Технические науки

УДК 004.422.8
Т.М. Салий, кандидат педагогических наук
Инновационный Евразийский университет (г. Павлодар),
З.Б. Баженеева
Инновационный Евразийский университет (г. Павлодар)
E-mail: toma_sal@mail.ru, zere.bazheneyeva@mail.ru

Принципы создания пользовательского интерфейса

Аннотация. *В статье рассматриваются принципы разработки графического пользовательского интерфейса. Описывается возможность создания удобной и понятной пользователю модели взаимодействия с программным обеспечением без необходимости изучения какого-либо специального языка. Авторы, основываясь на анализ разработок пользовательского интерфейса, указывают на проблему по переносимости программного обеспечения на другие операционные системы, так как графический интерфейс существенно зависит от возможностей, предоставляемых операционной системой для его создания. Статья посвящена анализу принципов создания пользовательского интерфейса. Предпринята попытка выявления факторов, являющихся обязательным условием создания эффективной модели пользовательского интерфейса.*

Ключевые слова: *программирование, компоненты, команды, процедуры, запросы, программное обеспечение.*

Существует множество различных стилей пользовательских интерфейсов. При их разработке используются разнообразные принципы и подходы. Рассмотрим два важнейших принципа, соблюдение которых рекомендуется при разработке интерфейсов [1]:

– пользовательский интерфейс должен базироваться на терминах и понятиях, знакомых пользователю;

– пользовательский интерфейс должен быть единообразным.

В основном используются командные и графические пользовательские интерфейсы.

При использовании *командного пользовательского интерфейса* пользователь имеет возможность обращаться к подсистеме (ПС) с некоторым запросом, представляемым командой. Преимуществом данного интерфейса – возможность минимизации требуемого от пользователя ввода с клавиатуры. Минусом данного интерфейса можно считать обязательное изучение командного языка и высокая вероятность ошибки при написании команды. Поэтому данный пользовательский интерфейс выбирают более опытные пользователи. С таким интерфейсом пользователь может взаимодействовать с компьютером быстрее и имеет возможность объединять команды в процедуры и программы.

Преимущество *графического пользовательского интерфейса* – возможность создания удобной и понятной пользователю модели взаимодействия с ПС. При этом пользователь может не изучать какой-либо язык программирования. При разработке такого интерфейса необходимы большие трудозатраты. Их можно сравнить с разработкой программного обеспечения. Кроме того, возникает серьезная проблема по переносимости ПС на другие операционные системы, так как графический интерфейс существенно зависит от возможностей, предоставляемых операционной системой для его создания.

Графический пользовательский интерфейс обобщает такие виды пользовательского интерфейса, как интерфейс типа меню. Для реализации первого принципа нужно знать последовательность переключения компонентов. Большинство пользователей предпочитают переключаться между полями ввода посредством использования клавиатуры клавишей Tab, а не мышкой. Это намного эффективней, чем выделять каждое поле мышью. Поэтому последовательность переключения компонентов должна соответствовать правильной очередности. Это имеет отношение к компонентам внутри компонентов-контейнеров (панель, GroupBox и т.п.), так и самих компонентов-контейнеров, если их на форме несколько.

Очередность переключения компонентов внутри контейнера осуществляет свойство TabOrder. Если TabOrder равняется 0, то этот компонент активный, вторым активным компонентом является компонент с TabOrder равным 1 и т. д., пока все компоненты не будут перебраны. У компонента также есть свойство TabStop. Оно демонстрирует получение фокуса для компонента при переключении клавишей Tab. При необходимости запрета переключения одного из компонентов, необходимо поставить TabStop = false. Тогда переключение на данный компонент осуществляется при использовании мыши. Пользователи, которые привыкли переключаться какой-либо определенной клавишей в одной программе, имеют привычку пользоваться ей и в остальных программах. Чаще всех это делают

пользователи 1С, где для перехода по полям ввода может использоваться клавиша Enter. Для этого необходимо установить у формы свойство KeyPreview в true и поместить в обработчик события OnKeyPress: (рисунок 1):

```
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin
 if ord(key)=vk_Return then
  Form1.SelectNext(PriemForm.ActiveControl, true, true);
end;
```

Рисунок 1 – Обработчик события OnKeyPress

Представленный обработчик события может осуществлять переход по элементам формы с помощью клавиши Enter. Этот метод не позволяет действия с кнопками, так как использование Enter приводит к ее нажатию, а при нажатии на клавишу Tab фокус ввода передается следующему в последовательности переключения компоненту.

