

# Глава 6

## «Интерфейсы информационного взаимодействия»






# Глава 6

## «Интерфейсы информационного взаимодействия»



### Тема 17. Анализ и синтез естественного языка взаимодействия

1. Структурная схема лингвистического транслятора.
  2. Морфологический анализ (синтез).
  3. Синтаксический анализ (синтез).
  4. Семантическая интерпретация (синтез) и проблемный анализ.
- 




## Интерфейсы информационного взаимодействия

Построение естественно-языкового интерфейса для компьютеров — очень привлекательная цель.

**SHRDLU** (шердлу) — ранняя программа понимания естественного языка, разработанная **Терри Виноградом** в MIT в **1968-1970** годах.

Она была написана на языках Micro Planner и Лисп на компьютере DEC PDP-6 и использовала графический терминал DEC.



Ранние системы, такие как **SHRDLU**, работая с ограниченным «миром кубиков» и используя ограниченный словарный запас, выглядели чрезвычайно хорошо, вдохновляя этим своих создателей. Однако *оптимизм быстро иссяк, когда эти системы столкнулись со сложностью и неоднозначностью реального мира.*

Понимание естественного языка иногда считают **AI-полной** задачей, потому как распознавание живого языка требует огромных знаний системы об окружающем мире и возможности с ним взаимодействовать. Само определение смысла слова «**понимать**» — одна из главных задач искусственного интеллекта.

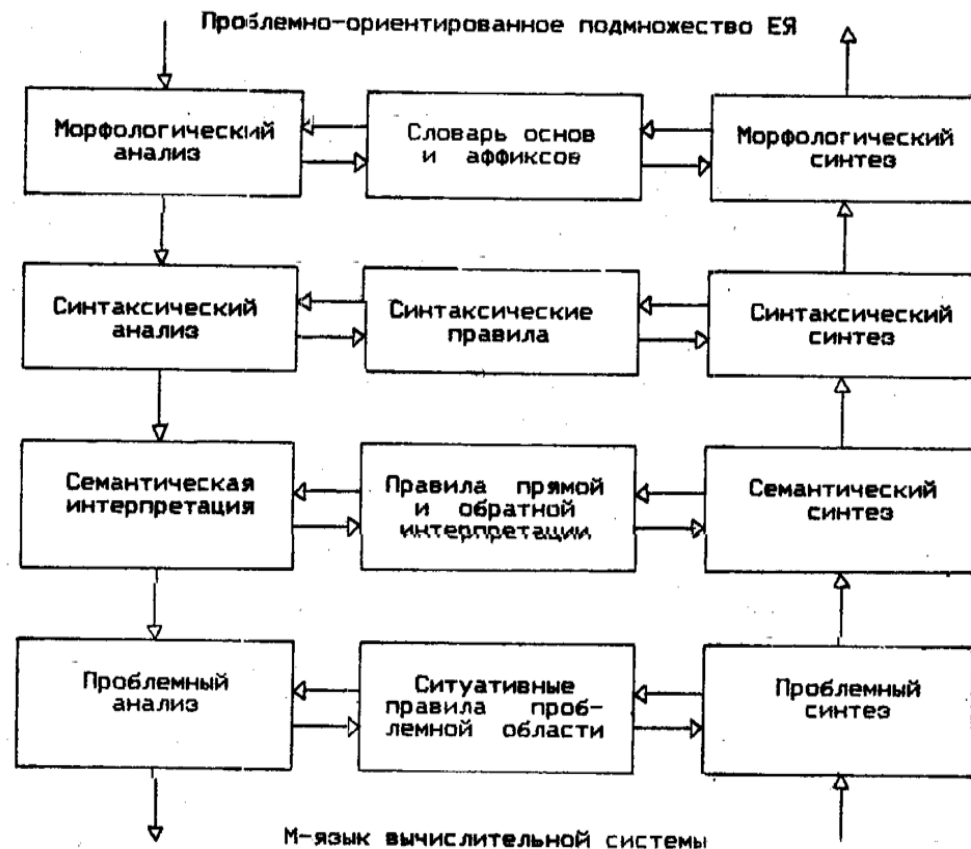
При определении задачи как AI-полной, подразумевается, что она не может быть решена простым алгоритмом.

Например, следующие задачи гипотетически являются AI-полными: компьютерное зрение, понимание естественного языка и пр.

# Структурная схема лингвистического транслатора

Цель анализа предложений естественного языка (ЕЯ) заключается в их переводе на машинный язык (МЯ) вычислительной системы.

Этот процесс можно представить наглядно в виде следующей схемы.



Структурная схема лингвистического транслатора



# Структурная схема лингвистического транслятора



Задача анализа ЕЯ сводится к:

- ☐ распознаванию правильно построенных предложений ЕЯ;
- ☐ исправлению ошибок в ЕЯ-тексте;
- ☐ декомпозиции предложения на фрагменты и построению его синтаксической структуры;
- ☐ семантической интерпретации фрагментов ЕЯ во фрагменты М-языка;
- ☐ композиции фрагментов М-языка в структуру, описывающую ситуацию проблемной среды.





# **Структурная схема лингвистического транслятора**




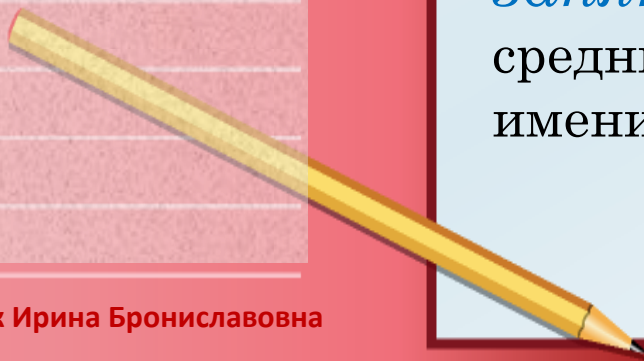
**Задача синтеза ЕЯ заключается в:**

- ☐ определении информации, которую необходимо сообщить пользователю;
- ☐ разбиении текста М-языка на фрагменты, соответствующие будущим фразам;
- ☐ определении лексем для синтезируемой фразы;
- ☐ построении синтаксической структуры фразы;
- ☐ определении порядка слов и осуществлении морфологического синтеза словоформ.



# Структурная схема лингвистического транслятора


## Морфологический анализ (синтез)



Цель морфологического анализа – построение для каждой словоформы предложения списка пар (х,у), где  
х – лексема или основа данной единицы,  
у – подписание значений грамматических признаков, определяющих данную словоформу. (Например, идут занятия):

*идти* – глагол, множественное число, настоящее время;

*занятие* – существительное, средний род, множественное число, именительный падеж.



