

## **ИНТЕРФЕЙСЫ ПОЛЬЗОВАТЕЛЯ**

Для статей по смежным темам см. ФАКТОРЫ ЧЕЛОВЕКА В КОМПЬЮТЕРНЫХ СИСТЕМАХ; ИНТЕРАКТИВНЫЕ ВХОДНЫЕ УСТРОЙСТВА; ИНТЕРАКТИВНАЯ СИСТЕМА; и ОКНО ОКРУЖАЮЩЕЙ СРЕДЫ.

Пользовательский интерфейс - это часть интерактивной компьютерной системы, которая взаимодействует с пользователем. Дизайн пользовательского интерфейса включает в себя любой аспект системы, который видим пользователю. Однажды все компьютерные пользователи были специалистами в области вычислений, а интерфейсы состояли из проводов перемычек в патч-платах, перфокартных карточек, подготовленных в автономном режиме, и пакетных распечаток. Сегодня широкий спектр неспециалистов использует компьютеры, а клавиатуры, мыши и графические дисплеи являются наиболее распространенным интерфейсом. Пользовательский интерфейс становится все большей и большей частью программного обеспечения в компьютерной системе - и более важной частью, поскольку более широкие группы людей используют компьютеры. По мере того как компьютеры становятся более мощными, критическое узкое место в применении компьютерных систем для решения проблем теперь чаще происходит в пользовательском интерфейсе, а не в аппаратном или программном обеспечении компьютера.

Поскольку дизайн пользовательского интерфейса включает в себя все, что видимо для пользователя, дизайн интерфейса глубоко проникает в дизайн интерактивной системы в целом. Хороший пользовательский интерфейс не может быть применен к системе после его создания, но должен быть частью процесса проектирования с самого начала. Правильный дизайн пользовательского интерфейса может существенно повлиять на время обучения, скорость выполнения, частоту ошибок, удовлетворенность пользователей и сохранение пользователем знаний об операциях с течением времени. Бедные проекты прошлого уступают место элегантным системам. Описательные таксономии пользователей и задач, прогностические модели производительности и пояснительные теории разрабатываются для руководства проектировщиками и оценщиками. Хафазардные и интуитивные стратегии развития с претензиями на «дружелюбие пользователя» уступают более научному подходу. Измерение времени обучения, производительности, ошибок и субъективного удовлетворения в настоящее время является частью процесса проектирования.

### **Дизайн пользовательского интерфейса**

Дизайн пользовательского интерфейса начинается с анализа задачи - понимания основных задач пользователя и проблемной области. Пользовательский интерфейс должен быть разработан с точки зрения терминологии пользователя и концепции его или ее работы, а не программиста. Таким образом, хорошее понимание когнитивных и поведенческих характеристик людей в целом, а также конкретной популяции пользователей. Хороший дизайн пользовательского интерфейса работает с возможностями и ограничениями пользователя, а не с машинами; это относится к универсальным интерфейсам для больших групп людей, а также к разработке специальных интерфейсов для пользователей с физическими или другими нарушениями. Важное значение имеет также знание характера работы пользователя и окружающей среды. Затем выполняемая задача может быть разделена, а части, назначенные пользователю или машине, основываются на знании возможностей и ограничений каждого из них.

## **Уровни дизайна**

Полезно рассмотреть пользовательский интерфейс на нескольких отдельных уровнях абстракции и разработать дизайн и реализацию для каждого. Это упрощает задачу разработчика, позволяя разделить ее на несколько меньших задач. Дизайн пользовательского интерфейса часто делится на концептуальные, семантические, синтаксические и лексические уровни.

Концептуальный уровень описывает основные сущности, лежащие в основе представления пользователя о системе, и возможные действия на них. Семантический уровень описывает функции, выполняемые системой. Это соответствует описанию функциональных требований системы, но не учитывает, как пользователь будет вызывать функции. Синтаксический уровень описывает последовательности входов и выходов, необходимых для вызова описанных функций. Лексический уровень определяет, как входы и выходы фактически формируются из примитивных аппаратных операций.

