

ДЖЕФ РАСКИН

# ИНТЕРФЕЙС

НОВЫЕ НАПРАВЛЕНИЯ В ПРОЕКТИРОВАНИИ КОМПЬЮТЕРНЫХ СИСТЕМ



По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 5-93286-030-8, название «Интерфейс: новые направления в проектировании компьютерных систем» – покупка в Интернет-магазине «Books.Ru – Книги России». Если Вы получили данный файл каким-либо другим образом, Вы нарушили международное законодательство и законодательство Российской Федерации об охране авторского права. Вам необходимо удалить данный файл, а также сообщить издательству «Символ-Плюс» ([piracy@symbol.ru](mailto:piracy@symbol.ru)), где именно Вы получили данный файл.

# THE HUMANE INTERFACE

NEW DIRECTIONS  
FOR DESIGNING INTERACTIVE SYSTEMS

JEF RASKIN

# ИНТЕРФЕЙС

НОВЫЕ НАПРАВЛЕНИЯ В ПРОЕКТИРОВАНИИ  
КОМПЬЮТЕРНЫХ СИСТЕМ

ДЖЕФ РАСКИН



---

*Санкт-Петербург-Москва*  
*2007*

Джеф Раскин

## Интерфейс: новые направления в проектировании компьютерных систем

Перевод Ю. Асотова

Главный редактор  
Зав. редакцией  
Научный редактор  
Редактор  
Художник  
Корректурa  
Верстка

*А. Галунов  
Н. Макарова  
А. Михайлов  
М. Буйневич  
С. Борин  
С. Беляева  
Н. Грищенко*

*Раскин Д.*

Интерфейс: новые направления в проектировании компьютерных систем. – Пер. с англ. – СПб: Символ-Плюс, 2007. – 272 с., ил.  
ISBN 5-93286-030-8

Это уникальное руководство по разработке интерактивных систем, отражающее опыт и взгляды Джефа Раскина, создателя проекта Apple Macintosh, должен прочитать каждый, кто задумывается об интерфейсе.

Сейчас много говорят об эффективности современных подходов к разработке интерфейсов. Раскин же демонстрирует, что многие из них ведут в тупик, и для создания компьютеров, с которыми было бы проще работать, требуются совершенно новые принципы разработки. Он объясняет, как осуществить эти необходимые сегодня изменения, и высказывает нестандартные идеи, демонстрируя дальновидность и способность к практическому взгляду на вещи. Эта книга, рассказывающая о научном подходе к разработке интерфейсов, может быть полезна как для создателей программного обеспечения, так и для руководителей проектов.

**ISBN-13: 978-5-93286-030-4**

**ISBN-10: 5-93286-030-8**

**ISBN 0-201-37937-6 (англ)**

© Издательство Символ-Плюс, 2003

**The Humane Interface. New Directions for Designing Interactive Systems by Jef Raskin, Copyright © 2000, All Rights Reserved. Published by arrangement with the original publisher, Pearson Education, Inc., publishing as ADDISON WESLEY LONGMAN.**

Интерфейс: новые направления в проектировании компьютерных систем. Джеф Раскин © 2000, все права защищены. Публикуется по соглашению с оригинальным издателем Pearson Education, Inc. (ADDISON WESLEY LONGMAN).

Все права на данное издание защищены Законодательством РФ, включая право на полное или частичное воспроизведение в любой форме. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих фирм.

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7,  
тел. (812) 324-5353, edit@symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Налоговая льгота – общероссийский классификатор продукции  
ОК 005-93, том 2; 953000 – книги и брошюры.

Подписано в печать 22.05.2007. Формат 70x100<sup>1</sup>/<sub>16</sub>. Печать офсетная.

Объем 17 печ. л. Доп. тираж 1000 экз. Заказ N

Отпечатано с готовых диапозитивов в ГУП «Типография «Наука»  
199034, Санкт-Петербург, 9 линия, 12.

*Мы угнетены нашими электронными рабами.  
Эта книга посвящается нашему освободителю.*



# Оглавление

<b>Предисловие</b> .....	9
<b>Благодарности</b> .....	11
<b>Введение. Важность основ</b> .....	15
<b>1. Предпосылки</b> .....	19
1.1. Определение интерфейса .....	20
1.2. Простое должно оставаться простым .....	20
1.3. Ориентация на человека и на пользователя .....	21
1.4. Инструменты, которые препятствуют новым идеям .....	22
1.5. Разработка интерфейса как часть общего цикла разработки .....	23
1.6. Определение человекоориентированного интерфейса .....	25
<b>2. Когнетика и локус внимания</b> .....	27
2.1. Эргономика и когнетика: что мы можем и чего не можем .....	27
2.2. Когнитивное сознательное и когнитивное бессознательное .....	29
2.3. Локус внимания .....	35
<b>3. Значения, режимы, монотонность и мифы</b> .....	53
3.1. Терминология и условные обозначения .....	53
3.2. Режимы .....	57
3.3. Модели «существительное-глагол» и «глагол-существительное» .....	81
3.4. Видимость и состоятельность .....	85
3.5. Монотонность .....	89
3.6. Миф о дихотомии «новичок–эксперт» .....	92
<b>4. Квантификация</b> .....	97
4.1. Количественный анализ интерфейса .....	97
4.2. Модель скорости печати GOMS .....	98
4.3. Измерение эффективности интерфейса .....	109
4.4. Закон Фитса и закон Хика .....	119