# Морфологический анализ (синтез)

Существуют два основных метода морфологического анализа:

- ❑ *декларативный*, когда в словаре системы хранятся все возможные словоформы с соответствующим комплексом морфологической информации;
- ❑ *процедурный*, когда наряду со словарем, в котором хранятся только основы слов, система имеет набор правил словообразования, позволяющих анализировать словоформу и приписывать ей соответствующий комплекс информации.



# Морфологический анализ (синтез)

Морфологический синтез предназначен для построения конкретных словоформ ЕЯ по словарю и заданной морфологической информации.

Аналогично анализу морфологический синтез реализуется декларативным и процедурным способами.

*Синтаксический анализ (синтез).*

Задача синтаксического анализа – построение синтаксической структуры предложена на основе морфологической информации и синтаксических правил объединения слов и словосочетаний.

Синтаксическая структура отражает связи, существующие между словами предложения. Известны два основных способа описания синтаксической структуры:

- ☐ система составляющих,
- ☐ дерево зависимостей.

# Способы описания синтаксической структуры

## Система составляющих

Элементы  $S$  называются  
составляющими.

Пусть имеется цепочка  $x$  (произвольная последовательность словоформ) длиной  $w$  (количество словоформ) .

Каждая словоформа цепочки называется *точкой*.

Для любых точек  $a$  и  $b$  цепочки  $x$  таких, что  $a < b$  ( $a$  левее  $b$ ) вводится понятие отрезка, представляющего множество точек  $s$ , удовлетворяющих неравенству  $a < s < b$  .

Множество  $s$  отрезков цепочки  $x$  называется системой составляющих этой цепочки, если:

- множество  $S$  содержит отрезок, состоящий из всех точек цепочки  $x$ , либо все одноточечные отрезки цепочки;
- любые два отрезка из множества  $B$  либо не пересекаются, либо один из них содержится в другом.

Например, для предложения "*лекция проводится в аудитории номер 232 главного корпуса*" допустима следующая система составляющих предложения:

(<sub>1</sub> ЛЕКЦИЯ ПРОВОДИТСЯ (<sub>2</sub> В АУДИТОРИИ 232 (<sub>3</sub> ГЛАВНОГО КОРПУСА ) ) )  
3 2 1

# Способы описания синтаксической структуры

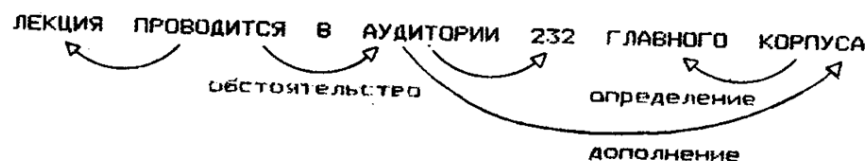
## Дерево зависимостей

Целью синтаксического синтеза является формирование синтаксической структуры фраз и заполнение их соответствующими лексемами.

Пусть  $x$  – произвольная непустая цепочка и  $X$  – множество всех точек  $x$ . Произвольное бинарное отношение  $R$ , определенное на  $X$ , при котором направленный граф  $(X, R)$  является деревом, называется *отношением зависимости*.

Само дерево  $(X, R)$  называется *деревом зависимостей* для  $X$ .

Если между точками  $a$  и  $b$  существует отношение  $a \rightarrow b$ , то точку  $a$  называют *управляющей*, а  $b$  – *подчиненной*.



Заключительным шагом синтеза является приписывание лексемам морфологических характеристик.



# Семантическая интерпретация (синтез) и проблемный анализ

Цель семантической интерпретации – формирование фрагментов на М-языке, соответствующих описанию проблемной ситуации.

Семантический синтез заключается в преобразовании текста М-языка таким образом, чтобы его части могли бы соответствовать будущим фразам и предложениям ЕЯ.

Получение таких фрагментов осуществляется на основе фрагментов ЕЯ-предложения, представленных в синтаксической структуре.

Получение фрагментов М-языка предполагает добавление в них информации, которая выражена неявно в ЕЯ-предложении.

На этапе проблемного анализа множество фрагментов проблемной ситуации структурируется с помощью правил описания ситуаций в проблемной среде.

В результате этого получается структура описания ситуации, заданной ЕЯ-текстом, которая затем подается на вход ВС (*Семантическая сеть предложения*).



Такое преобразование осуществляется за счет фрагментирования текста М-языка.



# Вывод

## Предложения

*«Мы отдали бананы обезьянам, потому что они были голодные»*

*«Мы отдали бананы обезьянам, потому что они были перезрелыми»* похожи по синтаксической структуре.

В одном из них местоимение **они** относится к обезьянам, а в другом — к бананам.

*Правильное понимание зависит от знаний компьютера, какими могут быть бананы и обезьяны.*

По нормам русского языка второе предложение некорректно, потому что в нем местоимение ссылается не на последнее подходящее слово, однако в *живой речи* такое предложение очень даже может встретиться.

# Глава 6

## «Интерфейсы информационного взаимодействия»

Первые компьютеры использовали для управления командный ввод (консоль). Само собой разумеющимся будет то, что для использования программ с таким интерфейсом, требовалась специальная подготовка и определенный уровень знаний.

Позже в **1973 году** в лаборатории Xerox появился первый компьютер использующий графический интерфейс, а еще позже, идея воплотилась в более успешных коммерческих системах Apple и Windows. С тех пор, даже само понятие «интерфейс», у многих ассоциируется именно с графическим интерфейсом пользователя (GUI).

## Тема 18. Классификация пользовательских интерфейсов.

1. Особенности графических и неграфических пользовательских интерфейсов.

# Классификация пользовательских интерфейсов

Существует ряд классификаций пользовательских интерфейсов, которые применяются в индустрии программных средств.

Рассмотрим общую классификацию, применимые в настоящее время.

В общем случае пользовательские интерфейсы можно разделить на две большие группы:

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

Наиболее широкое распространение в настоящее время получили интерфейсы группы WIMP.

Относительно реже используются интерфейсы группы SILK, которые используются в специальных областях.

# WIMP - интерфейсы

## Эволюция пользовательского интерфейса



Рассмотрим классификацию пользовательских интерфейсов, используемую в настоящее время, они разделяются на четыре группы:

- ❑ *неграфический пользовательский интерфейс командной строки (Command Line Interface – CLI);*
- ❑ *пользовательские интерфейсы карманных устройств (Hand User Interface – HUI);*
- ❑ *пользовательский Web-интерфейс (WEB User Interface – WUI);*
- ❑ *графический пользовательский интерфейс (Graphical User Interface – GUI).*



# CLI (Command Line Interface)

## 1. *Неграфический пользовательский интерфейс командной строки*

Данный интерфейс предназначен для подготовленного пользователя, регулярно работающим с приложением.

