

КУРС ДЛЯ АСПИРАНТУРЫ

«Пользовательский интерфейс: семиотические аспекты»

«User interface: semiotic aspects»

Читается в весеннем семестре

Лекции – 28 часов

Форма контроля – 2 коллоквиума и экзамен (в письменной форме)

За курс отвечает кафедра алгоритмических языков

Автор программы – профессор М.Г.Мальковский

Лектор – профессор М.Г.Мальковский

Аннотация

В курсе рассмотрены основные проблемы и перспективы создания удобных и надежных способов общения человека с компьютером. В центре внимания находятся эргономические, семиотические и лингвистические аспекты пользовательского интерфейса. Исследуются роль и место естественного языка и искусственных знаковых систем в обеспечении дружественного интерфейса, а также возможности невербального общения. Рассматриваются подходы к построению описаний естественного языка и анализу и синтезу текста и звучащей речи.

Содержание курса

Интерфейс "человек-компьютер". Основные понятия. Прикладное программное обеспечение; взаимодействие пользователя с компьютером; 5 поколений языков общения с ЭВМ; интеллектуализация ЭВМ. Удобство и эффективность взаимодействия; эргономические аспекты. Типы и схемы диалога. Многооконные интерфейсы. Программная и техническая поддержка.

Базовые понятия семиотики и лингвистики. Знаки и языки. Свойства языкового знака. Знак и значение; значение и смысл. Естественный язык; особенности его функционирования; семантика и прагматика; речевые ошибки. Язык и мышление. Гипотеза лингвистической относительности.

Естественный язык как средство человеко-машинного общения. Мифы и реальность; естественный язык и естественность общения. Описание естественного языка; история, методы. Формальное описание русской морфологии. Адаптивный диалог с ЭВМ на естественном языке.

ЛИТЕРАТУРА И WEB-ИСТОЧНИКИ

Основная литература

1. Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем – М.: Символ-плюс, 2005. – 272 с.[PDF]

(<https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbX1eGRlc2lnbm1zaWRpfGd4OjM2YWQzZjFmZDYwMmQyZGM>) (31.01.2020) Шифр: 5ВГ66, Р-242

2. Коутс Р., Влейминк И. Интерфейс "человек-компьютер" – М.: Мир, 1990.
(Несколько месяцев назад доступ к электронной версии был заблокирован с сообщением: Доступ к информационному ресурсу ограничен на основании Федерального закона «Об информации, информационных технологиях и о защите информации»)

Шифр: 4Ск, 2249 | 5ВГ66, К-739

3. Мальковский М.Г., Грацианова Т.Ю., Полякова И.Н. Прикладное программное обеспечение: системы автоматической обработки текстов. – М.: МГУ, 2000.

[HTML] (<http://knigosite.ru/library/books/46298>) (31.01.2020)

Шифр: 5ВГ66, М-211

Дополнительная литература

1. Искусственный интеллект. Междисциплинарный подход. / под ред. Д.И. Дубровского и В.А. Лекторского – М. : ИИнтеЛЛ, 2006.

[PDF](<https://klex.ru/mwd>) (31.01.2020)

Шифр: 5В, И-868

2. Звегинцев В.А. Язык и лингвистическая теория. – М. : URSS : Изд-во ЛКИ, 2008.

[HTML](http://www.ae-lib.org.ua/texts/zveginsev_language_and_linguistic_theory_ru.htm) (31.01.2020)

Шифр: 7М, З-430

3. Новое в лингвистике: сб.ст.: Вып.23. Когнитивные аспекты языка : [пер. с англ.] / сост., ред. и вступ. ст. В. В. Петрова и В. И. Герасимова – М. : Прогресс, 1988

[PDF](<https://www.booksite.ru/fulltext/novoelingv/text.pdf>) (31.01.2020)

Шифр: 7М, Н-740

Литература для самостоятельной работы учащихся в соответствии с тематическим планом:

1. Пospelов Д.А. Интеллектуальные интерфейсы для ЭВМ новых поколений // Электронная вычислительная техника. Сборник статей. Вып.3. –М.: Радио и связь, 1989. – С.4-20. [DOC]

(<http://www.raai.org/library/getauthor.php?author=%CF%EE%F1%EF%E5%EB%EE%E2%20%C4.%C0>)

2. Тидвелл Дж. Разработка пользовательских интерфейсов – СПб.: Питер, 2008. – 416 с. [DJVU] (<https://ru.b-ok.cc/book/2370470/5374d0>)

3. Ehlert Patrick Intelligent User Interfaces: Introduction and Survey (2003)

[PDF] (<http://www.kbs.twi.tudelft.nl/docs/report/DKS03-01.pdf>)

4. Энциклопедия КРУГОСВЕТ. Лингвистической относительности гипотеза

(http://www.krugosvet.ru/enc/gumanitarnye_nauki/lingvistika/LINGVISTICHESKO_OTNOSITELNOSTI_GIPOTEZA.html)

5. Shimei Pan, Oliver Brdiczka, Giuseppe Carenini, Duen Horng Chau, Per Ola Kristensson What's Hot in Intelligent User Interfaces // Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2016.

[PDF] (<https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11847/12280>)

6. Уорф Б. Язык, сознание и реальность / Эпистемология и философия науки – т. 50, № 4, 2016. – С. 220-243

(<https://cyberleninka.ru/article/v/yazyk-soznanie-i-realnost>)

Кроме того для знакомства с лекционным материалом лектор после каждой очередной лекции рассылает слушателям курса по электронной почте (на коллективный почтовый ящик) конспект этой лекции.

Раздел 1. Интерфейс "человек-компьютер". Основные понятия.

О главном термине курса:

интерфейс — в вычислительной технике, система унифицированных связей, посредством которых устройства компьютера взаимодействуют между собой, а также набор средств диалога, взаимодействия между пользователем и компьютером. С помощью интерфейса пользователь управляет работой компьютера: выдаёт задания, отвечает на его запросы, получает информацию о работе программы.

интерфейс — Совокупность средств и правил, обеспечивающих взаимодействие устройств вычислительной машины или системы обработки информации и (или) программ. [ГОСТ 15971 90] [ГОСТ Р 50304 92].

интерфейс — в широком смысле определенная стандартами граница между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

// Словари и энциклопедии на Академике

https://dic.academic.ru/dic.nsf/enc_tech/471/%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81

(05.02.2020.).

Интерфейс человек-компьютер — "...Интерфейс человек-компьютер (ИЧК) (human/computer interface (HCI)): граница, через которую реализуется физическое взаимодействие человека с прикладной платформой..."

// Словари и энциклопедии на Академике

https://official.academic.ru/8262/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D1%87%D0%B5%D0%BB%D0%BE%D0%B2%D0%B5%D0%BA-%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80

(05.02.2020.).

Пользовательский интерфейс (UI — англ. *user interface*) — совокупность средств, при помощи которых пользователь общается с различными устройствами, чаще всего — с компьютером или бытовой техникой, либо иным сложным инструментарием (системой).

// Словари и энциклопедии на Академике

<https://dic.academic.ru/dic.nsf/ruwiki/1100993> (08.02.20202.)

Немного об истории внедрения ЭВМ

Прикладное программное обеспечение; взаимодействие пользователя с компьютером; интеллектуализация ЭВМ

Традиционно основное внимание уделялось *точности результатов* работы вычислительных систем (ВС). Гораздо меньше внимания уделялось проблеме *удобства* работы с ВС.

60-е.-70-е гг. – ЭВМ окружена кондиционерами, обслуживается десятками инженеров, техников и операторов. Единственный режим использования ЭВМ – *пакетный*.

Посредник во взаимодействии пользователя с ЭВМ – оператор.

УСЛОВИЯ РАБОТЫ «ДИКТУЕТ» МАШИНА.

Решение задач на ЭВМ (основные этапы):

Содержательная постановка задачи	Формальная постановка задачи	Разработка алгоритма	Написание программы	Получение программы на маш.яз.	Отладка, тестирование	Эксплуатация	Анализ результатов
----------------------------------	------------------------------	----------------------	---------------------	--------------------------------	-----------------------	--------------	--------------------

Программное обеспечение (ПО) – комплекс программ, обеспечивающих обработку или передачу данных и предназначенных для многократного использования и применения разными пользователями (часто считают, что в состав ПО следует включать программную документацию).

По видам выполняемых функций программное обеспечение подразделяется на: *системное, прикладное и инструментальное*.

Системное ПО – нужно для обеспечения работы компьютера и вычислительных сетей (создание операционной среды функционирования других программ; обеспечение надежной и эффективной работы самого компьютера и вычислительной сети; диагностика и профилактика аппаратуры компьютера и вычислительных сетей; выполнение вспомогательных технологических процессов: копирование, архивация, восстановление файлов и др.).

Инструментальное ПО – используется в ходе разработки, корректировки или развития других программ: редакторы, компиляторы, отладчики, вспомогательные системные программы, графические пакеты и др.

Прикладное ПО – программное обеспечение, состоящее из отдельных прикладных программ, библиотек программ, пакетов прикладных программ, интегрированных систем, предназначенное для решения различных прикладных задач пользователей.

Некоторые классы прикладных программ и систем:

- текстовые процессоры,
- электронные таблицы,
- информационно-поисковые системы,
- веб-обозреватели,
- базы данных,
- системы деловой графики,
- оболочки,
- интегрированные системы типа Microsoft Office.

По мере развития ПО происходил переход:

Отдельные программы → Библиотеки программ → Пакеты прикладных программ → Интегрированные системы

Отдельная программа (может быть модульной)

- пишется на некотором языке программирования
- обращение к ней осуществляется с помощью штатных средств системы программирования
- настройка на конкретную задачу осуществляется с помощью параметров

Библиотека программ – совокупность программ для решения **типовых задач** некоторой прикладной области.

Свойства программ библиотеки:

общее программное обеспечение,

единообразие способов внутренней организации и использования.

Пакет прикладных программ – комплекс взаимосвязанных прикладных программ (библиотека) и системных средств для решения некоторого класса **характерных задач** прикладной области.

Интегрированная система – комплекс программ (элементами которого могут быть отдельные программы, библиотеки, пакеты прикладных программ, системные средства) для решения **комплексных (сложных, составных, разнородных) задач**.

Основные составляющие классической интегрированной системы (период 1990 - 2000 гг.):

- база данных,
- электронная таблица,
- подсистема деловой графики,
- текстовый процессор.

Система деловой графики – система, поддерживающая работу с различными видами графиков и диаграмм: гистограммы, круговые и секторные диаграммы и т.д.

Оболочка – программа, создаваемая для упрощения работы со сложными программными системами. Оболочки преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа меню. Обычно оболочка реализуется в виде отдельной программы.

На схеме показан – выделен **желтым** цветом – примерный объем текста (фрагменты программы и/или текста на языке управления заданиями, описание входных данных), который приходится писать пользователю при создании отдельной программы и при использовании библиотек программ и пакетов прикладных программ.

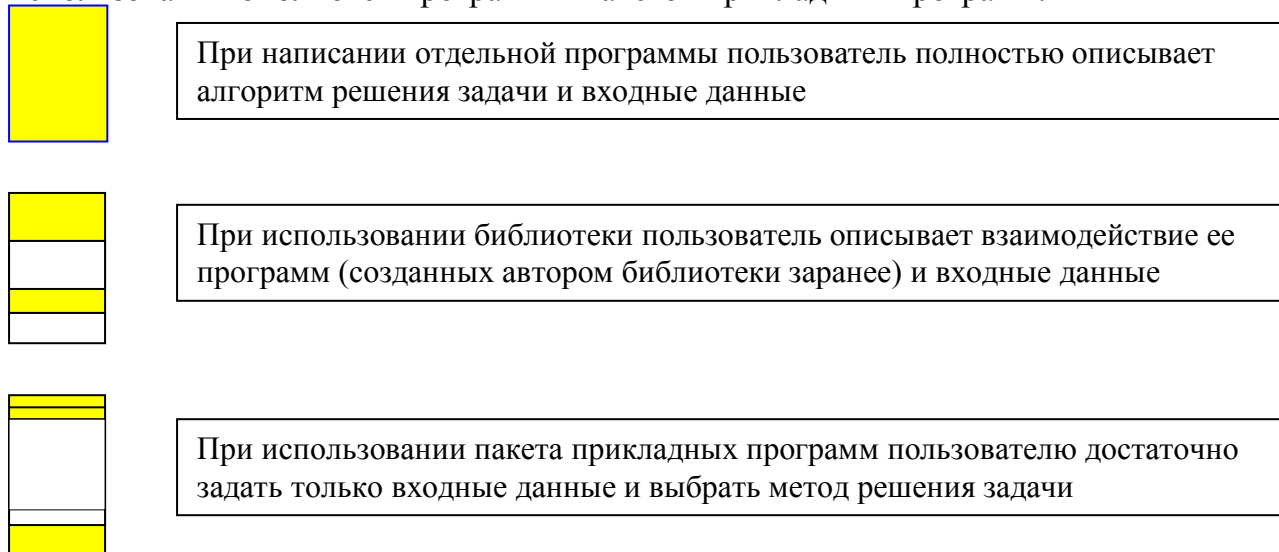


Рис. 1. Отдельная программа - библиотека программ - пакет прикладных программ.

Проиллюстрируем эту схему на примере Задания № 1 из книги: Трифонов Н.П., Пильщиков В.Н. Задания практикума на ЭВМ (1 курс). Учебное пособие, 2-е исправленное издание. – М.: МГУ, 2001.

ПОСТАНОВКА ЗАДАЧИ

С заданной точностью eps вычислить площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых $y=y_1(x)$, $y=y_2(x)$ и $y=y_3(x)$ определяются вариантом задания.

При решении задачи необходимо:

- с некоторой точностью eps_1 вычислить абсциссы точек пересечения кривых, используя предусмотренный вариантом задания метод приближенного решения уравнения $F(x)=0$; отрезки, где программа будет искать точки пересечения и где применим используемый метод, определить вручную;
- представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью eps_2 по квадратурной формуле, предусмотренной вариантом задания.

Величины eps_1 и eps_2 подобрать вручную так, чтобы гарантировалось вычисление площади фигуры с точностью eps .

Варианты задания различаются:

1. Методами приближенного решения уравнений (метод деления отрезка пополам, метод хорд, метод касательных, комбинированный метод);
2. Методами приближенного вычисления определенных интегралов решения – квадратурными формулами (формула прямоугольников, формула трапеций, формула Симпсона).
3. Набором функций (y_1 , y_2 , y_3).

Студенты первого курса должны написать программу (на языке Паскаль) для решения одного из вариантов задачи. В описании задания сформулированы требования к программе:

1. В программе предусмотреть печать как площади заданной фигуры, так и абсцисс точек пересечения кривых.

2. Описать в программе процедуру $root(f,g,a,b,eps,x)$, вычисляющую с точностью eps корень x уравнения $f(x)=g(x)$ на отрезке $[a,b]$. (Если используется метод касательных или комбинированный метод, то у $root$ должны быть еще параметры $f1$ и $g1$ - производные функций f и g .)

3. Описать в программе функцию $integral(f,a,b,eps)$, вычисляющую с точностью eps величину определенного интеграла от функции $f(x)$ на отрезке $[a,b]$.

4. Процедуру $root$ и функцию $integral$ следует предварительно протестировать.

В разделе «Методические указания» даются пояснения и указания, которые следует учесть при разработке алгоритма и его реализации:

Для корректного применения предложенных методов приближенного решения уравнения $F(x)=0$ (где $F(x)=f(x)-g(x)$) необходимо найти отрезок $[a,b]$, на котором уравнение имеет ровно один корень. Достаточное условие для этого таково: на концах отрезка функция $F(x)$ имеет разные знаки и на всем отрезке производная функции не меняет знак. Кроме того, для методов хорд и касательных, а также комбинированного метода обязательно требуется, чтобы на данном отрезке первая и вторая производные функции не меняли свой знак (не обращались в ноль).

В *методе деления отрезка пополам* определяется средняя точка c отрезка $[a,b]$ и из двух отрезков $[a,c]$ и $[c,b]$ выбирается тот, на концах которого функция $F(x)$ имеет разные знаки. К выбранному отрезку применяется та же процедура. Процесс деления отрезков прекращается, когда длина очередного отрезка станет меньше требуемой точности eps ; за корень уравнения можно принять любую точку этого отрезка. В остальных трех методах приходится учитывать свойства производных заданных функций.

Квадратурная формула для приближенного вычисления интеграла I от функции $F(x)$ на отрезке $[a,b]$ по формуле прямоугольников имеют следующий вид (n - число разбиений отрезка $[a,b]$):

$$I \cong I_n = h(F_0 + F_1 + \dots + F_{n-1}), \quad \text{где } F_i = F(a + (i+0.5)h), \quad h = (b-a)/n$$

Для обеспечения требуемой точности eps при приближенном вычислении интеграла I по квадратурной формуле нужно подобрать соответствующее число n разбиений отрезка интегрирования. Для достижения требуемой точности обычно используется следующий метод: берется некоторое начальное число разбиений n_0 (например, 10 или 20) и последовательно вычисляются значения I_n при n , равном $2n_0$, $4n_0$, $8n_0$ и т.д. Известно *правило Рунге*

$$|I - I_n| \cong p |I_n - I_{2n}|$$

(для формулы прямоугольников $p=1/3$). Согласно этому правилу, когда на очередном шаге величина $p|I_n - I_{2n}|$ окажется меньше eps , в качестве приближенного значения для I можно взять I_n или, что лучше, I_{2n} .

(Иллюстративный материал, не входит в программу коллоквиумов и экзамена).

Если бы при выполнении задания можно было воспользоваться некоторой *библиотекой*, в состав которой входят программы:

- **приближенного решения уравнений** (метод деления отрезка пополам, метод хорд, метод касательных, комбинированный метод);
- **приближенного вычисления определенных интегралов** (формула прямоугольников, формула трапеций, формула Симпсона),

объем работы существенно уменьшился бы. Автору программы нужно было бы выбрать (для указанного варианта задания) версии процедуры $root$ и функции $integral$ и написать относительно небольшую по объему управляющую программу.

Использование подходящего пакета прикладных программ позволяет еще более существенно сократить объем работы.

Вот как выглядит диалог (некоторые мелкие детали опущены) пользователя, решающего задачу на вычисление площади криволинейного треугольника, при применении пакета прикладных программ **MathCAD** – системы, которая используется для автоматизации инженерных расчетов. Текст, набираемый пользователем на клавиатуре, выделен полужирным шрифтом и **зеленым** фоном. Этот текст и ответы системы (координатная сетка, графики функций, результаты вычислений) появляются на экране; результаты вычислений выделены синим цветом. Для формирования графиков функций, написания некоторых формул и т.п. используются функциональные и горячие клавиши.

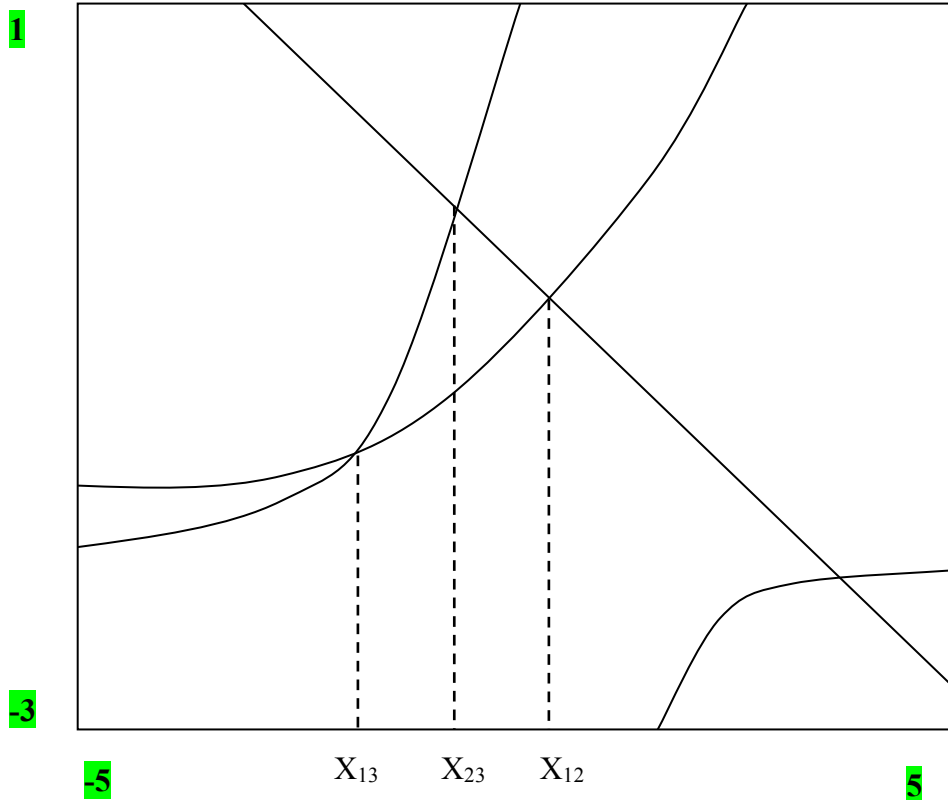
В системе используется метафора листа бумаги (бесконечного снизу и справа), на котором отображается существенная для решения задачи информация.

$$x := -5, -4.95 \dots 5$$

$$y1(x) := \exp(x) + 2$$

$$y2(x) := -2x + 8$$

$$y3(x) := -5/x$$



$$x := -3$$

$$x13 := \text{root}(y1(x) - y3(x))$$

$$x13 = -2.391$$

$$x := -1$$

$$x23 := \text{root}(y2(x) - y3(x))$$

$$x23 = -0.550$$

$$x := 1$$

$$x12 := \text{root}(y1(x) - y2(x))$$

$$x12 = 1.252$$

$$sq := \int_{x13}^{x23} y3(x) dx + \int_{x23}^{x12} y2(x) dx - \int_{x13}^{x12} y1(x) dx$$

$$sq = 9.807$$

Пять поколений языков общения с машиной. Языки 4GL и 5GL

В компьютерных науках выделяют **пять поколений языков общения с машиной (Generation Languages – GL)**:

1GL –машинные языки

2GL –автокоды (языки ассемблера)

3GL –языки программирования высокого уровня

4GL –простейшие языки запросов информационных систем,
языки запросов, ЯОД, ЯМД (баз данных),
языки генераторов отчетов,
языки поддержки принятия решений (статистический / прогнозный анализ),
языки генераторов приложений (RPG – Report Program Generator)

5GL – должны обеспечивать автоматическое построение прикладных программ на основании данных, которые вводятся в компьютер в виде максимально удобном и привычном для пользователя, не знакомого с программированием.

Считается, что переход к каждому новому поколению повышает производительность работы с компьютером примерно на порядок (в 10 раз).

Качественное отличие языков 4GL и 5GL от языков предыдущих поколений – заключается в том, что эти языки служат *не для описания алгоритмов, а для спецификации решаемых пользователем задач.*

Функциональные требования к 4GL:

интегрированное множество функций,
язык очень высокого уровня,
совместимость с языками 3GL,
дружественный интерфейс,
средства анализа и моделирования данных,
высокая эффективность при сильной нагрузке,
поддержка стандартных СУБД

Интеллектуализация ЭВМ

Внедрение программного обеспечения, позволяющего конечному пользователю решать на ЭВМ *любые* задачи, возникающие в сфере его профессиональной деятельности без каких бы то ни было посредников (в том числе, программистов).

