

Оглавление

Предисловие	9
Введение	11
Принцип I. Прямые действия	21
Глава 1. Непосредственное редактирование	24
Непосредственное редактирование однострочного текста	25
Непосредственное редактирование многострочного текста	29
Редактирование в оверлее.....	33
Редактирование таблицы	37
Групповое редактирование	40
Конфигурирование модуля.....	43
Рекомендации по выбору шаблона редактирования.....	45
Глава 2. Перетаскивание	46
Интересные моменты.....	47
Перетаскивание модулей	53
Перетаскивание элементов списка	66
Перетаскивание объектов	73
Действия при перетаскивании	79
Группирование объектов путем перетаскивания	84
Возможные трудности при реализации перетаскивания	87
Глава 3. Непосредственное выделение	88
Выделение с помощью переключателя	89
Множественное выделение	94
Выделение объекта.....	97
Смешанное выделение	100
Принцип II. Легким касанием	105
Глава 4. Контекстные инструменты	108
Взаимодействие в контексте страницы	108
Контекстные инструменты	110
Видимые инструменты	111

Всплывающие инструменты	114
Переключаемые инструменты	122
Многоуровневые инструменты	124
Вспомогательное меню	129
Принцип III. Не уходя со страницы	133
Глава 5. Оверлеи	136
Диалоговый оверлей	138
Информационный оверлей	144
Оверлей для ввода данных	151
Глава 6. Инлей	155
Диалоговый инлей	155
Инлей в списке	159
Информационный инлей	164
Набор вкладок	166
Инлей или оверлей?	168
Глава 7. Виртуальные страницы	169
Виртуальная прокрутка	169
Постраничная навигация внутри страницы	175
Постраничная навигация с прокруткой (карусель)	180
Виртуальное панорамирование	183
Масштабируемый интерфейс пользователя	185
Постраничная навигация или прокрутка?	189
Глава 8. Выполнение операции	191
Google Blogger	191
Волшебный принцип	192
Интерактивное одностороннее выполнение	195
Внутристраничный помощник	199
Выполнение в диалоговом оверлее	203
Конфигуратор	207
Статическое одностороннее выполнение	208
Принцип IV. Очевидность приглашений	215
Глава 9. Статические приглашения	217
Приглашение с призывом к действию	217
Приглашение к ознакомлению	222
Глава 10. Динамические приглашения	227
Всплывающее приглашение	227
Привычное приглашение	233

Приглашение при перетаскивании	236
Логичное приглашение	245
Приглашение к дополнительному контенту	247
Преимущества приглашений	250
Принцип V. Применение переходов	251
Глава 11. Шаблоны переходов	253
Приглушение и подсветка	253
Разворачивание/сворачивание	258
Заполнение пустот	263
Анимация	263
Освещение	267
Глава 12. Зачем нужны переходы?	270
Привлекательность интерфейса	270
Информирование пользователя	271
Сохранение контекста при смене вида на экране	271
Объяснение происходящего	279
Отражение взаимоотношений между объектами	283
Привлечение внимания к происходящему	283
Повышение воспринимаемой производительности	286
Создание иллюзии виртуального пространства	288
Принцип VI. Мгновенная реакция	291
Глава 13. Шаблоны поиска	294
Автозаполнение	294
Предложение поисковых запросов	298
Поиск в режиме реального времени	304
Расширенный поиск	311
Глава 14. Шаблоны обратной связи	319
Предварительный просмотр в режиме реального времени	319
Постепенное появление	329
Индикатор выполнения	332
Периодическое обновление	338
Заключение. Принципы и шаблоны для расширенного взаимодействия	341
Алфавитный указатель	344

Предисловие

В архитектуре *композицией (parti)* называют основополагающую концепцию здания.¹ Что предпочесть – строгий академичный подход с заранее четко обозначенной функциональностью или определенную гибкость, допускающую быстрые оперативные изменения? Архитектор, претворяя в жизнь каждый конкретный проект, должен не только воплотить его суть, но и учесть огромное количество нюансов, от которых в конечном счете зависит успех дела.

Принципы проектирования – ключевой фактор любой архитектурной композиции. Они задают основные характеристики здания, которые должны иметь в виду все участники проекта, включая клиентов, градостроителей, рабочих и инженеров. Принципы проектирования являются своего рода «краеугольными камнями», служащими критериями оценки всех принимаемых решений; благодаря им отдельные части проекта слагаются в единое целое. Но для успешного завершения строительства недостаточно знания только этих принципов.

Каждая часть здания, от чердака до внутреннего дворика, обладает индивидуальным набором функциональных возможностей и ограничений, которые должны вписываться в общую композицию. Речь идет о геометрических размерах, положении в пространстве, эстетических аспектах, условиях эксплуатации и т. д. Для того чтобы достигнуть гармонии, архитектор обязан тщательно изучить все эти нюансы и, взвесив все за и против, выбрать наиболее подходящие из доступных вариантов.

Принимая во внимание принципы проектирования с одной стороны и учитывая все субъективные соображения с другой, архитекторы создают функциональные здания, полностью отвечающие своему предназначению.

Идея сочетания общих принципов проектирования и особенностей, связанных с конкретной ситуацией, пригодится не только тем, кто занимается архитектурной композицией, но тем, кто создает современные веб-приложения с расширенными возможностями. Именно этой точки зрения придерживаются Билл Скотт и Тереза Нейл, авторы книги, которую вы держите в руках. Имея за плечами более чем 30-летний опыт

¹ Термин *parti* был впервые употреблен в книге Мэтью Фредерика (Matthew Frederick) «101 Things I Learned in Architecture School», MIT Press, 2007.

проектирования и разработки программного обеспечения, Билл и Тереза являются высококлассными специалистами в области систематизации. В данной книге они представляют собственную теорию, подробно описывая причины провала некоторых известных интерактивных ресурсов, и предлагают свои методы создания успешных проектов.

Богатый практический опыт позволил им выделить ряд принципов проектирования успешных веб-приложений. Эти принципы – от «прямых действий» до «мгновенной реакции» – позволят вам оценить интерактивные возможности своих сайтов. Билл и Тереза подробно рассматривают сопутствующие каждой ситуации обстоятельства и возможные компромиссы, снабжая описание каждого принципа соответствующими соображениями и примерами наиболее удачного использования на практике, так что в вашем распоряжении будет вся необходимая информация для принятия верного решения.

Теперь, дополняя фундаментальные принципы проектирования индивидуальными особенностями, веб-дизайнеры могут создавать полноценные структуры, позволяющие пользователям взаимодействовать, общаться и выполнять свою работу.

Итак, в путь, к новым идеям и секретам, с которыми вы сможете реализовать самые невероятные композиции веб-приложений!

Льюк Вроблевски (Luke Wroblewski),
руководитель Product Ideation and Design, Yahoo! Inc.,
автор книг «Web Form Design: Filling in the Blanks» (Rosenfeld Media)
и «Site-Seeing: A Visual Approach to Web Usability» (Wiley)

Октябрь, 2008

Введение

С чего все начиналось

Свой первый персональный компьютер я (Билл) приобрел примерно в 1981 году; это был Radio Shack Color Computer с резиновой клавиатурой. Несколько месяцев моим основным пользовательским интерфейсом была командная строка, где я вводил код COLOR BASIC.

Последующий переход на Apple IIe принес более удобную клавиатуру и множество игр. Но интерфейс по сути остался тем же. В те дни господствовали командная строка и текстовые меню. Выход на сцену IBM PC ничего не изменил. Lotus 123, самый современный тогда табличный процессор, управлялся набором замысловатых клавиатурных команд. Ничего нового для пользователя.

