

Раздел 2. Пользовательский интерфейс.

При работе с данным разделом Вам предстоит:

1. Изучить темы:

2.1. Определение, классификация пользовательских интерфейсов.

2.2. Место интерфейса в жизненном цикле разработки ПО.

2.3. Модели и методы построения интерфейсов.

2. Ответить на контрольные вопросы для подготовки к тестам.

3. Ответить на вопросы рубежного теста № 2.

4. Материалы для самостоятельной подготовки указаны в файле Приложение. Источники литературы.

2.1. Определение, состав, классификация пользовательских интерфейсов.

В ГОСТ Р 55241.1-2012/ISO/TR 9241-100:2010 Эргономика взаимодействия человек-система. Часть 100. Введение в стандарты, относящиеся к эргономике программных средств, даны определения интерактивной системы, пользовательского интерфейса.

Интерактивная система (interactive system): система компонентов аппаратного и программного обеспечения, которая получает информацию, вводимую пользователем, и передает ему свой ответ, помогая в работе или выполнении задачи. [см. п.10 Приложение. Источники литературы].

Термин «интерфейс» в переводе с английского языка означает сопряжение, граница.

Для взаимодействия человеком с системой, для ввода и получения информации на естественном языке, требуется промежуточное сопряжение, называемое интерфейсом.

Рассмотрим несколько определений пользовательского интерфейса.

Опр. 2.1 Пользовательский интерфейс (интерфейс пользователя) (user interface): Все компоненты интерактивной системы (программное обеспечение или

аппаратное обеспечение), которые предоставляют пользователю информацию и являются инструментами управления для выполнения определенных задач. [ИСО 9241-110: 2006]

Опр. 2.2 Пользовательский веб-интерфейс (Web user interface, World Wide Web user interface): Все аспекты веб-сайта или веб-приложения, относящиеся к информационному наполнению, функциональным возможностям, навигации, взаимодействию и представлению, которые существенны для использования веб-приложения или веб-сайта (см.п.9 Приложение. Источники литературы).




Термин «пользовательский интерфейс» применим к автоматизированным рабочим местам (АРМ) специалистов и прикладным программам.

Опр. 2.3. АРМ - программно-технический комплекс автоматизированной системы, предназначенный для автоматизации деятельности определенного типа.

Опр. 2.4. Прикладная программа, или приложение, - программа, предназначенная для выполнения определённых задач и рассчитанная на непосредственное взаимодействие с пользователем.

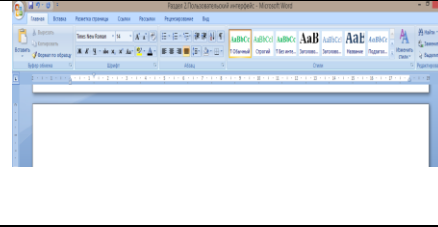
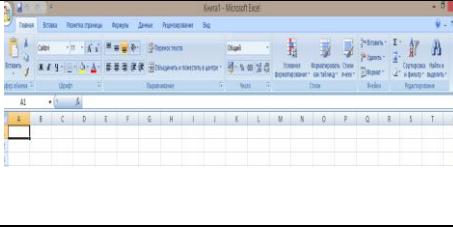
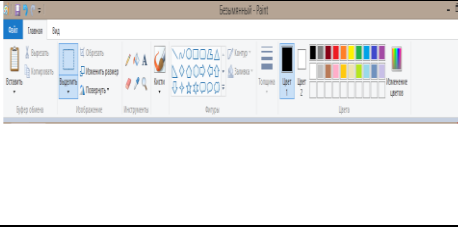
Рассмотрим виды интерфейсов АРМ специалистов (см.табл.2.1):

- ✓ бухгалтера,
- ✓ юриста,
- ✓ финансового директора,
- ✓ банковского оператора,
- ✓ энергетика,
- ✓ пилота авиалайнера,
- ✓ диспетчера,
- ✓ врача
- ✓ тренажеров и т.д.

		
АРМ оператора АЭС	АРМ (тренажеры)	АРМ диспетчеров взлетно-посадочной полосы.

Рассмотрим виды интерфейсов прикладных программ (см.табл.2.2):
 текстовый редактор,
 редактор электронных таблиц,
 редактор создания, обработки изображений и т.д.

Таблица 2.2

		
текстовый редактор	редактор электронных таблиц	редактор создания, обработки изображений

Интерфейс не существует сам по себе и в единственном числе. Он является частью какой-либо системы, состоящей из программных и/или аппаратных средств.

Практически все рассмотренные интерфейсы (табл.2.1, 2.2) программных, аппаратных средств содержат графические, текстовые элементы отображения информации.

Пользовательский интерфейс включает в себя все аспекты эргономики и дизайна, которые оказывают влияние на взаимодействие пользователя с системой.

