



http://

@

www

internet

Тенденции развития компьютерных технологий

Кравченко Ю.А.

Разделы дисциплины

Основные этапы, тенденции и риски развития компьютерных технологий

Проблемы и тенденции развития программного обеспечения и вычислительной техники

Перспективы развития средств интеллектуального анализа данных и управления знаниями, интегрированными из разных предметных областей

Проблемы компьютерного моделирования сложных систем

Развитие технологий проектирования информационных, автоматизированных и автоматических систем

Направления развития систем поддержки жизненного цикла наукоёмкой продукции



http://

@

www

internet

1. Основные этапы, тенденции и риски развития компьютерных технологий

средства создания и использования информационных ресурсов

1

■ научная методология, используемая в информационной сфере общества;

2

■ программно-аппаратные средства информатизации;

3

■ современные информационные технологии.

Развитие глобального процесса информатизации

1

Формирование информационной технологии, как самостоятельной научной дисциплины о методах создания высокоэффективных информационных технологии

Объект исследования и предметная область

1

Объектом исследований информационной технологии, как научной дисциплины, являются информационные технологии, т.е. способы рациональной организации информационных процессов

2

Предметом исследований должны стать теоретические основы и методы создания информационных технологий, а также их проектирование и эффективная реализация

Цикл формирования этого нового научного направления

классификация различных
видов информационных
технологий



разработка
критериев для
их
сравнительного
анализа и
количественной
оценки
эффективности

создание
методов
синтеза
высоко-
эффективных
технологий

Предметная область НИТ

ЗАДАЧИ

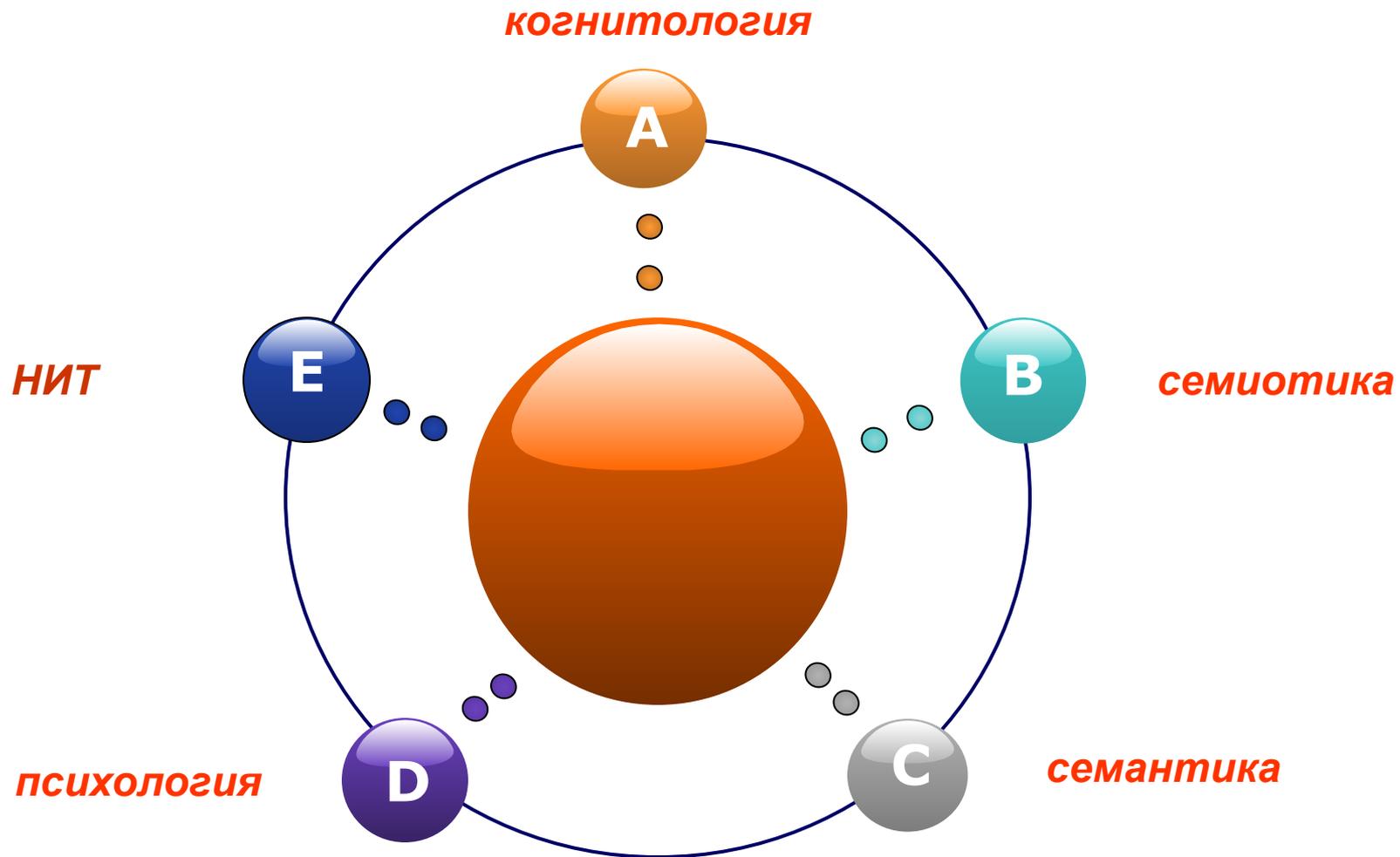
разработка методов структуризации и классификации Информационных технологий различного вида и назначения по их характерным признакам;

разработка критериев эффективности информационных технологий, методов их оптимизации и сравнительной количественной оценки;

определение перспективных направлений развития информационных технологий, а также научных методов, которые должны лежать в их основе;

определение принципов построения перспективных средств для реализации высокоэффективных информационных технологий нового поколения.

Креативные технологии



Новая информационная технология

Технология при переводе с греческого означает **искусство, мастерство, умение**, а это **процессы**. Под *процессом* следует понимать определенную **совокупность действий, направленных на достижение поставленной цели.**

Новая информационная технология

Информационная технология (ИТ) - **процесс**, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) **для получения информации нового качества** о состоянии объекта, процесса или явления (информационного продукта).

Новая информационная технология

Информационная система (ИС) -
взаимосвязанная **совокупность**
средств, методов и персонала,
используемых для хранения,
обработки и выдачи информации в
интересах достижения поставленной
цели.

Новая информационная технология

В основу концепции **новой ИТ**, базирующейся на широком применении компьютерной техники, положены три основных принципа: **интегрированность, гибкость, интерактивность.**

Характеристики НИТ

работа пользователя в режиме манипулирования данными

сквозная информационная поддержка на всех этапах прохождения информации на основе интегрированной базы данных

безбумажный процесс обработки документа

интерактивный (диалоговый) режим решения задачи

с широкими возможностями для пользователя

возможность коллективного исполнения документов на основе группы ПЭВМ, объединенных средствами коммуникации

возможность адаптивной перестройки форм и способа представления информации в процесс решения задачи

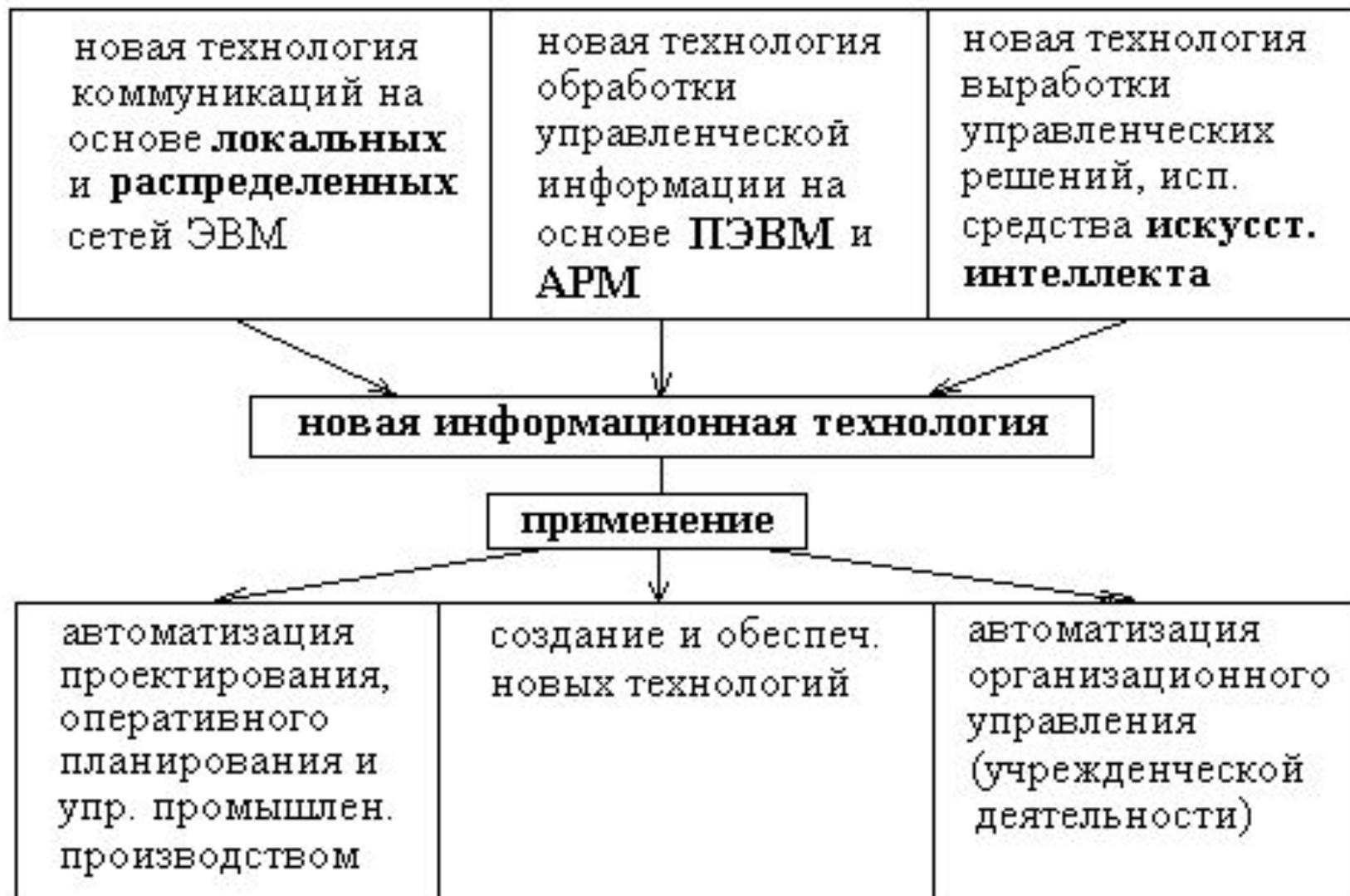
Два способа внедрения НИТ



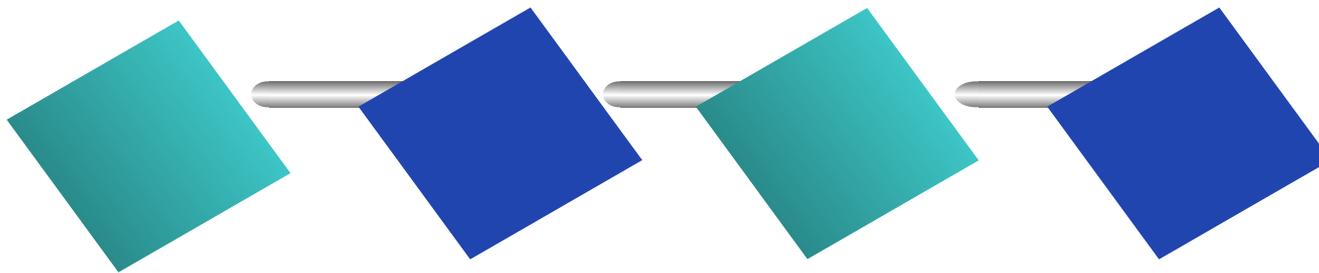
Области применения НИТ



Составные части и области применения НИТ



Особенности НИТ



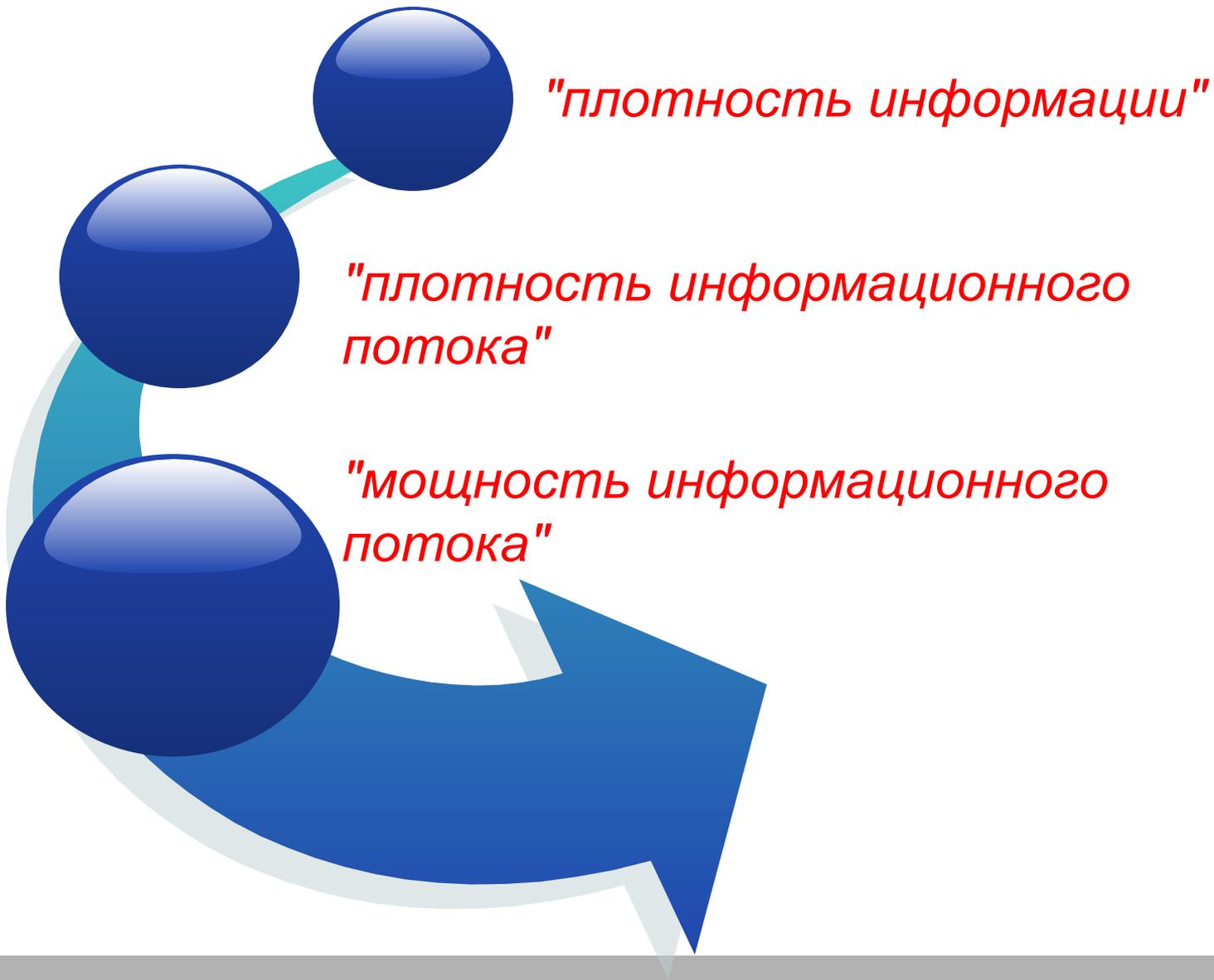
динамичность
(технология, поколения технических и программных средств изменяются дважды за 5 лет)

регулярное **повышение квалификации** разработчиков и пользователей информационных систем

глубокое и долговременное влияние на **развитие производительных сил и производственных отношений**

высокая степень потенциальной эффективности в условиях: стандартизации, масштабируемости внедрения, своевременного внедрения новых средств и методов НИТ

Новые понятия ИТ



Проблема семантического сжатия информации

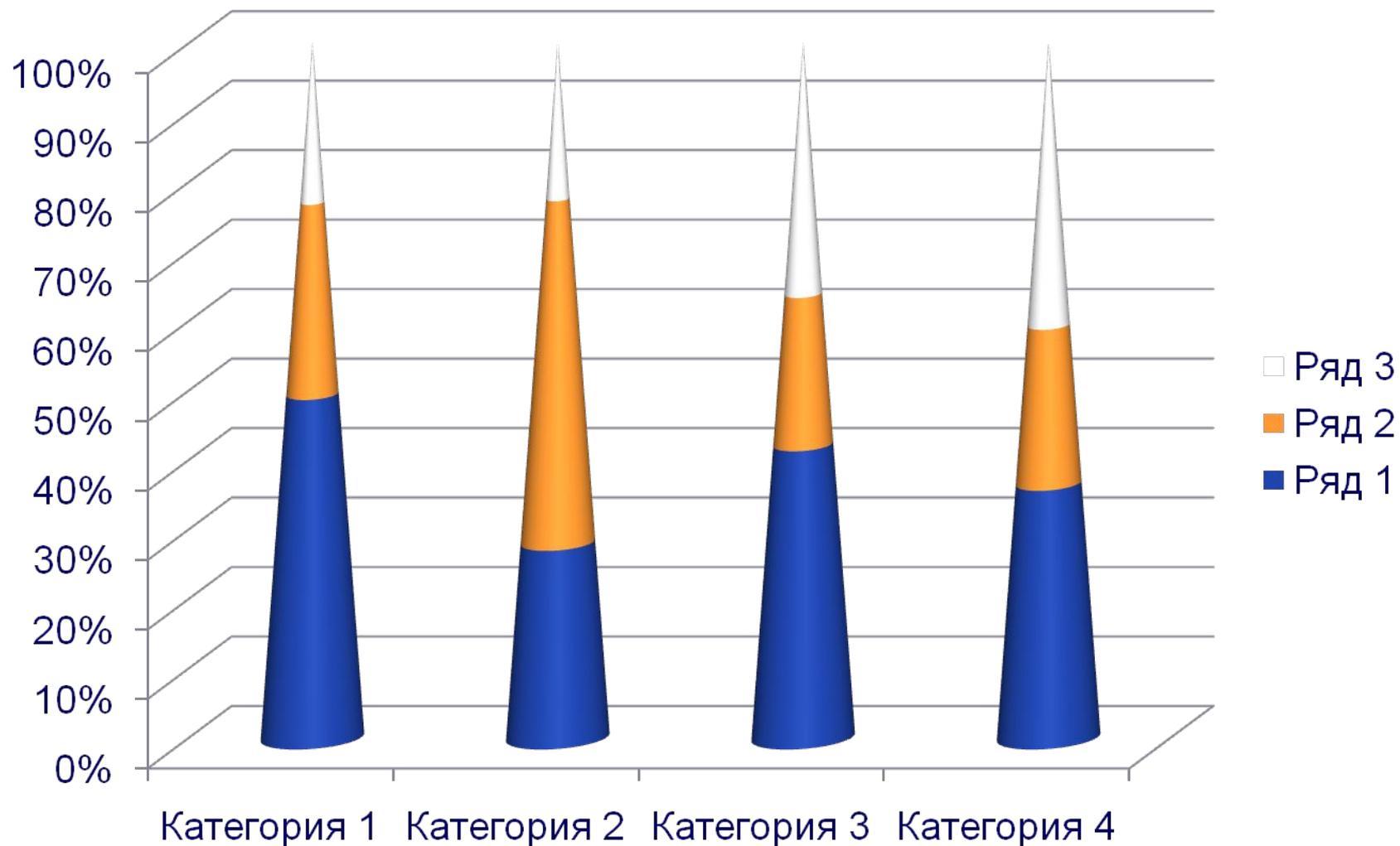


*Целевая
функция*

Сегмент 1

Сегмент 1n

Информационные конусы



Методологический аппарат науки как информационная технология

Изложенные выше подходы к рассмотрению основных проблем информационной технологии как науки позволяют рассматривать и методологию науки как своеобразную информационную технологию достаточно высокого уровня.

Методологический аппарат науки как информационная технология

Формирование информационной технологии как самостоятельного научного направления может оказаться весьма полезным для развития и самой науки в части дальнейшего совершенствования ее методологического аппарата.

Эволюция информационных технологий

1

- 1951 год – программируемый микропроцессор;

2

- 1969 год Intel Corporation – универсальный многоцелевой процессор;

3

- 1973 год – Bob Metcalfe создал Ethernet.

Эволюция информационных технологий

4

- 1975 год – первый серийный персональный компьютер;

5

- 1976 год Wang Laboratories – текстовый редактор Word Processing;

6

- 1978 год – VisiCalc первая программа для работы с таблицами.

Эволюция информационных технологий

7

- 1979 год – первый текстовый редактор для ПК WordStar;

8

- 1979 год – Oracle;

9

- 1982 год – TCP/IP.