- ❑ Пользователь взаимодействует с приложением с помощью команд (директив), имеющих определенную жестко заданную внутреннюю структуру.



- ❑ Код команды однозначно идентифицирует функцию приложения, исполняемую по данной команде.
- ❑ Код команды отражает название функции и записывается в виде слова на естественном языке, либо его сокращения или мнемокода.

Команды дают возможность опытному пользователю сразу специфицировать необходимую функцию вместо того, чтобы отвечать на серию запросов системы.

# HUI (Hand User Interface)

## 2. Пользовательский интерфейс карманных устройств



Сегодня широко известны два основных класса PDA (Personal Digital Assistant-персональный цифровой ассистент – "карманный" компьютер, предназначенный, выполнения некоторых специальных функций):

- ❑ в одних используется настоящий GUI-стиль как по внешнему виду, так и по поведению,
- ❑ других применяется подмножество GUI-интерфейса. Для ввода данных пользователем применяется "жестикуляционный" стиль с пером и сенсорным экраном.

Обычно подобные устройства обладают очень маленьким экраном.

Каждая область дисплея PDA меньше, чем большинство окон GUI-ориентированных приложений для настольных и портативных систем.

Для поддержки PDA обычно используется GUI-ориентированное ПО для портативных или настольных компьютеров.

# HUI (Hand User Interface)



HUI-интерфейс обеспечивает некоторые возможности GUI-интерфейса, а именно пиктограммы, меню и аналогичное поведение указателя.

В окне устройства одновременно отображается один объект.

Общий стиль для HUI-интерфейса можно назвать SIMP– стилем (Screen– экран, Icon– пиктограмма, Menu– меню и Pointer– указатель). При этом обеспечиваются многие свойства GUI-интерфейса, некоторые из них приведены ниже.

- ❑ *Пиктограммы* используются во многих PDA, их разрешающая способность изменяется в соответствии с типом устройства отображения. Как и в GUI-интерфейсе, пиктограммы применяются для представления объектов, действий и атрибутов.
- ❑ *Строка меню* и сами *меню* отображаются по требованию и обладают обычным для таких компонентов поведением.
- ❑ *Перо* служит в качестве указателя по большей части для взаимодействия с помощью одного щелчка.
- ❑ *Диалоги* отображаются как окна, которые перекрывают вызывающий объект. Такие окна не обладают стандартным оформлением GUI-окон, их нельзя перемещать и изменять их размер.



# HUI (Hand User Interface)



- ❑ К некоторым PDA можно подключить клавиатуру, однако пользователь должен обучиться ориентированному на указатель взаимодействию и письму с помощью пера для работы непосредственно с PDA.
- ❑ Некоторые команды можно выполнять с помощью "жестикмуляционных" комбинаций клавиш, эквивалентных клавишам быстрого выбора команд GUI-интерфейса.

Основные проблемы проектирования HUI-ориентированных приложений:

- ❑ Упрощение требований к пользователю по вводу данных и взаимодействию.
- ❑ Использование ограниченной области дисплея.



# WUI (Web User Interface)

## 3. Пользовательский Web-интерфейс

В зависимости от структуры гиперссылок приложения навигация в пределах WUI-интерфейса приводит к отображению Web-страниц в иерархии приложения *по одной за раз внутри одного окна.*

Базовый WUI-стиль весьма схож с меню иерархической структуры, которые пользователи знают по опыту работы в средах с неграфическим интерфейсом за исключением более наглядного представления и использования гиперссылок. Необходимая навигация выполняется в рамках одного или нескольких приложений с использованием текстовых или визуальных гиперссылок.



# WUI (Web User Interface)

Основные  
особенности  
приложения,  
использующего  
WUI-стиль:

- ❑ информация обычно отображается в единственном окне, называемом браузером, хотя для представления данных в приложении могут использоваться несколько окон браузеров;
- ❑ браузер обеспечивает меню для Web-приложения;
- ❑ выбор действий ограничен, так как меню, обеспечивающее обращение к функциям, не является легкодоступным для приложения;
- ❑ Web-страница обладает небольшой степенью внутреннего контроля над клиентской областью для открытия специализированных всплывающих меню;
- ❑ создание специализированных меню требует дополнительной работы по программированию;
- ❑ клиентская область не содержит традиционных пиктограмм;
- ❑ многие приложения используют графику и анимацию в эстетических или навигационных целях. Это таит в себе потенциальную угрозу возникновения внешнего визуального шума и увеличения времен отклика при загрузке и раскрытии графических файлов;



# WUI (Web User Interface)

- ☐ браузер и приложения обеспечивают возможности отключения графики, содержащейся в Web-страницах, так что на экране отображается только их текстовая версия;
- ☐ поддержка указателя осуществляется в основном для выбора с помощью одного щелчка мышью или выбора по навигационным ссылкам;
- ☐ технология "draganddrop" ("перетащить и поместить") не поддерживается за исключением случаев специального программирования в определенных средах;
- ☐ ограничены действия 2-й кнопки мыши.


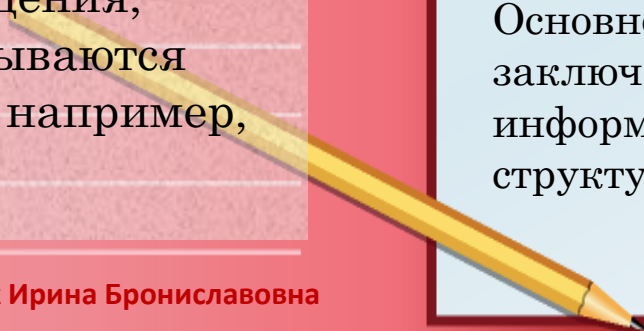




# WUI (Web User Interface)

## Навигация

По сравнению с GUI-ориентированными приложениями WUI-ориентированные приложения включают *несчетное* количество элементов поведения, которые не вызываются пользователем, например, анимационных.




Переход от одной страницы к другой с использованием гиперссылок или поискового механизма – наиболее часто выполняемая функция WUI-интерфейса.

Страницы, с которыми встречается пользователь, существуют в пределах того же самого или другого Web-узла.

Web-браузер обеспечивает базовые возможности навигации для перемещения по Web-узлам и в пределах Web-узлов *линейным способом с помощью кнопок панели инструментов Back (Назад) и Forward (Вперед)*.

Навигация от одной страницы приложения к другой в пределах одного и того же Web-узла приложения выполняется с использованием гиперссылок, схемы Web-узла, кнопок и навигационной панели.

Основное назначение Web-страницы заключается в обеспечении полезной информацией, включая навигационную структуру в организацию Web-узла.





