

**Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Уфимский государственный авиационный технический университет**

ЧЕЛОВЕКО-МАШИННОЕ ВЗАИМОДЕЙСТВИЕ

**Лабораторный практикум
по дисциплине «Человеко-машинное взаимодействие»**

Уфа 2008

Федеральное агентство по образованию
Государственное образовательное учреждение
Высшего профессионального образования
Уфимский государственный авиационный технический университет

Кафедра вычислительной математики и кибернетики

ЧЕЛОВЕКО-МАШИННОЕ ВЗАИМОДЕЙСТВИЕ

Лабораторный практикум
по дисциплине «Человеко-машинное взаимодействие»

Уфа 2008

Составители: О. С. Нургаянова, Г. Р. Шакирова

УДК 004.5 (07)

ББК 32.973-018.2 (Я7)

Человеко-машинное взаимодействие: Лабораторный практикум по дисциплине «Человеко-машинное взаимодействие» / Уфимск. гос. авиац. техн. ун-т;

Сост.: О. С. Нургаянова, Г. Р. Шакирова. – Уфа, 2008. – 30 с.

Рассматриваются основные этапы проектирования: от проведения предпроектного обследования предметной области и разработки концептуальных моделей до практической реализации интерфейса и его оценки

Предназначен для студентов специальности 230105 – «Программное обеспечение вычислительной техники и автоматизированных систем» с целью закрепления навыков проектирования и реализации пользовательских интерфейсов при построении сложных программных систем..

Табл. 1. Ил. 3. Библиогр.: 5 назв.

Рецензенты: канд. техн. наук, доц. Попов Д.В.

канд. техн. наук, доц. Федорова Н.И.

© Уфимский государственный
авиационный технический университет, 2008

СОДЕРЖАНИЕ

Введение	4
Лабораторная работа № 1. Формирование концептуальных требований к системе и пользовательскому интерфейсу	6
Лабораторная работа № 2. Построение графа диалога интерфейса пользователя	11
Лабораторная работа № 3. Разработка проекта прототипа пользовательского интерфейса	17
Лабораторная работа № 4. Разработка исполняемого прототипа пользовательского интерфейса	21
Лабораторная работа № 5. Тестирование исполняемого прототипа пользовательского интерфейса	25
Лабораторная работа № 6. Анализ эффективности пользовательского интерфейса	28
Список литературы	32

ВВЕДЕНИЕ

Организация человеко-машинного взаимодействия является важным аспектом проектирования и реализации сложных технических систем. Эффективное функционирование подобных систем возможно только при участии человека, поэтому особое место в их разработке отводится созданию механизмов взаимодействия человека и машины.

Человеко-машинная система выполняет свои задачи благодаря совместной работе устройств и людей, которые рассматриваются как неотъемлемые составляющие части всей системы. Стремительное развитие компьютерных технологий привело к созданию нового класса человеко-машинных систем, где в качестве устройства выступает компьютер. При организации информационного взаимодействия в таких системах возникают сложности, связанные с переводом естественного языка в машинные коды и обратно. Все это определяет высокие требования к уровню подготовки пользователей таких систем. Поэтому актуальной задачей является разработка специального интерфейса, являющегося связующим звеном между человеком и машиной. Это требует от современного специалиста в области компьютерных технологий умения разработать или адаптировать интерфейс под широкий класс пользователей, обеспечить эффективное использование компьютерных систем в разных приложениях.

Как следует из понятия «человеко-машинное взаимодействие» одной из его сторон выступает человек – пользователь. При этом разработчику интерфейса следует учитывать, что взаимодействие с системой будет осуществляться различными категориями пользователей, отличающихся разной степенью технической подготовки. Пользователи редко задумываются над тем, как устроена машина и как она справляется со своими задачами. Для них важнее всего удобство и результаты. Однако все, что видят пользователи, — это интерфейс. Поэтому интерфейс должен обеспечивать интуитивно понятный, максимально эффективный механизм взаимодействия с компьютером.

Пользовательский интерфейс – это не только структура и форма представления информации (ее оформление), но и совокупность всех элементов взаимодействия пользователя с системой. Проектирование интерфейса – это установление того, что пользователь будет видеть, какие действия выполнять и что получать в ответ. Это итерационный процесс, в основе которого лежат результаты анализа предметной

области, выявление потенциальных пользователей, постановке задачи, а итогом является полнофункциональный пользовательский интерфейс, учитывающий все особенности взаимодействия человека и машины в каждой конкретной ситуации.

Процедура проектирования пользовательского интерфейса может быть представлена как совокупность этапов, рассмотренных в рамках лабораторных работ:

1. Проектирование концептуальной модели интерфейса (лабораторная работа № 1), целью которого является разработка теоретической базы интерфейса, конкретизируемой на последующих этапах.

2. Детализация процедуры взаимодействия пользователя с системой посредством графа диалога (лабораторная работа № 2), в основе которого лежит концептуальная модель интерфейса. Целью данного этапа является последовательное описание итераций взаимодействия пользователя с системой.

3. Прототипирование пользовательского интерфейса (лабораторная работа № 3), целью которого является построение шаблона, макета интерфейса для его последующего функционального наполнения.

4. Разработка исполняемого прототипа пользовательского интерфейса (лабораторная работа № 4), целью которого является установление соответствия между целями функционирования системы и компонентами прототипа интерфейса.

5. Тестирование интерфейса на удобство применения (лабораторная работа № 5), целью которого является выявление возможных ошибок, допущенных на этапах проектирования и разработки, для их последующего анализа и исправления.

6. Анализ эффективности разработанного интерфейса (лабораторная работа № 6) с целью количественной оценки интерфейса для определения его конкурентоспособности среди аналогичных программных продуктов.

В ходе выполнения лабораторных работ студенты получают навыки итерационного проектирования пользовательского интерфейса, когда в рамках определенной задачи выполняется разработка концептуального проекта и его последующая практическая реализация. Тестирование и количественная оценка интерфейса позволяют оценить его место среди однотипных программных продуктов.

ЛАБОРАТОРНАЯ РАБОТА № 1

ФОРМИРОВАНИЕ КОНЦЕПТУАЛЬНЫХ ТРЕБОВАНИЙ К СИСТЕМЕ И ПОЛЬЗОВАТЕЛЬСКОМУ ИНТЕРФЕЙСУ

1. Цель работы

Целью работы является изучение методологии проведения предпроектного обследования предметной области с целью выявления решаемых задач и определения их информационных потребностей с использованием современных CASE-средств.

2. Задачи

В рамках лабораторной работы решаются следующие задачи: овладение методами разработки функциональных моделей, приобретение навыков построения на основе информационных моделей и навыков применения специализированных программных продуктов – *Logic Works BPWin* и *ERWin*.