Эволюция информационных технологий

10

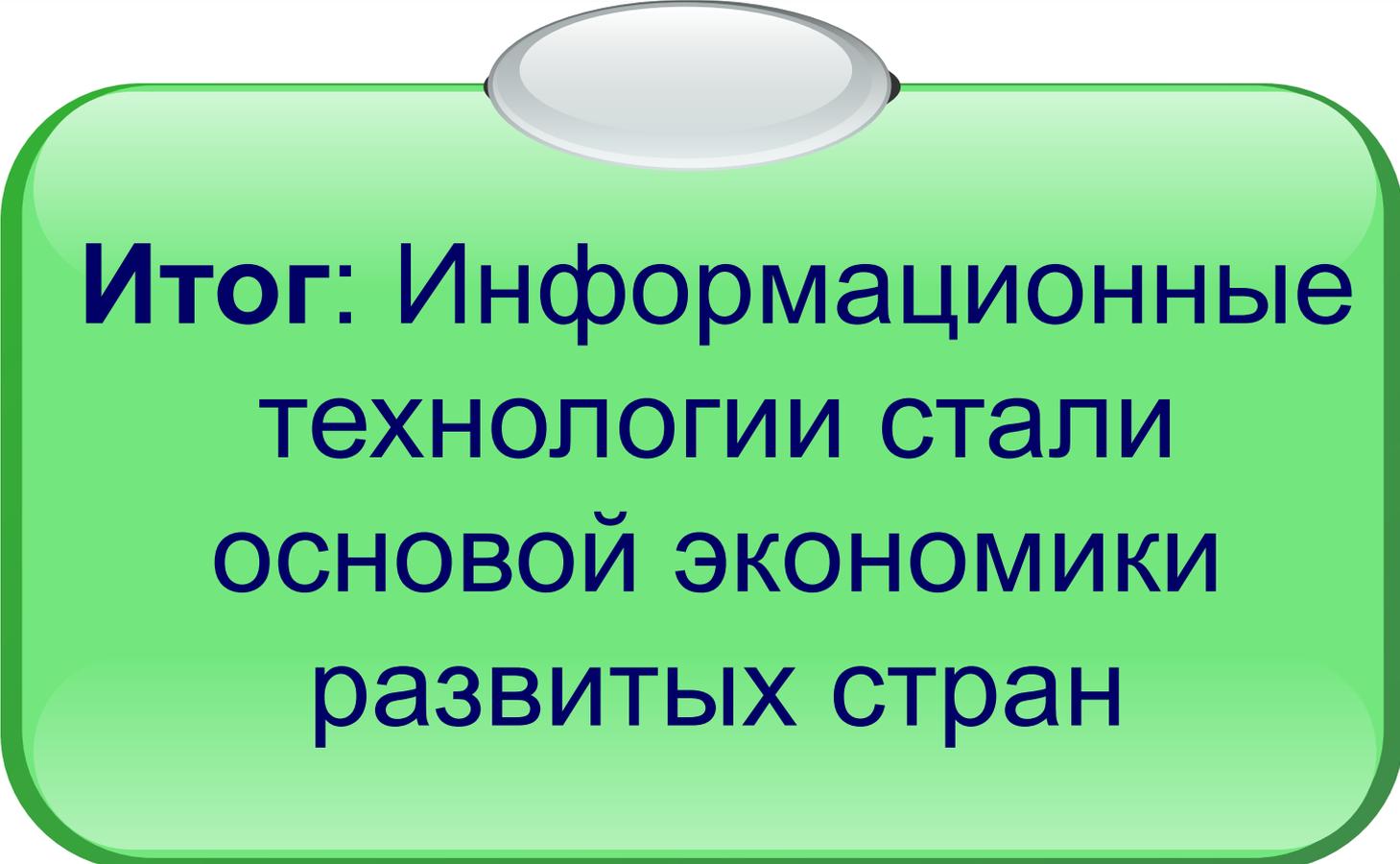
- 1984 год – Macintosh и первый настольный лазерный принтер;

1
1

- 1989 год – Электронная почта;

12

- 1990 год – Tim Berners-Lee (World Wide Web).



Итог: Информационные технологии стали основой экономики развитых стран

Переворот в сознаний (IT – commodity input)

Ежегодные затраты компаний во всем мире на аппаратные средства, ПО, коммуникации и обслуживание – более **2 трлн. долл (более 50%)**.
Ошибочное мнение – по мере развития и роста доступности ИТ их стратегическая значимость также возрастает!?

На самом деле – вы имеете преимущество перед конкурентами только в том случае, если обладаете чем-то, чего у них нет, или делаете что-то, чего они сделать не могут!!!

Стратегический подход

Дифференциация – главная цель и показатель эффективности любого стратегического планирования.

Функциональная и структурная дифференциации могут дать возможность компаниям в сфере IT уйти от губительных последствий ценовой конкуренции.

Принципиально важно отличать товарные общедоступные ресурсы (commodity input**) от ресурсов, способных создать преимущества**

Взгляд в будущее и прошлое



Превращение IT из источника конкурентных преимуществ в рядовую статью затрат ставит перед руководителями компаний ряд проблем.

Снижение рисков становится важнее инноваций, а сокращение затрат – важнее новых инвестиций!

IT проще понять, если рассматривать их в качестве «последнего звена» в ряду технологий, получивших широкое распространение и изменивших экономику за последние два столетия, которые по мере распространения превращались в обычный товарный ресурс.

Анализ современного состояния информационных систем (ИС)

Тенденции развития ИС

1

Появление у систем так называемых интеллектуальных (семантических) свойств

2

Способность систем к самоорганизации, саморазвитию (аналогия с организмами живой природы)

World Wide Web Consortium (W3C) стандарты

Extensible Markup Language (XML)
Язык построения
структурированных документов

**Resource Description Framework
(RDF)**
Система описания ресурсов

Ontology Web Language (OWL)
Язык описания онтологий

**Основные
стандарты
организации
интеллектуальной
сети
(структурирования)
информации**



Данные стандарты устанавливают механизм структурирования информации в сети Интернет таким образом, что её восприятие становится доступным программам, способным выполнять логический анализ и генерировать необходимые выводы.

Применение эволюционного моделирования для создания ИС



В основе второй тенденции создания глобальной сетевой структуры (саморазвивающейся искусственной интеллектуальной системы) лежит использование концепции **эволюционного моделирования (ЭМ)**.

ЭМ идеально приспособлено для решения задач, требующих оптимизации множества гетерогенных критериев. Тогда как методы математического анализа зачастую не способны найти компромисс между несколькими конкурирующими факторами.

Применение эволюционного моделирования для создания ИС

Два преимущества ЭМ

1

Отсутствие жестко заданной логики принятия решений

2

Широкий спектр анализируемых решений (в отличие от классических подходов)

Применение эволюционного моделирования для создания ИС



Данные преимущества открывают перед исследователем новые перспективы и дают возможность использовать адаптивные самонастраивающиеся алгоритмы, позволяющие «выращивать» решения NP- сложных задач.

Фундаментальная научная проблема



Технологии создания информационных систем применяют **дискретный способ проектирования**, при этом **эволюция систем имеет непрерывный характер**.

Данное противоречие является фундаментальной научной проблемой.

Поэтому понятен интерес к исследованиям и разработке новых подходов к созданию ИС, свободных от имеющихся ограничений и условностей. Парадигма, использующая принципы эволюционного моделирования, является весьма перспективной.

Развитие поисковых систем



В современном состоянии Всемирной паутины (WWW) веб-страницы являются пассивными, а «мыслящей частью» являются именными поисковые средства. Они не только индексируют страницы Сети, но и всячески анализируют их, организуя собственные базы знаний. **Контроль поисковых систем приведет в будущем к коммерциализации сервисов обработки информации.**

Развитие поисковых систем



Группы разработчиков (IBM, HP, W3C) провели формирование онтологий для широкого ряда предметных областей, которые являются одним из ключевых элементов технологии и позволяют описывать отношения объектов реального мира с помощью языков OWL и DAML (DARPA Agent Markup Language, DARPA – Defence Advanced Research Projects Agency)

Развитие поисковых систем



Правила вывода, задаваемые в онтологиях, предоставляют дополнительные возможности. Поисковая система не «понимает» в полном смысле этого слова ничего из содержащейся в них информации, но теперь уже может манипулировать терминами гораздо более эффективно.

Компьютер способен разобраться со структурой документа, но не может обрабатывать его семантику.

Сеть GGG (IntellectNet)

NET-WEB-GRAPH

GGG (G3) – Глобальный Гносеологический Граф (Global Gnoseology Graph).
Глобальная информационная сеть нового поколения
Сетецентрические принципы создания

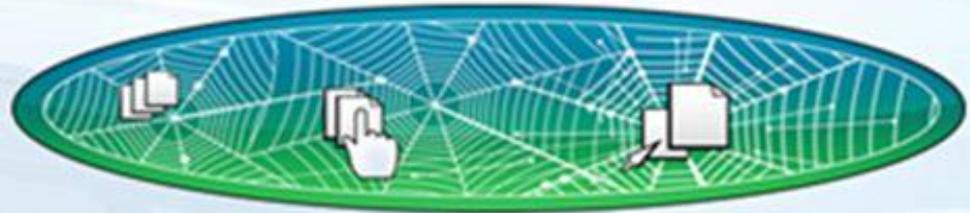
3. GRAPH – GGG (G3)

*СЕТЬ ЗНАНИЙ
СРЕДА ВЗАИМОДЕЙСТВИЯ*



2. WEB - WWW

*СЕТЬ ДОКУМЕНТОВ
СРЕДА ВЗАИМОСВЯЗИ*



1. NET

*СЕТЬ КОМПЬЮТЕРОВ
СРЕДА КОММУТАЦИИ*



Сеть GGG (IntellectNet)

Информационная эпоха завершится созданием и всемирным использованием **децентрализованной** глобальной информационной сети GRAPH – **GLOBAL GNOSEOLOGY GRAPH (GGG)**.

Сеть GGG логически дополняет имеющиеся глобальные информационные сети и продолжает взаимосвязанную цепочку **NET-WEB-GRAPH**. Где **NET** – осуществляет глобальную физическую связь различных компьютеризированных устройств, в ней нам всем мобильно и оперативно. **WEB** – формирует глобальную логическую сеть участников взаимодействия пакетами данных, в ней нам свободно и доступно. **GRAPH** – будет обеспечивать коллективное создание глобального сознания общества и гармонизацию реализации общей деятельности цивилизации.

Свойства сети GGG

Виртуальность – удобство описания реальностей мира информацией о нем

Доступность – фактическое отсутствие барьеров

Универсальность – унификация языка, знаний, культуры

Автоматическая генерация программного продукта

Способность отражать всю сложность реальных явлений

Индивидуальность



2. Проблемы и тенденции развития программного обеспечения и вычислительной техники

Уровни языков программирования





Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется **языком программирования низкого уровня**. Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

Язык ассемблера, который представляет каждую команду машинного кода, но не в виде чисел, а с помощью символьных условных обозначений, называемых **мнемониками**. Однозначное преобразование одной машинной инструкции в одну команду ассемблера называется **транслитерацией**.

Уровни языков программирования



Языки программирования **высокого** уровня понятнее человеку. Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускается гораздо меньше.

Fortran

Fortran (Фортран). Это первый компилируемый язык, созданный Джимом Бэкусом в 50-е годы. Программисты, разрабатывавшие программы исключительно на ассемблере, выражали серьезное сомнение в возможности появления высокопроизводительного языка высокого уровня, поэтому основным критерием при разработке компиляторов Фортрана являлась эффективность исполняемого кода.

Cobol



Cobol (Кобол). Это компилируемый язык для применения в экономической области и решения бизнес-задач, разработанный в начале 60-х годов. Он отличается большой «многословностью» — его операторы иногда выглядят как обычные английские фразы. В Коболе были реализованы очень мощные средства работы с большими объемами данных, хранящимися на различных внешних носителях.

Algol

Algol (Алгол). Компилируемый язык, созданный в 1960 году. Он был призван заменить Фортран, но из-за более сложной структуры не получил широкого распространения. В 1968 году была создана версия Алгол 68, по своим возможностям опережающая многие языки программирования, однако из-за отсутствия достаточно эффективных аппаратных платформ для нее не удалось своевременно создать хорошие компиляторы.

Pascal

Pascal (Паскаль). Язык Паскаль, созданный в конце 70-х годов основоположником множества идей современного программирования Никлаусом Виртом, во многом напоминает Алгол, но в нем ужесточен ряд требований к структуре программы и имеются возможности, позволяющие успешно применять его при создании крупных проектов.

Basic

Basic (Бейсик). Для этого языка имеются и компиляторы, и интерпретаторы, а по популярности он занимает одно из первых мест в мире. Он создавался в 60-х годах в качестве учебного языка и очень прост в изучении.

С

С (Си). Данный язык был создан в лаборатории Bell и первоначально не рассматривался как массовый. Он планировался для замены ассемблера, чтобы иметь возможность создавать столь же эффективные и компактные программы, и в то же время не зависеть от конкретного типа процессора.

Си во многом похож на Паскаль и имеет дополнительные средства для прямой работы с памятью (указатели).

C++ (Си++) — это объектно-ориентированное расширение языка Си, созданное Бьярном Страуструпом в 1979 году. Множество новых мощных возможностей, позволивших резко повысить производительность программистов, наложилось на унаследованную от языка Си определенную низкоуровневость, в результате чего создание сложных и надежных программ потребовало от разработчиков высокого уровня профессиональной подготовки.

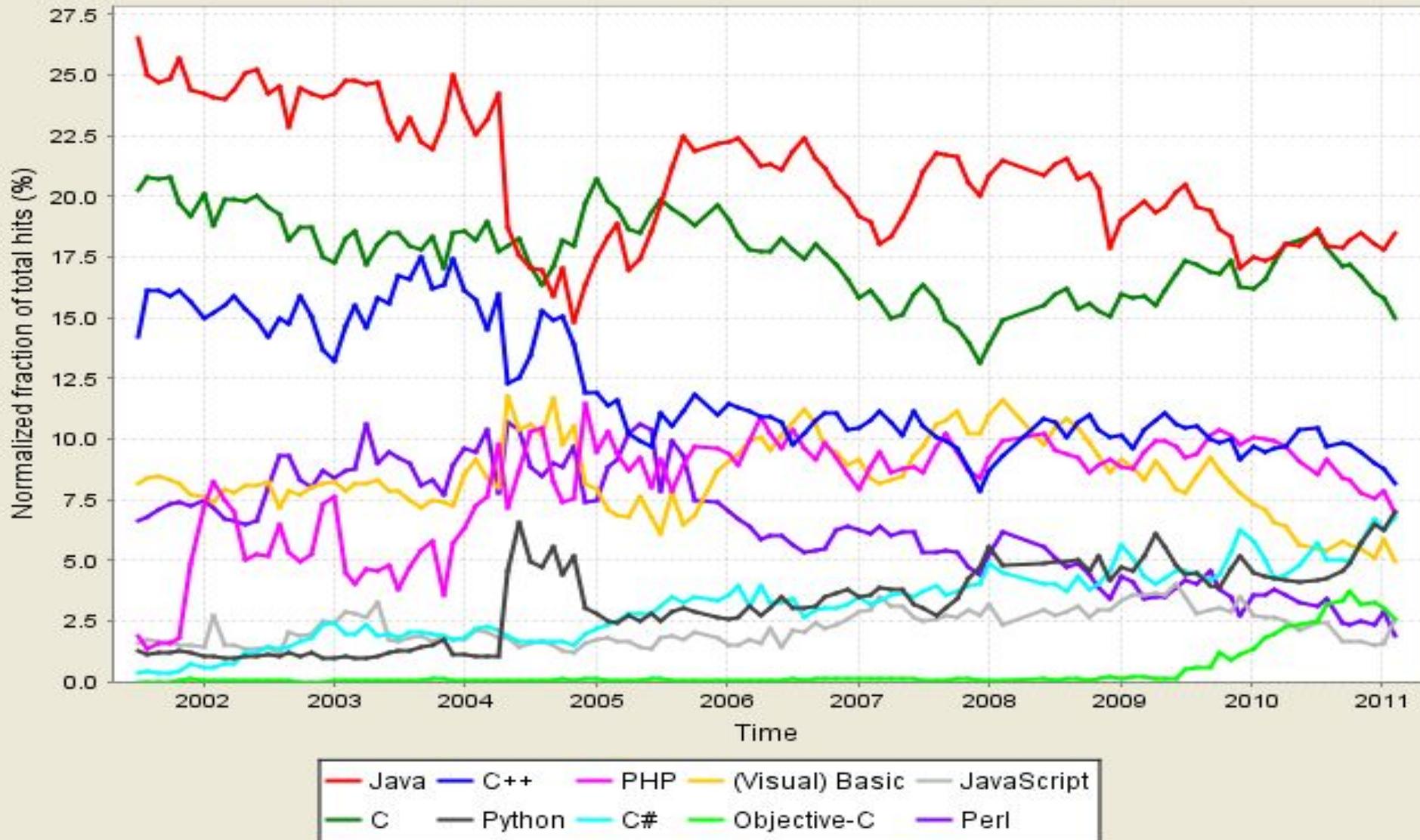
Java



Java (Джава). Этот язык был создан компанией Sun в начале 90-х годов на основе Си++. Он призван упростить разработку приложений на основе Си++ путем исключения из него всех низкоуровневых возможностей. Но главная особенность этого языка — компиляция не в машинный код, а в платформно-независимый байт-код, который потом выполняется с помощью интерпретатора — виртуальной Java-машины JVM (Java Virtual Machine), версии которой созданы сегодня для любых платформ. Благодаря наличию множества Java-машин программы на Java можно переносить не только на уровне исходных текстов, но и на уровне двоичного байт-кода.

Рейтинг языков программирования

Tiobe Programming Community Index



Рейтинг языков программирования 2010

Позиция	Язык программирования	Рейтинг
1	Java (groovy- динамическая компиляция)	18.166% (-0.48%)
2	C	17.177% (+0.33%)
3 (+1)	C++	9.802% (-0.08%)
4 (-1)	PHP	8.323% (-2.03%)
5	(Visual) Basic (макросы MS)	5.650% (-3.04%)
6	C#	4.963% (+0.55%)
7	Python	4.860% (+0.96%)
8 (+4)	Objective-C (Apple (C&Smalltalk))	3.706% (+2.54%)
9 (-1)	Perl	2.310% (-1.45%)
10	Ruby	1.941% (-0.51%)
11 (-2)	Javascript	1.659% (-1.37%)
12 (-1)	Delphi (object Pascal)	1.558% (-0.58%)
13 (+4)	Lisp	1.084% (+0.48%)
14 (+10)	Transact-SQL	0.820% (+0.42%)
15	Pascal	0.771% (+0.10%)
16 (+2)	RPG	0.708% (+0.12%)
17 (+12)	Ada	0.704% (+0.40%)
18 (-4)	SAS	0.664% (-0.14%)
19	MATLAB	0.627% (+0.05%)
20 (new)	Go	0.626% (+0.63%)

Рейтинг языков программирования 2010

Что нового в данном рейтинге? В первую очередь это быстро набирающий обороты Objective-C от Apple, который набрал 2.54% за год. Появление в первой двадцатке языка Go уже было в начале этого года.

Помимо языков от Google и Apple в 2009 высокого уровня достигли C# от Microsoft и Actionscript от Adobe. Что касается Java, то он по прежнему остался на первой строчке, хотя его популярность продолжает падать, а Си снова медленно, но верно идёт к первому месту.

Интересно также неожиданное падение популярности PHP, и Javascript.

Пожалуй, самое удивительное — появление в двадцатке легендарного военного американского языка Ada, интерес к которому вознёс его аж на 17 строчку рейтинга (по сравнению с 29 местом год назад).

Lisp

LISP (Лисп). Интерпретируемый язык программирования, созданный в 1960 году Джоном Маккарти. Ориентирован на структуру данных в форме списка и позволяет организовывать эффективную обработку больших объемов текстовой информации.



Serial Attached SCSI (SAS) — последовательный компьютерный интерфейс, разработанный для подключения различных устройств хранения данных. SAS разработан для замены параллельного интерфейса SCSI и использует тот же набор команд SCSI.

Последняя реализация SAS обеспечивает передачу данных со скоростью до 12Гбит/с на одну линию. К 2017-му году ожидается появление спецификации SAS со скоростью передачи данных 24Гбит/с.

MATLAB (matrix laboratory)

MATLAB — пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете. **MATLAB** используют более 1 млн. инженерных и научных работников, он работает на большинстве современных операционных систем, включая Linux, Mac, Solaris, Windows.

RPG (Report Program Generator)



RPG (Report Program Generator) — синтаксис изначально сходен с командным языком механических табуляторов компании IBM. Был разработан для облегчения перехода инженеров, обслуживавших эти табуляторы, на новую технику и переноса данных. Наиболее распространённой версией языка, по всей видимости, являлась RPG II. Компания IBM продолжает поддержку языка и в настоящее время, так как на нём написан громадный объём кода, который невыгодно переводить на другие языки программирования.

PERL

Perl — высокоуровневый интерпретируемый динамический. Название языка представляет собой аббревиатуру, которая расшифровывается как **Practical Extraction and Report Language** — «практический язык для извлечения данных и составления отчётов».

Первоначально аббревиатура состояла из пяти символов и в таком виде в точности совпадала с английским словом *pearl* («жемчужина»). Но затем стало известно, что такой язык существует, и букву «a» убрали.

Основной особенностью языка считаются его богатые возможности для работы с текстом, в том числе работа с регулярными выражениями, встроенная в синтаксис. Перл унаследовал много свойств от языков Си и скриптовых языков командных оболочек UNIX.

Ada

Ada (Ада). Назван по имени леди Августы Ады Байрон, дочери английского поэта Байрона и его отдаленной родственницы Анабеллы Милбэнк. В 1980 году сотни экспертов Министерства обороны США отобрали из 17 вариантов именно этот язык, разработанный небольшой группой под руководством Жана Ишбиа. Он удовлетворил на то время все требования Пентагона, а к сегодняшнему дню в его развитие вложены десятки миллиардов долларов. Структура самого языка похожа на Паскаль. В нем имеются средства строгого разграничения доступа к различным уровням спецификаций, доведена до предела мощность управляющих конструкций.

Prolog

Prolog (Пролог). Создан в начале 70-х годов Аланом Колмероз. Программа на этом языке, в основу которого положена математическая модель теории исчисления предикатов, строится из последовательности фактов и правил, а затем формулируется утверждение, которое Пролог будет пытаться доказать с помощью введенных правил. Человек только описывает структуру задачи, а внутренний «мотор» Пролога сам ищет решение с помощью методов поиска и сопоставления.

Smalltalk



Smalltalk (Смолток). Работа над этим языком началась в 1970 году в исследовательской лаборатории корпорации XEROX, а закончилась спустя 10 лет, воплотившись в окончательном варианте интерпретатора SMALLTALK-80. Данный язык оригинален тем, что его синтаксис очень компактен и базируется исключительно на понятии объекта. В этом языке отсутствуют операторы или данные. Все, что входит в Смолток, является объектами, а сами объекты общаются друг с другом исключительно с помощью сообщений (например, появление выражения $1+1$ вызывает посылку объекту 1 сообщения «+», то есть «прибавить», с параметром 1, который считается не числом-константой, а тоже объектом). Больше никаких управляющих структур, за исключением «оператора» ветвления в языке нет, хотя их можно очень просто смоделировать. Сегодня версия Visual Age for Smalltalk активно развивается компанией IBM.