# WUI (Web User Interface)

## *Компоненты WUI-интерфейса*

К наиболее распространенным компонентам WUI-интерфейса относятся:

- ☐ баннеры (заголовки),
- ☐ навигационные панели
- ☐ визуальные или текстовые гиперссылки, упорядоченные различными способами.

Также применяются разнообразные подходы к использованию графики, анимации и цвета.



# Компоненты WUI- интерфейса

**Баннер** представляет собой визуальный заголовок, отображаемый сверху Web-страницы.

**Навигационная панель** – это список вариантов выбора гиперссылок, обеспечивающих доступ к информации.

**Гиперссылка** представляет собой вариант выбора, который отображает следующую страницу информации или перемещает фокус отображения на другую область той же страницы;


**Браузер** – обладает заголовком, навигационной панелью областью, отображаемой в пределах экрана.

**Каталог** представляет собой визуальный поисковый механизм, в котором перечислены варианты выбора гиперссылок, используемых для навигации по дополнительным вариантам выбора до тех пор, пока не будет найден искомый результат, допускаются навигационные панели в виде заголовков другие типы навигации по вариантам выбора гиперссылок.



# WUI (WebUser- Interface)

## *Компоненты WUI-интерфейса*



**Поиск и результаты поиска** – один или несколько элементов управления, с помощью которых пользователь осуществляет ввод или выбор критерия поиска информации, результаты поиска отображаются в том же и другом окне Web-браузера.


**Документ** – во многом похожий на свой бумажный двойник Web-документ отображает текстовую информацию вместе со ссылками на дополнительные источники или развернутое представление информации.

**Записная книжка** – некоторые Web-узлы представляют визуальную записную книжку в качестве метафоры для организации данных. Она почти не отличается от навигационной панели, с той лишь разницей, что содержит меньшее количество вариантов выбора.





# WUI (Web User Interface)



Факторы успеха, которые, влияют на уровень практичности приложений, использующих WUI-интерфейс – это:

- ☐ простота навигации по иерархическим информационным структурам;
- ☐ легкость и быстрота поиска.

К другим важным факторам относятся:

- ☐ эстетические характеристики;
- ☐ ценность текущего содержания информации.



# GUI (Graphical User Interface)

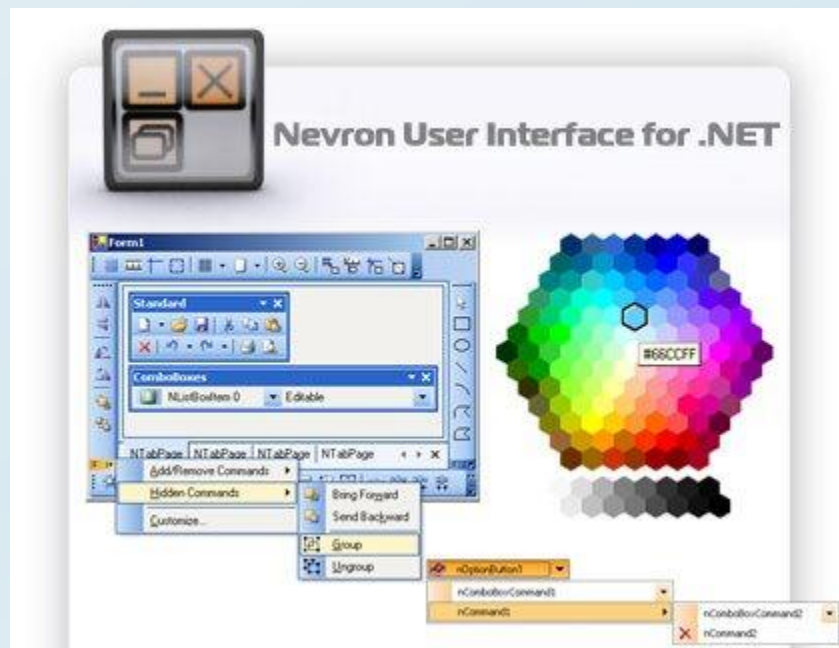
## 4. Графический пользовательский интерфейс

Важнейшие свойства GUI-интерфейса – это возможность непосредственного манипулирования, поддержка мыши или указателя, использование графики и наличие области для функций и данных приложения.

Графический пользовательский интерфейс определяется как стиль взаимодействия "пользователь-компьютер", в котором применяются такие основные элементы:

- ☐ окна,
- ☐ пиктограммы,
- ☐ меню,
- ☐ указатели.

Иногда GUI-интерфейс называют **WIMP-интерфейсом**.



# Компоненты GUI

## Окно

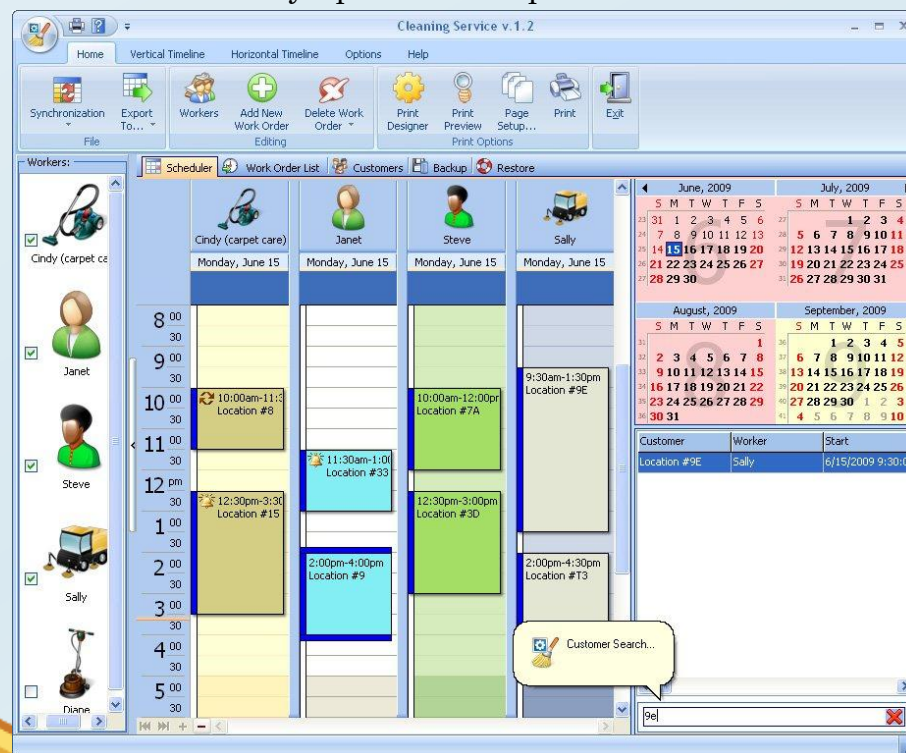
Частичное использование устройства отображения позволяет просматривать несколько окон для одновременного взаимодействия с несколькими объектами или управляющими диалогами. Определение окна также подразумевает использование графики или визуализации вместо текстовой информации для указания доступного объема информации (например, использование полосы прокрутки вместо указания типа "1-я строка из 45-ти").