Понятие об автоматическом синтезе программ

Автоматический синтез программ – автоматическое/автоматизированное построение программы по описанию ее назначения или действия, называемому обычно *спецификацией программы*.

Спецификация программы vs. спецификация задачи.

Эти два термина часто используются как синонимы. В то же время, *спецификация задачи* – это именно описание, того **что** нужно получить без каких бы то ни было указаний по поводу того, **как** это получить. Спецификация программы содержит те или иные данные о способах достижения поставленной в спецификации задачи цели (например, указания алгоритмического характера или рекомендации по интерфейсу).

Три базовых подхода к автоматическому синтезу программ:

- ***дедуктивный*** (построения доказательства возможности получения значений выходных переменных по значениям входных и "извлечение" программы из схемы доказательства);
- ***индуктивный*** (в том числе, синтез по примерам работы программы или по промежуточным ее результатам);
- ***трансформационный*** (в том числе, синтез по прецедентам).

"Судьба" автоматического синтеза программ.

В начале исследований в этой области обычно предполагалось, что на вход поступает сформированная пользователем спецификация необходимой ему программы, а по окончании работы синтезатора выдается эта программа. Такая схема предполагает исчерпывающую полноту и точность исходной спецификации, а также ее неизменность (то есть невозможность дополнения и исправления ее в процессе синтеза).

Написать подобную спецификацию программы зачастую сложнее, чем написать эту программу.

Удобство и эффективность человеко-машинного взаимодействия; эргономические аспекты

Вспомним: Традиционно основное внимание уделялось *точности результатов* работы вычислительных систем. Гораздо меньше внимания уделялось проблеме *удобства* работы с ВС. Это было вызвано объективными причинами: узкий круг пользователей, недостаточная надежность работы ЭВМ первых поколений.

Мы говорили об одном из направлений (одной из тенденций) развития прикладного программного обеспечения – передаче компьютеру (ВС) функций, которые на предыдущих этапах развития ППО выполнял человек.

По мере резкого увеличения и расширения круга пользователей, имеющих непосредственный доступ к ЭВМ, стало ясно, что необходим учет физиологических, эмоциональных, психологических и социологических особенностей человека, взаимодействующего с ЭВМ.

Второе направление (вторая тенденция) – достижение максимального *удобства* работы с ВС.

В этой связи мы обратимся к специальным исследованиям проблем взаимодействия человека с машиной (в широком понимании этого слова).

Соответствующие вопросы изучаются самостоятельной дисциплиной – эргономикой.

Эргономика (социальная инженерия) Ergonomics; Human engineering

От греч. Ergon – работа + Nomos - закон

(Мальковский М.Г., Соловьев С.Ю. Glossary Commander Служба тематических толковых словарей

http://www.glossary.ru/cgi-bin/gl_find.cgi?ph=%FD%F0%E3%EE%ED%EE%EC%E8%EA%E0&action.x=36&action.y=8
) (02.02.2020.).

некоторые традиционные проблемы эргономики

по материалам сайта Эргономика рабочего пространства: основные факты

<http://www.office-solutions.ru/ru/office-space/ergonomics/work-space/> (02.02.2020.)

Эргономика рабочего пространства: основные факты

Эргономика – это наука об удобстве, об организации рабочего пространства для комфортного и эффективного труда работника, исходя из физических и психологических особенностей человеческого организма.

Фактически, это свод основных правил для обеспечения безопасности труда и комфорта в условиях современного офиса. День за днем мы проводим в неестественном для человеческого тела положении, за столом, перед монитором компьютера при

искусственном освещении. Это пагубно влияет на наше здоровье, хоть и медленно, поэтому многие не замечают вреда, а иногда и не связывают свои хронические болезни с собственным невниманием к рабочему месту.

Важность знания эргономики и правильного положения за работой – это наше здоровье.

Для организации внимание к эргономике – это рост производительности труда, инвестиции в человеческий капитал компании. Здоровый человек трудится продуктивней и дольше, чем больной и отвлекающийся на неудобство, болезненные ощущения и внешние раздражения.

Немного цифр:

Европейское исследование о потерянном рабочем времени

3,6 миллионов.... потерянных рабочих дней (боли в спине)

33 дня средняя продолжительность отсутствия на работе (боли в спине)

500 миллионов Евро расходы на лечение проблем со спиной

1,3 миллиардов Евро расходы на болезни, связанные с проблемами со спиной

Источник: “Les Assises Internationales de Grenoble” I.N.R.S.



А теперь об экономии и эффективности:

До 85 — 90% функциональных расходов компаний связаны с сотрудниками, тем не менее, большинство административных служб направляют основные ресурсы на управление расходами оставшихся — 10 — 15 % (чаще всего, это материальные активы компании).

В то же время, 63 % офисных сотрудников считают, что удобная и приятная рабочая обстановка — главный фактор, чтобы работать более продуктивно.

Что важно для комфортной работы?

удобство рабочего кресла;

аксессуары для рабочего места (удобство, доступность, расстояние вытянутой руки);

акустика, низкое шумовое загрязнение;

освещение;

температурное окружение;

комфортность цветов.

Эргономика – научно-прикладная дисциплина, занимающаяся изучением и созданием эффективных систем, управляемых человеком. Эргономика изучает движение человека в процессе производственной деятельности, затраты его энергии, производительность и интенсивность при конкретных видах работ. Эргономика подразделяется на микро-, миди- и макроэргономику.

Микроэргономика – исследование и проектирование систем «человек-машина».

Эргономика программного обеспечения – подраздел микроэргономики, ориентированный на системы «человек-компьютер», «человек-компьютер-человек», «человек-компьютер-процесс», «человек-программа» и т.п.

Мидиэргономика – исследование и проектирование систем «человек-коллектив», «коллектив-машина», «человек-сеть», «коллектив-организация».

Мидиэргономика исследует взаимодействия на уровне рабочих мест и производственных задач. В сферу интересов мидиэргономики входят:

- проектирование организаций;
- планирование работ;
- обитаемость рабочих помещений;

- гигиена труда;
- проектирование интерфейсов сетевых программных продуктов и др.

Макроэргономика – исследование и проектирование систем «человек-общество», «организация-система организаций».

Интерфейс – полная совокупность сведений о входных и выходных сигналах, которыми могут обмениваться процессоры данных (устройство – устройство, программа – датчики состояния среды, человек – компьютерная система).

Другая (встречающаяся) трактовка термина

Интерфейс – совокупность программных и аппаратных средств, поддерживающих подобные обмены данными.

Пользовательский интерфейс – интерфейс между конечным пользователем и компьютерной системой.

Основные требования к интерфейсу:

- **точность,**
- **удобство (комфорт для пользователя):**
 - **социальные факторы** (возраст, образование),
 - **физический комфорт** (не может прочитать сообщение из-за неудачного шрифта, фона)
 - **психологический комфорт** (может прочитать сообщение, но не понимает его).

Эргономические характеристики:

Конструктивные особенности оборудования.

Доступность и надежность системы.

Качество разработки диалога.

Некоторые эргономические факторы (при распределении функций "человек – машина"):

- стоимость выполнения некоторой функции (учитывать обучение персонала и пр.);
- время выполнения этой функции;
- принцип **преимущественных возможностей**;
- принцип **ответственности**;
- принцип оптимального темпа обмена информацией;
- принцип **гуманизации труда**

(**гуманизация труда** - "совершенствование управления трудовой деятельностью с целью предоставления работнику возможности раскрыть свои производительные резервы, прежде всего интеллектуальные и психологические".)

Гуманизация труда [HTML] <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/gumanizaciya-truda.html> (08.02.2020).

Отдельный аспект пользовательского интерфейса – его информационная безопасность. Она предполагает защиту пользователя от нежелательного, в том числе, социально неприемлемого контента, появляющегося (будем надеяться, что редко) по запросам/командам самого пользователя, либо в результате незаконных действий других пользователей сети (например, контекстная реклама).

Особо важны эти проблемы (как и проблемы времени работы за компьютером, правильной позы и т.п.) в случае пользователя-ребенка. Поэтому существенным компонентом современного программного обеспечения должны быть **системы родительского контроля**. К сожалению, в рамках нашего курса мы не имеем возможности даже предварительного знакомства с этой первостепенной важности проблематикой, не связанной напрямую с нашей главной темой – **интеллектуаль-ный интерфейс**.

Вспомним введенный ранее термин "**Интеллектуализация ЭВМ**":

Внедрение программного обеспечения, позволяющего конечному пользователю решать на ЭВМ **любые** задачи, возникающие в сфере его профессиональной деятельности без каких бы то ни было посредников (в том числе, программистов).

Другое определение Интеллектуализации ЭВМ (^{def def} создание средств интеллектуального интерфейса) устанавливает непосредственную связь с основным понятием нашего курса!

Интеллектуальный интерфейс – совокупность программных и аппаратных средств, позволяющая конечному пользователю решать на компьютере характерные для его повседневной деятельности задачи без помощи посредников-программистов.

Расширение взаимодействия между человеком и компьютером с помощью:

- увеличения диапазона способов ввода и вывода;
- обогащения грамматики ввода и вывода;
- попытки кооперации с пользователем в достижении целей.

В идеале система должна иметь «модель мира задачи», над которой работают система и пользователь и которая близка модели этого мира в представлении пользователя.

Иногда **интеллектуальный интерфейс** рассматривается как развитие **дружественного интерфейса**.

Дружественный интерфейс:

- естественные языковые конструкции и структуры меню (не требуется знание синтаксиса формальных языков общения с компьютером);
- «интуитивный» уровень взаимодействия с компьютером, не требующий длительного обучения (для профессионала – несколько часов);

- разнообразные средства общения, пригодные для пользователей различного уровня подготовки (командный язык, меню, пиктограммы, диаграммы и др.);
- для каждого уровня пользователя адекватные возможности в: меню, запросах, подсистеме помощи;
- работа в реальном времени (необходимая скорость в диалоге);
- использование манипулятора типа «мышь», «горячих клавиш», сенсорного экрана и др.;
- минимальное использование клавиатуры;
- «интеллектуальные» средства (устойчивость к ошибкам, широкое использование принципа «по умолчанию», опережающий ввод).

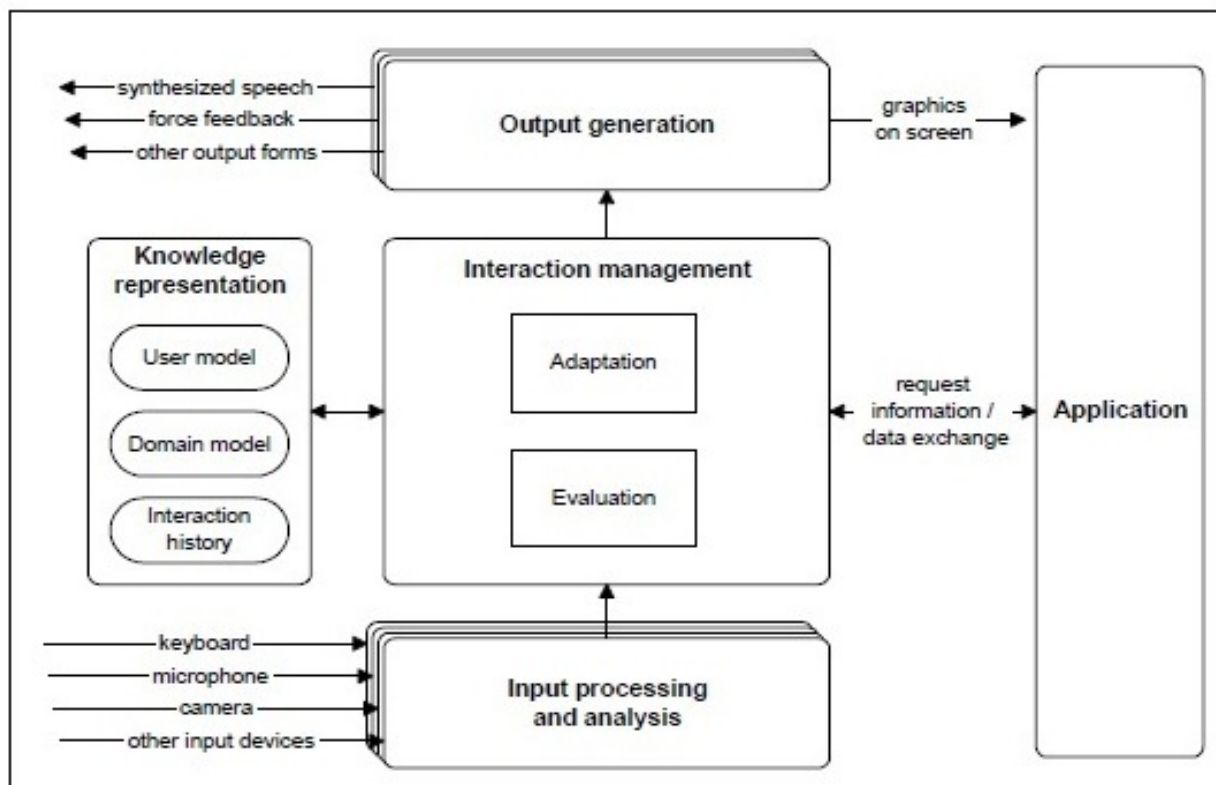


Рис. 1. Общая архитектура Интеллектуального интерфейса [2-литература для самостоятельной работы, стр. 9].

Рисунок из книги Дженфера Тидвелла "Разработка пользовательских интерфейсов" представляется достаточно банальным и не в полной мере отражающим суть и особенности интеллектуального интерфейса.

Блок обработки и анализа входных данных (*Input processing and analysis*) получает информацию, передаваемую с клавиатуры, микрофона, видеочамеры или других устройств ввода.

Управляющий блок (*Interaction management*) содержит модули адаптации и оценки.

Он получает информацию от первого блока и может взаимодействовать (обмен информацией – *request information/data exchange*), как с блоком **Приложение** (*Application*), так и с Базой знаний (*Knowledge representation*), в которой представлены: Модель пользователя (*User model*), Модель предметной/проблемной области (*Domain model*) и "История диалога" (*Interaction history*).

Что имеется в виду под Базой знаний, Моделью пользователя, Моделью предметной области и Историей диалога не очень понятно ни по этому рисунку, ни по сопроводительному тексту.

Блок вывода данных (*Output generation*) обеспечивает генерацию и выдачу синтезированных речевых сообщений (*synthesized speech*), тактильных данных (*force feedback*) и других (не указанных на рисунке явно) видов выходных сообщений.

Этот же блок может передавать графические изображения **Приложению** (*Application*).

Возвращаемся к рассмотрению общих характеристик человеко-машинного интерфейса.

Основные аспекты интерфейса:

- процессы ввода-вывода
- диалог (связывает фоновые процессы в одно целое)

(о термине *диалог* vs. *дилог*)

Процессы ввода-вывода.

Процессы ввода-вывода служат для того, чтобы принять от пользователя данные и передать ему данные через различные физические устройства.

Продолжаем рассматривать: **Основные аспекты интерфейса: процессы ввода-вывода и диалог**

С каждым устройством связан свой процесс ввода-вывода, задача которого – воспринять данные от пользователя (например, воспринять цифру 5 после нажатия соответствующей клавиши) и преобразовать их во внутреннее представление, с которым может работать процесс диалога (например, в ASCII код – 00110101). Если использовалось устройство речевого ввода, после произнесения слова «пять» должно быть получено такое же внутреннее представление.

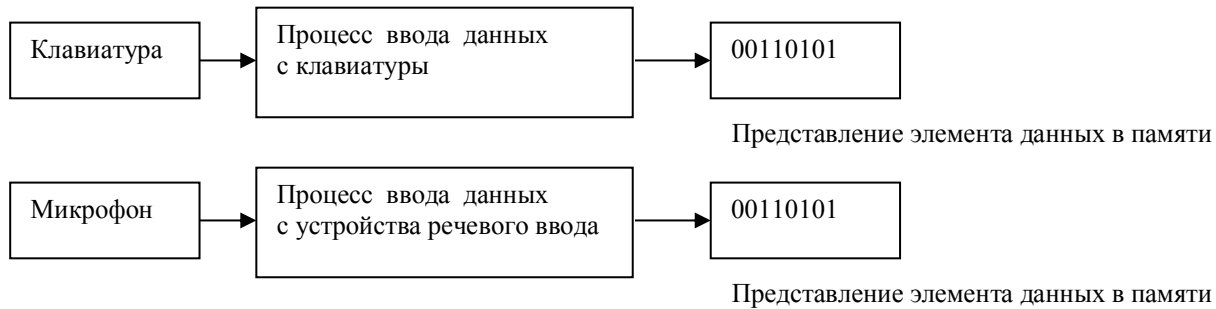


Рис. 2. Процессы ввода-вывода [2, с. 35].

Диалог «человек-компьютер» – обмен информацией между пользователем и вычислительной системой (с помощью терминала и по определенным правилам).

Задачи диалога:

- Определение задания.
- Передача/Прием входных данных и размещение их в переменных соответствующего процесса в нужной форме.
- Вызов процесса.
- Вывод/Получение результатов обработки.

Пользователь вычислительной системы через интерфейс посылает входные данные и получает выходные. Интерфейс обеспечивает связь между пользователем и процессами, выполняющими те или иные задания (в нашем примере это задания, поручаемые операционной системе: *копировать один файл в другой, удалить файл* и т.п.).

Виды сообщений в диалоге:

Пользователь → *команда (входные управляющие сообщения), данные (входные данные)*

Возможны и сложные сообщения, объединяющие команду и данные.

Система → *подсказка* (побуждение ввести команду или данные в некотором формате; в случае общения между людьми: поднятые брови или список вариантов, из которого участнику диалога нужно выбрать один),

данные (результаты, т.е. данные, которые возвращает процесс по окончании обработки),

сообщение о состоянии (что произошло/происходит в системе, например, сообщение о получении и начале обработки

введенных по запросу системы

или о том, что система

– при большой

данных

продолжает работу

задержке),

сообщение об ошибке (сигнал диалогового процесса о том, что дальнейшее выполнение работы невозможно, т.к. невозможно, например, обработать очередное сообщение пользователя),

справка (что можно/нужно делать и почему в ситуации, когда пользователь этого не знает).

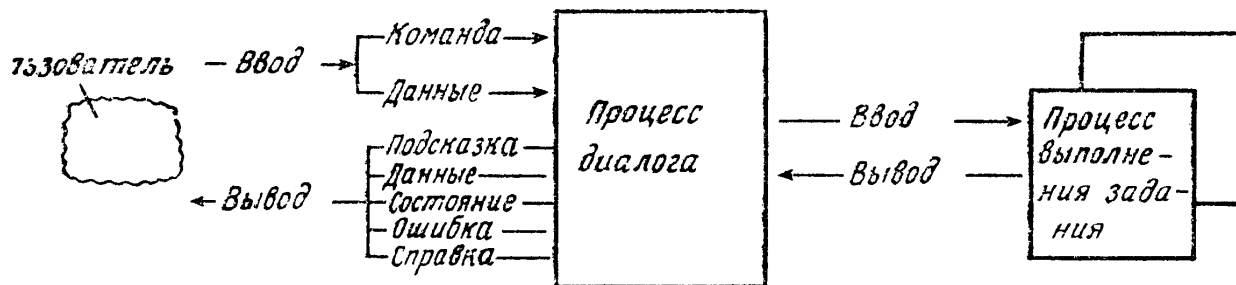


Рис. 2.5. Классификация сообщений в соответствии с выполняемыми ими функциями.

Рис. 3. Классификация сообщений в диалоге – рисунок из книги Коутс Р., Влейминк И. Интерфейс "человек-компьютер" [2, с. 39]

Пример диалога (телефонный разговор между продавцом и покупателем):

Продавец: **Доброе утро. Магазин запчастей.** (*Состояние*)

Чем могу помочь? (*Подсказка*)

Покупатель: Меня интересует цена некоторых деталей для моего автомобиля. (*Команда*)

Продавец: **Каких деталей?** (*Подсказка*)

Покупатель: Зубчатого колеса номер P12347. (*Входные данные*)

Продавец: **Боюсь, что для вашей марки автомобиля вы указали неправильный номер.**

Номер должен начинаться с Q7. (*Сообщение об ошибке*)

(ПАУЗА. ПРОДАВЕЦ ЖДЕТ.) (*Подсказка*)

Покупатель: А какие детали имеются для ремонта карбюратора? (*Команда*)

Продавец: **Итак, имеются . . .** (*Выходные данные*)

Покупатель: Мне нужна деталь Q7914.

(Команда + Входные данные)

Продавец: Сейчас посмотрю цену.

(Состояние)

Она стоит 5 фунтов.

(Выходные данные)

Что-нибудь еще?

(Подсказка)

...

Всегда в основе диалога лежит следующий цикл:

- 1) явный или неявный запрос на ввод данных;
- 2) ввод данных через процесс ввода;
- 3) проверка входных данных,

который повторяется пока не будут приняты приемлемые входные данные. Если выводится запрос на ввод команды, следующий шаг обработки будет зависеть от введенной команды.

Диалог, управляемый системой, – система жестко задает, какое задание можно выбрать и какие данные вводить.

Пример:

Продавец: Что желаете?

Покупатель: Яблоки, пожалуйста?

Продавец: Антоновку, коричные, мельбу?

Покупатель: Антоновку.

Продавец: Сколько?

...

Диалог, управляемый пользователем, – пользователь подает команду на выполнение нужного на данном этапе задания; если ответ системы его устраивает, подаются новые команды. В некоторых случаях возможно обращение к подсистеме помощи.

Пример:

Покупатель: 2 кило антоновки, пожалуйста.

Проверка входных данных (необходима для предотвращения неверных действий компьютерной системы – вплоть до аварийного завершения работы – в случае некорректных/ошибочных входных данных):

- 1) сравнение с элементами набора;
- 2) сравнение с шаблоном (дд/мм/гг);

3) проверка грамматической правильности.

К этой проблеме мы еще вернемся на Лекциях № 6 и № 7, рассматривая вопросы ошибок человека при вводе текстов в компьютер и вопросы поддержки работы пользователя.

Подсказки

Подсказка, как мы знаем, – выходное сообщение системы, побуждающее пользователя вводить данные. В ответ на подсказку пользователь может вызвать процесс выполнения некоторого задания или же передать процессу выполнения задания некоторые данные.

Примеры подсказок:

Коутс и Влейминк отмечают, "что существует много способов, с помощью которых человек может выразить свою готовность принять сообщение от другого участника диалога. Аналогично существует ряд форматов вывода подсказок в диалоге человек-компьютер [2, с. 48].

1) Меню (будет позже) – самый сложный формат, когда наряду с запросом на ввод сообщения выводятся допустимые форматы ввода; очень удобно в ситуациях, когда диапазон правильных входных данных не слишком велик.

```

DATE : установить текущую дату
DIR   : вывести каталог
ERA   : удалить файл или файлы с диска
RIP   : скопировать файлы
REN   : переименовать файлы
STAT : вывести данные об использовании дисковой памяти
TIME : установить текущее время
TYPE : вывести содержимое файла на экран
Команда?
```

Рис. 2.8 Пример меню.