А затем началась новая революционная эпоха в развитии интерфейса. В 1984 году появился Macintosh, которой вскоре поселился и у меня дома. Использование мыши открывало совершенно новый интерактивный мир. Теперь для навигации не нужно было запоминать устаревшие текстовые команды – новая среда предлагала интуитивно понятные способы взаимодействия.

Хорошо, подумаете вы, ну и что из того? Одно дело 1984 год, другое – сегодня. Какое отношение все это имеет к книге, посвященной проектированию веб-интерфейсов?

Самое прямое.

На протяжении бóльшей части истории Всемирной паутины интерфейс сайтов и приложений был весьма примитивным, что напоминало раннюю эру настольных компьютеров. Интерактивные возможности сайтов (возможности взаимодействия с пользователем) в основном предполагали два события:

- Нажатие гиперссылки
- Отправку данных с помощью формы

Попробуй заинтересовать этим пользователя! А для пущей сложности каждый щелчок мышью на ссылке или кнопке отправки данных вызывал обновление страницы. Казалось, создать идеальные условия для взаимодействия с пользователем попросту невозможно.

Любопытно, что технологии, позволявшие избавиться от этих ограничений, уже давно существовали. Но разработчики обратились к ним лишь после того, как их стали поддерживать наиболее популярные браузеры. В 2004 году Google запустил сервисы Gmail и Google Maps, основанные на комплексе технологий, для которого Джесси Джеймс Гарретт позднее придумал термин Ajax.

Отличие было существенным. Сервис Google Maps на основе Ajax позволял перемещать и увеличивать изображение в реальном времени, как в обычных настольных приложениях, и все это без обновления страницы. В то же время Mapquest, как и большинство веб-приложений того времени, при каждом смещении карты или изменении масштаба перезагружал страницу. Разница между старой Всемирной паутиной и совершенно новым миром, предлагаемым Ajax, была очевидной.

Почему мы решили написать книгу

Мне повезло быть свидетелем первой революции – в настольных приложениях (я даже написал одну из первых игр для Macintosh¹), а мой соавтор Тереза Нейл стала свидетелем второй революции – во Всемирной паутине.

Через несколько лет наши пути пересеклись в Sabre (головной компании Travelocity). Вместе мы создали команду по проектированию пользовательских интерфейсов и помогли улучшить многие продукты, выполняя эвристическую оценку и участвуя в полном редизайне веб-приложений. Эта работа позволила нам выявить много удачных шаблонов проектирования пользовательского интерфейса, а также антишаблоны – типичные ошибки.

Затем я перешел в Yahoo! и много сделал для революционного продвижения интерфейсов Ajax во Всемирной паутине. Одно из моих достижений в Yahoo! – запуск общедоступного сервиса Yahoo! Design Pattern Library. Как евангелист Ajax в Yahoo! я встречался и обсуждал с ведущими специалистами компании значение этих новых способов взаимодействия и важность их применения в особом контексте веб-среды. В результате за несколько лет я провел бесчисленное количество переговоров на эту тему, делясь наилучшими практиками с веб-разработчиками и дизайнерами по всему миру.

За это время Тереза состоялась как консультант по дизайну пользовательских интерфейсов. В ходе работы она продолжала совершенствовать первоначальный набор шаблонов и принципов, под ее руководством было спроектировано более 30 действующих насыщенных веб-приложений для сайтов как делового, так и общего назначения. Тереза и ее клиенты постоянно задействовали эти шаблоны в качестве инструментария при разработке новых приложений и редизайне уже существующих.

¹ GATO, выпущена компаний Spectrum Holobyte в 1985 году.

Эта книга – результат нашего опыта, тщательно отобранные знания и идеи, накопленные мной и Терезой более чем за 30 лет. К нам часто обращаются с вопросами, и мы пришли к выводу, что лучшим решением, позволяющим поделиться информацией с широкой публикой, будет написание книги.

О чем эта книга

Эта книга не об информационной архитектуре, хотя это понятие также будет рассматриваться на ее страницах. И не о визуальном дизайне, хотя мы затронем и эту тему, важную для качественного оформления приложения.

Эта книга посвящена проектированию *взаимодействия*, а именно – проектированию взаимодействия в веб-среде. А еще точнее – проектированию *расширенного* взаимодействия в веб-среде. Это квинтэссенция самых эффективных методов, шаблонов и принципов создания интерактивных приложений с расширенными возможностями для использования именно в веб-среде.

Дело в том, что веб-среда имеет свои *уникальные особенности*, отличающие ее от настольной. Со временем границы между ними становятся все более расплывчатыми, но тем не менее остается определенная специфика создания именно веб-приложений с широкими интерактивными возможностями. Возможность редактирования контента прямо на странице уходит корнями в традиции настольных приложений, однако в веб-среде имеет свои особенности. В данной книге подробно рассматриваются эти по-своему уникальные интерактивные возможности в виде набора шаблонов проектирования в контексте нескольких ключевых принципов проектирования.

Шаблоны проектирования

Что мы понимаем под шаблоном проектирования?

Кристофер Александер в своей замечательной работе «A Pattern Language: Towns, Buildings, Construction» (Oxford University Press) ввел термин «шаблон» (pattern), чтобы отразить наиболее распространенные архитектурные решения. Согласно его определению шаблон это:

...периодически возникающая задача и описание основных принципов ее решения...

Позднее термин «шаблон» был применен к программному обеспечению в книге «Design Patterns: Elements of Reusable Object-Oriented Software» (Addison-Wesley) Эриха Гаммы (Erich Gamma), Ричарда Хелма (Richard Helm), Ральфа Джонсона (Ralph Johnson) и Джона М. Влиссидеса (John

М. Vlissides).¹ Несколько лет спустя сфера его употребления расширилась за счет области проектирования пользовательских интерфейсов.²

Именно в этом значении шаблон и понимается в данной книге: *шаблон проектирования взаимодействия*. Здесь вы найдете более 75 шаблонов, иллюстрирующих самые распространенные методы организации расширенного взаимодействия в веб-среде. Каждый шаблон авторы сопровождают конкретным примером действующего сайта. Поскольку речь идет об интерактивности, для более полного разъяснения понятий приводится множество рисунков. Мы не только подробно рассмотрим все нюансы применяемого решения, но и укажем «плохие» шаблоны (анти-шаблоны), которых следует избегать. В каждом разделе приводятся соображения по поводу лучших методов проектирования.

Шаблоны распределены в соответствии с шестью принципами проектирования, составляющими основу данной книги:

Принцип первый: прямые действия

Как говорит Алан Купер, «там, где выводятся данные, должна быть и возможность их ввода». В этом состоит принцип прямых манипуляций. К примеру, вместо отдельной страницы для редактирования данных предоставляется возможность их редактирования по месту. Посвященные данному принципу главы 1–3 включают шаблоны «Непосредственное редактирование», «Перетаскивание» и «Непосредственное выделение».

Принцип второй: легким касанием

В ходе редизайна Yahoo! 360 дизайнер Эриксон де Джесус (Ericson deJesus) употребил выражение «легким касанием», говоря о необходимости сокращения усилий, прилагаемых пользователем для нормальной работы с сайтом. Верный путь к этому – использование контекстных инструментов. Шаблоны, иллюстрирующие данный принцип, рассмотрены в главе 4 «Контекстные инструменты».

Принцип третий: не уходя со страницы

Необходимость обновления страницы нарушает нормальный ход мысли. Вместо того чтобы обновлять страницу при каждом действии пользователя, стоит взглянуть на вещи его глазами. В зависимости от этого можно принимать решения, когда будет разумнее не перезагружать страницу. Вывод информации в перекрывающем слое или во внутреннем слое страницы рассматривается в главе 5 «Оверлеи» и главе 6 «Инлеи». Представление динамического контента обсуж-

¹ Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. «Приемы объектно-ориентированного проектирования. Паттерны проектирования». – Пер. с англ. – СПб: Питер, 2001.

