



Шаблоны проектирования веб-приложений



Паван Вора

Паван Вора

**Шаблоны проектирования
веб-приложений**

«ЭКСМО»

Вора П.

Шаблоны проектирования веб-приложений / П. Вора —
«Эксмо»,

Шаблоны проектирования веб-приложений, сходные по принципам с шаблонами проектирования веб-сайтов и программного обеспечения, предлагают множество эффективных решений. В книге приведены шаблоны проектирования веб-приложений исходя не только из проблем взаимодействия пользователей, но и рассматривая их эффективность и показывая, как они должны применяться. Из книги вы сможете узнать, как быстрее и качественнее проектировать отличные интерфейсы, как применять на практике и улучшать в дальнейшем многочисленные готовые решения. Вы сможете достичь небывалых высот в нелегком деле веб-программирования.

Содержание

Посвящается	5
Благодарность	6
Об авторе	7
Глава 1	8
Введение	8
Достоинства веб-приложений	11
Трудности в разработке интерфейсов для веб-приложений	13
Шаблоны проектирования	18
Документирование шаблонов	20
Структурирование шаблонов в этой книге	21
Применение шаблонов в этой книге	24
Глава 2	25
Введение	25
CLEAR BENEFITS (ЧЕТКИЕ ПРЕИМУЩЕСТВА)	26
SHORT FORMS (КОРОТКИЕ ФОРМЫ)	30
LOGICAL GROUPING (ЛОГИЧЕСКОЕ ГРУППИРОВАНИЕ)	34
LABEL ALIGNMENT (ВЫРАВНИЕ МЕТОК)	37
REQUIRED FIELD INDICATORS (ИНДИКАТОРЫ ОБЯЗАТЕЛЬНЫХ ПОЛЕЙ)	41
SMART DEFAULTS («УМНЫЕ» ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ)	44
FORGIVING FORMAT (ВЕЛИКОДУШНЫЙ ФОРМАТ)	47
KEYBOARD NAVIGATION (УПРАВЛЕНИЕ КЛАВИАТУРОЙ)	50
INPUT HINTS/PROMPTS (ПОДСКАЗКИ ПРИ ВВОДЕ/ ПРИГЛАШЕНИЕ К ВВОДУ)	53
Конец ознакомительного фрагмента.	56

Паван Вора
Шаблоны проектирования веб-приложений

Посвящается

Моей маленькой принцессе Суми

Благодарность

Выражаю искреннюю благодарность следующим людям:

- техническим редакторам – Венди Каслман, Дэвиду Дику, Каарен Хэнсон, Арни Лунду и Дейву Роджерсу – за потраченное время, хорошие советы и полезные комментарии. Их вдумчивые предложения в несколько раз улучшили эту книгу. Однако если в книге остались какие-либо ошибки или слабые места, это моя ответственность и, очевидно, результат того, что я невнимательно следовал их советам;

- Лену Бисли – за помощь в проведении исследования, а также помощь в редактировании многочисленных черновых вариантов каждой главы книги;

- команде издательства Elsevier: Мэри Джеймс, моему издателю, за проявленное терпение во время дискуссий со мной, а также за то, что помогала мне следовать намеченному пути и соблюдать сроки; Денис Пенроуз и Диане Серра, за то, что предоставили мне возможность написать книгу на ту тему, которая мне столь интересна; а также технологической группе, включая выпускающего редактора Джоди Аллен и корректора Деборе Прато, за их отличную корректуру и верстку книги;

- клиентам, коллегам и друзьям, которые наставляют, направляют, выслушивают и поддерживают меня;

- моей любимой жене Соне – за ее поддержку и терпение в течение долгого времени моей работы над книгой и за то, что не позволяла мне опускать руки в трудные периоды;

- и Суми – моей маленькой принцессе – за то, что понимала, что папе нужно время на написание книги, и за предложение написать эту книгу, чтобы у него было время играть с ней.

Паван Вора

Об авторе

Паван Вора – основатель и президент компании Alpha Cube, Inc., которая занимается консультированием в области проектирования взаимодействия программы с пользователем. Основные направления деятельности компании – проектирование, анализ и оценка пользовательских интерфейсов программного обеспечения и веб-приложений.

Он работает в этой сфере более 14 лет и спроектировал пользовательские интерфейсы для широкого диапазона приложений, работающих по схеме «предприятие—потребитель», «предприятие—предприятие», «потребитель для потребителя» и «предприятие—сотрудник».

Он издал и провел множество уроков и корпоративных практических тренингов по проектированию интернет-сайтов, веб-приложений и шаблонам проектирования в США и по всему миру.

Паван Вора получил степень магистра, а затем и степень доктора наук в сфере промышленного строительства (в Государственном университете Нью-Йорка в Буффало). Степень бакалавра в сфере промышленного строительства и машиностроения он получил в Техническом институте юбилея Королевы Виктории в Мумбае, Индия.

Глава 1

Введение

Введение

Все чаще компьютерные приложения разрабатываются на основе веб-технологий, и к ним можно получить доступ с помощью веббраузеров (например, Internet Explorer, Firefox, Safari и Opera). Обычно такие приложения называются *веб-приложениями*, или размещаемыми приложениями (hosted applications) – приложениями, в основе которых лежит модель программного обеспечения как услуги (SaaS)¹ или облачные вычисления (cloud computing)². Эти веб-приложения отличаются от более традиционных веб-сайтов в том смысле, что их предназначение – помочь пользователям в выполнении таких задач, как отправка электронных писем, бронирование авиабилетов, поиск домов, оплата счетов, перевод денег, покупка продуктов, рассылка приглашений и т. п. (рис. 1.1–1.4). Веб-сайты, напротив, ориентированы на контент и разрабатываются для поиска и ознакомления с достаточно статической информацией (рис. 1.5).

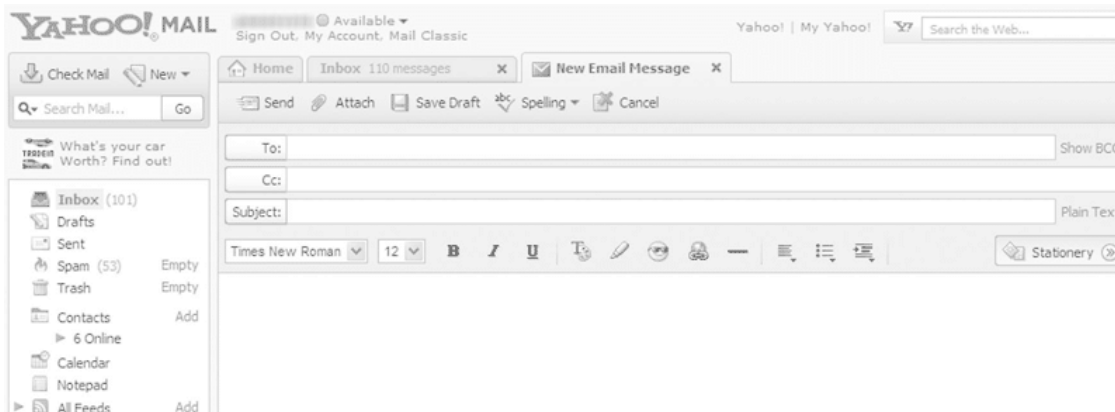


Рис. 1.1. Пользователи могут настроить свою электронную почту как в этом примере с Yahoo! Mail, и то же самое можно сделать и в таких клиентских приложениях, как Microsoft Outlook, Mozilla Thunderbird и Eudora

¹ SaaS – это модель продажи программного обеспечения, при котором поставщик разрабатывает веб-приложение, занимается его хостингом и управляет им (либо самостоятельно, либо через посредника), предоставляя клиентам возможность пользоваться им через Интернет. Клиенты не платят за право собственности на это программное обеспечение; они оформляют на него платную подписку.

² Веб-приложения считаются одним из видов «облачных вычислений», поскольку приложениями и файлами управляют в пределах «вычислительного облака», состоящего из тысяч компьютеров и серверов, соединенных друг с другом и доступных через Интернет.

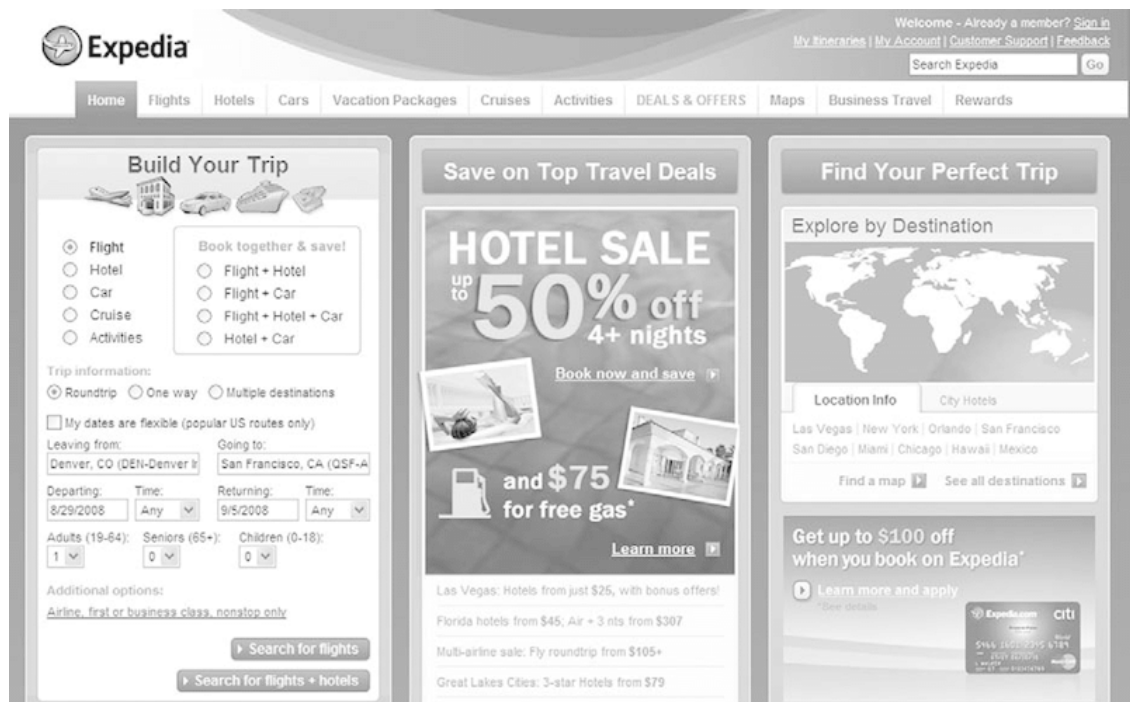


Рис. 1.2. Пользователи могут искать варианты туристических путевок и осуществлять бронирование билетов с помощью веб-приложений, таких как Expedia



Рис. 1.3. Пользователи могут найти продающийся дом, узнать его стоимость и увидеть, какие еще дома выставлены на продажу в этом районе. Все это можно сделать на таких сайтах, как Zillow.com

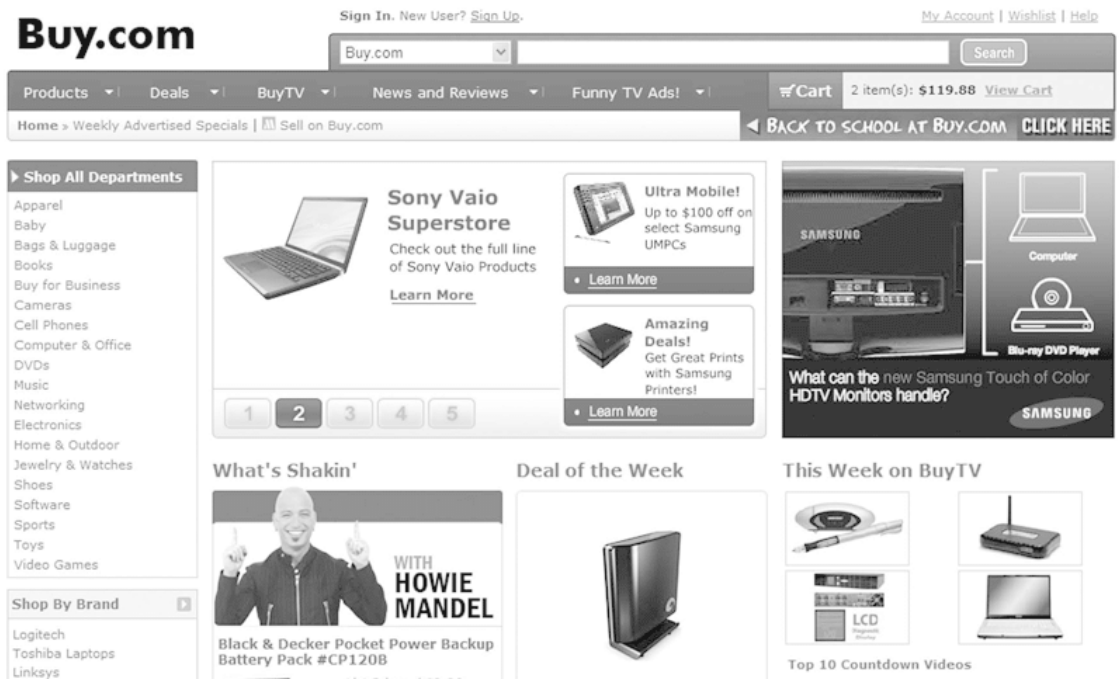


Рис. 1.4. Пользователи могут совершать покупки на таких сайтах, как Buy.com

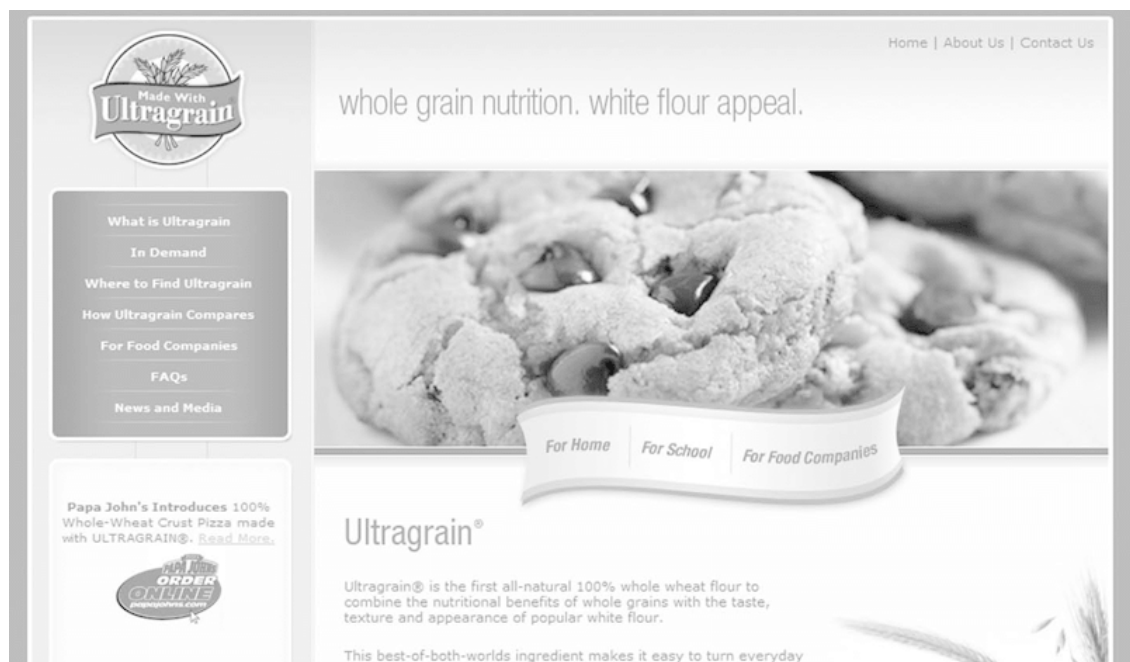


Рис. 1.5. Ознакомиться со статической информацией о компании Ultragrain и ее товарах можно на сайте этой компании (www.ultragram.com)

Достоинства веб-приложений

Популярность веб-приложений объясняется тем, что они обладают рядом достоинств. Эти достоинства описаны в данном разделе (Baxley, 2003; Turnbull, 2006).