1

Java - один из самых популярных языков для back-end разработки современных корпоративных веб-приложений. И это его основное преимущество. С помощью языка Java и его фреймворков веб-разработчики могут создавать масштабируемые веб-приложения для широкого круга пользователей. Java - основной язык для разработки нативных приложений (отдельная программа с собственным программным обеспечением) под Android и других приложений для смартфонов и планшетов. Большим плюсом этого языка называют WORA ("Write once, run anywhere") - "пишешь один раз, работает везде" принцип, объявленный Sun Microsystems, чтобы доказать кросс-платформенность Java. Но этот плюс не отменяет тот факт, что этот язык работает медленнее, чем другие подобные.

2

JavaScript - популярный язык среди молодых разработчиков. Он подходит для создания интерактивности сайта, или построения пользовательских интерфейсов с помощью одного из десятков популярных фреймворков. Этот язык редко встретишь за пределами браузера, вероятно, потому, что это единственное место, где он полезен. Тем не менее, JavaScript стоит учить, не только потому, что он и его десятки фреймворков набирают популярность, но и потому, что в будущем язык позволит создавать более сложные вещи.

3

C# стоит учить, потому что его знание поможет достаточно легко получить работу. Это самый популярный язык сейчас для разработки приложений для Windows, и очень популярный для мобильных устройств. Кроме того, движок для разработки игр Unity также использует C# в качестве одного из основных языков. Он очень похож на другие объектно-ориентированные языки программирования и достаточно легко учится при наличии базовых знаний C++ или Java.

4

PHP Хороший язык для создания веб-приложений для работы с данными. Это основная технология для создания мощных систем управления контентом, которые впоследствии можно расширять, чтобы сделать сайт более мощным. Был подвергнут критике как небезопасный язык, однако ситуация изменилась к лучшему после обновления в 2004 году. Тем, кто хочет учить PHP рекомендуется знать HTML, CSS и Javascript.

5

C++ Созданный в 1979 году, язык по-прежнему очень популярен и используется для построения различных типов приложений - от игр до офисных приложений. C++ предназначен для системного программирования, и подходит для разработки мощного программного обеспечения, аппаратно-ускоренных игр и приложений, требующих больших объемов памяти на настольных компьютерах, консолях и мобильных устройствах. Среди недостатков C++ программисты называют «неуклюжесть» в сравнении с Java.

Python Стоит изучить хотя бы потому, что Python - выбор Google и Ubuntu (операционная система, основанная на Debian GNU/Linux. Основным разработчиком и спонсором является компания Canonical). Но это не единственная положительная особенность языка Python, среди которых также его отличная читаемость и элегантный код. Python не требует такое количество кода для выполнения программы, как другие языки. Синтаксис ядра минималистичен, но библиотека включает большой объем полезных функций.

7

Ruby - динамический, рефлексивный, интерпретируемый высокоуровневый язык программирования для быстрого и удобного объектно-ориентированного программирования. Язык простой в освоении и невероятно мощный, плюс на нем написаны тысячи популярных веб-приложений по всему миру. Если вы любите объекты, этот язык вам подходит. Его основным преимуществом является скорость. Ruby очень похож на Python, но менее "человеческий".

Перспективные языки программирования

1

Erlang - функциональный язык программирования, разработанный компанией Ericsson, для разработки распределенных систем реального времени. Его главная особенность - параллельность. Его стоит изучать, потому что крупные банки с миллионами пользователей используют Erlang для банковских систем. Например, ПриватБанк, крупнейший украинский банк, ищет Erlang разработчиков для работы с системой Интернет-банкинга Приват24, которая ранее была написана на Java.

Перспективные языки программирования

2

R - Широко используется для разработки статистического программного обеспечения, но не очень популярен среди разработчиков. Этот язык рекомендуется знать тем, кто нуждается в серьезном анализе данных. Он работает на всех платформах и интегрируется со многими языками программирования, такими как Java, Ruby, C++, Python. Хотя он и не так популярен сейчас, ситуация может измениться в лучшую сторону. В январе 2015 Microsoft приобрела компанию Revolution Analytics, по их словам, для вклада в дальнейшее развитие языка R.

Перспективные языки программирования

3

Язык программирования **Swift** захватил разработчиков, как новый, более быстрый и легкий путь разрабатывать под Mac и iOS, по сравнению с Objective-C. Тем не менее, он актуален только в экосистеме Apple. Хороший для Apple - плохой для разработчика, который не хочет работать исключительно для Apple, особенно учитывая популярность Android. Стоит учесть, если вы хотите внести свой вклад в мир игр на iOS.

Перспективные языки программирования

4

Go (Golang) В Интернете гораздо больше информации о том, почему больше Go плохой язык, чем хороший. Этот язык разработан Google. Так, по данным Google, Go обеспечивает фундаментальную поддержку параллельного выполнения программ и коммуникации, и предлагает подход к построению системного программного обеспечения на многоядерных компьютерах. Этот язык может быть включен в список перспективных, однако есть определенные сомнения в отношении его будущего.

Go (Golang)



Он мультипоточен, со строгой типизацией, высокоуровневый. Есть поддержка определений функций и процедур, а также взаимосвязей между ними. Потоки исполняют процедуры, вызывают функции и запрашивают необходимые связи по мере надобности. Потоки разных агентов взаимодействуют между собой при помощи асинхронных сообщений. Потоки одного агента могут устанавливать динамические связи друг с другом, образуя своего рода общую память.

Сверхвысокоуровневый язык программирования



Сверхвысокоуровневый язык программирования (язык программирования сверхвысокого уровня, англ. *very high-level programming language*, **VHLL**) — язык программирования с очень высоким уровнем абстракции. В отличие от языков программирования высокого уровня, где описывается принцип «**как нужно сделать**», в сверхвысокоуровневых языках программирования описывается лишь принцип «**что нужно сделать**». Термин впервые появился в середине 1990-х годов для обозначения группы языков, используемых для быстрого прототипирования, написания одноразовых скриптов и подобных задач (Ruby, Haskell, Perl, AWK).

Другие языки программирования



Haskell. стандартизированный функциональный язык программирования общего назначения. Является одним из самых распространённых языков программирования с поддержкой отложенных вычислений. Поскольку язык функциональный, то основная управляющая структура — это функция.

Есть встроенная поддержка многозадачного и параллельного программирования, развитый инструментарий (средства автоматического тестирования, отладки и профилирования, в том числе для параллельных программ), существует несколько тысяч библиотек с открытым исходным кодом.

Другие языки программирования



AWK. интерпретируемый скриптовый C-подобный язык построочного разбора и обработки входного потока (например, текстового файла) по заданным шаблонам (регулярным выражениям). AWK рассматривает входной поток как список записей. Каждая запись делится на поля. На основе этой информации выполняется некоторый определённый программистом алгоритм обработки. По умолчанию разделителем записей является символ новой строки, разделителем полей — символ пробела или табуляции, или последовательность таких символов. Символы-разделители можно явно определить в программе. Символ-разделитель полей можно определить и в командной строке. Каждая запись поочерёдно сравнивается со всеми шаблонами, и каждый раз когда она соответствует шаблону, выполняется указанное действие.

Другие языки программирования



Delphi (Object Pascal) — императивный, структурированный, объектно-ориентированный язык программирования со строгой статической типизацией переменных. Основная область использования — написание прикладного программного обеспечения (Windows, Mac OS, iOS, Android). Структурированный язык запросов **SQL (Structured Query Language)**. Он основан на мощной математической теории и позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а группами записей.

HTML. Общеизвестный язык для оформления документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц.

Другие языки программирования



Tcl/Tk (*Tool Command Language*). В конце 80-х годов Джон Оустерхаут придумал скрипт-язык **Tcl** и библиотеку **Tk**. В **Tcl** он попытался воплотить видение идеального скрипт-языка. **Tcl** ориентирован на автоматизацию рутинных процессов и состоит из мощных команд, предназначенных для работы с абстрактными нетипизированными объектами. Он независим от типа системы и при этом позволяет создавать программы с графическим интерфейсом.

VRML (предок X3D). В 1994 году был создан язык **VRML** для организации виртуальных трехмерных интерфейсов в Интернете. Он позволяет описывать в текстовом виде различные трехмерные сцены, освещение и тени, текстуры (покрытия объектов), создавать свои миры, путешествовать по ним, «облетать» со всех сторон, вращать в любых направлениях, масштабировать, регулировать освещенность и т. д.

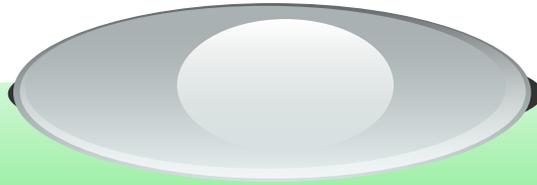
Другие языки программирования



PL/I (ПЛ/1). В середине 60-х годов компания IBM решила взять все лучшее из языков Фортран, Кобол и Алгол. В результате в 1964 году на свет появился новый компилируемый язык программирования, который получил название **Programming Language One**. В этом языке было реализовано множество уникальных решений. По своим возможностям ПЛ/1 значительно мощнее многих других языков (Си, Паскаля). Этот язык и сегодня продолжает поддерживаться компанией IBM.



Forth (Форт). Результат попытки Чарльза Мура в 70-х годах создать язык, обладающий мощными средствами программирования, который можно эффективно реализовать на компьютерах с небольшими объемами памяти, а компилятор мог бы выдавать очень быстрый и компактный код — то есть служил заменой ассемблеру. Однако сложности восприятия программного текста, записанного в непривычной форме, сильно затрудняли поиск ошибок, и с появлением Си язык Форт оказался забытым.



Командный интерпретатор Shell, используемый в операционных системах семейства UNIX, в котором пользователь может либо давать команды операционной системе по отдельности, либо запускать скрипты, состоящие из списка команд.

Основное назначение этого языка состоит в том, чтобы предоставить пользователю удобные средства взаимодействия с системой.

Скриптовые языки и языки программирования

1

Bash (Bourne Again shell), Csh (C shell), JavaScript, Ksh (Korn shell), Perl, Python, Ruby, Sh (Bourne shell), Tcl

2

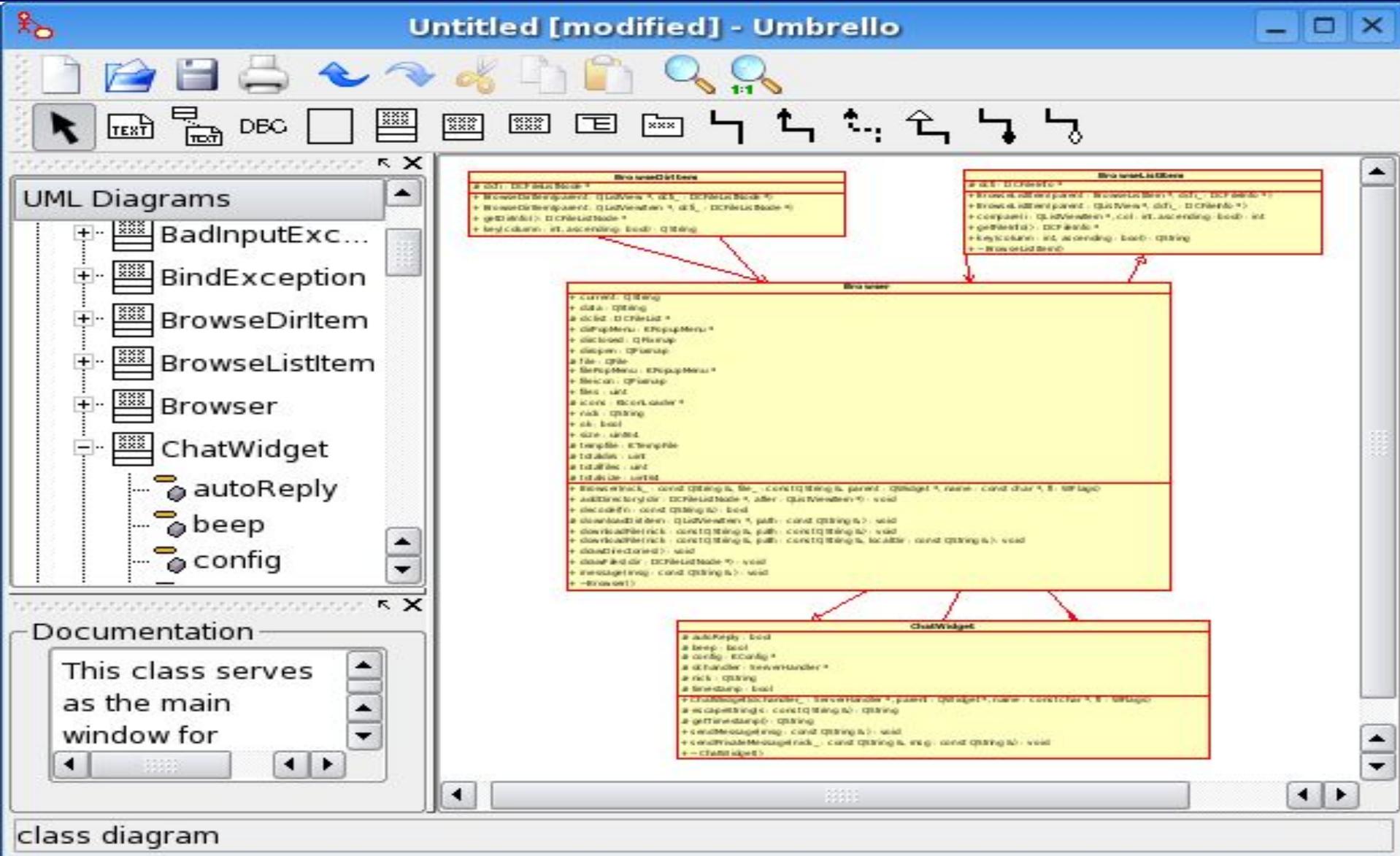
Assembler, C, C++, C#, FORTRAN, Forth, Java, LISP, Pascal

Case- технологии



Основной целью CASE-технологии является разграничение процесса проектирования программных продуктов от процесса кодирования и последующих этапов разработки, максимально автоматизировать процесс разработки. Для выполнения поставленной цели CASE-технологии используют два принципиально разных подхода к проектированию: **структурный (IDEF)** и **объектно-ориентированный (UML)**.

Case-технологии



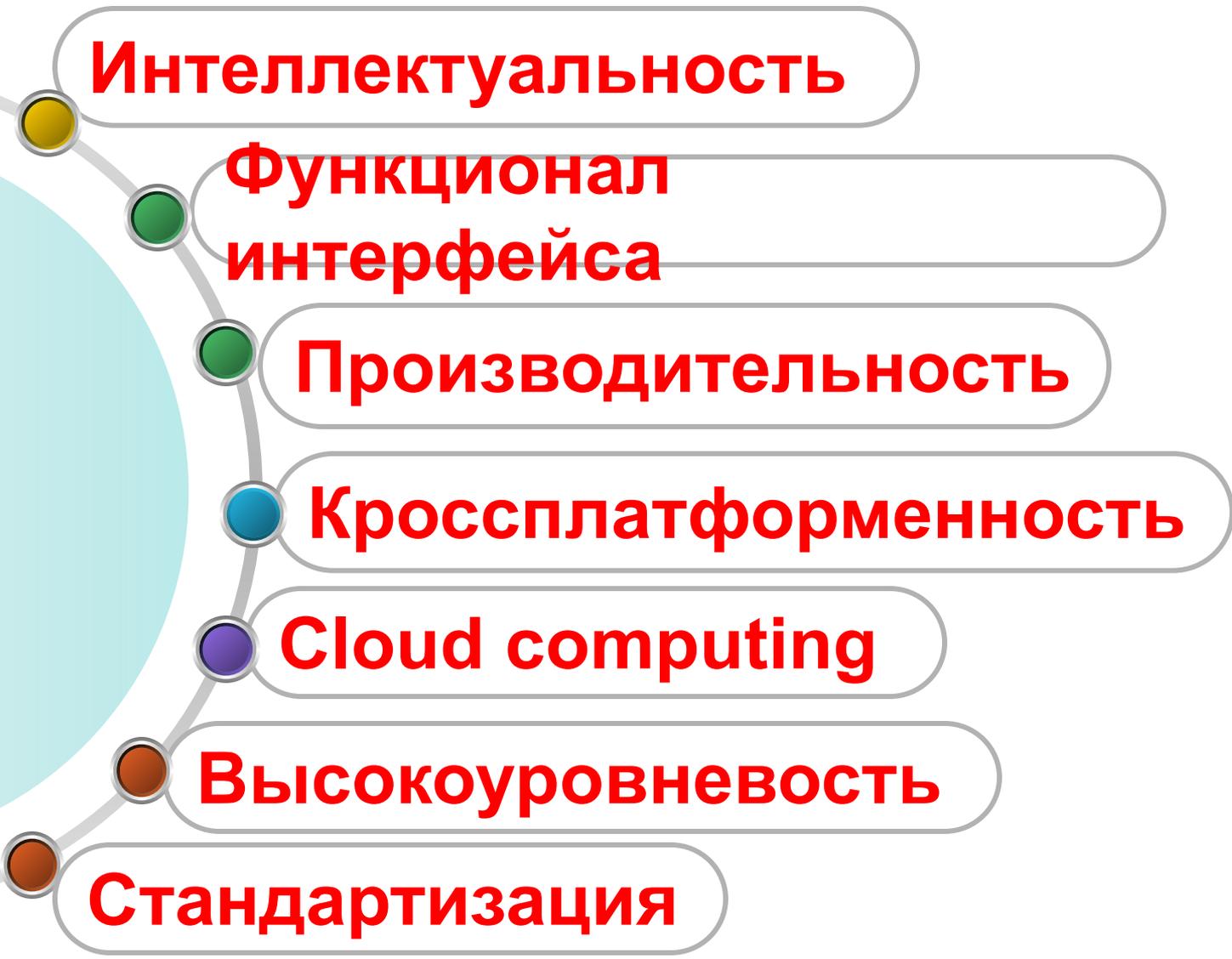
ARIS

Одной из современных методологий бизнес-моделирования, получившей широкое распространение является методология **ARIS**, которая расшифровывается как **Architecture of Integrated Information Systems** - проектирование интегрированных информационных систем. Ее использует программное средство ARIS Toolset

ARIS



Тенденции развития ПО



Интеллектуальность

**Функционал
интерфейса**

Производительность

Кроссплатформенность

Cloud computing

Высокоуровневость

Стандартизация

Интеллектуальность



Прогнозирование последующих действий пользователя. **Создание адаптирующихся интерфейсов.**

Повышение конкурентоспособности ПО лежит в области повышения интеллектуальности продукта. Даже в простых программах где, казалось бы, некуда “приткнуть” интеллектуальность, можно предпринять ряд шагов, делающих программу более удобной в использовании.

Функционал интерфейса



Лет десять-пятнадцать назад удобство пользовательского интерфейса не было решающим фактором при выборе ПО. Ценилась больше функциональность. Это было связано с тем, что программы были не столь функциональны, инструментарий программиста был не такой мощный. В результате, программирование одной функции было огромной работой. Если ваше ПО имело на 2-3 функции больше, чем у конкурента, то у вас были большие шансы на успех. Сегодня практически любой функционал легко и быстро повторяется конкурентами. Получить длительное по времени конкурентное преимущество можно, внедрив более **интеллектуальный функционал** и, как ни удивительно, разработав хороший интерфейс пользователя.

Производительность



Ярким примером, иллюстрирующим тему данного абзаца, можно назвать гонку браузеров за производительностью их движков. Разработчики активно оптимизируют операционные системы, различное прикладное ПО.

У пользователя несколько процессоров, много памяти, графический ускоритель- так **почему бы не задействовать это на “полную катушку”?** В конечном итоге, быстрый, отзывчивый и удобный интерфейс очень понравится пользователю.

Кроссплатформенность



Обеспечить кроссплатформенность тому или иному алгоритму на сегодняшний день не сложно. Но вот **интерфейсы... с ними заговоздка.**

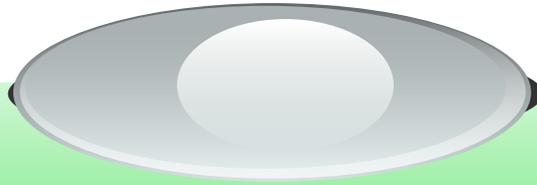
Разработчики приноровились использовать web-интерфейсы даже в исключительно офф-лайн приложениях. Такой способ позволяет снять ряд проблем при создании кроссплатформенных интерфейсов.

