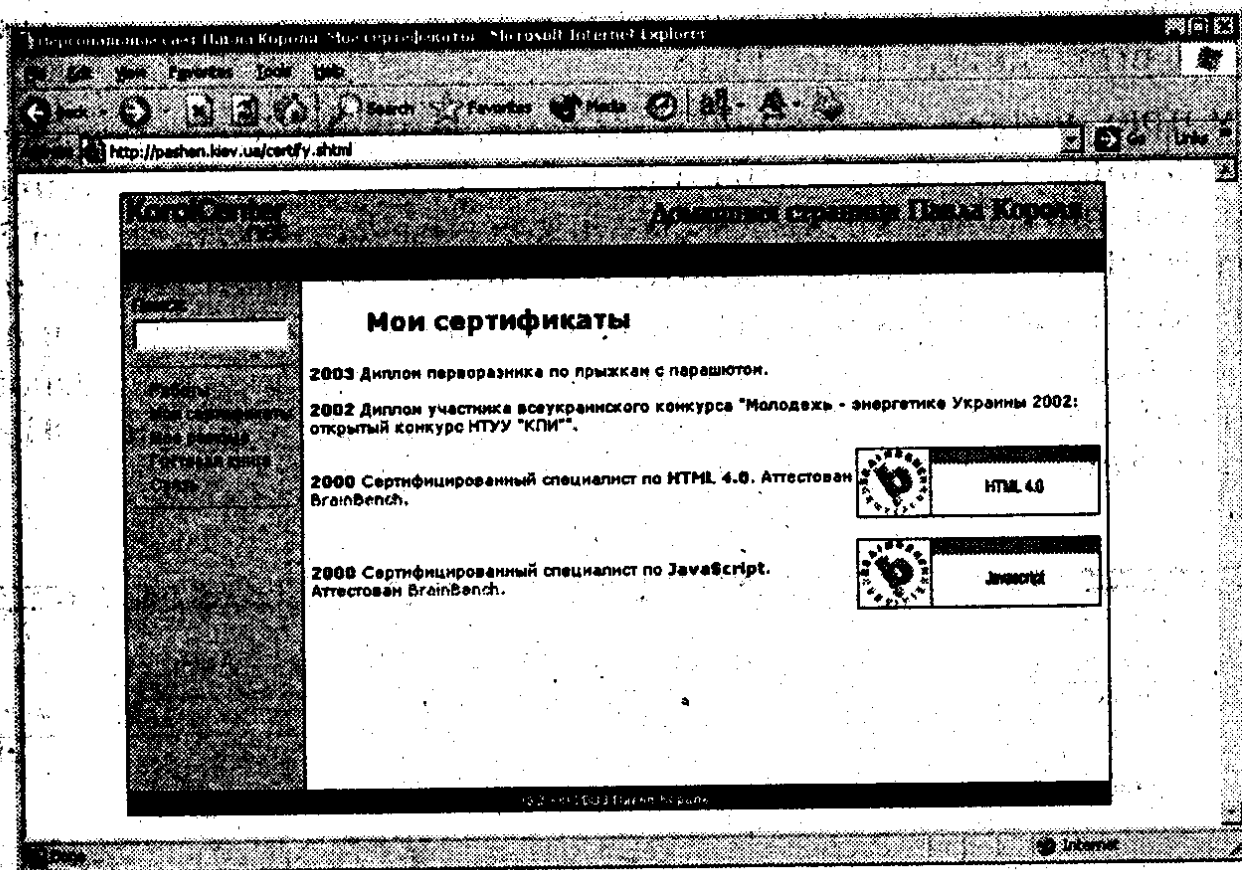


Рис. 11.16. Вот как бывает, когда человек верит в то, что все думают так, как он

## 9. Читаемость текста

Путешествуя по интернету, я встречал самые разные варианты оформления текстового материала. Недавно я попал на страницу с результатами поиска по запросу «Установка унитаза» по адресу <http://otdelka.h1.ru/> (я делаю дома ремонт).