Простота доступа

В большинстве случаев единственный вид программного обеспечения, который необходим для того, чтобы получить доступ и пользоваться веб-приложениями, – это браузер, такой как Internet Explorer, Firefox, Safari и Opera. Чтобы пользоваться веб-приложениями, пользователям не нужно скачивать и устанавливать дополнительное программное обеспечение, хотя в некоторых случаях все же приходится загружать вспомогательные приложения или плагины, такие как Adobe Flash, Java, Microsoft Silverlight и т. д., чтобы получить доступ ко всем частям веб-приложения.

Более того, так как и приложение, и информация хранятся на серверах поставщика, а не на пользовательских компьютерах, пользователи могут получить доступ к веб-приложениям из практически любого места, где есть подключенный к Интернету компьютер, на котором установлен браузер. Благодаря удаленному хранению данных, пользователи могут делиться информацией и сотрудничать друг с другом; например, они могут совместно пользоваться закладками приложений, такими как Delicious (www.delicious.com) и Furl (www.furl.net), и удаленно совместно работать над одними и теми же документами, применяя повышающие эффективность работы приложения, такие как Google Docs and Spreadsheets (docs.google.com) и Zoho (www.zoho.com).

Простота применения

Веб-приложения также популярны среди бизнесменов и разработчиков программного обеспечения, поскольку их можно дорабатывать, обновлять и отлаживать удаленно, при этом пользователям не приходится их устанавливать (или переустанавливать). С этим связано очередное достоинство веб-приложений – их поведение не зависит от операционной системы, установленной на компьютере пользователя. Один и тот же вариант приложения может использоваться практически любым пользователем, вместо того, чтобы создавать отдельные версии приложения для Windows, Macintosh OS X, GNU/Linux и других операционных систем.

«Обученный» базовый контингент пользователей

Развитие и распространение Интернета (от 16 миллионов пользователей в декабре 1995 года до почти 1,5 миллиардов пользователей в июне 2008 года согласно данным сайта Internet World Stats www.internetworldstats.com) способствовало тому, что взаимодействие в сети Интернет стало привычным для огромного количества пользователей. Большинство пользователей Интернета сейчас знакомы с терминологией веб-браузеров, например, *домашняя страница, назад, вперед, избранное, гиперссылки, кнопки подтверждения* и т. д. Принимая во внимание все это и учитывая тот факт, что для пользования веб-приложениями не нужны замысловатые установки, осталось очень мало преград для их использования (или, по крайней мере, для того, чтобы хотя бы с ними ознакомиться). Более того, многими популярными веб-приложениями теперь можно пользоваться бесплатно или бесплатно с ними ознакомиться.

Надежность и высокий уровень развития связности сети и веб-технологий

Раньше существенным препятствием для развития веб-приложений являлась ненадежность связности сети и очень нестабильная поддержка веб-стандартов – а именно, HTML, CSS и JavaScript – в веббраузерах. Теперь ситуация изменилась. Поддержка веб-стандартов становится все существеннее, а их несовместимость с браузерами, доводившая раньше разработчиков до отчаяния, уменьшается. Кроме того, связность сети и широкополосный доступ становятся все надежнее, все распространеннее и дешевле в использовании. Согласно результатам исследования Website Optimization, процент домов в США, имеющих широкополосный доступ в Интернет, в марте 2008 года вырос до 57 %, а для активных пользователей сети Интернет он составил 90 % (www.websiteoptimization.com/bw/0807/). Это послужило надежной основой для появления визуальных средств разработки и инфраструктур, способствующих развитию веб-приложений.

Трудности в разработке интерфейсов для веб-приложений

Несмотря на все эти достоинства и возрастающую пользу, разработка интерфейсов для веб-приложений остается проблематичной. Сложности при создании удобных в применении способов взаимодействия в основном обусловлены тем, что лежащая в основе веб-архитектура «слабо связана», а также с тем, что веб-браузеры поддерживают ограниченный набор интерактивных средств управления, и с нехваткой руководств по разработке таких аспектов, как например, каким образом осуществлять взаимодействие пользователей с системой.

«Слабо связанная» веб-архитектура

Важная проблема, с которой сталкиваются разработчики веб-приложений – это «слабо связанная» природа Интернета. Схема веб-взаимодействия очень проста: пользователь запрашивает веб-страницы с помощью веб-браузера, и серверы в ответ отправляют браузеру запрашиваемые страницы или сообщают пользователям, что при загрузке этой страницы возникли проблемы. Однако когда веб-сервер выполняет просьбу пользователя и отправляет браузеру веб-страницу, осуществляется соединение между веб-сервером и веб-браузером. Когда пользователь производит следующий запрос, снова устанавливается соединение с сервером до тех пор, пока новая страница не загрузится в браузере пользователя.

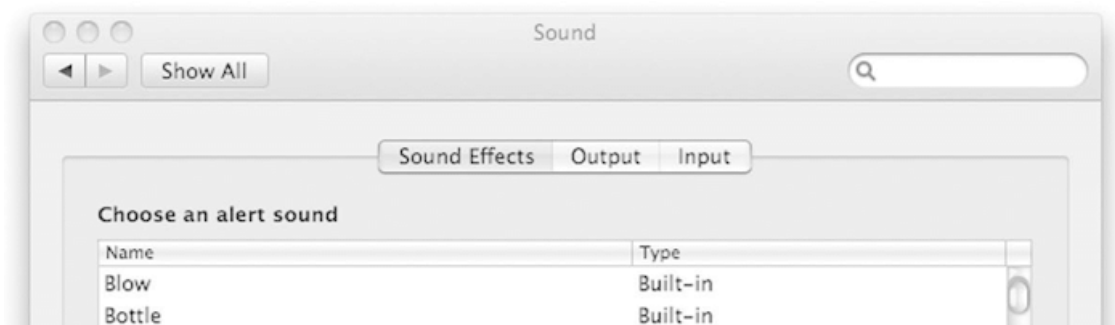
Загрузка каждой новой страницы или обновление страницы сопровождается приемлемыми задержками, связанными с необходимостью установить соединение, отправкой ответа на запрос, получением страницы и перезагрузкой страницы в браузере. Все это может показаться пользователям веб-приложений довольно скачкообразным и нестабильным. Например, пользователю, который просматривает данные, структурированные в виде иерархического дерева, возможно, придется ждать после каждого клика, пока узел отсчета будет дополнен или удален на перезагружаемой странице и можно будет увидеть изменения. Хотя эта проблема в большей степени связана с применением технологий создания сценариев, такими как JavaScript и богатые веб-приложения (Rich Internet Applications, см. главу 8), с ней часто сталкиваются пользователи при работе с большинством веб-приложений.

Ограниченный набор средств управления, или виджетов, для поддержки архитектуры приложения

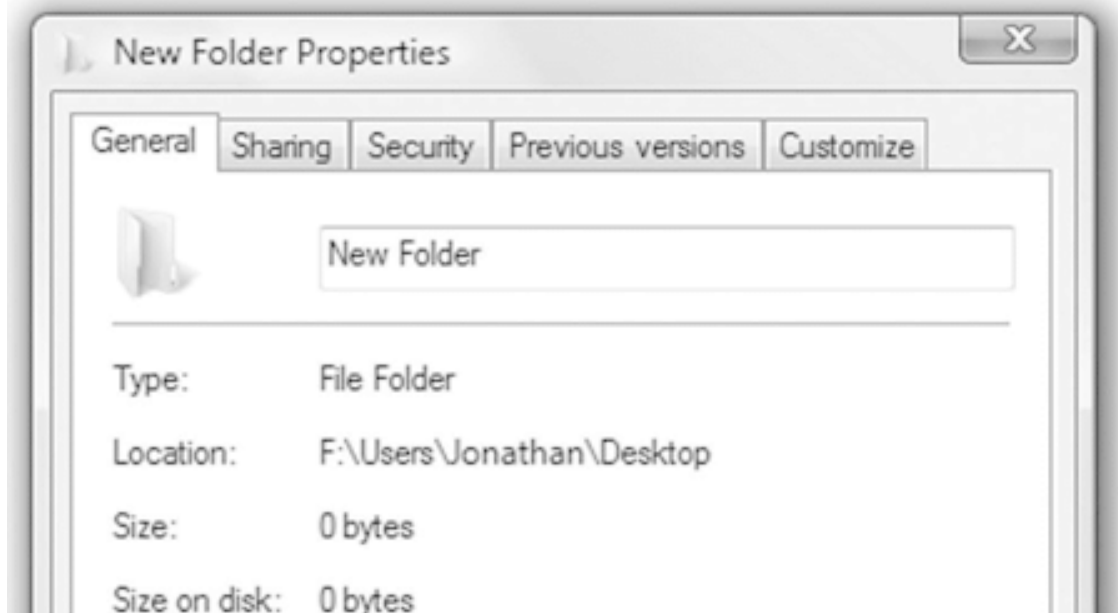
В текущей версии стандарта HTML (версия 4.01) поддержка средств управления ограничивается текстовыми окнами, зависимыми клавишами, флажками, раскрывающимися списками и командными или управляющими кнопками. Эта версия не поддерживает сложные взаимодействия, типичные для клиентских приложений, таких как регуляторы прокрутки, календари, экспертные системы, закладки, панели инструментов, перетаскивание, плавающие панели, диалоговые окна, контекстно-зависимые меню и т. д., которые встречаются даже в базовых клиентских приложениях. Хотя подобные элементы управления можно разработать с помощью сценариев JavaScript и каскадных таблиц стилей (Cascading Style Sheets, CSS), недостаток изначальной поддержки привел к появлению разнообразных способов их реализации, обладающих неполным функционалом.

Несогласованные способы взаимодействия

Как базовая технологическая архитектура Интернета, так и ограниченный набор доступных элементов управления затрудняют создание таких способов взаимодействия пользователя с веб-приложением, которые были бы на уровне со способами взаимодействия в клиентских приложениях. Кроме этого поскольку большинство веб-приложений разработано так, чтобы с ними можно было работать через браузер, способы взаимодействия и оформление нельзя спроектировать так, чтобы они подходили ко всем операционным системам; например, закладки в интерфейсе Macintosh OS X Aqua внешне выглядят совершенно иначе, нежели закладки в интерфейсе Windows Vista (рис. 1.6).



(a)

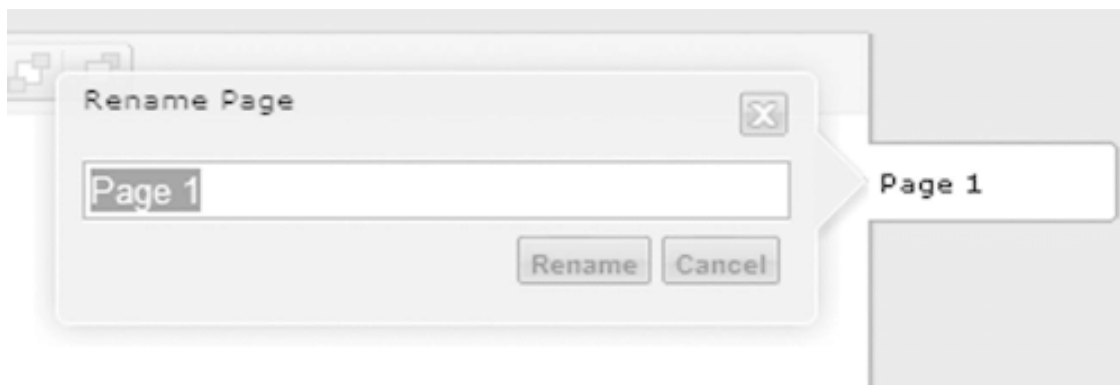


(б)

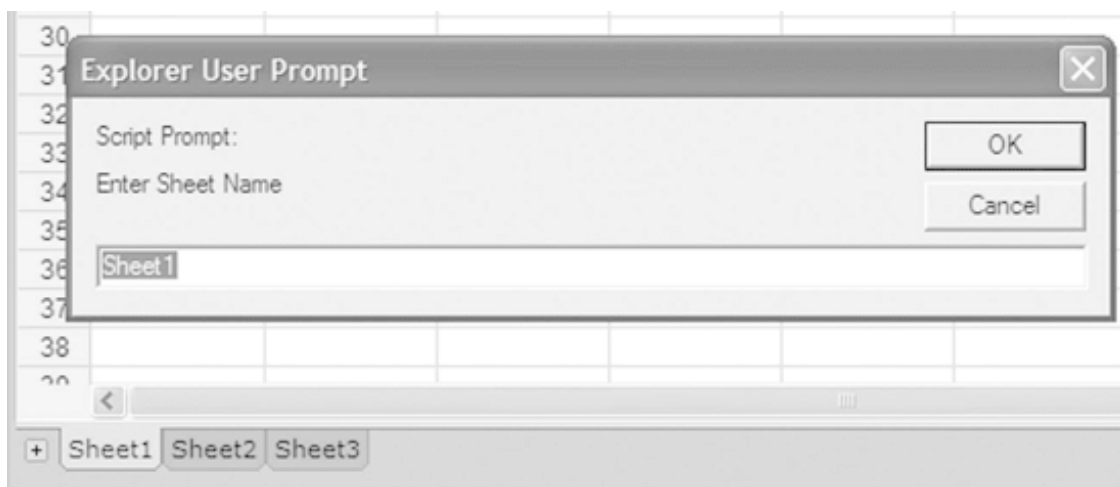
Рис. 1.6. Закладки в интерфейсе Macintosh OS X Aqua (a) и в интерфейсе

Хотя сравнительно свободная среда проектирования Интернета предоставляет разработчикам простор для творчества, она также приводит к многообразию и неоднородности пользовательских интерфейсов и способов, что во многих случаях вызывает определенные трудности для пользователей. Сложности связаны с тем, что пользователи сталкиваются с различными вариантами интерфейсов и способов взаимодействия, каждый из которых обладает собствен-

ной терминологией для названия объектов, действий и изображений, объединенных в одном и том же приложении (см. Тидвелл, 2008). На рисунке 1.7 показан пример изменения названия закладки в Zoho Notes (веб-приложение для записей, похожее на Microsoft OneNote) и Zoho Sheet (веб-приложение для работы с таблицами, похожее на Microsoft Excel). Чтобы изменить название закладки в Zoho Notes, пользователи должны дважды щелкнуть по названию закладки, после чего появится диалоговое окно Rename (Переименовать). Чтобы изменить название закладки в Zoho Sheet, пользователи должны щелкнуть правой кнопкой мыши по закладке и выбрать пункт Rename (Переименовать), который откроет всплывающее окно, позволяющее пользователю сменить имя; при этом с помощью двойного щелчка мыши можно выделить текст и заменить его своим.