² См. книгу Дженифер Тидвелл (Jenifer Tidwell) «Designing Interfaces: Patterns for Effective Interaction Design» (O'Reilly) и библиотеку шаблонов Мартижина ван Уэлли (Martijn van Welie), <http://www.welie.com/>.

дается в главе 7 «Виртуальные страницы». В последней главе этого раздела мы поговорим о том, как вместо перехода с одной страницы на другую организовать потоки информации на одной странице.

Принцип четвертый: очевидность приглашений

Заметность, легкость обнаружения опций – одна из самых насущных проблем при создании интерактивности в Сети. Нет смысла добавлять какую-либо опцию, если пользователь ее не заметит. Хороший способ решить эту проблему – предоставлять пользователю соответствующие приглашения, то есть указательные знаки, помогающие ему перейти на новый уровень взаимодействия с приложением. Главы 9 и 10 данного раздела соответственно посвящены статическим приглашениям, постоянно находящимся на странице, и динамическим приглашениям, появляющимся в ответ на действия пользователя.

Принцип пятый: применение переходов

Применение анимационных эффектов и иных визуальных переходов – очень полезный прием. Мы поговорим о нем в главе 11, рассмотрим набор самых распространенных шаблонов перехода. В главе 12 мы обсудим, для чего нужны переходы. Также будет рассмотрен ряд ошибочных шаблонов.

Принцип шестой: мгновенная реакция

Умный интерфейс отзывчив. Данный принцип объясняет, как создается расширенное взаимодействие с быстрой реакцией на действия пользователя. В главе 13 рассматривается набор шаблонов поиска, включая поиск в режиме реального времени, подбор возможных запросов, расширенный поиск и функцию автозаполнения. В главе 14 рассматриваются шаблоны обратной связи, включая динамический предварительный просмотр, постепенный вывод информации, индикаторы выполнения и периодическое обновление.

Для кого предназначена эта книга

Эта книга адресована всем, кто специфицирует, проектирует или создает веб-интерфейсы.

Изложенные здесь принципы особенно полезны веб-дизайнерам, поскольку образуют идейный каркас, основу философии создания продуманных до мелочей интерактивных веб-интерфейсов. Представленные в книге шаблоны послужат прекрасным дополнением к их инструментарию, а сотни приведенных примеров – полезным справочным материалом. И, конечно, лучшие методы займут свое место в списке средств эффективного взаимодействия.

Для руководителей предлагаемые шаблоны и примеры станут отличной отправной точкой в решении бизнес-задач. Хотя в книге не описана программная сторона решений, веб-разработчиков заинтересуют пред-

ставленные здесь шаблоны, поскольку каждый из них имеет реальное воплощение в коде. Для всех участников проекта эти шаблоны составляют общий «словарь», используемый и в управлении, и в проектировании, и в разработке. В итоге формируется прочная основа для более эффективной работы в команде.

Представленные в книге разнообразные примеры реализации принципов проектирования и шаблонов послужат подспорьем в работе как новичку, так и специалисту в области разработки веб-интерфейсов.

Дополнительные материалы

У этой книги есть справочный веб-сайт (www.designingwebinterfaces.com), который содержит обновленные примеры, дополнительную информацию о принципах, шаблонах и лучших методах проектирования, а также ссылки на статьи и полезные ресурсы по проектированию веб-интерфейсов.

Все использованные в книге диаграммы и рисунки доступны по лицензии Creative Commons; их можно свободно загружать с сервиса Flickr (www.flickr.com/photos/designingwebinterfaces/) и использовать в своих собственных презентациях.¹

Обозначения, принятые в книге

В данной книге приняты следующие типографические обозначения:

Курсив

Для URL-адресов, имен каталогов и файлов, опций, иногда просто для выразительности.

Полужирное начертание

Для названий шаблонов.

Рубленый шрифт

Для выделения элементов интерфейса – кнопок, пунктов меню и т. д.

Примечание

Так оформлены советы, примечания или заметки общего характера.

Использование примеров

Все приводимые в качестве примеров изображения можно найти на сайте Flickr (<http://www.flickr.com/photos/designingwebinterfaces/>). Их

¹ Также все *цветные* версии иллюстраций можно скачать единым архивом по адресу <http://www.symbol.ru/library/design-web-interface/all.zip> или поглавно (<http://www.symbol.ru/library/design-web-interface/01.zip>, ...[02.zip](http://www.symbol.ru/library/design-web-interface/02.zip) и т. д.).

можно использовать в собственных презентациях и т. п. при соответствующем соблюдении лицензии Creative Commons и со ссылкой на источник, включая название книги, автора, издательство и ISBN, например: «Designing Web Interfaces», Copyright 2009 Bill Scott and Theresa Neil, 978-0-596-51625-3.

Если использование материалов выходит за эти рамки, обратитесь за разрешением по адресу permissions@oreilly.com.

Нам интересно ваше мнение

Представленная в данной книге информация протестирована и проверена самым тщательным образом, но остается вероятность, что определенные функции изменились или же что в текст вкралась ошибка (как бы ни было трудно в это поверить). Сообщения о найденных ошибках, а также пожелания по поводу будущих изданий просим присылать по адресу:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (в США или Канаде)
707-829-0515 (международный или местный)
707-829-0104 (факс)

У этой книги есть страница в Интернете, которая содержит примеры и наметки для последующих изданий. Она доступна по адресу:

<http://www.oreilly.com/catalog/9780596516253>

Также с нами можно связаться по электронной почте. Для подписки на рассылку или запроса каталога напишите по адресу:

info@oreilly.com

Свои комментарии к этой книге присылайте по адресу:

bookquestions@oreilly.com

Дополнительную информацию о наших книгах, конференциях, ресурсных центрах и сети O'Reilly можно найти на нашем сайте:

<http://www.oreilly.com>

Safari Books Online



Если на обложке технической книги есть пиктограмма «Safari® Books Online», это означает, что книга доступна в Сети через O'Reilly Network Safari Bookshelf.

Safari предлагает намного лучшее решение, чем электронные книги. Это виртуальная библиотека, позволяющая без труда находить тысячи лучших технических книг, вырезать и вставлять примеры кода, за-

гружать главы и находить быстрые ответы, когда требуется наиболее верная и свежая информация. Она свободно доступна по адресу <http://safari.oreilly.com>.

Благодарности

Благодарности Билла

Эта книга – не только результат наших с Терезой совместных усилий. Нас вдохновляли многие, большинство – не напрямую.

Самую большую благодарность я хочу выразить Руфи. Вот уже 30 лет ты – моя чудесная жена и мой друг, а также замечательная мать. Без твоей поддержки и терпения я бы никогда не закончил эту книгу.

Я многим обязан моим редакторам из издательства O'Reilly. Отдельное спасибо Мэри Треслер (Mary Treseler) за ее хитроумные уловки, заставившие нас с Терезой закончить книгу, и за критические замечания в процессе нашей работы над ней. Я признателен всем остальным участникам команды, активно помогавшим этой книге увидеть свет: Рейчел Монаган (Rachel Monaghan), Марлоу Шеффер (Marlowe Shaeffer), Рону Билодью (Ron Bilodeau), Коллин Горман (Colleen Gorman), Адаму Уитверу (Adam Witwer), Роберту Романо (Robert Romano) и многим другим.

Каждый, кто когда-либо писал книгу, знает, что ее проверяют научные редакторы. Спасибо за приятные слова одобрения и конструктивную критику Кристиану Крамлишу (Christian Crumlish), Дэну Сафферу (Dan Saffer), Люку Вроблевски (Luke Wroblewski), Юхану Сонину (Juhan Sonin), Кевину Артуру (Kevin Arthur) и Алану Баумгартену (Alan Baumgarten). Я внимательно отнесся к каждому замечанию, и все они существенно повлияли на окончательный вариант книги.