На выполнение лабораторной работы отводится 4 академических часа.

3. Теоретическая часть

Эффективность пользовательского интерфейса непосредственно зависит от возможности предварить его разработку и внедрение описанием всего комплекса проблем, которые необходимо разрешить, с указанием того, какие функции системы должны быть автоматизированы, определением того, как взаимодействует система со своим окружением. Решение этой задачи базируется на проведении предпроектного обследования, позволяющего выявить основные задачи, которые должна выполнять проектируемая система. Формальные методики проведения такого анализа базируются на разработке *системного проекта*, который предусматривает создание процессных, функциональных, информационных и динамических моделей.

Для любой системы определяющим является ее функциональное содержание, так как оно определяет ее основные свойства. Поэтому начальным этапом разработки системного проекта является построение функциональной модели.

Функциональная модель представляет с требуемой точностью детализации систему функций, которые отражают свои взаимоотношения через объекты системы [1, 2].

В основе функциональной модели лежит исходная информация по рассматриваемому проекту, которая формируется в виде мнемосхем, отражающих особенности существующего и предлагаемого подходов к решению выявленных задач.

Наиболее распространенным стандартом проектирования функциональных моделей является *IDEF0* (первоначальное название *SADT – Structured Analysis and Design Technique*). Построенная с помощью этой методологии модель и отображает функциональную структуру системы, то есть производимые ей действия и связи между этими действиями.

Процесс моделирования в *IDEF0* начинается с определения контекста, то есть определения наиболее абстрактного уровня описания системы в целом. В контекст входит определение субъекта моделирования, цели и точки зрения на модель [1, 2].

Модель должна рассматриваться всегда с одной и той же позиции, которая называется *точкой зрения* данной модели. Этот аспект разработки функциональной модели занимает особое место, поскольку одни и те же задачи по-разному решаются отдельными категориями пользователей. Например, функции, связанные с проведением сессии на факультете вуза, решаются деканатом, преподавателями и даже студентами, а каждый из них уже со своих позиций подходит к решению этой задачи.

Модель не может быть построена без четко сформулированной *цели*. Формулировка цели позволяет команде аналитиков сфокусировать усилия в нужном направлении. При этом если цель будет слишком общей, то функциональная модель станет весьма громоздкой, что существенно усложнит и без того непростой этап концептуального проектирования.

При проектировании *контекста функциональной модели* должна быть сформулирована основополагающая функция системы, соответствующая определенным ранее цели и точке зрения. На следующем этапе необходимо выполнить *декомпозицию* контекстной функции, т.е. разбиения выбранной сложной задачи на более простые, каждая из которых, в свою очередь, также может быть декомпозирована и т. д. Каждое из таких описаний называется *диаграммой*.

IDEF0-модель объединяет и организует диаграммы в иерархические структуры, в которых диаграммы наверху модели менее детализированы, чем диаграммы нижних уровней. Каждая диаграмма представляет собой отдельный уровень декомпозиции, общее количество которых задает *степень декомпозиции* модели.

Каждая *IDEF0*-диаграмма содержит *блоки* и *дуги*. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними. В *IDEF0* каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов нижняя – для механизмов (рис. 1.1).

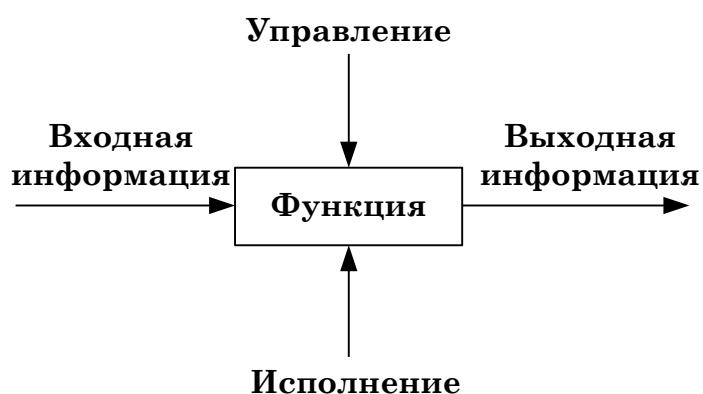


Рис. 1.1. Структура функционального блока *IDEF0*

Приведенное обозначение отражает следующие системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кто, что и как выполняет функцию.

В целом построение функциональной модели позволяет выделить функции, которые следует автоматизировать в рамках разрабатываемой системы.

Информационная модель – это формализованное описание информационных структур и операции над ними. Информационное моделирование представляет собой, прежде всего, анализ логической структуры информации об объектах системы. Эта логическая структура является необходимым дополнением функциональной модели, детализируя объекты, которыми манипулируют функции системы.

Методологической основой информационного моделирования является стандарт *IDEF1x*, использующий концепцию «сущность-

связь» (Entity-Relationship). Фактически с помощью диаграммы «сущность-связь» документируются сущности системы и способы их взаимодействия, включая идентификацию объектов, важных для предметной области (*сущностей*), свойств этих объектов (*атрибутов*) и их отношений с другими объектами (*связей*).

Построение информационной модели выполняется на базе функциональной модели в соответствии со следующими правилами:

- входная, выходная, управляющая и исполнительная информация функциональной модели становится сущностями в информационной модели;

- функциональные блоки преобразуются в связи между сущностями в информационной модели;

- каждая сущность должна содержать уникальный идентификатор, однозначно определяющий каждый ее экземпляр. Например, сущность «студент» может содержать в качестве идентификатора номер зачетной книжки. При этом каждый экземпляр этой сущности, т.е. каждый конкретный студент (например, Иванов) однозначно характеризуется этим значением: нет двух студентов с одинаковым номером зачетной книжки.

Построение информационной модели позволяет выявить информационные потребности каждой из автоматизируемых функций.

CASE-средства ERwin и BPwin, разработанные фирмой Logic Works, являются наиболее распространенными механизмами разработки функциональных и информационных моделей. *CASE-средство* верхнего уровня *BPwin* поддерживает методологию *IDEF0* (функциональная модель), а *ERwin* — методологию *IDEF1X* (информационная модель) [3].

4. Порядок выполнения работы

1. Выполнить предпроектное исследование предметной области согласно варианту задания и результат представить в виде мнемосхемы класса «как есть».

2. Определить не менее пяти функциональных возможностей проектируемой системы. Реализовать задачу анализа входных данных и последующего принятия решения из заранее определенного множества. Выбранные функции должны отвечать некоторым общим критериям и не должны решать диаметрально противоположные задачи. Например, если задана предметная область «Отдел кадров», то могут быть выделены следующие функции:

- прием на работу;
- увольнение сотрудников;
- премирование / штрафные санкции.