(a)



(б)

Рис. 1.7. Два различных способа взаимодействия пользователя с системой для смены названия закладки в Zoho Notes (a) и в Zoho Sheet (б)

Примечание

HTML5 будет поддерживать дополнительные элементы формы, которые на данный момент входят в спецификацию Web Forms 2.0, разработанную консорциумом World Wide Web Consortium (W3C) (www.w3.org/TR/web-forms-2/). Эта новая версия предоставит дополнительные элементы управления (например, элемент <datalist> для создания комбинированных окон и элемент <output> для того, чтобы показывать значения, производные от других элементов управления), а также в ней появятся расширения для

существующих элементов управления (например, `<input type="date"/>`, `<input type="email"/>` и др.), что немного облегчит разработку вебприложений. Браузер Орега (версия 9 и выше) сегодня поддерживает Web Forms 2.0 и предоставляет хорошую базу для разработки интерактивных прототипов.

Для разрешения этих трудностей и сопутствующих им проблем, связанных с удобством и простотой использования приложений, многие корпорации создают руководства разработки пользовательского интерфейса и руководства по стилю оформления для управления графическим пользовательским интерфейсом веб-приложений.

Однако применить эти принципы для разработки удобных в использовании веб-приложений не так уж и просто. Принципы разработки обладают ограниченной эффективностью, поскольку они часто предлагают высокоуровневые принципы (например, сделать видимым системный статус; использовать распознавание вместо повторного вызова; предотвращение ошибок) или предоставляют очень узкоспециализированные инструкции (например, выравнивать заголовки таблиц относительно выравнивания данных в таблице). С другой стороны, руководства по стилю оформления фокусируются на создании имиджа и аспектах визуального проектирования и обычно предназначены для определенной платформы (например, Macintosh OS X Aqua и Windows Vista) или для приложений, разработанных определенной корпорацией (например, руководство по графическому пользовательскому интерфейсу браузера от компании Oracle: «Oracle Browser Look and Feel [BLAF] Guidelines», 2004), и вовсе необязательно предоставляют подробные инструкции для разработки удобных средств взаимодействия.

Руководства по разработке и стилю оформления обычно довольно объемные и сложные в использовании, поскольку для реализации одного способа взаимодействия обычно необходимо, чтобы разработчики пользовательского интерфейса ознакомились с несколькими пунктами документа. Например, руководство по проектированию пользовательских интерфейсов от компании Apple: «Human Interface Guidelines» – документ объемом почти в 400 страниц, и чтобы создать диалоговое окно с элементами управления, разработчику пришлось бы ознакомиться с несколькими пунктами части номер 3, называющейся: «Part III: The Aqua Interface» (рис. 1.8), которая содержит примерно 250 страниц, посвященных этому вопросу.

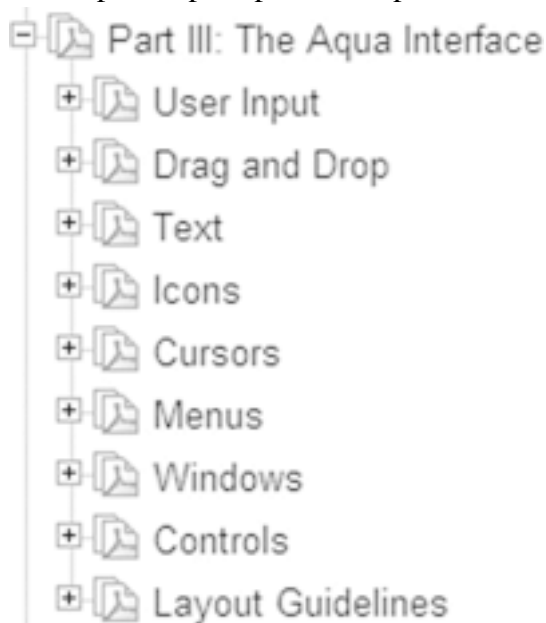


Рис. 1.8. Раздел «Часть III: Аквавзаимодействие» в «Руководстве по проектированию пользовательских интерфейсов» от компании Apple составляет примерно 250 из 400 страниц документа

В конце концов руководства по разработке и стилю оформления могут быть слишком сложными, чтобы быть полезными для тех, кто не знаком с системой взаимодействия человека с компьютером. Это делает их менее эффективным средством коммуникации между членами команды разработчиков веб-приложения, к которым относятся покупатели, эксперты по отдельным вопросам, клиенты, аналитики, разработчики программного обеспечения, руководители проектов и маркетологи, которые вряд ли хорошо подкованы в вопросах взаимодействия человека с компьютером и удобства и простоты использования.

Применение шаблонов проектирования помогает решить многие вопросы и может служить хорошим дополнением к руководствам по разработке и оформлению при создании еще лучших и единообразных по стилю интерфейсов, а так же при усовершенствовании простоты и удобства использования веб-приложений.

Шаблоны проектирования

Упоминание о шаблонах впервые появилось применительно к архитектуре в работах Кристофера Александра и его коллег: «Язык шаблонов» (Alexander et al., 1977) и «Извечный путь строительства» (Alexander, 1979). Они объясняют суть шаблонов следующим образом:

Каждый шаблон описывает некую повторяющуюся в нашей среде проблему, а затем таким образом описывает ключ к ее решению, что это решение можно применять многократно, ни разу не придя к одному и тому же результату.

Alexander et al., 1977, стр. x

Таким образом, шаблоны фокусируются на проблемах контекста применения и служат ориентиром для разработчиков в том, когда, как и для чего применять данное решение. Шаблоны имеют практическое применение и отвечают требованиям «хорошего» проектирования, при этом они воплощают высокоуровневые принципы и стратегии.

В последнее время шаблоны все чаще стали применяться разработчиками пользовательских интерфейсов и программного обеспечения³, которых привлекают следующие преимущества.

- Проверенные решения и руководства по их применению. Шаблоны указывают на реальные решения, а не на некие абстрактные принципы или директивы. Помимо этого, благодаря указанию контекста, выявлению проблемы и ее логическому обоснованию, шаблоны объясняют и то, как можно решить проблему, и то, почему это решение подходит для данного конкретного контекста. Однако поскольку это универсальное «ключевое» решение, оно может применяться по-разному в различных реализациях, при этом реализация не будет казаться «типовой» или недостаточно творческой.

- Усовершенствованный процесс проектирования. Определение шаблонов проекта и их каталогизация может помочь разработчикам пользовательского интерфейса увеличить производительность, сократив время, которое тратится на «изобретение велосипеда». Более того, если элементы пользовательского интерфейса используются в качестве шаблонов и представлены в виде библиотеки шаблонов проектов (см. главу 13), проекты можно разрабатывать, тестировать и дорабатывать довольно быстро, а также можно сократить цикл выпуска.

- Возможность многократного применения и согласованные интерфейсы. Формирование библиотеки допускающих многократное применение элементов пользовательского интерфейса также может способствовать разработке согласованных интерфейсов приложений. Особенно это удобно для больших корпораций, в которых много рассредоточенных проектных групп, где решения для одной и той же проблемы могут применяться различными проектными группами, что приводит к несогласованности интерфейсов, разработанных одной и той же компанией. Благодаря каталогизации и обмену информацией о шаблонах проекта, группы могут повысить согласованность, предсказуемость, простоту и удобство использования своих проектов (Leacock et al., 2005) и могут сформировать корпоративную память об опыте проектирования (Borchers, 2001).

³ После выхода работы Кристофера Александра в сфере архитектуры, понятие шаблонов было популяризировано и стало применяться в области программного обеспечения. Этому способствовала также работа Эрика Гаммы, Ричарда Хелма, Ральфа Джонсона и Джона Влиссидеса (часто эту группу называют GoF, от англ. Gang of Four – «банда четырех»): «Приемы объектно-ориентированного проектирования. Паттерны проектирования» (Питер, 2007). Впоследствии шаблоны также стали популярны в области взаимодействия компьютера и человека, этому способствовали работы Тидвелла (Tidwell, www.designinginterfaces.com), Уэли (Welie, www.welie.com), Борчерса (Borchers, 2001), Грэхэма (Graham, 2003) и Ван Дюйна (Van Duijn et al., 2002, 2006).

- **Общий, совместно используемый язык.** Шаблоны помогают поддерживать и совершенствовать обмен информацией между членами команды с различной специализацией благодаря развитию общего языка или терминологии при объяснении и обсуждении проектных решений (Borchers, 2001; Erickson, 2000). Это очень важно, поскольку проектировщики пользовательского интерфейса часто работают в междисциплинарных группах, в состав которых входят также разработчики, специалисты в прикладных областях и пользователи или представители пользователей, а в этих группах обычно отсутствует общая терминология для обмена идеями и мнениями по поводу проекта.

- **Эффективное средство обучения и справочное руководство.** Для опытных проектировщиков шаблоны также могут быть эффективным средством предоставления инструкций для тех, кто не получил формального образования в области проектирования (Chen, 2003). При наличии наглядных и текстовых описаний неопытным проектировщикам интерфейсов легче увидеть примеры успешного применения шаблонов.

- **Удобные в применении веб-приложения.** В конечном итоге, поскольку шаблоны основаны на успешном опыте применения, их использование может сделать веб-приложения удобными в применении, потому как способы взаимодействия, лежащие в основе шаблонов, знакомы пользователям.

Документирование шаблонов

Важно, чтобы шаблоны были документально оформлены, чтобы из этой документации можно было понять, что они собой представляют, почему они работают и как они должны применяться для решения проблемы проектирования, чтобы получить вышеупомянутые преимущества. Однако интересен тот факт, что не существует какого-либо «шаблона» для документирования шаблонов (см. главу 13). Авторы шаблонов применяют различные подходы, начиная с тех, кто основывается на более описательном повествовании в духе Кристофера Александра (Borchers, 2001; Graham, 2003; van Duynе et al., 2002, 2006), заканчивая теми, кто следует более структурированному, хотя и минималистическому, подходу (Laasko, 2003); Тидвелл, 2008; www.welie.com)⁴.

В этой книге представлен довольно краткий обзор шаблонов. В частности, в описание каждого шаблона входит:

- *Название шаблона.* Короткий заголовок, описывающий проектное решение. По примеру других наборов шаблонов названия шаблонов написаны ЗАГЛАВНЫМИ БУКВАМИ, чтобы они выделялись среди остального текста. В тексте приведены названия шаблонов на английском языке, а в конце книги вы найдете список всех шаблонов на английском языке и смысловой перевод в скобках (русские названия шаблонов не утверждены).
- *Проблема.* Краткое описание проблем(ы) проектирования, которую решает шаблон, и описание сопутствующих трудностей проектирования и альтернативных вариантов, с которыми сталкиваются проектировщики пользовательских интерфейсов.
- *Решение.* Ключевое проектное решение проблемы. Это краткое описание решения (т. е. шаблона), которое сопровождается примером (т. е. наглядным изображением), иллюстрирующим его применение для веб-приложения.
- *Зачем.* Логическое обоснование эффективности проектного решения.
- *Как.* Перечисление наиболее успешных опытов применения, описывающее применение этого решения и возможные вариации. Для каждой вариации приводится один или больше наглядных примеров, демонстрирующих, как эти шаблоны могут эффективно применяться в различных ситуациях.
- *Связанные шаблоны проектирования.* Поскольку очень часто происходит так, что несколько шаблонов вместе применяются для создания удобного проектного решения, в этом разделе указаны смежные шаблоны, которые могут пригодиться проектировщикам либо потому, что они применяются совместно с данным шаблоном, либо потому, что они влияют на применение данного шаблона.

⁴ Чтобы упорядочить разнообразие и несогласованность форм используемых шаблонов и чтобы найти способ объединения шаблонов и их языков от разных авторов в особые тематические коллекции шаблонов, участники конференции CHI 2003 предложили язык разметки на основе XML под названием Pattern Language Markup Language (PLML; произносится как «плэмэл»). Чтобы получить больше информации, см. работу Финчера (Fincher, 2003).

Структурирование шаблонов в этой книге

Шаблоны в этой книге структурированы по главам следующим образом.

Глава 2: Формы. Формы являются отличительной чертой веб-приложений. Именно с помощью элементов формы, таких как текстовые окна, раскрывающиеся меню, зависимые кнопки, флажки и прочие, пользователи вводят информацию и взаимодействуют с веб-приложениями. В этой главе рассказывается о шаблонах, связанных с разработкой форм для веб-приложений и обеспечивающих заполнение формы таким образом, чтобы оно не было трудоемким занятием, а также чтобы эти формы помогали пользователям достигать своих целей. Это такие шаблоны, как CLEAR BENEFITS, SHORT FORMS, LOGICAL GROUPING, LABEL ALIGNMENT, REQUIRED FIELD INDICATORS, SMART DEFAULTS, FORGIVING FORMAT, KEYBOARD NAVIGATION, INPUT HINTS/ PROMPTS, ACTION BUTTONS и ERROR MESSAGES.