Я многим обязан времени, проведенному в компании Yahoo!. Спасибо Эрин Малоун (Erin Malone) за письмо, подобное грому среди ясного неба, которое привело меня в Yahoo!. Я смог пообщаться с замечательными людьми и многого добиться. Спасибо Эрин, Мэтту Ликоку (Matt Leacock), Шанель Уиллер (Chanel Wheeler) за основание Yahoo! Design Pattern Library. Спасибо Ларри Теслеру (Larry Tesler) и Эрин за то, что дали мне возможность возглавить и пропагандировать запуск общедоступной Yahoo! Design Pattern Library. Именно руководя работой с шаблонами проектирования, я вынашивал большинство идей, содержащихся в этой книге. И особенно благодарен многим талантливым дизайнерам и разработчикам за полезные комментарии и вдохновлявшее меня мастерство.

Выражаю благодарность команде YUI, особенно Нейту Кохли (Nate Koechley) и Эрику Мираглия (Eric Miraglia), в частности за формулирование «интересных моментов» (стр. 47) и за возможность привязки шаблонов к реальному коду. Спасибо моим коллегам Дугласу Крокфорду (Douglas Crockford), Иену Лэмбу (Iain Lamb), Жульену Лекомте (Julien

Lecomte), Адаму Платти (Adam Platti) и моему хорошему другу Даррену Джеймсу (Darren James) за поддержку во время написания книги. Спасибо многим талантливым дизайнерам, с которыми мне посчастливилось сотрудничать и чьи идеи также отражены на страницах книги: Карону Веберу (Karon Weber), Саманте Триподи (Samantha Tripodi), Эриксону де Жесусу (Ericson deJesus), Мике Лаакеру (Micah Laaker), Люку Вроблевски, Тому Чи (Tom Chi), Лукасу Петтинати (Lucas Pettinati), Кевину Ченгу (Kevin Cheng), Кэтлин Уоткинс (Kathleen Watkins), Кирстен Ламмердинг (Kiersten Lammerding), Аннетт Леонг (Annette Leong), Лансу Нишире (Lance Nishihira) и многим другим.

За пределами Yahoo! мои идеи получили поддержку и развитие благодаря Дэну Сафферу (Dan Saffer) и Райану Фрейтасу (Ryan Freitas) из Adaptive Path, Азе Раскину (Aza Raskin) и Скотту Робинсу (Scott Robbins) из Humanized, Питеру Моерхольцу (Peter Moerholz) и Дэвиду Верба (David Verba) из Adaptive Path. Хотелось бы поблагодарить участников сообщества, посвященного шаблонам. Спасибо Дженифер Тидвелл (Jenifer Tidwell) за то, что направила меня именно в эту область, Мартижину ван Уэлли (Martijn van Welie) – за составление отличной библиотеки шаблонов, Джеймсу Рефеллу (James Refell) и Люку Вроблевски (Luke Wroblewski) – за работу над шаблонами в eBay, Кристиану Крамлишу (Christian Crumlish), нынешнему руководителю разработки шаблонов в Yahoo!, – за его интересные идеи. Спасибо Джесси Джеймсу Гарретту (Jesse James Garrett), который не только дал имя технологии Ajax, но и пригласил и сопровождал меня на первый посвященный ей саммит. Преподавание на Designing for Ajax Workshops убедило меня в необходимости написания этой книги и позволило протестировать материал с реальной аудиторией.

Спасибо многим компаниям и организаторам конференций, пригласившим меня выступить перед публикой. Обсуждение данного материала с тысячами слушателей стало бесценным опытом, позволившим выявить наиболее актуальные и животрепещущие темы для дизайнеров и разработчиков. Благодарю этих людей (перечисляю их в произвольном порядке с указанием компании, в которую меня пригласили для выступления): Джаред Спул (Jared Spool) (UIE), Бен Галбрейс (Ben Galbraith) и Дион Элмер (Dion Almer) (Ajaxian/Ajax Experience), Кэтрин Маккиннон (Kathryn McKinnon) (Adobe), Джереми Гиллан (Jeremy Geelan) (SysCon), Рашими Синха (Rashmi Sinha) (BayCHI/Slideshare), Аарон Ньютон (Aaron Newton) (CNET), Брайан Кромпи (Brian Kromrey) (курсы Yahoo! UED), Люк Ковальски (Luke Kowalski) (Oracle), Шон Кейн (Sean Kane) (Netflix), Решма Кумар (Reshma Kumar) (Silicon Valley Web Guild), Эммануэль Леви-Валенси (Emmanuel Levi-Valensi) (People in Action), Бруно Фигуиредо (Bruno Figueiredo) (SHiFT), Мэттью Мороз (Matthew Moroz) (Avenue A Razorfish), Питер Боерсма (Peter Boersma) (SIGCHI.NL), Кит Сиборг (Kit Seeborg) (Web-Visions), Уилл Тшуми (Will Tschumy) (Microsoft), Боб Бэксли (Bob Baxley) (Yahoo!), Джей Циммерман (Jay Zimmerman) (Rich WebExperience), Дэйв Верба (Dave Verba)

(UX Week). Также хочу поблагодарить следующие компании и конференции: Web Builder 2.0, eBig, PayPal, eBay, CSU Hayward, City College San Francisco, Apple и многие другие.

Выражаю глубокую признательность Sabre Airline Solutions, в особенности Брэду Йенсену (Brad Jensen), давшему мне замечательную возможность заниматься проектированием пользовательских интерфейсов в своей организации, а также Дэвиду Эндикотту (David Endicott) и Деймону Хугленду (Damon Hougland), поддержавшим мою идею поделиться своими мыслями с широкой публикой. И всей команде, помогавшей нам с Терезой воплотить свои замыслы. Многие шаблоны, описанные в этой книге, появились благодаря практической разработке приложений в этой компании.

Наконец, спасибо Netflix, моему нынешнему месту работы и, несомненно, одному из лучших мест работы в мире. Спасибо за то, что поддержали мои начинания и научили проектировать и создавать по-настоящему удобные для пользователя продукты.

Благодарности Терезы

Мне хотелось бы выразить свою признательность следующим людям:

Аарону Эрлофу (Aaron Arlof), создавшему иллюстрации для нашей книги. Они в точности отражают все шесть принципов проектирования.

Брэду Йенсену (Brad Jensen), моему вице-президенту в Sabre Airline Solutions, направившему меня первым делом поговорить с Биллом. Без наставлений Билла я бы здесь ничего не добилась.

Деймону Хугленду (Damon Hougland), который помог нам с Биллом создать команду по вопросам взаимодействия с пользователем в Sabre.

Джо Балдерасу (Jo Balderas), благодаря которому я смогла сделать первые шаги в программировании.

Даррену Джеймсу (Darren James), сделавшему из меня программиста.

Всем моим клиентам, принимавшим участие в презентациях, с живым интересом изучавшим шаблоны и принципы проектирования пользовательских интерфейсов, в частности Стивену Смиту (Steven Smith), Дэйву Уилби (Dave Wilby), Сури Бала (Suri Bala), Джеффу Комо (Jeff Como) и Сету Элсбури (Seth Alsbury), которые доверили мне проектирование корпоративных приложений на заре революции насыщенных веб-приложений. Особая благодарность моим нынешним коллегам Скотту Бомсу (Scott Boms) из Wishingline, Паоло Виера (Paulo Viera), Джессике Дуглас (Jessica Douglas), Алану Баумгартену (Alan Baumgarten) и Робу Джонсу (Rob Jones).

Наконец, самое главное: я хочу поблагодарить своего мужа за постоянную поддержку и родителей за одобрительное отношение к моей работе. И спасибо моему сыну Аарону за то, что позволил мне провести столько времени за компьютером.