<b>5. Унификация</b> .....	125
5.1. Унификация и элементарные действия .....	127
5.2. Каталог элементарных действий .....	129
5.3. Имена файлов и файловые структуры .....	145
5.4. Поиск строк и механизмы поиска .....	152
5.5. Форма курсора и методы выделения .....	162
5.6. Позиция курсора и клавиша <LEAP> .....	166
5.7. Ликвидация приложений .....	169
5.8. Команды и трансформаторы .....	173
<b>6. Навигация и другие аспекты человекоориентированных интерфейсов</b> .....	181
6.1. Интуитивные и естественные интерфейсы .....	182
6.2. Улучшенная навигация: ZoomWorld .....	184
6.3. Пиктограммы .....	202
6.4. Способы и средства помощи в человекоориентированных интерфейсах .....	209
6.5. Письмо от одного пользователя .....	224
<b>7. Проблемы за пределами пользовательского интерфейса</b> .....	229
7.1. Более человекоориентированные среды программирования .....	230
7.2. Режимы и кабели .....	234
7.3. Этика и управление разработкой интерфейсов .....	237
<b>Заключение</b> .....	244
<b>А. Однокнопочная мышь: история и будущее</b> .....	246
<b>В. Теория работы интерфейса для SwyftCard</b> .....	250
<b>Библиография</b> .....	253
<b>Алфавитный указатель</b> .....	257

# Предисловие

*В любом деле, где мы используем компьютеры, какую-то долю времени мы всегда тратим на то, чтобы наладить их работу. Но если бы мой садовник тратил столько же времени на починку садового совка, сколько у нас уходит на мороку с нашими компьютерами, то я лучше купил бы ему другой, хороший совок. По крайней мере, вы можете купить хороший совок.*

*Эразмус Смамс*

Создание интерфейса во многом напоминает строительство дома. Никакая внутренняя или внешняя отделка не спасет сооружение, если его фундамент не положен надлежащим образом. В книге, которую вы держите в руках, рассматриваются когнитивные принципы, лежащие в основе процесса взаимодействия между машиной и человеком и позволяющие ответить на главный вопрос: почему один интерфейс является удачным, а другой – нет. Существующие сегодня графические пользовательские интерфейсы, например те, которые применяются в операционных системах Windows и Macintosh и архитектура которых основана на самой операционной системе и прикладных программах, являются изначально порочными. Для того чтобы сделать компьютеры более удобными и эффективными в использовании, требуется совершенно иной подход. В этой книге описываются некоторые принципиальные недостатки пользовательских интерфейсов и предлагаются пути их преодоления.

Приемы, описанные здесь, могут быть применены к широкому кругу продуктов, включая веб-сайты, прикладное программное обеспечение, ноутбуки и другие устройства для работы с информацией, а также операционные системы. Однако эту книгу нельзя рассматривать как обзор вопросов, связанных с разработкой интерфейсов «человек-машина». Скорее, она открывает новые направления в этой сфере и в то же время затрагивает уже существующие области разработки интерфейсов, на основе которых можно создавать что-то новое.

Если мы собираемся преодолевать проблемы, присущие сегодняшним «человеко-машинным» интерфейсам, нам необходимо усвоить все то, что здесь изложено; однако этого *недостаточно*. Многие важные аспекты разработки интерфейсов не упоминаются в этой книге, поскольку они хорошо освещены в других работах. Данная книга была задумана как дополнение к существующим (или как введение к будущим) подходам к проблемам разработки интерфейсов.

Книга рассчитана на следующий круг читателей:

- Веб-дизайнеров и менеджеров, которые хотят, чтобы их сайты обладали особой легкостью в использовании, что привлекает аудиторию и помогает пользователям находить требуемую им информацию и покупать необходимые товары
- Разработчиков новых программных продуктов и менеджеров, ответственных за их выпуск, которые желают научиться создавать сайты или продукты, отличающиеся простотой в изучении и использовании, а также первоклассным набором дополнительных компонентов, что в совокупности делает эти продукты привлекательными для потребителей
- Менеджеров компаний, которые настаивают на том, чтобы их продукты не требовали больших затрат на обслуживание и не вынуждали пользователей часто обращаться за помощью к службам поддержки
- Программистов, которые разрабатывают интерфейсы (а кто из них сегодня не занимается этим?) и хотят узнать, как сделать свой труд более ценным
- Менеджеров по информационным технологиям, которым необходимо иметь представление о том, какие свойства интерфейса могут снизить затраты на его изучение и обеспечить его продуктивность
- Потребителей, которые желают знать, чего ждать от разработчиков в будущем и что является неверным в том подходе, который используется сегодня при разработке программного обеспечения
- Студентов, изучающих вычислительную технику и когнитивную психологию, которые хотят узнать, что лежит в основе опыта разработки интерфейсов

И наконец, эта книга также окажется полезной для исследователей, занимающихся проблемами разработки «человеко-машинных» интерфейсов. Возможно, что после прочтения этой книги они уже никогда не смогут относиться к интерфейсам так же, как относились к ним прежде.