Глава 3: Аутентификация пользователя. Веб-приложения осуществляют взаимно однозначное взаимодействие с пользователями и сохраняют информацию о конкретных пользователях. Для этого требуется, чтобы пользователи создавали учетную запись и получали к ней доступ с помощью уникальных учетных данных. В этой главе описываются шаблоны проектов, имеющие отношение к доступу и завершению работы с веб-приложениями, включая REGISTRATION, CAPTCHA, LOG IN, LOG OUT, AUTOMATIC LOGOUT и FORGOT USERNAME/ PASSWORD.

Глава 4: Главная страница приложения. Важный для проектировщиков вопрос – что пользователь увидит или на какую страницу будет направлен, когда он входит в свою учетную запись в приложении? В этой главе описаны шаблоны, которыми могут воспользоваться проектировщики при направлении посетителей на эту «главную» страницу приложения, сюда относятся INBOX, CONTROL PANEL, DASHBOARD, PORTAL, PERSONALIZATION, CUSTOMIZATION и BLANK SLATE.

Глава 5: Навигация. Навигация – это то, как пользователи работают с веб-приложением. Если процесс навигации спроектирован удачно, он становится прозрачным (т. е. незаметным для пользователя) и пользователи могут быстро осуществить свои задачи, не прибегая к ненужному поиску с возвратом. В центре внимания этой главы – шаблоны проектирования навигационных систем, такие как PRIMARY NAVIGATION, SECONDARY NAVIGATION, UTILITY NAVIGATION, FACETED NAVIGATION, SUPPLEMENTARY NAVIGATION, TAG CLOUDS, BREADCRUMBS и WIZARDS.

Глава 6: Поиск и фильтрация. Для большинства веб-приложений получение информации только с помощью навигационных систем может быть затруднительным и нанести ущерб удобству и простоте использования. По этой причине информация в веб-приложении доступна для поиска, чтобы пользователи могли быстро и эффективно найти то, что им нужно. Кроме тех случаев, когда критерии поиска очень особые, пользователи, скорее всего, получают большое количество результатов. Проектировщики должны предоставлять пользователям возможность с помощью механизмов фильтрации сократить список результатов до удобного в обработке набора вариантов. В этой главе рассказывается о шаблонах проектирования, связанных с поиском и фильтрацией информации в веб-приложениях, сюда относятся SIMPLE SEARCH, PARAMETRIC SEARCH, ADVANCED SEARCH, SEARCH TIPS, SEARCH RESULTS, SORTING, PAGINATION, CONTINUOUS SCROLLING, FILTERING, FACETED SEARCH и SAVED SEARCHES.

Глава 7: Списки. Списки применяются для того, чтобы представить пользователям несколько элементов. Способ этого представления может быть различным, в зависимости от того, какие именно элементы должны быть представлены. В этой главе рассказывается о шаб-

лонах проектирования для различных типов списков, а именно SIMPLE LIST, TABULAR LIST, HIERARCHICAL LIST, EVENT LIST, TIMELINES, IMAGE LIST/ GRID, MAPS, LIST ACTIONS и LIST UTILITY FUNCTIONS.

Глава 8: Богатые веб-приложения. Богатые веб-приложения (RIA) предоставляют такую же ответную реакцию и интерактивность, что и настольные клиентские приложения, поскольку пользователям не приходится ждать, пока основные данные на странице обновляются и пока вносятся изменения в оформление; по этой причине результаты их действий можно увидеть немедленно. В этой главе рассказывается о часто применяющихся шаблонах проектирования для богатых веб-приложений, к которым относятся RICH-TEXT EDITOR, RICH FORM, AUTOSUGGEST/ AUTOCOMPLETION, EDIT-IN-PLACE, OVERVIEW-PLUS-DETAIL, DYNAMIC QUERYING, LIVE PREVIEW, DRAG-AND-DROP, SLIDER, ANIMATIONS/TRANSITIONS, DELAY/PROGRESS INDICATOR, SPOTLIGHT/YELLOW-FADE и CAROUSEL

Глава 9: Социальные приложения. Современная тенденция в сфере веб-приложений – это разработка социальных приложений, которые не просто способствуют тому, чтобы пользователи вносили свой вклад и выкладывали свой контент (например, фотографии, видео, закладки и т. д.), но также развивают взаимодействие и помогают создавать сообщества. В этой главе рассказывается о шаблонах проектирования, которые были созданы на основе подобных социальных приложений, включая ADD/UPLOAD CONTENT, TAGGING, RATING, REVIEWS, VOTE TO PROMOTE, USER PROFILE, REPUTATION, DISCOVER NETWORK MEMBERS, FRIEND LIST, GROUPS/SPECIAL INTEREST COMMUNITY, MESSAGING, PRESENCE INDICATOR, SHARING и COLLABORATION.

Глава 10: Интернационализация. Интернационализация вебприложений является важным шагом к локализации – в данном случае к адаптации этих приложений к определенному региону или языку. Это помогает сократить усилия, необходимые для последующей локализации, и избавляет от необходимости разрабатывать отдельные региональные и языковые версии приложений. В этой главе рассказывается о шаблонах проектирования, которые помогают сделать приложение в достаточной степени универсальным и адаптируемым на первых этапах разработки, сюда относятся EXTENSIBLE DESIGN, DATE FORMAT, TIME FORMAT, NUMBER FORMAT, CURRENCY AND CURRENCY FORMAT, GLOBAL GATEWAY и LANGUAGE SELECTOR.

Глава 11: Доступность. Шаблоны проектирования, о которых говорится в этой главе, помогают сделать веб-приложения доступными и поддерживающими рекомендации и правила доступности информации в Интернете, например, правила, сформулированные на консорциуме W3C; к этим шаблонам относятся PROGRESSIVE ENHANCEMENT, SEMANTIC MARKUP, UNOBTRUSIVE STYLE SHEETS, UNOBTRUSIVE JAVASCRIPT, ACCESSIBLE FORMS, ACCESSIBLE IMAGES, ACCESSIBLE TABLES, ACCESSIBLE NAVIGATION и ACCESSIBLE ALTERNATIVE.

Глава 12: Визуальное проектирование. Визуальное проектирование веб-приложений сильно влияет на то, насколько полезным пользователи считают это приложение. Эта глава посвящена тем шаблонам проектирования, которые определяют, как выглядят и какое впечатление производят веб-приложения; сюда относятся LIQUID-WIDTH LAYOUT, FIXED-WIDTH LAYOUT, PROGRESSIVE LAYOUT, GRID STRUCTURE, VISUAL HIERARCHY, HIGHLIGHT и ICONS.

Глава 13: Библиотеки шаблонов. Несмотря на популярность шаблонов и библиотек шаблонов, в настоящее время еще не достигнуто соглашения о том, как шаблоны должны документироваться, соблюдаться и распространяться. По этой причине в данной главе представлен «шаблон» библиотеки шаблонов и показаны ее ключевые элементы, а также предложены лучшие варианты ее расширения.

Глава 14: Помощь. Несмотря на то что при соблюдении основных принципов, процессов и шаблонов проектирования создается удобный и эффективный в применении интерфейс, необходимо предоставить помощь, доступную на каждом этапе взаимодействия пользователя с системой. В главе перечислены шаблоны проектирования, связанные с предоставлением помощи в веб-приложениях, к ним относятся CONTEXTUAL HELP, FREQUENTLY ASKED QUESTIONS, APPLICATION HELP, GUIDED TOURS, HELP WIZARDS, HELP COMMUNITY и CLICK-TO-CHAT.

Примечание

В эту книгу включено много снимков экрана с веб-приложениями, эти снимки были сделаны в течение девяти месяцев. Хотя большая их часть актуальна с ноября 2008 по настоящий момент, некоторые могли измениться после выхода в печать этой книги. Это могло произойти, поскольку люди, занимающиеся разработкой вебприложений, меняют характеристики, вводят новые возможности и убирают устаревшие интерфейсы.

Применение шаблонов в этой книге

В этой книге предоставлено практическое руководство по проектированию пользовательских интерфейсов для разработки веб-приложений, предлагающее «работающую» отправную точку, с которой проектировщики могут начать внедрение и оптимизацию творческих решений.

Рассматривайте шаблоны, описанные в этой книге, как рекомендации, а не как конкретные точные проектные решения, и применяйте их, если только они подходят для решения вашей проблемы. Не применяйте шаблон для решения проблемы, только чтобы его применить! Как обозначил Грэхэм (Graham, 2003): «Шаблоны – это абстрактные, ключевые решения проблем, которые возникают в различных контекстах... Практическое применение данного решения может приобретать различные формы в различных приложениях. Поэтому шаблоны не являются готовыми к применению решениями». Поэтому сосредоточьтесь внимание на сущности шаблона и затем подумайте, как этот шаблон может помочь решить проблему, поскольку шаблоны указывают не на одно-единственное решение, а скорее на стратегию решения проблемы.

Глава 2

Формы

Введение

Формы являются отличительной чертой веб-приложений. С помощью элементов формы (например, текстовые поля, раскрывающиеся и прокручиваемые списки, переключатели, флажки и управляющие кнопки) веб-приложения позволяют пользователям выполнять такие задачи, как покупка товаров и услуг, бронирование авиабилетов, поиск местоположения, загрузка и выкладывание фотографий и т. д. Чтобы пользователи могли успешно выполнить свои задачи, важно, чтобы формы не были сложными и были спроектированы так, чтобы:

- их цель была ясна (CLEAR BENEFITS);
- они запрашивали только минимальную необходимую релевантную информацию (SHORT FORMS);
- их структура четко передавала взаимоотношения между элементами формы (LOGICAL GROUPING);
- метки и соответствующие элементы формы были выровнены, чтобы усовершенствовать процесс просмотра (LABEL ALIGNMENT);
- они четко указывали, что должен сделать пользователь (REQUIRED FIELD INDICATORS, INPUT HINTS/PROMPTS);
- они сводили к минимуму объем данных, которые должен ввести пользователь, и не требовали от пользователей дважды вводить одну и ту же информацию (SMART DEFAULTS);
- процесс заполнения формы был результативным (SMART DEFAULTS, FORGIVING FORMAT, KEYBOARD NAVIGATION);
- они четко указывали, как выполнить задание (ACTION BUTTONS);
- они давали пользователям четкие инструкции в случае возникновения ошибки (ERROR MESSAGES).

Хотя выбор подходящих элементов (например, когда использовать флажки, а когда – переключатели) крайне важен для создания удобной в использовании формы, здесь мы об этом говорить не будем, поскольку уже существуют отличные источники, посвященные этому вопросу; например, работа Мейхью (Mayhew, 1991), Галитца (Galitz, 2002), «Руководство по проектированию пользовательских интерфейсов» от компании Apple (2007) и руководство по разработке интерфейса от Windows Vista («User Experience Guidelines», 2007).

Однако важно помнить, что веб-приложения создаются с помощью HTML и предоставляют не все способы управления, доступные для популярных платформ, таких как Windows или Macintosh. В частности, процесс взаимодействия в веб-приложениях ограничивается следующими элементами формы: текстовые поля (в одну строку или состоящие из нескольких строк), переключатели, флажки, раскрывающиеся списки, прокручиваемые списки, кнопки (включая кнопки с изображением) и особый элемент управления для просмотра файлов. К недостающим элементам управления относятся, например, управление прокруткой, комбинированное окно, элемент управления «дерево» и вкладки. Хотя эти элементы управления можно внедрить с помощью сложных комбинаций HTML, каскадных таблиц стилей CSS и сценариев JavaScript, это всего лишь обходные пути решения проблемы, поскольку в данном случае эти элементы не являются частью основного языка разметки.

CLEAR BENEFITS (ЧЕТКИЕ ПРЕИМУЩЕСТВА)

Проблема

Увидев форму, пользователи могут не понять, как ее заполнение и отправка поможет им выполнить свои задачи и достичь своих целей.

Решение

При разработке четко обозначайте преимущества заполнения и отправки форм. Это особенно важно, когда пользователи создают новые учетные записи (т. е. в регистрационных формах), которые являются первым этапом получения доступа к возможностям веб-приложений (рис. 2.1).

To join LinkedIn, sign up below... it's free!

The image shows a registration form for LinkedIn. The form is titled "To join LinkedIn, sign up below... it's free!". It contains several input fields: "First Name", "Last Name", "Email" (with a "We don't spam" note), "Password" (with a "6 or more characters" note), "Country" (a dropdown menu showing "United States"), "Postal Code" (with an example "eg: 94043" and a note "Only your region will be public, not your postal code"), "I am currently:" (a dropdown menu showing "Employed"), "Company", and "Title". To the right of the form is a sidebar titled "LinkedIn helps you...". It contains three sections: "Re-connect" (with an icon of two people and the text "Connect and stay in touch with colleagues, classmates, & clients."), "Power your career" (with an icon of a briefcase and the text "Discover inside connections when you're looking for a job or new business opportunity."), and "Get Answers" (with an icon of a question mark and the text "Your network is full of industry experts willing to share advice. Have a question? Just ask.").

Рис. 2.1. Социальная сеть LinkedIn четко перечисляет преимущества регистрации в пункте LinkedIn helps you (LinkedIn поможет вам)

Зачем

Возможно, пользователи не хотят заполнять форму и предоставлять личную информацию, пока не поймут, какие преимущества они получают в этом случае и как это поможет им в достижении целей. Кроме того, они могут быть обеспокоены вопросами конфиденциальности, если не будут знать, как их личная информация будет использована. Четкое обозначение преимущества и объяснение пользователям, что нет причин волноваться по поводу конфиденциальности, – это первые шаги к тому, чтобы пользователи не отказывались от заполнения форм.

Как

Пользователи должны быть осведомлены о преимуществах заполнения формы, даже если это форма содержит всего одно поле для заполнения, например как форма подписки на электронную новостную рассылку (рис. 2.2).



Рис. 2.2. Сайт UIE перечисляет преимущества подписки на свою электронную новостную рассылку, «UIETips»

Объясняйте преимущества регистрации в регистрационных формах

Когда незарегистрированный пользователь просматривает регистрационную форму, вам предоставляется прекрасная возможность рассказать ему о преимуществах регистрации. Это облегчает принятие решения, регистрироваться ему или нет (рис. 2.3).