Рис. 4. Пример меню – рисунок из книги Коутс Р., Влейминк И. Интерфейс "человек-компьютер", [2, с. 49]

Коутс Р., Влейминк И. Интерфейс "человек-компьютер" – М.: Мир, 1990.

2) Указание на требуемый формат (при этом система, не выводя возможные ответы пользователя – как это было в случае меню – описывает, с использованием общепринятых обозначений, только формат ожидаемого ответа):

(Счет от (дд/мм/гг)).

3) Если нужно ввести несколько подобных сообщений, используется форма для заполнения (предшественник "экраных форм" – Лекция № 4):

Номер счета	[]
От	[/ /]
Код поставщика	[]
Всего по счету	[]
Налог	[]

4) Компьютерный эквивалент фразы "Чем могу служить?" – сообщение о том, что система просто ожидает входные данные, не указывая, какими они могут быть (**A >**); *запрос на ввод команды*.

Типы диалога (традиционные структуры диалога):

Два основных вида: "вспоминай и набирай" и "смотри и выбирай".

Диалог типа ВОПРОС-ОТВЕТ (Q&A).

Диалог на основе КОМАНДНОГО ЯЗЫКА.

Диалог типа МЕНЮ.

Диалог на основе ЭКРАННЫХ ФОРМ

В современных системах часто применяются комбинированные типы диалога.

Диалог типа ВОПРОС-ОТВЕТ

Построен по аналогии с обычным интервью. Система берет на себя роль интервьюера и получает информацию от пользователя в форме ответов на ее вопросы.

Исторически первый вид диалога (именно такой вариант назывался "диалоговым режимом").

Достоинства:

возможна подсказка со стороны системы (например, формат данных: **YY/MM/DD**);
возможно (в любой момент) обращение к подсистеме ПОМОЩИ (HELP);
гибкость, удобство при большой ветвистости дерева диалога.

Недостаток: много нажатий клавиш

Описание структуры диалога типа ВОПРОС-ОТВЕТ с помощью сетей переходов.

Вершина = состояние (выводится сообщение системы, ожидается ответ П)

Дуга = переход

Метка (на дуге) = условие перехода

Пример: Схема диалога с гипотетическим ППП, решающим задачу вычисления площади фигуры (Лекция 1)

Вершина1 – «ПОСТАНОВКА ЗАДАЧИ: задание функций и требуемой точности»

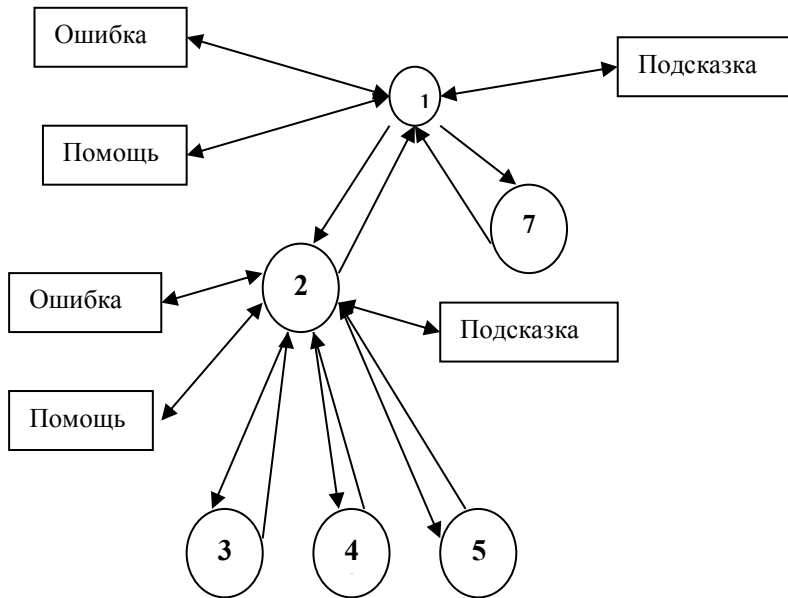
Вершина2 – «ВЫБОР МЕТОДА РЕШЕНИЯ УРАВНЕНИЯ: метод деления отрезка пополам, и др.)»

Вершина3 – «РЕШЕНИЕ УРАВНЕНИЯ МЕТОДОМ ДЕЛЕНИЯ ОТРЕЗКА ПОПОЛАМ»

Вершина4 – «РЕШЕНИЕ УРАВНЕНИЯ МЕТОДОМ ХОРД»

Вершина5 – «РЕШЕНИЕ УРАВНЕНИЯ МЕТОДОМ КАСАТЕЛЬНЫХ»

Вершина7 – «ВЫБОР МЕТОДА ВЫЧИСЛЕНИЯ ОПРЕДЕЛЕННОГО ИНТЕГРАЛА: формула прямоугольников, формула трапеций, формула Симпсона»



Фрагмент диалога с гипотетическим ППП, решающим задачу вычисления площади фигуры
(Лекция 1)

СИСТЕМА: ВВЕДИТЕ ФУНКЦИИ И ЗАДАЙТЕ ТОЧНОСТЬ ВЫЧИСЛЕНИЙ

П: $y_1(x) := \exp(x) + 2$

$y_2(x) := -2x + 8$

$y_3(x) := -5 / x$

СИСТЕМА: ОШИБКА НЕ ЗАДАНА ТОЧНОСТЬ

П: 0.001

СИСТЕМА: ОШИБКА НЕ ЗАДАНА ТОЧНОСТЬ

П: ПОМОЩЬ

СИСТЕМА: ТОЧНОСТЬ ЗАДАЕТСЯ ТАК: eps = <вещественное число>

П: eps = 0.001

СИСТЕМА: ВЫБЕРИТЕ МЕТОД РЕШЕНИЯ УРАВНЕНИЯ (ДЕЛ / ХОР / КАС)

П: МЕТОД ХОРД

СИСТЕМА: НЕИЗВЕСТНЫЙ МЕТОД

ВЫБЕРИТЕ МЕТОД РЕШЕНИЯ УРАВНЕНИЯ (ДЕЛ / ХОР / КАС)

...

Диалог на основе КОМАНДНОГО ЯЗЫКА

Система обычно ничего не выводит, кроме подсказки (имя устройства, символ "приглашения": **A>**, который означает готовность системы к работе/вводу) и, разумеется результатов выполнения команды пользователя.

Примеры:

```
A>copy
```

```
Required parameters missing
```

```
A>copy lect7.doc d:\tmp
```

```
1 file(s) copied
```

```
D:\TMP>dir
```

```
Volume in drive D has no label
```

```
Volume Serial Number is D8F8-C43C
```

```
Directory of D:\TMP
```

```
.          <DIR>                03-07-02  4:04p
..         <DIR>                03-07-02  4:04p
EE                330      03-07-02  4:04p
LECT7  DOC  49152    03-07-02  3:57p
          4 file(s)  49482 bytes
          1023932928 bytes free
```

```
D:\TMP>lect7.doc
```

```
Bad command or file name
```

Недостатки:

Пользователь должен знать синтаксис командного языка.

Часто: плохая диагностика (минимальный контроль со стороны системы).

Сравнение (в книге [2]) с сержантом (пользователь) и беспрекословно выполняющим любые команды солдатом. Ответственность за правильность команд лежит на пользователе. "Если команду невозможно выполнить, солдат может сказать об этом, не уточняя той части команды, которая невыполнима" ([2, с. 164]).

Диалог типа МЕНЮ

Исторически первый вид дружественного интерфейса.

Происхождение термина – от *меню*, которым пользуются официант и посетитель ресторана.

Суть – система предлагает набор возможных вариантов ответа (можно вводить и команды, и данные).

Основные форматы (структуры) МЕНЮ:

БЛОК ДАННЫХ:

Заголовок	МЕНЮ
ОБЪЕКТ1	Текст1
ОБЪЕКТ2	Текст2
...	...
ОБЪЕКТN	ТекстN
МЕНЮ	
Окончание	

СТРОКА:

ОБЪЕКТ1	ОБЪЕКТ2	...	ОБЪЕКТN
----------------	----------------	-----	----------------

НАБОР ПИКТОГРАММ:

ОБЪЕКТ1

ОБЪЕКТ2

ОБЪЕКТN

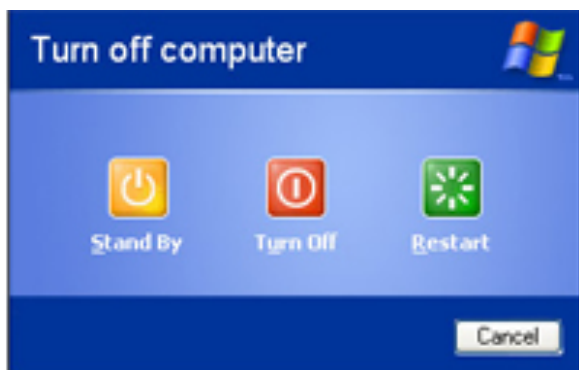


Рис. 1. Меню как набор пиктограмм.

Источник: Компьютерна система. Първи стъпки с компютъра [болгарский язык]
[HTML] <http://www-it.fmi.uni-sofia.bg/ReDisInfo/courses/modules/module1/parts/module1/part2/lesson2.html>
(03.02.2020.)

Достоинства:

возможные варианты ответов пользователя предлагает сама система (подсказка), пользователю не нужно вводить тексты, тем самым уменьшается вероятность ошибки ввода.

удобно в ситуациях

- диапазон ответов мал и все они могут быть явно отражены на экране,
- пользователю необходимо видеть возможные варианты ответов.

Выбор из МЕНЮ: просмотр списка объектов, указание на объект или ввод идентификатора (Посетитель ресторана может указать выбранное блюдо в меню либо сразу, либо, пробежав пальцем по списку; или же назвать блюдо из меню).

Иерархические МЕНЮ (Trail menu/Tree menu).

Всплывающие/спадающие МЕНЮ (Pop-up).

Скрытые МЕНЮ ("Гамбургер-меню", "выдвижное меню").

Примеры:

Основное МЕНЮ Microsoft Word (одной из старых версий):

<u>Ф</u> айл	<u>П</u> равка	<u>В</u> ид	<u>В</u> ставка	<u>Ф</u> ормат	<u>С</u> ервис	<u>Т</u> аблица	<u>О</u> кно	<u>С</u> правка
--------------	----------------	-------------	-----------------	----------------	----------------	-----------------	--------------	-----------------

Спадающее МЕНЮ для позиции Окно:

<u>Н</u> овое	<u>У</u> порядочить все	<u>Р</u> азделить
√	Lect04-2017	

Скрытое МЕНЮ при выходе из ОС Windows 2000

Предлагается (по умолчанию) "основной" вариант (например, **Shut down**).

После нажатия кнопки ▼ появляются все варианты

Log off <login>
Shut down
Restart
Stand by

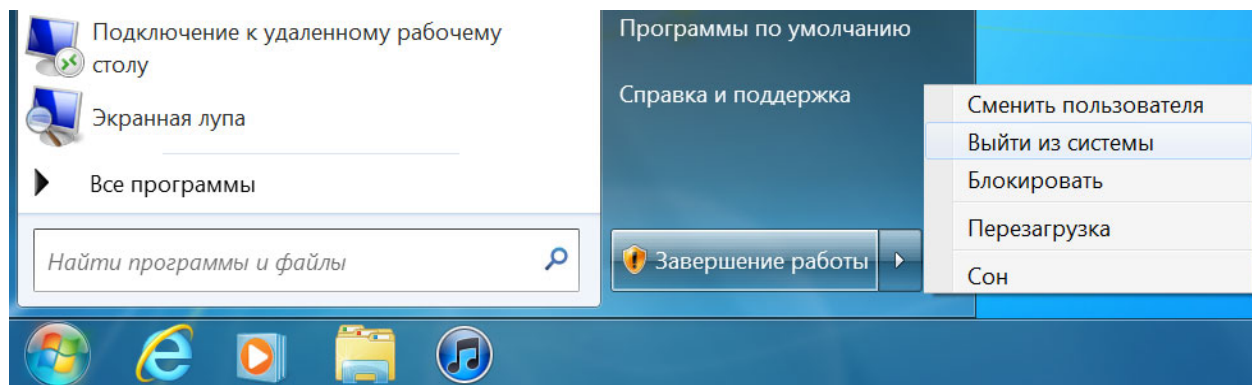


Рис. 2. Вызов скрытого меню при выходе из системы.

Источник: Compbegin.ru http://www.compbegin.ru/articles/view/_53 (03.02.2020.)

Иногда **скрытым меню** называют меню, в котором предьявляется только один вариант, а доступ к другим обеспечивается с помощью клавиш "→" и "←" или каким-то другим образом.

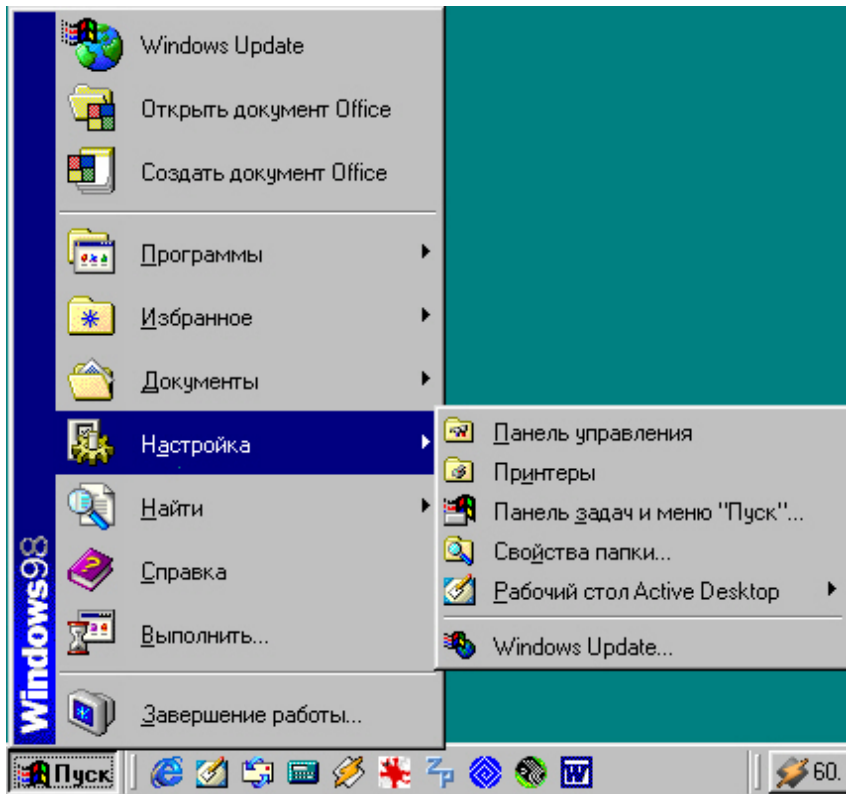


Рис. 3. Вызов иерархического меню в одной из старых версий ОС Windows.

Источник: Компьютерная библиотека КРАО [HTML]

http://web.krao.kg/13_economica/10/Glava4/Glava4.htm (03.02.2020.)

Диалог на основе ЭКРАННЫХ ФОРМ

Суть – задается сразу несколько вопросов (есть установки по умолчанию); ввод после ответа на несколько нужных вопросов.

Примеры:

1. Экранная форма для системы резервирования билетов.

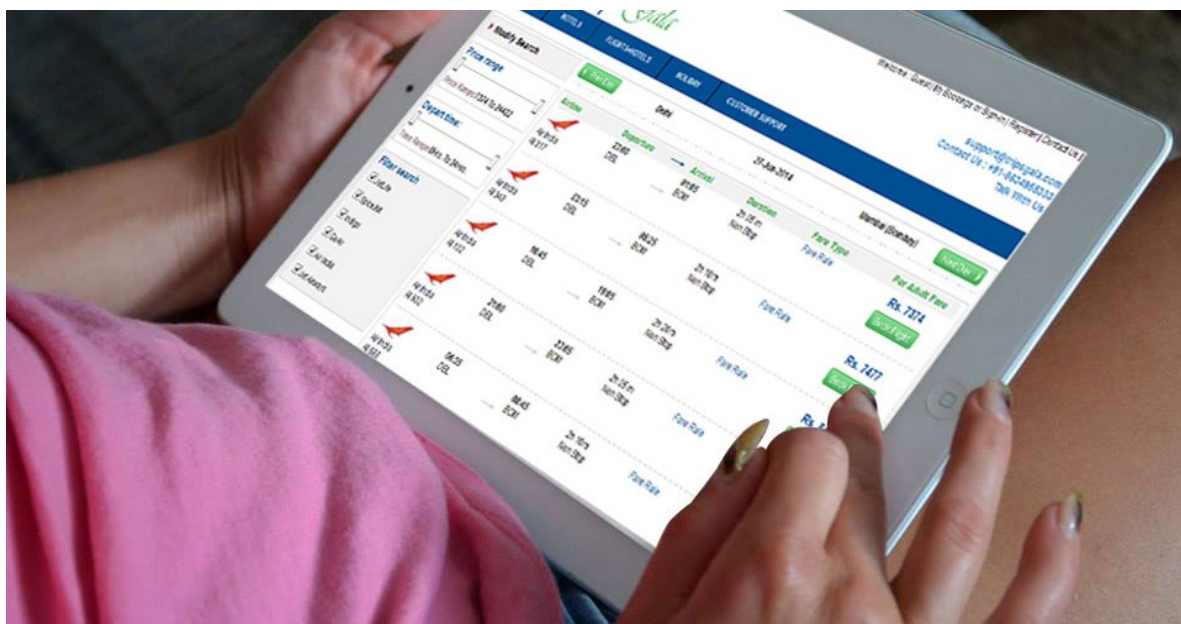


Рис. 4. Система онлайн-бронирования авиабилетов от компании Axis-softtech.

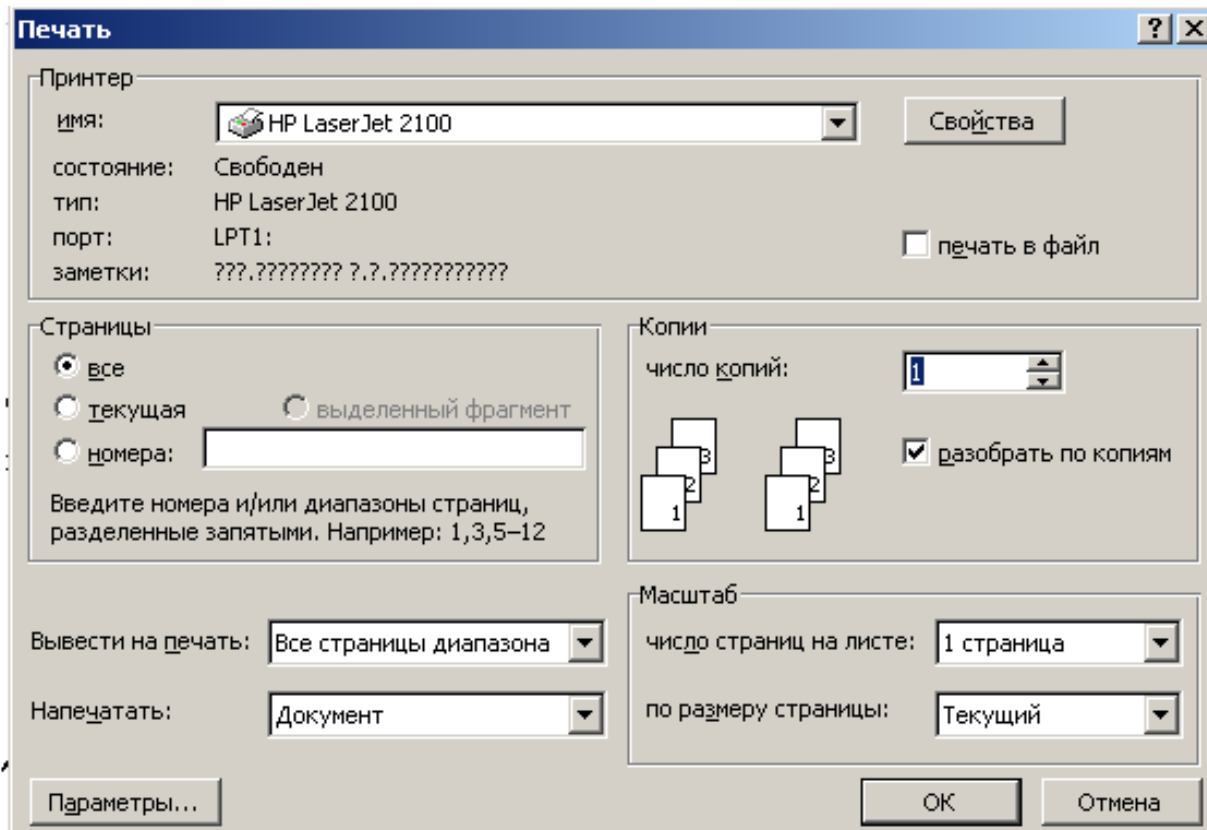
Источник: Системы бронирования авиабилетов

[\(02.02.2020.\)](http://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B_%D0%B1%D1%80%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%B0%D0%B2%D0%B8%D0%B0%D0%B1%D0%B8%D0%BB%D0%B5%D1%82%D0%BE%D0%B2)

2. Экранная форма в старых версиях текстового процессора Microsoft Word, вызываемая через МЕНЮ: Файл → Параметры страницы.

3. Экранная форма "Печать" в одной из старых версий MS Word (скриншот подготовлен лектором).

Достоинства экранных форм:



- привычно для человека (заполнение таблиц/бланков/анкет),

- быстрее чем работа по схеме ВОПРОС-ОТВЕТ или по схеме МЕНЮ,

- удобно, если можно заранее определить стандартную последовательность вводимых данных.

Недостаток: схема плоха в ситуации, когда процесс выбора ответа имеет структуру дерева.

Традиционно рассматриваются 4 основные структуры диалога, каждая из которых соответствует некоторому типу взаимодействия между людьми; эти структуры пригодны для организации взаимодействия пользователей с прикладными программами.

Сравнение четырех традиционных структур диалога (типов диалога):

Диалог **типа ВОПРОС-ОТВЕТ** пригоден и для ввода данных с клавиатуры и для выбора из списка. На диапазон значений входных данных не накладывается никаких ограничений. Однако эта структура неудобна для использования способа прямого указания.

Диалог **на основе КОМАНДНОГО ЯЗЫКА** ориентирован прежде всего на подготовленного пользователя и требует в ответ на стандартный запрос ввода ответов, описывающих (в соответствии с жесткими синтаксическими ограничениями) задачу и ассоциированные с ней данные.

Диалог **типа МЕНЮ** позволяет предъявить пользователю точный список вариантов ввода (возможно, с использованием иерархии, скрытого меню и др.) и дает возможность выбора одного из них: вводом идентификатора с клавиатуры, просмотром списка на экране и прямым указанием на объект и т.д.).

Диалог **на основе ЭКРАННЫХ ФОРМ** требует от пользователя ответов на группу вопросов. Обычно пользователь может отвечать на вопросы в любом порядке, может не отвечать на вопросы с предустановленным значением по умолчанию, может редактировать свои ответы. Обеспечиваются и гибкость, и удобство для пользователя.