**Окно** – это область устройства отображения, используемая для представления и взаимодействия с объектами, информацией об объектах, или для выполнения действий, применяемых к объекту.

Окно обладает строкой заголовка, набором операций перемещения, изменения размера, набором меню и областью для отображения информации об объектах.

Обычно окно представляет собой прямоугольник. При котором приложение, использующее окно, является GUI-ориентированным приложением.

Окно отображает информацию только на определенную часть или область устройства отображения.





# Компоненты GUI

## Пиктограммы

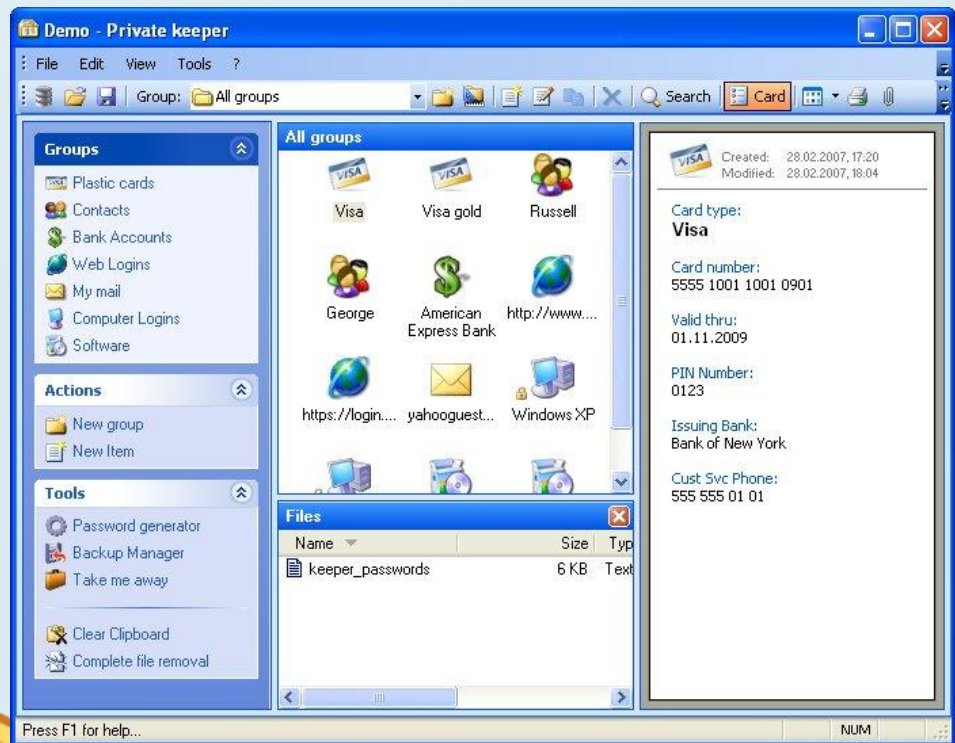
Существует множество графических символов, применяемых в GUI-интерфейсе, которые формально не являются пиктограммами.

Графические символы, используемые для представления действий, атрибутов объектов и состояния могут восприниматься конечными пользователями как пиктограммы, однако с точки зрения GUI-интерфейса и разработчиков стандартов их следует рассматривать как графические кнопки. Для подобных случаев использования графических символов термины "пиктограмма" и "графика" взаимозаменяемы.

Пиктограмма во многих отношениях похожа на окно, хотя согласно формальному определению **пиктограмма** – это область устройства отображения, используемая для наглядного представления объекта.

Типичные свойства пиктограммы включают графический символ для представления объекта, заголовок или имя, а также операции непосредственного манипулирования.

Наиболее важная операция которую выполняют над пиктограммой, представляющей объект – это операция Open (Открыть) для отображения окна, содержащего детализированную информацию о объекте.



# Компоненты GUI

## Меню

Меню отображает набор альтернатив, с помощью которых пользователь, может осуществить выбор. Обычно альтернативы GUI-ориентированного меню представляют собой имена выбираемых пользователями команд для выполнения действий над объектами.

*Меню* содержат в себе полный набор пользовательских команд. Системы, отличные от графических, напротив, требуют, чтобы под меню использовался весь дисплей, при этом меню строятся иерархическим способом.

Существует несколько типов меню:

- ☐ строки меню,
- ☐ выпадающие,
- ☐ всплывающие
- ☐ каскадные.

Каковы бы ни были их цель и назначение, компоненты наподобие панели инструментов, представленных пиктограммами, являются меню.





# Компоненты GUI

## Указатели

Графические системы обычно содержат координатно-указательные устройства в виде мыши или шарового манипулятора.

С координатно-указательным устройством ассоциируется определенное место на экране, куда пользователь может осуществить ввод с помощью этого устройства.

**Указатель** – это графический символ, визуально показывающий местоположение входа в систему для координатно-указательного устройства.

Указатели, используемые в GUI интерфейсе, включают системный указатель в виде стрелки, графическое перекрестие и I-образный или "балочный" указатель (указатель в форме двутавровой балки).

Во многих отношениях указатель аналогичен курсору, определяющему место вставки вводимых с клавиатуры символов на экране устройства отображения.



# Свойства GUI- интерфейса

## Непосредственное манипулирование

Многие действия, выполняемые с помощью выбора альтернатив или меню, можно произвести, воспользовавшись непосредственным манипулированием.

Например, во многих системах результатом перетаскивания пиктограммы документа на пиктограмму принтера на рабочем столе является печать документа.

Наиболее значительное свойство GUI-интерфейса заключается в непосредственном манипулировании, которое позволяет пользователю взаимодействовать с объектами с помощью указателя.


Например, окно можно переместить по экрану с помощью мыши, установив указатель на строку заголовка окна, нажав и удерживая кнопку мыши и перемещая мышь (иногда эту операцию называют "захватить и перетащить" – "grabanddrag").



К другим действиям, которые выполняются с помощью непосредственной манипуляции, относятся такие операции, как Move (Переместить), Copy (Копировать), Delete (Удалить) и Link (Связать).



# Свойства GUI- интерфейса




К некоторым другим методам работы, присущим GUI-интерфейсу, относятся:

- ☐ буфер обмена,
- ☐ комбинации клавиш,
- ☐ ускоряющие клавиши в меню и диалогах,
- ☐ дополнительные возможности взаимодействия мышь-клавиатура.