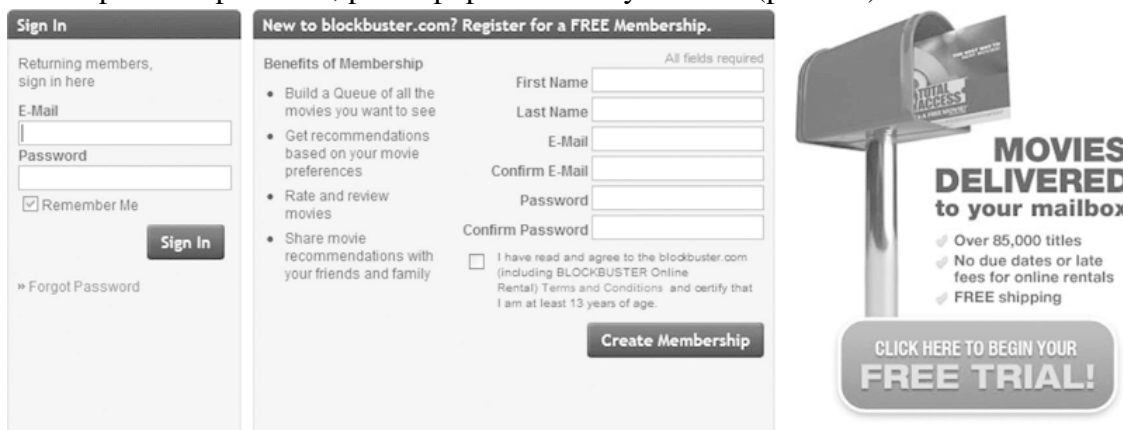


Рис. 2.3. Сайт Blockbuster перечисляет преимущества регистрации на Blockbuster.com и предлагает «бесплатный» пробный период каждому регистрирующемуся

Объясняйте преимущества, прежде чем перенаправлять пользователей на форму

Во многих случаях пользователей приходится перенаправлять на форму. Если они не знают о преимуществах, они могут не щелкнуть по ссылке или кнопке, ведущей их к форме. Поэтому очень важно, чтобы преимущества заполнения формы были обозначены до того, как пользователи увидят саму форму. Один из способов это осуществить – это поставить информативные метки на ссылки, например «Перевести деньги» или «Оплатить счета», вместо стандартных меток, таких как «Узнать больше» или «Продолжить». Хотя преимущества могут быть пользователям и непонятны, это помогает сообщить им о том, какое действие будет следующим (рис. 2.4).

Not a member yet?

Sign up and get the only online address book and calendar that syncs with Microsoft, Google, Yahoo!, Apple, AOL — and your mobile phone.

Sign up *It's free!*

[Learn more](#) ►

[Video demo](#) ►

Рис. 2.4. Сервис Plaxo размещает описание преимуществ регистрации рядом с кнопкой регистрации, и, что весьма оригинально, сайт предоставляет пользователям возможность узнать больше, посмотрев демонстрационный или обучающий ролик

Все чаще веб-приложения предоставляют такие возможности и преимущества, которые сложно описать в нескольких предложениях. Или даже когда преимущества очевидны, пользователи могут захотеть узнать больше о том, как получить эти преимущества при использовании приложения. Чтобы дать подробные объяснения, предложите пользователям узнать больше о работе веб-приложения, и они станут меньше беспокоиться по поводу заполнения необходимых форм (рис. 2.5 и 2.6).

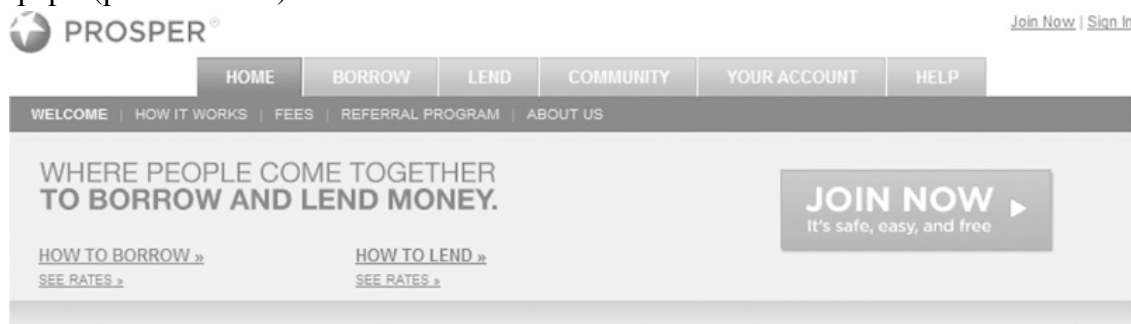


Рис. 2.5. Сайт Prosper (где можно дать или взять займы деньги) предоставляет информацию о том, как можно взять или дать деньги в долг с помощью ссылок How to borrow (Как взять в долг) и How to lend (Как дать займы)

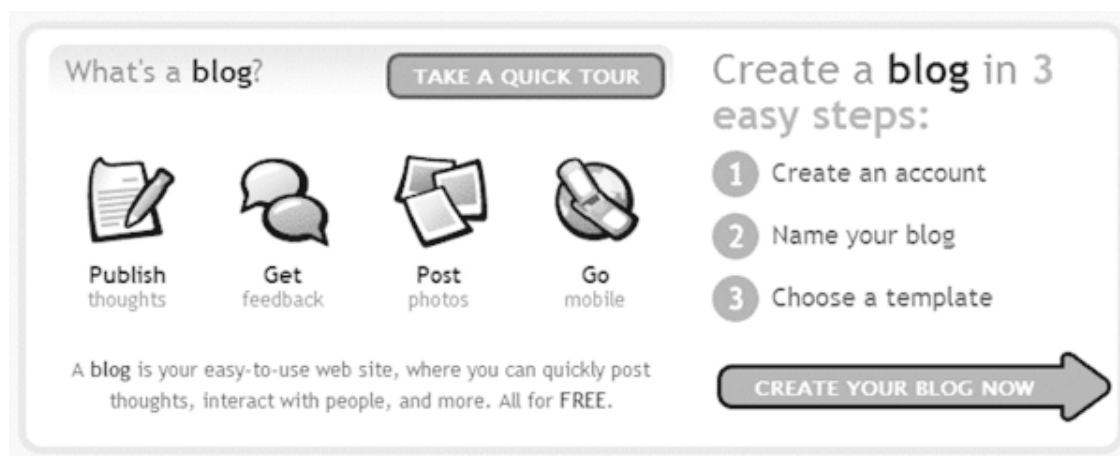


Рис. 2.6. Помимо перечисления преимуществ ведения блога (описание своих мыслей, получение комментариев и т. д.) сайт Blogger предоставляет информацию о ведении блога с помощью функции Take a Quick Tour (Короткий тур)

Связанные шаблоны проектирования

Для многих сложных веб-приложений и тех, где пользователь должен внести предоплату, попробуйте предоставить пользователям возможность воспользоваться интерактивным чатом CLICK-TO-CHAT (см. главу 14), который позволит им задавать вопросы непосредственно квалифицированному представителю компании.

SHORT FORMS (КОРОТКИЕ ФОРМЫ)

Проблема

Чем больше сведений о себе указывают пользователи, тем легче их понять и оптимизировать приложение, чтобы оно больше отвечало их потребностям и предоставляло более релевантную и персонализированную информацию. Однако объемные формы увеличивают возможность, что пользователи не заполнят форму или предоставят недостоверную информацию.

Решение

Делайте формы как можно короче (рис. 2.7). Не выясняйте у пользователей информацию, которая «может пригодиться» в будущем. Если дополнительные поля могут предоставить полезную информацию и их нельзя убрать, сделайте их необязательными и соответствующим образом обозначьте (см. далее в этой главе шаблон REQUIRED FIELD INDICATORS).

Choose a plan that fits your needs

Plans can be changed at any time, so feel free to start using Crazy Egg with a free plan.

The screenshot shows a registration form for Crazy Egg. On the left, there is a pricing table with a highlighted 'Basic' plan. The table lists features like 'Visits you can track per month' (10,000), 'Pages you can track at once' (10), 'Advanced Features' (checked), and 'Live Reporting' (checked). A 'SIGN UP' button is at the bottom of the table. To the right, the registration form is titled 'Grab an account now'. It includes fields for 'Email', 'Password', and 'Re-type password'. Below these is a checkbox for 'I have read and agree to the Terms of Use and the Privacy Policy.' and two buttons: 'CANCEL' and 'SUBMIT >'.

	Basic \$9 / month
Visits you can track per month	10,000
Pages you can track at once	10
Advanced Features	✓
Live Reporting	✓
SIGN UP	

Grab an account now

Email

Password Re-type password

☐ I have read and agree to the [Terms of Use](#) and the [Privacy Policy](#).

[CANCEL](#) [SUBMIT >](#)

Рис. 2.7. Для регистрации на сайте Crazy Egg пользователи должны заполнить всего три поля и принять условия использования и политику конфиденциальности. Более того, благодаря объединению формы с соответствующими преимуществами, пользователи точно знают, чего им ожидать

Зачем

Исследование, проведенное Relevant Ads, показало, что более короткие формы чаще заполняются (рис. 2.8). Сократив формы, можно также сократить вероятность возникновения ошибок, поскольку пользователям приходится иметь дело с меньшим количеством элементов формы. В дальнейшем это увеличивает шансы успешного заполнения формы, которое приводит к более высокому показателю эффективности.

	Short Form	Medium Form	Long Form
Conversion Rate	13.4%	12.0%	10.0%
Cost Per Conversion	\$31.24	\$34.94	\$41.90

Рис. 2.8. В исследовании, проведенном Relevant Ads, показатель эффективности падал с увеличением количества полей в форме

Как

Подумайте, насколько важен каждый элемент формы, и что будет, если не включать его в форму. Помимо этого подумайте, придется ли пользователям прилагать большие усилия, чтобы предоставить информацию. Если пользователям приходится задумываться о том, как ответить на вопрос в форме, рассматривайте это как препятствие заполнению формы, и позаботьтесь о том, чтобы убрать этот вопрос. Самая известная «простая» форма – это, вероятно, форма Google, которая находится на странице поиска и просто предоставляет поле для ввода текста и кнопку поиска (рис. 2.9).

Рис. 2.9. Форма поиска у Google самая простая и короткая. Хотя в нее входит кнопка I'm Feeling Lucky (Мне повезет!), большинство людей обычно вводят ключевые слова и щелкают по кнопке Google Search (Поиск в Google) или нажимают клавишу Enter на клавиатуре

Сервис Jottit предоставляет еще один пример самой короткой и, возможно, самой эффективной формы (рис. 2.10). Чтобы создать интернет-страницу, пользователи просто печатают

в поле ввода то, что они хотят разместить на своей странице и щелкают по кнопке «Create a page».



(a)



(б)

Рис. 2.10. На сайте Jottit, чтобы создать веб-страницу, пользователи просто вводят текст и щелкают по кнопке Create a page (Создать страницу) (а). Тогда пользователи получают свою интернет-страницу и возможность ее редактирования (б)

Разделяйте объемные формы на несколько страниц с более короткими формами

Группируйте информацию в большой форме так, чтобы у каждой группы была своя задача, и выкладывайте элементы формы, относящиеся к каждой группе, на отдельной странице (рис. 2.11). Помимо этого расставляйте группы в таком порядке, чтобы самая важная и необходимая информация располагалась вначале, а дополнительная информация – после нее.

Create a meebo account!

Instant Message anywhere with no downloads!

Access all of your IMs from any web browser with just one meebo account!

Join now! It's free!

By signing up for a Meebo account, you agree that you have read and understand Meebo's [terms of service](#) and [privacy policy](#).

Choose a username and enter a password for your meebo account

meebo id:
(A-Z, 0-9, underscores, and periods)

Password:
(minimum 8 characters)

Confirm:

Next

Рис. 2.11. На сайте Меебо регистрационная форма разделена на несколько страниц. Самая важная часть формы располагается вначале, а на остальных страницах пользователи могут установить дополнительные настройки своей учетной записи и указать свои предпочтения

Разделение формы ускоряет процесс заполнения каждой страницы, и пользователям такая форма может показаться короче, чем если разместить ее целиком на одной странице.

Связанные шаблоны проектирования

Когда формы максимально сокращены, пусть они кажутся еще более простыми для заполнения. Это можно сделать, указав пользователям, какая информация обязательна (REQUIRED FIELD INDICATORS), сгруппировав и расположив в логическом порядке элементы формы (LOGICAL GROUPING) и сообщив пользователям о преимуществах заполнения формы (CLEAR BENEFITS). Помимо этого попробуйте применить шаблон WIZARDS в формах, которые разделены на несколько страниц (см. главу 5).

LOGICAL GROUPING (ЛОГИЧЕСКОЕ ГРУППИРОВАНИЕ)

Проблема

Чтобы достичь результата, пользователям приходится заполнять довольно большие формы. Однако разработчикам хотелось бы, чтобы пользователям казалось, что форму легко заполнить и чтобы они охотно ее заполняли.

Решение

Группируйте элементы формы так, чтобы пользователям было понятно, какие данные нужно предоставлять в той или иной части формы (например, адрес доставки, платежная информация и т. д.; рис. 2.12).

1. Tell us about yourself...

My Name

Gender

Birthday

I live in

Postal Code

2. Select an ID and password

Yahoo! ID and Email @

Password Password Strength

Re-type Password

3. In case you forget your ID or password...

Alternate Email

Security Question

Your Answer

Рис. 2.12. Yahoo! разделяет элементы регистрационной формы на логические группы, благодаря чему создается впечатление, что форму можно легко и быстро заполнить

Зачем

Благодаря распределению информации по группам, так, чтобы пользователи могли четко понимать, для каких целей служит информация из каждой группы и какие элементы формы относятся к той или иной группе, формы кажутся легче для заполнения. Пускай лучше поль-

зователям форма представляется состоящей из небольшого количества групп, чем из большого количества отдельных элементов.

Как

Объедините элементы формы в группы в зависимости от их функций, например, адрес доставки, адрес выставления счета, контактная информация и т. д. Убедитесь, что порядок расположения элементов в каждой группе соответствует представлению пользователей о том, в каком порядке нужно вводить информацию. Например, для пользователей из США расположите элементы формы, относящиеся к информации об адресе, так, чтобы сначала нужно было вводить название улицы и дом, затем город, а только затем название штата и индекс, или, например, при создании учетной записи, сначала просите ввести имя пользователя (или адрес электронной почты), а затем пароль (рис. 2.13).

Secure connection Sign in **Billing & Shipping** Gift options Payment Verify

Billing And Shipping Information

Billing Address

First Name Last Name

Address (We cannot ship to PO boxes)

Town/City State Zip Code

Country
United States

Daytime Phone Ext.

Evening Phone

☐ Shipping address same as billing address

Shipping Address (for all shippable items)

First Name Last Name

Address (We cannot ship to PO boxes)

Town/City State Zip Code

Country
United States

Daytime Phone Ext.

[Click here for APO/FPO shipping options.](#)

Shipping Location

☐ My shipping address is a business address.
(This helps us choose the best shipping method for you.)