Пользовательский интерфейс состоит из множества составляющих, таких как:

- набор задач пользователя, которые он решает при помощи системы,
- используемая системой метафора,
- элементы управления системой,
- навигация между блоками системы,
- визуальный (и не только) дизайн экранов программы.

В дизайне пользовательского интерфейса можно условно выделить декоративную и активную составляющие. К декоративным составляющим относятся элементы, отвечающие за эстетическую привлекательность программного изделия. К активным элементам относятся управляющие средства пользовательского интерфейса, посредством которых пользователь управляет программой. Управляющие средства различных классов программных продуктов могут значительно различаться.

Существует ряд классификаций пользовательского интерфейсов, которые применяются в индустрии программных средств. Рассмотрим общую классификацию и их разновидности, применимые в настоящее время. В общем случае пользовательские интерфейсы можно разделить на две большие группы:

- ✓ WIMP-интерфейс, компонентами которого являются: window- окно, icon - пиктограмма, menu - меню и pointer - указатель.
- ✓ SILK-интерфейс, speech - речь, icon - пиктограмма, language - язык, knowledge - знание.

Следующая классификация:

- ✓ графический пользовательский интерфейс (Graphical User Interface - GUI);
- ✓ пользовательский Web-интерфейс (WEB-User Interface WUI);
- ✓ пользовательские интерфейсы карманных устройств (Hand-User Interface - HUI);
- ✓ текстовый пользовательский интерфейс командной строки (Command Line Interface - CLI).
- ✓ тактильный (биометрический, виртуальный, жестовый) интерфейс (WIMP) - организация взаимодействия методами осязания и ощущений,

пространственного расположения. [см. п.8 Приложение «Источник литературы»].

- ✓ трехмерный/пространственный интерфейс (WIMP) – интерфейс прямой манипуляции; создание эффектов объемности, способами затенения, удаленности графических элементов способами уменьшения или увеличения и т.д. [см. п.7 Приложение. Источник литературы].
- ✓ речевой интерфейс (SILK) – основан на методах голосового распознавания, имеет признаки взаимодействия: естественный язык, диалог, неограниченный словарный запас и грамматика. Примеры реализации голосового интерфейса: Alisa/Yandex, Alexa/Amazon, Siri/Apple, OK/Google, Кортана/Microsoft. Также голосовой интерфейс используется в системах навигации. К рассмотренной технологии применяются методы искусственного интеллекта.
- ✓ семантический (общественный) интерфейс - возник в конце 70-х годов XX века, связан с развитием искусственного интеллекта. Он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейс. Основная его отличительная черта - это отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По своей сути это трудно называть интерфейсом - это уже моделирование "общения" человека с компьютером.
- ✓ нейрокомпьютерный интерфейс (НКИ) - система, созданная для обмена информацией между мозгом и электронным устройством, например, компьютером (см.табл.2.5). В однонаправленных интерфейсах внешние устройства могут либо принимать сигналы от мозга, либо посылать ему сигналы. Двухнаправленные интерфейсы позволяют мозгу и внешним устройствам обмениваться информацией в обоих направлениях. В основе нейрокомпьютерного интерфейса, часто используется метод биологической обратной связи.

Тактильные и осязательные устройства могут быть использованы во многих различных ситуациях и для решения многих различных задач. Для некоторых задач они могут быть заменены другими устройствами, для других - могут быть использованы совместно с другими устройствами, для остальных задач природа данного типа обратной связи означает, что эти устройства являются наиболее эффективными.

Благодаря своим компактным размерам и минимальным требованиям к питанию, тактильные дисплеи предлагают доступные средства обеспечения доступа к данным посредством осязания (см.табл.2.5). Тактильные дисплеи часто достаточно малы, что позволяет устанавливать их на другие интерактивные устройства, такие как мышь, клавиатура или игровой контроллер, или на портативные устройства, такие как мобильный телефон и

карманный персональный компьютер (или даже на тактильные/осязательные устройства, обеспечивающие обратную связь по усилию). Тактильная информация широко используется в видеоиграх как недорогой способ предоставления сенсорной обратной связи посредством игровых джойстиков. Тактильные ощущения крайне важны для успешной манипуляции предметами, обнаружения границ и восприятия текстуры. Их также используют для более выразительных и качественных взаимодействий, таких как невизуальные коммуникации (например, похлопывание по плечу или поглаживание руки), и для восприятия качества продукции.

Для управления компьютером используется выражение лица человека, направление его взгляда, размер зрачка и другие признаки. Для идентификации пользователя используется рисунок радужной оболочки его глаз, отпечатки пальцев и другая уникальная информация. Изображения считываются с цифровой видеокамеры, а затем с помощью специальных программ распознавания образов из этого изображения выделяются команды. Эта технология, по-видимому, займет свое место в программных продуктах и приложениях, где важно точно идентифицировать пользователя компьютера.