Коутс Р. и Влейминк И. [2, с. 192] отмечают, что "все четыре традиционные структуры диалога представляют собой разновидности структуры типа **ВОПРОС-ОТВЕТ**:

МЕНЮ = ВОПРОС-ОТВЕТ с опережающей справкой;

ФОРМА = ВОПРОС-ОТВЕТ с опережающим вопросом;

КОМАНДА = ВОПРОС-ОТВЕТ с опережающим ответом."

Они отмечают также, что (несмотря на эту общность), разные части достаточно крупной программной системы обладают различными характеристиками, что предполагает использование комбинаций основных структур.

Напомним, что об этом мы говорили в самом начале лекции ("комбинированные типы диалога").

Многооконные WIMP-интерфейсы

В середине 80-х гг. XX века появилось понятие – **Многооконные WIMP-интерфейсы**. В наше время они обычно реализуются на уровне операционной системы.

Аббревиатура **WIMP** трактуется так:

W - информация представлена на экране в виде **окон (Windows)**.

I - объекты представлены в виде **пиктограмм (Icons)**.

M - выборка производится с помощью манипулятора типа «**мышь**» (**Mouse**).

P - используются меню: «всплывающие» (**Pop-up**), «вытягиваемые» (**Pull-down**).

Для их реализации необходимы достаточные ресурсы:

- память прямого доступа (RAM) для представления изображений в несколько раз превышающих объем физического экрана;
- экран с поточечной адресацией для представления графических изображений (в том числе, пиктограмм);
- адекватные вычислительные мощности для быстрой перерисовки экрана и работы в мультипрограммном режиме.

Три метафоры:

Конкретный объект (папка, файл).

Не нужно изучать, запоминать и вводить директивы типа **copy MyFile/My Directory**.

Файл может быть скопирован "буксировкой" его с помощью манипулятора "мышь" по нужную иконку каталога (*прямое манипулирование*).

«*Что вижу, то и получу*» (фактический эффект любого действия немедленно отображается на экране). Более точно было бы – "*видишь то, что получил*".

В прежних интерфейсах не выполнялся компьютерный эквивалент таких действий, как, например, копирование (желательно увидеть и оригинал, и копию).

Поэтому при работе пользователь вынужден был работать более внимательно (→ напряженно) и выполнять лишние действия по проверке того, **что получилось**.

Рабочий стол и Буфер вырезок.

Метафора **Рабочий стол** предполагает, что интерфейс поддерживает доступ к множеству различных информационных источников и форматов, позволяет пользователю легко переключаться с одного источника на другой ("тасовать бумаги на столе"). Столь же легко можно менять тип задания (финансовое планирование с помощью электронных таблиц → подготовка текста для написания отчета; с передачей/пересылкой информации).

При этом доступны вспомогательные средства: часы, калькулятор, записная книжка и др.). Как и в прежних интерфейсах есть возможность временно хранить в специальном буфере изъятые части документа. В **WIMP**-интерфейсах используется "магический" **Буфер вырезок**, который дает возможность вставки стольких копий вырезанного фрагмента, сколько требуется пользователю.

Окно как элемент интерфейса.

Окно – специальная область (обычно прямоугольная) физического окна (экрана), с помощью которой пользователь обозревает отдельные аспекты своего взаимодействия с компьютерной системой/задачей.

Концепция окон в компьютерных системах не нова. Однако ранее в компьютерных системах использовались окна фиксированных размеров, занимающие фиксированное положение на экране (такие окна называли иногда *черепащей*; использовался и термин "*покрытие черепащей*"). Так, в текстовом редакторе WordStar при редактировании текста использовались:

"окно статуса" (1 строка);

"окно инструкций" (строки 2-9),

"окно табуляции" (1 строка);

"текстовое окно" (оставшаяся часть экрана);

в некоторых случаях можно было вызвать "окно для ответа" по схеме вопрос-ответ).

В случае многооконного интерфейса отображение на физическом экране тех частей буфера, которые соответствуют окнам, реализует процесс *управления отображением*.

Он реализует основные **Операции с окнами**:

- открытие,
- закрытие,
- свертка,
- выбор активного окна,
- перемещение окна (относительно его буфера),
- перемещение окна (относительно экрана),
- изменение атрибутов окна.

При выполнении этих операций обычно требуется перерисовка физического экрана.

Активное окно (часто, последнее открытое).

В **WIMP**-интерфейсах выбор активного окна и другие операции с окнами осуществляются обычно прямым манипулированием с помощью какого-либо позиционирующего устройства.

Пиктограмма (Icon) – небольшое окно с изображением, отражающим содержание/функции некоторого буфера/файла/программы.

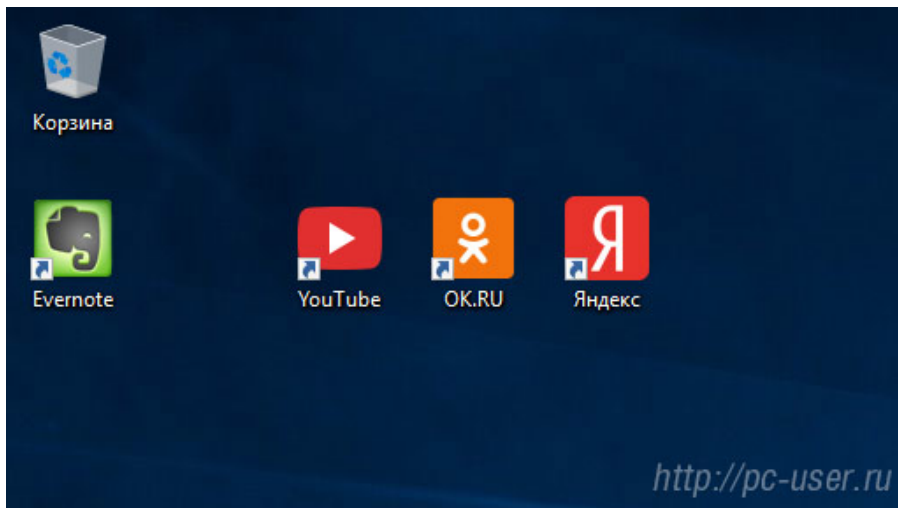


Рис. 5. Примеры пиктограмм.

https://www.pc-user.ru/view_post.php?id=247 (28.02.20.)

При раскрытии пиктограммы содержимое такого буфера обычно появляется в окне большего размера, возможно, содержащем новые пиктограммы (пример, раскрытие папки с файлами).

Пиктограммы появляются, например, при установке нового программного обеспечения.

Есть возможность определять свои собственные пиктограммы.

Ярлык (ShortCart) – пиктограмма + ссылка на соответствующий буфер/файл/программу/адрес в Интернете.

Баннер (в Интернете) – окно (графическое изображение или текстовый блок) рекламного характера, являющийся гиперссылкой на веб-страницу с расширенным описанием продукта или услуги. Обычно при создании баннера используются анимационные (а иногда и звуковые) эффекты.

Отметим, что термин **баннер** (англ. **banner** — флаг, транспарант) использовался и в докомпьютерные времена для обозначения рекламных транспарантов, например, вывешиваемых на растяжках над улицей).

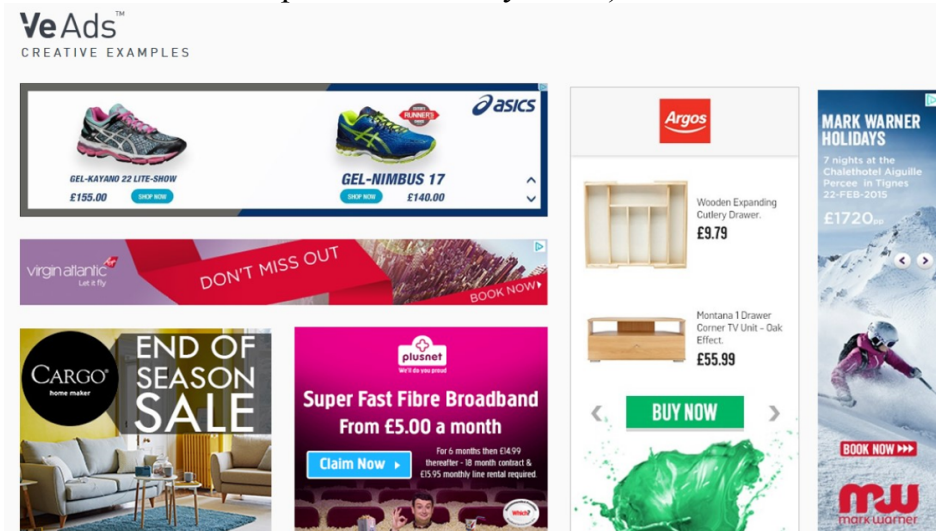


Рис. 6. Примеры баннеров.

<https://www.ve.com/ru/blog/bannernaya-reklama-v-internete> (28.02.2020.)

Прямое манипулирование:

Прямое манипулирование – непосредственный результат метафоры конкретного объекта. Если окна, буферы и другие элементы данных являются "физическими сущностями" то для работы с ними "нужны физические действия", а не команды.

Прямое манипулирование:

- выборка,
- открытие,
- буксировка,
- резиновая нить (изменение размеров окна),
- экранные кнопки.

Указание и выбор:

Пользователь может определить любое окно как активное подведением курсора и нажатием клавиши; пиктограмма может быть раскрыта до нужного окна. Для файла можно открыть

буфер. Если файл текстовый, может быть открыт буфер редактирования; если он исполняемый (программа), он может быть активизирован.

Указание также используется для позиционирования курсора.

Буксировка и резиновая нить:

Буксировка используется при выполнении операций перемещения и копирования. Выбранный объект "буксируется" к некоторой новой позиции. Так можно передвинуть окно/пиктограмму по экрану и нажать кнопку выбора. При нажатой кнопке указатель передвигается до нужной позиции. Файл можно скопировать, удалить ("отбуксировать") в "корзину".

Метод **резиновой нити** – специальный вариант буксировки, используемый, например, для изменения размеров окна. Движение курсора перпендикулярно ребру окна вызывает движение только этого ребра.

Экранные кнопки и скользящие барьеры:

Рассмотренные механизмы использовали кнопки манипулятора "мышь" или некоторые управляющие клавиши. Число кнопок можно увеличить, представляя их как объекты на дисплее. Такая **экранная кнопка** выбирается и "нажимается".

Если кнопки управляют объектом дискретно, то **скользящие барьеры** обеспечивают непрерывно изменяемый ввод/вывод. Пользователь буксирует барьер "мышью" вместо нажатия кнопок. Иногда при рассмотрении аппарата скользящих барьеров используются термины **ползунок** (показывает позицию окна над буфером) и **барьер просмотра** (перемещаемый для установки позиции).

Окна меню и блоки диалога:

Поскольку пространство физического окна ограничено, в **WIMP**-интерфейсах используются динамические окна.

Всплывающее меню – окно, которое появляется на экране с целью предъявления пользователю меню. Пользователь выбирает необходимый пункт, и после завершения выборки окно исчезает.

Постоянно доступные (и часто используемые) функции можно классифицировать по группам и заголовки групп представить в качестве **линейки меню**; **выпадающие меню** появляются автоматически при движении по линейке; откидное меню появляется только при активном выборе соответствующего элемента линейки.

Блок диалога / диалоговое окно:

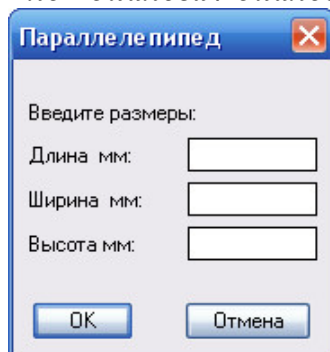


Рис. 7. Диалоговое окно DCL (AutoCAD)

https://acad-prog.ru/dialogovoe_okno/ (08.02.2020.)

Диалоговое окно (*Dialog box*)

прямоугольник на экране монитора персонального компьютера, служащий для ввода данных, необходимых для работы программы, подтверждения действия, ответа на запрос системы или настройки каких-либо параметров. Как правило, имеет кнопки ОК и «Отменить» (Cancel). Для облегчения работы пользователя многие диалоговые окна имеют в верхнем правом углу кнопку с изображением вопросительного знака. При щелчке «мышью» по этой кнопке, а затем по любому элементу диалогового окна появляется всплывающая подсказка с описанием этого элемента. Диалоговые окна часто состоят из нескольких вкладок. Так, для настройки мультимедиа-устройств в системе Windows имеются ярлычки с надписями «Аудио», «Видео», CD и др. в верхней части диалогового окна, предназначенные для доступа к различным вкладкам. В этом же диалоговом окне имеются элементы, предназначенные для регулирования «мышью», – ползунки для настройки уровня записи и воспроизведения. Если список файлов и папок каталога Windows не умещается в окне, появляется полоса прокрутки, позволяющая прокручивать информацию. Работа с диалоговым окном заканчивается подтверждением или отменой выполненных в нём действий. Для подтверждения открытия выбранного файла необходимо нажать кнопку «Открыть», а для отмены – кнопку «Отмена».

Энциклопедия «Техника». — М.: Росмэн. 2006.

http://dic.academic.ru/dic.nsf/enc_tech/368/%D0%B4%D0%B8%D0%B0%D0%BB%D0%BE%D0%B3%D0%BE%D0%B2%D0%BE%D0%B5 (08.02.2020.).

Оценка многооконных WIMP-интерфейсов

Оценка многооконной технологии

Многооконная технология обеспечивает готовый доступ к большему объему информации, чем при работе с одним экраном.

Система управления окнами допускает повышенную плотность информации на экране.

Окна дают доступ к многочисленным источникам информации; эту информацию можно сравнивать, объединять (в том числе разнородные данные).

Пользователь может использовать информацию на различных уровнях детализации.

Окна и пиктограммы позволяют эффективно использовать оперативную память человека и организовать/привлечь его внимание.

Частично перекрытые окна "напоминают о своем существовании"

Многооконная технология расширяет осознаваемое пространство обзора (много информации, которую можно увидеть), но не увеличивает объем информации, которую пользователь может усвоить.

Является ли метафора *рабочего стола* аналогией только с аккуратным рабочим столом?

Как соотносятся порядок/беспорядок на виртуальном *рабочем столе* пользователя и на рабочем столе в его кабинете/офисе?

Существует опасность: при использовании многооконного интерфейса большее количество времени тратится на работу с дисплеем, а не с решаемой задачей.

Преимущества и недостатки пиктограмм

Пиктограммы:

- полезное средство напоминания;
- могут (из-за небольшой площади) постоянно присутствовать на экране;
- сторонники пиктограмм считают, что они обеспечивают эффективную классификацию системных объектов с использованием зрительной памяти человека;
- оппоненты утверждают, что некоторые из используемых обычно пиктограмм двусмысленны, непонятны.

Преимущества и недостатки прямого манипулирования

Основное достоинство – нет необходимости изучать лексику, синтаксис и семантику языка взаимодействия с системой.

Интерфейсы прямого манипулирования позволяют достичь большого разнообразия операций с помощью небольшого набора механизмов (что удобно при разработке крупных проектов).

Хотя работа может осуществляться операциями указания и выборки, многим опытным пользователем более привычно и удобно использование традиционной структуры команд.

Некоторые дополнительные возможности интерфейсов

Динамические визуальные сигналы

Изменение изображения на экране с целью дать пользователю дополнительную информацию.

Так, при выполнении программой длительных действий курсор мыши приобретает форму песочных часов (или же такие часы появляются рядом с курсором). Это – сигнал о том, что на действия пользователя система временно реагировать не будет.

Второй пример – изменение изображения кнопки при нажатии на нее. Это – сигнал о том, что система «считает», что пользователь взаимодействует именно с этой кнопкой.

Анимационный интерфейс

Основан на том очевидном факте, что движущийся объект легче привлекает внимание.

"Как правило, в дизайне интерфейсов началу анимации сопутствует какое-либо событие. Это может быть событие, инициируемое пользователем или системой: щелчок мышью, наведение курсора, прокрутка страницы до определенного места, завершение загрузки элементов и т.д. Однако встречается и автоматическая анимация, чаще всего это бегущая строка или автоматическая прокрутка с периодическим интервалом, скажем, внутри блоков новостей или слайдеров. Отдельно можно рассматривать "контролируемую" анимацию, привязанную к прокрутке страницы. В онлайн-СМИ она часто задействуется в интерактивных материалах, предназначенных для "долгого" чтения.

В целом мы можем выделить три общих функции, которые анимация выполняет в графических интерфейсах:

Во-первых, анимация **облегчает восприятие** некоторых процессов и структур в интерфейсе.

Во-вторых, анимация **привлекает внимание**.

В-третьих, анимация может **показывать соответствие** между действиями пользователя (например, проведением пальцем по экрану или перетаскиванием объекта с помощью "мыши") и событиями в интерфейсе. Плавный скроллинг без преувеличения можно назвать значительно более удобным, чем, скажем, постраничное пролистывание, в первую очередь потому, что пользователь имеет возможность отслеживать интересующий его элемент в процессе прокрутки, не теряя его из виду. Также – в первую очередь на устройствах с сенсорным экраном – процессы, управляемые продолжающимися действиями пользователя, воспринимаются более естественно, будучи анимированными. Анимация отражает плавность движений управляющего интерфейсом человека и придает взаимодействию между человеком и машиной иллюзию прямого управления. Наконец, на более общем уровне восприятия анимация может повлиять на степень **«правдоподобности» интерфейса** – в живом мире человек не наблюдает резких изменений, и анимация точно так же, как и, например, текстуры или округлые линии, делает интерфейс более доступным благодаря углублению его сходства с объектами и процессами реального мира".

(Беляев А.А. Анимация в дизайне интерфейса информационных сайтов // электронный научный журнал "Медиаскоп", вып. 4, 2014 – <http://www.mediascope.ru/node/1618> – 15.02.2020.)

Звуковые эффекты

Простейшие звуковые эффекты, например, подача звукового сигнала при получении нового письма по электронной почте, обнаружении компьютерного вируса и т.п., обращают внимание пользователя на некоторое (иногда достаточно важное) событие в системе.

Они важны, например, в ситуациях, когда пользователь увлеченно работает с активными окнами, закрывающими другие окна или подсистемы, непосредственно связанные с событиями, требующими его внимания или срочного вмешательства.

Мультимедиа

Компьютерные технологии, обеспечивающие одновременное использование нескольких информационных сред: текст, графика, фото, видео, анимация, звуковые эффекты.

Мультимедиа — взаимодействие визуальных и аудиоэффектов под управлением интерак-тивного программного обеспечения с использованием современных технических и программных средств, они объединяют текст, звук, графику, фото, видео в одном цифровом представлении.

Например, в одном объекте-контейнере (англ. *container*) может содержаться текстовая, аудиальная, графическая и видео информация, а также, возможно, способ интерактивного взаимодействия с ней.

Термин *мультимедиа* также, зачастую, используется для обозначения носителей информации, позволяющих хранить значительные объемы данных и обеспечивать достаточно быстрый доступ к ним (первыми носителями такого типа были Компакт-диски).

Основные составляющие мультимедиа:



Текст



Аудио



Изображения



Анимация



Видео



Интерактивность

Словари и энциклопедии на Академике

<https://dic.academic.ru/dic.nsf/ruwiki/274779> (15.02.2020.)

Аппаратные средства мультимедиа:

— основные средства: компьютер с высокопроизводительным процессором и памятью большого объема, манипуляторами и мультимедиа-монитором со встроенными стереодинамиками;

— специальные средства: приводы CD-ROM, TV-тюнеры и фрейм-грабберы, графические ускорители, платы видеовоспроизведения, звуковые платы, акустические системы и др.

Словари и энциклопедии на Академике

https://dic.academic.ru/dic.nsf/fin_enc/20142 (15.02.2020.)

Поддержка пользователя:

- хорошие (и своевременные) *сообщения об ошибках*; не «назидательные»/грубые; с информацией о том, что нужно/можно сделать;
- хорошо структурированная и полная *справочная информация* и встроенные системные утилиты-«мастера» (подсистема помощи; мастер установки оборудования и др.);
- хорошо структурированная и полная (при этом – обозримая) *внешняя документация*.

Пример:

В книге [2, с. 197-200] приводится пример мучительного для пользователя диалога с системой электронной почты. Пользователь правильно ввел свой запрос только с 7 попытки.

При этом он ошибся 4 раза:

- простая ошибка ввода слова с клавиатуры (**haeders** вместо **headers**),
- второстепенная ошибка в наименовании (**intersection** вместо **intersect**),
- простая синтаксическая ошибка (пользователь не понял, что значения параметра следует заключать в кавычки),
- грубая синтаксическая ошибка (пользователь не понимает, что два ключевых параметра должны быть связаны логическим оператором).

Пользователь сразу же заметил первую ошибку, но при этом повторял ее несколько раз (когда терял сосредоточенность); он так и не заметил в сообщениях системы, что это слово можно было сократить до одной буквы (**h**).

Он сразу же исправил вторую и третью ошибки. Скорее из соображений здравого смысла, чем с помощью системы. Исправить четвертую ошибку без помощи системы пользователь, скорее всего, не смог бы.

Вывод: Причина ошибок – диалог ведется на основе командного языка & поддержка пользователя в системе явно неудачна (недостаток системы).

Ошибки

Последствия ошибок:

- а) система выходит из строя;
- б) система обнаруживает невозможность дальнейшей обработки полученных данных и просит пользователя скорректировать их;
- в) система выполняет задачу, но результат отличается от того, который ожидал пользователь (например, пользователь по ошибке выполнил команду **Move** вместо команды **Copy**).

Основные причины ошибочных действий системы – неверный ввод информации пользователем.

В случае а) проверка входных данных на корректность не предохраняет систему от невыполнимых команд.

В случае б) такая проверка показывает неосуществимость требований пользователя.

В случае в) такая проверка неполна (например, проверяется формат входных данных, а правильность самих данных не проверяется).

Основные причины ошибок пользователей:

- неправильное понимание сути требования (например, неправильное понимание подсказки);

- неправильное действие пользователя (в том числе случайная ошибка, например, нажатие соседней клавиши);
- недоработки авторов системы:

в одной из ОС были такие команды:

copy <ToFilename> = <FromFilename> - копирование файла

ren <FromFilename> <ToFilename> - переименование файла.

Процесс ввода данных в общем случае не должен быть сложным. Избежать ошибок ввода могут помочь: опережающий ввод, автозаполнение форм, максимально широкое использование **меню**; минимизация объема клавиатурного ввода, минимизация действий "по умолчанию", запрос и ожидание подтверждения.

Диалог не должен способствовать появлению ошибок восприятия или понимания.

Все системы нуждаются в механизме проверки входных данных для прекращения ошибочного ввода и информирования пользователя об этом (а, в лучшем случае, исправления ошибки ввода, если это возможно и допустимо с учетом полномочий системы – автоматическая замена **haeders** на **headers**).

Классический пример плохого сообщения об ошибках:

В одной из известных и распространенных в свое время крупных компьютерных систем при сбое центрального процессора в верхней части экрана каждого терминала появлялось сообщение:

ОЖИДАЮ ФУНКЦИЮ ЧЕТЫРЕ

Еще один пример плохого сообщения об ошибках:

ДИСК ЗАПОЛНЕН – НАЖМИТЕ КЛАВИШУ ESCape

Пользователь взволнован: пропадут ли 10 страниц текста, который он перед этим вводил? Можно ли (и как) сохранить результаты ввода?