# Благодарности

*Так дружеский совет врагов сметет.*

*Уильям Шекспир  
«Генрих VI», акт III, сцена 1<sup>1</sup>*

Перечислить всех, кто помог мне написать эту книгу, трудно, потому что число этих людей огромно, а мой долг им так велик. Многие мои друзья, коллеги, родственники, критики и некоторые щедрые люди, знакомство с которыми ограничивается только Интернетом, внесли свой вклад в виде идей, критики, предложений и редакторской работы. Прошу извинить меня, если вы помогли мне, но я забыл вас упомянуть или написал ваше имя или звание неверно, и прошу сообщить обо всех таких случаях.

Благодарю сотрудников из Addison Wesley Longman, а именно редакторов, дизайнеров, сотрудников отделов по связям с общественностью, маркетингу и других, которые все, несомненно, отличаются не только особой компетентностью, но и доброжелательностью и терпением. С другой стороны, выбранные ими рецензенты, имена которых мне неизвестны, были особенно безжалостны, за что я им также благодарен.

Среди тех, кто перечислен ниже, – мои друзья, знакомые, коллеги, мой брат, преподаватель игры на трубе, который учит моего сына, приятель, который, как и я, увлекается авиамоделированием, – в общем, список на первый взгляд невероятно обширный. Только некоторые из упомянутых людей являются экспертами в области разработки интерфейсов, но все они прочитали рукопись этой книги и либо сделали к ней важные добавления, либо в течение многих лет помогали мне в размышлениях над нею. Это Дэвид Альзофон (David Alzofon) (тот самый, который нарисовал Квазимодо), Билл Эткинсон (Bill Atkinson), Томас Этвуд (Thomas Atwood), Пол Бейкер (Paul Baker), Джерри Баренхолтс (Jerry Barenholtz), Джон Бумгарнер (John Bumgarner), Дэвид Кулкинз (David Caulkins), Ph.D. Уильям Бакстон (William Bux-

---

<sup>1</sup> Перевод Е. Бируковой, ПСС в восьми томах. Изд-во «Искусство», 1958, т. 1. – *Примеч. пер.*

ton), Ph.D. Ренуик Карри (Renwick Curry), Роберт Фаулз (Robert Fowles), Ph.D. Джош Гаррет (Josh Garrett), Жан-Франсуа Грофф (Jean-Franzois Groff), Ph.D. Скот Ким (Scott Kim), Катлин Мандис (Kathleen Mandis), Памела Мартин (Pamela Martin), Трой Мей (Troy May), Ph.D. Мириам Мейслер (Miriam Meisler), Дуглас Маккенна (Douglas McKenna), Майкл С. Миллер (Michael S. Miller), доктор медицины (M.D.) Дэвид Мошэл (David Moshal), Эндрю Нильсен (Andrew Nielsen), Якоб Нильсен (Jakob Nielsen), Джули Ососк (Julie Ososke), Иан Паттерсон (Ian Patterson), Ph.D. Майкл Раскин (Michael Raskin), Эразмус Смамс (Erasmus Smums), Спайдер Робинсон (Spider Robinson), Минору Таояма (Minoru Taoyama), Шэй Телфер (Shay Telfer), Йессо Текериан (Yesso Tekerian), Брюс Тоньяццини (Bruce Tognazzini), Дэвид Уинг (David Wing), Ph.D. Терри Уиноград (Terry Winograd), участники местного отделения (BayCHI) исследовательской группы по проблемам взаимодействия человека и компьютера Ассоциации по вычислительной технике (ACM), которые давали мне возможность высказывать свои идеи и обсуждали их, а также студенты Центра компьютерных исследований музыки и акустики Стенфордского университета во главе с его директором Джоном Чоунингом (John Chowning).

Я счастлив, что у меня есть грамотная и в то же время любящая меня жена Линда Блум, дипломированная медсестра. Она никогда не переживала, что написание технических книг – не самый подходящий способ содержания семьи, если только это дело стоило того, чтобы я им занимался. Ее внимание к идеям, целям и отдельным деталям этой книги позволило сделать лучше многие ее страницы. Хотя не в моих силах было выбирать родителей, но они заслуживают особого уважения, ибо научили ценить людей больше, чем вещи, а также помогли мне полюбить искусство так же, как и науку, что и привело в итоге к этой работе. Мой сын Аза, несмотря на свой юный возраст, внес в эту книгу намного больший вклад, чем можно было бы ожидать, включая идеи, редактирование и кропотливую работу над иллюстрациями. Кроме того, необходимо отметить, что он и его сестры проявляли поразительное терпение в то время, как я писал эту книгу. Особенно важную роль в моей жизни играет Л. Роланд Дженайз (L. Roland Genise), мой лучший учитель. В университетские годы он подарил мне два таких качества, как уверенность в своих силах и любовь к математике. Также я счастлив упомянуть имена Брайена Хауэрда (Brian Howard) и Дугласа Уайета (Douglas Wyatt), которые разделили со мной дружбу, философию и музыку и были беспощадными редакторами моих первых работ. В этой книге я позволил себе не согласиться с некоторыми идеями доктора Дональда Нормана (Donald Norman). Несмотря на эти небольшие разногласия, я считаю его работы весьма ценными в данной области. Кроме того, без его критических замечаний и подсказок эта книга вряд ли смогла бы появиться. Я благодарен Биллу Верпланку (Bill Verplank), спокойному и приятному человеку, который способен столь мягко и тактично делать замечания, что только оказавшись но-