Cloud computing



Web-приложения, устанавливаемое ПО- у каждого типа приложений есть свои достоинства и недостатки. С одной стороны, web-приложения позволяют нам просто делать свое дело и не заботиться об установке ПО, о резервных копиях данных. С другой стороны, устанавливаемое ПО позволяет работать в офф-лайнном режиме, задействовать все имеющиеся ресурсы на компьютере. Возможность получить преимущества от обоих типов ПО дают облачные вычисления. **Предлагается расширить использование облаков, задействуя их в полноценном устанавливаемом ПО.**

Высокоуровневость



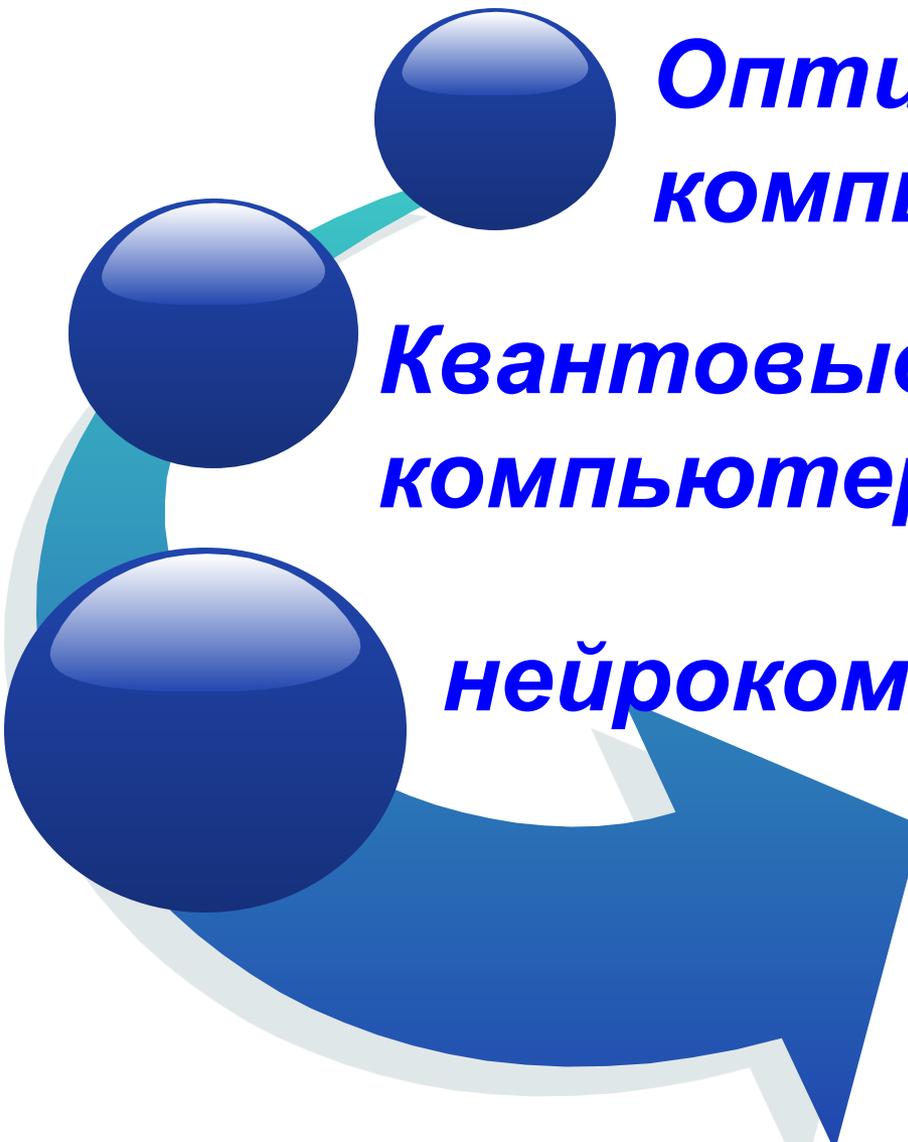
Наделение коммерческого ПО элементами искусственного интеллекта- тема еще свежая, “мало раскопанная”, поэтому тут есть, где развернуться, есть интерес со стороны потребителей, есть деньги. Можно констатировать, что обладание своими низкоуровневыми технологиями на сегодняшний день - недостаток. Ведь на поддержку и развитие их надо тратить время и деньги. При этом, конкурентных преимуществ никаких. С другой стороны- высокоуровневое программирование с новыми перспективами, денежными рынками.

Стандартизация



Все больший приоритет приобретают **стандарты при проектировании ПО**. Формируются приемы программирования, нарабатываются методики ведения проектов. Хотя проблема быстрого устаревания знания еще актуальна, но она уже явно менее остра, чем 20 лет назад. ИТ стабилизируются в своем развитии, выходят на плоское плато S-образной кривой развития.

Тенденции развития вычислительной техники



*Оптические
компьютеры*

*Квантовые
компьютеры*

нейрокомпьютеры

Оптические компьютеры

**аналоговые интерференционные
оптические вычисления**

**использование оптических
соединений для передачи
сигналов**

**создание компьютера,
полностью состоящего из
оптических устройств
обработки информации**

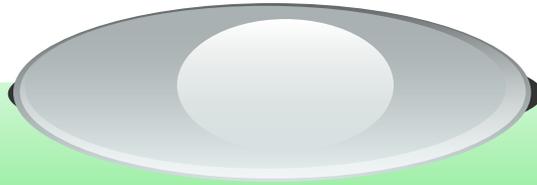
**Три основных
направления
развития
оптических
методов в ВТ**

Оптические компьютеры



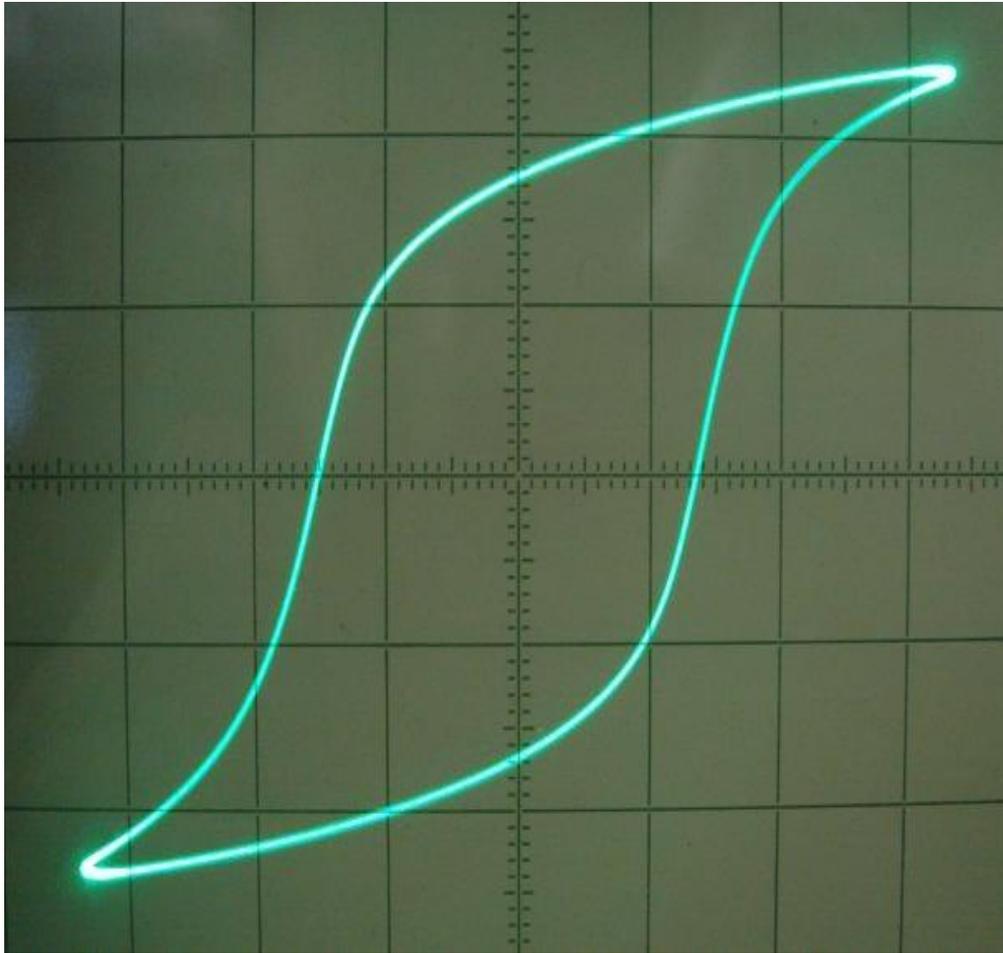
В основе работы различных компонентов оптического компьютера (трансфазаторы-оптические транзисторы, триггеры, ячейки памяти, носители информации) лежит **явление оптической бистабильности**. Оптическая бистабильность - это одно из проявлений взаимодействия света с веществом в нелинейных системах с обратной связью, при котором определенной интенсивности и поляризации падающего на вещество излучения соответствуют два (аналог 0 и 1 в полупроводниковых системах) возможных стационарных состояния световой волны, прошедшей через вещество, отличающихся амплитудой и (или) параметрами поляризации.

Принцип работы



Увеличение интенсивности падающего на вещество светового луча до некоторого значения I_1 приводит к резкому возрастанию интенсивности прошедшего луча; на обратном же ходе при уменьшении интенсивности падающего луча до некоторого значения $I_2 < I_1$ интенсивность прошедшего луча остается постоянной, а затем резко падает. Таким образом, интенсивности падающего пучка I , значение которой находится в пределах петли гистерезиса, соответствуют два значения интенсивности прошедшего пучка, зависящих от предыдущего оптического состояния поглощающего вещества.

Гистерезис



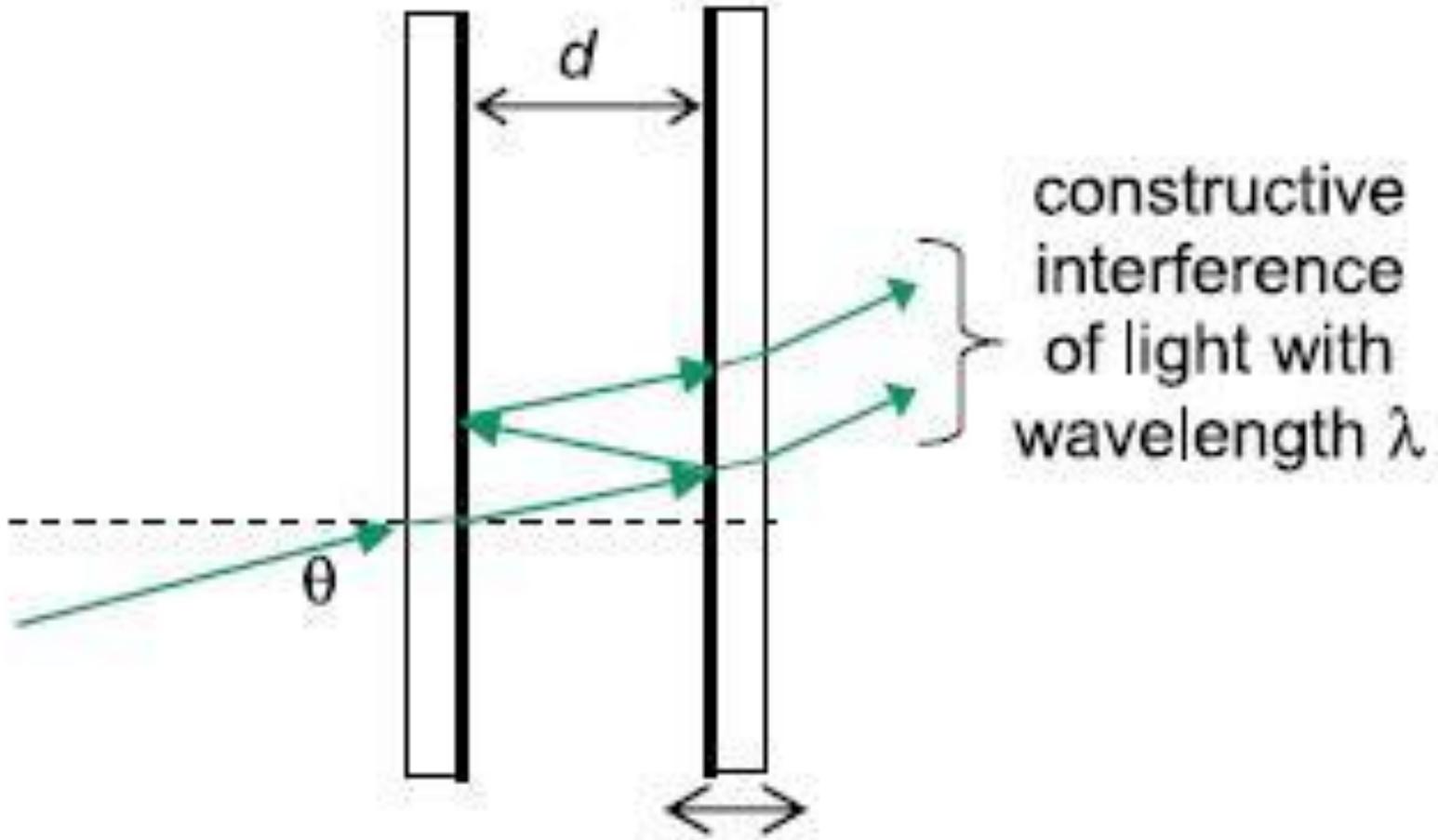
Гистерезис – это свойство биологических, физических и прочих систем, в которых мгновенный отклик на воздействия зависит от их текущего состояния, а на интервале времени поведение системы определяется ее предысторией. **Петлей гистерезиса** называется график, демонстрирующий это свойство.

Оптические логические устройства



Весь набор полностью оптических логических устройств для синтеза более сложных блоков оптических компьютеров реализуется на основе пассивных **нелинейных резонаторов-интерферометров**. В зависимости от начальных условий (начального положения пика пропускания и начальной интенсивности оптического излучения) в пассивном нелинейном резонаторе, **нелинейный процесс завершается установлением одного из двух устойчивых состояний пропускания падающего излучения**. А из нескольких нелинейных резонаторов можно собрать любой, более сложный логический элемент (триггер).

Нелинейные резонаторы



Оптические элементы памяти



Элементы памяти оптического компьютера представляют собой полупроводниковые нелинейные оптические интерферометры, в основном, созданными из арсенида галлия (GaAs).

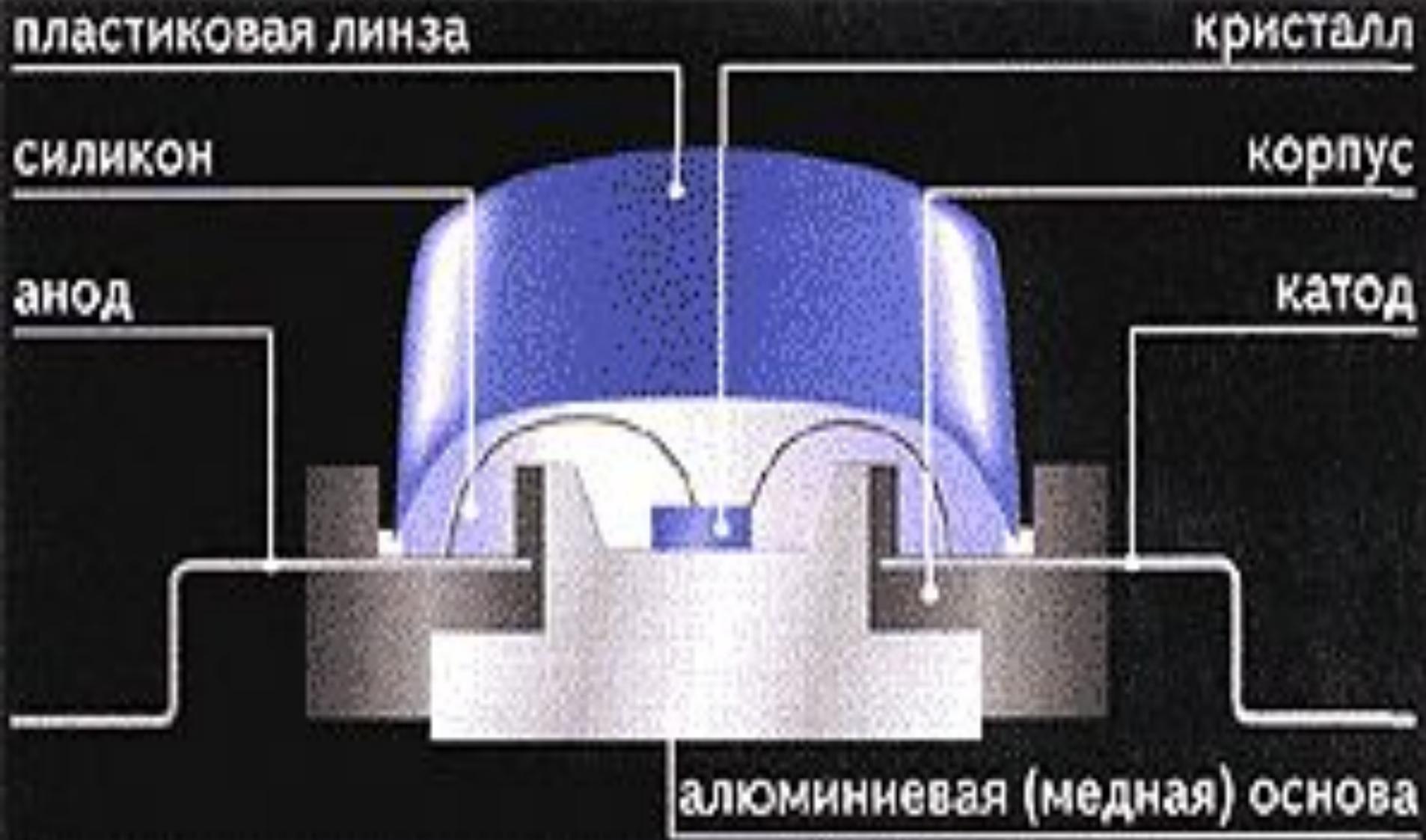
Минимальный размер оптического элемента памяти определяется минимально необходимым числом атомов, для которого устойчиво наблюдается оптическая бистабильность. Это число составляет ~1000 атомов, что соответствует **1-10 нанометрам**.

light emitting diode



Светодиод состоит из полупроводникового кристалла на токонепроводящей подложке, корпуса с контактными выводами и оптической системой. Для повышения живучести пространство между кристаллом и пластиковой линзой заполнено прозрачным силиконом. Алюминиевая основа служит для отвода избыточного тепла. Которого, надо сказать, выделяется совсем небольшое количество.

light emitting diode



light emitting diode



Свечение в полупроводниковом кристалле возникает при **рекомбинации электронов и дырок в области р-n-перехода**. Область р-n-перехода, образуется контактом двух полупроводников с разными типами проводимости.

Очевидно, что **чем больший ток проходит через светодиод, тем он светит ярче**, поскольку чем больше ток, тем больше электронов и дырок поступают в зону рекомбинации в единицу времени.

light emitting diode



Чтобы р-п-переход стал излучать свет, ширина зоны в активной области светодиода должна быть близка к энергии квантов света видимого диапазона. Во-вторых, полупроводниковый кристалл должен содержать мало дефектов, из-за которых рекомбинация происходит без излучения. Чтобы соблюсти оба условия, зачастую одного р-п-перехода в кристалле оказывается недостаточно, и производители вынуждены идти на изготовление многослойных полупроводниковых структур, так называемых **гетероструктур**.

Проблема цвета



Белый свет от светодиодов можно получить несколькими способами.

Первый — смешать цвета по технологии **RGB**. На одной матрице плотно размещаются красные, голубые и зеленые светодиоды, излучение которых смешивается при помощи оптической системы, например линзы. В результате получается белый свет. **Второй** способ заключается в том, что на поверхность светодиода, излучающего в ультрафиолетовом диапазоне (есть и такие), наносится три люминофора, излучающих, соответственно, голубой, зеленый и красный свет. По принципу люминесцентной лампы. **Третий** способ - это когда желто-зеленый или зелено-красный люминофор наносится на голубой светодиод. При этом два или три излучения смешиваются, образуя белый или близкий к белому свет.

Последние разработки



Суть последних разработок: **избавиться от преобразования светового сигнала (фотонов) в электрический (электронов)** на всем протяжении прохождения по цепи. Если раньше процессы протекающие в чипе можно было описать как **фотон-электрон-фотон**, то сейчас, благодаря «**диоду для света**» встроенному в чип — **фотон-фотон**. Оптический компьютер становится реальностью.

Последние разработки



Это значит, что по волоконно-оптической сети к компьютеру может подводиться сразу **несколько пучков информации** без преобразования и замедления сигнала. Скорость света превосходит скорость компьютера на электронах. По медным проводам может проходить **только один электронный поток данных**.

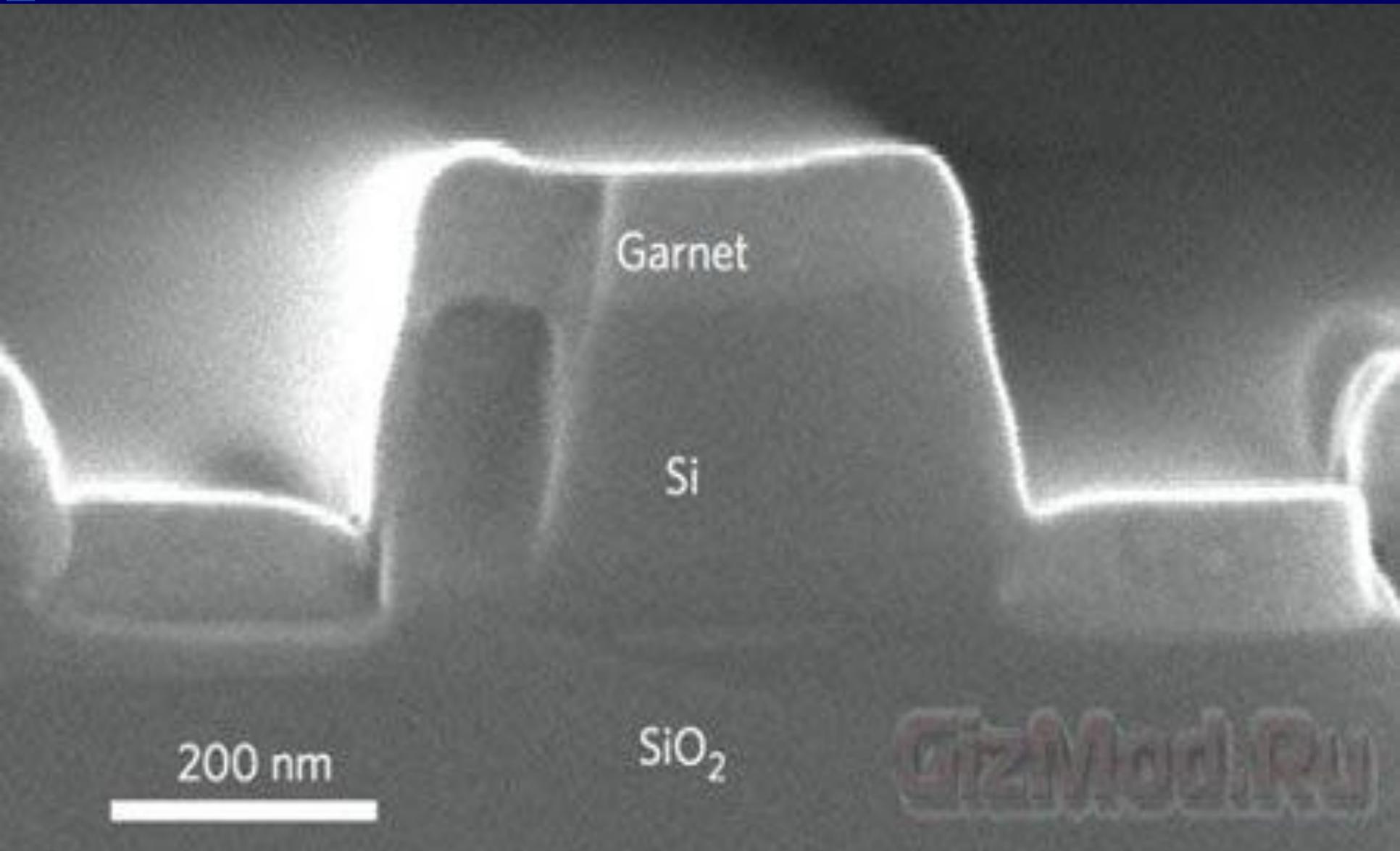
«Диод для света»



Он работает так же, как его электрический аналог, пропуская лишь в одном направлении, только **не ток, а свет**.