Рис. 2.13. Форма оформления заказа на сайте Apple кажется удобной для заполнения, во-первых, благодаря тому, что она разделена на несколько страниц, а во-вторых, благодаря логической группировке элементов на каждой странице

Разделив элементы формы на группы, расположите их в том порядке, который соответствует задачам пользователей и системным требованиям. Например, на странице оформления заказа лучше сначала задавать вопросы, касающиеся адреса доставки, поскольку эту информацию можно использовать при подсчете транспортных сборов и стоимости доставки так, чтобы на странице совершения платежа пользователям можно было показать итоговую стоимость и предоставить возможность указать, что адрес выставления счета совпадает с адресом доставки (рис. 2.14).

1 Sign in & Shipping **2 Payment** **3 Review & Submit Order**

Enter Shipping Information

Print This Page * Indicates Required Fields

Shipping

Select Shipping Profile
Pawan Vora 11152 Huron Street, Suite 211

Add New Shipping Information
Edit Your Shipping Information

Shipping Information

* First Name MI * Last Name
Pawan [] Vora

* Address (Dell cannot ship to a P.O. Box)
[]
[]

* City * State
Boulder Colorado

* Zip Code 4-digit Ext.
80501 []

* Daytime Phone Number Ext.
() - [] []

Evening Phone Number
() - [] []

Shipping Method

- ☒ 3-5 Day Delivery
\$0.00 (\$4.99 - \$4.99)
- ☐ 2nd Business Day Delivery
\$12.00
- ☐ Next Business Day Delivery
\$16.00

Continue

Рис. 2.14. Приложение Dell, как и многие другие приложения для электронной коммерции, отображает информацию о доставке и о платежах отдельно друг от друга, чтобы формы казались короче и пользователи уделяли отдельное внимание каждой порции информации

Связанные шаблоны проектирования

При группировке элементов формы можно разделить форму на множество страниц, чтобы она казалась короче (SHORT FORMS), или структурировать ее так, чтобы данные, введенные в начале формы, при необходимости автоматически вносились повторно в последующих частях формы (SMART DEFAULTS), и пользователям не приходилось возвращаться к ранее введенной информации.

LABEL ALIGNMENT (ВЫРАВНИЕ МЕТОК)

Проблема

Между метками и соответствующими элементами формы должна быть отчетливо видна связь, чтобы облегчить заполнение формы и сократить количество ошибок при вводе информации.

Решение

Существуют три неплохих варианта расположения меток по отношению к элементам формы: над элементом, с выравниванием левой границы метки относительно элемента; слева, с выравниванием левой границы метки относительно других меток; и слева, с выравниванием правой границы метки относительно других меток (рис. 2.15).

Tell us about yourself - All fields are required

First name Last name

Street address

City

State / Province ZIP / Postal code Country or region

Select United States

(a)

Join Now! It's quick and easy. [Already a member? Please login above.](#)

* Choose a Username

* Email Address

* Re-type Email Address

* Choose Password

(b)

Sign up for a free Ning account

Your Ning ID lets you create new social networks and join existing social networks on Ning.
Already have a Ning ID? **Sign In.**

Name

Email Address
You'll use this email to sign into any network on Ning.

Password

Retype Password

(в)

Рис. 2.15. Во многих случаях метки располагаются а) над элементами формы; б) выровнены по левому краю и в) выровнены по правому краю

Зачем

Чем быстрее пользователи могут соотнести метку с соответствующим элементом формы, тем быстрее они смогут заполнить форму. Изучив движения глаз, Пензо (Penzo, 2006) обнаружил, что пользователям достаточно просто соотнести метку с элементом формы, если метки расположены любым из способов, показанных на рис. 2.15.

Однако результаты изучения движений глаз показали, что когда метки полей выровнены по левому краю, большие расстояния между некоторыми метками и соответствующими им полями ввода (возникающие, поскольку метки обладают различной длиной – например, сравните «Название компании» и «Город») приводят к тому, что у пользователей уходит больше времени на зрительное восприятие формы. По сравнению с выравниванием по левому краю, при выравнивании по правому краю общее количество исправлений в 2 раза меньше, что значительно сокращает усилия, которые должен приложить пользователь для заполнения формы.

В этом же исследовании описаны преимущества размещения меток полей над элементами формы; при таком расположении меток формы заполняются быстрее всего. Недостатком в данном случае является то, что при таком расположении меток требуется дополнительное вертикальное пространство. Однако если форму нужно перевести на несколько языков, благодаря такому расположению сохраняется внешнее единообразие оформления, поскольку одни и те же метки при переводе на другие языки могут обладать разной длиной. При размещении меток над элементами формы для меток остается достаточно места, не зависимо от изменения их длины (см. шаблон EXTENSIBLE DESIGN в главе 10).

Как

Чтобы правильно соотнести метки с соответствующими элементами формы (для языков с письменностью в направлении слева направо), разместите метки либо слева, либо над элементами управления (Penzo, 2006; Wroblewski, 2008). При размещении меток полей слева от

элементов формы выровняйте метки по правому краю так, чтобы они находились близко друг к другу (рис. 2.16).

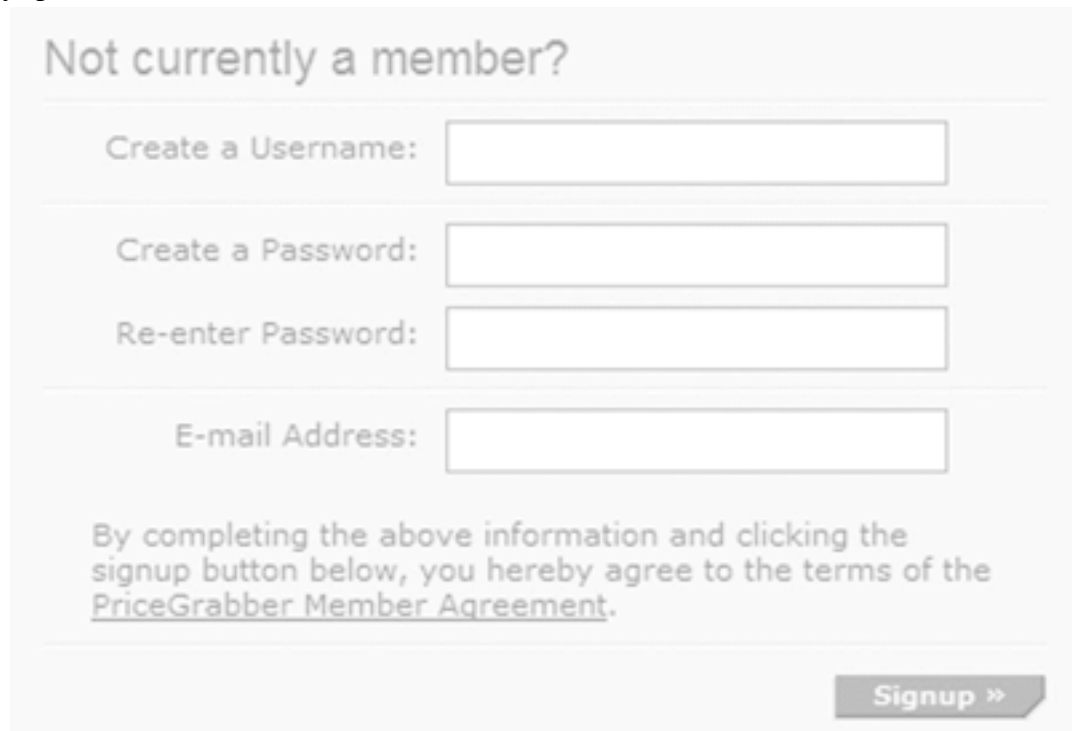


Рис. 2.16. На сайте PriceGrabber метки расположены слева от элементов формы, но выровнены относительно них по правому краю

При размещении меток над элементами формы важно, чтобы визуально метка одного элемента формы находилась на достаточном расстоянии от предыдущего элемента (Penzo, 2006). Врублевски (Wroblewski, 2008) советует, чтобы расстояние составляло примерно 50 %-75 % высоты отдельного поля для ввода данных. Помимо этого Пензо (Penzo, 2006) советует применять метки с неформатированным шрифтом вместо меток с жирным шрифтом, поскольку жирный шрифт несколько сложнее воспринимать (рис. 2.17).

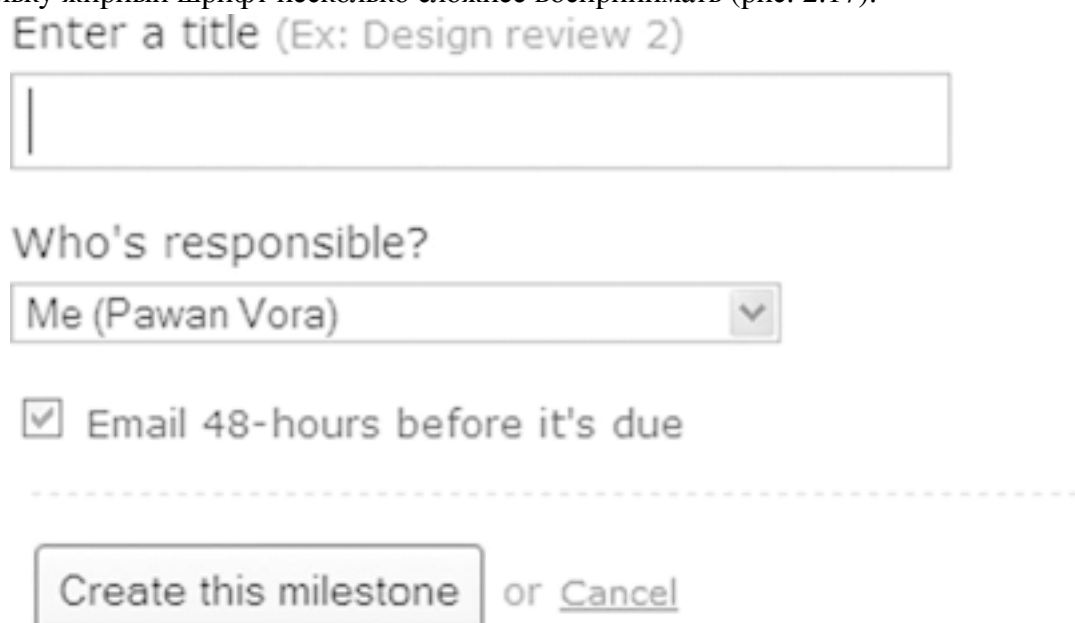
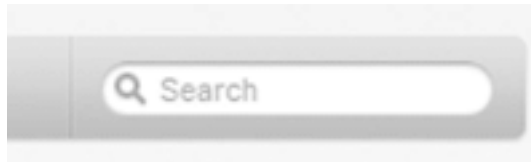


Рис. 2.17. На сайте Basecamp над полями расположены метки с неформатированным текстом

Не злоупотребляйте встраиваемыми метками

Встраиваемые метки допустимы в формах, в которых совсем немного полей ввода (обычно одно), и в тех случаях, когда большинство пользователей должны знать, какие данные им нужно вводить, например, на рис. 2.18 это ключевые слова поиска. Поскольку встраиваемые метки сами находятся внутри поля и пользователю приходится их удалять, их не видно, когда пользователи ввели данные в это поле. Поэтому они неудобны в больших формах или в тех случаях, когда маловероятно, что пользователи знают, какие данные они должны ввести.



(a)



(б)

Рис. 2.18. На сайте Apple в поле поиска используются встраиваемые метки (a). Метка исчезает, когда пользователь щелкает по полю поиска или начинает вводить искомые слова (б)

Связанные шаблоны проектирования

Обычно в формах метки сопровождаются индикаторами обязательной к заполнению информации (REQUIRED FIELD INDICATORS).

REQUIRED FIELD INDICATORS (ИНДИКАТОРЫ ОБЯЗАТЕЛЬНЫХ ПОЛЕЙ)

Проблема

Чтобы выполнить задание, пользователи должны предоставить определенную информацию. Например, при создании учетной записи в большинстве случаев пользователи должны указать электронный адрес и пароль. Однако для усовершенствования процесса взаимодействия в формах могут присутствовать дополнительные вопросы, на которые пользователи могут не захотеть отвечать или ответы на которые им могут быть на данный момент неизвестны. Если не заполнены обязательные поля, пользователи могут получить сообщение об ошибке, и у них уйдет больше времени на заполнение формы.

Решение

Четко укажите обязательную информацию в форме, чтобы пользователи могли ее успешно ее заполнить и чтобы уменьшить вероятность появления сообщения «заполнены не все обязательные поля» (рис. 2.19).

The image shows a registration form for Domino's. At the top, a red asterisk is followed by the text "Required Information". The form contains several input fields: "First" and "Last" (combined as "Name*"), "Address Type*" (a dropdown menu with "House" selected), "Street Address*", "City*", "State*" (a dropdown menu with "Select..." selected), "Zip Code*", "Phone Number*" (with hyphens for area code and number), "Extension*", "E-mail Address*", "Confirm E-mail Address*", and "Age*" (with a checkbox for "13 and over"). To the right of the form, a "NOTE:" box states: "This will be your default address. You may add more addresses later." Below the main form, there is a section for a cell phone number with the text "Enter your cell phone # to receive great offers straight to your cell phone." and a checkbox for "Same as main phone". At the bottom of this section, a note states: "Note: Standard text messaging rates apply. To opt out of Domino's text messaging program, send the word STOP to DOMINO (366466) at any time."

Рис. 2.19. На сайте Dominos обязательные поля обозначены красными звездочками, так пользователю становится понятно, что номер мобильного телефона вводить необязательно. Также указано, какие преимущества пользователь получит, если укажет свой мобильный телефон

Зачем

Указание обязательных полей не только экономит время, которое пользователи тратят на то, чтобы решить, какую информацию они должны предоставить, но также экономит время,

которое у них отнимают сообщения об ошибке в случае, если не все поля заполнены, и которое уходит на то, чтобы заполнить форму заново.

Как

В веб-приложениях необходимые для заполнения поля часто обозначаются звездочками (обычно красными). Использование красных звездочек эффективно потому, что они понятны даже людям с нарушением восприятия цветов. Не рекомендуется использовать другие виды индикации (например, использовать красный и жирный шрифт для меток полей), поскольку часто именно таким образом указаны ошибки на странице. Помимо этого программы экранного доступа могут не уловить изменение цвета, а также люди с нарушенным цветовым зрением могут не различить обязательные и дополнительные поля.

Хотя могут быть и другие учитывающие перечисленные выше нюансы способы указать обязательные поля, красные звездочки все же предпочтительны, поскольку большинство пользователей привыкло видеть именно их при указании обязательных полей; цвет звездочек может варьироваться в зависимости от фона страницы.

Совет

Пользователи знают, что на странице регистрации обязательно вводить имя пользователя и пароль. Поэтому не обязательно указывать их как обязательные.