гами на полу, вы поймете, что на самом деле из-под вас вытащили ковер. Именно он убедил меня совершенно изменить тон и ориентацию книги, о чем я несколько не пожалел. Лин Дюпре (Lyn Durr), жесткий и весьма придирчивый редактор, также среди тех, чье мнение повлияло на окончательную форму этой книги. Советую вам обязательно прочитать ее работу «Ошибки в письменной речи». Многие идеи, часть которых изложена в этой книге, рождались и оттачивались в разговорах и работе с моим другом Джеймсом Уинтером (James Winter), доктором медицины (M.D.) и философии (Ph.D.). Специалист по вычислительной технике Дик Карпински (Dick Karpinski), весьма ироничный человек, который сам говорит, что ему палец в рот не клади, помогал мне в самых разных аспектах: объяснял непонятные технические детали, знакомил меня с важными людьми и ценными книгами или просто заскакивал с какой-нибудь хорошей идеей. Напоследок хочу еще вспомнить Питера Гордона (Peter Gordon), мудрого, настойчивого и, что особенно важно, терпеливого человека, который представлял меня в издательстве Addison-Wesley. Переписка с ним, будучи опубликована, открыла бы всему миру нашу склонность к постоянной игре слов и ужасным каламбурам, что могло бы навеки запятнать наши имена, но пролило бы свет на бесчисленные подробности, сопровождавшие создание такой с виду небольшой книжки.

Благодарю сотрудников корпорации Agfa за цифровую камеру, с помощью которой сделаны некоторые иллюстрации.

Кроме того, благодарю тех читателей, которые первыми предложили сделать отдельные исправления или изменения, принятые впоследствии. Это Эрик Блоссом (Eric Blossom), Джон Бонди (Jon Bondy), Поль Каббидж (Paul Cabbage), Питер Джонс (Peter Jones), Дж. А. Митчел (G. A. Michael), Кэм Митчнер (Cam Mitchner), Рич Морин (Rich Morin), Мартин Портмэн (Martin Portman) и Элизабет Рибо (Elisabeth Riba). Автор также благодарен Рич Морин за поддержку сайта *www.jefras-kin.com*.



# Введение. Важность основ

*Один человек, один компьютер.*

*Слоган компании  
Apple Computer*

Представьте себе, что вы поднялись на борт сияющего шикарной отделкой авиалайнера, оснащенного просторными, комфортабельными кожаными креслами с целым набором встроенной аудио- и видеотехники; в буфете вас ожидают отличная еда и напитки. Вы садитесь в свое кресло и смотрите в большой, чисто вымытый иллюминатор. Со вздохом предвкушения особенно приятного полета вы протягиваете руку к шкафчику впереди вас, чтобы поглядеть, что там. Сначала вы достаете весьма объемистую бутылку любимого напитка, а затем буклет с описанием этого замечательного воздушного лайнера.

В то время как двери закрываются и идут приготовления к взлету, вы усаживаетесь поудобнее и начинаете читать. Из буклета вы узнаете, что интерьер самолета создан трудами самых лучших в мире дизайнеров, что повара из пятизвездочных отелей лично составляли меню и готовили блюда и что в группу разработчиков самолета не были включены инженеры-авиаконструкторы, поскольку всемирно признанные дизайнеры сделали внешний вид самолета таким, что и без того создается впечатление авиалайнера, способного летать во много раз быстрее, чем любой другой.

Еще в буклете мелким шрифтом сообщается, что путешествие на этом самолете нередко даже в хорошую погоду сопровождается болтанкой и что достаточно регулярно с ним случаются катастрофы. Если же перелет обойдется без этих инцидентов, то в целом, как обещают авторы, ваше путешествие будет комфортным и интересным.

Теперь звук закрывающихся дверей внезапно принимает угрожающее значение, вы теряете спокойствие и чувствуете, что попали в ловушку. Вы начинаете думать, что именно этот рейс обречен и что вы предпочли бы сейчас сидеть в более жестком кресле, без любимого напитка и даже без бокового иллюминатора, лишь бы только самолет был оборудован хорошей и надежной техникой.

Представленная абсурдная ситуация довольно точно описывает суть большинства существующих сегодня «человеко-машинных» интерфейсов. Наши компьютеры и сотовые телефоны оснащены самыми последними моделями чипов и другой электронной начинкой. Современные операционные системы способны радовать глаз великолепными цветными заставками и стремительными трехмерными эффектами. Вы щелкаете по кнопке, и вот! – вы видите, как она движется самым реалистичным образом, слышите, как звук щелчка мыши весьма точно передается с помощью цифрового стереофонического воспроизведения, а затем, как только на экране открывается панель, до ваших ушей доносится чарующее глассандо арфы.

Но когда вы начинаете пользоваться этой системой, выясняется, что в некоторых случаях она неприятно ограничивает вас своим непредсказуемым поведением. Из тысяч команд, предусмотренных в системе, вам не удастся найти ту, которая нужна в данный момент, а простые стандартные процедуры выполняются бесконечно долго. Программа, приобретенная в прошлом году, вдруг перестает запускаться под улучшенной версией той же самой операционной системы, и вам приходится покупать новую версию программы. Ко всему прочему оказывается, что операционная система имеет свойство время от времени зависать.

