**АНАЛИЗ ИНТЕРФЕЙСА ПРОГРАММНОГО ПРОДУКТА**

Специалисты обычно формулируют некоторый набор принципов и правил, позволяющих как оценивать удобство интерфейса, так и предлагать решения, повышающие его удобство. Приведем эти правила. Правило доступности. Система должна быть настолько понятной, чтобы пользователь, никогда раньше не видевший ее, но хорошо разбирающийся в предметной области, мог без всякого обучения начать ее использовать. Это правило служит некоторым идеалом, к которому надо стремиться, поскольку на практике достичь такой степени доступности почти никогда не удается. Тем не менее, специалисты продолжают делать в этом направлении все возможное. Правило эффективности. Система не должна препятствовать эффективной работе опытных пользователей, работающих с ней долгое время. Очевидным примером нарушения этого правила является нацеленность системы только на новичков, использование средств, которые хорошо подходят для неопытного пользователя, ограничивая его в возможности сделать что-то не так, но неэффективны для эксперта, который и так знает, что и где ему нужно сделать. Правило непрерывного развития. Система должна способствовать непрерывному росту знаний, умений и навыков пользователя и приспосабливаться к его меняющемуся опыту. Плохие результаты приносит предоставление только базовых возможностей или оставление начинающего пользователя наедине со сложным интерфейсом, которым уверенно пользуются эксперты. Нарушение непрерывности при переходе от одного набора возможностей к другому также приносит неудобства, поскольку пользователь вынужден разбираться с добавленными возможностями в новом контексте.

Большинство пользователей можно поместить в три группы: новички, опытные и средние, которые уже знают больше, чем новички, и не делают столько ошибок, но еще не приобрели автоматизма при выполнении большинства операций и иногда путаются в интерфейсе. Новичкам необходима помощь в освоении новой для них системы и контроль за их действиями, опытным пользователям — высокая эффективность выполнения часто требующихся действий и возможность гибкого управления системой в таких ситуациях, которые встречаются реже, но способны вызвать проблемы при их неадекватной поддержке. Про средних же пользователей часто забывают, хотя подавляющее большинство пользователей программного обеспечения относится именно к этой категории. Им нужны достаточно высокие эффективность и гибкость вместе с возможностью быстро получать адекватную помощь по возникающим время от времени разнообразным вопросам. Правило соблюдения контекста. Система должна быть согласована с контекстом, в котором ей предстоит работать. Это правило требует от системы быть работоспособной не «вообще», а именно в том окружении, в котором ею будут пользоваться. В контекст могут входить специфика и объемы входных и выходных данных, тип и цели организаций, в которых система должна работать, уровень пользователей, зашумленность помещений и пр.

**ПРИНЦИПЫ РАЗРАБОТКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА**

Представленные выше правила определяют общие требования, которым должен удовлетворять удобный интерфейс. Следующие принципы позволяют находить решения, повышающие удобство пользовательского интерфейса. Принцип структуризации. Пользовательский интерфейс должен быть целесообразно структурирован. Близкие по смыслу, родственные его части должны быть связаны видимым образом, а независимые — разделены; похожие элементы должны выглядеть похоже, а непохожие — различаться. Принцип простоты. Наиболее распространенные операции должны выполняться максимально просто. При этом должны быть видимые ссылки на более сложные процедуры. 6.2. Принцип видимости. Все функции и данные, необходимые для решения определенной задачи, должны быть видны, когда пользователь пытается ее решить. Принцип обратной связи. Пользователь должен получать сообщения о действиях системы и о важных событиях внутри нее. Сообщения должны быть информативными, краткими, однозначными и написанными на языке, понятном пользователю. Принцип толерантности. Интерфейс должен быть гибким и терпимым к ошибкам пользователя. Ущерб от ошибок должен снижаться за счет возможности отмены и повтора действий и за счет разумной интерпретации любых разумных действий пользователя и введенных им данных. По возможности следует избегать обязывающего взаимодействия (модальных диалогов), основанного на ограничении свободы пользователя. Принцип повторного использования. Следует стараться многократно использовать внутренние и внешние компоненты, обеспечивая тем самым унифицированность интерфейса и сходство между его похожими элементами. Рассмотрим некоторые вопросы разработки эргономичного интерфейса пользователя.