Расшифруйте значение индикаторов обязательных полей

Хотя большинство пользователей Интернета понимают, что красная звездочка рядом с меткой поля указывает на то, что это поле является обязательным, в некоторых веб-приложениях этот символ указывает на дополнительные поля. Поэтому следует внести ясность и указать наверху формы, что поля, отмеченные звездочкой (*), обязательны.

Размещайте индикаторы обязательных полей одинаково во всех формах

На данный момент не существует рекомендаций по поводу того, как лучше располагать индикаторы обязательных полей относительно меток полей; их можно расположить в любом месте: после метки, перед меткой, перед полем для ввода данных и после него.

Однако предположив, что при заполнении формы глаза пользователя направлены больше на элементы формы, имеет смысл расположить индикаторы обязательных полей как можно ближе к элементам формы. Помимо этого, если располагать эти индикаторы в одном и том же месте, пользователи смогут быстро просматривать формы и определять, какие поля являются обязательными. Можно расположить индикаторы обязательных полей слева от элементов формы, при этом метки будут расположены слева от них и выровнены по правому краю (см. рис. 2.19). Для меток, находящихся над элементами формы, расположите индикаторы обязательных полей слева от меток (рис. 2.20), поскольку так они будут ближе как к меткам полей, так и к элементам формы, где посетитель вводит или выбирает данные; если разместить их слева от элементов формы, их можно будет перепутать с флажками или зависимыми кнопками.

Write a hotel review

Thank you for writing a hotel review. By taking a few minutes to share your opinions, you are helping other travelers make informed decisions. Your review is subject to our review guidelines.

* Name of Hotel:

* Hotel's Location: Ex. "Rome, Italy", "Miami South Beach"

* Review Headline:

> Show examples

* Your Rating: Click numbers to rate the hotel

1 2 3 4 5 6 7 8 9 10 ?

> Show detailed scoring

Рис. 2.20. На сайте TravelPost указатели на обязательные поля находятся слева от меток, расположенных над элементами формы. Однако обратите внимание, что в метках используется полужирный шрифт, что не рекомендуется, поскольку выделенные жирным шрифтом метки увеличивают время на заполнение формы (Penzo, 2006)

Не указывайте дополнительные поля

В случаях если форма содержит меньше дополнительных полей, чем обязательных, иногда, чтобы избежать скопления символов, отмечают дополнительные поля вместо обязательных. Это неудачное решение. Поскольку пользователи привыкли использовать веб-приложения, в основной части которых применяются индикаторы обязательных полей, они, скорее всего, будут воспринимать отмеченные поля как обязательные или, по меньшей мере, такое обозначение внесет путаницу.

Объясняйте, для чего необходимо предоставлять конфиденциальную информацию

Если вам необходимо получить личную информацию, например дата рождения пользователя, пол или раса, четко обозначьте, почему эта информация вам необходима. Кроме этого предоставьте ссылку на «Политику конфиденциальности», где пользователи смогут узнать, каким образом их информация будет использоваться.

Связанные шаблоны проектирования

Даже если в форме четко указаны обязательные поля, проектировщики все равно должны стремиться максимально сократить общее количество полей в форме – как дополнительных, так и обязательных (SHORT FORMS). Кроме того, если пользователь не заполняет обязательное поле, покажите ему сообщение на той же самой странице, на которой он не заполнил одно из обязательных полей (ERROR MESSAGES).

SMART DEFAULTS («УМНЫЕ» ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ)

Проблема

С добавлением каждого элемента формы увеличивается как время, которое пользователь тратит на заполнение формы, так и вероятность возникновения ошибок. Кроме того, если в формах из нескольких частей пользователю приходится несколько раз вводить одну и ту же информацию, увеличивается не только время, необходимое для заполнения формы, но также возникает необходимость, чтобы приложение проверяло, каждый раз вводилась одна и та же информация.

Решение

Применяйте соответствующие значения по умолчанию, в зависимости от того, что, скорее всего, выберут или введут пользователи; можно исходить из того, что пользователь выбрал или ввел ранее (рис. 2.21). Однако не стоит автоматически повторно вводить те элементы формы, которые должны быть подтверждены (например, повторно вводить пароль или согласие пользователя с условиями и положениями использования) или имеют отношение к безопасности (например, смена пароля).

The screenshot shows a web interface for adding a new milestone. At the top, there are navigation tabs: Overview, Messages, To-Do, Milestones (selected), Writeboards, and Chat. Below the tabs is the heading 'Add a new milestone'. The form is divided into two main sections. The left section, titled 'When's it due?', contains a calendar for October 2007. The date '21' is highlighted, and the text 'Due: October 21, 2007' is displayed below the calendar. The right section contains a text input field for 'Enter a title (Ex: Design review 2)', a dropdown menu for 'Who's responsible?' with 'Me (Pawan Vora)' selected, and a checked checkbox labeled 'Email 48-hours before it's due'. At the bottom of the form are two buttons: 'Create this milestone' and 'Cancel'.

Рис. 2.21. В сервисе Basecamp по умолчанию задана сегодняшняя дата в поле When's it due? (Когда будет готово) и данные о зарегистрированном пользователе в поле Who's responsible? (Кто отвечает?), а также установлен флажок Email 48-hours before it's due (Напомнить мне по электронной почте за 48 часов до того, как срок истечет), чтобы упростить процесс добавления новой задачи

Зачем

Применение соответствующих значений по умолчанию сводит к минимуму необходимость вводить или выбирать данные. Это сокращает общее время, необходимое для выполнения задания, и количество ошибок при вводе данных. В некоторых случаях, использование значений по умолчанию может устранить необходимость заполнять форму и сведет задачу пользователя просто к подтверждению информации. Например, в приложениях для электронной коммерции пользователям, которые ранее предоставляли информацию об адресе доставки и адресе выставления счета (и разрешили сохранить эту информацию), не нужно заново заполнять все поля на страницах оформления покупки. Вместо этого после нажатия кнопки «Сделать заказ» их можно направить на страницу предварительного просмотра/подтверждения. Поскольку у большинства пользователей адрес доставки и адрес выставления счета меняются нечасто, время оформления заказа сильно сокращается; однако у пользователей должна быть возможность внести изменения в эту информацию.

Значения по умолчанию выполняют еще одну функцию: у пользователей появляется пример, какие данные и в каком формате они должны вводить, что сокращает вероятность возникновения ошибки при вводе данных (van Duynе et al., 2006).

Как

Анализируйте каждый элемент формы – поля ввода текста, переключатели, раскрывающиеся списки и т. д. – и если можно логически предположить, исходя из ранее введенной пользователем информации или из контекста, какие данные будет вводить пользователь, введите эти значения самостоятельно.

Настройки по умолчанию можно применить и к последовательности выполнения заданий; например, если пользователь выполнил задание А и, скорее всего, далее он приступит к выполнению задания Б, направьте пользователей непосредственно на страницу выполнения задания Б (рис. 2.22).

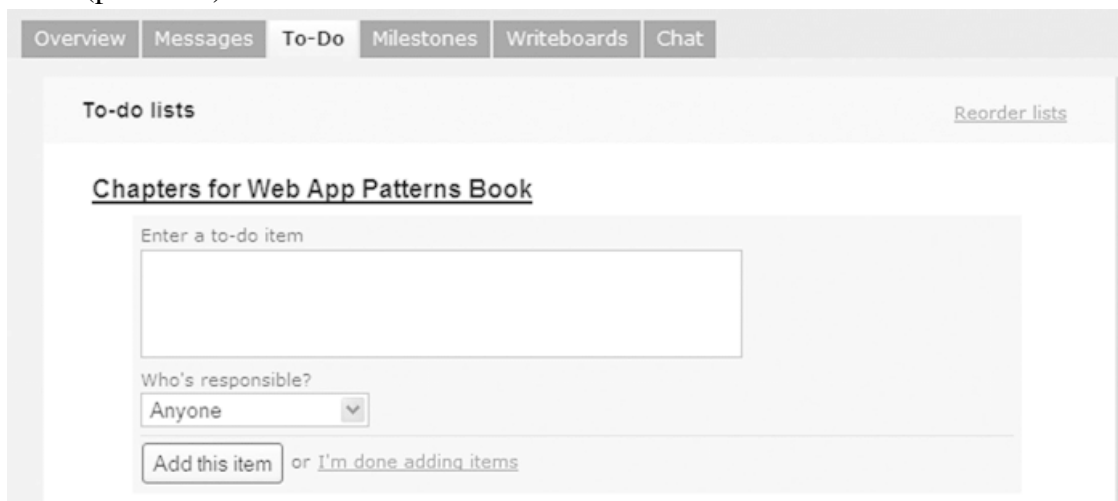


Рис. 2.22. В сервисе Bascamp для усовершенствования процесса смены заданий применяются настройки по умолчанию. Когда пользователи создают список заданий («To-do»), на странице отображается форма Enter a to-do item (Введите задание), и пользователям не приходится щелкать по элементу Add a To-Do (Добавить задание)

Не применяйте значения по умолчанию для персональной информации

Для персональной информации, такой как пол, обращение, возраст, раса и т. д., не меняйте значения по умолчанию, поскольку это может показаться оскорбительным некоторым пользователям или будет воспринято как пристрастность. Например, если установить пол – мужской или женский – по умолчанию, у пользователей может возникнуть ощущение, что вы пристрастны; то же самое произойдет, если вы установите по умолчанию обращение Господин или Госпожа.

Кроме того, когда неясно, необходим ли для данного приложения определенный вид персональных данных, лучше обозначить соответствующие поля как необязательные или предоставить пользователям возможность скрыть эту информацию. То есть в приложениях, посвященных поиску пары, пользователи понимают, зачем нужно указывать свой пол. Однако при регистрации электронного ящика пользователям может быть непонятно, зачем нужно указывать свой пол.

Не устанавливайте по умолчанию значения там, где предоставлена возможность выбора

Когда пользователю предоставляется возможность выбора, например, при оформлении подписки на новостную рассылку от компании или посредника, убедитесь, что выбор по умолчанию не отражает интересы компании. Если пользователи читают формы невнимательно, они могут подписаться на рассылки или услуги, на которые на самом деле подписываться не хотят. А впоследствии они будут воспринимать рассылку компании как спам или перестанут доверять этой компании.

Связанные шаблоны проектирования

Разумные настройки по умолчанию являются еще одним способом упрощения и сокращения формы (SHORT FORMS). Кроме того, они уменьшают вероятность возникновения ошибки и появления сообщений об ошибке (ERROR MESSAGES).

FORGIVING FORMAT (ВЕЛИКОДУШНЫЙ ФОРМАТ)

Проблема

В формах пользователям часто приходится предоставлять такую информацию, как номера телефонов, номера банковских карт, даты и т. д., а подобную информацию можно вводить в различных форматах. Например, пользователь может по-разному ввести телефонный номер: без пробелов (например, 3035555555), разделив цифры пробелами или дефисами (например, 303 555 5555 или 303-555-5555), или разделив цифры иным образом (например (303) 555-5555). Даже когда у пользователей есть пример, они все равно могут ввести данные не в том формате. Сообщение об ошибке может замедлить процесс заполнения формы и оттолкнуть пользователей, если несколько элементов формы обладают жесткими требованиями к формату.

Решение

Пусть пользователи вводят данные в различных форматах: спроектируйте веб-приложение таким образом, чтобы эти форматы были допустимы и чтобы в случае ввода данных в каком-либо формате не появлялось сообщение об ошибке (рис. 2.23).

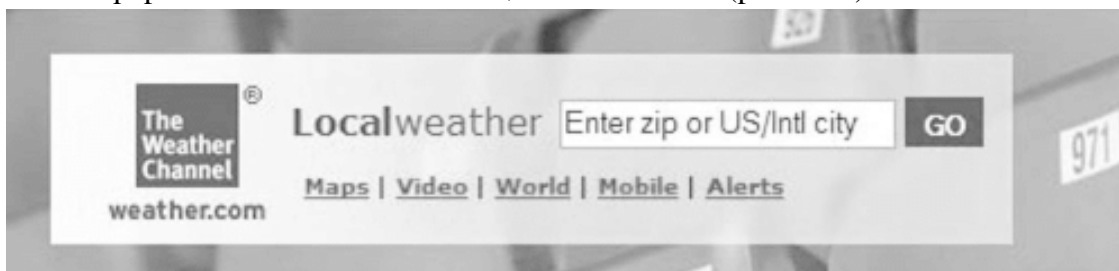


Рис. 2.23. На сайте канала погоды пользователи могут ввести индекс или название своего города, чтобы узнать информацию о погоде. Для этого используется только одно поле для ввода текста вместо двух отдельных

Зачем

Для некоторых видов информации (например, даты, номера телефонов, почтовый индекс и т. д.) существует несколько «правильных» способов ввода данных. Во время заполнения формы не обременяйте пользователей необходимостью определенным образом форматировать данные, гораздо проще, чтобы приложение допускало различные форматы данных и самостоятельно обрабатывало их необходимым образом.

Как

Продумайте альтернативные способы ввода данных и затем доработайте приложение так, чтобы оно принимало эти варианты и верно их интерпретировало. Для допускающих двоякое толкование данных предоставьте пользователям возможность выбрать из нескольких вариантов, так чтобы у них не возникло ощущения, будто они допустили ошибку и чтобы они

чувствовали, что успешно продвигаются вперед (рис. 2.24). Также можно предложить пользователям выбор из нескольких вариантов (рис. 2.25; см. также шаблон AUTOSUGGEST/AUTOCOMPLETION в главе 8).

Please help us with a little more information

1 We found more than one airport that matched 'San Francisco'. Please select an airport from the list below.

Going to:

- ☐ San Francisco, CA (QSF-All Airports)
☐ Oakland, CA (OAK-Oakland Intl.)
☐ San Francisco, CA (SFO-San Francisco Intl.)
☐ San Jose, CA (SJC-San Jose Intl.)
 Or
☐ Enter a new location or use [Airport lookup](#)

Рис. 2.24. На сайте Expedia пользователям предлагается выбрать один аэропорт из нескольких вариантов, когда введенные данные (в данном примере Сан-Франциско) могут трактоваться по-разному

The screenshot shows the Kayak website's flight search section. The 'From' field is set to 'DEN' and the 'To' field is set to 'san'. A dropdown menu is open under the 'To' field, displaying a list of suggested destinations starting with 'san'. The suggestions include San Diego, CA - San Diego (SAN), San Francisco, CA - San Francisco (SFO), San Juan, Puerto Rico - Luis Munoz Marin (SJU), San Jose, CA - San Jose (SJC), Santa Ana, CA - J. Wayne/Orange Cnty (SNA), San Antonio, TX - San Antonio (SAT), San Jose, Costa Rica - Juan Santamaria Intl (SJO), Montego Bay, Jamaica - Sangster Intl (MBJ), San Jose Cabo, Mexico - Los Cabos (SJD), and Santo Domingo, Dominican Republic - Las Americas (SDQ). The interface also includes options for trip type (Round-trip, One-way, Multi-city, Weekend), travel dates (11/16/2007), cabin class (Economy), and the number of travelers (1).