В основе разработки хороших интерфейсов лежат некоторые основные принципы, которые на сегодня не являются общеизвестными. И вопрос о необходимости изучения этих принципов не возникает, поскольку кажется, что уже определено, как должны выглядеть и работать интерфейсы: ведь они непрерывно совершенствовались в течение двух десятилетий, основные разработчики программного обеспечения опубликовали руководства по созданию интерфейсов, чтобы обеспечить совместимость между ними, а существующие средства разработки позволяют быстро создавать любые интерфейсы, которые выглядят по-современному – подобно тому, как и упомянутый авиалайнер создавался, чтобы быть *похожим* на безопасный и комфортабельный летательный аппарат.

Все эти интерфейсы неспособны выполнять многие важные для нас задачи. Например, чтобы записать какую-то мысль, вы хотели бы просто подойти к компьютеру или другому устройству для обработки информации и начать набирать ее – без всякой загрузки, без необходимости открывать текстовый процессор, создавать файл, вообще без использования операционной системы. (Мое определение операционной системы звучит следующим образом: «То, с чем приходится возиться перед тем, как начать возиться с программой».) Чтобы добавить к репертуару системы несколько средств для выполнения простых операций, вы не обязаны изучать целую прикладную программу. К сожалению, в разработке интерфейсов изначально было взято неверное направление, и это привело к тому, что уровень их сложности стал неоправдан-

но высоким с точки зрения как технологической, так и логической необходимости.

Миллионы из нас имеют противоречивые отношения с информационными технологиями. Мы не можем жить без них, и в то же время нам трудно жить с ними. Тем не менее, проблема создания удобных и простых технологий имеет свои решения, хотя мы и не можем ими воспользоваться – они станут доступными, только если мы оставим груз прошлого. Привычный вариант интерфейса в виде рабочего стола, ориентированный на работу с прикладными программами, является частью этой проблемы. В этой книге предлагаются некоторые альтернативные варианты. В конце концов, компьютерные проблемы – это не погода, и мы все-таки *можем* что-то сделать для их разрешения.

С учетом широкого распространения Интернета, а также очевидной важности компьютерных продуктов, предназначенных для группового взаимодействия, может показаться странным, что содержание этой книги касается, главным образом, разработки однопользовательских интерфейсов. Одной из причин этого является тот факт, что проблема разработки однопользовательских интерфейсов еще не решена. Но главная причина состоит в том, что качество любого интерфейса в конечном итоге определяется качеством взаимодействия между одним человеком и одной системой – между ней и вами. *Если индивидуальное взаимодействие с некоторой системой не проходит для пользователя легко и комфортно, то в результате этот недостаток негативным образом отражается на качестве работы всей системы, независимо от того, насколько она хороша в других своих проявлениях.*



# 1

## Предпосылки

*Нет ничего более невозможного, чем написать книгу, которая бы получила одобрение каждого читателя.*

*Мигель де Сервантес*

В этой главе говорится о распространенном непонимании сущности таких систем, как интерфейсы, а также методов их разработки. Интерфейс – это нечто большее, чем окна, пиктограммы, выпадающие меню и мышь. Необходимость проектирования интерфейса уже на ранних стадиях разработки продукта иногда упускается из виду. Другой фактор, который часто недооценивается, состоит в том, что все мы наделены познавательными аппаратами, имеющими между собой много общего. При разработке интерфейсов следует сперва учесть общие факторы, а потом уже рассматривать индивидуальные различия. Но, к сожалению, существующие на сегодня средства конструирования интерфейсов не позволяют подойти к задаче именно таким образом.

Я не согласен с мнением, что пользоваться компьютерами сложно потому, что с их помощью мы пытаемся делать безнадежно сложные вещи. В действительности независимо от того, насколько сложной является задача, выполняемая тем или иным продуктом, составные части этой задачи все равно должны оставаться простыми. Эта глава заканчивается определением человекоориентированного интерфейса.

## 1.1. Определение интерфейса

*Позвоните по вышеуказанному номеру и испытайте невероятное разочарование от нашей системы голосовой почты.*

*Надпись под рекламным объявлением одной из марок обуви*

В этой книге выражения *интерфейс «человек-машина»* или *интерфейс «человек-компьютер»* я обычно буду сокращать до *пользовательского интерфейса* или просто *интерфейса*. Многие считают, что термин *пользовательский интерфейс* относится только к современным графическим пользовательским интерфейсам (graphical user interface, GUI), основанным на окнах и меню, управляемых с помощью мыши. Например, в одной из статей в журнале «Mobile Office» было сказано: «Уже недалеко то время, когда вам совсем не нужно будет задумываться об интерфейсе, вы будете просто разговаривать со своим компьютером». В ответ на это я мог бы заметить, что системы, управляемые голосом, действительно могут обходиться без окон, но телефонные автоответчики их также не имеют, и, тем не менее, их интерфейсы зачастую оказываются чрезвычайно плохими. Итак, способ, которым вы выполняете какую-либо задачу с помощью какого-либо продукта, а именно совершаемые вами действия и то, что вы получаете в ответ, и является интерфейсом. (См. также Raskin, 1993.)