Так получилось, что не догрузились таблицы стилей, и сайт предстал передо мной в совершенно ином виде. Предоставляю вам возможность оценить разницу в читаемости текста (рис. 11.17, 11.18).

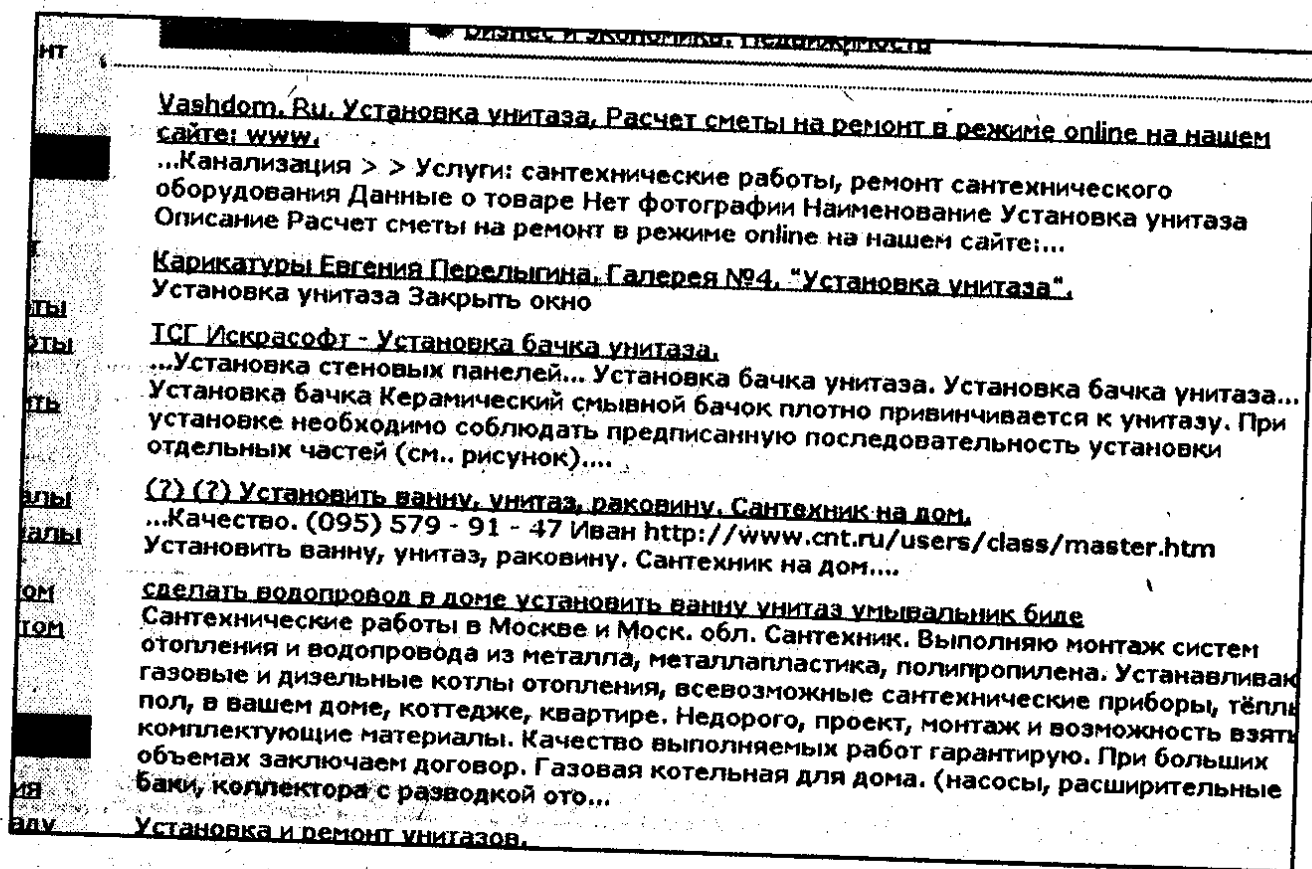


Рис. 11.17. Так выглядит сайт otdelka.h1.ru в корректном состоянии

## 10. «Попрыгунчики»

Раньше меня раздражали стандартные в MS Windows закладки, расположенные в два уровня (рис. 11.19). Вернее, раздражали не сами закладки, а то, что при переключении с нижнего ряда на верхний они начинают «прыгать» и оказываются уже в другом ряду. Наверняка многие из вас видели этих «попрыгунчиков».