Их можно объединить в единую группу «Учет кадров».

Если же выбраны, например, функции:

- прием на работу;
- увольнение сотрудников;
- учет расхода канцелярских принадлежностей в отделе,

то такой набор некорректен, поскольку не может быть отнесен к какой-то одной обобщающей функции.

3. Построить функциональную модель. Функциональный блок контекстной диаграммы должен соответствовать обобщающей функции. Декомпозиция первого уровня – это функциональные блоки, соответствующие 5 выбранным в п.2 функциям. Специфицировать не менее 3 входных, выходных, управляющих и исполнительных воздействий с точки зрения пользователя будущей системы. Степень декомпозиции каждого функционального блока – не менее 3. Количество функциональных блоков – не менее 4.

4. Построить информационную модель – не менее 5 сущностей. Дать подробное описание всех субъектов и объектов функционирования с указанием их информационной нагрузки, т.е. тех свойств, которые в контексте данных автоматизируемых функций имеют значение.

5. Варианты заданий

В качестве вариантов заданий предлагаются следующие предметные области:

- кинопрокат;
- библиотека;
- автосалон;
- издательство;
- мебельный магазин;
- гостиница;
- ателье;
- кадровое агентство;
- книжный магазин;
- деканат. Сессия.

6. Контрольные вопросы

1. С какой целью выполняется предпроектное обследование предметной области?
2. Что входит в состав системного проекта?
3. Что такое функциональная модель и какую контекстную нагрузку она несет?
4. На базе какой методологии строится функциональная модель?
5. Что такое информационная модель? Какова ее роль в предпроектном обследовании?
6. На базе какой методологии строится информационная модель?
7. Каковы правила перехода от функциональной модели к информационной?
8. В чем заключается различие понятий «сущность» и «экземпляр сущности»?

7. Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. Титульный лист, название и цель работы.
2. Вариант задания и постановку задачи.
3. Мнемосхему.
4. Обоснование выбора автоматизируемых функций и их описание.
5. Функциональную модель.
6. Информационную модель.
7. Выводы.

ЛАБОРАТОРНАЯ РАБОТА №2

ПОСТРОЕНИЕ ГРАФА ДИАЛОГА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

1. Цель работы

Целью работы является изучение методики разработки графа диалога, описывающего механизм работы проектируемой системы в контексте взаимодействия с пользователем.

2. Задачи

В рамках лабораторной работы решаются следующие задачи: приобретение навыков разработки графа состояний и переходов, отражающего взаимодействие экранных форм проектируемого интерфейса на основе графа диалога.

На выполнение лабораторной работы отводится 4 академических часа.

3. Теоретическая часть

Отличительной особенностью информационного взаимодействия между пользователем и компьютером является его *диалоговый* характер. Каждая итерация этого взаимодействия предполагает поочередный обмен информацией между его участниками.

Для того чтобы диалог был конструктивным, должны соблюдаться следующие правила:

- участники диалога должны понимать язык друг друга;
- участники диалога не должны говорить одновременно;
- очередное высказывание должно учитывать как общий контекст диалога, так и последнюю информацию, полученную от собеседника.

При проектировании диалога, необходимо определить:

- структуру диалога;
- возможный сценарий развития диалога;
- содержание управляющих сообщений и данных, которыми могут обмениваться человек и приложение (семантику сообщений);
- визуальные атрибуты отображаемой информации (синтаксис сообщений).

Общая структура диалога определяется его типом. Выделяют четыре типа диалога [4]:

1. Диалог на основе командного языка. Исторически это первая из реализованных структур диалога. Она очень часто используется в операционных системах. При такой организации диалога система не выводит ничего, кроме постоянной подсказки (приглашения на ввод команды), которая означает готовность системы к работе. Каждую команду вводят с новой строки и обычно заканчивают нажатием клавиши «ввод». Ответственность за правильность задаваемых команд ложится на пользователя. Система информирует о невозможности выполнения неверной команды, не поясняя, как правило, причин.

2. Диалог типа «вопрос-ответ». Структура диалога типа «вопрос-ответ» (Q&A) основана на аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информацию от пользователя в виде ответов на вопросы. Это наиболее известная структура диалога; все диалоги, управляемые компьютером, в той или

иной степени состоят из вопросов, на которые пользователь отвечает, однако в структуре Q&A этот процесс выражен явно. В каждой точке диалога система выводит в качестве подсказки один вопрос, на который пользователь дает один ответ. В зависимости от полученного ответа система может решить, какой следующий вопрос задавать. Такую структуру диалога обычно применяют при проектировании экспертных систем.

3. Диалог на основе меню. Меню является наиболее популярным вариантом организации запросов на ввод данных во время диалога, управляемого компьютером. Структура типа меню является наиболее естественным механизмом для работы с устройствами указания и выбора: меню представляет собой изображение тех объектов, которые выбираются пользователем. Меню можно с равным успехом применять для ввода как управляющих сообщений, так и данных. Приемлемая структура меню зависит от его размера и организации, от способа выбора пунктов меню и реальной потребности пользователя в поддержке со стороны меню.

4. Диалог на основе экранных форм. Диалог на основе экранных форм допускает обработку на одном шаге диалога нескольких ответов. На практике формы используются там, где учет какой-либо деятельности требует ввода стандартного набора данных. Человек работает с формой до тех пор, пока не заполнит ее полностью и не передаст системе. Система может проверять каждый ответ непосредственно при вводе или по окончании заполнения всей формы. Такая структура позволяет повысить скорость ввода данных по сравнению со структурой типа «вопрос-ответ» и манипулировать более широким диапазоном входных данных, нежели меню; кроме того, с ней могут работать пользователи любой квалификации.

Независимо от типа диалога его развитие во времени можно рассматривать как последовательность переходов системы из одного состояния в другое. Ни одно из этих состояний не должно быть тупиковым, т.е. пользователь должен иметь возможность перейти из любого текущего состояния диалога в требуемое (за один или несколько шагов). Для этого в ходе разработки интерфейса необходимо определить все возможные состояния диалога и пути перехода из одного состояния в другое – разработать *сценарий диалога*.

В качестве примера рассмотрим формирование графа диалога выбора студенческой группы в рамках решения задачи управления

учебным процессом на уровне факультета ВУЗа (рис. 2.1). Каждая группа характеризуется множеством параметров (специальность, курс, номер группы и т. д.). Поэтому ее выбор может осуществляться как по одному из этих параметров, так и по их совокупности. Все эти возможности должны быть учтены в графе диалога: следует разработать альтернативные переходы между всеми допустимыми состояниями. В качестве отдельного состояния выступает запрос одного параметра, а в качестве перехода – передача выбранного параметра следующему состоянию. В нашем случае такими состояниями являются последовательные запросы на выбор специальности, номера группы, факультета. Каждый последующий запрос приводит к ограничению множества допустимых данных, т.е. выбор факультета сокращает список доступных специальностей, который в свою очередь определяет множество групп. В реальности вопросы могут задаваться в разной последовательности, поэтому в графе диалога все состояния связаны друг с другом, а зачастую просто дублируются.