Синтаксически-семантическая модель объекта-действия - это связанный подход; он также отделяет концепцию задачи и компьютера (т. е. семантику в предыдущем абзаце) от синтаксиса для выполнения задачи. Например, задача написания статьи научного журнала может быть разложена в подзадачи для написания титульной страницы, тела и ссылок. Аналогичным образом титульная страница может быть разложена на уникальный заголовок, один или несколько авторов, абстрактный и несколько ключевых слов. Чтобы написать научную статью, пользователь должен понять эту семантику задачи. Чтобы использовать текстовый процессор, пользователь должен узнать о семантике компьютера, такой как каталоги, имена файлов, файлы и структуру файла. Наконец, пользователь должен изучить синтаксис команд для открытия файла, вставки текста, редактирования и сохранения или печати файла. Новичок часто пытается научиться выполнять свои задачи на компьютере и запоминать синтаксические детали. После изучения семантика задачи и компьютера относительно стабильна в человеческой памяти, но синтаксические детали часто нужно репетировать. Знающий пользователь одного текстового процессора, который хочет изучить второй, должен только изучить новые синтаксические детали.

## **Системы управления пользовательским интерфейсом**

Система управления пользовательским интерфейсом (UIMS) является программным компонентом, который отделен от прикладной программы, выполняющей основную задачу. UIMS осуществляет взаимодействие с пользователем, реализуя синтаксический и лексический уровни, в то время как остальная часть системы реализует семантический уровень. Подобно операционной системе или графической библиотеке, UIMS отделяет функции, используемые многими приложениями, и перемещает их в общую подсистему. Он централизует реализацию пользовательского интерфейса и позволяет некоторые из усилий по разработке инструментов для пользовательских интерфейсов амортизироваться во многих приложениях и совместно использоваться ими. Он также поощряет последовательный «внешний вид» пользовательских интерфейсов к различным системам, поскольку они совместно используют компонент пользовательского интерфейса. UIMS также поддерживает концепцию независимости диалога, где могут быть внесены изменения в дизайн интерфейса (диалог пользователь-компьютер), не затрагивая код приложения. Это поддерживает разработку альтернативных пользовательских интерфейсов для одного и того же приложения (семантики), что облегчает итеративное усовершенствование интерфейса посредством прототипирования и тестирования и, в будущем, альтернативные интерфейсы для пользователей с различными физическими или другими недостатками. UIMS требует языка или метода для точного определения пользовательских

интерфейсов; это также позволяет дизайнеру интерфейса описывать и изучать различные возможные пользовательские интерфейсы перед их созданием. UIMS становятся мощными инструментами, которые не только уменьшают усилия в области развития, но и стимулируют поисковые прототипы.

### **Синтаксический дизайн уровней: стили взаимодействия**

Основными классами пользовательских интерфейсов, которые в настоящее время используются, являются командные языки, меню, формы, естественный язык, прямые манипуляции, виртуальная реальность и их комбинации. Каждый стиль взаимодействия имеет свои достоинства для конкретных пользовательских сообществ или наборов задач. Выбор стиля или комбинации стилей является ключевым шагом, но в каждом из них есть несколько минутных решений, которые определяют эффективность получаемой системы.

#### **Командный язык**

Пользовательские интерфейсы языка команд используют искусственные языки, подобно языкам программирования. Они кратки и недвусмысленны, но для новичков часто бывает трудно учиться и помнить. Однако, поскольку они обычно позволяют пользователю комбинировать конструкции новыми и сложными способами, они могут быть более мощными для продвинутых пользователей. Для них командные языки дают сильное чувство, что они отвечают за то, что они берут на себя инициативу, а не реагируют на компьютер. Пользователи командного языка должны изучать синтаксис, но часто могут быстро выражать сложные возможности, не заставляя читать отвлекающие подсказки. Тем не менее, частота ошибок обычно высока, требуется обучение, а удержание может быть неудовлетворительным. Сообщения об ошибках и онлайн-помощь трудно обеспечить из-за разнообразия возможностей и сложности связанных задач с компьютерными концепциями и синтаксисом. Языки команд и более длинные запросы или языки программирования являются областью частых пользователей-экспертов (опытных пользователей), которые часто получают удовлетворение от овладения сложным набором понятий и синтаксиса. Интерфейсы командного языка также являются стилем, наиболее подходящим для программирования, то есть написанием программ или сценариев команд ввода пользователя.