## 1.2. Простое должно оставаться простым

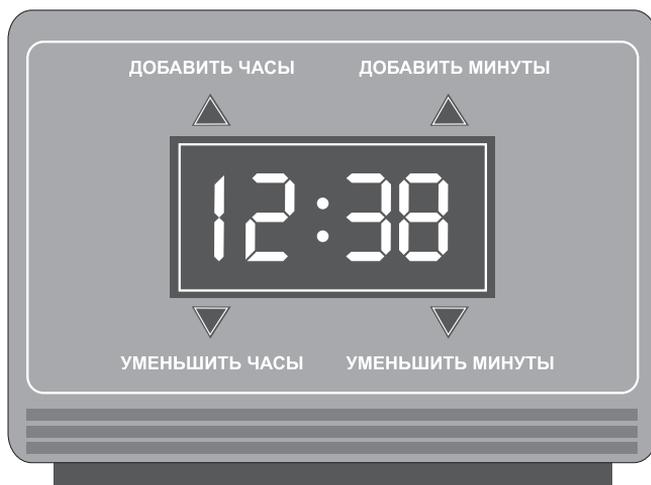
*Технология – странная вещь. Одной рукой она дает вам великие дары, а другой – наносит удар в спину.*

*С. П. Сноу (цитата из Jarman, 1992)*

Несмотря на рост количества специалистов по разработке интерфейсов, мало кто из потребителей заявляет, что новые продукты, например электронные четырехкнопочные наручные часы, стали проще в использовании, чем несколько десятилетий назад. Если вы скажете, что наручные часы, так же как и компьютеры, сегодня имеют намного большую функциональность (с чем можно согласиться) и что, следовательно, интерфейсы этих устройств должны стать более сложными (что сомнительно), то я позволю себе заметить, что эта сложность неоправданно возникает в отношении даже тех задач, которые раньше удавалось выполнять без усилий. Сложные задачи могут требовать сложных интерфейсов, но это не оправдывает усложнения простых задач. Сравните, например, насколько труднее установить время на электронных наручных часах с четырьмя кнопками, чем выполнить *то же самое* действие на механической модели часов. Простые задачи должны оставаться простыми независимо от уровня сложности всей системы.

Из всех нелепостей, создаваемых абсурдными конструкциями интерфейсов, именно усложнение простого чаще всего оказывается поводом для высмеивания в комиксах или комедийных сценах. Например,

в фильме «Городские жулики» (City Slickers) три товарища гонят стадо коров. Один из героев, его играет Билли Кристал (Billy Crystal), безуспешно пытается, видимо, уже не один час, объяснить друзьям, как с помощью видеомэгафона записать какую-нибудь программу на одном канале во время просмотра другого. Когда, в конце концов, друзья выходят из себя от длинного и непонятного объяснения, персонаж Кристала с радостью соглашается сменить тему и предлагает вместо этого рассказать, как устанавливать время на часах в том же видеомэгафоне. Это предложение приводит друзей в ярость, что вызывает смех у зрителя. Комический эффект порождается несоответствием между очевидной простотой задачи и сложностью интерфейса. Если бы лицевая панель видеомэгафона была снабжена специальными кнопками, расположенными над и под цифрами часов, как это показано на рис. 1.1, тогда мало у кого возникали бы трудности при установке времени.



*Рис. 1.1. Легко настраиваемые электронные часы для видеомэгафона. Еще лучшим вариантом были бы часы, в которых время автоматически устанавливается по сигналам точного времени, передаваемым по радио*

## 1.3. Ориентация на человека и на пользователя

*Мы слишком усложнили программное обеспечение и забыли главную цель.*

*Джим и Сандра Сандфорс*

Не только разработчики интерфейсов, но и руководители предприятий электронной и компьютерной промышленности понимают необходимость ориентации разработок на нужды пользователей и покупателей. И первым шагом в этом направлении является стремление узнать своего пользователя, что на практике обычно означает обращение за

помощью к специалистам в той или иной области. Специалисты действительно могут хорошо разбираться в особенностях и деталях решаемой проблемы, но их экспертные знания, как правило, не касаются вопросов человеческой психологии. Хотя у пользователей могут быть разные потребности в зависимости от конкретной задачи, тем не менее, в целом они проявляют много общих ментальных характеристик. Прежде чем приступать к разработке самой программы или пытаться учесть различия между отдельными пользователями, разработчики интерфейса могут облегчить свой труд, сосредоточив внимание на том, что является общим для всех людей с точки зрения требований к интерфейсу. По завершении этой стадии разработчики интерфейса уже могут приступить к согласованию различий между отдельными пользователями и группами пользователей и, в конечном итоге, к поиску оптимального варианта, удовлетворяющего широкому диапазону требований пользовательских задач. Однако этот первый важный шаг, во время которого проект интерфейса приводится в соответствие с общими законами психологии, в процессе разработки обычно пропускается. Разработчики интерфейсов предпочитают не задумываться об этом и больше полагаются на так называемые «промышленные стандарты». В результате все широко используемые сегодня модели интерфейсов построены без учета закономерностей мышления и поведения человека. Например, почти во всех компьютерных системах файлы должны иметь собственные имена. Между тем часто возникают ситуации, когда нам трудно вспомнить, под каким именем мы сохранили файл полгода назад. (Одно из возможных решений этой проблемы обсуждается в разделе 5.3.) Таким образом, мы хотим, чтобы программное обеспечение было простым и понятным, своим безупречным поведением показывая нам, что его создатели больше работали над удобством использования, нежели над привлекательным внешним видом своего продукта.