Несмотря на свою полезность, эти механизмы не рассматриваются как существенные свойства GUI-интерфейса.

GUI-интерфейс не гарантирует более высокого уровня практичности, однако надлежащим образом спроектированное GUI-ориентированное программное приложение может превосходить его аналог с неграфическим интерфейсом с точки зрения эффективности работы пользователя и степени его удовлетворенности (при условии верной постановки задач и соответствующем уровне навыков пользователя).





# NUI (Natural user interface)



Многие UI/UX дизайнеры знакомы с **NUI (Natural user interface)**.

Многие просто видели reacTable, музыкальный инструмент.

Строго говоря, reacTable не является идеальным примером устройства с интерфейсом NUI, потому что для его управления используются материальные предметы, но увидев его можно наглядно представить, какими интерфейсы будут в ближайшем будущем.

Мы не будем следовать точной терминологии, а просто *помечаем* и подумаем какие способы взаимодействия NUI доступны нам уже сейчас.

Отличительной особенностью **NUI** является интерпретация опыта полученного еще в реальном мире.

Например с рождения мы понимаем как перемещать предметы в пространстве, точно также мы можем передвигать объекты в виртуальной среде.



# NUI (Natural User Interface)

Все те же лидеры отрасли, что и 20 лет назад, уделяют много внимания NUI.

Microsoft создали Kinect (ранее Project Natal) — игровой «контроллер без контроллера», и Surface (сенсорный «рабочий стол»).

Apple совершили революцию с помощью iPad (интернет-планшет).


Такая технология как дополненная реальность еще в большей мере способствует развитию NUI.

NUI — это интуитивно понятный интерфейс. Теоретически, человек который впервые в жизни увидел компьютер, с таким интерфейсом, способен решать на нем определенные задачи. Многие приложения с тактильным интерфейсом во многом приближены к идеальному NUI. Это соображение по большей части касается приложений разработанных для детей, так как в них, мы не можем в полной мере использовать приобретенный опыт пользователя. **Для детей нам приходится разрабатывать интерфейсы понятные быстро и сразу.**





# Вывод



Мы не знаем еще пока как создавать интуитивно понятные интерфейсы, для таких специальных программ как например Photoshop или Eclipse.


***Но пока это и не надо, кажется наше поколение никогда не сможет избавиться от GUI, настолько прочно оно засело в нашем сознании.***

*А вот то поколение которое подрастает, наверно с недоумением будет смотреть на существующие сейчас способы взаимодействия, если вообще узнают о них когда-нибудь.*




# Глава 6

## «Интерфейсы информационного взаимодействия»



### Тема 19. Интерфейсы информационного взаимодействия компьютерных иерархических систем.


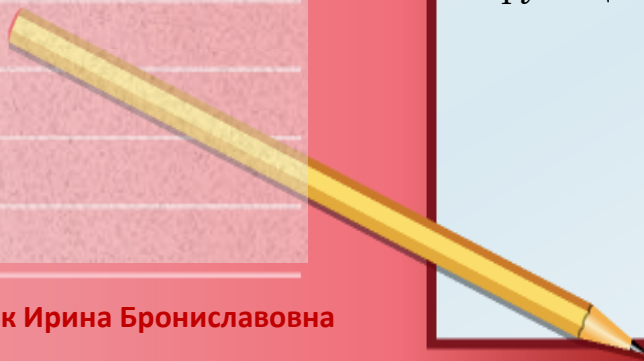
1. Система уровней  
информационного  
взаимодействия.
  2. Интерфейсы и протоколы  
взаимодействия уровней  
информационного  
взаимодействия.
- 





# Интерфейс информационного взаимодействия

Рассмотрим интерфейс информационного взаимодействия на примере WEB-приложений, отметим, что рассмотренный ранее пользовательский WUI непосредственно связан с программным интерфейсом уровня приложений.




Взаимодействие различных объектов в сети Интернет строится в соответствии с правилами и требованиями общего международного стандарта **ISO 7498** (ISO – International Organization for Standardization).

Этот стандарт имеет тройной заголовок "Информационно-вычислительные системы – Взаимодействие открытых систем – Эталонная модель".

Обычно его называют короче – "Эталонная модель взаимодействия открытых систем".


Публикация этого стандарта в **1983** году подвела итог многолетней работы многих известных телекоммуникационных компаний и стандартизирующих организаций.

Основной идеей, которая положена в основу этого документа, является разбиение процесса информационного взаимодействия между системами на уровни с четко разграниченными функциями.





# Интерфейс информационного взаимодействия



В качестве *прообраза модели взаимодействия OSI* (Open System Interconnection) была использована структура, предложенная ANSI (American National Standards Institute).

**ISO 7498** является стандартом в области телекоммуникаций.

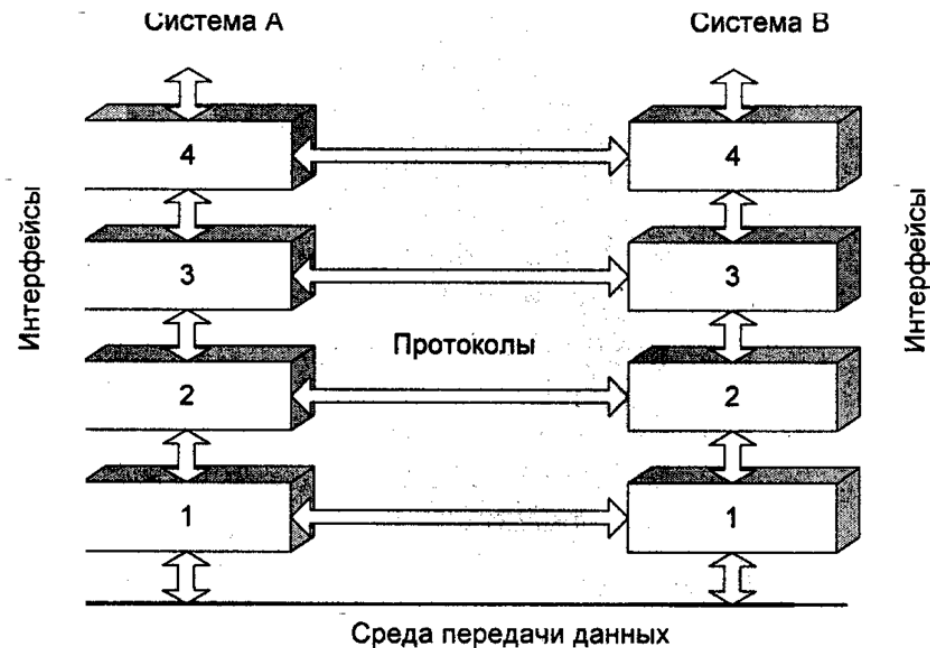
Преимущества слоистой организации взаимодействия заключаются в том, что она обеспечивает независимую разработку уровней стандартов, модульность аппаратуры и программного обеспечения информационно-вычислительных систем и способствует тем самым техническому прогрессу в данной области.