**ВЗАИМОДЕЙСТВИЕ МЕЖДУ ПОЛЬЗОВАТЕЛЕМ И КОМПЬЮТЕРОМ**

Человекомашинный интерфейс обеспечивает связь между пользователем и компьютером, он позволяет достигать поставленных целей, успешно находить решение поставленной задачи. Взаимодействие — обмен действиями и реакциями на эти действия между компьютером и пользователем. Существует ряд стилей взаимодействий, которые подразделяются на два основных вида. 1. Использование интерфейса языка команд — ввод команд текстовыми средствами. 2. Непосредственное манипулирование. Таким образом, имеется ряд способов, которыми пользователь мог бы связываться с компьютером:

■ языки команд — пользователь управляет системой, вводя соответствующие команды в текстовом режиме;

■ вопрос и ответ — диалог, где компьютер задает вопросы, а пользователь отвечает ему (или наоборот);

■ формы — пользователь заполняет формы или поля диалога, вводя данные в соответствующие поля;

■ меню — пользователь обеспечен рядом опций и управляет системой, выбирая необходимые пункты;

■ прямое манипулирование — пользователь управляет объектами на экране посредством устройства манипулирования типа мыши.

В разрабатываемой программной системе также применен комплексный подход к созданию интерфейса. Здесь используется прямое манипулирование, меню, формы и диалоги. Цель создания эргономичного интерфейса состоит в том, чтобы отобразить информацию настолько эффективно, насколько это возможно для человеческого восприятия, и структурировать отображение на дисплее таким образом, чтобы привлечь внимание к наиболее важным единицам информации. Основная же цель в том, чтобы минимизировать общую информацию на экране и представить только то, что является необходимым для пользователя.

**РАЗМЕЩЕНИЕ ИНФОРМАЦИИ НА ЭКРАНЕ**

Количество информации, отображаемой на экране, называется экранной плотностью. Исследования показали, что чем меньше экранная плотность, тем отображаемая информация наиболее доступна и понятна для пользователя, и наоборот, если экранная плотность большая, это может вызвать затруднения в усвоении информации и ее ясном понимании. Однако опытные пользователи могут предпочитать интерфейсы с большой экранной плотностью. Информация на экране может быть сгруппирована и упорядочена в значимые части. Это может быть достигнуто использованием кадров (фреймов), методов типа цветового кодирования, рамок, негативного изображения или других методов для привлечения внимания. Выделение элементов интерфейса яркостью. Для привлечения внимания к каким-либо элементам интерфейса можно воспользоваться выделением этих элементов большей яркостью на фоне других, более темных. Однако важно не переусердствовать с этим методом, поскольку большое количество ярких элементов способно вызвать дискомфорт у пользователя и можно получить обратный эффект — перегрузку интерфейса. Применять этот метод следует только при необходимости. Существует несколько способов выделения яркостью:

■ движение (мигание или изменение позиции) — очень эффективный метод, поскольку глаз имеет специальный детектор движущихся элементов;

■ яркость — не очень эффективный метод, так как люди способны различить лишь несколько уровней яркости;

■ цвет — использование может быть чрезвычайно эффективным;

■ форма (символ, шрифт, форма символа) используется для того, чтобы дифференцировать различные категории данных;

■ использование разных алфавитов (шрифтов) в раличных формах;

■ размер (текста, символов) — обычно применяют увеличение выделенного объекта в 1,5 раза;

■ оттенение (различная текстура объектов) — эффективный метод для привлечения внимания к какой-либо части экрана;

■ окружение (подчеркивание, рамки, инвертированное изображение).