Биометрический тип интерфейса используется во многих областях для взаимодействия с компьютером, мобильными устройствами, робототехникой, в медицине, игровой индустрии, искусстве и творчестве, тренажерах.

Рассмотрим более подробно вопросы, относящиеся к графическому интерфейсу (WIMP).

Важнейшие свойства GUI-интерфейса:

- ✓ возможность непосредственного манипулирования,
- ✓ поддержка мыши,
- ✓ указателя,
- ✓ тактильное управление,
- ✓ использование графики,
- ✓ наличие области для функций и данных приложения.

Рассмотрим подробнее компоненты GUI (см.табл.2.3).

Окно - это область устройства отображения, используемая для представления и взаимодействия с объектами, информацией об объектах, или для выполнения

действий, применяемых к объекту. Окно обладает строкой заголовка, набором операций перемещения, изменения размера, набором меню и областью для отображения информации об объектах. Окно отображает информацию только на определенную часть или область устройства отображения. Частичное использование устройства отображения позволяет просматривать несколько окон для одновременного взаимодействия с несколькими объектами или управляющими диалогами. Определение окна также подразумевает использование графики или визуализации вместо текстовой информации для указания доступного объема информации.

Набор элементов графического пользовательского интерфейса:

- ✓ меню,
- ✓ кнопки,
- ✓ списки,
- ✓ текстовые поля
- ✓ панель инструментов,
- ✓ строка состояния,
- ✓ ползунки,
- ✓ пиктограммы и т.п

Таблица 2.3

	
<p>Графический интерфейс игры</p>	<p>Графический интерфейс рабочего стола</p>

Пользовательский Web-интерфейс (WEB-User Interface WUI).

Пользовательский веб-интерфейс (Web user interface, World Wide Web user interface): Все аспекты веб-сайта или веб-приложения, относящиеся к

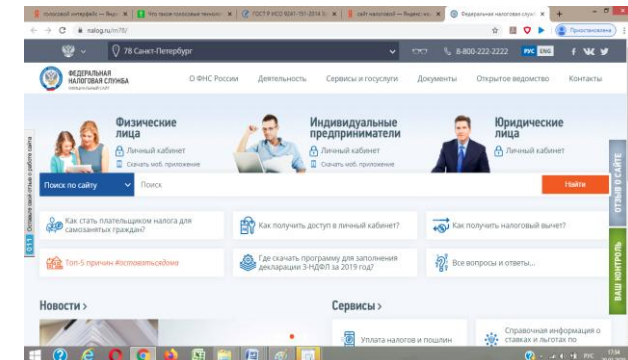
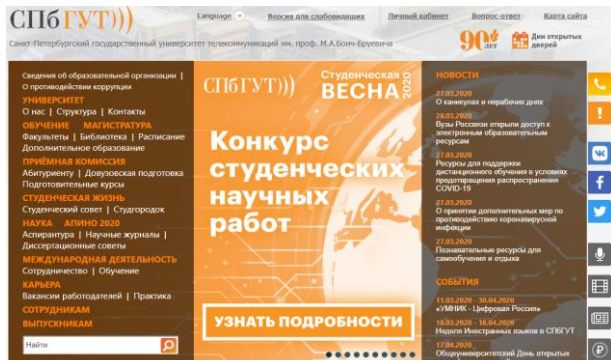
информационному наполнению, функциональным возможностям, навигации, взаимодействию и представлению, которые существенны для использования веб-приложения или веб-сайта [п.4 Приложение. Источники литературы].

Диалог пользователя осуществляется через интерфейс выбранного браузера. Браузер - специальное программное обеспечение, используемое для запроса, обработки, манипулирования и отображения содержания веб-страниц.

Основные элементы взаимодействия веб-интерфейса (см.табл.2.4):

- ✓ меню,
- ✓ ссылки на станицы сайта,
- ✓ ссылки на социальные сети и ресурсы,
- ✓ изображения,
- ✓ текстовые поля,
- ✓ полосы прокрутки,
- ✓ мультимедиа,
- ✓ видео,
- ✓ аудио и т.д.





Таблица 2.4

	
<p>Интерфейс сайта услуг налоговой службы</p>	<p>Интерфейс сайта СПбГУТ</p>

Пользовательские интерфейсы мобильных устройств (Hand-User Interface -HUI). Пользовательский интерфейс мобильного устройства аналогичен тому, что используется на компьютере, но с адаптацией под размеры экрана устройства (см. табл. 2.5). На мобильных платформах под окнами понимаются элементы интерфейса, занимающие все пространство экрана устройства. Переход между такими окнами осуществляется при помощи графических элементов-навигаторов

или тактильным перетягиванием Особенности проектирования мобильных интерфейсов определяются в первую очередь спецификой операционной системы.