При использовании многоуровневой модели проблема перемещения информации между узлами сети разбивается на более мелкие и, следовательно, более легко разрешимые проблемы.

Многоуровневая модель четко описывает, каким образом информация проделывает путь через среду сети от одной прикладной программы, к примеру, обработки таблиц, до иной прикладной программы обработки тех же таблиц, находящейся на другом компьютере сети.

# Многоуровневая модель информационного взаимодействия

Перемещения  
информации между  
узлами сети



Предположим, например, что система А, изображенная на рисунке, имеет информацию для отправки в систему В. Прикладная программа системы А начинает взаимодействовать с уровнем 4 системы А (верхний уровень), который, в свою очередь, начинает взаимодействовать с уровнем 3 системы А, и т. д. - до уровня 1 системы А.

Задача уровня 1 забирать информацию из физической среды сети, отдавать, а потом после того как информация проходит через физическую среду сети и поступает в систему В, она последовательно обрабатывается на каждом уровне системы В в обратном порядке – сначала на уровне 1, затем на уровне 2 и т. д., пока, наконец, не достигнет прикладной программы системы В.



# Многоуровневая модель информационного взаимодействия

Информация по оказываемым услугам передается между уровнями в специальном информационном блоке, который называется *заголовком*. Заголовок обычно предшествует передаваемой прикладной информации.

Многоуровневая модель не предполагает наличия непосредственной связи между одноименными уровнями взаимодействующих систем.

Следовательно, каждый уровень системы А должен полагаться на услуги, предоставляемые ему смежными уровнями системы А, чтобы помочь осуществить связь с соответствующим уровнем системы В. Предположим, что уровень 4 системы А должен связаться с уровнем 4 системы В. Для того чтобы выполнить эту задачу, уровень 4 системы А должен воспользоваться услугами уровня 3 системы А, тогда уровень 4 будет называться *"пользователем услуг"*, а уровень 3 – *"источником услуг"*.

# Многоуровневая модель информационного взаимодействия

Предположим, что система А хочет отправить в систему В какой-либо текст, называемый "данные" или "информация". Этот текст передается из прикладной программы системы А в верхний уровень этой системы. Прикладной уровень системы А должен передать определенную информацию в прикладной уровень системы В, поэтому он помещает управляющую информацию своего уровня в виде заголовка перед фактическим текстом, который должен быть передан. Построенный таким образом информационный блок передается в уровень 3 системы А, который может предварить его своей собственной управляющей информацией, и т. д.

# **Многоуровневая модель информационного взаимодействия**

Размеры сообщения увеличиваются по мере того, как оно проходит вниз через уровни до тех пор, пока не достигнет сети, где оригинальный текст и вся связанная с ним управляющая информация перемещаются в систему В и поглощаются уровнем 1 системы В.

Уровень 1 системы В отделяет от поступившей информации и обрабатывает заголовок уровня 1, после чего он определяет, как обрабатывать поступивший информационный блок. Уменьшенный в размерах информационный блок передается в уровень 2, который отделяет заголовок этого же уровня, анализирует его, чтобы узнать о действиях, которые он должен выполнить и т. д.

Когда информационный блок наконец доходит до прикладной программы системы В, он должен содержать только оригинальный текст.

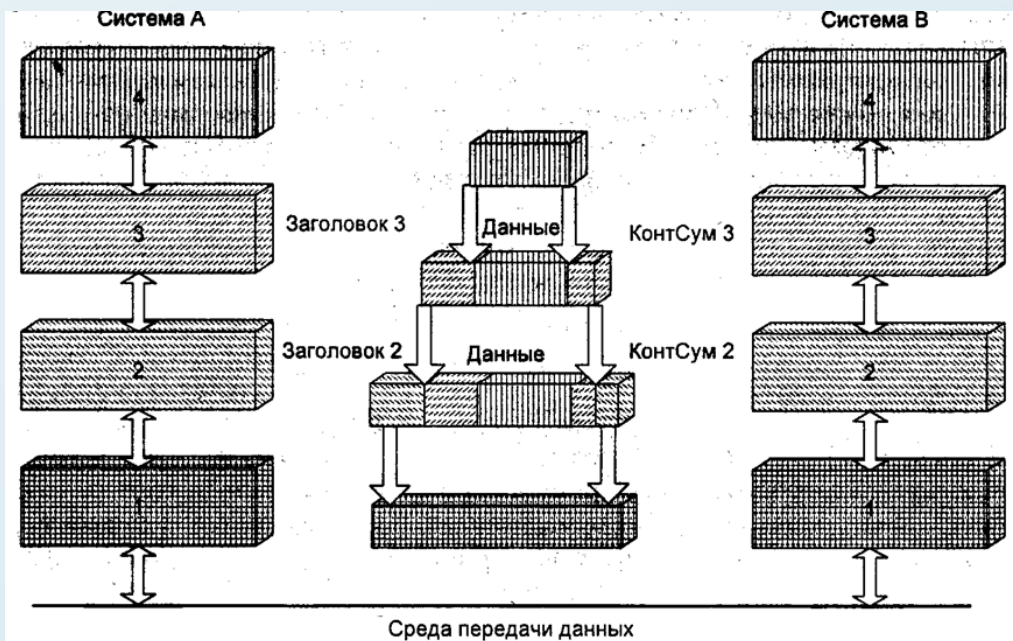


# Многоуровневая модель информационного взаимодействия

## Инкапсуляция блоков данных различных уровней

Данная модель напоминает собой вложенные друг в друга матрешки. Самая маленькая из них — это и есть пользовательские данные, а все остальные служат для доставки данных в точку назначения.

Иными словами, в результате работы этого механизма каждый пакет более высокого уровня вкладывается в "конверт" протокола нижнего уровня.



Структура заголовка и собственно данных относительно и зависит от уровня, который в данный момент анализирует информационный блок. Например, на уровне 2 информационный блок состоит из заголовка этого же уровня и следующих за ним данных. Однако данные уровня 2 могут содержать заголовки уровней 3 и 4.

Кроме того, заголовок уровня 2 является просто данными для уровня 1.

Помимо заголовка на каждом уровне системы информационный блок завершается соответствующей контрольной суммой - КонтСум.

# Модель ISO

В соответствии с **ISO 7498** выделяются семь уровней (слоев) информационного взаимодействия:

Информационное взаимодействие двух или более систем, таким образом, представляет собой совокупность информационных взаимодействий уровней подсистем, причем каждый слой локальной информационной системы взаимодействует только с соответствующим слоем удаленной системы.