В подобных ситуациях более удачно сообщение такого вида:

ДИСК ЗАПОЛНЕН – НЕВОЗМОЖНО СОХРАНИТЬ ТЕКУЩИЙ ФАЙЛ

НАЖМИТЕ КЛАВИШУ ESCape И ВСТАВЬТЕ НОВЫЙ ДИСК ДЛЯ СОХРАНЕНИЯ

ФАЙЛА

Сообщение об ошибке должно точно объяснить, в чем заключается ошибка и какие действия следует предпринять, чтобы ее устранить.

Введенные пользователем данные, которые расценены системой как некорректные, могут на самом деле быть правильными (появление незнакомого текстовому процессору слова). Можно пометить такие данные флажком "**новое**", а в дальнейшем уточнить ситуацию (например, с помощью пользователя).

К проблеме субъективного аспекта правильности данных мы вернемся в конце курса.

Справочная информация

В диалогах, управляемых системой, справочная информация вызывается с помощью стандартного ответа (**Help** и т.п.) на любой запрос системы. В диалогах, управляемых пользователем, может быть предусмотрена специальная команда (типа **Help**); возможно, с уточняющими параметрами.

Справочная информация (как и сообщения об ошибках) должна быть:

- информативна;
- своевременна (появляется на экране тогда и только тогда, когда она требуется);
- обзрима;

- сформулирована в понятных пользователю терминах;
- доступна в любой момент;
- отвечать контексту (соответствовать конкретной проблеме пользователя).

Целесообразно использовать иерархическую систему справочной информации.

Внешняя документация

Помощь пользователю преследует три главные цели:

- описание возможностей и принципов работы программной системы;
- снабжение пользователя инструкцией по решению конкретной задачи;
- помощь в работе с системой и выдача справочной информации.

Вопрос – всю ли справочную информацию следует "встраивать" в систему или же часть ее можно оформить в виде внешнего документа?

Учитывать: пользователь не может эффективно использовать объемное руководство во время работы с системой.

Проблема речевых ошибок при вводе в компьютер текста на естественном языке

Значительная часть информации, которую человек вводит в компьютерную систему представлена текстами на том или ином естественном языке (например, русском, английском).

Подробно о роли естественного языка в человеко-машинном общении, о связанных с этим вопросом мифах и реальных проблемах мы поговорим позже (Раздел 3). На сегодняшней лекции мы рассмотрим проблему ошибок при вводе в компьютер текста на естественном языке.

Использование естественного языка в качестве средства общения (*речевая деятельность* человека) неизбежно сопровождается теми или иными нарушениями языковых правил. Такие нарушения – вне зависимости от того, обусловлены они неполнотой знаний человека о языке или же явлениями подсознания или случайными сенсомоторными "сбоями" (описки, опечатки, оговорки) – мы будем называть *речевыми ошибками*.

Обнаружить речевую ошибку не всегда просто. Действительно, для получателя сообщения (реципиента) внешним признаком речевой ошибки служит появление в тексте какой-либо незнакомой ему речевой единицы. Однако такая "подозреваемая" речевая единица может оказаться и правильной конструкцией или формой (например, термином), не знакомой реципиенту.

С другой стороны, абсолютно правильная на первый взгляд единица может быть ошибкой, обнаружить которую удастся лишь на "высших" этапах анализа. Так, в предложении: *Пуск ракеты осуществляется нажатием краской кнопки* – все слова известны, синтаксические связи правильны; опечатка обнаруживается только на семантическом/смысловом уровне.

Если одним из участников общения является компьютерная система, положение становится еще более сложным. И лингвистические знания, и интеллектуальные способности (в том числе – в плане работы с языком) такого "собеседника" пока весьма скромны.

Отметим еще одно обстоятельство.

Как бы ни различались характер использования и назначение компьютерных систем, оснащение их средствами обнаружения и исправления речевых ошибок повышает устойчивость и эффективность их функционирования, облегчает процесс общения человека с компьютером.

Классификация речевых ошибок

Первый критерий классификации речевых ошибок, в соответствии с которым ошибки подразделяются на мотивированные и случайные, связан с понятием индивидуальной языковой модели.

Индивидуальная языковая модель (ИЯМ) – это то подмножество языковых единиц и правил, которое усвоил и использует в своей речевой практике конкретный носитель некоторого естественного языка. Субъективное преломление языка (как знаковой системы социального уровня) в процессе его усвоения приводит к тому, что в ИЯМ не попадают (или попадают в искаженном варианте) некоторые языковые единицы и правила языка.

Поэтому в речи конкретных носителей языка начинают проявляться некоторые индивидуальные особенности, либо вступающие в противоречие с языковыми нормами, либо нет.

В первом случае мы имеем дело с мотивированными речевыми ошибками – точнее, с ошибками, мотивированными особенностями ИЯМ конкретного носителя языка (пользователя компьютерной системы).

К ошибкам такого рода относятся, например, ошибки в словоизменении (*контейнерá* – в форме именительного падежа множественного числа), орфографические ошибки в основах (*еденица*), некоторые пунктуационные ошибки, смешение слов-паронимов (*представить* – *предоставить*), нарушение лексической сочетаемости (*делать горе*), искажение фразеологизмов (*не так страшен черт, как его малютки*).

Ошибки, обусловленные внешними по отношению к ИЯМ факторами: сбой речевого аппарата человека, несвоевременное переключение регистра клавиатуры, нажатие соседней клавиши, сбой на линии связи с компьютером – мы будем называть случайными.

Как правило, мотивированные речевые ошибки регулярно повторяются в речи носителя языка, а случайные ошибки могут как повторяться (например, при западании одной из клавиш), так и не повторяться. Отметим, что иногда отличить случайную ошибку от мотивированной сложно. Так, употребление слова *представить* вместо *предоставить* в контексте *представлено право* может быть или результатом случайной ошибки (пропуск буквы), или результатом мотивированной ошибки (смещения паронимов).

Мотивированные речевые ошибки могут различаться степенью серьезности (грамматичности). Помимо серьезных, абсолютно недопустимых грамматических ошибок – типа орфографических ошибок в основах или смешения слов – рассматриваются и ошибки, в результате которых появляются "полуграмматичные" формы (*контейнерá, сидевши*), которые имеют в словарях стилистические пометы: просторечное, устарелое, разговорное, областное и др.

Следующий критерий классификации ошибок (мотивированных и случайных) связан с языковыми уровнями, нормы (правила) которых оказываются нарушенными в результате речевых ошибок. В соответствии с этим критерием можно выделить:

1) орфографические ошибки: пропуск одной буквы, замена одной буквы, перестановка двух рядом стоящих букв, одна лишняя буква (отдельно может рассматриваться случай удвоения буквы), замена буквы русского алфавита буквой латиницы и др.;

2) морфологические (словоизменительный уровень) ошибки: ошибки в окончаниях (флексиях) при склонении и спряжении слов (рассматриваются различные подклассы таких ошибок), употребление отсутствующих в языке форм слов, несоблюдение правил чередования в основе, употребление незнакомых системе вариантов слов, испытывающих колебания в роде, одушевленности;

3) синтаксические ошибки: ошибки в моделях управления слов-предикатов, пунктуационные ошибки, нарушение обычного порядка слов (в том числе – в устойчивых словосочетаниях), вставка пробела внутрь слова, пропуск пробела (отдельно могут рассматриваться случаи слитного и раздельного написания частиц *не* и *ни*);

4) лексико-семантические ошибки: употребление слов в необычном значении, нарушение лексической сочетаемости, семантические противоречия.

Ошибки при речевом вводе

При речевом вводе с помощью микрофона возникают дополнительные ошибки, связанные как с особенностями устной речи пользователей, так и с использованием специального программно-аппаратного обеспечения распознавания речи.

Известно, что в устной речи встречается так называемая **омофония** – конструкции/ слова произносятся одинаково, а пишутся по-разному. Известный пример омофонии – *и скота* ↔ *из kota*. Еще один пример (найденный в свое время лектором): фразы *I have red books* и *I have read books* произносятся одинаково, обе они допустимы в речи. Однако, система распознавания, которая должна по произносимому построить письменный текст, столкнется с неоднозначностью. Очевидная причина – в английском языке три формы неправильного глагола *to read* пишутся одинаково, но начальная форма (Base) произносится так [ri:d], а формы Past Simple и Past Participle – [red] (как и прилагательное *red*).

Нечеткая дикция, невнятная речь, изменение скорости речи, орфографические ошибки (*проишествие* вместо *происшествие*), ошибки в ударении (*зво'нит* вместо *звони't*), интонационные ошибки, появление лишних или отсутствие ожидаемых пауз, слов-паразитов (*э-э-э, да*) также приводят к неверному распознаванию произнесенного.

(Источник части примеров: "14 русских слов, в которых каждый хоть раз делал ошибку" – сайт Flytothesky.ru, Автор: Редакция Фев 21, 2014

(<https://fishki.net/1437326-14-russskih-slov-v-kotoryh-kazhdyj-hot-raz-delal-oshibku.html>) (02.02.20..)

В итоге, построенный системой распознавания и переданный приложениям текст, может отличаться от того, что хотел сказать пользователь.

Надежность распознавания, особенно в ситуации без предварительной настройки на диктора (пользователя системы с речевым вводом) пока не очень высока.

Ошибки при вводе печатного и рукописного текста

Ввод печатного или рукописного текста также предполагает использование средств распознавания, разумеется не последовательности звуков, а цепочки символов.

Печатный текст (без дефекта бумаги и/или шрифта) распознается достаточно надежно (точность близка к 100%). При работе с рукописным текстом сложности вызваны нечеткостью почерка: слишком маленький пробел между словами может привести к ошибке типа "склеивание соседних слов", в русскоязычных текстах рукописное *т* можно спутать с рукописным *и*) и др.

Замечание об опыте лектора в решении задач распознавания речи и текста и о динамике цитирования его работ в этой области.

Диагностика речевых ошибок

Методы обнаружения и исправления орфографических и морфологических ошибок в текстах широкой тематики базируются на представлении о тексте как о цепочке независимо появляющихся словоформ. Известно три основных метода обнаружения орфографических ошибок – статистический, полиграммный и словарный.

При статистическом методе словоформы, обнаруживаемые в тексте, упорядочиваются согласно частоте их встречаемости. Искаженные слова оказываются среди малоупотребительных слов в конце списка.

При полиграммном методе все встречающиеся в тексте двух- или трёхбуквенные сочетания (полиграммы) проверяются по таблицам, содержащим информацию об их допустимости в русском языке. Если в словоформе имеются недопустимые полиграммы, то она считается неправильной.

При словарном методе все входящие в текст словоформы проверяются по компьютерному словарю. Если словарь такую форму допускает, она считается правильной, а иначе либо сразу признаётся ошибочной, либо предъявляется человеку.

В настоящее время первые два метода практически не используются, т.к. уже есть хорошие компьютерные словари, достаточно большие по объёму и с эффективным доступом.

Диагностика же и исправление синтаксических, пунктуационных и лексико-семантических ошибок предполагает взгляд на текст как на последовательность связанных единиц, комбинирование которых имеет свои закономерности.

Подходы к автоматизации выявления и коррекции этих ошибок можно разбить на две группы: синтаксически-ориентированные подходы и подходы, основанные на концептуальных фреймах.

Последние больше пригодны для систем, работающих в строго ограниченных предметных областях. Для текстов широкой тематики предназначены синтаксически ориентированные подходы. Сначала поступившее на вход предложение обрабатывается средствами грамматики, рассчитанной на синтаксически правильный текст. Если такая проверка обнаруживает дефекты

синтаксической структуры, некоторые условия ослабляются. Какие грамматические правила смягчаются, зависит от учитываемых системой ошибок. Например, в русских текстах иногда оказывается пропущенной запятая, обособляющая причастный оборот в постпозиции. Для того, чтобы такое предложение могло быть обработано, требуется временная отмена условия (присутствующего в каноническом правиле) обязательного наличия запятой.

Однако ослабление канонических правил неизбежно влечёт за собой возрастание числа возможных интерпретаций. При этом нельзя опознать ошибочный текст прежде, чем будет закончен анализ средствами канонической грамматики. Другой подход предлагает сначала использовать слабую грамматику, а затем подвергнуть обрабатываемое предложение фильтрации на основе строгих требований правильности. Но при этом наличие ошибки предполагается более вероятным, чем соблюдение норм грамматики.

Также отметим, что описанные методы позволяют автоматически обнаружить ошибку только тогда, когда не удаётся построить связный синтаксический граф для рассматриваемого предложения. Однако ошибки, при которых возможно получение формально приемлемой, но по сути неверной интерпретации, остаются невыявленными. При этом никаких сообщений об ошибках не поступает.

Процессы и устройства ввода-вывода

Процессы ввода-вывода служат для того, чтобы принять от пользователя данные и передать ему данные через различные физические устройства.

Основные процессы ввода-вывода [2, с 55]

Ввод текстового сообщения:

- с использованием стандартных процедур;
- в режиме посимвольного ввода;
- с использованием *специальных символов*.

Ввод предложений типа указать и выбрать:

- просмотр предлагаемого списка операций и выбор нужной;
- выбор данных из любого места экрана.

Ввод графического сообщения.

Вывод текстового сообщения:

- в текущую позицию на устройстве;
- в заданную позицию на устройстве;
- с указанием конкретного формата изображения.

Вывод графического сообщения.

Текстовое сообщение – строка символов (в том числе, символов псевдографики).

Графическое сообщение – сообщение, которое нельзя представить в виде строки символов.

Ввод текстового сообщения.

Стандартный ввод с клавиатуры.

Цикл: - **ожидание** готовности устройств ввода;
- **интерпретация управляющих** последовательностей для редактирования (**Backspace, Delete**) или позиционирования курсора на экране;
- **эхо-печать** на экране введенной информации (до ввода **Enter**).

Посимвольный ввод.

Ввод с использованием специальных клавиш.

Помимо обычных клавиш клавиатура имеет и **специальные** (редактирования, управления курсором, функциональные, например, **Backspace, F1**).

Иногда для упрощения работы со специальными клавишами поступают так: каждой клавише клавиатуры присваивается числовой код (код сканирования); даже если при нажатии клавиши не вырабатывается обычный код символа (клавиши: **Shift, Control** и др.), эти клавиши при нажатии одновременно с обычными символьными служат для установки **флага** состояния клавиатуры.

Так, с помощью драйвера клавиатуры мы можем в зависимости от флага сформировать код символа и ввести **с, С** или **Control-С**.

Ввод предложений типа указать и выбрать:

Относительный выбор – просмотр предлагаемого списка объектов (в том числе, операций) и выбор нужного объекта. Нужно определить: как предьявляется список объектов, как происходит перемещение по этому списку; как пользователь фиксирует выбранный объект.

Используется клавиатура и/или **мышь**.

Пользователю нужно ясное и оперативное подтверждение того, на какой объект он указывает.

Абсолютный выбор – дает возможность пользователю указать любое место на экране (есть там объект или нет). Предполагается, что соответствующий процесс ввода может получить от устройства ввода точно указываемые координаты. В основном используется при вводе данных с помощью манипулятора типа **мышь** или **сенсорного экрана**.

Вывод текстового сообщения в текущую позицию на устройстве (экран).

Что должно быть выведено (строка символов).

Как текст должен быть выведен (атрибуты, определяющие формат выводимых данных).

Вывод текстового сообщения в заданную позицию на устройстве.

В текстовом режиме экран условно разделяется на **знакоместа**. Каждому знакоместу могут быть сопоставлены цвет шрифта и цвет фона (или оттенок одного цвета).

Текущая (следующая доступная) позиция указывается на экране **курсором**.

- 1) Перевод курсора в начало поля, задаваемое координатами (**строка, столбец**).
- 2) Вывод сообщения в заданную позицию.

Вывод текстового сообщения с указанием конкретного формата изображения.

Выделения в сообщении:

- шрифт;
- шрифтовые выделения (курсив, полужирный, подчеркивание);
- размер;

цвет;
мерцание;
яркость;
фон.

Важное сообщение может сопровождаться звуковым сигналом.

Ввод и вывод графических сообщений

В графическом режиме экран монитора условно разделяется на более мелкие элементы чем знакоместа – *пиксели*. На монохромном дисплее пикселю соответствует один бит. На цветных экранах число битов, необходимых для представления каждого пикселя, соответствует числу основных цветов. Количество точек по горизонтали и по вертикали называется разрешающей способностью экрана (чем она выше, тем выше качество изображения).

Устройства ввода:

клавиатура – текстовый ввод, экспорт нетекстовых объектов;

виртуальная (экранная) клавиатура;

дигитайзер / графический планшет (состоит из собственно планшета/экрана и пера/стилуса) –

устройство для преобразования изображений/рукописного текста в цифровую форму;

сканер – ввод графических изображений и текстов обычно с бумажных носителей (для преобразования полученного при сканировании текста из графического изображения в текстовое нужны специальные прикладные программы – системы распознавания текста);

устройства считывания документов фиксированного формата;

датчики (для автоматического сбора и передачи в компьютер информации);

световое перо;

сенсорный экран;

манипулятор *мышь* (механическая, оптико-механическая, оптическая, беспроводная);

трекбол (перемещается не корпус, а шар; может располагаться на клавиатуре);

тачпад (указательное сенсорное устройство; также может располагаться на клавиатуре компьютера, а кроме того, на панелях бытовых электронных устройств);

микрофон – устройство для речевого ввода;

веб-камера (вебкамера) – видео- или фотокамера, переводящая оптическое изображение в видеосигнал или поток видеоданных, с целью ввода изображения в компьютер или непосредственной передачи по сетям;

цифровые фото- и видеокамеры;

джойстик (в англ. – "ручка управления самолетом") – устройство ввода в виде ручки управления, обычно с управляющими кнопками);

игровые манипуляторы/геймпады;

игровые рули;

средства, поддерживающие невербальную коммуникацию (в том числе "умные" очки, напульсники, специальные датчики): мимика, жесты, позы; физиологические показатели, например, проводимость кожи, расширение зрачков и частота сердечных сокращений; бесконтактные сенсорные игровые контроллеры.

Устройства вывода:

монитор;

проектор;

(эти два устройства служат для вывода оперативной текстовой и графической информация);

принтер (матричный, струйный, лазерный);

графопостроитель / плоттер (устройство для автоматического вычерчивания с большой точностью рисунков, схем, сложных чертежей, карт и другой графической информации на бумаге размером до А0 или кальке);

газоразрядный индикатор (ионный прибор для отображения информации, использующий тлеющий разряд);

каттер / режущий плоттер (совмещает процесс построения изображения с разрезанием листа на фрагменты нужного формата);

3D-принтер;

синтезатор речи – программно-аппаратные (как правило) средства перевода текста или других данных в звучащую речь;

звукогенератор (не путать с видом оружия);

акустические системы (наушники, колонки и др.) – средства звукового вывода.

Устройства ввода-вывода:

интерактивная доска;

дисковод;

картридер / кардридер;

USB-флеш-накопитель;

стример;

шлем, очки, перчатки, комнаты виртуальной реальности,

средства *имитации тактильных ощущений*, [виртуальный ретинальный монитор](#)

(Виртуальная реальность имитирует как воздействие, так и реакции на воздействие.);

средства *прямого подключения к нервной системе* ;

гаптоклон (новое – [см. ниже](#)).

Устройства связи:

сетевая карта; модем; роутер/маршрутизатор; линии связи (кабельные, проводные, беспроводные).

модем – для подключения одного компьютера к сети;

роутер – для распределения сетевого трафика между несколькими компьютерами;

беспроводные линии связи:

спутниковые каналы передачи данных;

радиорелейные каналы передачи данных;

сотовые каналы передачи данных;

спутниковые каналы передачи данных;

радиоканалы передачи данных (WiMAX, MMDS, для локальных сетей [Wi-Fi], Bluetooth).

Технология Ethernet –сетевой кабель, который подключается к компьютеру (сетевая карта).

Технология ADSL (Asymmetric Digital Subscriber Line).

Что следует учитывать при выборе устройств?

- Содержание и формат обрабатываемых данных.
- Объем ввода-вывода.
- Ограничения, накладываемые пользователем и рабочей средой.
- Ограничения, связанные с другими аппаратными и программными средствами, используемыми в системе.

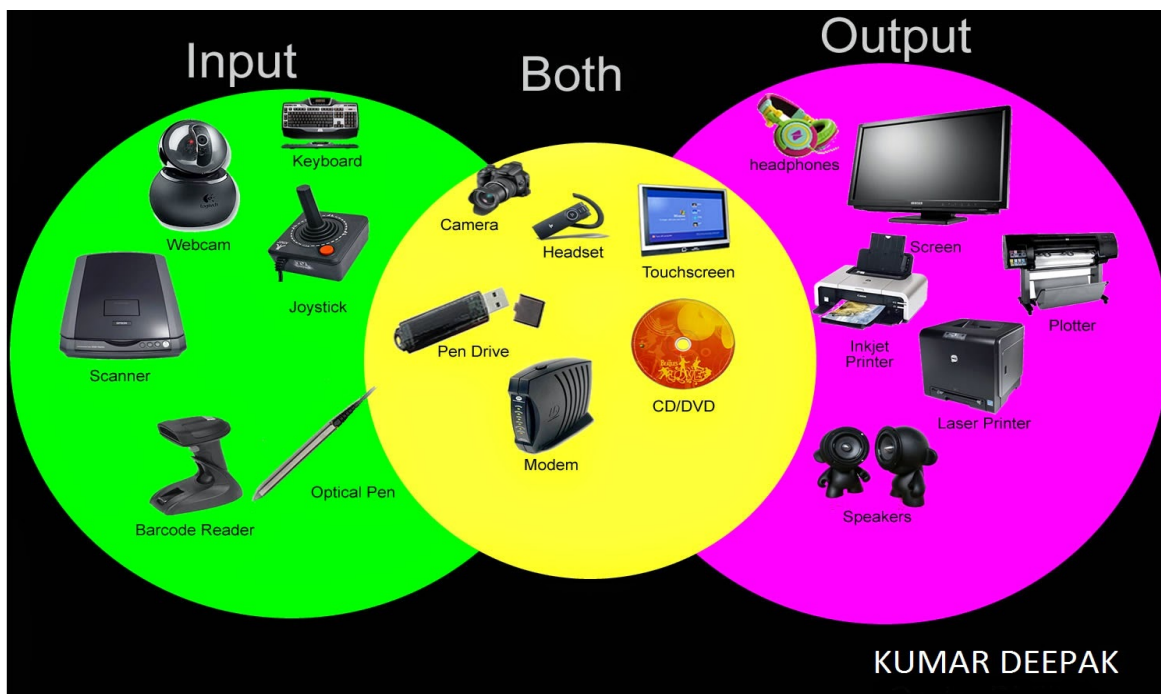


Рис. 1. Некоторые устройства ввод-вывода.

<http://zerombaccess.blogspot.ru/2015/02/input-output-devices.html> (02.02.2020.)