Принцип I

Прямые действия

Во время недавней поездки в Биг Сюр (Калифорния) я сфотографировал несколько живописных видов, открывавшихся с шоссе № 1. Загрузив полученные снимки на фотосервис Flickr, я решил дать им описательные названия. К примеру, «Вид побережья с моста Биксби» – гораздо понятнее, чем «IMG_6420.jpg».

Обычно в Сети для такого переименования требуется нажать кнопку Edit на странице с фотографией. При этом открывается отдельная страница с названием, описанием и прочими сведениями о фотографии, которые можно изменить. При нажатии кнопки Save введенные данные сохраняются и происходит возврат к странице с фотографией, где она отображается уже с новым названием. Данный метод схематично изображен на рис. I.1.¹

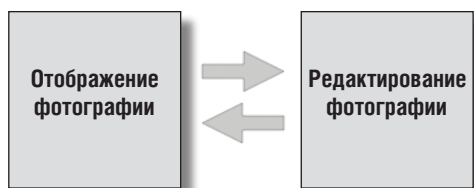


Рис. I.1. Как правило, для редактирования информации пользователь должен перейти на другую страницу

Отредактировать название фотографии в Flickr можно и так. Однако обычно Flickr предлагает и более прямой способ редактирования. Как показано на рис. I.2, при щелчке мышью на надписи IMG_6420.jpg во-

¹ Напоминаем, что все *цветные* версии иллюстраций можно скачать единым архивом по адресу <http://www.symbol.ru/library/design-web-interface/all.zip> или поглавно (<http://www.symbol.ru/library/design-web-interface/01.zip>, ...02.zip и т. д.).

круг нее появляются элементы управления для редактирования. Таким образом, переход в режим редактирования осуществляется одним щелчком мыши.

Такой прямой способ редактирования гораздо удобнее, поскольку позволяет не покидать текущую страницу. В этом есть еще одно преимущество: благодаря легкости изменения названия, описания, меток и иной информации пользователь вводит больше данных, что облегчает поиск и просмотр фотографий.

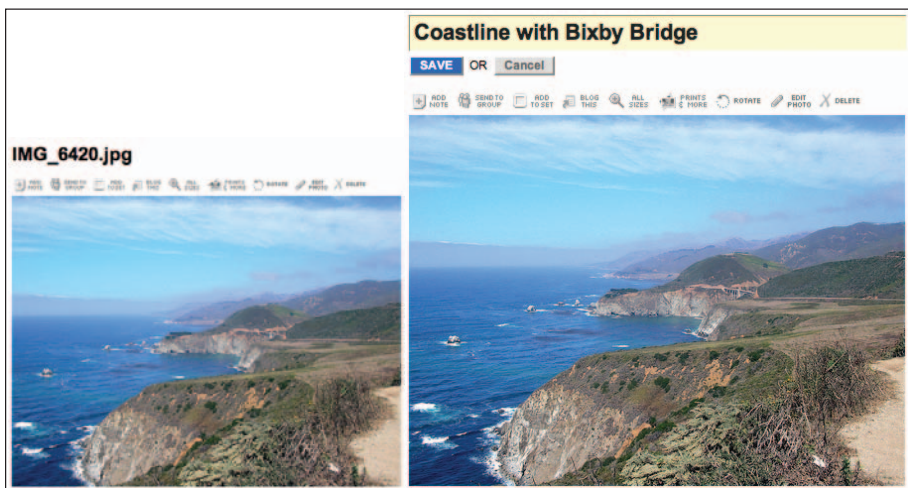


Рис. I.2. Flickr: щелкните на названии фотографии, чтобы изменить его

Действовать напрямую

Основной задачей первых веб-сайтов было отображение информации и обеспечение легкости ориентирования в ней. О большой интерактивности говорить не приходилось. Ранние версии языка HTML не предоставляли возможность создания форм для ввода информации пользователем. И даже когда ввод и вывод данных стали стандартом для веб-сайтов, Сеть в основном предлагала информацию для чтения, и лишь изредка у пользователя появлялась возможность ввести свои данные. Такое разделение было обусловлено не дизайном, а технологическими ограничениями.

Алан Купер так описал эту ложную дихотомию в книге «About Face 3: The Essentials of Interaction Design»¹:

¹ Купер А., Рейман Р., Кронин Д. «Алан Купер об интерфейсе. Основы проектирования взаимодействия». – Пер. с англ. – СПб.: Символ-Плюс, 2009.

...во многих программах функции ввода и вывода трактуются как два разных процесса и расположены в совершенно разных местах. Если посмотреть на это глазами пользователя, ... никакой разницы нет.

Купер подводит итог сказанному простым правилом: «Где есть вывод, должен быть и ввод». В более широком смысле интерфейс должен непосредственно откликаться на действия пользователя – то есть важны *прямые действия*.¹

Чтобы более наглядно проиллюстрировать это правило, мы рассмотрим несколько общих шаблонов взаимодействия, которые можно использовать при создании собственных интерфейсов. Им посвящены следующие три главы:

Глава 1 «Непосредственное редактирование»

Непосредственное редактирование содержимого страницы (контента).

Глава 2 «Перетаскивание»

Перемещение объектов с помощью мыши.

Глава 3 «Непосредственное выделение»

Совершение различных операций с непосредственным выделением объектов.

¹ Это повторяет принцип «Прямого управления», выведенный Беном Шайдерманом (Ben Scheiderman) в статье «Direct manipulation: a step beyond programming languages», *IEEE Computer* 16[8] [August 1983], с. 57–69.

1

Непосредственное редактирование¹

Традиционно на веб-страницах информация представлялась только в режиме отображения. Для ее редактирования была предусмотрена специальная форма с несколькими полями и кнопкой для сохранения изменений. Принцип прямых действий предполагает возможность редактирования контента непосредственно на странице.

В данной главе мы рассмотрим серию шаблонов проектирования² для непосредственного редактирования контента веб-страницы. Ниже приведены шесть шаблонов, соответствующих наиболее распространенным методам изменения данных прямо на странице:

Непосредственное редактирование однострочного текста

Изменение текстовой строки.

Непосредственное редактирование многострочного текста

Изменение более сложных данных в специальной форме.

Редактирование в оверлее

Изменение данных в дополнительном оверлейном (перекрывающем) слое.

Редактирование таблицы

Изменение содержимого ячеек таблицы.

¹ Цветные иллюстрации к этой главе можно скачать по адресу <http://www.symbol.ru/library/design-web-interface/01.zip>.

² Мы используем термин «шаблоны проектирования», поскольку речь идет о типичных решениях типичных проблем. Впервые он был употреблен в книге Кристофера Александера (Christopher Alexander) «A Pattern Language» (Oxford University Press). Множество статей о шаблонах проектирования (как моих – Билла, так и других авторов) можно найти на сайте <http://www.lukew.com/ff/entry.asp?347>.

Групповое редактирование

Прямое изменение группы элементов.

Конфигурирование модуля

Изменение настроек непосредственно на странице.

Наиболее прямой метод редактирования *на странице* подразумевает редактирование в контексте текущей страницы: во-первых, не уходим с нее и, во-вторых, редактируем собственно на ней.

Преимущество **непосредственного редактирования** в том, что пользователю доступен контекст – ведь в процессе редактирования нередко возникает необходимость обращения к прочей информации, представленной на странице. Например, как сказано в следующем разделе, пользователю удобно изменять название фотографии, глядя на нее.

Это также полезно при редактировании отдельного элемента из набора. К примеру, система комментариев Disqus предлагает возможность непосредственного редактирования комментариев (рис. 1.1). После отправки комментария и до того, как кто-либо ответит на него, доступна ссылка для его редактирования. Редактирование осуществляется в контексте остальных комментариев, видимых на странице.