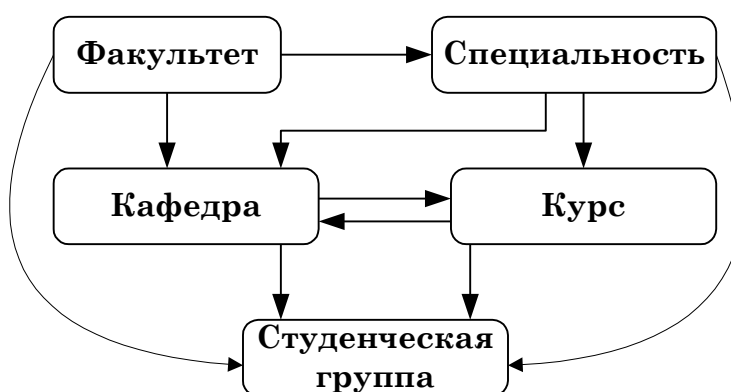


Рис. 2.1. Построение графа диалога на примере организации выбора студенческой группы

Для формального описания сценария диалога применяются модели дискретно-событийного класса. На сегодняшний день их спектр достаточно широк: сети Петри, конечные автоматы, графы состояний и переходов, иерархические ситуационные модели. Каждая из этих моделей определяет множество состояний и переходов между ними при выполнении определенных внешних и внутренних условий.

Формальное описание диалога в контексте пользовательского интерфейса представляется экранными формами – состояниями и

элементами управления (кнопки, ссылки и т.п.) – переходами – и реализуется в форме графа состояний и переходов интерфейса.

При разработке графа состояний и переходов интерфейса должны соблюдаться следующие правила:

1. Состояние графа диалога переходит в экранную форму графа состояний и переходов интерфейса.

2. В рамках одной экранной формы могут быть представлены как одно состояние графа диалога, так и их совокупность.

Графическое представление графа состояний и переходов представлено двумя элементами: состояниями – «О», и переходами – «—», при этом выделяют две группы состояний. Первая из них соответствует новой экранной форме и обозначается как «О». Вторая соответствует текущей экранной форме, определенным образом видоизмененной в результате выполнения переходов, и обозначается как «●». Каждый переход должен содержать идентификатор (имя) того элемента управления, с которым он связан.

Для рассматриваемого примера граф состояний и переходов может быть представлен единственной экранной формой (рис. 2.2), на которой предусмотрен выбор параметров поиска студенческой группы: факультет, специальность, номер и т.д. Каждая итерация выбора параметра приводит к видоизменению исходной экранной формы с учетом выполненной фильтрации данных. В качестве переходов могут быть выбраны раскрывающиеся списки, позволяющие сделать выбор соответствующего параметра.

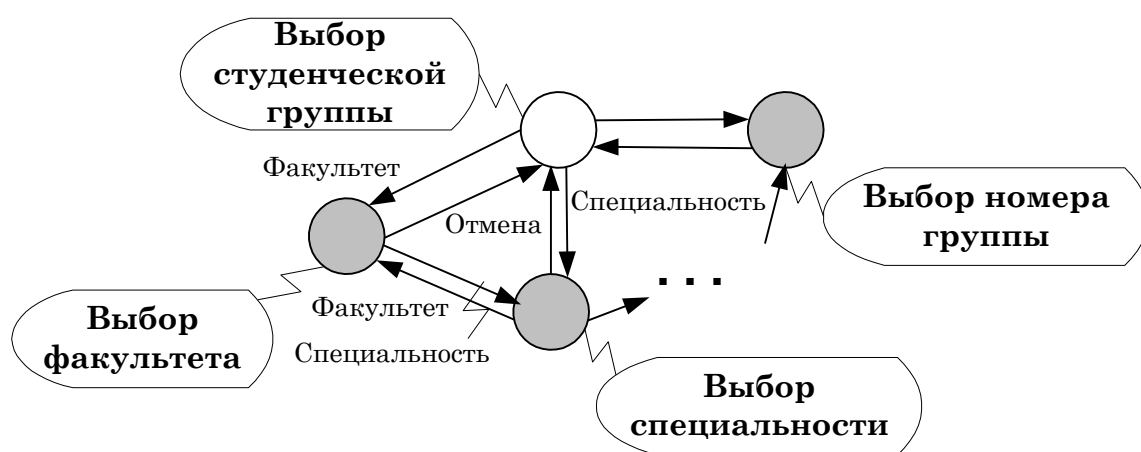


Рис. 2.2. Построение графа состояний и переходов пользовательского интерфейса на примере организации выбора студенческой группы

Построение графа диалога и последующая разработка графа состояний и переходов интерфейса позволяет еще на этапе проектирования выявить и устранить слабые стороны разрабатываемого интерфейса.

4. Порядок выполнения работы

1. На основании разработанных функциональной и информационной моделей (лабораторная работа № 1) построить граф диалога, предусмотрев возможность выбора не менее 10 параметров.

2. Разработать граф состояний и переходов интерфейса проектируемой системы. Выявить тупиковые ситуации и описать способы их устранения.

5. Контрольные вопросы

1. Что такое диалог? Какие типы диалога вы знаете?
2. Что такое граф диалога? Какова его роль в процессе проектирования интерфейса?
3. Что такое граф состояний и переходов интерфейса? Каковы его составляющие?
4. Приведите примеры классов моделей, которые могут быть использованы для построения графа диалога.
5. Какие типы состояний предусмотрены в графе состояний и переходов интерфейса?

6. Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. Титульный лист, название и цель работы.
2. Вариант задания и постановку задачи.
3. Граф диалога. Обоснование выбора представленных параметров в форме таблицы:

Нпп	Название параметра	Обоснование выбора

4. Граф состояний и переходов интерфейса.
5. Выводы.

ЛАБОРАТОРНАЯ РАБОТА №3

РАЗРАБОТКА ПРОЕКТА ПРОТОТИПА

ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

1. Цель работы

Целью работы является изучение особенностей построения прототипов пользовательского интерфейса и принципов их стандартизации в контексте решаемой задачи, сформулированной при изучении и анализе действий пользователя в рамках предпроектного обследования.