Гаптоклон

"Японские учёные сделали возможным тактильный контакт на расстоянии"

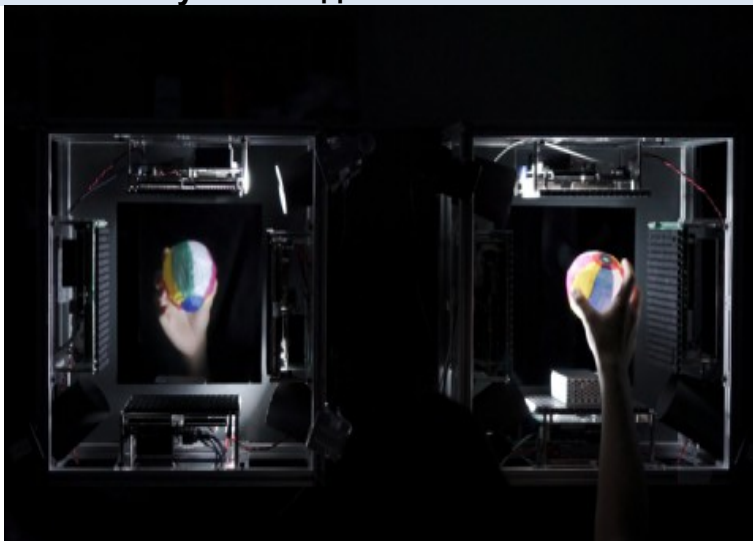


Рис. 2 Иллюстрации к статье про гаптоклон (источник указан ниже).

Проект, получивший название Haptoclone, предлагает небывалый опыт - так называемую "телетактильность". Это значит, что пользователь сможет отправлять

своё голографическое изображение, имитирующее прикосновение, другому пользователю, использующему такой же аппарат. Это позволит полноценно ощутить иллюзию прямого тактильного контакта.

Идея была впервые представлена на конференции по вопросам компьютерной графики SIGGRAPH 2015.

Продукт пока что недоступен для покупки, однако, вкупе с другими VR- и AR-проектами, Haptoclone может стать настоящим прорывом в будущем и предложить пользователям ещё более яркий цифровой опыт. Ведущий разработчик Haptoclone Ясутоши Макино заявил:

Представьте, насколько это здорово, - предложить людям, находящимся в разных городах и странах ощутить прикосновение друг друга. Или представьте, что находитесь в зоопарке и хотите погладить льва. Наш проект позволит безо всякого риска почувствовать, что это такое.

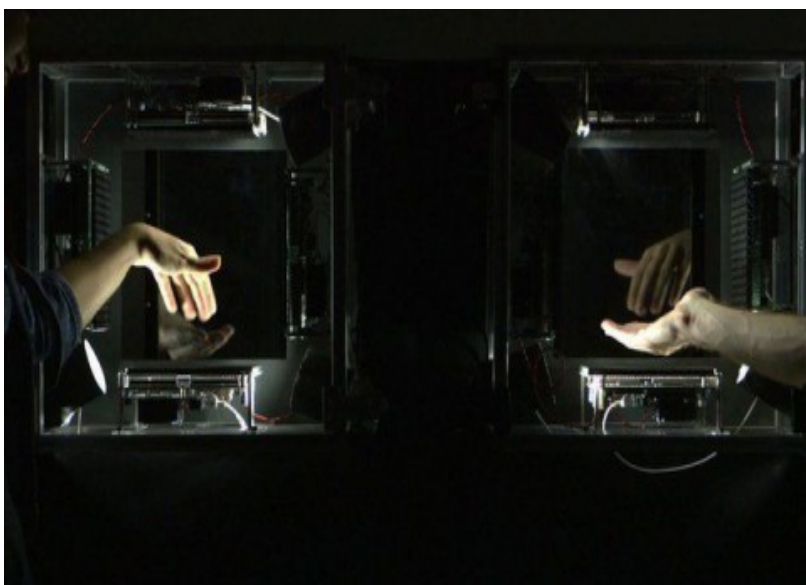


Рис. 3 Иллюстрации к статье про гаптоклон (источник указан ниже).

Его коллега Ёшиказу Фуруяма добавил:

Изображение настолько реалистичное, что, в корреляции с ощущением настоящего прикосновения, будет непросто отличить голограмму от реальности.

Haptoclone использует технологию Kinect для захвата движений в реальном времени и четыре ультразвуковые матрицы, создающие сверхзвуковое лучевое давление. Таким образом компьютер заставляет пользователя поверить в абсолютную реальность цифровых прикосновений.

Одним из немногочисленных минусов инновационной идеи является повышенная сложность создания агрегата, вызванная, по большей части, высокой стоимостью сборки. В данный момент исследователи используют особые воздушные панели для создания изображения. В будущем они планируют заменить их обычными [камерами](#), таким образом сделав тактильно-оптическую коммуникацию на расстоянии более доступной для широкой аудитории.

Также стоит принимать во внимание вопрос безопасности подобной технологии. Хироюки Шинода, исследователь из Токийского университета, также занятый в разработке проекта, отметил, что использующийся в машине уровень ультразвука вполне приемлем и безопасен, однако малейшая ошибка в расчётах может нанести вред здоровью. Именно поэтому сейчас учёные заняты вопросом о "необходимых ограничениях".

Источник: bgr.com

<http://odnako.su/hi-tech/smart-mobile-phone/-452047-yaponskie-uchyonye-sdelali-vozmozhnym-taktilnyj-kontakt-na-rasstoyanii/> (02.02.2020.)

Разработка интерфейса "человек-компьютер"

Общие рекомендации:

- Интерфейс необходимо проектировать отдельно (как, например, отдельно можно разрабатывать структуру файлов). Состав и форма представления входных и выходных данных должны стать предметом тщательного анализа разработчиков интерфейса.
- При проектировании интерфейса необходимо максимально полно учитывать особенности используемых аппаратных и программных средств.
- Крайне желательно соблюдение стандартов (либо общих, если таковые есть, либо принятых в некотором коллективе разработчиков).
- Необходимо соблюдение общепринятых в эргономике рекомендаций (разумеется, с учетом конкретных особенностей разработки).
- Необходим учет особенностей, целей, проблем потенциальных пользователей. Необходимо привлечение пользователей к оценке (и необходимой модернизации) разрабатываемого интерфейса.
- Интерфейс должен быть адаптируемым к изменению круга пользователей, к изменению потребностей пользователей.

Критерии оценки качества интерфейса:

- простота освоения и запоминания операций системы (*сколько времени требуется определенному пользователю для достижения заданного уровня знаний; какова должна быть его подготовка; насколько легко возобновить деятельность после некоторого перерыва в работе и т.п.*);
- быстрота достижения целей задачи, решаемой с помощью системы (*учитывать не быстрое действие системы, а время, необходимое для достижения некоторой цели, например, «обработка за час не менее 20 банковских счетов с ошибкой менее 1 %»*);
- субъективная удовлетворенность при эксплуатации системы (*явное мнение пользователя или же количество невыходов на работу*).

Общие требования к диалогу:

- **Естествен** (для человека, решающего привычные задачи), т.е. не вынуждает пользователя существенно изменять свои традиционные способы решения задачи; стиль ведения диалога должен быть разговорным, а не письменным; следует избегать как чрезмерной напыщенности, так и «фамильярности»; диалог должен вестись на родном языке пользователя (или на другом понятном и привычном ему языке).
- **Последователен** (логически); последовательность (единообразие) в организации диалога с различными модулями системы, в построении сообщений, в использовании форматов данных, в размещении данных (однотипных) на экране.
- **Неизбыточен**; ввод минимума информации, необходимой для работы системы; выходные сообщения должны содержать именно ту информацию, которая

требуется пользователю, причем в приемлемой для восприятия форме; следует широко использовать принцип «по умолчанию».

- **Обеспечивает поддержку пользователя;** поддержка пользователя – мера помощи, которую система оказывает пользователю при его работе; основные аспекты: количество и качество имеющихся инструкций, характер выдаваемых сообщений об ошибках, подтверждение тех или иных действий системы.
- **Гибок;** гибкость диалога – мера того, насколько хорошо он соответствует различным уровням подготовки и производительности труда пользователя.

Рассмотренные нами критерии эффективного взаимодействия человека с компьютером касаются не только структур диалога и содержания отображаемых сообщений системы, но и того, как эти сообщения выглядят на экране.

Процесс размещения данных на экране – Форматирование экрана

Прежде всего разработчику интерфейса следует определить:

- **какая** информация должна появляться на экране (с учетом избыточности диалога);
- каков основной **формат** этой информации;
- **где** эта информация должна появляться (область вывода для каждого поля; пустые места);
- какие средства используются для **выделения** (цвет изображения и фона, инверсное изображение, мерцание текста или фона, яркость и др.).

После этого можно:

- **разработать проект** размещения данных на экране;
- **оценить эффективность** этого размещения.

Рекомендуется оценка эффективности предложений во взаимодействии с пользователями / потенциальными пользователями.

Этот процесс часто носит итеративный характер.

Данные должны располагаться на экране так, чтобы пользователь мог просматривать экран в логической последовательности и мог легко:

- выводить нужную информацию;
- идентифицировать связанные группы данных;
- легко определять исключительные ситуации (сообщения об ошибках, предупреждения);
- определять, какое действие он должен выполнить (и должен ли вообще что-то делать) для продолжения выполнения задания.

Какая информация должна выводиться на экран?

Желательно – на экране находится та и только та информация, которая действительно необходима пользователю на данном этапе работы. Например, в меню не следует включать пункты, соответствующие недоступным в данный момент действиям, или же пункты, с которыми работает узкий круг пользователей.

Однако есть и общие правила/рекомендации: оставлять примерно половину площади экрана пустой; оставлять пустую строку после 4-5 строк таблицы (или выделять строки, например, поочередно, фоном); оставлять несколько пробелов между столбцами таблицы.

Логически связанные данные должны быть представлены отдельной группой.

Фрагменты текста должны располагаться на экране так, чтобы взгляд пользователя сам перемещался по экрану в нужном направлении.

Общепринятую схему просмотра (начать из левого верхнего угла и перемещаться слева направо и сверху вниз) желательно менять только в крайних случаях (расположение материала на экране задает другую схему).

"Нельзя допустить, чтобы пользователь при заполнении экранной формы мучился так же, как разработчик при ее составлении". [2, с. 246].

В каком виде информация должна выводиться на экран?

Естественность диалога предполагает, что информация представлена таким образом, что ее можно сразу же использовать. Недопустимо требовать, чтобы пользователь обращался к справочникам или выполнял какие-то промежуточные вычисления.

Общепринятая система больших (прописных) и малых (строчных) букв в тексте облегчает его восприятие. "Программу для вывода формы, представленной на рис. 7.2, наверняка составил программист с устаревшими взглядами, который даже мыслит заглавными буквами!" [2, с. 243].

В таблицах поля и данные (и их наименования) должны четко различаться.

В каком месте экрана должна располагаться информация?

Плотность расположения данных (это понятие субъективное) зависит и от задачи, и от категории пользователя.

Данные на экране для опытного оператора/пользователя могут располагаться плотнее, чем для начинающего.

Выделение информации на экране

Выделение информации – использование средств, позволяющих привлечь внимание пользователя к некоторой области экрана (и, следовательно, к представленной в этой области информации).

Неудачное выделение информации может помешать работе пользователя. Поэтому при выборе способов/средств выделения информации следует учитывать психофизиологические особенности человеческого восприятия (а при проектировании интерфейса разработчику рекомендуется сначала расположить информацию без выделения полей и только после этого решить какие поля следует выделить и как).

Рекомендации по использованию цвета:

- минимальное количество цветов (не более 3-4 на одном экране)
- для больших прямоугольников следует выбирать цвет фона;
- яркие цвета для выделения данных, а более спокойные тона – для фона;
- для выделения двух областей уместны контрастные цвета (из разных концов спектра);
- цвет должен соответствовать представлениям (ассоциациям) пользователя;
- целесообразны эксперименты с различными композициями цветов на реальном экране.

Время ответа (интервал между событием и реакцией системы на него):

Время ответа системы – время от момента ввода последнего символа команды/запроса до момента вывода первого символа ответа системы.

Быстрый ответ системы:

- не задерживает пользователя,
- впечатляет, создает благоприятное представление о системе.

Психологические аспекты скорости ответа системы на запрос:

- преемственный и предсказуемый отклик системы;
- ответ системы не должен быть слишком быстрым, иногда необходимы паузы;
- гипотеза Ингве (7 ± 2) об объеме оперативной памяти человека;
- ограниченность времени хранения информации в оперативной памяти (около 2 секунд).

Рекомендации по допустимому времени ответа:

Подтверждение факта	- 0.1 – 0.2 с
Выполнение простой команды	- 0.5 – 1.0 с
Выполнение команды связного диалога	- 1.0 – 2.0 с
Обработка сложного запроса	- 2.0 – 4.0 с
Пакетная обработка	- 10 – 20 с

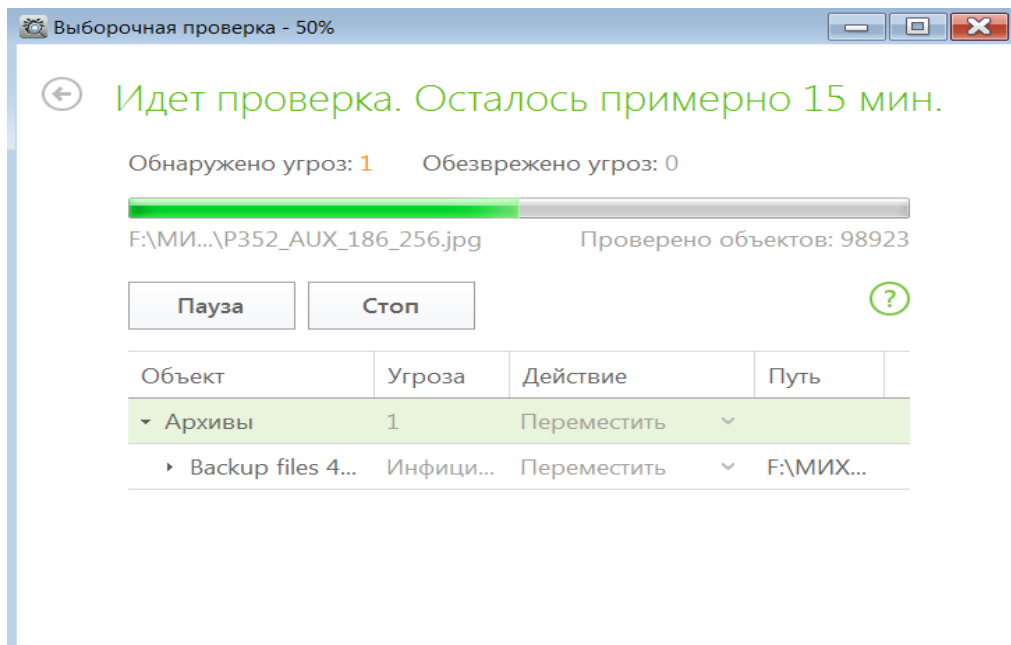
Иногда считается, что если время ответа более 20 секунд, то система не является интерактивной.

Если все же процесс обработки длителен по сути (например, при поиске вирусов на жестком диске большого объема), необходимо информировать пользователя (постоянно или время от времени) о нормальном протекании процесса.

Для этого используются, в частности:

- подвижная шкала, наглядно показывающая объем проделанной и объем оставшейся работы;
- оценка времени объема оставшейся работы;
- информация о проценте объема проделанной работы;
- комбинации этих приемов (например, шкала и оценка времени оставшейся работы).

(скриншот подготовлен лектором)



Почему трудно добиться быстрой реакции системы?

Время ответа зависит от загрузки ресурсов системы:

- времени доступа к файлам,
- времени работы ЦП,
- времени передачи информации по сетям.

При возрастании нагрузки на систему:

- возрастает загрузка ресурсов,
- образуются очереди,
- возрастает время ответа (резко возрастает при использовании системы на 80% и выше).

Реакция на загрузку зависит от:

- среднего числа задач,
- среднего «размера» задач,
- разброса размера задач относительно среднего.

Необходимость гарантии того, что система работает (сообщения о состоянии).

Различные варианты таких сообщений.

Адаптация

Фиксированная адаптация – пользователь явно выбирает уровень диалоговой поддержки. Первое решение – выделение двух уровней:

- **подробный диалог**, обеспечивающий всемерную поддержку начинающих;
- **краткий диалог**, предназначенный для экспертов и обеспечивающий небольшую поддержку либо совсем лишенный ее.

Недостатки: навыки меняются со временем; пользователь может хорошо знать одну часть системы и совсем не знать другие; выбор уровня (пользователем) может быть неверен; автоматическое определение уровня затруднительно.

Косметическая адаптация – приспособление компьютерной системы к развитию у пользователя навыков работы с ней без учета поведения пользователя и без однозначного выбора им конкретного стиля диалога:

- умолчания,
- сокращения,
- синонимы,
- "горячие клавиши",
- опережающий ввод,
- многоуровневая помощь.

Косметическая адаптация (названа так потому, что вносит лишь поверхностные изменения в базовую структуру) весьма полезна, поскольку снижает утомляемость и делает интерфейс более универсальным.

Гибкость при сравнении (входного сообщения и "цели").

1. Полное совпадение

2. Совпадение сокращений:

Слово	Правильные сокращения
мотор	мо мот мото
содержимое	с со сод содерж содержим
собственность	с со соб собств собствен
конец	к ко кон

Ситуации: нет совпадений, однозначное совпадение с целью, неоднозначное совпадение.

3. Частичное совпадение:

Простейший вариант. Удаление по одному символу с конца "цели" до совпадения с входными данными (так, можно работать с опечаткой типа – **prpo** вместо **prop**).

Более серьезные методы предполагают использование метрик, статистики ошибок, учета их причин (в том числе, расположения клавиш на клавиатуре, регистра и др.).

Синонимы

Пример: в MS-DOS синонимичны **DEL**, **ERASE** и **del**.

Преимущество синонимов – пользователь может выбирать наиболее естественный/привычный для него вариант написания идентификатора/команды.

Умолчания

Система заранее настраивается на стандартный (наиболее вероятный) ответ/сообщение пользователя (например, дата – обычно текущая дата; номер следующего документа – обычно следующее натуральное число, принтер – обычно основной принтер, подключенный к компьютеру). Для принятия соответствующего значения: "нулевой ввод" (**Enter**), нажатие "активной" экранной кнопки.

Проблемы: неэффективно, если вводятся разные данные; стандартный ответ может меняться при смене пользователя и с течением времени.

Опережающий ввод

Опережающий ввод символов.

Если при вводе первых символов сообщения система обнаруживает, что сообщение целиком хранится в некотором буфере (файле, адресной книге и т.п.), система дополняет введенную цепочку и выдает один или несколько вариантов полного сообщения. Это сообщение хранится в системе и было когда-то введено пользователем (или другими пользователями) ранее.

Можно использовать и механизмы, обеспечивающие опережающий ввод цепочки ответов/команд.

Многоуровневая помощь

Система циклически выводит сообщения и запрос о необходимости более детального объяснения (до тех пор, пока пользователь не будет удовлетворен либо не будут исчерпаны все уровни помощи).

Полная адаптация – формирование в компьютерной системе модели пользователя (или группы пользователей) и модификация этой модели в процессе работы.

Изыщная адаптация и модели пользователя

Любые интерфейсы предполагают наличие (обычно неявной, "*вшитой*" в них) модели мира, лежащей в основе их взаимодействий с пользователем. Их мир – некоторый класс прикладных задач. Каждый пользователь имеет свою собственную модель этого мира приложений, используемую им при общении с компьютерной системой (формирование задания и входных данных, интерпретация сообщений системы).

Требование *естественности диалога* – требование (как минимум) сопоставимости этих моделей.

Вспомним об индивидуальных языковых моделях и проблемах человеческого общения на естественном языке.

Отличия в моделях приводят к ошибкам "взаимопонимания" и затрудняют общение пользователя с системой.

Любой интерфейс содержит неявную модель мира приложений, "внесенную" в него разработчиками и отражающую их взгляд на эту модель.

"Большинство из более или менее удачных интерфейсов человек-компьютер потерпели провал из-за того, что модель на которой проектировщик основывал систему, существенно отличается от модели, сложившейся у пользователя". [2, с. 401].

Пример с выходом из системы с помощью **F1**.

Проблемы:

пользователей может быть несколько;

даже в однородной группе пользователей могут существовать индивидуальные различия в моделях;

даже единственный пользователь со временем меняется.

"Почему пользователь должен подстраиваться под системную модель, а не системная адаптируется либо сама, либо кем-то, подстраиваясь к текущей модели пользователя". [2, с. 402].

Путь – отделение содержания и представления системных сообщений и правильных ответов от структуры диалога; в дальнейшем, персонификация диалога.

Добавление ускорителей (опережающий ответ, гибкость при сопоставлении) обеспечивает большую адаптивность. Желательно, чтобы и пользователь мог сам модифицировать содержание диалога (например, перефразировать подсказки).

Мы говорили ранее о различных уровнях подготовки пользователей (групп пользователей) в связи с выбором схемы диалога.

Но как определить автоматически уровень подготовленности пользователя?

Интерфейсу/его разработчику необходимо иметь не только модель задач приложений, но и модели пользователей (*профили пользователей*).

Сложности создания таких профилей.

Учет психофизиологических факторов (характер, способность к обучению, знание задачи, знание интерфейса). Часть из них (личностные) относительно устойчивы, другие – динамичны.

Что можно учесть? Время отклика пользователя; характер ошибок ввода, степень, в которой используется опережающий ответ; историю работы с системой.

Изящная адаптация (вероятно, *smart*) – термин, введенный для описания такого типа интерфейса, который обеспечивает разумную меру самоадаптации:

гибкость с точки зрения возможных входных сообщений и выходной информации;

способность к персонализации;

не прерывает пользователя, но информирует о неоднозначных и нераспознанных входных данных; интерфейс должен "следить" за вниманием пользователя;

в идеале желательны "совместные" действия интерфейса и пользователя, аналогичные тем, что возникают при общении людей.

Изящное взаимодействие – лежит в основе **Интеллектуального интерфейса** (как мы его определяли на Лекции № 2).

Интеллектуальный интерфейс – совокупность программных и аппаратных средств, позволяющая конечному пользователю решать на компьютере характерные для его повседневной деятельности задачи без помощи посредников-программистов.

Расширение взаимодействия между человеком и компьютером с помощью:

- увеличения диапазона способов ввода и вывода;

- обогащения грамматики ввода и вывода;

- попытки кооперации с пользователем в достижении целей.

Эти расширения отражают тенденцию приближения человеко-машинного интерфейса к общению между людьми. Они требуют наличия в системе модели проблемной среды, в которой работают человек и компьютер и которая близка к модели, существующей в представлениях человека.

Еще несколько замечаний об интерфейсе.

(Факультативный материал; в программу второго коллоквиума и экзамена не входит! Но прочитать рекомендуется!)

Джефф Раскин в своей книге "Интерфейс: новые направления в проектировании компьютерных систем" [4, с. 7-13] отмечает (в цитируемые фрагменты лектором внесены отдельные мелкие изменения):

"1) Разработчики интерфейса должны не только включать в него четкие указания для пользователя, но и предусматривать возможность нелогичных действий с его стороны (в программистском сленге для обозначения этого есть термин "защита от дурака"). На самом деле, нежелательные последствия – это, конечно, в первую очередь результат непредусмотрительности программиста. Необходимо продумывать возможные стратегии поведения пользователя, чтобы не позволять ему "попадать в ловушку" (например, неожиданный выход из системы) или идти единственным путем: у него должен быть хотя бы небольшой выбор (например, чтобы не повторять каждый раз уже отработанную цепочку).