Примечание

Если для редактирования необходим доступ к контексту страницы, предпочтительнее непосредственное редактирование.

Первые два шаблона, **непосредственное редактирование однострочного текста** и **непосредственное редактирование многострочного текста**, описывают методы добавления на страницу возможности прямого редактирования текста.

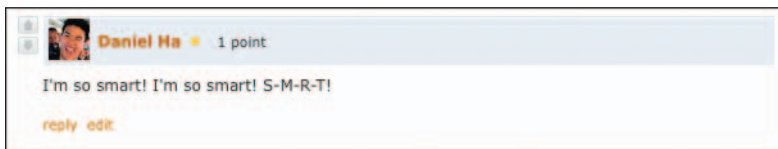


Рис. 1.1. Disqus предоставляет возможность редактирования комментариев в контекстном окружении

Непосредственное редактирование однострочного текста

Простейший вид редактирования на странице – **непосредственное редактирование однострочного текста**. При этом редактирование выполняется непосредственно на текущей странице, а не в отдельном окне или на другой странице. Flickr можно назвать классическим примером этого метода (рис. 1.2).

Обычный режим отображения

Разработчики Flickr не стали добавлять к названию элементы, указывающие на возможность его редактирования. Интерфейс не загроможден инструментами редактирования и заголовок выглядит совершенно обычно.

Такое решение чревато некоторыми проблемами. Пользователь может не сразу догадаться о возможности редактирования. В качестве альтернативного решения в данном случае можно поместить рядом с заголовком ссылку «edit», щелчок на которой активирует соответствующий режим.

Приглашение к редактированию

При наведении указателя мыши на название оно подсвечивается желтым. Появляется всплывающая подсказка — приглашение для пользователя «Click to edit» (Щелкните для редактирования).

Такие подсказки предназначены для того, чтобы вывести пользователя на новый уровень взаимодействия с приложением (от наведения указателя мыши до щелчка). Заметим, что такие подсказки вообще полезны. Создатели Flickr не сомневались, что пользователь наведет указатель мыши на название (это же его собственное фото).

Редактирование

Если пользователь щелкнул на названии, оно переходит в режим редактирования: название оказывается непосредственно в поле редактирования.

Кнопки «Save» и «Cancel» создают знакомый пользовательский интерфейс (форма для ввода данных), и у него не возникает сомнений в том, что он редактирует название.

Недостаток данного подхода: при переходе в режим редактирования сама фотография смещается вниз, чтобы освободить место для дополнительных элементов интерфейса.

Завершение редактирования

Есть много способов показать пользователю, что выполняется сохранение данных. В данном примере текст названия временно заменяется словом «saving...». По завершении операции вместо него выводится новое название в режиме отображения.

Можно предложить и другой вариант, например во время сохранения отображать индикатор выполнения.

Рис. 1.2. Flickr: достаточно прямолинейный способ непосредственного редактирования названия фотографии

Соображения

Описанный процесс достаточно прост. Чтобы отредактировать название, щелкните на нем. По окончании редактирования нажмите кнопку Save, и на прежнем месте появится новое название. Flickr стал одним из первых сайтов, применивших этот метод взаимодействия с пользователем. За последние несколько лет он практически не изменился, что служит лучшим доказательством его эффективности.

Однако и здесь есть над чем подумать.

Заметность

Легко ли пользователю обнаружить данную функцию? В рассмотренном примере можно выделить несколько указаний пользователю на возможность редактирования:

- Всплывающая текстовая подсказка (Click to edit)
- Желтая подсветка области с возможностью редактирования
- Превращение указателя мыши в текстовый курсор (в виде символа I)

Однако все эти указания появляются только *после* того, как пользователь наведет указатель мыши на название. Вероятность заметить функцию редактирования зависит от того, наведет ли он указатель мыши на название и обратит ли внимание на все эти подсказки.¹

Чтобы возможность редактирования стала более очевидной, можно поместить на этой же странице несколько дополнительных указаний. К примеру, рядом с названием можно разместить ссылку Edit, щелчок на которой активирует режим редактирования. Если такая ссылка отображается постоянно, возможность редактирования намного очевиднее.

При этом нужно следить за уровнем визуального шума, допускаемого на странице. Каждая дополнительная ссылка или кнопка усложняет восприятие страницы, а ее перегруженность предлагаемыми возможностями может привести к тому, что пользователь просто не заметит данную функцию.

Примечание

Если отображение информации важнее возможности ее редактирования, последнюю можно скрыть до тех пор, пока пользователь не начнет взаимодействовать с контентом.

¹ При запуске Yahoo! Design Pattern Library (<http://developer.yahoo.com/upatterns/>) данный шаблон не был включен в первоначальный набор шаблонов из-за внутренней дискуссии об очевидности функции редактирования. Оказалось, что один из критиков, старший дизайнер и постоянный пользователь сервиса Flickr, обнаружил ее совсем недавно. В результате мы исключили данный шаблон из общедоступной версии.

Сервис Yahoo! Photos¹ задействует этот подход для редактирования названий (рис. 1.3). Отображение ссылки для редактирования каждой фотографии в группе визуально нагружало бы страницу. Поэтому все названия отображаются без дополнительных элементов. При наведении указателя мыши на название его фон подсвечивается, а при щелчке открывается поле для редактирования. Щелчок мышью вне данного поля или переход к следующему названию автоматически сохраняет внесенные изменения. Это помогает уменьшить визуальную загроможденность как страницы, так и самого процесса редактирования. В результате получаем визуально спокойную фотогалерею.



Рис. 1.3. Yahoo! Photos: метод редактирования названий фотографий сводит визуальный шум к минимуму – при редактировании просто появляется соответствующая подсветка

Доступность

Другим существенным недостатком **непосредственного редактирования** является его недоступность для пользователей с ограниченными возможностями, а их гораздо больше, чем принято думать. Людям с физическими недостатками, различными заболеваниями, например имеющим плохое зрение, необходимы специальные вспомогательные средства.

Как правило, такие средства анализируют разметку страницы, отыскивая контент, якоря, всплывающие текстовые подсказки с названием изображения и прочие структурные элементы страницы. Если применяемый метод **непосредственного редактирования** не предполагает некоторую разметку в коде страницы (вроде явной ссылки на режим редактирования), то вспомогательные средства не смогут обнаружить эту функцию.

Если полагаться только на пользовательские манипуляции с мышью, часть посетителей сайта не смогут воспользоваться функцией прямого редактирования. Как уже говорилось, явная ссылка на режим редактирования делает эту функцию более заметной (что было наглядно показано на рис. 1.1). В качестве «побочного эффекта» она также сделает данную функцию более доступной для пользователей с ограниченными возможностями.

¹ В 2007 году сервис Yahoo! Photos был заменен на Flickr.

Примечание

Добавление альтернативной возможности редактирования на отдельной странице сделает функцию более доступной.

Естественно, возникает необходимость выбора между **непосредственным редактированием** и менее прямым, но более доступным для всех категорий пользователей методом. Это противоречие можно разрешить, обеспечив обе возможности в одном интерфейсе. Реальный пример такого решения дает Flickr, где предлагается дополнительная возможность редактирования данных (рис. 1.4).