Надо заметить, что пользователи быстро привыкают к тому, что в диалоговых окнах приложений есть кнопки по умолчанию. Например, клавиша Enter подтверждает свой выбор, клавиша Esc отменяет выбор. Для этого для кнопки, реагирующей на Enter, устанавливаем свойство Default в true. Для кнопки, реагирующей на Esc, устанавливаем свойство Cancel в true.

Второй принцип – пользовательский интерфейс может быть единообразным. По данному принципу диалоговое окно должно иметь не меньше двух кнопок (сохранить/отменить, да/нет, и т.п.), которые запрашивают действия пользователя: отказ действия или подтверждение. С помощью кнопки [X] в заголовке окна можно отказаться от какого-либо действия. Недопустимо, если возможность отказа вообще отсутствует, или имеется только одна кнопка для подтверждения действия, а для отказа предполагается закрывать в заголовке окна кнопкой [X]. Это путает пользователя, и он не может понять, как отказаться от действия. Диалоговые окна необходимо располагать по центру экрана, а не там, где они были разработаны в режиме проектирования.

Отметим, что это нагляднее и устраняет проблему разного разрешения экрана у разных пользователей. Не допускается превышение размеров окон по сравнению с размером экрана. Уменьшение же размеров окна иногда приводит к исчезновению оконных элементов, что не допустимо.

Допускается использование закладки (типа PageControl), чтобы разбить компоненты на группы, в том случае если не получается разместить компоненты в одном окне. Чтобы понять предназначение какой-либо кнопки на панели инструментов (типа ToolBar) нужно задать подсказки (hint).

Форма в большинстве случаев может быть прямоугольником или кругом. Часто используемые основные элементы интерфейса должны быть выделены, например, размером или цветом. Иконки в программе должны быть очевидными. Если такого нет, то их необходимо подписать. Пиктограммы зачастую сами требуют для себя объяснений вместо того, чтобы объяснять (рисунок 2).

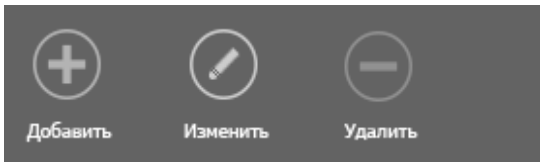

Рисунок 2 – Основные элементы интерфейса

При проектировании интерфейса большое значение имеет использование цветов. Такие цвета, как желтый, оранжевый, красный являются теплыми. Холодными цветами считаются синий и зеленый, а нейтральным цветом принято считать серый. Обычно для элементов интерфейса используют теплые цвета. Это связано с психологией восприятия. Конечно, мнение о цвете очень субъективно и каждый пользователь может менять его, даже в зависимости от своего настроения.

Удобство использования интерфейса пользователем является одним из важных показателей качества программного обеспечения. Оно описывается с помощью таких характеристик, как легкость обучения работе с ним, понятность пользовательского интерфейса, производительность работы пользователя с программным обеспечением, трудоемкость решения определенных задач с его помощью, частота появления ошибок и жалоб на неудобства. Для построения действительно удобных программ нужен учет контекста их использования, психологии пользователей, необходимости помогать начинающим пользователям и предоставлять все нужное для работы опытных. Самым значимым фактором является то, действительно ли данная программа помогает решать задачи для пользователей.

В силу большого разнообразия видов ПС и пользователей существует множество различных стилей пользовательских интерфейсов. Могут использоваться разные подходы и принципы при разработке пользовательских интерфейсов. Рассмотрим следующие принципы, которые также следует соблюдать при проектирования интерфейса, ориентированного на пользователя:

– *Контроль* – на стороне пользователя. Смысл данного принципа заключается в том, что пользователь инициирует какое-либо действие, и если в результате этого контроль переходит к программе, то пользователь получает необходимую *обратную связь* (в виде курсора в форме песочных часов, индикатора ожидания).