## 1.4. Инструменты, которые препятствуют новым идеям

Создание хороших интерфейсов требует большой и напряженной работы. Считается, что такие известные на рынке инструменты для построения интерфейсов, как Visual Basic и Visual C++, позволяют снизить стоимость разработки и ускорить ее внедрение. Несмотря на все свои полезные свойства, эти инструменты нечасто будут упоминаться в этой книге. Причина состоит в том, что они основаны на традиционных парадигмах и, следовательно, слишком ограничивают ваши возможности. Аналогичным образом принципы создания интерфейсов в таких системах, как Macintosh или Windows, а также часть подходов, предлагаемых в различных книжных изданиях, посвященных разработке интерфейсов, иногда оказываются явно ошибочными – зачастую из-за корпоративной необходимости поддерживать совместимость с ранними версиями интерфейса, а также из предубеждения, что поль-

зователи непременно отнесутся с неодобрением к попыткам отойти от старых, привычных принципов построения интерфейсов. Действительное усовершенствование интерфейсов возможно, если подходы к их разработке будут серьезно пересмотрены. При этом разработчику необходимо найти компромисс между оправданным применением уже устоявшихся парадигм, которые облегчают изучение интерфейса пользователем, и новыми подходами, которые позволяют сделать интерфейс более удобным и практичным. Конечно, в ситуации, когда часто меняется состав группы разработчиков или круг потребителей продукта, стремление придерживаться известных подходов, возможно, было бы лучшим решением. *Но в тех случаях, когда известно, что большая часть времени у пользователей будет уходить на рутинные, повторяющиеся операции, а обучение в то же время не потребует больших затрат, верным решением является разработка интерфейса с максимальной продуктивностью, даже если впоследствии от пользователя потребуются некоторые усилия по его изучению.*

## 1.5. Разработка интерфейса как часть общего цикла разработки

Применяемые сегодня методы разработки проектов зачастую не считаются с необходимостью разработки интерфейса. Это упущение может быть следствием того, что специалисты по разработке интерфейсов привлекаются к проекту слишком поздно, когда возможности улучшения качества взаимодействия между пользователем и продуктом большей частью уже потеряны. Интерфейсом удобнее всего заниматься именно на начальных стадиях разработки. И если специалисты по интерфейсам привлекаются уже после того, как программное обеспечение спроектировано и определены его инструменты или когда разработка программы уже почти завершена, то их рекомендации могут потребовать переделки всей выполненной работы, что, естественно, является неприемлемым. Когда бюджет проекта уже исчерпан и рабочий план почти завершён, перспектива отказа от большей части или даже всего дизайна и готового кода, конечно, не может вызвать энтузиазма у менеджеров проекта. Так что даже в такой современной книге по управлению проектами, как «UML Toolkit» (Eriksson and Magnus, 1998), не говорится о необходимости рассматривать интерфейс уже на стадии анализа требований к проекту, которую авторы обозначают как первую фазу его разработки. Однако в действительности разработка интерфейса не должна откладываться до стадии технической реализации, которая в плане Эриксона и Магнуса является третьей фазой. *Определив задачу, для которой продукт предназначен, сначала спроектируйте интерфейс, после чего приступайте к его реализации.* Это повторяющийся процесс. Определение задачи будет меняться во время разработки интерфейса. Поэтому весь процесс разработки продукта будет проходить в соответствии с изменениями в задаче продукта и его интерфейсе. Здесь необходимо стремиться к максимальной

гибкости. На первом этапе разработки следует определить, что именно должен сделать пользователь для получения того или иного результата и как система должна отвечать на каждое его действие.

*Пользователи не задумываются над тем, как устроена машина, пока она справляется со своими задачами.* При этом не имеет значения, какой именно процессор используется и является ли язык программирования объектно-ориентированным, многопоточным или, быть может, называется какими-то другими умными словами. Для пользователей важнее всего удобство и результаты. Но все, что они видят, – это интерфейс. Другими словами, *с точки зрения потребителя именно интерфейс является конечным продуктом.*

---

### **Ваше время бесценно, ваша работа священна**

Я приучился часто сохранять проделанную работу, чтобы даже в случае системного сбоя не потерять большую часть своего труда. В конце каждого абзаца или даже после нескольких предложений я при помощи сочетания клавиш вызываю команду сохранения. Эта команда создает копию текста на диске, где он может оставаться относительно защищенным от потери в случае сбоя. Приблизительно каждый час я создаю резервную копию своей работы с помощью энергонезависимого запоминающего устройства, которое может быть физически извлечено из компьютера и таким образом защищено от любых неожиданностей в его работе. Кроме того, каждую неделю я сохраняю резервную копию всей системы на внешнем диске. Это не значит, что я параноик, – я всего лишь считаю, что такой подход практичен. Однако необходимости во всех этих сложных процедурах не должно возникать. *Система должна рассматривать все данные, вводимые пользователем, как бесценные.* И если перефразировать Первый закон робототехники Азимова: «Робот не может причинить вред человеку или своим бездействием допустить, чтобы человеку был причинен вред», то первый закон проектирования интерфейсов должен звучать примерно так: «Компьютер не может причинить вред данным пользователя или своим бездействием допустить, чтобы данным был причинен вред».