2) Это позволит как новичку, так и пользователю, выполняющему работу, быстро пройти дальше. Одновременно это обеспечит возможность человеку, который захочет исследовать интерфейс, поиграть во "что – если".

[Раскин подробно рассматривает требования, которым должны соответствовать исследуемые интерфейсы:]

- **Не следует помещать в главное меню пункты, срабатывающие сразу после нажатия.** Пользователи привыкли, что любой пункт главного меню в программах всегда содержит «подменю». Помещение в главное меню пункта, который срабатывает сразу при нажатии на него, делает программу непредсказуемой. Чаще всего это бывает пункт "Выход".
- **Делайте действия обратимыми.** Люди исследуют интерфейс не только с помощью навигации. Иногда они хотят знать, что случится, если они сделают что-то потенциально опасное. Иногда они не хотят этого, но такое может произойти случайно. Если действия обратимы, пользователь может исследовать интерфейс и не бояться за последствия.
- **Всегда предоставляйте отмену (undo).** Если Ваша программа не предоставляет возможности отмены, Вам неизбежно придется создавать массу диалогов с вопросами типа "Вы действительно уверены?". Это, безусловно, замедляет работу, но отсутствие таких диалогов (если Вы опустили "отмену") будет замедлять работу еще больше. Специальные исследования показали, что люди в опасной среде делают не больше ошибок, чем в отзывчивой и более ясной среде, но работают гораздо медленнее и осторожнее, чтобы избежать необратимых ошибок.

- **Всегда предоставляйте запасной выход.** Пользователи не должны чувствовать себя пойманными в ловушку. Путь назад должен всегда быть четко обозначен. Однако делайте так, чтобы остаться в программе было легче. Более ранние программы создавались так, чтобы их трудно было покинуть. С приходом Internet чаще появляются программы, в которых трудно остаться. Браузеры напоминают гирлянды объектов и опций, которые не имеют ничего общего с самой программой. Если пользователь имеет на экране набор элементов, в котором 49 элементов ведут напрямую к разрушению работы и только один-два способны реально помочь – такой интерфейс не является исследуемым.
- **Предлагайте пользователю постоянные наглядные подсказки, чтобы он чувствовал себя "как дома" (очень эффективны "всплывающие" подсказки).** Стабильные визуальные элементы не только позволяют людям быстро перемещаться, но и работают как указатели, давая пользователю ощущение психологического комфорта. Если в ставший привычным пользовательский интерфейс начать вносить некоторые изменения, то его, как и любой язык в такой ситуации, будет труднее понимать. Люди в большинстве своем консервативны по природе, не любят отказываться от привычек и привыкать к чему-то новому, особенно когда этого нового много. Этот фактор необходимо учитывать. Поэтому, если разрабатывается программа, аналог которой уже существует и которым пользуются многие, то имеет смысл посмотреть, как организовано взаимодействие с пользователем в уже существующей программе, и позаимствовать ее основные принципы, тогда новая программа не вызовет отторжения, так как сделана в привычном для пользователя стиле. Она будет более привычной для пользователя, а, главное, ему будет легче перейти к новой программе, если она лучше аналогичной.
- **Стремитесь к единообразию интерфейса в разных подсистемах (частях программы).** Необходимо тщательно продумать и осознать схему программы, приведя ее к системе (структуре, выстроенной по определенным правилам), и реализовать пользовательский интерфейс в соответствии с этой системой. Пользователю достаточно будет «схватить» суть, чтобы по аналогии разобраться в остальном. Следует избегать лишних элементов: чем меньше их будет, тем проще пользователю будет разобраться.
- **Обеспечивайте защиту работы пользователя.** Удостоверьтесь в том, что пользователь никогда не потеряет свою работу в результате либо своей ошибки, либо неустойчивости соединения, или по какой-то другой причине, кроме неизбежных, таких, как неожиданная потеря питания".

[Еще одна проблема, на которую обращает внимание Д. Раскин [там же], – **Сообщения, подтверждающие действие системы**]

"Сообщения, подтверждающие какое-либо действие системы, требуются для того, чтобы пользователь мог еще раз убедиться в том, что система выполнила или будет выполнять требуемое действие. Подтверждать ввод данных следует тогда, когда эти данные приведут к необратимым действиям системы (например, удаление записи из файла) или когда вводится код и пользователю необходимо проверить по соответствующему описанию, нужна ли запись выбрана (например, о конкретном энергоресурсе или административной области). Также может оказаться

желательным подтверждение, что было выполнено некоторое действие (например, добавлена запись в файл).

Может оказаться полезным не только подтверждение выполнения какого-то действия, но и вывод сообщения о состоянии системы (чтобы пользователь знал, где он находится). Это можно реализовать, отведя на экране специальную область, в которой будет отображаться путь, "пройденный" пользователем до текущего состояния.

Механизм *статуса* используется, чтобы держать пользователя в курсе того, что происходит. Механизмы статуса жизненно необходимы для предоставления информации человеку, чтобы он мог соответственно реагировать на изменение условий. Простой пример: человек, не получая информации о статусе, в течение периодов ожидания, пока процесс действительно не завершится, будет напрягаться и уставать без необходимости, так что в следующий раз у него может не хватить физических и умственных ресурсов на это.

Информация о статусе должна быть актуальной и легкодоступной. Пользователи не должны искать информацию о статусе. Они должны иметь возможность, одним взглядом окинув свою рабочую среду, получить хотя бы приближенное понятие о ходе работы и состоянии системы. Информация о статусе может быть, например, такой: иконка папки "Входящие" может иметь три состояния - пустая, наполненная и переполненная. Однако здесь нельзя переусердствовать. Иконка "мусорной корзины" на Макинтоше долгие годы изображала неминуемую опасность взрыва, когда хотя бы один документ находился внутри, так что люди старались очищать корзину сразу же после каждого документа. Это не только превратило одношаговую операцию в двухшаговую (перетащить в корзину, очистить корзину), но и лишило смысла саму идею мусорной корзины - возможность отмены удаления".

Раздел 2. Базовые понятия семиотики и лингвистики.

Семиотика

(от греч. σημεῖον — знак, признак)

(семиология) — научная дисциплина, изучающая общее в строении и функционировании различных знаковых (семиотических) систем, хранящих и передающих информацию, будь то системы, действующие в человеческом обществе (главным образом язык, а также некоторые явления культуры, обычаи и обряды, кино и т. д.), в природе (коммуникация в мире животных) или в самом человеке (например, зрительно и слуховое восприятие предметов; логическое рассуждение).

Словари и энциклопедии на Академике

<https://les.academic.ru/1010/%D0%A1%D0%B5%D0%BC%D0%B8%D0%BE%D1%82%D0%B8%D0%BA%D0%B0> (25.02.20.)

Лингвистика

(языкознание, языковедение) — наука, изучающая языки. Это наука о естественном человеческом языке вообще и о всех языках мира как индивидуальных его представителях. В широком смысле является частью семиотики как науки о знаках.

Словари и энциклопедии на Академике

<https://dic.academic.ru/dic.nsf/ruwiki/1006728> (25.02.20.)

Введение

Деятельность человека предполагает создание и использование абстрактных объектов (понятий), освобожденных от второстепенных, привходящих характеристик и отражающих наиболее существенные стороны действительности. Она реализует высшую форму регуляции деятельности человека в предметной среде.

В свою очередь существенность, значимость отражаемого определяется деятельным контекстом: в процессе управляемой мышлением деятельности и в соотношении с ее целями и средствами. Очевидно, что деятельность человека нельзя рассматривать в отрыве от его общественных отношений. В социальной среде, в условиях общественного разделения труда структура человеческой деятельности усложняется. Человек может выполнять лишь отдельные этапы решения задачи, а цель деятельности может быть связана с его непосредственными потребностями косвенным путем. При этом функции регуляции совместной и дифференцированной деятельности выполняет **коммуникативная деятельность (общение)** – обмен информацией о деятельности индивидуальной.

Язык, языковой знак

В качестве средства такого обмена (средства общения) используется знаковая система социального уровня – **язык**. Связи языковых знаков (элементов языка) с психическими явлениями и объектами психического взаимодействия объективированы в социальной группе носителей данного языка и относительно константны для нее. Каждый член группы воспринимает эти связи как данные извне, как существующие объективно, а процесс усвоения этих связей (и знаков языка) – одна из необходимых предпосылок формирования личности, включения человека в систему общественных отношений. Овладевая языком, человек

приобретает детально разработанную систему объективации и реализации собственной психической деятельности, активно используемую им в ходе социальных взаимодействий.

Основными элементами языка как знаковой системы являются **языковые знаки**, образующие субстанцию языка. Они связаны внутрисистемными отношениями, определяющими как структуру языка в целом, так и структуру конкретных знаковых конструкций, и участвуют в отношениях с внеязыковыми объектами.

Свойства и аспекты языкового знака

Языковой знак – материальный объект, поставленный в соответствие некоторому другому объекту и заменяющий последний в ходе деятельности (свойство **знаковости**). Другими словами, если некоторый объект является знаком, то он поставлен в соответствие некоторому другому объекту и способен его заменять. Отметим также, что знаки следует отличать от единиц языка: фонем, морфем и слов (в естественном языке). Фонемы свойством знаковости не обладают и служат лишь исходным материалом для построения знаков языка, а морфемы и слова являются знаками.

Второй атрибут языкового знака – его **конвенциональность**, или немотивированность. Это свойство означает, что устанавливается указанное соответствие соглашением людей, использующих язык (всюду в данном разделе, если специально не оговаривается иное, речь идет о языке человека). Знак может не иметь никакого сходства с объектом, в соответствие которому он поставлен, и не быть связанным с ним причинно-следственными отношениями.

Однако выбор материального объекта на роль знака не может быть произволен абсолютно. Объект должен обладать так называемыми системообразующими свойствами – необходимыми для включения его в систему языка.

К числу системообразующих свойств языка относятся **дискретность** («членораздельность») и **неоднородность** («различаемость») его элементов. Неоднородность языка проявляется и в его **иерархичности** – в языке может быть выделено несколько иерархических неоднородных уровней, единицы каждого из которых относительно однородны (морфемы – как единицы морфологического уровня, например). С особенностями речевого аппарата человека связан принцип **линейности** в языке – в конструкциях, построенных по правилам языка, знаки могут располагаться лишь в линейной последовательности, то есть цепочкой.

Как элементы системы (языка) знаки обладают еще рядом свойств. Связь знака с внеязыковыми объектами задается отношениями: **сигматическими** – связь знака с реальными объектами и явлениями действительности или отдельными аспектами их; **семантическими** – связь с психическими моделями соответствующих сторон реальности или с моделями реально не существующих объектов; **прагматическими** – связь с людьми, использующими знаки языка в своей деятельности.

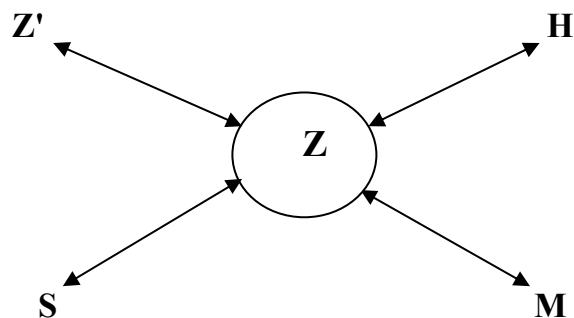


Рис. 1. Четыре аспекта языкового знака.

Синтаксические отношения (при описании естественных языков обычно рассматриваются два уровня синтаксических отношений: морфологический и синтаксический в узком смысле) характеризуют связи между знаками как элементами языка: иерархические – отношения вхождения знака в сложный знак; синтагматические – отношения взаимодействия знаков или их классов; парадигматические – отношения между элементами одного класса, например, формами одного слова.

Характеристика языка как знаковой системы, элементы которой связаны отношениями с различными сторонами действительности, полезна и при рассмотрении процессов его использования. Язык является знаковой системой социального уровня. Выбор знаков, правил их комбинирования и соотнесения с явлениями объективного и субъективного планов закреплен «общественным соглашением», которое разделяют все владеющие данным языком. Тем самым язык задает нормы интерпретации и употребления знаков. А в реальных процессах использования языка на основе этих общих норм строятся и анализируются конкретные знаковые конструкции, описывающие конкретные ситуации.

Таким образом, процесс использования языка – **речевая деятельность** (РД) – может быть охарактеризован как актуализация и конкретизация существующих в языке в потенциальной форме синтаксических отношений и отношений знаков с внеязыковыми объектами: выбор нужной формы слова при синтезе, установление актуального сигматического отношения с реальным объектом – **денотатом** и т.п. Результатом РД при синтезе является **сообщение**, несущее определенную информацию и о самом языке (по правилам которого оно построено), и о ситуации, в которой оно было синтезировано, и об авторе.

В структурном плане сообщение представляет собой наделенную структурой линейную последовательность знаков. Каждый знак характеризуется его позицией в цепочке и актуальными синтаксическими отношениями с другими знаками. «Нечленимые» в языке знаки – материальные оболочки «элементарных» языковых единиц (морфем, неизменяемых слов, фразеологизмов) – «воспроизводятся» в речевом произведении в готовом виде. Структурно сложные знаки строятся из составляющих по правилам языка.

Общение, или коммуникативная деятельность, предполагает наличие не менее двух носителей языка – **автора** сообщения и субъекта, которому это сообщение адресовано, – **реципиента** (в случае диалога они естественным образом поочередно «меняются местами»), а также наличие некоторой совместной деятельности, которую общение должно регулировать.

Естественные и искусственные языки

Определенное сходство в структурном и функциональном плане с естественными языками имеют и представители класса искусственных языков.

Хотя проблема соотношения естественных и искусственных языков представляется на первый взгляд (и является на самом деле) тривиальной, при ее рассмотрении часто допускаются ошибки. Одна из наиболее распространенных – абсолютизация временных и несущественных различий (существенные при этом как правило упускаются из виду), учет особенностей искусственных языков, существующих в данный момент, и неоправданный перенос выявляемых при этом различий на весь класс искусственных языков вообще. Очевидно, что в этом случае и сама проблема подменяется другой.

Единственное непреходящее отличие естественного языка от всех прочих – которые и следует называть искусственными – связано с историей его возникновения. *Естественный язык* – продукт, естественно-исторически возникший из объективных общественных потребностей (в первую очередь из потребности в общении, регулирующей совместную и дифференцированную деятельность) на ранних этапах общественного развития, когда человеческое познание было практически нерелексивным и никакое активное сознательное регулирование процесса создания языка было невозможно.

Объективность существования ЕЯ объясняет, почему многие языковые реликты, исключения и нерегулярности "живут" в языке до сих пор, а попытки избавиться от них, модернизировать, улучшить язык часто заканчиваются неудачей. Для того, чтобы некоторые новшества попали в язык, "закрепились" в нем, их должно усвоить большинство носителей языка (и при спонтанном развитии языка новшества, возникшие в речи, попадают в язык лишь в этом случае). Это и есть объективация. В противном же случае (без объективации) модернизатор, пусть даже руководствующийся самыми благими намерениями, окажется в положении Шалтая-Болтая из "Алисы в Зазеркалье".

Примечательно, что хотя существенные отклонения от общих языковых правил и норм недопустимы, индивидуальные модели ЕЯ (*индивидуальные языковые модели* – ИЯМ), усвоенные его носителями и определяющие особенности речи последних, могут иметь некоторое своеобразие. Они описывают своего рода "подмножество-расширение" эталонного языка, не содержащее неизвестных конкретному носителю языка знаков и правил и включающее ряд конструкций и форм, не входящих в общелитературный язык (заведомо неграмматичных) или используемых в рамках узких социальных групп (формы просторечные и разговорные, жаргонизмы, профессионализмы). Несмотря на формальную недопустимость или ограничения на употребление подобных форм, они (эти формы) не только встречаются в повседневной речи (соответствующие запреты либо вообще не входят в ИЯМ носителя языка, либо случайно или намеренно им игнорируются), но и понимаются людьми.

Понятие ИЯМ подчеркивает факт субъективного преломления естественного языка как знаковой системы социального уровня в ходе его усвоения и использования. Формирование ИЯМ – сложный и длительный процесс, включенный в процесс развития человеческой психики и детерминируемый как

внешними (социальная среда), так и внутренними (индивидуальные психофизиологические особенности) факторами. Примечательно, что с уменьшением социальной группы (производственный коллектив, семья) своеобразие используемого в ней подмножества-расширения ЕЯ возрастает. И это естественно: усвоить отклонения от языка-эталона должно в этом случае небольшое число лиц.

Характерной, но не обязательно отличительной чертой любого естественного языка является его **универсальность**. Возникнув как средство регуляции самых разнообразных видов человеческой деятельности (протекающей в различных контекстах, предполагающей использование различных средств, направленной на различные объекты), ЕЯ может быть использован для **выражения качественно различных видов содержания, для описания реальной действительности и процессов психической деятельности с разной степенью строгости, полноты, эксплицитности**.

Автор сообщения, например, может описать некоторую ситуацию в весьма общих чертах и, не потеряв при этом объективно существенные (но не сочтенные им таковыми) детали, возложить задачу выявления этих деталей и дальнейшей конкретизации совместной деятельности на реципиента.

И хотя в ходе последующих языковых взаимодействий могут потребоваться новые языковые средства (необходимые для выражения новых видов информации), соответствующее расширение ЕЯ, представляющего собой открытую систему, всегда возможно. При этом способность ЕЯ служить метаязыком для себя позволяет описать такое расширение средствами самого языка.

Поскольку ЕЯ является средством объективации психической деятельности человека (построения моделей отражаемого) и средством передачи информации об отражаемом, сигматические, семантические и прагматические аспекты знаков чрезвычайно существенны в процессах использования языка. Важную роль играют и синтаксические отношения, существующие в потенциальной форме в языке и актуализируемые в РД. Возможность передачи информации с необходимостью предполагает структурированность языковых выражений, репрезентующих эту информацию, а многие тонкие оттенки содержания передаются исключительно синтаксическими средствами (например, порядком слов).

Знаки и значения

Синтаксическое значение существует в языке как система ассоциированных с данным сообщением эталонных парадигматических, синтагматических и иерархических связей с другими знаками языка – аналог «значимости» в смысле одного из лингвистов-классиков Ф. де Соссюра. Являясь соотносительной характеристикой знаков языка, оно определяет выбор некоторого знака как средства выражения других видов значения. **Сигматическое значение** – класс реальных объектов («денотатов», или «обозначаемых»), в соответствие которым может быть поставлено сообщение, в то время как **семантическое значение** отсылает к классу эталонных психических моделей денотатов (к «десигнатам», «означаемым», или «концептам»). **Прагматическое значение** представляет собой класс нормативно соотносенных с сообщением действий потенциальных

реципиентов или же класс действий и целей потенциального автора сообщения, побуждающих его к речевой деятельности.

Полное **значение сообщения** является комплексом четырех указанных видов значения. Возможны и «вырожденные» случаи. Так, например, сигматическое значение некоторых знаков может быть пустым (всегда – *кентавр*, в определенном временном интервале – *Великий инквизитор*); пустая морфа не изображается никаким знаком (хотя, например, в словоформе *слов* отсутствие флексии передает грамматическое значение родительный падеж, множественное число); служебные морфемы обладают лишь грамматическим и крайне абстрактным семантическим (грамматический род, число) значением.

Отметим, что для ЕЯ характерно отсутствие взаимнооднозначного соответствия между знаками и связанными с ними обозначаемыми, означаемыми и действиями.

Знак или сообщение называется **омонимичным**, если связанные с ним в языке или в речевом произведении классы обозначаемых, означаемых или деятельных актов содержат более одного элемента. Частным случаем омонимии является **полисемия** – наличие у слова нескольких обозначаемых (означаемых): *оросительный канал* ↔ *канал ствола орудия* ↔ *канал связи с ЭВМ* ↔ *дипломатические каналы*. Возможна и грамматическая омонимия: *второй параграф* ↔ *о второй главе* ↔ *для второй главы*, *цинковые белила* ↔ *белила потолок*, *прием посла* = *посол принял* <кого-то> ↔ <кто-то> *принял посла*.

В письменной речи неразличимы формы: *профессора́* ↔ *профэссора*, *ко́лет* ↔ *колёт*, (если нет проставлены ударения), *это все твои вещи?* ↔ *это всё твои вещи?* (если не используется буква ё), а в устной (**омофония**) – конструкции *и скота* ↔ *из кота*. Определенный интерес представляют сходные по строению, но имеющие несовпадающие значения слова (**паронимы**): *языковой* ↔ *языковой*, *представить* ↔ *предоставить* (*языковая колбаса* ↔ *языковой знак*, *представить справку* ↔ *предоставить возможность*).

Если же различные знаки имеют «общее значение» и могут, в частности, заменять друг друга в сообщении (общее в синтаксическом значении, «значимости»), то их обычно называют **синонимами**. «Абсолютная» синонимия в ЕЯ встречается редко: *языкознание* = *лингвистика* = *языковедение*; флексии *-е* и *-ей* с грамматическим значением сравнительная степень прилагательного, находящиеся в отношении свободного чередования: *весел-е* = *весел-ей*.

Как правило, возможность замены знака другим, то есть употребления синонима, ограничена нетождественностью «общего значения», контекстными условиями. Так, например, морфологические синонимы – флексии с «общим значением» мужской род, множественное число, именительный падеж: *-ы*, *-и*, *-е*, *-а*, *-я* – употребляются с существительными различных типов склонения (*лингвисты*, *филолог-и*, *южан-е*, *профессор-а*, *кра-я*). К контекстным условиям относятся также лексическая сочетаемость и стилистическая окраска. Например, замена *хочу* на *желаю* во фразе *Я не хочу его видеть* допустима, а во фразе *Кроме нас хотят переселиться в колхоз еще несколько семей* – нет.

Описывая одну и ту же ситуацию, одно обозначаемое с помощью различных синонимичных конструкций (синонимичными, естественно, могут быть и

предложения, тексты), носитель языка имеет возможность переструктурировать не только сообщение, но и репрезентуемую им модель ситуации, выделять то одни, то другие компоненты и аспекты.

Пусть Z и Z' - некоторые знаки, а $v(Z)$ и $v(Z')$ - их значения (полные значения в языке). В общем случае возможны следующие ситуации:

Рассмотрим ситуации сочетания пустых/непустых знаков (морфем) и их значений.

$Z = \emptyset \quad v(Z) = \emptyset$ - тривиальный случай;

$Z = \emptyset \quad v(Z) \neq \emptyset$ - "нулевой аффикс":

sheep - *sheep*[- \emptyset] - его значение "мн. число сущ.",

стол[- \emptyset] - его значение "сущ. в форме им. или вин. падежа",

шуб[- \emptyset] - его значение "сущ. в форме род. падежа"

$Z \neq \emptyset \quad v(Z) = \emptyset$ - "пустая морфа", например интерфиксы (соединительные гласные): *пар-о-воз*. Я (в отличие от некоторых лингвистов) не считаю, что в этом случае $v(Z) = \emptyset$;

$Z \neq \emptyset \quad v(Z) \neq \emptyset$ - тривиальный случай.