2. Задачи

В рамках лабораторной работы решаются следующие задачи: приобретение навыков проектирования прототипов пользовательских интерфейсов и обоснования выбора управляющих элементов с учетом возможной взаимосвязи объектов и субъектов интерфейса.

На выполнение лабораторной работы отводится 4 академических часа.

3. Теоретическая часть

При построении интерфейса важно придерживаться определенных принципов, способствующих обеспечению его унификации и дружелюбности:

- контроль пользователем интерфейса. У пользователя создается субъективное ощущение управления системой, делающей его взаимодействие с системой более комфортным;
- уменьшение загрузки памяти пользователя. Элементы, редко или вовсе не используемые пользователем должны располагаться «на заднем плане» или быть скрыты от пользователя, уступая место наиболее актуальным, часто используемым. Элементам управления;
- последовательность пользовательского интерфейса. Все элементы управления в своей совокупности должны «читаться»: у пользователя не должно возникать сомнения, в каком порядке ему следует использовать те или иные элементы.

Поэтому важнейшим этапом разработки пользовательского интерфейса является проектирование его прототипа. Решение этой задачи не всегда однозначно – обилие доступных элементов управления затрудняет выбор каждого конкретного интерфейсного

элемента, предназначенного для решения той или иной задачи.

Еще одна особенность, которая должна быть учтена при проектировании интерфейса, состоит в обеспечении его объектности, заключающейся в возможности сопоставления объектов и субъектов предметной области отдельным элементами управления.

Решение этих задач носит вполне формализованный характер и базируется на определении объектов и субъектов разрабатываемой системы в контексте доступных элементов управления.

На начальном этапе разработки прототипа интерфейса должен быть составлен *перечень объектов и данных* [4], которые должны быть задействованы в данном проекте. Для каждого объекта указывается его обобщенный тип (например, данные или устройство), позволяющий заранее выделить необходимые классы объектов. При этом каждый объект разрабатываемого прототипа будет представлять собой экземпляр одного из выделенных классов.

Все объекты можно разделить на две основные группы – активные и пассивные. Первые предназначены для выполнения какой-то конкретной операции или операций, т.е. являются объектами, на которые может быть направлено действие. Например, для активного объекта «Список студентов» могут быть определены три операции (действия) – заполнить, найти, проверить. Все эти действия предполагают участие объекта «Список студентов» в качестве операнда – «заполнить список студентов», «найти в списке студентов», «проверить список студентов».

Вторая группа объектов – пассивные – предназначена для объектов, на которые не направлены никакие действия. Иными словами, это такие объекты, все действия с которыми сводятся к единственной операции – просмотреть их содержимое без активного воздействия на него.

При этом следует иметь в виду, что одни объекты могут представлять подмножество для других. Например, объект «Студент» является подмножеством объекта «Список студентов». Эти взаимосвязи должны быть учтены при описании их свойств и допустимых действий, в частности, такие объекты должны, по меньшей мере, относиться к одному и тому же типу. Например, объекты «Студент» и «Список студентов» должны относиться к единому типу – данные, – в противном случае механизм их сопоставления не будет однозначным.

Следующий этап разработки прототипа связан с определением

взаимосвязей между описанными ранее объектами. Эти взаимосвязи носят характер передачи потоков информации от одного объекта к другому или передачи одного объекта другому в форме потока. Решение этой задачи позволит заранее спланировать воздействие одного объекта на другой и при их последующей практической реализации сформировать логику функционального описания этих объектов. Например, взаимосвязь объектов «Список студентов» и «Экзаменационная ведомость» может быть определена как «Сформировать», что означает, что список студентов как поток информации преобразуется в новый объект – экзаменационную ведомость. Поэтому соответствующие элементы управления в функциональном отношении должны придерживаться механизма подобного взаимодействия: содержимое экземпляра класса «Список студентов» является источником данных для функции, предусмотренной для экземпляра класса «Экзаменационная ведомость».

Последний этап создания проекта прототипа пользовательского интерфейса предполагает агрегирование данных, полученных на двух предшествующих этапах, в форме *матрицы прямого манипулирования объектами*. Эта матрица представляет собой двумерную таблицу, строки и столбцы которой описывают исходные и конечные объекты взаимодействия экземпляров классов, выявленных на предыдущих этапах. Для того, чтобы описать все возможные варианты взаимодействия, необходимо и в столбцах, и в строках таблицы указать *все* выявленные на первом этапе проектирования объекты и данные, независимо от того, к какому виду они относятся (активные или пассивные). На пересечении строк и столбцов таблицы должны быть заданы связывающие их действия. Например, в ячейке, заданной пересечением строки «Студент» и столбца «Список студентов», должно помещаться значение «Добавить студента в список», что соответствует результатам, полученным на предшествующих этапах проектирования прототипа пользовательского интерфейса.

Полученная матрица манипулирования позволяет из множества доступных элементов управления выбрать те, что наиболее эффективно реализуют заданные объекты и их взаимодействие.

Как известно, проектирование интерфейса является итерационным процессом. Негативная оценка проекта пользователями обуславливает необходимость его корректировки, т.е. возвращению на предыдущий этап разработки. Поэтому еще на этапе проектирования должны быть

предусмотрены альтернативные варианты интерфейсного представления всех заявленных объектов, которые могут быть использованы в случае корректировки проекта прототипа интерфейса. Такие элементы должны быть подобны в плане их функциональной нагрузки, однако их внешнее представление может коренным образом отличаться.

Так, например, объект «Категория студента» может быть визуализирован с помощью трех альтернативных интерфейсных элементов:

- раскрывающийся список – ListBox, позволяющий выбрать соответствующее значение;
- компонент CheckBox, с единственным значением, например «Бюджетник»: если он выбран, то значение данного объекта – бюджетник, в противном случае – контрактник;
- компонент RadioButton, содержащий оба возможных значения и гарантирующего единственный выбор.

4. Порядок выполнения работы

1. На основании системного проекта и графа диалога выделить основные объекты и действия, которые должны участвовать в процессе функционирования разрабатываемой системы.

2. Определить взаимосвязи между выделенными объектами.

3. Составить матрицу прямого манипулирования объектами.

4. На основе функциональной модели разработать визуальное представление разрабатываемой системы, определить основные компоненты интерфейса будущей системы.

5. Обосновать выбор используемых элементов управления и предусмотреть альтернативные варианты.

6. Организовать возможность получения информации в режиме диалога. Для этого следует использовать разработанные в лабораторной работе №2 вопросы к пользователю.

7. Предусмотреть в проекте интерфейса возможность вывода результата пользователю на основе исходных данных, полученных от пользователя.

8. При проектировании предусмотреть согласование с созданной в лабораторной работе №1 функциональной моделью. В случае необходимости, доработайте свой проект в соответствии с моделью.