Таблица 2.5

	
<p>Тактильный интерфейс Materiable, меняющий свою форму и реагирующий на прикосновения.</p>	<p>Тактильные интерфейсы телефона, часов</p>
	
<p>Интерфейс виртуальной реальности</p>	<p>Жестовый (сенсорный) интерфейс</p>

	
Интерфейс биометрический (глазной)	Нейрокомпьютерный интерфейс

Текстовый пользовательский интерфейс командной строки (Command Line Interface - CLI).

Интерфейс командной строки пользовательского интерфейса или язык командного интерпретатора, является средством взаимодействия с компьютерной программой, где пользователь вводит команды с клавиатуры на запрос программы (см.рис.2.1). Программа, которая обрабатывает интерфейс называется интерпретатор команд языка или оболочки (вычисления).

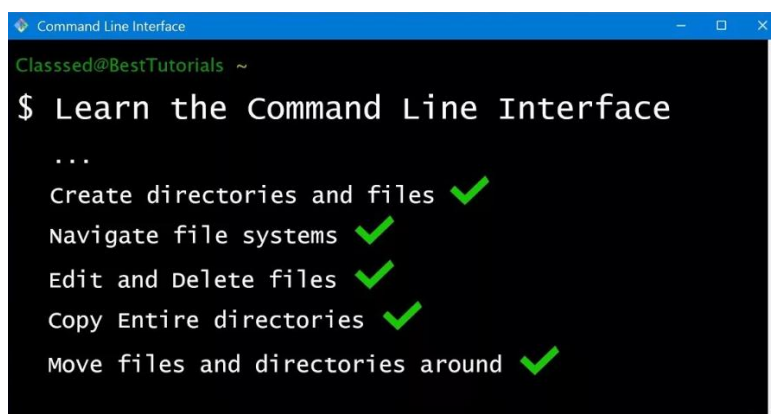


Рис.2.1 Командный интерфейс

2.2. Место интерфейса в жизненном цикле разработки ПО.

Интерфейс не существует сам по себе, он является частью программного продукта, посредством которого осуществляется взаимодействие. Интерфейс может относиться к веб-сайту или к специальной программе, организованной в виде АРМ.

Поэтому, в совокупности с программным продуктом, интерфейс имеет жизненный цикл. И важно, уже на первой стадии разработки, продумывать все вопросы, связанные с качеством интерфейса.

Создание максимально эффективного прототипа интерфейса является чрезвычайно важной задачей. Прототип должен хорошо выглядеть, чтобы понравиться заказчику и не вызывать вопросов у субъектов тестирования, он должен быть максимально дешев, максимально полон и, что немаловажно, должен с легкостью обновляться.

При создании прототипа наиболее частой ошибкой является чрезмерное наведение глянца и вообще стремление сделать прототип возможно более похожим на результирующую систему.

Требования к прототипу изменяются в процессе проектирования:

Первый прототип стоит делать максимально простым. Только после того, как тестирование подтверждает его правильность, стоит делать более детализированный прототип.

Рассмотрим возможные этапы разработки и жизненного цикла интерфейса, представленные на рисунках 2.2 (а-д).