#### **Меню**

Пользовательские интерфейсы на основе меню явно предоставляют параметры, доступные пользователю в каждой точке диалога. Пользователи читают список элементов, выбирают наиболее подходящую для их задачи, тип или точку, чтобы указать их выбор, убедитесь, что выбор правильный, иницилируйте действие и наблюдайте эффект. Если терминология и смысл элементов понятны и понятны, пользователи могут выполнять свои задачи с небольшим изучением или запоминанием и несколькими нажатиями клавиш. Меню требует только того, чтобы пользователь мог распознавать желаемую запись из списка, а не отзывать его, размещая меньшую нагрузку на долговременную память. Наибольшая выгода может заключаться в том, что существует четкая структура для принятия решений, поскольку одновременно предлагается только несколько вариантов. Этот стиль взаимодействия подходит для начинающих и прерывистых пользователей. Он также может привлекать частых пользователей, если механизмы отображения и выбора очень быстрые. Основным недостатком является то, что они могут раздражать опытных пользователей, которые уже знают, что они хотят сделать, и не должны видеть их в списке. Однако хорошо продуманные системы меню могут обеспечивать обход для опытных пользователей. Меню также

трудно применять к «неглубоким» языкам, которые имеют большое количество вариантов в нескольких точках, потому что отображение опций становится слишком большим. Для дизайнеров системы выбора меню требуют тщательного анализа задач, чтобы обеспечить удобство поддержки всех функций и тщательно выбирать терминологию и использовать ее последовательно. Программные средства для поддержки выбора меню помогают в обеспечении согласованного дизайна экрана, проверки полноты и поддержки обслуживания.

### Заполнение формы

Выбор меню обычно становится громоздким, когда требуется ввод данных; заполнение формы (также называемое заполнением пробелов) полезно здесь. Пользователи видят отображение связанных полей, перемещают курсор между полями и вводят данные по желанию, так же, как и с бумажной формой для счета-фактуры, листа данных персонала или формы заказа. Часто наблюдение за полным набором связанных полей на экране в знакомом формате часто очень полезно. Взаимодействие с заполнением формы требует, чтобы пользователи понимали метки полей, знали допустимые значения, были знакомы с полями ввода и редактирования и могли отвечать на сообщения об ошибках. Эти требования подразумевают, что пользователи должны иметь некоторую подготовку или опыт.

### Естественный язык

Главное преимущество пользовательских интерфейсов на естественном языке - это, конечно же, то, что пользователь уже знает язык. Надежда, что компьютеры будут реагировать должным образом на произвольные фразы или фразы на естественном языке, привлекла многих исследователей и разработчиков систем, но с ограниченным успехом до сих пор. Взаимодействие на естественном языке обычно не имеет большого контекста для выпуска следующей команды, часто требует «диалога уточнения» и может быть более медленным и более громоздким, чем альтернативы. Поэтому, учитывая современное состояние, такой интерфейс должен быть ограничен некоторым подмножеством естественного языка, и подмножество должно быть выбрано тщательно - как в словаре, так и в ряде синтаксических конструкций. Такие системы часто плохо себя ведут, когда пользователь слегка отклоняется от подмножества. Поскольку они начинают с представления иллюзии о том, что компьютер действительно может «говорить по-английски», системы могут ловушки или разочарования начинающих пользователей. По этой причине техники человеческого фактора могут помочь. Изучение человеческой задачи задачи и терминов и конструкций, которые люди обычно используют для ее описания, можно использовать для надлежащего ограничения подмножества естественного языка на основе эмпирического наблюдения. Изучение человеческих факторов может также определять задачи, для которых естественный язык является хорошим или плохим. Хотя будущие исследования на естественном языке дают надежду на человеко-компьютерное общение, настолько естественное, что «так же, как разговаривать с человеком», такой разговор не всегда может быть самым эффективным способом командования машиной. Это часто более подробный и менее точный, чем компьютерные языки. В таких ситуациях, как операция, управление воздушным движением и диспетчеризация аварийных транспортных средств, люди развивали короткие, высокоформатные языки, похожие на компьютерные языки, для общения с другими людьми. Для частых пользователей усилия по изучению такого искусственного языка перевешивают его краткость и точность, и часто предпочтительнее естественный язык.