5. Контрольные вопросы

1. Перечислите основные принципы проектирования пользовательского интерфейса.
2. Какие типы объектов вы знаете?
3. Какие типы данных циркулируют между объектами? Как они выражаются?
4. Что представляет собой матрица прямого манипулирования?

6. Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. Титульный лист, название и цель работы.
2. Вариант задания и постановку задачи.
3. Таблицу объектов и действий в виде:

Объекты и данные	Тип объекта	Действия

4. Схему взаимодействия объектов.
5. Матрицу прямого манипулирования.
6. Перечисление элементов интерфейса, их функциональной нагрузки и обоснование их выбора.
7. Выводы.

ЛАБОРАТОРНАЯ РАБОТА №4

РАЗРАБОТКА ИСПОЛНЯЕМОГО ПРОТОТИПА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

1. Цель работы

Целью работы является изучение механизмов обеспечения функциональной нагрузки интерфейсных элементов проекта прототипа в контексте задач, решаемых с помощью соответствующих объектов и данных.

2. Задачи

В рамках лабораторной работы решаются следующие задачи: разработать исполняемый прототип пользовательского интерфейса на основе составленного проекта с учетом специфики предметной области и ориентаций на разные категории пользователей.

На выполнение лабораторной работы отводится 4 академических часа.

3. Теоретическая часть

Разработанный на начальной стадии создания пользовательского интерфейса проект задает просто визуальное представление определенной задачи, а не ее решение. Поэтому необходимо дополнить этот проект соответствующим функциональным содержанием, которое должно полностью соответствовать специфике решаемой задачи или комплекса задач в рамках заданной предметной области.

Современные языки программирования высокого уровня позволяют поставить в соответствие каждому интерфейсному элементу управления методы, отражающие их специфику. Использование *RAD*-систем (например, *Delphi*) позволяет максимально упростить эту задачу за счет, как визуального проектирования управляющих элементов, так и более гибкого их программного описания.

Согласно концепции построения графа состояний и переходов интерфейса каждая последующая итерация взаимодействия с пользователем приводит изменению состава данных. Эту особенность и следует реализовать посредством соответствующего функционального содержания. Так, например, рассмотрим два класса объектов «Факультет» и «Специальность». Множество данных первого класса определяет информационное обеспечение второго класса, т.е. при выборе факультета список специальностей ограничивается их принадлежностью к нему. Пользователю должны быть предоставлены возможности оперировать данными в контексте того состояния, в котором находится система в конкретный момент времени. Пусть для визуального представления рассматриваемых классов объектов выбран компонент *ListBox*. Для решения поставленной задачи необходимо предусмотреть метод связанный с изменением выбранного значения. Например, в рамках *RAD*-системы *Delphi* такой метод уже предусмотрен (*onChange*), т. е. достаточно его просто объявить. Однако такая ситуация не всегда имеет место. Зачастую разработчику недостаточно имеющейся функциональности системы и он вынужден создавать собственные. Такой способ, хотя и является трудоемким, зато обеспечивает большую гибкость по сравнению с традиционным подходом.

Важно обеспечить однозначное соответствие между предлагаемой функциональностью и содержанием матрицы прямого манипулирования объекта разработанной на предыдущем этапе.

Поскольку проектирование пользовательского интерфейса процесс итерационный, необходимо предусмотреть, по крайней мере, один альтернативный вариант для каждой предлагаемой функции.

Разработка исполняемого прототипа пользовательского интерфейса не будет эффективной без тщательно продуманной поддержки пользователей, значит, необходима справочная система, детально описывающая все аспекты функционирования разрабатываемой системы.

Для того чтобы документация, поставляемая совместно с системой была полезна всем системным пользователям, она должна содержать пять описанных ниже документов.

1. Функциональное описание, в котором кратко представлены функциональные возможности системы. Прочитав функциональное описание и вводное руководство, пользователь должен определить, та ли это система, которая ему нужна?

2. Документ по инсталляции системы, в котором содержится информация по установке системы. Здесь должна быть представлена информация о дисках, на которых поставляется система, описание файлов и минимальные требования к конфигурации.

3. Вводное руководство, предоставляющее неформальное введение в систему, описывающее ее «повседневное» использование. Как начать работу с системой, как использовать общие возможности. Все описания снабжаются примерами и содержат сведения о том, как восстановить систему, после совершения ошибки и как начать работу заново.

4. Справочное руководство, в котором описаны возможности системы и их использование, представлен список сообщений об ошибках и возможные причины их появления.

5. Руководство администратора, необходимое для некоторых типов программных систем. В нем дано описание сообщений, генерируемых системой при взаимодействии с другими системами, и описаны способы реагирования на эти сообщения. Если в систему включена аппаратная часть, то в руководстве администратора должна быть информация о том, как выявить и устранить неисправности связанные с аппаратурой, как подключить новые периферийные устройства и т. п.

Таким образом, справочная система, разработанная согласно вышеописанной концепции, позволит обеспечить полнофункциональную поддержку пользователей, независимо от их

категории и уровня подготовки. В целом, процедуру разработки исполняемого прототипа пользовательского интерфейса можно считать завершенной.

4. Порядок выполнения работы

1. В выбранной *RAD*-системе разработать исполняемый прототип пользовательского интерфейса.

2. Разработать компоненты справочной системы.

2.1. Функциональное описание.

2.2. Документ по инсталляции системы.

2.3. Вводное руководство.

2.4. Справочное руководство.

2.5. Руководство администратора.

5. Контрольные вопросы

1. Что такое исполняемый прототип пользовательского интерфейса? На основе чего он строится?
2. Для чего необходимо предусматривать альтернативные варианты функционального представления элементов управления?
3. Какие компоненты справочной системы вы знаете?

6. Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. Титульный лист, название и цель работы.
2. Вариант задания и постановку задачи.
3. Таблицу интерфейсных элементов и их функционального описания с указанием альтернативных операций:

Интерфейсный элемент	Метод	Альтернативный метод	Обоснование выбора

4. Скриншоты, иллюстрирующие работу исполняемого прототипа.
5. Разработанную справочную систему.
6. Выводы.

ЛАБОРАТОРНАЯ РАБОТА №5

ТЕСТИРОВАНИЕ ИСПОЛНЯЕМОГО ПРОТОТИПА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

1. Цель работы

Целью работы является изучение механизмов тестирования программных продуктов в контексте обеспечения удобства пользователя и однозначного внешнего и внутреннего представления составных элементов интерфейса.

2. Задачи

В рамках лабораторной работы решаются следующие задачи: освоение существующих методик проведения тестирования программного продукта на удобство применения пользователем.

На выполнение лабораторной работы отводится 4 академических часа.