Работая над этой книгой, я по совету своих редакторов стал использовать опцию, позволяющую либо принять, либо отклонить изменения, внесенные в документ. Каждый раз, сделав несколько изменений, я запускал команду сохранения. Когда произошел сбой системы, я не стал беспокоиться, полагаясь на сделанные мной периодические сохранения. Однако когда я попытался найти файлы с самыми последними изменениями, это не удалось, и мне пришлось делать ту же работу заново. Немного поэкспериментировав, я выяснил, что при включенной опции «принять или отклонить» команда сохранения, подаваемая с клавиатуры, перестает действовать. Однако пользователю никакого предупреждения об этом не дается. В ре-

зультате пропало больше трех часов моего труда, и мне пришлось тратить время на эксперименты и выяснять, что же произошло и как это предотвратить в будущем. Если не считать излишней сложности сегодняшних компьютерных систем, именно такие досадные мелочи говорят о необходимости усовершенствования подходов к разработке интерфейсов.

Наилучшей формулировкой второго закона интерфейса может быть следующее утверждение: *«Компьютер не должен тратить впустую ваше время или вынуждать вас выполнять действия сверх необходимых»*. В разделе 4.3 будет рассматриваться измерение объема работы, необходимого для выполнения той или иной задачи.

---

## 1.6. Определение человекоориентированного интерфейса

*Можно создать самолет с любыми техническими характеристиками, которые только пожелает Министерство военно-воздушных сил, если при этом не требуется, чтобы он мог летать.*

*Вилли Мессершмидт (выдающийся немецкий авиаконструктор времен второй мировой войны)*

*Интерфейс является ориентированным на человека, если он отвечает нуждам человека и учитывает его слабости.* Чтобы создать такой интерфейс, необходимо иметь представление о том, как действуют люди и машины. Кроме того, следует развить в себе способность чувствовать те трудности, с которыми сталкиваются люди. И это не всегда просто. Мы настолько привыкли к тому, как работают программы, что соглашаемся принять их методы работы как данность, – даже в тех случаях, когда их интерфейсы неоправданно сложны, запутанны, неэкономны и побуждают людей к ошибкам.

Многие из нас испытывают раздражение, например, от того, что для запуска (иначе говоря, загрузки) компьютера требуется какое-то время. В 1999 году была реклама одного автомобильного радиоприемника со встроенным компьютером, в которой утверждалось, что «в отличие от домашнего компьютера, эта система не заставит вас долго ждать, пока она загрузится». Внимательное изучение шести наиболее серьезных работ в области разработки интерфейсов показывает, что даже в этих книгах, написанных в основном в то время, когда разработке интерфейсов стали придавать важное значение, проблема загрузки не упоминается (Shneiderman, 1987; Norman, 1988; Laurel, 1990; Tognazzini, 1992; Mayhew, 1992; Cooper, 1995). Тем не менее, я уверен, что каждый из названных авторов всецело согласился бы с тем, что сокращение или устранение задержки при запуске компьютера улучшило бы эффективность его использования, тем более что я еще не встречал пользователя, у которого такая задержка не вызывала бы раздраже-

ние. Однако никогда не существовало технической необходимости в том, чтобы компьютер после включения начинал работать более чем через несколько секунд. Наши компьютеры долго загружаются только лишь потому, что многие дизайнеры и разработчики не потрудились сделать интерфейс в этом отношении ориентированным на человека. Кроме того, некоторые люди думают, что если компьютеры с медленной загрузкой продаются миллионами, то это якобы свидетельствует об их высокой производительности.

Нельзя сказать, что проблема долгой загрузки машины всегда игнорировалась. Уже вышедший из употребления Apple Newton, Palm Pilot и другие карманные компьютеры могут запускаться мгновенно, а появление на некоторых компьютерах «спящего режима» – состояния, в котором компьютер потребляет меньше энергии, чем в обычном режиме, и из которого он может быть быстро переведен в рабочее состояние, – это шаг в правильном направлении.

Инженерам удавалось с успехом решать и более сложные проблемы. Например, в ранних моделях телевизоров необходимо было ждать около минуты, пока разогревалась катодная трубка кинескопа. В некоторых моделях инженеры добавили специальную схему, которая поддерживала катодную трубку в теплом состоянии, что позволило сократить время достижения рабочей температуры. (Поддержание катодной трубки в разогретом состоянии потребовало бы большого расхода электричества и уменьшило бы срок ее службы.) В другом варианте был разработан кинескоп с катодной трубкой, которая разогревалась в течение нескольких секунд. И в том и в другом случае интересы пользователя были удовлетворены. В начале двадцатого столетия был создан автомобиль на паровой тяге, называвшийся Стенли СТИМЕР (Stanley Steamer). Несмотря на все свои очевидные достоинства, этот механизм не имел успеха из-за одного недостатка: чтобы тронуться с места, от момента зажигания до достижения необходимого давления в котле требовалось подождать 20 минут.

Принцип разработки, согласно которому программные продукты не должны вынуждать пользователя ждать без необходимости, можно считать очевидным и ориентированным на человека. Таким же является и стремление не подгонять пользователя. В общем виде этот принцип можно было бы сформулировать следующим образом: *«Ритм взаимодействия должен устанавливаться самим пользователем»*.

Не требуется обладать большими техническими знаниями, чтобы понять, что большая пропускная способность коммуникационных линий может ускорить передачу веб-страниц. Однако другие взаимосвязи иногда бывают не столь очевидны. Поэтому для разработчиков интерфейсов «человек-машина» важно знать внутренние механизмы технологии. В противном случае у них не будет возможности оценивать достоверность утверждений, высказанных, например, программистами или специалистами по аппаратной разработке относительно осуществимости тех или иных элементов интерфейса.