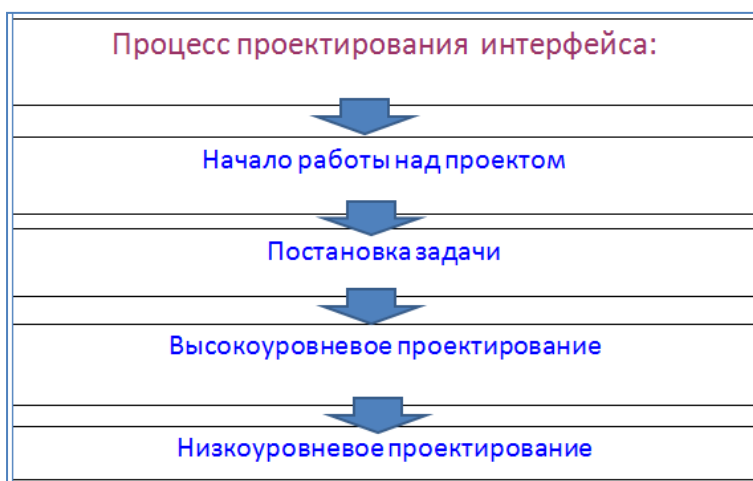


Рис.2.2 а) Процесс проектирования интерфейса



Рис.2.2 б) Начало работы над проектом

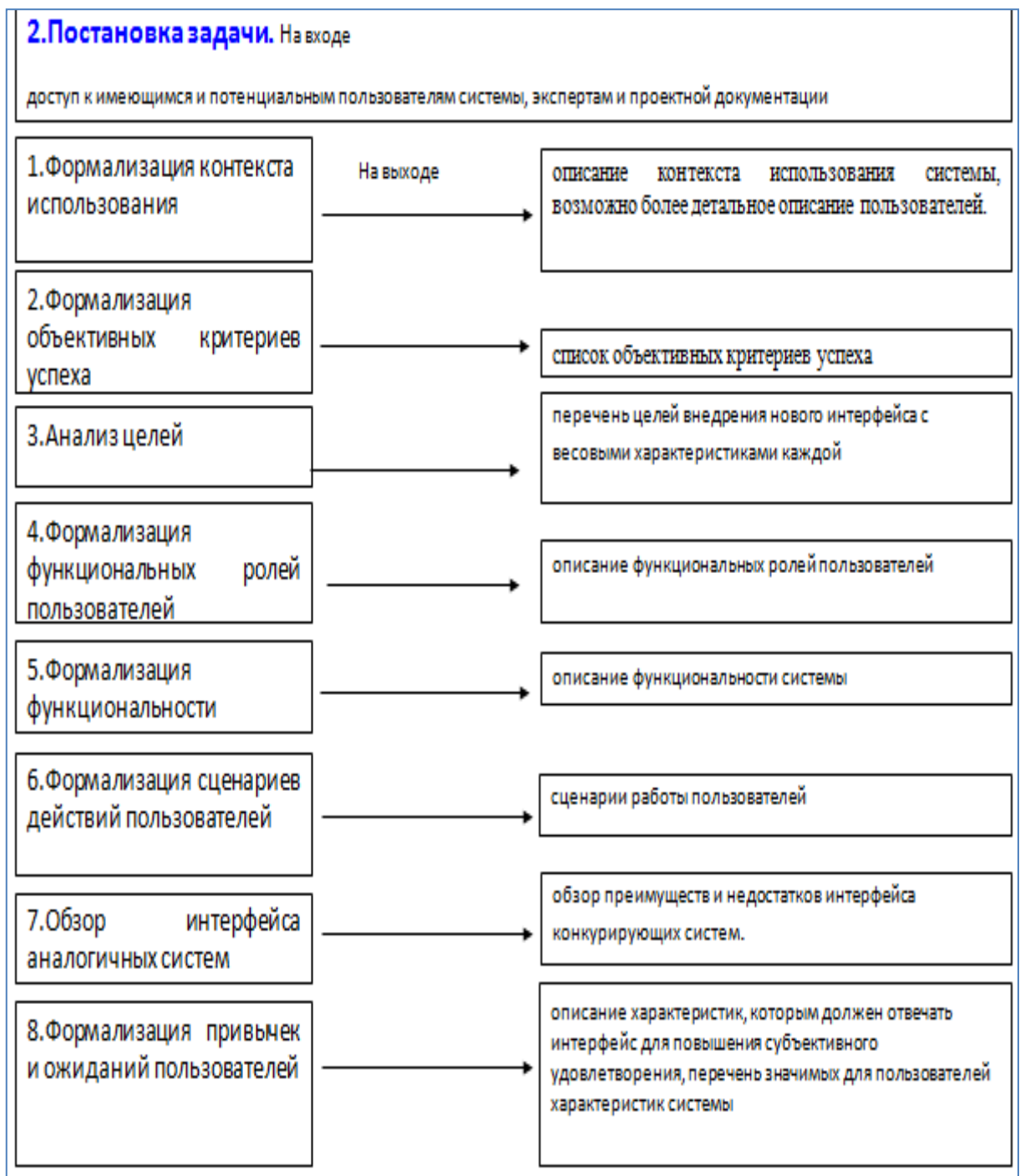


Рис.2.2 в) Постановка задачи на разработку интерфейса

3.Высокоуровневое проектирование интерфейса

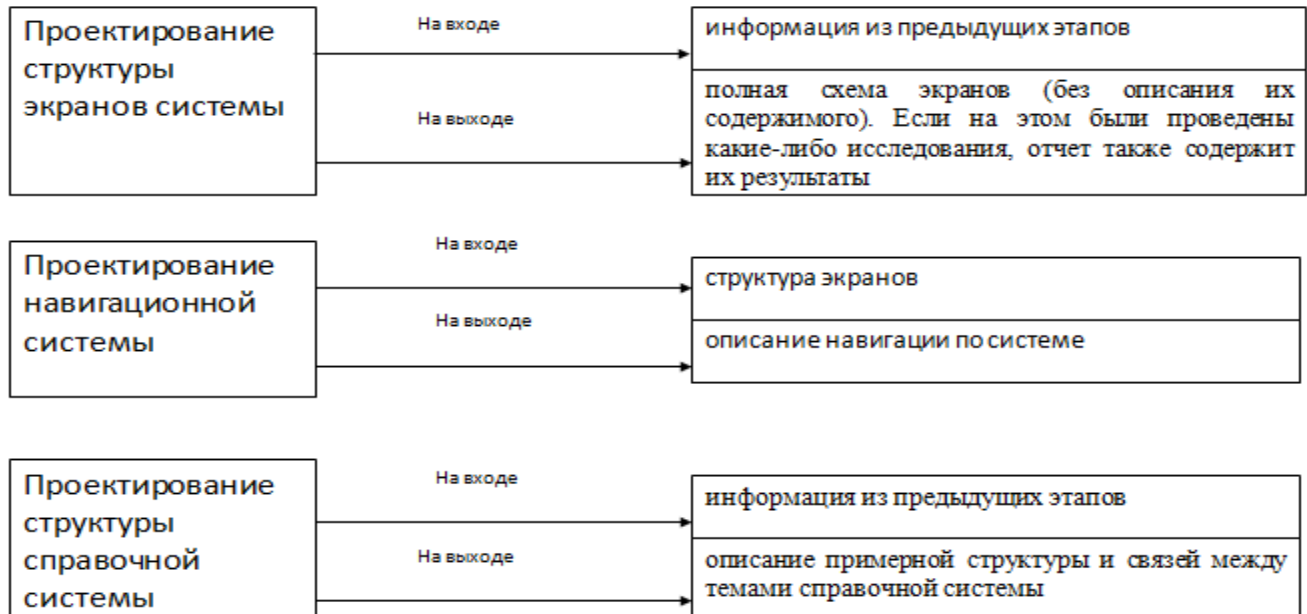


Рис.2.2 г) Высокоуровневое проектирование интерфейса

4.Низкоуровневое проектирование интерфейса



Рис.2.2 д) Низкоуровневое проектирование интерфейса

Рассмотренные этапы жизненного цикла относятся к ситуации, когда в исследуемой предметной области есть апробированные и используемые на практике готовые интерфейсные решения, которые по разным причинам уже не устраивают их потребителей. В случае, когда нужно разработать интерфейс, например, веб-сайта или АРМ с «нуля», тогда можно на первом этапе исследовать решения, близкие по сути и содержанию, но используемые другими организациями, конкурентами.

Если разработанная программа, включающая интерфейс становится неактуальной, то ее жизненный цикл на этом завершается. Неактуальность заключается в несоответствии заявленным требованиям и поставленным задачам, устареванию и т.д.

2.3. Модели и методы построения интерфейсов.

При создании интерфейса важно продумать вопросы концептуального, физического логического проектирования. Эти виды проектирования относятся к разным этапам разработки. Также, надо понимать, что интерфейс входит в состав АРМ, прикладной программы, полноценного сайта, игры. Поэтому, в пределах нашей дисциплины, акцент сделан на вопросах касающихся интерфейса, как звена взаимодействия между человеком и системой. Но требуется рассматривать интерфейс, как часть разработки полноценного продукта. Совокупность полученных знаний по ранее изученным дисциплинам «Информационные технологии», «Технологии программирования» и разделов настоящей дисциплины, позволят создать полноценный продукт, который студенты должны будут выполнить в виде контрольной работы.