Рис. 2.25. На сайте Kayak применяется шаблон AUTOSUGGEST/ AUTOCOMPLETION, предлагающий пользователю выбор из нескольких вариантов, чтобы сократить количество возможных ошибок

Предложите пользователям подсказки при вводе/приглашение к вводу

Даже если приложение разработано таким образом, что допустимы различные форматы, продемонстрируйте пользователям пример хотя бы одного приемлемого формата (см. шаблон INPUT HINTS/PROMPTS далее в этой главе). Такие инструкции помогают пользователям избавиться от сомнений по поводу надлежащего способа ввода данных.

Связанные шаблоны проектирования

Различные способы форматирования вводимых данных – это лишь один из способов уменьшить количество ошибок при вводе и упростить процесс заполнения формы. Ошибки

при вводе можно сократить также с помощью предоставления необходимых инструкций по форматированию (INPUT HINTS/PROMPTS) и предоставив пользователю возможность выбрать из нескольких вариантов (см. шаблон AUTOSUGGEST/ AUTOCOMPLETION в главе 8).

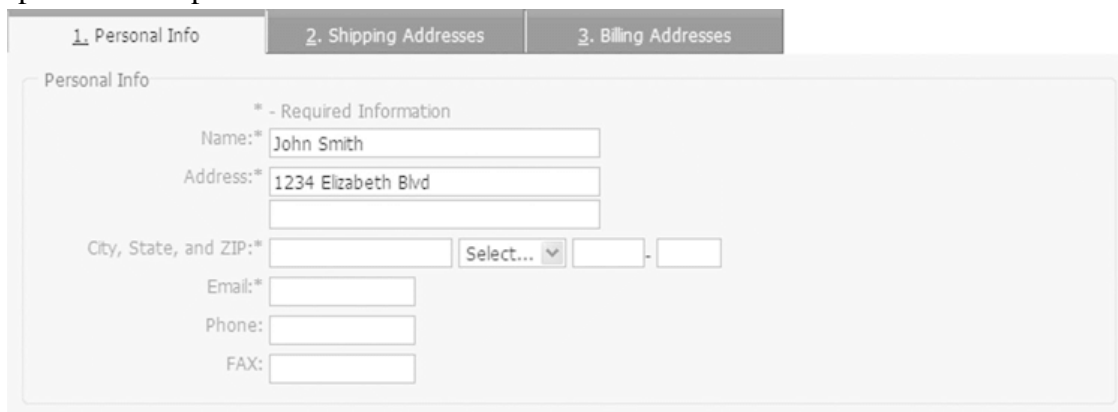
KEYBOARD NAVIGATION (УПРАВЛЕНИЕ КЛАВИАТУРОЙ)

Проблема

При заполнении форм пользователи часто перемещаются между полями с помощью клавиши Tab или выбирают какой-либо вариант из раскрывающегося списка с помощью клавиш клавиатуры. Если пользователям в обязательном порядке придется применять то мышь, то клавиатуру для заполнения определенных частей формы, тогда многие пользователи не просто не захотят, но и не смогут заполнить эту форму по техническим причинам.

Решение

Пускай пользователи будут иметь возможность применять клавишу Tab для перехода от одного элемента к другому. Помимо этого убедитесь, что пользователи могут нажать клавишу Enter для отправки формы; а также, если это поможет повысить эффективность процесса взаимодействия, предоставьте им возможность перемещаться по приложению с помощью сочетаний клавиш клавиатуры (рис. 2.26). Также убедитесь, что с помощью клавиш клавиатуры можно просмотреть раскрывающиеся списки, особенно те из них, которые созданы на основе сценариев JavaScript.



The image shows a web form with three tabs: "1. Personal Info", "2. Shipping Addresses", and "3. Billing Addresses". The "Personal Info" tab is currently selected. Below the tabs, there is a section titled "Personal Info" with a sub-header "* - Required Information". The form contains several input fields: "Name:" with the value "John Smith", "Address:" with the value "1234 Elizabeth Blvd", "City, State, and ZIP:" with a dropdown menu showing "Select..." and a text input field, "Email:", "Phone:", and "FAX:". The "Shipping Addresses" tab is highlighted in the tab bar.

Рис. 2.26. В этом примере пользователи могут перемещаться по форме с помощью клавиши Tab, а также переходить от закладки к закладке с помощью сочетаний клавиш. Например, на вкладку Billing Address (Адрес для выставления счета) можно перейти, нажав сочетание Alt +3 (Windows) или Ctrl+3 (Mac)

Зачем

Позволив пользователям пользоваться клавиатурой для навигации, вы не только сделаете форму доступнее, но также увеличите скорость ее заполнения, поскольку в этом случае пользователям не придется во время заполнения формы регулярно менять клавиатуру на мышь и наоборот.

Как

Разрешите навигацию с помощью клавиатуры во всех формах, т. е. пользователи должны получить возможность заполнять формы, даже если они будут использовать только клавиатуру, либо только мышь. Уделите особенное внимание процессу перемещения между полями. По умолчанию веб-браузеры предоставляют возможность перемещаться между элементами страницы (вкладками, элементами формы и ссылками) в зависимости от того порядка, в котором они расположены в HTML-коде. На самых хорошо организованных страницах проектировщикам не приходится указывать порядок перемещения. Однако в тех случаях, когда HTML-код не соответствует последовательности смены заданий пользователя (т. е. тому порядку, в котором пользователь будет заполнять форму), замените заданный по умолчанию порядок перемещения с помощью атрибута `tabindex`, как показано в примере:

```
<input type="text" name="fieldname" id="fieldname" tabindex="110" />
```

Это необходимо в большинстве случаев, когда формы расположены в несколько колонок и важно, чтобы клавиша `Tab` перемещала курсор к следующему по логике элементу формы, а не просто слева направо или сверху вниз – в порядке, в котором расположены элементы формы в HTML коде.

Примечание

Значением атрибута `tabindex` может быть число от 1 до 32 767. При кодировании формы и применении атрибута `tabindex` увеличьте значение `tabindex` на 10. Если возникает необходимость изменить форму и вставить один или несколько элементов между существующими элементами формы, можно использовать числа между существующими значениями `tabindex`, это не повлияет остальные элементы формы.

Будьте внимательны при установке курсора в первом поле формы

Когда первоочередной задачей для пользователей является заполнение формы (например, при поиске, регистрации, входе в учетную запись и т. д.), обычно допускается, чтобы при загрузке страницы курсор располагался в первом поле – так, чтобы пользователи могли сразу приступить к заполнению формы. Однако избегайте размещения курсора в каком-либо поле формы на тех страницах, на которых располагаются навигационные элементы, позволяющие пользователям перейти к другим частям приложения, или на страницах, содержащих необходимый для чтения контент (например, инструкции по заполнению формы или сообщения об ошибках). Автоматическое размещение курсора в таких случаях может сделать страницу непригодной для пользователей, у которых установлена программа экранного доступа, поскольку они, скорее всего, не увидят информацию, расположенную выше данного поля формы.

Попробуйте настроить горячие клавиши

Разработайте горячие клавиши для приложений, которые будут часто использоваться и основным приоритетом которых будет эффективность взаимодействия (например, приложения для поддержки покупателей). Горячие клавиши можно разработать на основе HTML с помощью атрибута `accesskey`, как в следующем примере:

```
<input type="button" value="buttonName" accesskey="k" />
```

Благодаря спецификации атрибута `accesskey`, пользователи смогут переходить к тому или иному элементу формы при нажатии клавиши `Alt` (или `Ctrl` (Mac)) плюс той буквы или цифры, которая указана в атрибуте `accesskey`.

При создании горячих клавиш важно не создавать и не переопределять горячие клавиши, которые часто используются в браузерах. Список этих горячих клавиш приведен в табл. 2.1.

Таблица 2.1. Горячие клавиши в браузерах

F	Вызывает меню Ф айл (File) в IE (Internet Explorer) и Firefox
E	Вызывает меню П равка (Edit) в IE и Firefox
V	Вызывает меню В ид (View) в IE и Firefox
A	Вызывает меню И збранное (Favorites) в IE
T	Вызывает меню И нструменты (Tools) в IE и Firefox
H	Вызывает меню С правка (Help) в IE и Firefox
B	Вызывает меню З акладки (Bookmarks) в Firefox
S	Вызывает меню Ж урнал (History) в Firefox

Связанные шаблоны проектирования

Предоставление пользователям возможности пользоваться клавиатурой для навигации не только помогает быстро заполнить форму, но также необходимо для того, чтобы веб-страницы были доступными (см. шаблон ACCESSIBLE FORMS в главе 11).

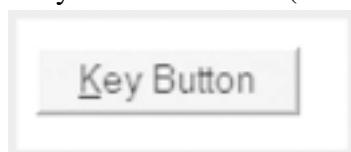
Примечание

Чтобы пользователям было известно о существовании горячих клавиш, подчеркните соответствующие клавиши с помощью таблицы стилей.

В случае с управляющими кнопками для этого необходимо применить элемент button так, как показано в следующем примере:

```
<style type="text/css">
quick-key {
text-decoration: underline;
}
</style>
<button type="button" accesskey="k"><span class="quick-key">K</span>ey Button</button>
```

В этом случае кнопка выглядит следующим образом, а пользователи могут нажать Alt (или Ctrl) +k на клавиатуре, чтобы ей управлять:



Подчеркивание невозможно для элемента input type="button", поскольку теги HTML недействительны в пределах значения его атрибута, где указывается текст управляющей кнопки.

INPUT HINTS/PROMPTS (ПОДСКАЗКИ ПРИ ВВОДЕ/ПРИГЛАШЕНИЕ К ВВОДУ)

Проблема

В определенных случаях пользователи могут не знать, какие данные вводить в определенные поля и нужно ли при этом придерживаться какого-либо формата. По этой причине пользователи могут предоставить либо неверную информацию, либо ввести данные в неверном формате.

Решение

Предоставьте пользователям необходимые подсказки, объяснения или инструкции, и покажите им, как они должны вводить данные (рис. 2.27). В тех ситуациях, когда данные могут быть введены несколькими способами, применяйте шаблон FORGIVING FORMAT (о котором упоминалось ранее).

The screenshot shows the 'Create a Group' interface on LinkedIn. At the top, there's a navigation bar with 'Groups', 'My Groups', 'Groups Directory', and 'Create Groups'. Below this is the 'Create a Group' heading. The 'Upload Group Logos' section contains two rows. Each row has a placeholder image (a square with 'in' and 'Group'), a text input field, a 'Browse...' button, and a help icon (?). Below the first row, text specifies '100x50 pixels; PNG, JPEG, GIF; max size 100 KB'. Below the second row, text specifies '60x30 pixels; PNG, JPEG, GIF; max size 100 KB'. The 'Group Information' section follows, with a 'Group name:' label and a text input field. Below that is a 'Group description:' label and a larger text area containing the instruction: 'Enter a brief description of your group and its purpose, i.e., "This is a group for User Experience Design professionals to expand our network of people and ideas."' At the bottom, there's a 'Group type:' label and a dropdown menu currently showing 'Choose...'.

Рис. 2.27. При создании группы на сайте LinkedIn пользователи получают необходимые подсказки и инструкции по загрузке логотипов группы, описание таких требований, как размеры изображения, форматы и размеры файлов, а также здесь показан пример описания группы, встроенный в поле для ввода текста

Зачем

Разъяснив, что должен сделать пользователь, вы избавите его от необходимости догадываться и сократите вероятность возникновения ошибок, что увеличивает скорость заполнения формы.

Как

Существует несколько способов, как предоставить пользователям подсказки и инструкции (рис. 2.28):

The screenshot shows the 'Create a Windows Live ID' registration form. It includes several sections with instructions and hints:

- Create a Windows Live ID:** A text input field for the ID, a dropdown for the domain (currently 'live.com'), and a 'Check availability' button. A hint above the ID field says 'is available'.
- Choose your password:** Three input fields: '*Type password:', 'Password strength:', and '*Retype password:'. A hint to the right states: 'Strong passwords contain 7-16 characters, do not include common words or names, and combine three of these character types: uppercase letters, lowercase letters, numbers, or symbols. Get help with this'.
- Enter password reset information:** Three input fields: 'Alternate e-mail:', '*Question:', and '*Secret answer:'. Hints below the fields state: 'This must be different from your Windows Live ID', 'Select one' (for the question dropdown), and 'Five-character minimum; not case sensitive'.

Рис. 2.28. При заполнении регистрационной формы Windows Live пользователи получают инструкции по вводу пароля, альтернативного электронного адреса и секретного вопроса. Кроме того, когда пользователи приступают к заполнению поля, им предоставляется дополнительная информация и возможность получить помощь, пройдя по специальной ссылке

- *Предоставьте примеры*, как можно ввести данные. Например, если в поле ввода электронного адреса можно ввести несколько адресов, покажите пример того, как несколько введенных электронных адресов разделены запятыми или другим разделительным знаком.
- *Покажите, какие форматы приемлемы* для таких данных, как даты, номера телефонов, номера кредитных карт и т. д. Для дат покажите допустимые форматы следующим образом: мм/дд/гг, дд/мм/гг или мм/дд/гггг; для номеров телефонов в России покажите формат ввода как xxx-xxx-xxxx и т. д.).
- *Покажите, какие ограничения* действуют для каждого поля, например, какое минимальное или максимальное количество символов в него можно вводить. Например, в поле для ввода пароля пользователям нужно ввести хотя бы шесть символов, среди которых будет хотя бы один специальный символ, отличный от пробела.

Пусть подсказки и разъяснения будут короткими – не больше нескольких слов или одного предложения – чтобы пользователи их не игнорировали. Кроме того, располагайте подсказки и примеры как можно ближе к соответствующим элементам формы. Чтобы пользователи не путали подсказки и метки, сделайте подсказки менее заметными с помощью более светлого тона и/или меньшего шрифта.

Попробуйте использовать динамические контекстуальные подсказки

В тех случаях, когда подсказки или разъяснения должны содержать подробные или сложные инструкции, попробуйте показывать подсказки динамически, когда пользователь начинает вводить данные в элемент формы или подводит к этому элементу (или группе элементов) указатель мыши (рис. 2.29). Недостаток этого подхода заключается в том, что пользователи должны приступить к вводу текста, чтобы увидеть инструкции.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.