Ключевым для разработки является выбор материала, обладающего необходимыми свойствами. На его роль подходит **гранат**, пленка из которого на поверхности кремниевого чипа по-разному преломляет свет, падающий на нее под разными углами, в результате пропуская свет только в одном направлении. Основная заслуга специалистов MIT как раз заключается в том, что они нашли способ формирования этой пленки с применением стандартного оборудования для выпуска кремниевых чипов.

«Диод для света»



Проблемы создания ОК



К настоящему времени **уже созданы и оптимизированы отдельные составляющие оптических компьютеров** – оптические процессоры, ячейки памяти, однако до полной сборки еще далеко. Основной проблемой, стоящей перед учеными, является синхронизация работы отдельных элементов оптического компьютера в единой системе, поскольку уже существующие элементы характеризуются различными параметрами рабочей волны светового излучения (интенсивность, длина волны), и уменьшение его размера. Если для конструирования оптического компьютера использовать уже разработанные компоненты, то обычный РС имел бы размеры легкового автомобиля.

Потенциальные преимущества ОК

световые потоки, в отличие от электрических, могут пересекаться друг с другом

световые потоки могут быть локализованы в поперечном направлении до нанометровых размеров и передаваться по свободному пространству

скорость распространения светового сигнала выше скорости электрического

взаимодействие световых потоков с нелинейными средами распределено по всей среде, что дает новые степени свободы в организации связи и создании параллельных архитектур

Потенциальные преимущества ОК



Создание большего количества параллельных архитектур, по сравнению с полупроводниковыми компьютерами, является **основным достоинством оптических компьютеров, оно позволяет преодолеть ограничения по быстродействию и параллельной обработке информации, свойственные современным ЭВМ.** Развитие оптических технологий все равно будет продолжаться, поскольку полученные результаты важны не только для создания оптических компьютеров, но также и для оптических коммуникаций и сети Internet.

Квантовые компьютеры



Идея о квантовых вычислениях была высказана Юрием Маниным в 1980 году, одна из первых моделей квантового компьютера была предложена Ричардом Фейнманом в 1981 году. Вскоре Пол Бениофф описал теоретические основы построения такого компьютера.

Необходимость в квантовом компьютере возникает тогда, когда мы пытаемся исследовать методами физики сложные многочастичные системы, подобные биологическим. Пространство квантовых состояний таких систем растет как экспонента от числа n составляющих их реальных частиц, что делает невозможным моделирование их поведения на классических компьютерах уже для $n=10$.

Квантовые компьютеры



Бит имеет лишь два состояния - 0 и 1, тогда как состояний **кубита (qubit, Quantum Bit)** значительно больше. Существуют волновые функции, которые называются собственными для какой-либо определенной величины. Квантовая система может находиться в состоянии с волновой функцией, равной линейной комбинации собственных функций, соответствующих каждому из возможных значений (такое состояние называется сложным), т. е. физически - ни в возбужденном, ни в основном состоянии (**суперпозиция**). Это означает, что кубит в одну единицу времени равен и 0, и 1, тогда как классический бит в ту же единицу времени равен либо 0, либо 1. Как для классических, так и для квантовых компьютеров были введены элементарные логические операции: дизъюнкция, конъюнкция и квантовое отрицание, при помощи которых будет организована вся логика квантового компьютера.

Квантовые компьютеры



Квантовую суперпозицию можно проиллюстрировать, например, так: «Вообразите атом, который мог бы подвергнуться радиоактивному распаду в определённый промежуток времени. Или не подвергнуться. Мы можем ожидать, что у этого атома есть только два возможных состояния: „распад“ и „не распад“, но в квантовой механике у атома может быть некое объединённое состояние — „распада — не распада“, то есть ни то, ни другое, а как бы между. Вот это состояние и называется „суперпозицией“»

Квантовые компьютеры



Согласно законам квантовой механики, **энергия электрона, связанного в атоме, не произвольна**. Она может иметь лишь определенный прерывный (дискретный) ряд значений E_0, E_1, \dots, E_n называемых уровнями энергии (**спектр атома**). Самый нижний уровень энергии E_0 , при котором энергия атома наименьшая, называется **основным**. Остальные уровни (E_1, E_2, \dots, E_n) соответствуют более высокой энергии атома и называются **возбужденными**. Излучение и поглощение атомом электромагнитной энергии происходит отдельными порциями - квантами, или фотонами. При поглощении фотона энергия увеличивается - он переходит «вверх» - с нижнего на верхний уровень, при излучении фотона атом совершает переход вниз.

Квантовые компьютеры



Если атом в данный момент времени находится в одном из возбужденных состояний E_2 , то такое состояние атома неустойчиво, даже если на него не влияют другие частицы. Через очень короткое время атом перейдет в одно из состояний с меньшей энергией, например E_1 . Такой самопроизвольный (спонтанный) переход с одного уровня на другой и сопровождающее его спонтанное излучение столь же случайны во времени, как радиоактивный распад ядра атома.

Предсказать точно момент перехода принципиально невозможно - можно лишь говорить о вероятности того, что переход произойдет через такое-то время.

Квантовые компьютеры



Но атом может перейти с уровня E_2 на E_1 не спонтанно, а под действием электромагнитной волны, если только частота этой волны достаточно близка к частоте перехода атома. Такая резонансная волна как бы «расшатывает» электрон и ускоряет его «падение» на уровень с меньшей энергией. Переходы, происходящие под действием внешнего электромагнитного поля, называются вынужденными (или стимулированными).

При создании квантового компьютера основное внимание уделяется вопросам управления кубитами при помощи вынужденного излучения и недопущении спонтанного излучения, которое нарушит работу всей квантовой системы.

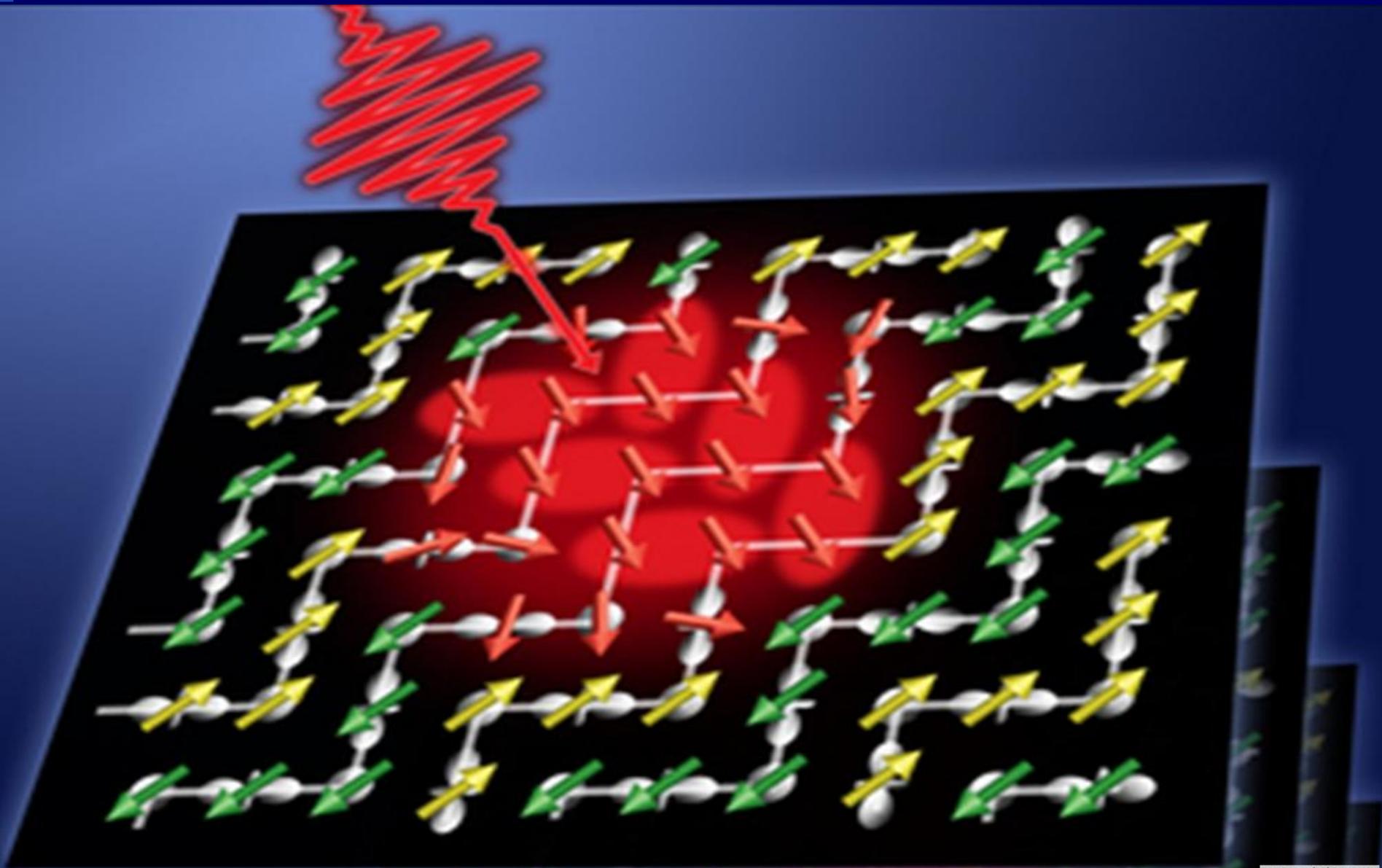
Квантовые компьютеры



Идея квантовых вычислений состоит в том, что квантовая система из L двухуровневых квантовых элементов (квантовых битов, кубитов) имеет 2^L линейно независимых состояний, а значит, вследствие принципа квантовой суперпозиции, пространство состояний такого квантового регистра является 2^L -мерным гильбертовым пространством. Операция в квантовых вычислениях соответствует повороту вектора состояния регистра в этом пространстве. Таким образом, квантовое вычислительное устройство размером L кубит фактически задействует одновременно 2^L классических состояний.

Физическими системами, реализующими кубиты, могут быть любые объекты, имеющие два квантовых состояния: поляризационные состояния фотонов, электронные состояния изолированных атомов или ионов, спиновые состояния ядер атомов, и т. д.

Квантовые компьютеры



Квантовые компьютеры



Квантовый бит, называемый кубитом, находится в состоянии $a|0\rangle + b|1\rangle$, так что $|a|^2$ и $|b|^2$ — вероятности получить **0** или **1** соответственно при измерении этого состояния; $|a|^2 + |b|^2 = 1$. Сразу после измерения кубит переходит в базовое квантовое состояние, соответствующее классическому результату.

Пример:

Имеется кубит в квантовом состоянии $\frac{4}{5}|0\rangle - \frac{3}{5}|1\rangle$. В этом случае, вероятность получить при измерении **0** составляет $(\frac{4}{5})^2 = \frac{16}{25} = 64\%$, **1** $(-\frac{3}{5})^2 = \frac{9}{25} = 36\%$.

В результате измерения кубит переходит в новое квантовое состояние, то есть, при следующем измерении этого кубита мы получим **0** со стопроцентной вероятностью.

Квантовые компьютеры



Перейдем к системе из двух кубитов. Измерение каждого из них может дать **0** или **1**. Поэтому у системы есть **4 классических состояния**: **00, 01, 10 и 11**. Общее квантовое состояние системы имеет вид $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$. Теперь $|a|^2$ — вероятность измерить 00 и т. д. Отметим, что $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$ как полная вероятность.

В общем случае системы из L кубитов, у неё 2^L классических состояний (**00000... (L-нулей), ...00001 (L-цифр), ..., 11111... (L-единиц)**), каждое из которых может быть измерено с вероятностями 0—100 %.

Одна операция над группой кубитов затрагивает все значения, которые она может принимать, в отличие от классического бита. Это и обеспечивает беспрецедентный параллелизм вычислений.

Квантовые компьютеры



Большая часть современных ЭВМ работают по схеме: n бит памяти хранят состояние и каждый такт времени изменяются процессором. В квантовом случае система из n кубитов находится в состоянии, являющимся суперпозицией всех базовых состояний, поэтому изменение системы касается всех 2^n базовых состояний одновременно. Теоретически новая схема может работать намного (в экспоненциальное число раз) быстрее классической. Практически (квантовый) алгоритм Гровера поиска в базе данных показывает квадратичный прирост мощности против классических алгоритмов.

Квантовые алгоритмы

Алгоритм Гровера позволяет найти решение уравнения $f(x) = 1, 0 \leq x < N$ за время $O(\sqrt{N})$.

Алгоритм Шора позволяет разложить натуральное число n на простые множители за полиномиальное от $\log(n)$ время.

Алгоритм Залки — Визнера позволяет моделировать

унитарную эволюцию квантовой системы частиц за почти линейное время с использованием $O(n)$

кубит

Алгоритм Дойча — Йожи позволяет «за одно вычисление» определить, является ли функция двоичной переменной $f(n)$ постоянной ($f_1(n) = 0, f_2(n) = 1$ независимо от n или «сбалансированной» ($f_3(0) = 0, f_3(1) = 1; f_4(0) = 1, f_4(1) = 0$).

Алгоритм Саймона решает проблему чёрного ящика экспоненциально быстрее, чем любой классический алгоритм, включая вероятностные алгоритмы.

Квантовые компьютеры (успешные применения)



Благодаря огромной скорости разложения на простые множители, квантовый компьютер позволит расшифровывать сообщения, зашифрованные асимметричным криптографическим алгоритмом **RSA** (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. До сих пор этот алгоритм считается сравнительно надёжным, так как эффективный способ разложения чисел на простые множители для классического компьютера в настоящее время неизвестен. Для того, например, чтобы получить доступ к кредитной карте, нужно разложить на два простых множителя число длиной в сотни цифр. Даже для самых быстрых современных компьютеров выполнение этой задачи заняло бы в сотни раз больше времени, чем возраст Вселенной. Благодаря **алгоритму Шора** эта задача становится вполне осуществимой, если квантовый компьютер будет построен.

Первые реализации квантовых компьютеров

В конце **2001** года **IBM** заявила об успешном тестировании **7-кубитного** квантового компьютера, реализованного с помощью **ЯМР**. На нём был исполнен алгоритм Шора и были найдены сомножители числа 15.

В **2005** году группой **Ю. Пашкина** (сотрудник лаборатории сверхпроводимости г. Москвы) при помощи японских специалистов был построен **двухкубитный квантовый процессор** на сверхпроводящих элементах.

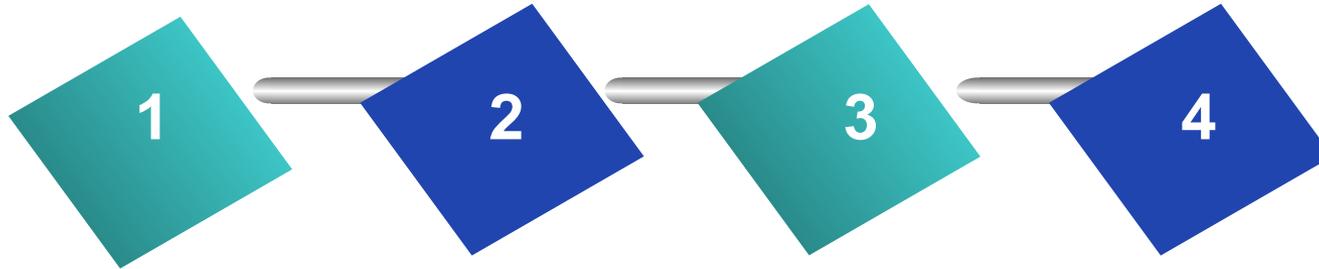
В ноябре **2009** года физикам из **Национального института стандартов и технологий в США** впервые удалось собрать программируемый квантовый компьютер, состоящий из двух кубит.

В феврале **2012** года компания **IBM** сообщила о достижении значительного прогресса в физической реализации квантовых вычислений с использованием

сверхпроводящих кубитов, которые, по мнению компании, позволят начать работы по созданию квантового компьютера.

В апреле **2012** года группе исследователей из Южно-Калифорнийского университета, Технологического университета Дельфта, университета штата Айова, и Калифорнийского университета удалось построить **двухкубитный квантовый компьютер на кристалле алмаза с примесями**. На нем реализован алгоритм Гровера для 4-х вариантов перебора (95%)

Принципы практической реализации квантовых компьютеров (1996 D.P. Divincenzo)



**Точно
известное
число
частиц
системы.**

**Возможность
приведения
системы в
точно
известное
начальное
состояние.**

**Высокая
степень
изоляции от
внешней
среды.**

**Умение менять
состояние системы
согласно заданной
последовательности
элементарных
преобразований.**

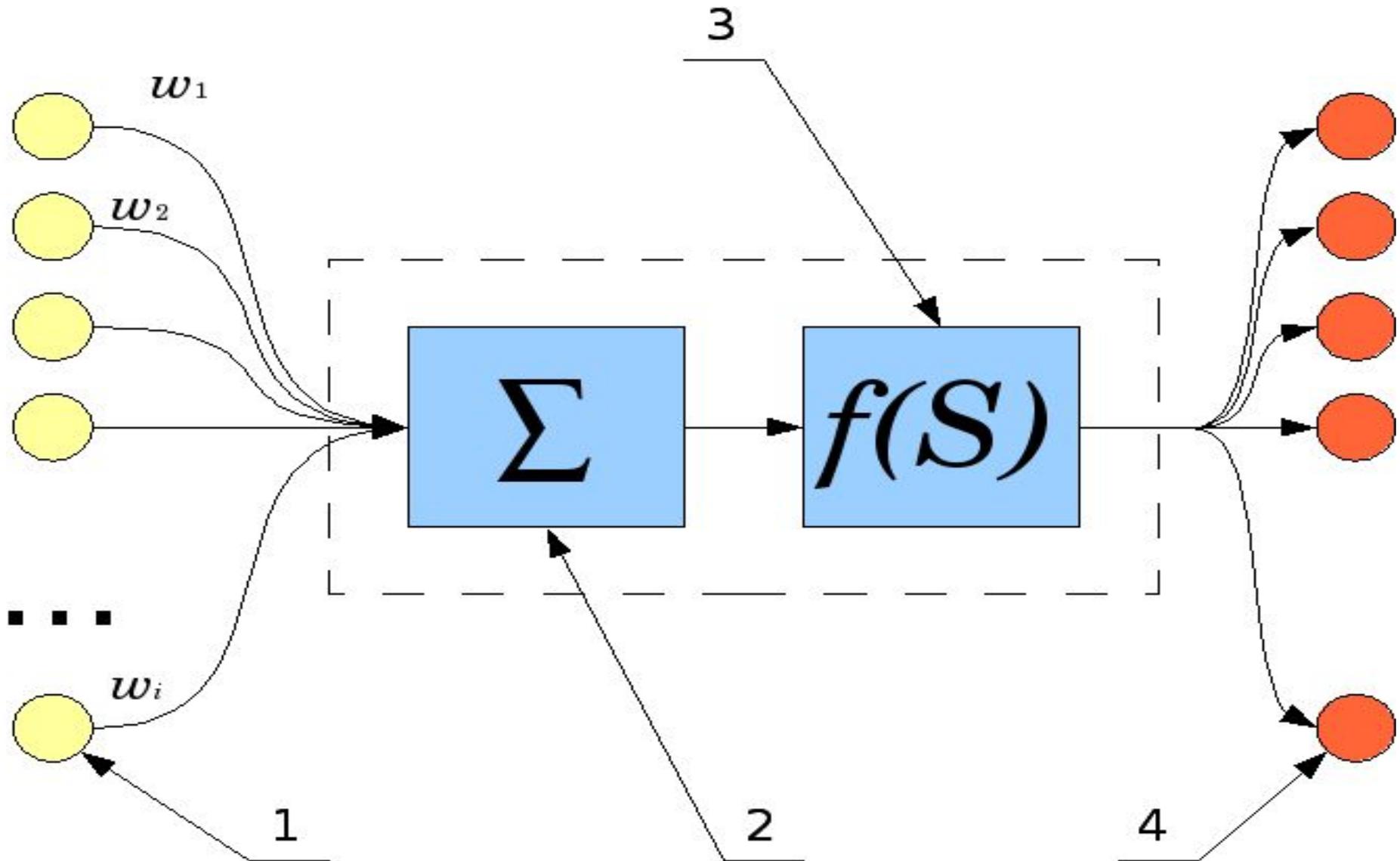
Нейрокомпьютеры



Нейрокомпьютер — устройство переработки информации на основе принципов работы естественных нейронных систем. Эти принципы были формализованы, что позволило говорить о теории искусственных нейронных сетей.

Проблематика же нейрокомпьютеров заключается в построении реальных физических устройств, что позволит не просто моделировать искусственные нейронные сети на обычном компьютере, но так изменить принципы работы компьютера, что станет возможным говорить о том, что они работают в соответствии с теорией искусственных нейронных сетей.