## Прямое управление

В графическом или прямом стиле управления пользовательским интерфейсом (GUI) на экране отображается набор объектов, и пользователь имеет репертуар манипуляций, который может выполняться на любом из них. Это означает, что пользователь не имеет командного языка для запоминания за пределами стандартного набора манипуляций, мало когнитивных изменений режима и напоминания о доступных объектах и их состояниях, которые постоянно отображаются на дисплее. Примеры такого подхода включают в себя программы рисования, электронные таблицы, системы производства или системы управления технологическими процессами, которые показывают схему завода, системы управления воздушным движением, некоторые учебные и летные модели, видеоигры и рабочий стол Xerox Star и его потомков (Macintosh, Windows, и различные менеджеры файлов X Window). Указывая на объекты и действия, пользователи могут быстро выполнять задачи, сразу наблюдать результаты и, при необходимости, отменить действие. Клавиши ввода команд или меню заменяются устройствами перемещения курсора, такими как светячок, джойстик, сенсорный экран, трекбол или мышь, чтобы выбрать из видимого набора объектов и действий. Прямая манипуляция привлекательна для новичков, легко запоминается для прерывистых пользователей, поощряет разведку и, с осторожным дизайном, может быть быстрой для опытных пользователей. Ключевой трудностью при разработке таких интерфейсов является поиск подходящих манипулируемых графических представлений или визуальных метафор для объектов проблемной области, таких как рабочий стол и шкаф для хранения. Главный недостаток прямой манипуляции заключается в том, что часто бывает сложно создавать скрипты или параметризованные программы на таком динамическом и эфемерном языке.

В хорошо продуманном интерфейсе прямой манипуляции действия пользователя должны быть максимально приближены к мыслям пользователя, которые мотивировали эти действия; следует сократить разрыв между намерениями пользователя и действиями, необходимыми для их ввода в компьютер. Цель состоит в том, чтобы опираться на оборудование и навыки, которые люди приобрели благодаря эволюции и опыту, и использовать их для общения с компьютером. Интерфейсы прямого управления пользовались большим успехом, особенно с новыми пользователями, в основном потому, что они используют аналогию с существующими человеческими навыками (указывая, захватывая, перемещая объекты в пространстве), а не обученные поведением.

## Виртуальная реальность

Среды виртуальной реальности несут иллюзию пользователя манипулирования реальными объектами и преимуществом естественного взаимодействия. Соединив движение головы пользователя с изменениями изображений, представленных на головном дисплее, создается иллюзия окружения мира образованных компьютером изображений или виртуальной среды. Сенсоры для рук позволяют пользователю взаимодействовать с этими изображениями, как если бы они были реальными объектами, расположенными в окружающем его пространстве. Интерфейсы дополненной реальности смешивают виртуальный мир с реальным миром через зеркальное зеркало с половиной серебра или телевизионную камеру, позволяя накладывать виртуальные изображения на реальные объекты и аннотации или другие компьютерные данные, которые должны быть привязаны к реальным объектам. Современное состояние виртуальной реальности требует дорогостоящего и громоздкого оборудования и обеспечивает очень низкий экран отображения, поэтому такие интерфейсы в настоящее время используются в основном там, где чувство «присутствия» в виртуальном мире имеет первостепенное значение, например,

обучение огню бойцов или лечения фобий. По мере совершенствования технологии они, скорее всего, будут использоваться более широко.