Очень эффективный способ, если не переусердствовать. Использование цвета при проектировании эргономичного интерфейса. Цвет может улучшить интерфейс, но для многих систем использование цвета практически не влияет на эффективность работы пользователя. Основное назначение цвета — создание интерфейсов, более интересных для пользователей; однако имеются случаи, когда цвет может помочь проектировщику интерфейса. Особенно это эффективно при группировке информации, выделении различий между информацией или простых сообщений (ошибки, состояния и т.д.). Цвет — мощный визуальный инструмент, и применять его надо очень осторожно, чтобы не вызвать у пользователя дискомфорт от ошибочных цветовых комбинаций.

Перечислим некоторые принципы использования цвета, которыми нужно руководствоваться при проектировании эргономичного интерфейса:

■ необходимо ограничить число цветов до 4 на экране и до 7 — для последовательности экранов;

■ для неактивных элементов лучше использовать бледные цвета;

■ если цвет служит для кодировки информации, следует удостовериться, что пользователь правильно понимает код; например, неоплаченные счета выделяются красным цветом;

■ важно при использовании цвета учитывать представления пользователя; например, для картографа зеленый — это лес, желтый — пустыня, синий — вода. Для химика красный означает горячий, синий — холодный и т.д.;

■ при отображении состояния, как правило, красный означает опасность (стоп), зеленый — продолжение работы, желтый — цвет предупреждения;

■ для привлечения внимания наиболее эффективны белый, желтый и красный цвета;

■ при необходимости упорядочения данных можно использовать спектр 7 цветов (радуга);

■ если надо разделить данные, то цвета выбирают из различных частей спектра: красный — зеленый, синий — желтый, любой цвет— белый;

■ для группировки данных, объединения и подобия нужно использовать соседние цвета спектра: оранжевые— желтые, синие— фиолетовые. Важно отметить, что около 9 % людей не различают цветов (обычно, красно-зеленые сочетания); однако эти люди могут отличать черно-белые оттенки, поэтому проектировщики автоматизированных систем должны проверять, не нарушает ли использование различных цветов в интерфейсах восприятия пользователей этой категории. Непротиворечивость и стандартизация. Данные на экране следует располагать таким образом, чтобы пользователь знал, где найти и где ожидать вывода необходимой информации:

■ информация, на которую следует немедленно обратить внимание, должна всегда отображаться на видном месте, чтобы захватить внимание пользователя (например, предупреждающие сообщения и сообщения об ошибках);

■ не очень часто востребуемая информация (например, справка) не должна отображаться, но должна быть доступна по необходимости. Например, иконка «справки» или соответствующая опция меню должна быть доступна на каждом экране. 90 Тексты и диалоги. Приведем некоторые принципы, которыми необходимо руководствоваться при создании текстовых диалогов и отображений:

■ текст в нижнем регистре читается приблизительно на 13% быстрее, чем текст, напечатанный полностью в верхнем регистре;

■ символы верхнего регистра наиболее эффективны для передачи информации, которая должна привлечь внимание. Не используйте верхний регистр, если вы не хотите выделять какую-либо информацию;

■ выровненный по правому краю текст труднее читать, чем равномерно распределенный текст с невыровненным правым полем;

■ оптимальный интервал между строками равен или немного больше, чем высота символов.

Меню. Необходимый элемент автоматизированной системы — меню, позволяющее пользователю выполнять задачи внутри приложения и управлять процессом решения. Меню — набор опций, отображаемых на экране, где пользователи могут выбирать и выполнять действия, тем самым производя изменения в состоянии интерфейса. Достоинство меню в том, что пользователи не должны помнить название элемента или действия, которое они хотят выполнить, они должны только распознать его среди пунктов меню. Таким образом, меню может использовать даже неопытный пользователь. Однако проект меню должен быть тщательно продуман — чтобы меню было эффективным, названия его пунктов должны быть очевидными. Меню может занимать много экранного места, но решением этой проблемы может быть использование всплывающего или ниспадающего меню, которое вызывается щелчком по пиктограмме, строке меню или другому объекту. Основные принципы создания меню. В процессе проектирования системы меню приложения необходимо принять наилучший способ отображения меню, чтобы оно было понятным и легким в использовании. Обычно команды меню упорядочены некоторым иерархическим способом. Основная проблема состоит в том, чтобы правильно распределить различные пункты меню по различным уровням и правильно их сгруппировать.