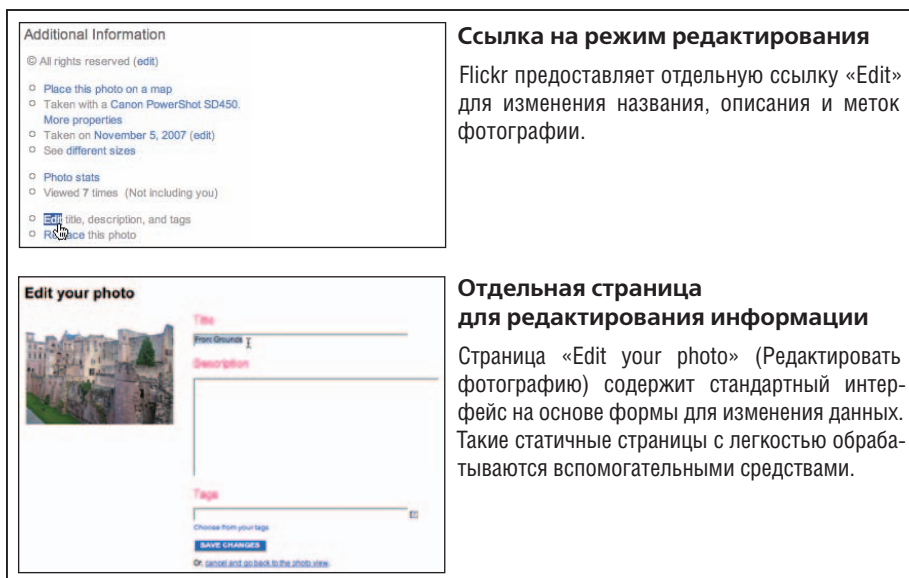


Рис. 1.4. Flickr: пользователь также может изменить название, описание и метки фотографии на отдельной странице

Непосредственное редактирование многострочного текста

В предыдущем примере мы редактировали одну строку. А если строк несколько или данные об элементе представлены в более сложной форме, чем однострочный текст, и нужно предоставить пользователю возможность прямого редактирования всех значений?

Шаблон **непосредственное редактирование многострочного текста** описывает эту возможность.

Приложение Backpack компании 37 Signals использует данный шаблон для редактирования заметки, состоящей из названия и тела (рис. 1.5). Для лучшей читаемости названия оно отображается как заголовок,

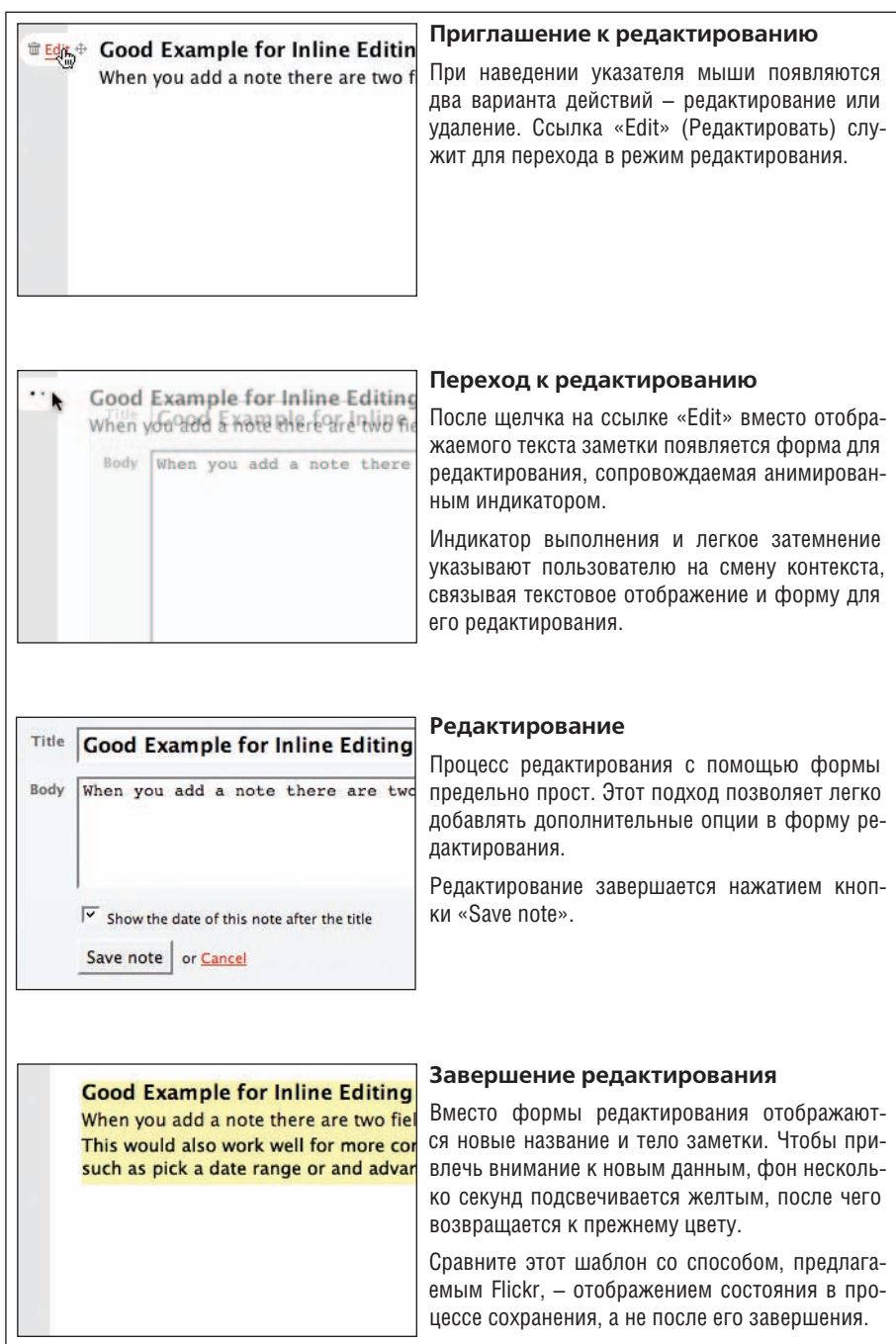


Рис. 1.5. Васкраск: для редактирования заголовка и тела заметки отображается форма с несколькими полями

а тело – как обычный текст. Во время редактирования заголовков и текст выводятся в полях ввода с приглашениями-подсказками.

Соображения

При непосредственном редактировании однострочного текста разница между режимами отображения и редактирования минимальна, поэтому переход от одного к другому не так бросается в глаза. При редактировании нескольких полей гораздо больше видимых различий между элементами в режимах отображения и редактирования.

Читаемость или возможность редактирования?

В режиме отображения наибольшее значение имеет читаемость данных. А лучший способ предоставить возможность редактирования – обычная форма для ввода данных. Пользователю могут пригодиться следующие опции:

- Инструменты редактирования
- Приглашения-подсказки
- Текстовая справка по пользовательскому вводу
- Обработка ошибок
- Вспомогательные инструменты ввода (например, всплывающий календарь или раскрывающийся список для выбора)
- Стили режима редактирования (например, объемные «вдавленные» поля для редактирования)

Режим редактирования отличается по размеру и расположению элементов, а также по их количеству и типу. Следовательно, переход между режимами, скорее всего, будет достаточно резким.

В приведенном примере форма редактирования занимает гораздо больше места, чем отображаемая заметка.

Смешивание режимов отображения и редактирования

В идеале хотелось бы достичь безболезненного переключения между режимами. Добавление режима редактирования прямо на страницу сильно повлияет на остальное ее содержимое. Один из способов смягчения перехода – умеренное использование анимации. В Vascrack анимационный эффект состоит в постепенном исчезновении отображаемой заметки и одновременном проявлении режима редактирования (рис. 1.5).

Другой способ решения проблемы – отвести под оба режима одинаковое пространство. Пользователь сервиса Yahoo! 360 может настроить свое статусное сообщение. Его текущий статус отображается в выноске-облаке, как реплики героев комиксов. Визуально оно выглядит единым, но редактировать можно три параметра – стиль облака, текст статуса и ссылку на веб-страницу, переход на которую выполняется щелчком по облаку. На рис. 1.6 показан вид облака перед редактированием.