Опр. 2.4 Модель- это объект, который имеет сходство в некоторых отношениях с прототипом и служит средством описания и/или объяснения, и/или прогнозирования поведения прототипа системы.

Опр.2.5 Под системой понимается - совокупность элементов и связей между ними, обладающая определенной целостностью.

Опр. 2.6 Ментальная или концептуальная модель – лишь внутреннее отображение того, как пользователь понимает и взаимодействует с системой.

Ментальная модель не обязательно точно отображает ситуацию и ее компоненты.

Она служит основой для понимания, анализа и принятия решений и позволяет пользователю:

- предсказать последующие события;
- найти причины замеченных событий;
- определить необходимые действия для осуществления нужных изменений;
- использовать их как мнемонические устройства для запоминания событий и связей(отношений);
- обеспечить понимание аналогичных устройств;
- применять стратегии, которые позволяют преодолеть ограничения, заложенные в алгоритме обработки информации.

В основе ментальной модели – все взаимоотношения между пользователями и их компьютерами, поэтому она является фундаментом для выработки принципов и правил ПИ.

Концептуальная модель проектирования интерфейса.

Концептуальное проектирование необходимо для уяснения целей и возлагаемых задач, изучения аудитории потенциальных пользователей. Этот вид относится к первому этапу разработки интерфейса.

При концептуальном проектировании определяются:

1. Все цели (главные и второстепенные).
2. Действия, осуществляемые для достижения поставленных целей.
3. Состав и категория пользователей.
4. Интересы групп пользователей (при создании игр, сайтов), пожелания пользователей (при создании интерфейсов АРМ).
5. Действия, выполняемые пользователями.
6. Состав и количество экранов, диалоговых окон.