Принципы проектирования меню:

■ структура меню должна соответствовать структуре решаемой системой задачи; организация меню должна отра­ зить наиболее эффективную последовательность шагов, ведущих к решению поставленной задачи;

■ пункты меню должны быть краткими, грамматически правильными и соответствовать своему заголовку. Порядок пунктов меню выбирается в соответствии с соглашением, частотой и порядком использования, а также в зависимости от потребностей задачи или пользователя;

■ выбор пунктов меню должен быть обеспечен несколькими способами — с помощью клавиатуры, с помощью мыши и через другие объекты пользовательского интерфейса. Важно зафиксировать легко запоминаемые сочетания клавиш для более быстрого доступа к пунктам меню, поскольку это очень экономит время. Формы. Формы — основной элемент интерфейса. Назначение форм — удобный ввод и просмотр данных, состояния, сообщений автоматизированной системы. Основные принципы проектирования форм:

■ форма проектируется для более удобного, понятного и быстрого решения поставленной задачи. Если форма переносится из бумажной формы, то передвижение по смежным полям не должно вызывать затруднений у пользователя;

■ размещение информационных единиц на пространстве формы должно соответствовать логике ее будущего использования: это зависит от необходимой последовательности доступа к информационным единицам, частоты их использования, а также от относительной важности элементов;

■ важно использовать незаполненное пространство, чтобы создать равновесие и симметрию среди информационных элементов формы, для фиксации внимания пользователя в нужном направлении;

■ логические группы элементов необходимо отделять пробелами, строками, цветовыми или другими визуальными средствами;