Рассмотрим теперь ситуации, в которых два знака "пересекаются" (имеют общую подцепочку) или не пересекаются (не имеют такой подцепочки), а их значения либо "пересекаются" (здесь пересечение определить формально чрезвычайно трудно), либо нет. Возможны, опять же, 4 варианта:

$Z \cap Z' = \emptyset \quad v(Z) \cap v(Z') = \emptyset$ - тривиальный случай (хотя есть и вопросы: *слон* - *моська*);

$Z \cap Z' = \emptyset \quad v(Z) \cap v(Z') \neq \emptyset$ - синонимы (языковед - лингвист), гипонимы/гиперонимы (*орел* - *птица*), супплетивные формы (*am* - *is* - *are*, *много* - *больше*, *ребенок* - *дети*), пары типа: *отец* - *сын*, *мальчик* - *девочка*, *теща* - *жена*;

$Z \cap Z' \neq \emptyset \quad v(Z) \cap v(Z') = \emptyset$ - омонимы; *опять* - *пять*, *окно* - *окунь*, *оса* - *осел* (ср. *коза* - *козел*);

$Z \cap Z' \neq \emptyset \quad v(Z) \cap v(Z') \neq \emptyset$:

$v(Z) = v(Z')$ - *волчиха* - *волчица*;

$v(Z) \supset v(Z')$ - *москвич* - *москвичка*,

$v(Z) \sim v(Z')$ - *перечитать* - *чтение*, *одеваться* - *переодеваться*

- *раздеваться*.

Значение и смысл

Согласно данному выше определению значение сообщения является комплексом эталонных для некоторого ЕЯ внутриязыковых связей знаков и ассоциированных со знаками внеязыковых объектов. А поскольку при усвоении естественного языка конкретным человеком, при формировании ИЯМ язык претерпевает субъективное преломление, отклонения от эталона имеют место и в сфере значений – субъективное сужение или расширение значения. Возможно, что отдельные виды значения при этом не меняются или меняются в разной степени. Свообразием может отличаться и значение, ассоциированное со знаком в той или иной социальной группе (научные термины, профессионализмы, диалекты языка, жаргонизмы).

Учет субъективной и социальной соотнесенности значения (как следствие преломления языка в ИЯМ или в модели языка, общей для членов социальной группы) – необходимая предпосылка эффективного использования ЕЯ для общения между людьми, между людьми и искусственными информационными системами.

В конкретных процессах РД происходит дальнейшая модификация значения – с сообщением связываются лишь отдельные компоненты преломленного в ИЯМ значения. Подобная актуализация предполагает выбор уместных в текущем контексте аспектов семантического и прагматического (учет конкретной цели автора сообщения, особенностей собеседника) значений; установление, если это возможно, связи с обозначаемым; выбор (раскрытие) синтаксических средств выражения значения.

Соотнесенная с сообщением в реальном процессе речевой деятельности подсистема значения (виртуально ассоциированного с данным сообщением в ИЯМ носителя языка) может быть названа *смыслом сообщения*.

Понимание выражений естественного языка

С привлечением понятий *значения* и *смысла* сообщения можно конкретизировать описание информационного аспекта общения, регулирующего совместную и дифференцированную деятельность. Автор очередного сообщения строит его таким образом и с использованием таких (представленных в его ИЯМ) языковых средств, чтобы смысл сообщения максимально точно отображал важнейшие в текущей контекстной ситуации аспекты деятельности, преследуемые им цели. Задача же реципиента – выявить этот смысл, то есть установить те стороны значения (допустимого с позиций его ИЯМ), которые наиболее существенны в текущей ситуации с его точки зрения и которые, как он предполагает, имел в виду автор сообщения.

Подобный процесс раскрытия смысла и назван *пониманием сообщения*.

Понимание сообщения можно трактовать как вид интеллектуальной деятельности, обладающий всеми атрибутами ее и, естественно, всеми атрибутами психического отражения: информативностью (выявляется информация об актуальных связях знаков) и субъективностью (учитывается информация, существенная с позиций реципиента). Процесс понимания может быть прерывистым: выдвижение подцелей, возврат на предыдущие этапы и коррекция. Кроме того, он может сопровождаться адаптацией как к языку в целом, так и к особенностям ИЯМ собеседника.

В силу того, что понимание языковых выражений – один из наиболее привычных для человека видов интеллектуальной деятельности, многие этапы понимания реализуются с помощью неосознаваемых человеком средств – вторичных автоматизмов, сформированных в ходе усвоения языка.

Однако «ключевыми» в текущем коммуникативном акте могут быть различные виды значения. Одно и то же сообщение может быть синтезировано с разными целями: установить актуальные семантические и сигматические связи и выполнить действия над соответствующими объектами, запомнить смысл сообщения и др.

Замечательный пример, иллюстрирующий возможность неправильного выделения ключевого вида значения описан Дж. Литлвудом. В печатном тексте одной из своих работ Литлвуд не обнаружил последней фразы – *Таким образом, σ следует сделать сколь возможно малым*. Однако вместо нее на пустом месте в конце статьи стояла миниатюрнейшая σ. Наборщик вместо того, чтобы воспроизвести данную фразу (что он и должен был сделать по роду своей профессии), выполнил потенциально предусматриваемые ее значением действия.

Примечательно, что своеобразие восприятия смысла часто определяется устойчивыми факторами, в первую очередь социальной ролью реципиента (поэтому ошибку наборщика следует признать экстраординарной). Так, лингвиста могут заинтересовать несущественные для прочих носителей языка проявления в речи языковых закономерностей.

В силу действия подобных осложняющих факторов понимание должно предполагать ориентацию на ИЯМ автора полученного сообщения, учет целей его обращения к реципиенту, а также адаптацию и обучение, направленные на сближение используемых собеседниками ИЯМ и повышение точности отражения

в них языка. Основная предпосылка близости ИЯМ – активные коммуникативные взаимодействия, протекающие в общем деятельном контексте (так, в небольшой социальной группе ИЯМ достаточно близки, хотя они, как отмечалось, могут отражать язык с искажениями). А для сближения смыслов можно использовать перефразирование сообщений, упоминавшееся в связи с синонимией, и уточняющий диалог.

Разумеется, такой диалог будет возможен и плодотворен лишь в том случае, когда реципиент осознает, что он понял сообщение «как-то не так», что в сообщении не указаны важные с его точки зрения детали: *Этого просто не вынести! А что вам нужно вынести? – спросила Алиса* (Л. Кэрролл, «Алиса в Стране Чудес»), что в его распоряжении вообще нет средств приписать сообщению какую-либо интерпретацию.

Для того, чтобы добиться правильного понимания адресуемых ему сообщений, каждый из участников процесса общения должен располагать информацией об определяющей предмет общения проблемной среде, о языке (эта информация представлена в его ИЯМ), о собеседнике, в том числе и об используемой им ИЯМ, и о себе. Эта информация соответствует **глобальному контексту** общения.

Естественно, что при обработке очередного сообщения (отдельной фразы, абзаца и т.п.) важную роль играет и информация, почерпнутая из предшествующих сообщений (из **локального контекста**). Именно учет глобального и локального контекстов: предмета обсуждения, собственных целей и целей собеседника, языковых и внеязыковых связей между отдельными сообщениями – и помогает реципиенту приписать очередному сообщению наиболее уместную интерпретацию, то есть правильно понять его.

Установив, о чем идет речь в сообщении, как должна быть использована содержащаяся в нем информация, реципиент может относительно легко разрешать проблемы, возникающие при анализе чисто знаковых (синтаксических) отношений, определяющих структуру сообщения.

Иллюстрирует эти возможности способность человека:

1) выбирать «наиболее разумную» интерпретацию сообщения, отсеивая интерпретации неестественные (но формально допустимые): *За безбилетный проезд и провоз одного места багажа взимается штраф 1 рубль, Сведения о войсках противника, которые помогли нашим партизанам, В черных костюмах выступают наши фигуристы, которые отделаны красными и зелеными цветами; Новый американский десантный корабль-док USS Portland введен в строй в городе Портленд в штате Орегон, на котором вскоре будет установлена лазерная пушка;*

2) понимать неграмматичные (ошибочные) конструкции: *Ошибка вслове лектор, В аудиторию вошли лектора [следует: лекторы], Предоставить [следует: представить] справку в бухгалтерию – и грамматически неоформленные квазифразы типа: ребен- спа- комнат- шир- распах- окн-;*

3) определять по контексту достаточные с точки зрения текущего этапа общения аспекты значений и функциональные роли в тексте незнакомых слов и конструкций. Читатель «Алисы в Зазеркалье», например, достаточно ясно

представляет себе, что произошло с головой Бармаглота (... *Взы-взы – стриждает меч, Ува! Ува! И голова Барабардает с плеч!*), хотя и не знает семантическое и сигматическое значения незнакомого глагола *барабардать*.

Примечательно, что ориентация на «высшие» аспекты значения (сигматический, семантический и прагматический), то есть на внеязыковые связи знака характерна и для более частных видов речевой деятельности. Так, согласно данным психолингвистики и при выборе слов из долговременной памяти человек ориентируется в первую очередь на их семантические значения и связи. Использование других критериев, звукового сходства, например, свидетельствует либо о невозможности обращения к семантическому уровню (незнание семантического значения слова), либо о нарушении психической деятельности (шизофрения).

В этой связи можно вспомнить знаменитую фразу Л.В.Щербы *Глокая кудра итеко будланула бокра и кудрячит бокренка*. Невозможность установить сигматические и семантические отношения квазислов этой фразы заставляет человека при ее анализе обратиться к чисто знаковым (синтаксическим) отношениям. Предполагая грамматическую корректность фразы, можно исследовать ее синтаксические свойства: порядок слов, словоизменение, словообразование (*бокр ↔ бокр-енок*). Определенные ассоциации могут возникнуть и при анализе знаковой (звуковой) структуры корневых морфем. Так, *глокость* может показаться кому-то очень нехорошим качеством, а глагол *кудрячить* может ассоциироваться либо с существительным *кудри*, либо с глаголами *корчить* или *корячить*.

Несомненно, «высшие» аспекты значения передаются с помощью знаковых (синтаксических) средств, а проникнуть на эти «высшие» уровни не удастся, не начав анализа структуры сообщения. Однако можно предположить, что по мере раскрытия внеязыковых связей знаков – даже до завершения анализа синтаксической структуры сообщения в целом – происходит переход на уровень информационной модели описываемой ситуации. Причем выявляемая информация (семантическая, сигматическая, прагматическая) не только пополняет эту модель, но и управляет дальнейшим анализом текста.

Некоторые сведения из морфологии (словообразование) естественных языков.

На примере словообразования – образования новых слов от однокоренных – мы рассмотрим сложность и разнообразие языковых структур.

Знаки естественного языка (ЕЯ) – **корни** и **аффиксы** (несколько типов):

корни – семантические и сигматические значения обычно очевидны; во всех ЕЯ их около 100 000.

аффиксы – не может быть словом (переходить на более высокий уровень иерархии); в ЕЯ их около 100.

Способы словообразования: *аффиксация, модификация, изменение сочетаемости.*

Аффиксация

Разрывается корень?	Разрывается аффикс?	Вид аффикса	Примеры
-	-	префикс (приставка) суффикс интерфикс	<i>переход, напечатать</i> <i>окошко, лукошко</i> (уменьшительно-ласкательная форма) <i>нефтеперевозки, первоцвет</i>
-	+	циркумфикс	<i>счетвером, подоконник</i>
+	-	инфикс	<i>vic-i</i> (я победил) – <i>vincō</i> (я побеждаю) (лат.), <i>stand-stood</i> (англ.)
+	+	трансфикс	<i>ktb</i> - корень "суть процессатисания", <i>kataba</i> (он написал), <i>maktabun</i> (канцелярия) (араб.) ¹

Основные определения:

"аффиксация

Использование аффиксов при образовании новых слов как один из способов словопроизводства (в лингвистике)".

[Словари и энциклопедии на Академике](https://dic.academic.ru/dic.nsf/efremova/272537/%D0%B0%D1%84%D1%84%D0%B8%D0%BA%D1%81%D0%B0%D1%86%D0%B8%D1%8F)

<https://dic.academic.ru/dic.nsf/efremova/272537/%D0%B0%D1%84%D1%84%D0%B8%D0%BA%D1%81%D0%B0%D1%86%D0%B8%D1%8F> (02.02.20.)

"аффикс

(лат. affixus — прикрепленный).

Служебная морфема, т. е. часть слова, видоизменяющая лексическое либо [грамматическое значение](#) корня (основы) или выражающая отношения между словами в словосочетании или предложении; общее название всех морфем, за исключением корня".

[Словари и энциклопедии на Академике](https://dic.academic.ru/dic.nsf/lingvistic/101/%D0%B0%D1%84%D1%84%D0%B8%D0%BA%D1%81)

<https://dic.academic.ru/dic.nsf/lingvistic/101/%D0%B0%D1%84%D1%84%D0%B8%D0%BA%D1%81> (02.02.20.)

¹ Пример с сайта LINGVOBLOG <http://www.lingvoblog.com/j-note37/> (02.02.20.)

"приставка

3. Составная часть слова, стоящая перед корнем; то же, что префикс (грам.)".

Словари и энциклопедии на Академике
<https://dic.academic.ru/dic.nsf/ushakov/976482> (02.02.20.)

"суффикс

(от лат. suffixus-подставленный) (линг.).

Словообразовательный элемент - часть слова, находящаяся между корнем и окончанием (флексией) и придающая определенное значение слову сравнительно с другими однокоренными словами, напр. *иц* или *ов* словах столица, столовый.

В некоторых лингвистических сочинениях (различающих суффиксы словообразова- тельные и словоизменительные) - всякая часть слова, стоящая за корнем, т.е. в том числе и т.н. флексия".

Словари и энциклопедии на Академике
<https://dic.academic.ru/dic.nsf/ushakov/1047990> (02.02.20.)

"интерфикс

Соединительная морфема, не имеющая самостоятельного словообразовательного значения и

располагающаяся между корнями сложного слова (*в лингвистике*)".

Словари и энциклопедии на Академике
<https://dic.academic.ru/dic.nsf/efremova/171253/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B8%D0%BA%D1%81> (02.02.20.)

"циркумфикс

Аффикс, морфема, которая ставится в начале и в конце другой морфемы. Циркумфиксы контрастируют с префиксами, которые прибавляются к началу слова, суффиксами, которые добавляются в конец слова, и инфиксами, которые вставляются в середину слова".

Словари и энциклопедии на Академике
<https://dic.academic.ru/dic.nsf/ruwiki/1344142> (02.02.20.)

"инфикс

(от лат. infixus — вставленный)

Аффикс вставляемый внутрь основы при словообразовании или словоизменении. И. является важным средством аффиксации, например, в индонезийских языках: тагальском s-um-ulat — «писать», s-in-ulat — «был написан» от sulat — «письмо». Так называемый носовой И. в индоевропейских языках (лат. vic-i — «я победил», vi-n-co — «я побеждаю» и др.) носит более спорный характер, поскольку он не имеет определенного значения. В русском языке имеются чередования, восходящие к носовому индоевропейскому И.: «лечь» — «лягу» «сесть» — «сяду» от leg-ti — le-n-g-o и sed-ti — se-n-do".

Словари и энциклопедии на Академике

<https://dic.academic.ru/dic.nsf/bse/90890/%D0%98%D0%BD%D1%84%D0%B8%D0%BA%D1%81> (02.02.20.)

"трансфикс

(лат. transfixus - пробитый, пронзенный)

лингв. разновидность аффикса, разрывающая корень слова, состоящий из согласных, путем вставки гласных (напр., в трехгласном корне арабского языка или в других семитских языках)".

Словари и энциклопедии на Академике

https://dic.academic.ru/dic.nsf/dic_fwords/36472/%D0%A2%D0%A0%D0%90%D0%9D%D0%A1%D0%A4%D0%98%D0%9A%D0%A1 (02.02.20.)

Модификация

"метатеза метатезис

перестановка согласных звуков в языке; одно из так называемых фонетических явлений языка. Наблюдается чаще всего у согласных, наиболее способных к образованию самостоятельных слогов без помощи гласных звуков, т. е. у звуков "плавных" р, л и носовых м, н. В прежние времена лингвисты сильно злоупотребляли понятием М.; теперь этим приемом объяснения пользуются сравнительно редко и с большой осторожностью. Так, например, в народной форме протупея вместо портупея (=porte-épre e), по-видимому, имеется М., но она могла возникнуть и путем переделки данного слова под влиянием так называемой "народной этимологии". На место непривычного звукового сочетания, не совпадающего с природными русскими корнями, подставлено имеющееся уже в составе языка. Протупея напоминает сложения с предлогом про-; корень туп- имеется в русском языке (туп-ой), также как и суффикс -ея. Здесь, таким образом, метатезис может быть мнимым, нефонетическим. Имеются, однако, и несомненные случаи М.: пол. broda, старослав. брада возникли из более древней формы borda, barda (ср. нем. Bart, фр. barbe, лит. barzd à); рус. лось — из олсь (ср. лат. alces, греч. άλκη) и т. д. Здесь мы имеем дело с широко развитым в славянских языках явлением М. плавных согласных, известным также под именем полногласия".

Энциклопедический словарь Ф.А. Брокгауза и И.А. Ефрона

Словари и энциклопедии на Академике

https://dic.academic.ru/dic.nsf/brokgauz_efron/66989/%D0%9C%D0%B5%D1%82%D0%B0%D1%82%D0%B5%D0%B7%D0%B0 (02.02.20.)

"редупликация в языкознании

РЕДУПЛИКАЦИЯ (от позднелат. reduplicatio — удвоение), полное или частичное повторение корня, основы или слова как способ образования слов, грамматических форм, фразеологических единиц и др. (напр., в малайском языке orang — «человек», orangorang — «люди», русском — «ходишь-ходишь»)".

Словари и энциклопедии на Академике

<https://dic.academic.ru/dic.nsf/es/86835/%D0%A0%D0%95%D0%94%D0%A3%D0%9F%D0%9B%D0%98%D0%9A%D0%90%D0%A6%D0%98%D0%AF> (02.02.20.)

В работе Е.Ф.Арсентьевой и Р.Б.Валиуллиной "Редупликация в современном английском и русском языках"

<https://cyberleninka.ru/article/n/reduplikatsiya-v-sovremennom-angliyskom-i-russkom-yazykah> (02.02.20.)

разделяются случаи:

полной:

большой-большой, далеко-далеко, bye-bye

и неполной:

тик-так, тиф-паф, humpty-dumpty

редупликации.

"чередование — Появление различных звуков в одном и том же месте в структуре одной и той же морфемы в различных условиях ее функционирования..."

[Словари и энциклопедии на Академике](#)

<https://dic.academic.ru/dic.nsf/ogegova/265662> (02.02.20.)

"аблаут — (нем. Ablaut) (апофония) чередования гласных в составе одной морфемы, нередко выступающие в качестве внутренней флексии. Напр., англ. sing sang sung, рус. собирать сбор"...

[Словари и энциклопедии на Академике](#)

<https://dic.academic.ru/synonyms/%D0%B0%D0%B1%D0%BB%D0%B0%D1%83%D1%82> (02.02.20.)

Чередования такого типа называют "значащими" — меняющими значение слова . — в отличие от альтернации ("незначащих" чередований, в результате которых мы получаем алломорфы одной и той же морфемы).

"альтернация (лат. alternatio - чередование) — лингв. смена фонем или графем в основах слов и аффиксах (напр., «пиСать» - «пиШу» - [с // ш] в основе); при а. одна МОРФЕМА выступает в виде двух или более вариантов (напр., «воз-/вос-», «из-/ис-» и т.п.). Ср. АЛЛОМОРФ".

[Словари и энциклопедии на Академике](#)

https://dic.academic.ru/dic.nsf/dic_fwords/3175/%D0%90%D0%9B%D0%AC%D0%A2%D0%95%D0%A0%D0%9D%D0%90%D0%A6%D0%98%D0%AF (02.02.20.).

Мы рассмотрим некоторые отношения между знаками и их значениями на примере морфологического уровня языка (как словоизменения, так и словообразования). Основные единицы морфологического уровня: **морфемы**, **[алло]морфы**. **Морфема** - собственно языковой знак; **[алло]морфа** - вариант морфемы, выбор которого обусловлен окружением. Основные ситуации:

контрастная дистрибуция (выбор алломорфы меняет значение: *she was charm-ed - she was charm-ing*).

дополнительная дистрибуция (явная зависимость от окружения/контекста): /-z, -s, -iz, -in, -Ø/ - алломорфы морфемы {-Z1} "множественное число существительного": *pen - pens [-z], map - maps [-s], box - boxes [-iz], ox - oxen [-in], sheep - sheep [-Ø]*. Вообще говоря, существуют и другие способы образования форм

множественного числа существительных (man - men, mouse - mice, datum - data, child - children), однако мы рассматриваем не проблему форм существительных, а проблему дополнительной дистрибуции.

Знак или сообщение называется **омонимичным**, если связанные с ним в языке или в речевом произведении классы обозначаемых, означаемых или деятельных актов содержат более одного элемента. Частным случаем омонимии является **полисемия** – наличие у слова нескольких обозначаемых (означаемых): *оросительный канал* ↔ *канал ствола орудия* ↔ *канал связи с ЭВМ* ↔ *дипломатические каналы*. Возможна и грамматическая омонимия: *второй параграф* ↔ *о второй главе* ↔ *для второй главы*, *цинковые белила* ↔ *белила потолок*, *прием посла* = *посол принял* <кого-то> ↔ <кто-то> *принял посла*.

В письменной речи неразличимы формы: *профессора́* ↔ *профэссора*, *ко́лет* ↔ *колéт*, (если нет проставлены ударения), *это все твои вещи?* ↔ *это всё твои вещи?* (если не используется буква ё), а в устной (**омофония**) – конструкции *и скота* ↔ *из кота*. Определенный интерес представляют сходные по строению, но имеющие несовпадающие значения слова (**паронимы**): *языковой* ↔ *языковой*, *представит* ↔ *предоставит* (*языковая колбаса* ↔ *языковой знак*, *представит справку* ↔ *предоставит возможность*).

Если же различные знаки имеют «общее значение» и могут, в частности, заменять друг друга в сообщении (общее в синтаксическом значении, «значимости»), то их обычно называют **синонимами**. «Абсолютная» синонимия в ЕЯ встречается редко: *языкознание* = *лингвистика* = *языковедение*; флексии *-е* и *-ей* с грамматическим значением сравнительная степень прилагательного, находящиеся в отношении свободного чередования: *весел-е* = *весел-ей*.

Как правило, возможность замены знака другим, то есть употребления синонима, ограничена нетождественностью «общего значения», контекстными условиями. Так, например, морфологические синонимы – флексии с «общим значением» мужской род, множественное число, именительный падеж: *-ы*, *-и*, *-е*, *-а*, *-я* – употребляются с существительными различных типов склонения (*лингвист-ы*, *филолог-и*, *южан-е*, *профессор-а*, *кра-я*). К контекстным условиям относятся также лексическая сочетаемость и стилистическая окраска. Например, замена *хочу* на *желаю* во фразе *Я не хочу его видеть* допустима, а во фразе *Кроме нас хотят переселиться в колхоз еще несколько семей* – нет.

Описывая одну и ту же ситуацию, одно обозначаемое с помощью различных синонимичных конструкций (синонимичными, естественно, могут быть и предложения, тексты), носитель языка имеет возможность переструктурировать не только сообщение, но и репрезентуемую им модель ситуации, выделять то одни, то другие компоненты и аспекты.