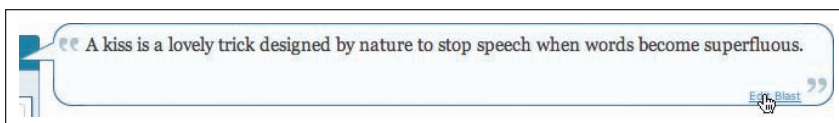


Рис. 1.6. Yahoo! 360: ссылка Edit Blast указывает пользователю на возможность редактирования

На рис. 1.7 показано, как облако выглядит при редактировании. Обратите внимание: форма редактирования спроектирована так, чтобы оба режима (отображения и редактирования) занимали равное пространство.



Рис.1.7. В сервисе Yahoo! 360 функция редактирования реализована в виде облака; различия между режимами отображения и редактирования сведены к минимуму

Соответствие размеров вовсе не случайно. Дизайнеры страницы старались сохранить ее визуальную целостность при переходе в режим редактирования, лишь немного увеличив занимаемое место по сравнению с режимом отображения и при этом уместив все необходимые инструменты редактирования внутри облака.

WYSIWYG¹

Если режимы редактирования и отображения задействуют совершенно разные области страницы, пользователю, возможно, будет трудно четко представить себе, как будут выглядеть внесенные изменения. В Yahoo! 360 есть возможность увидеть, как будет выглядеть облако сразу после редактирования его стиля. Переход от стиля облака quote (цитата) к thought (мысль) визуально заметен уже в режиме редактирования (рис. 1.8), что было бы невозможно при использовании отдельной формы редактирования.



Рис. 1.8. Yahoo! 360: обновленный вид облака мгновенно отображается прямо в режиме редактирования

¹ WYSIWYG (англ. What You See Is What You Get – «что видишь на экране, то и получишь») – интерфейс, отображающий контент в режиме редактирования в виде, максимально приближенном к тому, как он будет отображаться по окончании редактирования.

Редактирование в оверлее

В предыдущих двух шаблонах использовался метод непосредственного редактирования прямо на странице. Такое редактирование выполняется по месту, среди остальных элементов страницы.

Редактирование в оверлее выполняется в дополнительном слое, расположенном поверх контента страницы. Этот способ не предполагает перехода на другую страницу, но редактирование происходит и не прямо на странице. Для редактирования используется облегченный всплывающий слой (подобный диалоговому окну).

Есть ряд причин, побуждающих разработчиков предпочесть оверлей (overlay) редактированию прямо на странице.

Иногда сложный комплекс инструментов для редактирования невозможно разместить на самой странице. Если область редактирования достаточно велика, то ее размещение на странице приведет к существенному сдвигу контента, отвлекая внимание пользователя. Переход между режимами, сопровождаемый различными перестановками, также нежелателен.

В определенных случаях приходится нарушать структуру контента страницы, если редактируемая информация имеет самостоятельное значение. Дополнительный слой дает пользователю четко очерченное пространство для редактирования, не загромождая страницу.¹

Примечание

Редактирование в оверлее – прекрасное решение в том случае, если область редактирования занимает существенную часть экрана и при этом у пользователя нет необходимости видеть содержимое всей страницы в процессе редактирования.

Yahoo! Trip Planner – сетевой сервис для совместного планирования путешествий. План путешествий может содержать определенные события, привязанные к конкретным датам. В таком случае элемент события содержит дату, которую можно редактировать в оверлее (рис. 1.9).

Соображения

Дата в формате «Sun Jun 4 12:00am–Mon Jun 5 12:00am» воспринимается гораздо легче, чем в виде, удобном для редактирования (рис. 1.10). Редактирование в специальном окне исключает появление ошибок при вводе даты начала и окончания определенного события.

¹ В прошлом вспомогательное окно открывалось в новом окне браузера. Всплывающее окно позволяет разместить дополнительный контент поверх страницы. В результате не возникает ощущения перегруженности страницы. Более подробная информация приведена в главе 5.

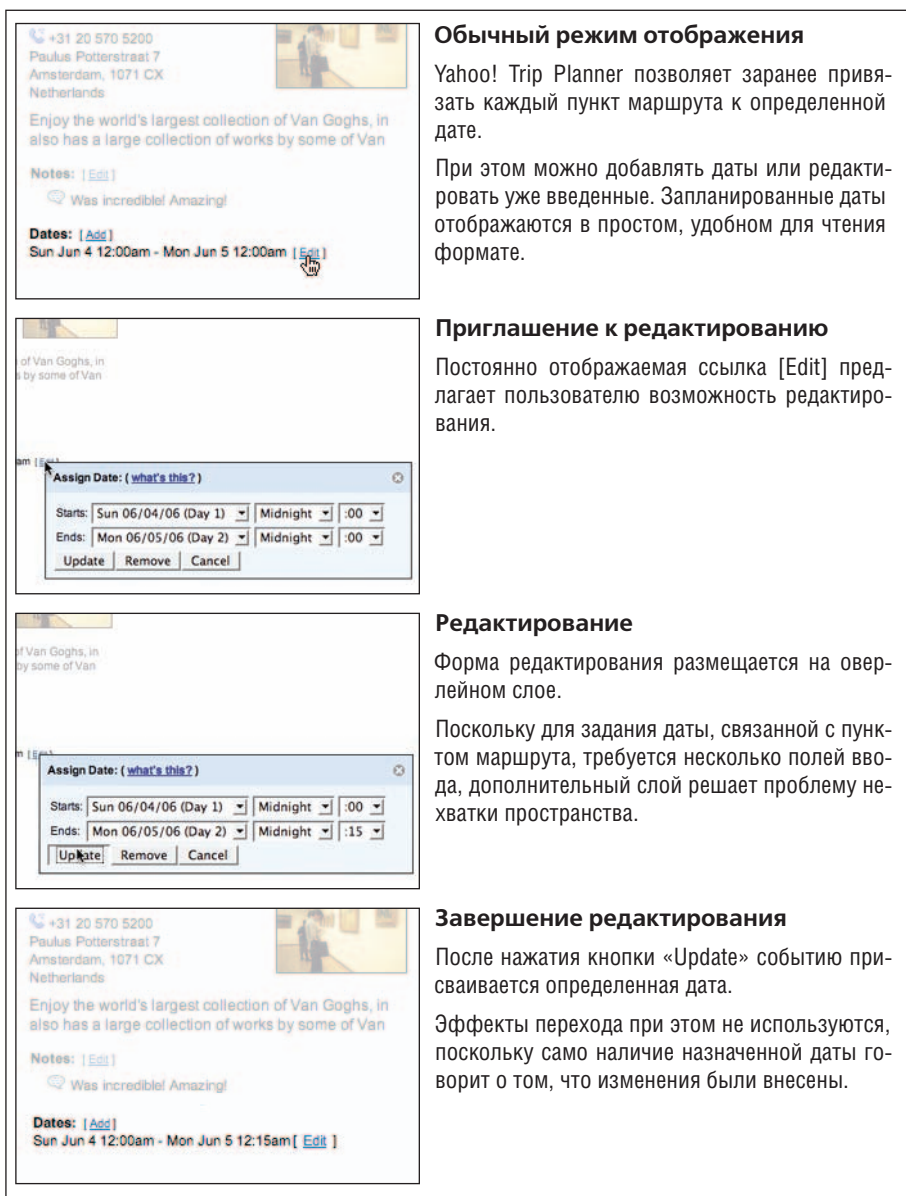


Рис. 1.9. Yahoo! Trip Planner: изменение даты события в оверлее

Поскольку диапазон дат известен, для выбора даты и времени начала и окончания события Trip Planner использует набор раскрывающихся списков.

Следует отметить, что несколько раскрывающихся списков для выбора часов и минут – не лучший вариант. Более удачный пример возмож-