■ взаимозависимые или связанные элементы должны отображаться в одной форме. При разработке форм необходимо продумать и указать, какие кнопки в полосе системного меню должны быть доступны в том или ином окне, должно ли окно допускать изменения пользователем его размера, каким должен быть заголовок окна. Без особой необходимости не нужно делать окна с изменяемыми размерами. При изменении размеров, если не применены специальные приемы, нарушается компоновка окна, что негативно сказывается на работе пользователя. Имеет смысл создать окно с изменяемыми размерами, если это позволяет пользователю изменять полезную площадь расположенных в нем компонентов отображения и редактировать информацию: текст, изображение, списки и т.д. Если проектируется изменяемое окно, то необходимо принять меры, чтобы компоненты в окне при этом тоже изменяли свои размеры или местоположение, равномерно распределяясь по площади окна и не оставляли пустых мест. При проектировании форм необходимо стремиться к использованию ограниченного набора цветов и уделять внимание их правильному сочетанию. Для фона формы избираются нейтральные цвета (светло-серые). Цвет не должен использоваться в качестве основного средства передачи информации, надо выбирать системные цвета, которые пользователь может перестраивать по своему усмотрению. Управляющие элементы и функционально связанные с ними компоненты экрана следует зрительно объединять в группы, заголовки которых коротко и четко поясняют их назначение. Каждое окно должно иметь некоторую центральную тему, которая подчиняется его композиции. Пользователь должен понимать, для чего предназначено данное окно и что в нем наиболее важно. Недопустимо перегружать окно большим числом элементов управления ввода и отображения информации. Также недопустимо, чтобы сходные по функциям органы управления в разных окнах назывались по-разному или размещались в разных местах окон. Важно позаботиться о том, как приложение впишется в общую организацию рабочего пространства системы и как оно будет взаимодействовать с другими приложениями. Дизайн заголовков и полей. Для отдельных полей заголовок должен быть выровнен по левому краю; для полей списков заголовок должен быть выше и левее по отношению к основному полю; числовые поля выравниваются по правому полю. Длинные колоночные поля или длинные столбцы информационных единиц с одиночными полями необходимо объединять в группы по пять элементов, разделяемых пустой строкой, — это помогает пользователю мысленно обрабатывать информацию по выделенным группам. В формах с большим количеством информации важно использовать названия разделов, которые однозначно свидетельствуют о характере принадлежащей им информации. Необходимо четко разделить отображение заголовков и непосредственно полей ввода. Заголовки должны быть краткими, знакомыми и содержательными. Поля, необязательные для заполнения либо не имеющие особой важности, должны отличаться визуально (цветом или другими эффектами) от полей важных и обязательных для заполнения. Форматы ввода. Следует обеспечить ввод значений по умолчанию во все поля, которые это допускают и где такая функция не будет раздражать пользователя. Можно назначить клавиши или коды для ввода часто повторяющихся значений. Входные данные должны быть значимыми и общепринятыми. Не следует объединять поля ввода чисел и символов, поскольку числовые и алфавитные клавиши находятся неудобно относительно друг друга на клавиатуре. Необходимо исключить частое переключение между верхним и нижним регистрами для ускорения ввода данных. Нельзя требовать от пользователя ввода незначимых цифр (например, вместо 00000010 пользователь должен ввести только 10). Аналогично, нельзя требовать от пользователя ввести информацию, которая была предварительно введена или которая может быть автоматически получена из системы. Желательно использовать значения по умолчанию, чтобы минимизировать процесс ввода информации. Организация системы навигации и системы отображения состояний. Навигация обеспечивает пользователю возможность перемещаться между различными экранами, информационными единицами и подпрограммами в автоматизированной системе. В полноценной системе пользователь всегда может получить информацию о состоянии системы, о процессе выполнения или активной подпрограмме. Общие принципы проектирования. Существует ряд навигационных средств и приемов, которые помогают пользователю ориентироваться в системе. Они включают использование заголовков страниц для каждого экрана, номеров страниц, строк и столбцов, отображение текущего имени файла вверху экрана. Тип системы навигации зависит от принятого стиля интерфейса. Для интерфейсов языка команд существует очень мало способов обеспечения полноценной навигации. В интерфейсах с меню можно использовать иерархически структурированное меню. Диалоговые интерфейсы сами по себе защищают пользователя от ошибочных действий. Информация состояния обычно отображается внизу экрана и содержит в себе данные о количестве запи94 сей, числе обработанных единиц, процессе печати, очереди печати и т.д. Проектирование сообщений. Сообщения необходимы для направления действий пользователя в нужную сторону, подсказок и предупреждений при выполнении необходимых действий на пути решения задачи. Они также включают в себя подтверждения действий со стороны пользователя и подтверждения со стороны системы, что задачи выполнены успешно либо по каким-то причинам не выполнены. Сообщения могут быть выведены в форме диалога, экранных заставок и т.п.

Сообщения могут предложить пользователю:

■ выбрать из предложенных альтернатив опцию или набор опций;

■ ввести информацию;

■ выбрать опцию из набора опций, которые могут изменяться в зависимости от текущего контекста;

■ подтвердить фрагмент введенной информации перед продолжением ввода. Сообщения могут быть помещены в модальные диалоговые окна, которые вынуждают пользователя ответить на вопрос прежде, чем начнет выполняться любое другое действие. Это может быть полезно, когда система принуждена заставить пользователя обдумать решение перед продолжением работы. Немодальные диалоговые окна позволяют работать с другими элементами интерфейса, в то время как само окно может игнорироваться.

**ПРЕДОТВРАЩЕНИЕ, ОБНАРУЖЕНИЕ И ИСПРАВЛЕНИЕ ОШИБОК**

Обычный человек в нормальном состоянии совершает множество ошибок разного рода. Можно сказать, что человек, в отличие от компьютера, является адаптивной аналоговой системой и успешность его «функционирования» в гораздо большей степени определяется не точностью выполнения действий и формулировки мыслей, а способностью быстро выдать хорошее приближение к нужному результату и достаточно быстро поправиться, если это необходимо. Ошибки пользователя могут быть основаны на неправильном понимании действия или порядка действий или быть случайными, непреднамеренными, например опечатка при вводе текста.