Формальный нейрон



Нейрокомпьютеры (основная идея)



В отличие от цифровых систем, представляющих собой комбинации процессорных и запоминающих блоков, **нейропроцессоры содержат память, распределённую в связях между очень простыми процессорами**, которые часто могут быть описаны как формальные нейроны или блоки из однотипных формальных нейронов. **Тем самым основная нагрузка на выполнение конкретных функций процессорами ложится на архитектуру системы**, детали которой в свою очередь определяются межнейронными связями. Подход, основанный на представлении как памяти данных, так и алгоритмов системой связей (и их весами), называется **коннекционизмом**.

Основные преимущества нейрокомпьютеров

1

- Все алгоритмы нейроинформатики высокопараллельны, а это уже залог высокого быстродействия;

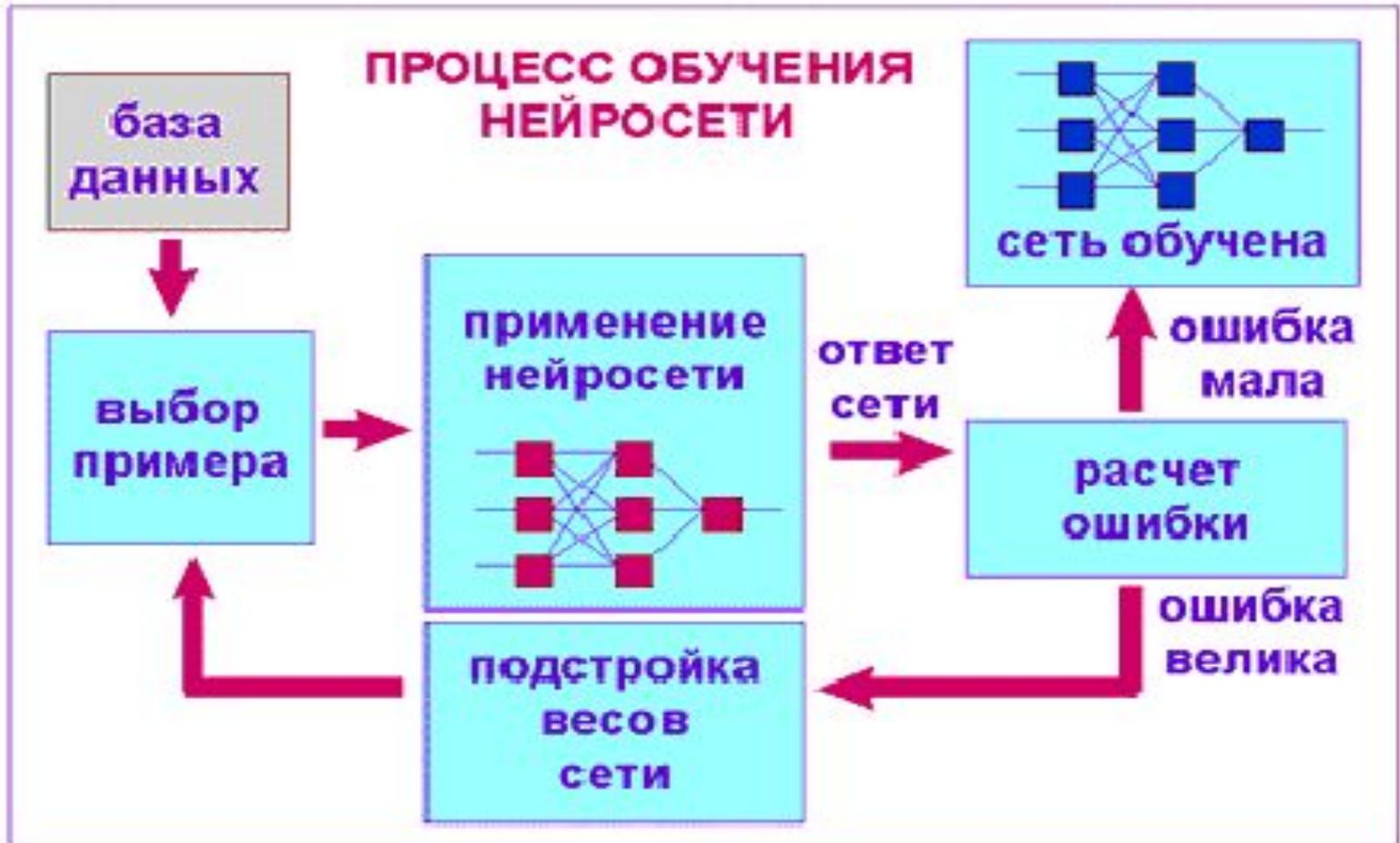
2

- Нейросистемы можно легко сделать очень устойчивыми к помехам и разрушениям;

3

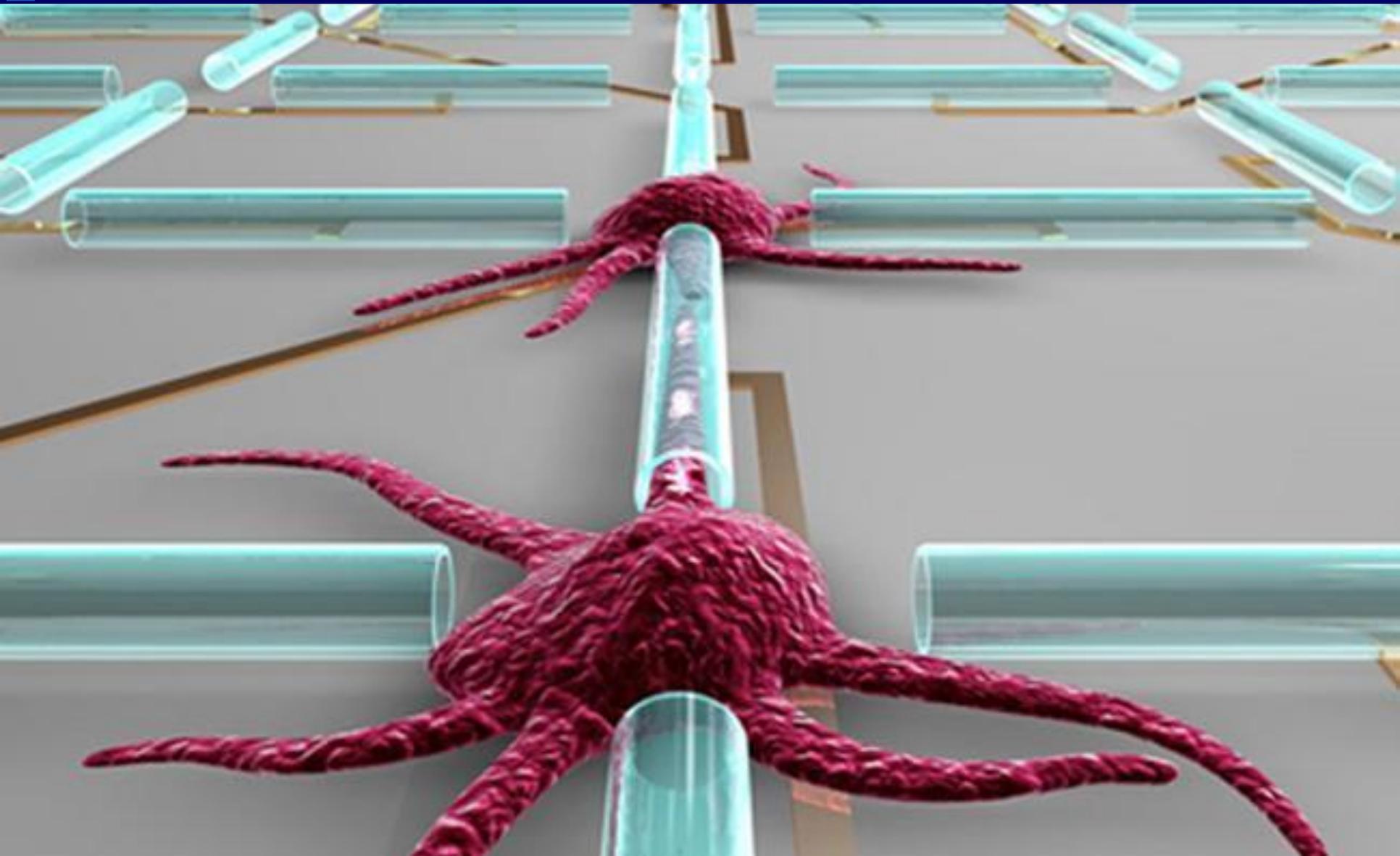
- Устойчивые и надёжные нейросистемы могут создаваться и из ненадёжных элементов, имеющих значительный разброс параметров.

Обучение нейросети



Процесс обучения нейронной сети.

WetWare



Нейрокомпьютеры (новая идея)



В нейрокомпьютинге постепенно созревает новое направление, основанное на **соединении биологических нейронов с электронными элементами**. Эти разработки получили наименование **Wetware (англ.)** — «влажный продукт». В настоящее время уже существует технология соединения биологических нейронов со сверхминиатюрными полевыми транзисторами с помощью нановолокон (**Nanowire (англ.)**). В разработках используется современная нанотехнология. В том числе, для создания соединений между нейронами и электронными устройствами используются углеродные нанотрубки.



http://



www

internet

3. Интеллектуальный анализ данных и управление знаниями

Процесс аналитического исследования больших массивов информации



Технологии и методы анализа и интерпретации данных

В реальной ситуации практически невозможно проверить экономическую модель на стадии анализа и поэтому начальные результаты имеют характер *эвристик*, которые можно использовать в процессе *принятия решения*.

Методы анализа данных основываются на классических принципах **разведочного анализа данных (РАД)**

Технологии и методы анализа и интерпретации данных



Технологии и методы анализа и интерпретации данных

Хранилища данных - способ хранения больших многомерных массивов данных, который позволяет легко извлекать и использовать информацию в процедурах анализа. Эффективная архитектура хранилища данных должна быть организована таким образом, чтобы быть составной частью информационной системы управления.

Технологии оперативной аналитической обработки данных OLAP

Термин **OLAP** (или **FASMI** - быстрый анализ распределенной многомерной информации) обозначает методы, которые дают возможность пользователям многомерных баз данных в реальном времени генерировать описательные и сравнительные сводки ("views") данных и получать ответы на различные другие аналитические запросы.

Технологии оперативной аналитической обработки данных OLAP

Технологии OLAP были разработаны для анализа данных в системах баз данных с целью поддержки принятия решений и ориентированы, главным образом, на обработку **нерегламентированных интерактивных** запросов.

Технологии оперативной аналитической обработки данных OLAP

В многомерной модели данных база данных представляется в виде одного или нескольких кубов данных, называемых иногда *гиперкубами*. Такой куб имеет несколько независимых измерений, своего рода систему координат представляемого им многомерного пространства данных. Каждому измерению соответствует некоторый *атрибут*, характеризующий какое-либо качественное свойство данных.

Технологии оперативной аналитической обработки данных OLAP

На множестве значений некоторых атрибутов измерений (элементов) могут быть определены *иерархические отношения*. Например, для атрибута-времени может использоваться иерархия «годы — кварталы — месяцы», для атрибута-территории — «регион — город — район».

Технологии оперативной аналитической обработки данных OLAP

Наборы значений измерений по одному для каждого из них определяют точки куба, называемые *ячейками*. С ячейками ассоциируются значения различных других количественных атрибутов, называемых *показателями*.

Технологии оперативной аналитической обработки данных OLAP

Для целей анализа могут строиться *сечения куба данных* (называемые также его *проекциями*) путем фиксации значений различных наборов атрибутов-координат. Может также осуществляться *сжатие куба* на основе использования значений атрибутов измерений более высоких уровней иерархии и соответствующего *агрегирования* значений ассоциированных с ними показателей.

Технологии оперативной аналитической обработки данных OLAP

Возможна также и обратная операция **детализации данных**. Таким образом, возможен анализ данных с нужной степенью детализации. Для удобства восприятия данных в процессе анализа используются различные операции визуализации данных, в частности **вращение куба** путем изменения порядка измерений.

Технологии оперативной аналитической обработки данных OLAP

Город	Товар	Январь	Февраль	Март	Итого
Москва	Позиция 1	10	22	15	47
	Позиция 2	2	7	5	14
	Позиция 3	17	34	20	71
Итого		29	63	40	132
Рязань	Позиция 4	2	0	3	5
	Позиция 3	5	6	3	14
	Позиция 5	12	22	7	41
Итого		19	28	13	60
Владивосток	Позиция 1	7	7	5	19
	Позиция 5	10	12	15	37
	Позиция 2	2	3	0	5
Итого		19	22	20	61

Технологии оперативной аналитической обработки данных OLAP

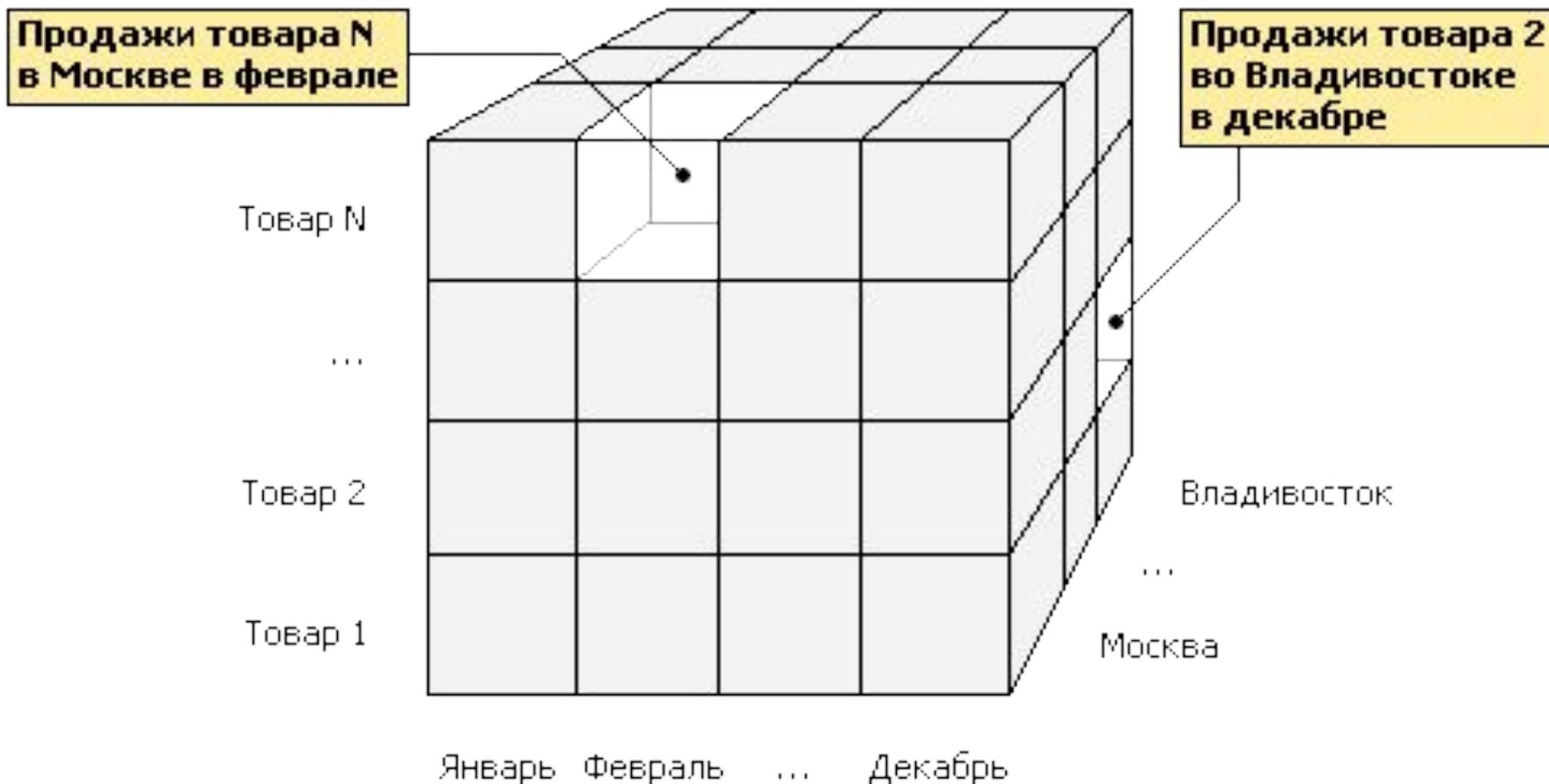


Рис. 1. Данные в трехмерном кубе

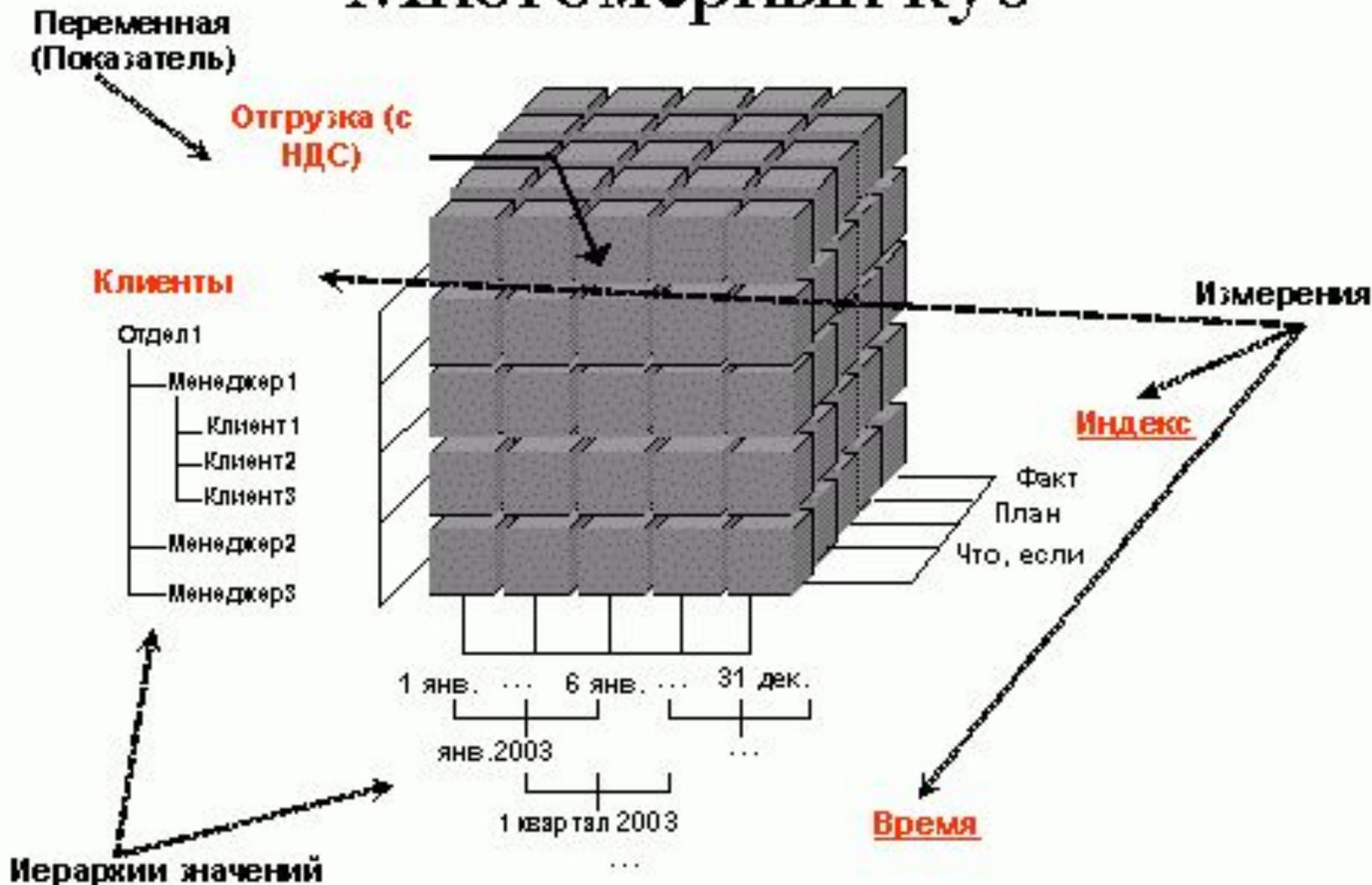
Технологии оперативной аналитической обработки данных OLAP

Сложим значения во всех ячейка по вертикали, то получим следующий отчет.

Город	Январь	Февраль	Март	Итого
Москва	29	63	40	132
Рязань	19	28	13	60
Владивосток	19	22	20	61
Итого	67	113	73	253

Технологии оперативной аналитической обработки данных OLAP

Многомерный куб



Технологии оперативной аналитической обработки данных OLAP

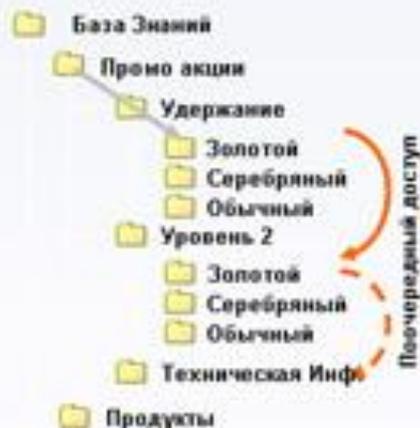
Технология InStranet

Контент распределен по размерностям многомерного куба для более быстрой навигации среди большого объема информации путем фильтрации.

Доступ пользователей к информации определяется распределением их прав.

Стандартная Структура

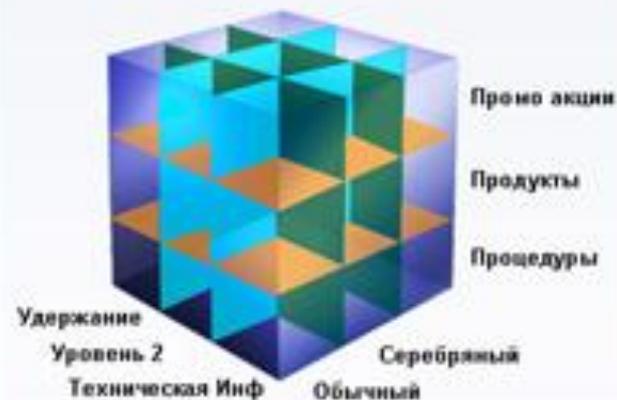
Легкий путь доступа к информации путем сортировки...