Я задумался, с чем связаны такие неудобства. Ответ оказался прост. Если вы щелкнете на верхней правой закладке, то нижний ряд закладок просто исчезнет — раскрывающаяся закладка закроет его собой. Разве можно такое допустить? Вот и прыгают закладки снизу вверх, сверху вниз.

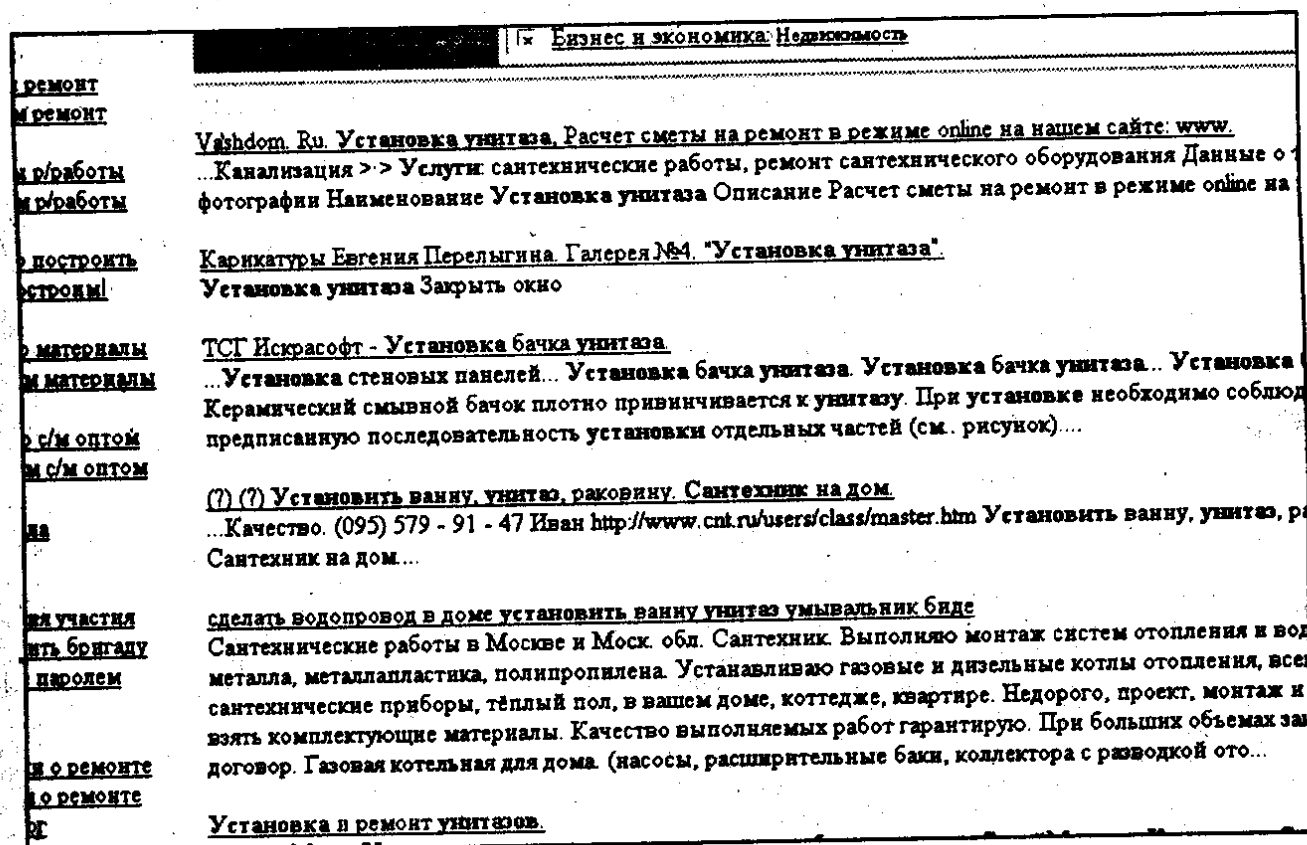


Рис. 11.18. Так выглядит сайт otdelka.h1.ru без таблиц стилей

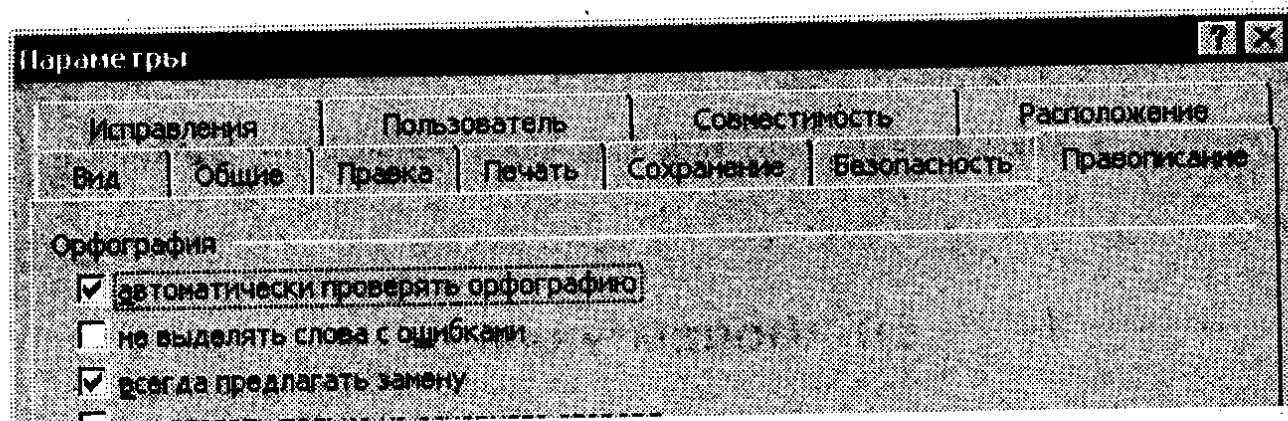


Рис. 11.19. Закладки в два ряда



Производственно-практическое издание

**Калиновский Андрей Иванович**

# **Юзабилити: как сделать сайт удобным**

Редактор *А.А. Пушкина*

Художник обложки *С.В. Ковалевский*

Компьютерная верстка *С.И. Лученок*

Корректор *Е.О. Кликунова*

Подписано в печать с готовых диапозитивов 02.02.2005.