Тип данных	Уровень	Функции
Данные	7. Уровень приложения или прикладной (Application Layer)	Доступ к сетевым службам
	6. Уровень представления (Presentation Layer)	Представление и кодирование данных
	5. Уровень сессии или сеансовый (Session Layer)	Управление сеансом связи
Сегменты	4. Транспортный уровень (Transport Layer)	Прямая связь между конечными пунктами и надежность
Пакеты	3. Сетевой уровень (Network Layer)	Определение маршрута и логическая адресация
Кадры	2. Канальный уровень (Data Link Layer)	Физическая адресация
Биты	1. Физический уровень (Physical Layer)	Работа со средой передачи, сигналами и двоичными данными

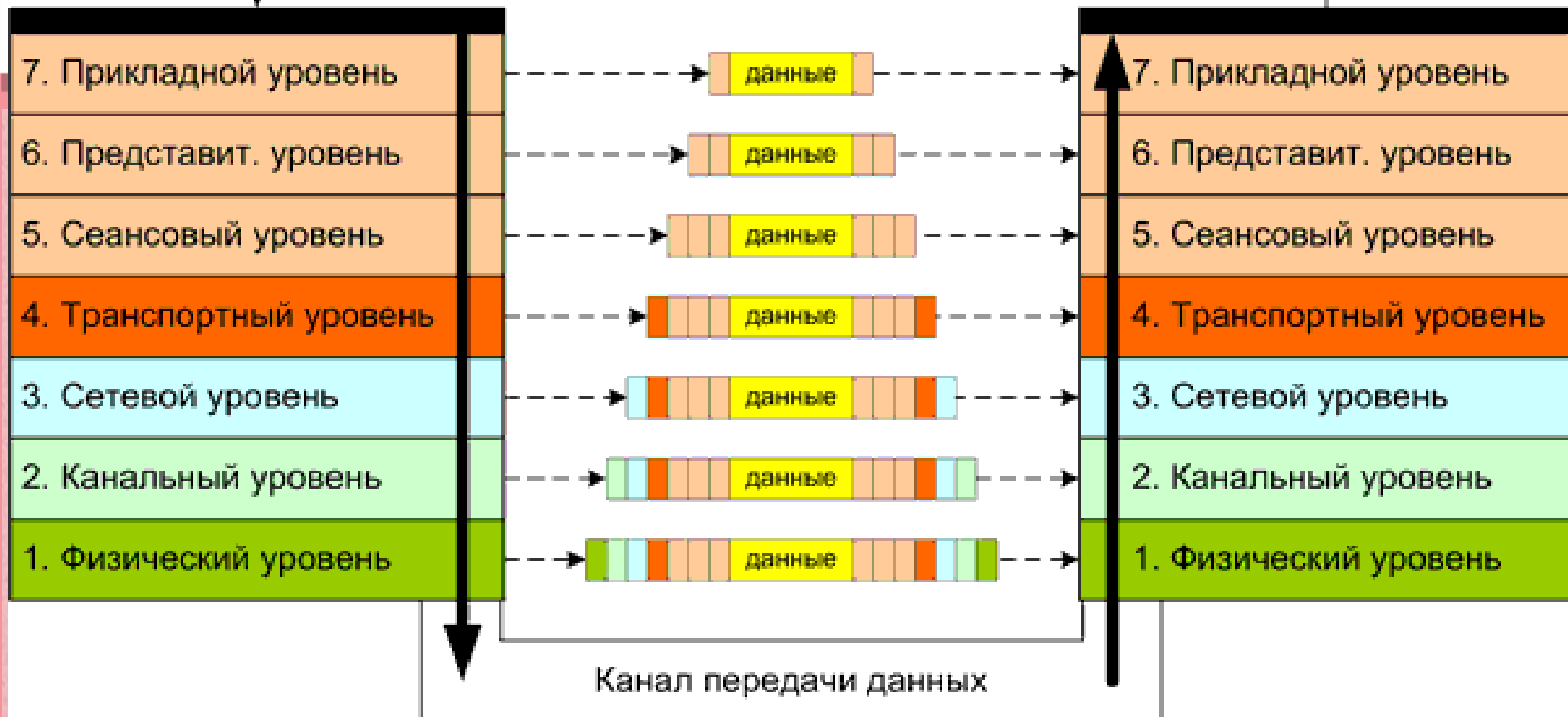
Система 1

Система 2

Приложение 1

данные

Приложение 2



----- Логическое соединение между уровнями

————— Реализация передачи данных



# Модель ISO

TCP/IP (*Transmission  
Control Protocol/Internet  
Protocol*)

**Протокол** – набор алгоритмов (правил) взаимодействия объектов одноименных уровней.

Слои (уровни) одной информационной системы также взаимодействуют друг с другом, причем в непосредственном взаимодействии участвуют только соседние уровни. Как правило, средний уровень пользуется услугами, которые ему предоставляет нижний уровень, а сам, в свою очередь, предоставляет услуги для верхнего уровня.

**Интерфейс** – совокупность правил, в соответствии с которыми осуществляется взаимодействие с объектом данного уровня.


Иерархическая организация сетевого взаимодействия позволяет обеспечивать преемственность разработанных структур и их быструю адаптацию к изменениям, происходящим в технологиях передачи данных. Например, при переходе на новый способ передачи данных по физическому носителю, изменения коснутся только нижних уровней и совсем не затронут верхние в том случае, если система протоколов организована в соответствии с требованиями ISO 7498. На практике требования данного стандарта реализуются в виде стека протоколов.

**Стек** – иерархически организованную группу взаимодействующих протоколов

Протоколы, которые входят в стек, имеют специализированный интерфейс и предназначены для взаимодействия только с протоколами соответствующих уровней данного стека. В качестве примеров таких стеков можно привести стек TCP/IP.



# Модель ISO



Уровни 7-5 считаются верхними и, как правило, не отражают специфики конкретной сети. Блок данных пользователя (сообщение) этими уровнями рассматривается как единое целое. Изменения могут испытывать только сами данные.

Уровни 1-3 и иногда 4 считаются нижними уровнями OSI. На каждом из этих уровней определяется свой формат представления данных. При прохождении по стеку с 4-го уровня до первого сообщение пользователя последовательно фрагментируется и преобразуется в последовательность блоков данных соответствующего уровня.

**Инкапсуляция** – процесс помещения фрагментированных блоков данных одного уровня в блоки данных другого уровня.

Обычно инкапсулируются данные протоколов верхних уровней в блоки данных протоколов нижних уровней (сетевой - канальный), но также может выполняться инкапсуляция для протоколов одноименных уровней.