Интерфейсы виртуальной реальности, такие как интерфейсы прямой обработки, набирают силу, используя ранее существовавшие возможности и ожидания пользователя. Для навигации по обычной компьютерной системе требуется набор изученных, неестественных команд, таких как ключевые слова, которые нужно ввести, или функциональные клавиши для нажатия. Навигация через систему виртуальной реальности использует существующие, естественные «навигационные команды» пользователя, например, позиционирование головы или глаз, поворот его тела или движение к чему-то интересному. Результатом является более естественный пользовательский интерфейс, поскольку взаимодействие с ним больше похоже на взаимодействие с остальным миром.

### Другие вопросы

Смешение нескольких стилей может быть подходящим, когда требуемые задачи и пользователи разнообразны. Команды могут привести пользователя к заполнению формы, когда требуется ввод данных, или всплывающие (или раскрывающиеся) меню могут использоваться для управления средой прямого манипулирования, когда подходящая визуализация операций не может быть найдена. Область совместной работы с компьютером расширяет понятие единого пользовательско-компьютерного интерфейса к интерфейсу, который поддерживает сотрудничество группы пользователей.

Хотя интерфейсы, использующие современные методы, такие как прямая манипуляция, часто легче изучать и использовать, чем обычные, их значительно сложнее построить, поскольку в настоящее время они обычно запрограммированы на низкоуровневом, специальном уровне. По-прежнему необходимы соответствующие концепции разработки программного обеспечения высшего уровня и абстракции для работы с этими новыми технологиями взаимодействия. Одним из решений является технология прямого манипулирования фактическим дизайном и построением интерфейсов прямой обработки. Указание графического внешнего вида пользовательского интерфейса («внешний вид» посредством прямой манипуляции относительно прост и обеспечивается многими текущими инструментами, такими как Visual Basic, но описание поведения диалога («чувство») больше сложный и еще не поддержанный, предопределенные или «консервированные» элементы управления и виджеты представляют собой современное состояние.

### **Лексический дизайн уровней: задачи, устройства и методы взаимодействия**

Лексический дизайн начинается с задач взаимодействия, необходимых для конкретного приложения. Это примитивные входы низкого уровня, требуемые от пользователя, такие как ввод текстовой строки или выбор команды. Для каждой задачи взаимодействия дизайнер выбирает подходящее устройство взаимодействия и метод взаимодействия (способ использования физического устройства для выполнения задачи взаимодействия). Существует несколько способов использования одного и того же устройства для выполнения одной и той же задачи. Например, можно было бы использовать мышь для выбора команды с помощью всплывающего меню, фиксированного меню (палитры или панели инструментов), множественного нажатия, обхода желаемой команды или даже записи имени команды с помощью мыши.

## Устройства ввода

Операции ввода варьируются от открытых текстовых обработок или программ рисования до простых повторных нажатий клавиш ENTER для поворота страницы в электронном документе. В то время как клавиатура и мышь были стандартным устройством ввода данных, все чаще привлекают альтернативы для многих задач. Высокоточные сенсорные экраны сделали это долговечное устройство более привлекательным для общественного доступа, домашнего контроля, управления процессом и других приложений. Джойстики, трекболы и планшеты данных со стилусами с многочисленными вариантами также полезны для различных задач указания и манипулирования. Речевой ввод для голосовой почты и распознавания речи для команд эффективен, особенно по телефону и физически отключен. Другие методы ввода включают в себя клавиши, которые могут быть динамически помечены, речь, 3D-указатель, жест рукой, движение всего тела и визуальная линия взгляда.

## Устройства вывода

Механизмы вывода должны быть успешными в передаче пользователю текущего состояния системы и каких действий в настоящее время доступны. Экран CRT стал стандартным подходом, но альтернативными являются плоские панели (светодиодные, жидкокристаллические, плазменные, электролюминесцентные и другие) и печатные устройства. Текущие экраны с высоким разрешением обеспечивают примерно 1000 x 1000 пикселей; но их разрешение (в точках на дюйм) все еще намного грубее, чем типичная распечатка или фотография бумаги, и их размер намного меньше, чем обычный рабочий стол пользователя, доска объявлений или другая рабочая поверхность. Дисплеи с высоким разрешением могут улучшить читаемость текстовых дисплеев, чтобы производительность могла соответствовать производительности машинописных документов. Синтезированный или оцифрованный речевой выход эффективен и экономичен, особенно в телефонах и физически отключен. Системы голосовой почты, которые хранят и пересылают оцифрованные голосовые сообщения, продолжают расти в популярности. Другие выходные носители включают анимированную графику, окна, значки, дисплеи активного значения, манипулируемые объекты, гипертекст и гипермедиа, дисплеи с поддержкой головок и неречевой звук.