7. Разделы экранов, окон.

8. Критерии достижения цели.

С учетом поставленных целей, а также интересов, пожеланий пользователей, формируется список функций, сервисов и разделов, которые будут располагаться в окне интерфейса.

Целесообразность создания или модификации оценивается с точки зрения его эффективности. Для оценки можно использовать критерии, которые важны при создании интерфейса, с помощью которого реализуются поставленные цели. Например, при оценке веб-интерфейса, важен критерий количества посетителей сайта, удобство, доступность и т.д. При оценке интерфейса АРМ важны критерии функциональности, удобства и т.п. Все показатели зависят от множества внутренних и внешних потребностей пользователей.

Эффективность интерфейса определяется набором свойств системы, влияющих на качество её функционирования. Чем лучше система, тем выше ее эффективность. В этом случае можно говорить, о достижении цели и выполнении поставленных задач. Критерий эффективности – усредненный показатель выбора лучшей системы или решения.

Логическая модель проектирования интерфейса.

Логическое проектирование – процесс организации и упорядочивания информации, построение его структуры, содержания, формирование навигации. Этот вид проектирования может относиться к первому и второму этапу разработки интерфейса.

Логическое проектирование предполагает:

1. Описание архитектуры модели (линейная, иерархическая, иная).
2. Создание списка экранных окон, разделов.
3. Создание содержимого.
4. Описание навигации между окнами, разделами, страницами.
5. Определение типа информации и графики, которая будет размещена в окне интерфейса.

Для наглядного восприятия и понимания, все данные процесса логического проектирования оформляются в графическом виде (схема, модель). Это вопрос будет рассмотрен в методах построения интерфейса.

Физическая модель проектирования интерфейса.

Физическое проектирование – этап разработки псевдореального и реального прототипов интерфейса. Этот вид проектирования может относиться ко всем этапам, начиная со второго этапа разработки.

При физическом проектировании осуществляется:

1. Выбор программного обеспечения.
2. Выбор методов и технологий проектирования.
3. Разработка необходимых информационных, графических, мультимедиа объектов.
4. Вёрстка псевдореального прототипа.
5. Вёрстка реального прототипа, в составе разработки АРМ, сайта, игры.
6. Проверка на ошибки и соответствие стандартным требованиям.

Физическое проектирование веб-сайта является ответственным и важным процессом, от которого зависит качество созданного продукта и его дальнейшая судьба. Уровень востребованности и заинтересованности пользователями, в конечном итоге определяет показатель его эффективности, с точки зрения достижения всех поставленных целей.

На первом этапе разработки интерфейса программного продукта должен быть создан документ «Техническое задание».

Техническое задание является основным техническим и юридическим документом, систематизирующим порядок и последовательность всех этапов работ. В этом документе прописываются все требования к разрабатываемому продукту, сроки и этапы выполнения работ, стоимость, ответственность сторон. Документ подписывается всеми заинтересованными физическими или юридическими лицами, к которым относятся заказчики и исполнители.

При составлении документа нужно ориентироваться на цели и выработанную концепцию, которая определяет весь набор требований.

Методы построения интерфейсов.

При создании интерфейса, необходимо понимать какие методы или совокупность методов могут быть использованы для его построения, для упрощения процесса его дальнейшего изучения пользователями.

Чтобы научиться пользоваться системой, пользователю нужно построить ментальную модель этой системы. Применение метафоры, позволяет пользователю не создавать новую модель, а воспользоваться готовой моделью, которую он ранее построил по другому поводу.

Метафорический метод построения интерфейса основан на интуитивном понимании. В интерфейс добавляются элементы, графические объекты, которые нам известны из жизни. Например, на рабочем столе Windows находятся графические объекты: корзина, папка, документ и т.п. Эти объекты и термины мы используем в обычной жизни.

У метафор есть несколько существенных недостатков.

1. Не для любой функциональности можно подобрать подходящую метафору, причем заранее узнать, есть ли хорошая метафора или нет, невозможно, так что можно потратить время на поиски и ничего не найти. Это, как минимум, неэффективно.

2. Даже подходящая метафора может оказаться бесполезной, если её не знает существенная часть аудитории или её тяжело однозначно передать интерфейсом.

3. Почти всегда метафора будет сковывать функциональные возможности. Что делать, если проектируемая система обладает большим количеством функций, чем копируемый образец? Следование метафоре в таких условиях будет только вредить, поскольку совпадающим функциям будет учиться легче, а несовпадающим – сложнее(они будут слишком иначе устроены).