Ошибки второго вида могут быть разделены еще на шесть подвидов:

■ неточность в выборе опции (например, пользователь случайно нажал кнопку «Выход» и программа закрылась);

■ ошибки при управлении данными (например, присвоение ошибочного имени файла из-за неточности отображения последнего);

■ ошибки ассоциативного характера (например, сохранение файла с именем какого-либо человека, так как пользователь думал о нем в момент сохранения);

■ потеря активности, когда пользователь забывает необходимую последовательность действий для продолжения работы;

■ ошибка режима или состояния, когда пользователь думает, что он находится в одном состоянии, а фактически — другом; например, режим вставки взамен режима печати поверх текста в текстовом процессоре. Пользователь всегда будет делать ошибки даже в отличной программной системе, поэтому в разрабатываемой системе всегда должна быть предусмотрена защита от ошибок. Техника такой защиты включает в себя следующие аспекты:

■ принудительные действия в системе, которые предотвращают или затрудняют появление ошибок;

■ обеспечение хороших и информативных сообщений об ошибках;

■ использование обратимых действий, позволяющих пользователям исправлять их собственные ошибки;

■ обеспечение нормальной диагностики системы, в процессе которой пользователю объясняется, в чем суть ошибки, и указываются пути ее исправления. Рассмотрим основные принципы обработки ошибок в формах ввода:

■ обеспечить возможность посимвольного редактирования введенных записей для исправления ошибок ввода (опечаток);

■ если ошибка обнаружена системой, желательно вернуть курсор в поле с ошибочными данными и каким-либо образом выделить это поле визуально;

■ выводить значимые сообщения об ошибках, использующие стиль языка пользователя и соответствующую терминологию;

■ выводить сообщения об ошибках, которые объясняют и предлагают пути ее устранения.

Эффективность предотвращения и преодоления ошибок пользователей тем выше, чем реже пользователи ошибаются при работе с данным интерфейсом и чем меньше времени и усилий требуется для преодоления последствий уже сделанных ошибок.

**ОБЩИЕ ТРЕБОВАНИЯ К ГРАФИЧЕСКОМУ ИНТЕРФЕЙСУ ПОЛЬЗОВАТЕЛЯ**

Под графическим интерфейсом пользователя подразумевается тип экранного представления, при котором пользователь может выбирать команды, запускать задачи и просматривать в списке файлы, указывая на пиктограммы или пункты списков меню, показанных на экране (например, система Windows).

Графический интерфейс пользователя любой программы должен включать в себя:

■ главное меню;

■ инструментальную панель быстрых кнопок, дублирующих основные разделы меню;

■ контекстное меню, всплывающее при щелчке пользователя правой кнопкой мыши на том или ином компоненте;

■ продуманную последовательность переключения фокуса управляющих элементов;

■ клавиши быстрого доступа ко всем разделам меню и всем управляющим элементам, горячие клавиши для доступа к основным командам;

■ ярлычки подсказок, всплывающие при перемещении курсора мыши над быстрыми кнопками и иными компонентами;

■ полосу состояния, используемую для развернутых подсказок и выдачи различной информации пользователю;

■ файл справки, темы которого отображаются при нажатии клавиши F1 или при выборе пользователем соответствующего раздела меню;

■ информацию о версии, доступную пользователю при щелчке на пиктограмме приложения правой кнопкой мыши;

■ возможность настройки приложения и запоминания настроек, чтобы при очередном сеансе работы восстанавливались настройки, установленные в прошлом сеансе;

■ средства установки приложения, регистрация его в Windows и удаление из Windows.

**ЗАДАНИЕ**

1. Провести анализ эргономичности графического интерфейса пользователя по прототипам, созданным в предыдущей работе или уже готового проекта. Составить таблицу соответствия (несоответствия) разработанного пользовательского интерфейса требованиям стандартного графического интерфейса пользователя. Подготовить рекомендации по исправлению выявленных нарушений. (Желательно анализировать интерфейс не своей программы, а другого студента.);

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Перечислите общие принципы и правила создания удобного интерфейса.

2. Как используется цвет при разработке эргономичного интерфейса?

3. Перечислите основные требования к компоновке форм.

4. Как можно предотвратить появление ошибок при вводе?

5. Как правильно создать меню?