## Выводы

Человеческая инженерия, однажды увиденная как краска, надетая в конце проекта, теперь чаще становится стальной рамкой, на которой построена конструкция. Академические и промышленные исследователи используют силу эмпирического наблюдения и традиционного научного метода в человеко-компьютерном взаимодействии. Классические экспериментальные методы психологии применяются для решения сложных познавательных задач человека с помощью информационных и компьютерных систем. Редукционистский подход, необходимый для контролируемых экспериментов, дает небольшие, но надежные результаты. Благодаря множественным повторениям с аналогичными задачами, предметами и условиями эксперимента, общность и валидность могут быть улучшены. Каждый небольшой экспериментальный результат становится плиткой в мозаике человеческой деятельности с компьютеризированными информационными системами.

В то же время целостные подходы к наблюдению участников, также известные как тестирование на удобство использования или формативная оценка, способствуют пониманию, которое может быть немедленно применено к проектам и гипотезам, которые приводят к

контролируемым экспериментам. Простое понятие просить пользователей «думать вслух», поскольку они используют компьютерную систему, дает большие преимущества. Видеопотоки пользователей, борющихся с системой, оказывают сильное влияние на дизайнеров и выявляют по крайней мере некоторые из недостатков в дизайне. Эмпирические методы также выгодно применяются для неофициальных оценок ранних прототипов и строгих приемочных испытаний перед доставкой.

Хорошим видением успешного дизайна пользовательского интерфейса является то, что идиосинкразии, связанные с компьютером, исчезают, и пользователи могут сосредоточиться на своих задачах. Дизайнеры стремятся уменьшить нагрузку на сложный синтаксис и неудобные компьютерные концепции, но многое остается. Эмпирические исследования дают много важных небольших результатов, обеспечивают основу для совершенствования возникающих теорий и вносят вклад в практические рекомендации. Разработчики коммерческих систем применяют рекомендации и свою интуицию для создания элегантных систем, но должны тщательно протестировать свои проекты у реальных пользователей.

### **Список источников**

1. J.D. Foley, "Interfaces for Advanced Computing," *Scientific American*, vol. 257, no. 4, pp. 127-135, October 1987.
2. J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, Mass., 1990.
3. H.R. Hartson and D. Hix, "Human-computer Interface Development: Concepts and Systems for its Management," *Computing Surveys*, vol. 21, no. 1, pp. 5-92, 1989.
4. E.L. Hutchins, J.D. Hollan, and D.A. Norman, "Direct Manipulation Interfaces," in *User Centered System Design: New Perspectives on Human-computer Interaction*, ed. by D.A. Norman and S.W. Draper, pp. 87-124, Lawrence Erlbaum, Hillsdale, N.J., 1986.
5. R.J.K. Jacob, "A Specification Language for Direct Manipulation User Interfaces," *ACM Transactions on Graphics*, vol. 5, no. 4, pp. 283-317, 1986.
6. J. Johnson and others, "The Xerox Star: A Retrospective," *IEEE Computer*, vol. 22, no. 9, pp.11-29, 1989.
7. B.A. Myers, "User Interface Software Tools," *ACM Transactions on Computer-Human Interaction*, vol. 2, no. 1, pp. 64-103, March 1995.
8. D.R. Olsen, *User Interface Management Systems: Models and Algorithms*, Morgan Kaufmann, San Francisco, 1992.
9. B. Shneiderman, "Direct Manipulation: A Step Beyond Programming Languages," *IEEE Computer*, vol. 16, no. 8, pp. 57-69, 1983.
10. B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Second Edition, Addison-Wesley, Reading, Mass., 1992.