...но экстремально медленный

Многомерная Структура

Легкий путь доступа к любой информации **НА ЛЕТУ!!!**



Специфика систем **глубинного анализа данных** состоит в том, что пользовательские **запросы** не только имеют, как правило, нерегламентированный характер, но и, в отличие от запросов в OLAP, **нечетко формулируются.**

Хотя **методы добычи данных** можно применять к любой, предварительно не обработанной и даже неструктурированной информации, их можно также использовать **для анализа данных и отчетов, полученных средствами OLAP**, с целью более углубленного исследования, как правило, в более высоких размерностях.

Методы построения математических моделей в Data mining



Основные задачи решаемые Data Mining

классификация

кластеризация

выявление ассоциаций

поиск типовых образцов на заданном множестве

выявление объектов данных, не соответствующих характеристикам и поведению, общим для всех рассматриваемых данных

моделирование тенденций во временных рядах

Разведочный анализ данных (РАД)
применяется для нахождения связей
между переменными в ситуациях,
когда отсутствуют (или недостаточны)
априорные представления о природе
этих связей

Разведочный анализ данных (РАД)

Процедура анализа
распределений переменных

Просмотр корреляционных
матриц

Анализ многовходовых
таблиц частот

**Основные
методы**

Анализ распределений переменных

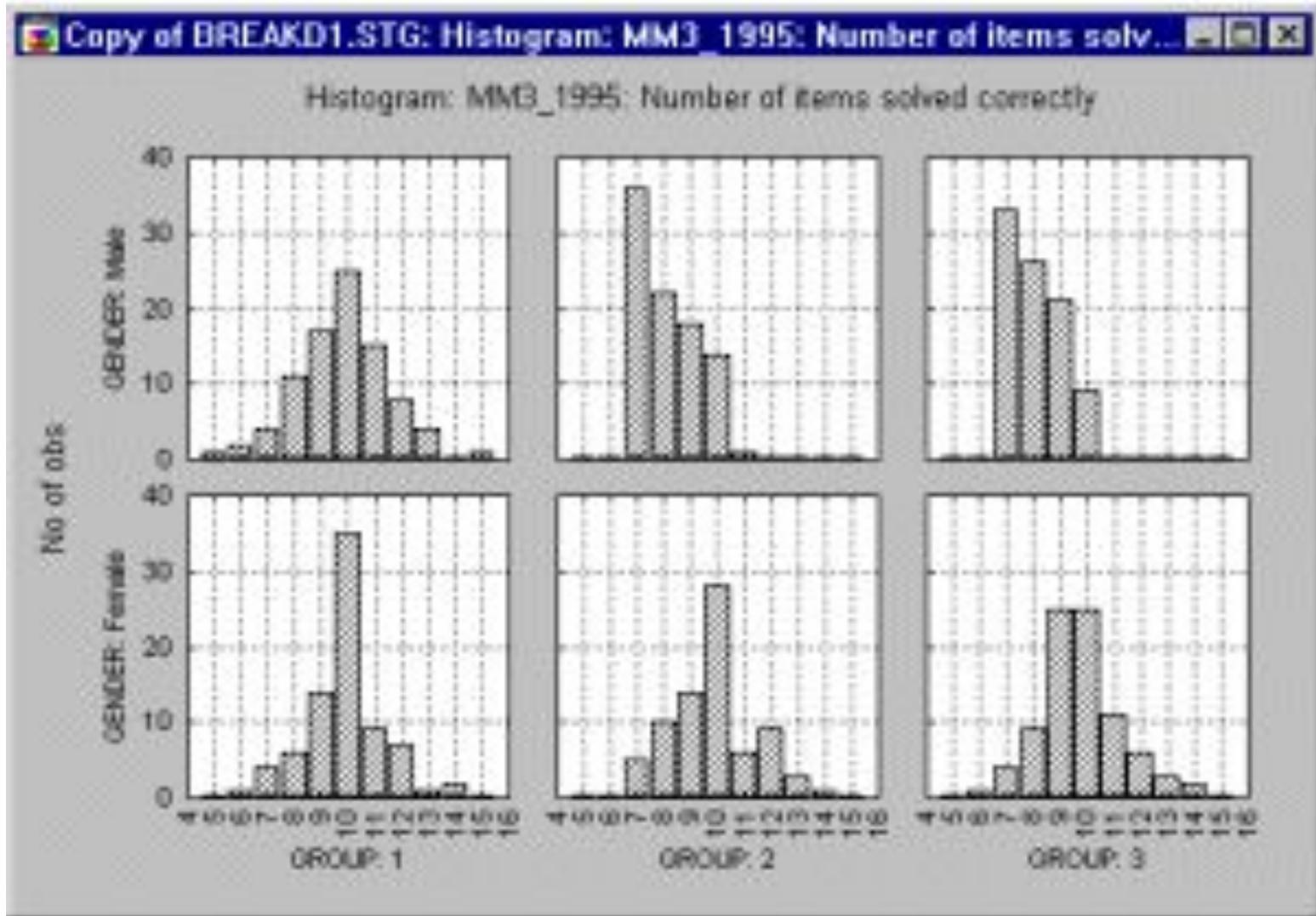
Важным способом описания переменной является **форма ее распределения**, которая показывает, с какой частотой значения переменной попадают в определенные интервалы. Эти интервалы, называемые *интервалами группировки*, выбираются исследователем. Обычно исследователя интересует, насколько точно распределение можно аппроксимировать **нормальным**.

Если асимметрия (показывающая отклонение распределения от симметричного) существенно отличается от 0, то распределение несимметрично, в то время как нормальное распределение абсолютно симметрично. Итак, у симметричного распределения асимметрия равна 0. Асимметрия распределения с длинным правым хвостом положительна.

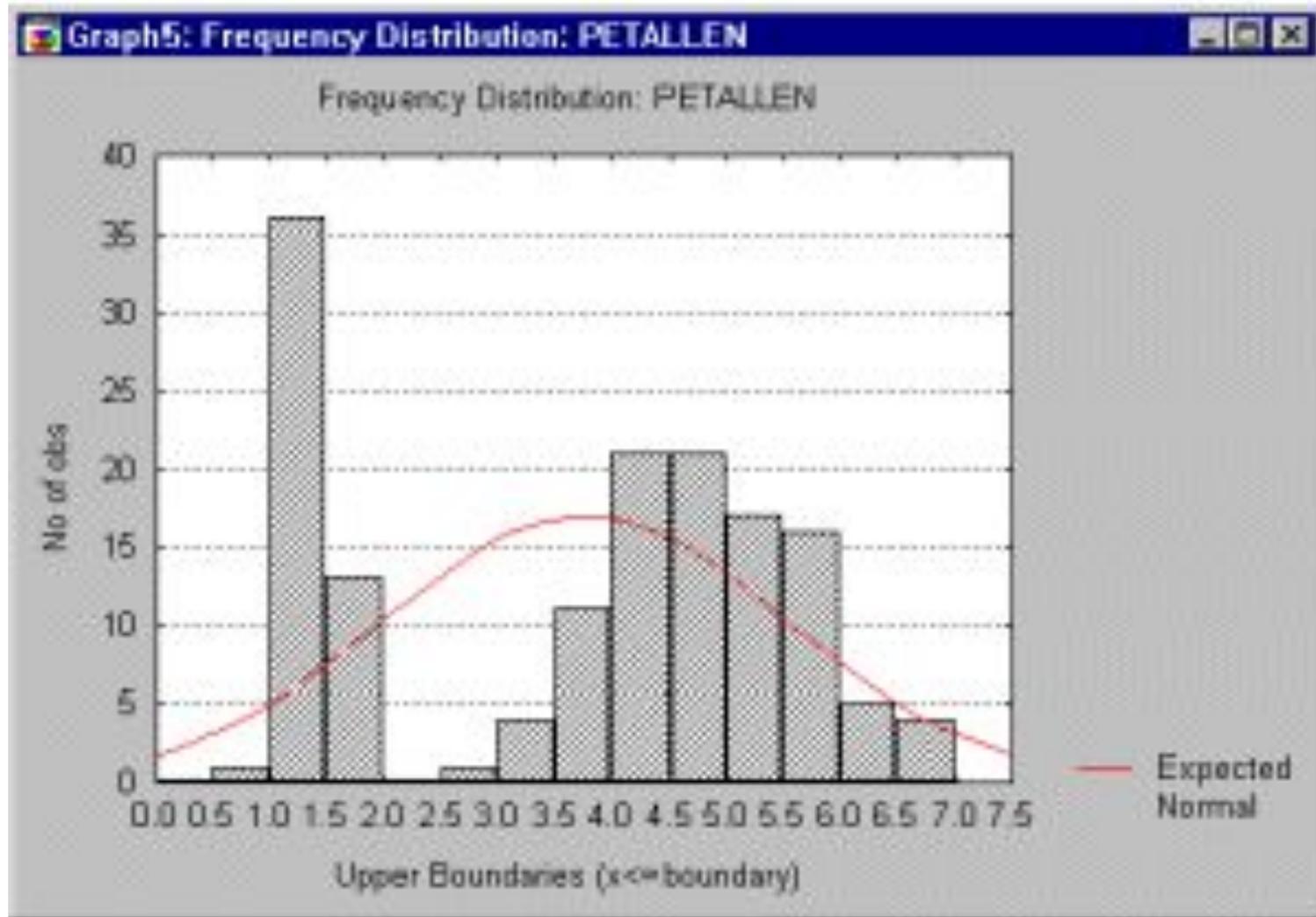
Анализ распределений переменных

Если распределение имеет длинный левый хвост, то его **асимметрия отрицательна**. Далее, если **эксцесс** (показывающий "остроту пика" распределения) существенно **отличен от 0**, то распределение имеет или более закругленный пик, чем нормальное, или, напротив, имеет более острый пик (возможно, имеется несколько пиков). Обычно, **если эксцесс положителен, то пик заострен**, если **отрицательный, то пик закруглен**. Эксцесс нормального распределения равен 0.

Анализ распределений переменных

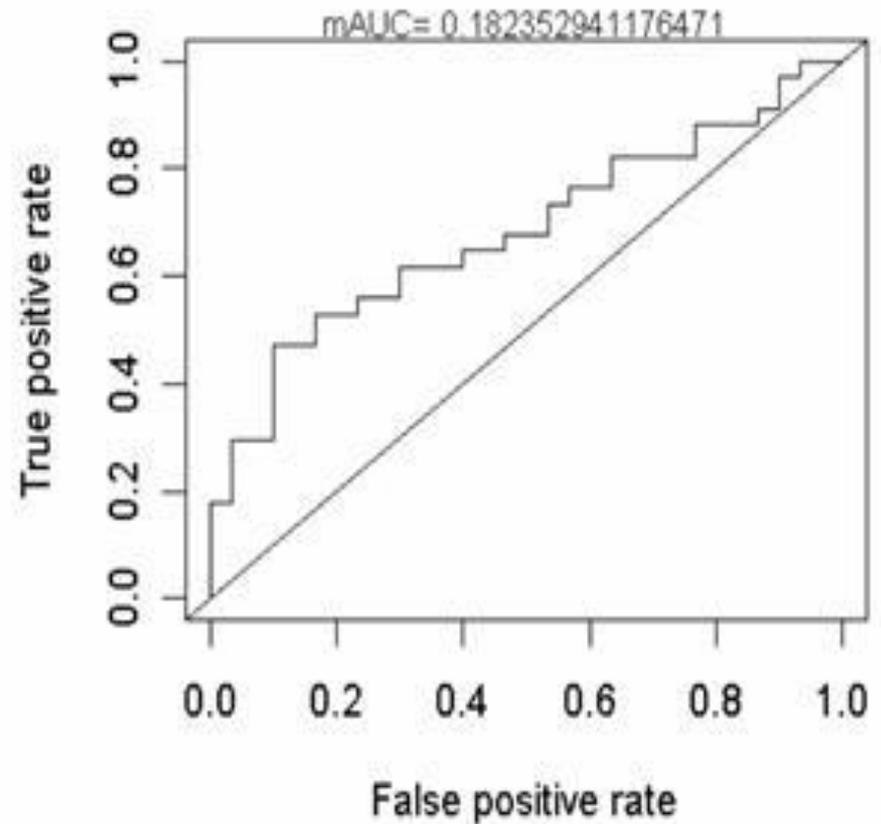
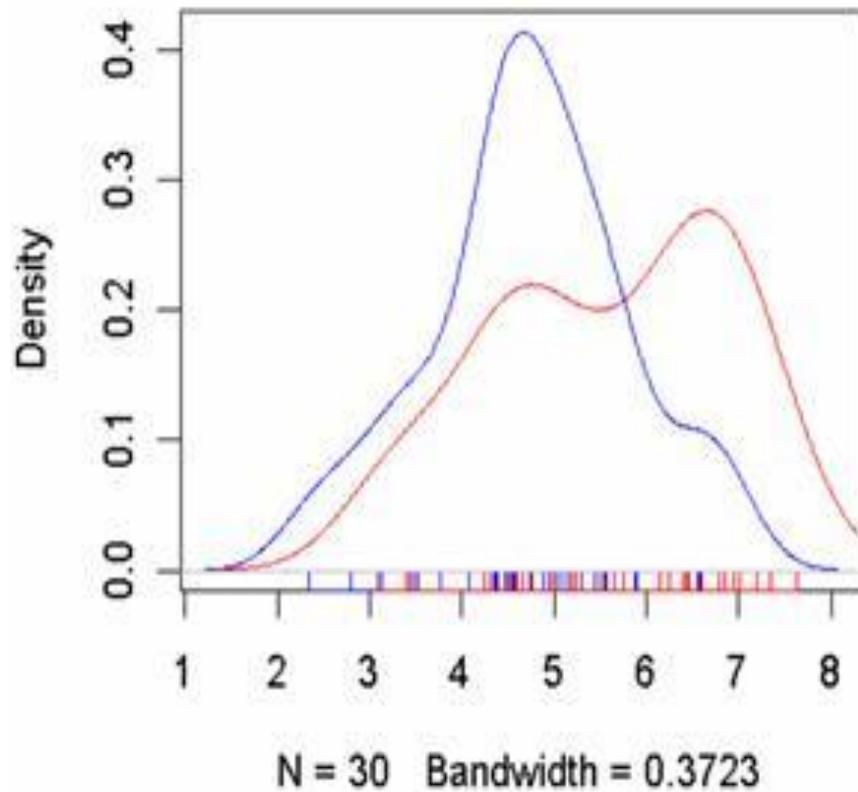


Анализ распределений переменных



Анализ распределений переменных

A_2174_28



Анализ распределений переменных



Разведочный анализ корреляционных матриц

Корреляция представляет собой меру зависимости переменных. Наиболее известна корреляция Пирсона. При вычислении корреляции Пирсона предполагается, что переменные измерены, как минимум, в интервальной шкале. Некоторые другие коэффициенты корреляции могут быть вычислены для менее информативных шкал.

Разведочный анализ корреляционных матриц

Коэффициенты корреляции изменяются в пределах от -1.00 до $+1.00$. Значение -1.00 означает, что переменные имеют строгую отрицательную корреляцию. Значение $+1.00$ означает, что переменные имеют строгую положительную корреляцию. Значение 0.00 означает отсутствие корреляции.

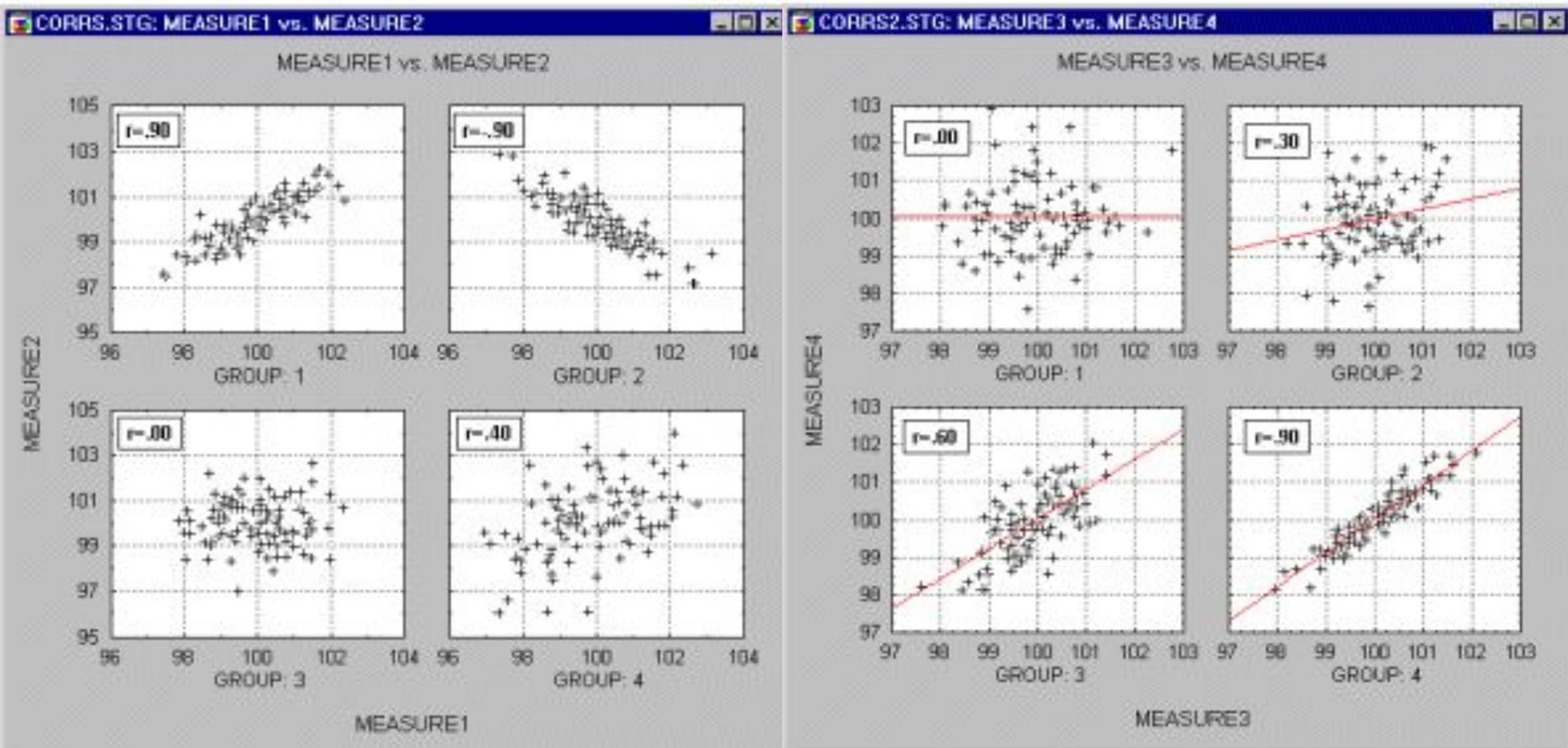
Разведочный анализ корреляционных матриц

Наиболее часто используемый коэффициент корреляции Пирсона (r) называется также линейной корреляцией, т.к. измеряет степень линейных связей между переменными. Важно, что значение коэффициента корреляции не зависит от масштаба измерения.

Разведочный анализ корреляционных матриц

Корреляция высокая, если на графике зависимость выражена прямой линией (с положительным или отрицательным углом наклона). **Интервальная шкала** позволяет не только упорядочить наблюдения, но и количественно выразить расстояния между ними (на шкале не обязательно присутствует абсолютная нулевая отметка).

Разведочный анализ корреляционных матриц



Разведочный анализ корреляционных матриц

Проведенная прямая называется **прямой регрессии** или прямой, построенной **методом наименьших квадратов**. Последний термин связан с тем, что сумма **квадратов** расстояний (вычисленных по оси Y) от наблюдаемых точек до прямой является минимальной. Заметим, что использование **квадратов** расстояний приводит к тому, что оценки параметров прямой сильно реагируют на выбросы.

Разведочный анализ корреляционных матриц

Выбросы. По определению, выбросы являются нетипичными, резко выделяющимися наблюдениями. Так как при построении прямой регрессии используется сумма *квадратов* расстояний наблюдаемых точек до прямой, то выбросы могут существенно повлиять на наклон прямой и, следовательно, на значение коэффициента корреляции. Поэтому единичный выброс (значение которого возводится в квадрат) способен существенно изменить наклон прямой и, следовательно, значение корреляции.