3. Теоретическая часть

Ошибки в программном обеспечении, ухудшающие его эксплуатационные характеристики, приводят к убыткам как потребителей, так и разработчиков. Однако эти характеристики зависят и от множества различных факторов, и одним из самых значительных из них является качество пользовательского интерфейса.

Ошибки в интерфейсе наиболее критичны для таких свойств продукта, как производительность пользователей системы (этот показатель определяется из расчета, сколько единиц продукции в час может создать пользователь), количество человеческих ошибок (как часто пользователь ошибается при выполнении определенной операции), скорость обучения работе с системой и субъективное удовлетворение пользователей. Удобство применения является «клеем», который скрепляет все части, которые должны соединиться вместе, чтобы составить какой-либо продукт.

Согласно определению международной организации стандартизации (ISO), удобство применения – это эффективность, рентабельность и удовлетворение, с которым пользователи смогут выполнить те или иные задачи в заданной среде.

Существуют следующие способы проведения тестирования:

– наблюдение;

- проведение опросов и исследований;
- контекстуальные опросы;
- эвристические оценки;
- работа с выделенными группами;
- лабораторное тестирование.

Каждый из этих способов характеризуется различной степенью сложности и ориентирован на конкретные категории пользователей.

Важной частью любого тестирования является правильный подбор пользователей (и достаточного их числа). Участники теста должны быть типичными представителями пользователей данного продукта. Их количество зависит от многих факторов, в том числе, времени, ресурсов, проекта теста, типа тестируемых задач и вида статистического анализа, который вы планируете провести по полученным результатам.

Прежде чем планировать и проводить тестирование на удобство применения, следует четко определить цели и задачи, стоящие перед ним. Эта ответственность лежит на владельцах продукта, специалистах, составляющих планы, проектировщиках и разработчиках. Цель – это обеспечение преимуществ продукта перед конкурирующими в области простоты изучения, эффективности, гибкости и т. д. Для большей точности и достоверности каждая цель делится на подзадачи.

В качестве факторов, определяющих удобство применения пользователем, можно выделить:

- полезность – степень, до которой продукт позволяет пользователю достичь стоящей перед ним цели. Оценка мотивации пользователя в использовании данного продукта. Оценка полезности, как правило, осуществляется с помощью оценки качества выполненной работы;
- эффективность – насколько успешно продукту удастся содействовать пользователю в выполнении стоящей перед ним задачи. Оценка эффективности, как правило, осуществляется с помощью оценки качества выполненной работы;
- простота изучения. Пользователи могут начать использовать продукт, приобретя определенный уровень знаний после прохождения тренинга. Оценка простоты изучения, как правило, осуществляется с помощью оценки качества выполненной работы;
- отношение пользователей – восприятие, ощущения и мнения пользователей по поводу изучения и использования данного продукта.

Оценка, как правило, осуществляется в соответствии с отзывами пользователей (устной или письменной обратной связи).

Можно сформулировать следующие рекомендации по тестированию:

- сравнительные тесты могут представить полезную информацию при исследовании удобства применения новых версий программных продуктов;
- используйте задачи общего характера и области известных проблем в качестве базовых задач, чтобы произвести точные сравнительные измерения между версиями продуктов;
- применяйте стандартные измерения удобства применения, например, качество выполнения работ пользователем (выполняемость задач и показатель успешности их выполнения), степень удовлетворенности пользователей (данные по степени удовлетворенности и предпочтений);
- используйте внешних, независимых производителей для планирования и проведения тестирования на удобство применения, чтобы уменьшить субъективность восприятия;
- необходимой частью каждого теста должно являться описание ожидаемых результатов работы программы;
- программе не должна тестироваться ее автором;
- организация-разработчик программного обеспечения не должна «единолично» его тестировать;
- необходимо подбирать тесты не только для правильных (предусмотренных) входных данных, но и для неправильных (непредусмотренных);
- при анализе результатов каждого теста необходимо проверять, не делает ли программа того, что она не должна делать.

4. Порядок выполнения работы

1. Спроектировать тесты для оценки удобства применения разработанного интерфейса.

2. Выполнить спроектированные тесты с привлечением сторонних пользователей.

3. Проанализировать полученные результаты.

5. Контрольные вопросы

1. Что такое тестирование на удобство применения?

2. Какие требования предъявляются к тестированию?

3. Какие государственные стандарты регламентируют проведение тестирования?

6. Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. Титульный лист, название и цель работы.
2. Вариант задания и постановку задачи.
3. Описание тестовых заданий.
4. Результаты выполнения тестов в форме таблицы:

№ п/п	Краткое описание тестируемого действия	Ожидаемый результат	Фактический результат

5. Анализ полученных результатов.

ЛАБОРАТОРНАЯ РАБОТА №6

АНАЛИЗ ЭФФЕКТИВНОСТИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

1. Цель работы

Целью работы изучение методик количественной оценки эффективности и производительности разработанного пользовательского интерфейса.

2. Задачи

В рамках лабораторной работы решаются следующие задачи: приобретение навыков применения модели GOMS для расчета количественных показателей эффективности и соотношений теории информации для оценки информационной производительности пользовательского интерфейса.

На выполнение лабораторной работы отводится 4 академических часа.

3. Теоретическая часть

Наиболее распространенным подходом к количественному анализу пользовательских интерфейсов является классическая модель *GOMS* — «правила для целей, объектов, методов и выделения» (*the model of goals, objects, methods and selection rules*). Моделирование GOMS позволяет предсказать, сколько времени потребуется

опытному пользователю на выполнение конкретной операции при использовании данной модели интерфейса [5].

В рамках модели *GOMS* определен перечень временных интервалов [5], необходимых для выполнения различных видов операций. Здесь каждый интервал обозначен одной буквой:

$K = 0,2$ с. Нажатие клавиши. Время, необходимое для того, чтобы нажать клавишу.

$P = 1,1$ с. Указание. Время, необходимое пользователю для того, чтобы указать на какую-то позицию на экране монитора.

$H = 0,4$ с. Перемещение. Время, необходимое пользователю для того, чтобы переместить руку с клавиатуры на графическое устройство ввода или с графического устройства ввода на клавиатуру.

$M = 1,35$ с. Ментальная подготовка. Время, необходимое пользователю для того, чтобы умственно подготовиться к следующему шагу.

R . Ответ. Время, в течение которого пользователь должен ожидать ответ компьютера.

С помощью таких типичных значений мы можем сделать правильную сравнительную оценку между какими-то двумя интерфейсами по уровню эффективности их использования. Если оцениваются сложные интерфейсы, включающие пересекающиеся временные зависимости, или если должны быть с точностью достигнуты определенные временные интервалы, то следует применять более сложные модели.