– *Индивидуализация и настройка* – два взаимосвязанных принципа разработки пользовательского интерфейса. Примером индивидуализации является изменение пользователем порядка и размеров колонки в программе просмотра строк (так называемые *сетки – grid)* с последующим сохранением этих изменений как его личных предпочтений. При обращении к этой же программе в будущем данные предпочтения учитываются. Примером настройки является различие в функционировании программы по отношению к пользователям-новичкам и опытным пользователям. Например, пользователю-новичку программа может предложить явную помощь и дополнительные предупреждающие сообщения, которые указывают на потенциальную опасность некоторых событий, инициируемых пользователем.

– *Согласованность*. При разработке качественного интерфейса этот принцип является основным. Согласованность означает фактически соблюдение стандартов и следование некоторым общепринятым правилам работы с пользовательским интерфейсом. Данный принцип может рассматриваться, по крайней мере, в двух направлениях: соответствие стандартам в области именования, программирования и соответствие стандартам поставщика пользовательского интерфейса, которые связаны с пользовательским интерфейсом. Оба направления одинаково важны. Если приложение разрабатывается для Windows, то следует обеспечить «ощущение и впечатление», которые свойственны при работе в системе Windows [2]. Разработчик пользовательского интерфейса не должен предлагать необычные новшества и слишком увлекаться творчеством. Это может плохо сказаться на умении и уверенности пользователей. Пользователям необходимо представлять среду, которая знакома им и поведение которой предсказуемо. Следует отметить также необходимость соответствия внутренним стандартам в области именования, программирования, аббревиатур и т.п. Сюда относятся именование и программирование меню, командных кнопок, полей экранов. Также необходимо отнести стандарты по расположению объектов на экране и последовательному использованию элементов пользовательского интерфейса в рамках всех приложений, разрабатываемых собственными силами.

– *Обратная связь.* Принцип обратной связи дополняет первый принцип – контроль должен находиться на стороне пользователя. Контролировать ситуацию – значит знать, что происходит, когда контроль временно передается программе. Разработчик должен встроить в систему для каждого события, инициируемого пользователем аудиоподсказки или визуальные подсказки. Индикатор ожидания или указатель в форме песочных часов в большинстве случаев представляет достаточный уровень обратной связи, чтобы пользователь мог понять, что программа что-то делает. Может потребоваться более ясная обратная связь (в виде отображения поясняющего сообщения), если некоторые компоненты приложения в определенный момент времени стали источником проблем с производительностью.

– *Терпимость к ошибкам.* Терпимость к ошибкам подразумевает многоуровневую систему отмены операций. Хорошо спроектированный интерфейс должен позволять пользователям совершать ошибки, экспериментировать и проявлять терпимость к ошибкам. Это позволяет пользователю выполнять ошибочные последовательности действий с возможностью в любой момент совершить при необходимости «откат» в начало. Подобная терпимость стимулирует исследовательскую активность пользователя.

– *Эстетичность и удобство.* На зрительное восприятие системы влияет эстетичность интерфейса. Удобство касается простоты, надежности, эффективности и продуктивности в использовании интерфейса. Оба принципа, конечно, касаются удовлетворенности пользователя. В этом сложном вопросе разработчик пользовательского интерфейса нуждается в помощи художника-графика и эксперта по социальной психологии. В настоящее время существует достаточно много разнообразных «золотых правил», которые касаются создания удобного и эстетичного интерфейса. При разработке интерфейса следует учитывать такие вопросы как движение человеческого глаза и фиксация, чувство уравновешенности и симметрии, чувство пропорции, выравнивание и отступ между элементами, группирование связанных элементов, использование цветов и т.д. В этом смысле неплохо помнить о том, что «простота – эталон красоты». В действительности «простоту» часто рассматривают как еще один принцип создания пользовательского интерфейса, прочно связанного с принципами эстетичности и удобства. В сложных приложениях простота лучше всего достигается за счет последовательного раскрытия информации таким образом, что она отображается только тогда, когда в ней возникает необходимость, возможно в различных окнах. Принцип эстетичности и удобства превращает разработчика пользовательского интерфейса в художника.