Формат 60×84 1/16. Бумага типографская № 2. Гарнитура Ньютон.

Печать офсетная. Усл. печ. л. 13,02. Уч.-изд. л. 12,5.

Тираж 3010 экз. Заказ № 996.

Общество с ограниченной ответственностью «Новое знание».

ЛИ № 02330/0133439 от 30.04.2004. Минск, пр. Пушкина, д. 15, ком. 16.

Почтовый адрес: 220050, Минск, а/я 79.

Телефон/факс: (10-375-17) 211-50-38. E-mail: nk@wnk.biz

В Москве:

Москва, Колодезный пер., д. 2а.

Телефон (095) 234-58-53. E-mail: ru@wnk.biz

<http://wnk.biz>

Унитарное полиграфическое предприятие

«Витебская областная типография».

210015, Витебск, ул. Щербакова-Набережная, 4.

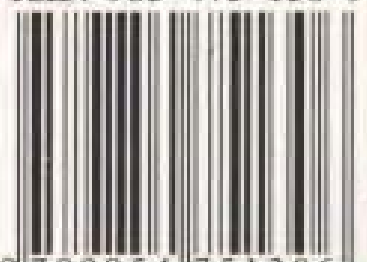




# Юзабилити:

как сделать сайт удобным

ISBN 985-475-130-9



9 789854 751306

<http://wnk.biz>