4. Для физических объектов как принтеры или документы легко найти визуальную метафору. Но для часто используемых в программах понятий как процессы, связи, службы и преобразования это сделать трудно или даже невозможно. Очень сложно найти хорошую визуальную метафору для покупки билета, смены каналов, приобретения товара, нахождения ссылки, установки формата, вращения (они будут слишком иначе устроены).

5. Метафора может ограничивать систему или вызывать противоречивость восприятия и понимания.

Применять метафорический метод к разработке интерфейса можно, но с большой осторожностью. Гораздо более перспективным является идеоматический подход.

Идиоматический метод основан на знании о том, как решать ту или иную задачу - естественный для человека процесс.

Идиоматический метод основан на том факте, что человеческий мозг - необычайно мощная обучающаяся машина, и что обучаться для нас - легко.

Большинство элементов управления в графическом интерфейсе пользователя - идиомы. Кнопки, выпадающие списки и полосы прокрутки - это то, что мы узнаем автоматически, а не догадываемся метафорически.

Например, мышь не является метафорой чего-либо. Люди обучаются работе с ней идиоматически. В мыши нет ничего, что указывало бы на цель ее применения. Она также не напоминает ничего из нашего опыта, так что обучение работе с ней не интуитивно. Однако научиться работать мышью очень легко. Большинство знакомых нам элементов GUI, которые считаются метафорическими на самом деле являются идиоматическими. Такие артефакты, как кнопки закрытия окна, окна с изменяемыми размерами, бесконечно вложенные папки с файлами, щелчки мышью и перетаскивание пиктограмм - не метафорические операции, потому что их нет в реальном мире. Их сила лишь в простой идиоматической узнаваемости.

Метод Аффорданс (affordance) – это использование свойств объекта, которые сами показывают способ своего использования.

Например, надпись «На себя» на двери не является аффордансом, а облик двери, который подсказывает человеку, что она открывается на себя, несет в себе аффорданс.

Польза аффорданса заключается в том, что он позволяет пользователям обходиться без какого-либо предварительного обучения, благодаря этому аффорданс является самым эффективным и надежным средством обеспечения понятности.

Проблема в том, что аффорданс на экране получить сложнее, нежели в предметах реального мира, поскольку единственным способом его передачи оказывается визуал, а такие способы, как тактильные свойства или приспособленность к человеческой анатомии (ложку, например, трудно держать неправильно) не подходят. Это ограничение приводит к тому, что доступными оказываются всего несколько способов передачи аффорданса, из которых самыми значительными являются четыре:

1. маппинг, или повторение конфигурации объектов конфигурацией элементов управления (этот способ работает хорошо в реальном мире, но не очень хорошо на экране, поскольку предпочтительней непосредственное манипулирование);

2. видимая принадлежность управляющих элементов объекту;

3. визуальное совпадение аффордансов экранных объектов с такими же аффордансами объектов реального мира (кнопка в реальном мире предлагает пользователю нажать на неё, псевдотрехмерная кнопка предлагает нажать на неё по аналогии);

4. изменение свойств объекта при подведении к нему курсора (бледный аналог тактильного исследования).

Стандарт- повторение во всех системах одних и тех же объектов, понятий, названий, правил.

Например, на светофоре красный цвет означает «стоп», желтый - «быть готовым», зеленый – «разрешение на выполнение действия».

Стандарт хорошо работает, когда популярен, в противном случае не работает вовсе. Популярен же он может быть двумя способами: во-первых, он может быть во всех системах, во-вторых, он может быть популярен внутри отдельной системы. Например, стандарт интерфейса MS Windows популярен почти во всех программах для Windows, именно поэтому его нужно придерживаться.

С другой стороны, этот стандарт оставляет неопределенным очень многое, и это многое в разных системах трактуется по-разному. Бесполезно пытаться найти общий знаменатель во всех системах, гораздо эффективнее установить собственный стандарт, не противоречащий стандарту ОС, но дополняющий его; после чего этого стандарта надо придерживаться.

Контрольные вопросы по теме «Пользовательский интерфейс»

1. В чем заключается неудобство командного интерфейса?
2. В чем сущность графического интерфейса?
3. Какими свойствами характеризуется интерфейс АРМ?
4. Какие этапы следует выполнить при разработке интерфейса?
5. Какова структура диалога?
6. Для чего требуется функция навигации?
7. Определение, классификация пользовательских интерфейсов.
8. Место интерфейса в жизненном цикле разработки ПО.
9. Модели построения интерфейсов.
10. Методы построения интерфейсов.
11. Важнейшие свойства GUI-интерфейса.
12. Составляющие пользовательского интерфейса.
13. Этапы разработки и жизненного цикла интерфейса.
14. Высокоуровневое проектирование интерфейса.
15. Низкоуровневое проектирование интерфейса.
16. Классификация интерфейсов.
17. Эффективность интерфейса.