Модель содержимого пользовательского интерфейса описывает набор взаимосвязанных контекстов взаимодействия или рабочих пространств (представляемых экранами, формами, окнами, диалогами, страницами и пр.) с содержащимися в них данными и возможными в их рамках действиями.

При построении этой модели нужно определить, что войдет в состав интерфейса (какие данные и функции), и не решать вопрос о том, как именно оно будет выглядеть [3].

Пользовательский интерфейс – это достаточно обширная тема. Она тесно переплетенная с психологией. И если соблюдать базовые принципы, то это позволит упростить сам процесс проектирования, а так же строить интерфейсы более дружелюбными к пользователю.

СПИСОК ЛИТЕРАТУРЫ

1 Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. – М., СПб.: Вильямс, 2012. – 432 с.

2 Петцольд Ч. Программирование с использованием Microsoft Windows Forms : [пер. с англ.] – М.: Русская редакция. – СПб.: Питер, 2006. – 410 с.

3 Вендров А. М. Проектирование программного обеспечения. М.: Финансы и статистика, 2012. – 352 с.

REFERENCES

1 Macyashek L.A. Analiz trebovanij i proektirovanie sistem. Razrabotka informacionnyh sistem s ispol'zovaniem UML. – M., SPb: Vil'yams, 2012. – 432 s.

2 Petcold C. Programmirovanie s ispol'zovaniem Microsoft Windows Forms: [per. s angl.] – M.: Russkaya redakciya. – SPb.: Piter, 2006. – 410 s.

3 Vendrov A.M. Proektirovanie programmnogo obespecheniya. – M.: Finansy i statistika, 2012. – 352 s.

ТҮЙІН

***Т.М. Салий**, педагогика ғылымдарының кандидаты*
Инновациялық Еуразия университеті (Павлодар қ.),
З.Б. Баженеева
Инновациялық Еуразия университеті (Павлодар қ.)

Пайдаланушы интерфейсін құру принциптері

Мақалада принциптері графикалық пайдаланушы интерфейсі. Сипатталады құру мүмкіндігі, ыңғайлы және түсінікті пайдаланушы моделін өзара іс-қимыл бағдарламалық қамтамасыз ету қажеттілігі жоқ зерделеу қандай да бір арнайы тілі. Авторлар негізге ала отырып, талдау әзірлемелер пайдаланушы интерфейсін көрсетеді проблеманы төзімділік бағдарламалық қамтамасыз ету басқа операциялық жүйелер сияқты графикалық интерфейс айтарлықтай байланысты мүмкіндіктерін ұсынатын операциялық жүйе үшін, оның құру. Мақала талдау принциптерін құру пайдаланушылық интерфейс. Талпыныс факторларды анықтау болып табылады тиімді моделін құру пайдаланушылық интерфейс.

***Түйін сөздер:** бағдарламалау компоненттері, командалар, процедуралар, сұраныстар, бағдарламалық қамтамасыз ету.*

RESUME

***T.M. Saliy,** Candidate of pedagogical sciences,*
Innovative University of Eurasia (Pavlodar)
Z.B. Bazheneyeva
Innovative University of Eurasia (Pavlodar)

Principles of creation of user interface

Principles of development of graphic user interface are considered in the article. Possibility of creation is described by comfortable and clear to the user of model of co-operating with software without the necessity of study of some special language. The authors, based on the analysis of development of user interface, pointed out a problem on bearableness of software on other operating systems, because a graphic interface essentially depends on the possibilities given by the operating system for his creation. The article is sanctified to the analysis of principles of creation of user interface. An attempt to identify factors that are a prerequisite for creating an effective user interface model was made.

***Keywords**: programming, components, commands, procedures, queries, software.*