Вычисления времени, необходимого на выполнение того или иного действия (например, «переместить руку с графического устройства ввода на клавиатуру и набрать букву»), с помощью модели *GOMS* начинаются с перечисления операций из списка жестов модели *GOMS*, которые составляют это действие (в приведенном примере это H и K). Перечисление движений (K , P и H) — это довольно простая часть модели *GOMS*. Более сложным, например, в модели скорости печати *GOMS*, является определение точек, в которых пользователь остановится, чтобы выполнить бессознательную ментальную операцию, — интервалы ментальной подготовки, которые обозначаются символом M .

Основные правила, позволяющие определить, в какие моменты будут проходить ментальные операции, представлены в таблице [5].

Информационная производительность интерфейса E определяется как отношение минимального количества информации,

необходимого для выполнения задачи, к количеству информации, которое должен ввести пользователь. Так же как и в отношении физической производительности, параметр E может изменяться в пределах от 0 до 1. Если никакой работы для выполнения задачи не требуется или работа просто не производится, то производительность составляет 1.

Производительность E может равняться и 0 в случаях, когда пользователь должен ввести информацию, которая совершенно бесполезна. Следует отметить, что в интерфейсах можно встретить немало деталей, которые имеют сомнительную ценность из-за параметра $E=0$. Примером такого бесполезного элемента может быть диалоговое окно, в котором есть только одна-единственная возможность для действия пользователя, например кнопка ОК.

В параметре E учитывается только информация, необходимая для задачи, и информация, вводимая пользователем. Два или более методов действия могут иметь одинаковую производительность E , но иметь разное время выполнения.

Правила определения моментов выполнения ментальных операций

Правило 0. Начальная расстановка операторов M	Операторы M следует устанавливать перед всеми операторами K , а также перед всеми операторами P , предназначенными для выбора команд. Однако перед операторами P , предназначенными для указания аргументов этих команд, ставить оператор M не следует
Правило 1. Удаление ожидаемых операторов M	Если оператор, следующий за оператором M , является полностью ожидаемым с точки зрения оператора, то этот оператор M может быть удален
Правило 2. Удаление операторов M внутри когнитивных единиц	Если строка вида $M K M K \dots$ принадлежит когнитивной единице, то следует удалить все операторы M , кроме первого. Когнитивной единицей является непрерывная последовательность вводимых символов, которые могут образовывать название команды или аргумент

Правило 3. Удаление операторов M перед последовательными разделителями	Если оператор K означает лишний разделитель, стоящий в конце когнитивной единицы, то следует удалить оператор M , стоящий перед ним
Правило 4. Удаление операторов M , которые являются прерывателями команд	Если оператор K является разделителем, стоящим после постоянной строки, то следует удалить оператор M , стоящий перед ним
Правило 5. Удаление перекрывающихся операторов M	Любую часть оператора M , которая перекрывает оператор R , означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует

Информация измеряется в битах. Один бит, который представляет собой один из двух альтернативных вариантов (таких как 0 или 1, да или нет), является единицей информации. Например, чтобы выбрать один из каких-либо четырех объектов, потребуется 2 бита информации. В общем случае при количестве n равновероятных вариантов суммарное количество передаваемой информации определяется по формуле Хартли: $\log_2 n$.

Если вероятности для каждой альтернативы не являются равными и i -я альтернатива имеет вероятность $p(i)$, то информация, передаваемая этой альтернативой, определяется с помощью формулы Шеннона: $p(i) \log_2 (1/p(i))$.

Рассмотрим информационное содержание интерфейса, в котором возможно сделать только нажатие единственной клавиши. В данном случае может иметь место одно из двух событий: кнопка нажата или кнопка не нажата. Эти события образуют полную группу, поэтому если вероятность нажатия на кнопку равна p , то вероятность ее «ненажатия» определяется как $(1 - p)$. Тогда информационное содержание данного интерфейса найдем как: $p \log_2 (1/p) + (1-p) \log_2 (1/(1-p))$.

Этот пример показывает важный момент, который заключается в том, что мы можем оценить объем информации, содержащейся в сообщении, только в контексте всего набора возможных сообщений [5].

4. Порядок выполнения работы

1. Выполнить расчет производительности разработанного в предыдущих лабораторных работах пользовательского интерфейса по модели *GOMS*.

2. Оценить информационную производительность разработанного интерфейса с помощью теоремы Шеннона (распределение вероятностей событий указать самостоятельно).

5. Контрольные вопросы

1. Какие способы количественной оценки производительности интерфейса Вы знаете?

2. В каких ситуациях применим метод оценки, основанный на модели *GOMS*?

3. Что такое информационная производительность интерфейса?

4. Чем отличается расчет информационной производительности интерфейса при работе с равновероятными и разноравновероятными объектами?

6. Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. Титульный лист, название и цель работы.

2. Вариант задания и постановку задачи.

3. Порядок и результаты расчета эффективности интерфейса с помощью модели *GOMS*.

4. Порядок и результаты расчета информационной производительности интерфейса.

5. Выводы.

СПИСОК ЛИТЕРАТУРЫ

1. Овчинников, В.Г. Методология проектирования автоматизированных информационных систем: основы системного подхода / В.Г. Овчинников. М. : Спутник+, 2005. 284 с.
2. Черемных, С.В. Структурный анализ систем: IDEF-технологии. М. : Финансы и статистика, 2005. 192 с.
3. Маклаков, С.В. ВРwin и ERwin. CASE-средства разработки информационных систем / С.В. Маклаков. М. : ДИАЛОГ-МИФИ, 2005. 256 с.
4. Мандел, Т. Дизайн интерфейсов / Т.Манделл. М. : ДМК Пресс, 2005. 416 с.
5. Раскин, Д. Интерфейс: новые направления в проектировании компьютерных систем / Д. Раскин. М. : Символ-Плюс, 2005. 272 с.

Составители: НУРГАЯНОВА Ольга Сергеевна
ШАКИРОВА Гульнара Равиловна

ЧЕЛОВЕКО-МАШИННОЕ ВЗАИМОДЕЙСТВИЕ

Лабораторный практикум
по дисциплине «Человеко-машинное взаимодействие»

Подписано в печать 19.05.2008. Формат 60x84 1/16 .
Бумага офсетная. Печать плоская. Гарнитура Times New Roman.
Усл. печ. л. 1,8. Усл.-кр.-отт. 1,8. Уч.-изд. л. 1,7.

Тираж 100 экз. Заказ №

ГОУ ВПО Уфимский государственный авиационный технический университет
Центр оперативной полиграфии УГАТУ
450000, Уфа-центр, ул. К. Маркса, 12