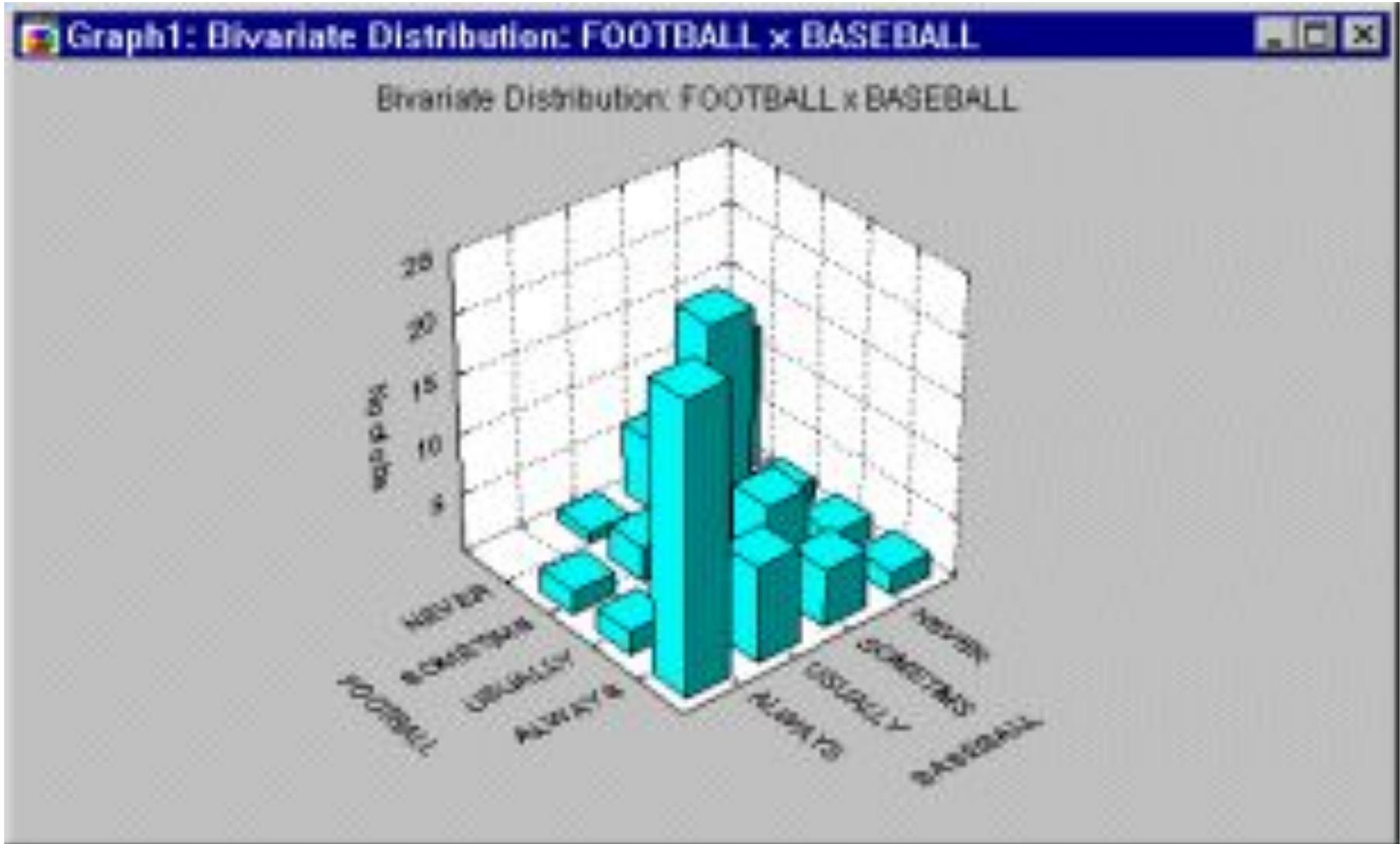
Анализ таблиц частот

Кросстабуляция - это процесс объединения двух (или нескольких) таблиц частот так, что каждая ячейка (клетка) в построенной таблице представляется единственной комбинацией значений или уровней табулированных переменных. Таким образом, кросстабуляция позволяет совместить частоты появления наблюдений на разных уровнях рассматриваемых факторов.

Анализ таблиц частот

В целях исследования отдельные строки и столбцы таблицы удобно представлять в виде графиков. Полезно также отобразить целую таблицу на отдельном графике. Таблицы с двумя входами можно изобразить на 3-мерной гистограмме.

Анализ таблиц частот



Анализ таблиц частот

Другой способ визуализации таблиц сопряженности - построение **категоризованной гистограммы**, в которой каждая переменная представлена индивидуальными гистограммами на каждом уровне другой переменной. Преимущество 3М гистограммы в том, что она позволяет представить на одном графике таблицу целиком. Достоинство категоризованного графика в том, что он дает возможность точно оценить отдельные частоты в каждой ячейке.

Методы многомерного разведочного анализа

Методы многомерного разведочного анализа специально разработаны для поиска закономерностей в многомерных данных (или последовательностях одномерных данных). К ним относятся: *кластерный анализ, факторный анализ, анализ дискриминантных функций, многомерное шкалирование, логлинейный анализ, канонические корреляции, пошаговая линейная и нелинейная регрессия, анализ соответствий, анализ временных рядов и деревья классификации.*

Кластерный анализ

Кластерный анализ. Термин в действительности включает в себя **набор различных алгоритмов классификации**. Общий вопрос состоит в том, как организовать наблюдаемые данные в наглядные структуры, т.е. **развернуть таксономии**.

Кластерный анализ является не столько обычным статистическим методом, сколько набором различных алгоритмов распределения объектов по кластерам.

Кластерный анализ

Независимо от предмета изучения применение кластерного анализа предполагает **следующие этапы**: — Отбор выборки для кластеризации. — Определение множества переменных, по которым будут оцениваться объекты в выборке. — Вычисление значений той или иной меры сходства между объектами. — Применение метода кластерного анализа для создания групп сходных объектов. — Проверка достоверности результатов кластерного решения.

Основные задачи решаемые Кластерным анализом

Разработка типологии или классификации

Исследование полезных концептуальных схем группирования объектов

Порождение гипотез на основе исследования данных

Проверка гипотез или исследования для определения, действительно ли типы (группы), выделенные тем или иным способом, присутствуют в имеющихся данных

Формальная постановка задачи кластеризации

Пусть X — множество объектов Y — множество номеров (имён, меток) кластеров. Задана функция расстояния между объектами $\rho(x, x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1, \dots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые *кластерами*, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ , а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i .

Алгоритм кластеризации — это функция $a: X \rightarrow Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного *критерия качества* кластеризации.

Кластерный анализ

не существует однозначно
наилучшего критерия
качества кластеризации

число кластеров неизвестно
заранее и устанавливается в
по субъективным критериям

результат кластеризации
зависит
от метрики, выбор которой
также субъективен

**Основные
недостатки**

Объединение (древовидная кластеризация)

Объединение (древовидная кластеризация). Назначение этого алгоритма состоит в объединении объектов в достаточно большие кластеры, используя некоторую меру сходства или расстояние между объектами. Типичным результатом такой кластеризации является иерархическое дерево.

Объединение (древовидная кластеризация)

Дерево классификации множества поставщиков



Объединение (древовидная кластеризация)

Кластерный анализ содержит эффективную двухходовую процедуру объединения. Однако **двухходовое объединение** используется (относительно редко) в обстоятельствах, когда ожидается, что и **наблюдения и переменные одновременно** вносят вклад в обнаружение осмысленных кластеров.

Объединение (древовидная кластеризация)

Можно предположить, что медицинскому исследователю требуется выделить **кластеры пациентов, сходных по отношению к определенным кластерам характеристик физического состояния**. Трудность с интерпретацией полученных результатов возникает вследствие того, что сходства между различными кластерами могут происходить из (или быть причиной) некоторого различия подмножеств переменных. Поэтому **получающиеся кластеры являются по своей природе неоднородными**.

Объединение (древовидная кластеризация)

В сравнении с другими описанными методами кластерного анализа, **двухходовое объединение является, наименее часто используемым методом.** Однако некоторые исследователи полагают, что он предлагает мощное средство разведочного анализа.

Метод К-средних (C-means)

Алгоритм разделительной кластеризации, основанный на разбиении множества элементов векторного пространства на заранее определенное число кластеров k . Алгоритм представляет собой итерационную процедуру, в которой выполняются **следующие шаги**:

Метод К-средних (C-means)

1. Выбирается число кластеров k .
2. Из исходного множества данных **случайным образом выбираются k записей, которые будут служить начальными центрами кластеров.**
3. Для каждой записи исходной выборки определяется ближайший к ней центр кластера. При этом **записи, «притянутые» определенным центром, образуют начальные кластеры.**

Метод К-средних (C-means)

4. Вычисляются **центроиды** – центры тяжести кластеров. Каждый центроид – это вектор, элементы которого представляют собой средние значения признаков, вычисленные по всем записям кластера. **Затем центр кластера смещается в его центроид.**

Метод К-средних (C-means)

Затем 3-й и 4-й шаги итеративно повторяются. Очевидно, что **на каждой итерации происходит изменение границ кластеров и смещение их центров**. В результате минимизируется расстояние между элементами внутри кластеров. **Остановка алгоритма производится тогда, когда границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации**, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор записей. Алгоритм обычно находит набор стабильных кластеров за несколько десятков итераций.

Метод К-средних (C-means)

Преимуществом алгоритма являются быстрота и простота реализации. К его **недостаткам** можно отнести неопределенность выбора начальных центров кластеров, а также то, что число кластеров должно быть задано изначально, что может потребовать некоторой априорной информации об исходных данных.



http://

@

www

internet

3.1. KNOWLEDGE MANAGEMENT

KNOWLEDGE MANAGEMENT

Поддержка эффективности любой проектной деятельности лежит в основе формирования задач современных информационных технологий, способствующих развитию перспективных направлений, связанных с решением проблем **идентификации предметной области и концептуального моделирования, анализа и извлечения данных, принятия решений и управления знаниями, адаптации и самоорганизации.**

KNOWLEDGE MANAGEMENT

Мнения ученых в прогнозных исследованиях показывают, что «...до 2020 года количество информации и потребности в ней будут расти **экспоненциально**...». Без технологий поиска и обработки таких объемов информации в будущем аналитики, эксперты и лица, принимающие решения, будут введены в состояние, которое можно назвать "аналитическим параличом"...

KNOWLEDGE MANAGEMENT

Таким образом, одной из актуальных проблем современной науки является «информационное переполнение». По мнению ученых, одним из основных способов решения проблемы информационного переполнения является переход от хранения и обработки данных к **накоплению и обработке знаний**.

KNOWLEDGE MANAGEMENT

Накопление и обработка знаний – сложные многоэтапные процессы, требующие соответствующего управления для обеспечения необходимой последовательности, согласованности и эффективности.

Отталкиваясь от того мнения, что знания не существуют в готовом виде, сформулируем

аксиому 1: Новые знания непрерывно производятся в процессе обработки знаний.

Управление знаниями –
непрерывное управление процессами
проверки знаний для выявления
новых проблем с целью создания и
удовлетворения спроса на новые
знания.

KNOWLEDGE MANAGEMENT

Сформулируем **концепцию** интеллектуальной поддержки принятия решений, основанную на процессах цикла управления знаниями. **Выявление новых знаний связано с изучением и поиском отношений между разнородными источниками знаний, так как именно междисциплинарный и мультидисциплинарный подходы накопления информации с большей вероятностью после обработки приведут к продуцированию нового знания.**

KNOWLEDGE MANAGEMENT

На основе рассмотренных процессов циклического сценария управления знаниями сформулируем **аксиому 2: Цель управления знаниями состоит в повышении их прозрачности.** Абстрактность данной аксиомы подразумевает определение широкого спектра задач, основным критерием эффективности решения которых будет прозрачность процессов сценария управления знаниями. Среди таких задач можно выделить: **идентификацию, приобретение, совершенствование, распределение, использование и сохранение** знаний.

IT как субъект эволюции



Информационные системы как эволюционирующие «организмы», совокупность которых образует некоторое связанное пространство, представляющее собой аналог биосферы

Признаки эволюции

1

■ «борьба за существование»;

2

■ «видовое изменение»;

3

■ «естественный отбор».

Различие биологической и информационной эволюций



Механизмы развития живого организма, основанного на копировании ДНК, сильно напоминают обычное компьютерное копирование

Подобие биологических и информационных «организмов»

Состав (типизация, дискретность, детерминированность)

Организация (глобальные сети, современное ПО)

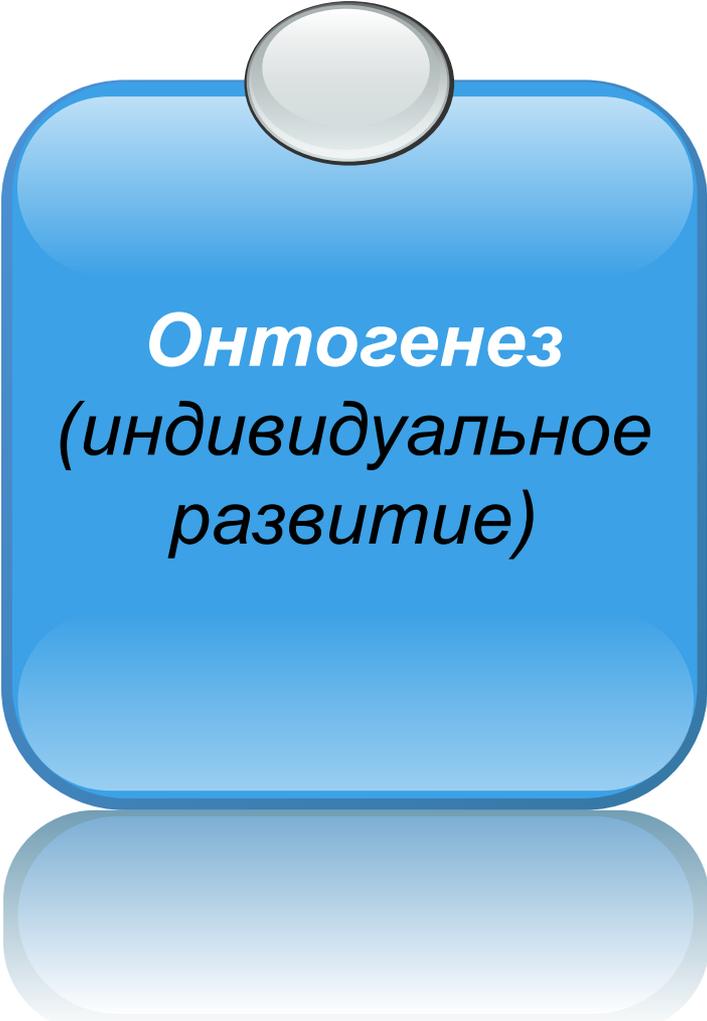
Обмен веществом и энергией (информация, ресурсы)

Раздражимость и психические функции

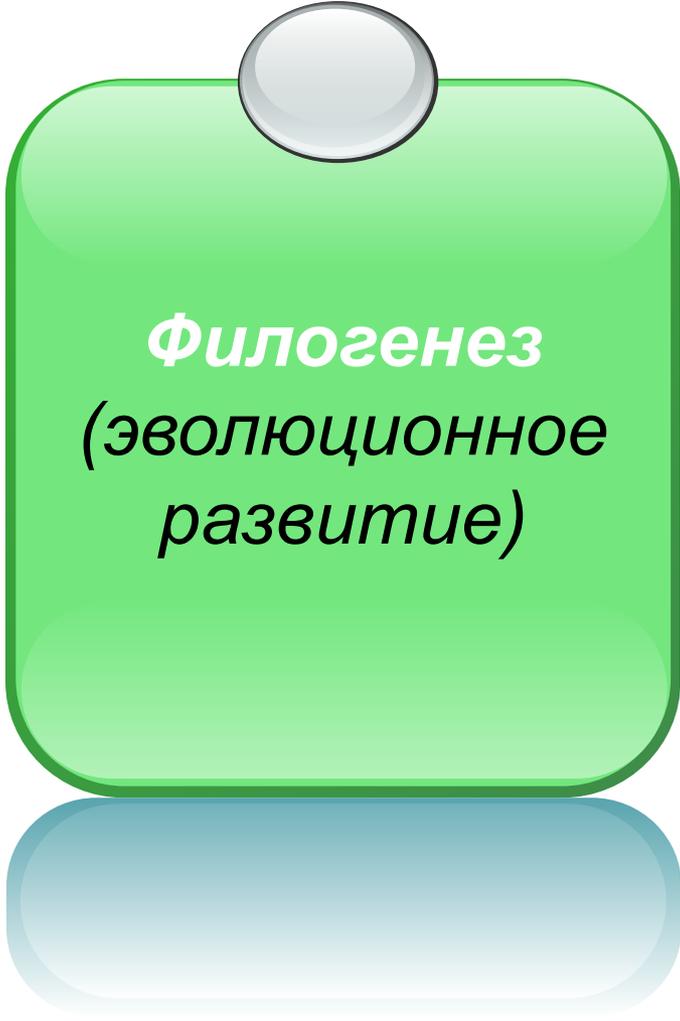
Гомеостаз (регуляторные системы) (целостность)

Наследование

Подобие биологических и информационных «организмов»

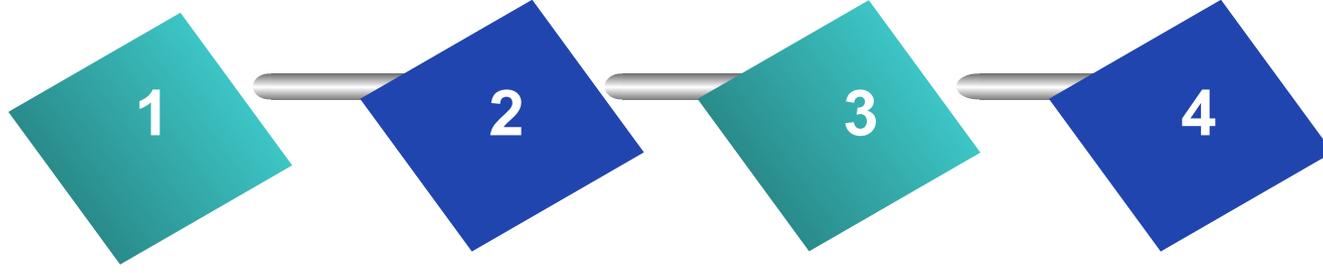


Онтогенез
*(индивидуальное
развитие)*



Филогенез
*(эволюционное
развитие)*

Фазы эволюционного процесса



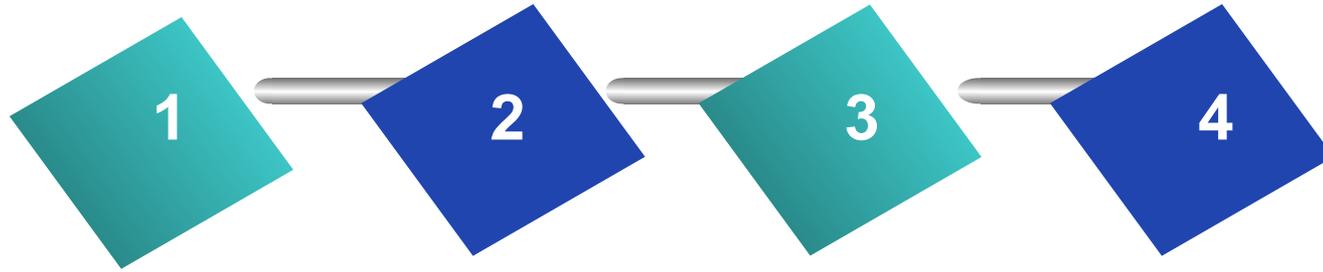
**Образование
базовой
ветви**

**«Взрыв»
видового
разнообрази
я**

**Отбор наиболее
жизнеспособны
х видов**

**Возникновение
новой базовой
ветви**

Компоненты эффективности КИС



СИГНАЛИНГ

КООРДИНАЦИЯ

КОНТРОЛЬ

АДАПТАЦИЯ

**Своевременность
формирования
управляющих
сигналов**

**Возможность
координации
исполнения
задач**

**Контроль
качества
исполнения
задач**

**Адаптация
жизненного
цикла при
появлении
новых задач**

«Портрет» активной КИС

Система, обеспечивающая управление, должна быть **активной**, т.е. сигнализировать о состоянии тех или иных процессов (участков) и их показателей в случае отклонения от заданного диапазона.

«Портрет» активной КИС

Активность системы – принципиально важный аспект. Такая система хранит «представление» об идеальной модели процессов. **В перспективе такая система будет «самостоятельно» осуществлять управление необходимыми процессами, входящими в спектр функций системы.**

«Портрет» активной КИС

Следующее свойство подобной системы –
обработка событий.

Логика системы будет выглядеть как
**структура возможных событий с ярко
выраженной временной составляющей.**

**Обязательно наличие средств
прогнозирования развития объекта
управления и проработки альтернативных
вариантов планирования.**

«Портрет» активной КИС



Выделение в «идеальной» модели областей принятия решений при достижении критических показателей или других совокупных условий позволит достичь **необходимой скорости реакции системы на внешние или внутренние события (воздействия)**. При этом **обратные связи позволят динамически модифицировать модель жизненного цикла**

Проблемы реализации



Эволюционная логика утверждает, что такую систему нельзя купить, ее можно только «вырастить» параллельно с организацией процессов для управления которыми она создается!

Объемы данных не определяют уровень информированности, манипулирование потоками исполнения задач не заменяет организации, а алгоритмы, в которых детерминированы условия и результаты, не являются дисциплиной!

Синергетика и устойчивость систем



Синергетика утверждает, что степень устойчивости системы зависит от сложности ее организации. **Сложные системы в целом устойчивее (ИНС).** Если система начинает «умирать», то на самом первом этапе это реализуется как упрощение состава системы, ее организации и регуляторных функций. Т.е. «умирает» целостность системы, подменяемая конкурирующими элементами.

Самоорганизация как аспект развития



Информационная система – не конвейер обработки данных, а среда оперирования знанием. **Самоорганизация** – ключевой аспект эволюционного развития и приобретения конкурентных преимуществ. *Именно через организацию работ реализуется* управление и его функции, поддерживаются важнейшие характеристики системы, происходят **технические революции.**

Составляющие организационного решения для ИС

Технологическая – организация пространства решений предметных задач, формализация процессов и сценариев управления

Коммуникативная – организация пространства использования системы (отношения, права доступа, защита информации)



Операционная – организация пространства для эффективного накопления и обработки знаний (системные и сетевые решения, инструменты проектирования)

Логика эволюционного развития для ИС



Логика эволюционного развития информационных технологий позволяет вести непрерывную модернизацию ИС, когда опережающее развитие модели процессов будет формировать задачи информационной поддержки, а логика развития ИС будет требовать модернизации модели процессов.

Автоматизация управления знаниями



Проблема автоматизации управления знаниями, как непрерывного процесса проверки знаний для выявления закономерностей с целью создания и удовлетворения спроса на новые знания, напрямую связана с решением задачи **структуризации информации.**

Проблема информационного переполнения



Проблема информационного переполнения возникает в среде с интенсивным обменом и информационными потоками. Одним из **следствий тенденции к децентрализации информационных ресурсов** является постепенное усиление информационной активности субъектов информационного поиска и обмена. В этих условиях **сеть гетерогенных информационных ресурсов, задействованная в обмене информацией, неизбежно проходит через "точку роста", в которой привлекательность результатов накопления и обработки знаний падает**, т.к. пользователи начинают испытывать трудности информационного переполнения.

СУЗ как развитие концепции ИС



Под **системой управления знаниями (СУЗ)** будем понимать совокупность **элементов**, таких как модели, методы и алгоритмы накопления, синтеза, обработки, обмена, хранения знаний, и **отношений** между ними, реализующих повторяемые на регулярной основе управленческие действия и процедуры, призванные повысить эффективность процессов идентификации, приобретения, совершенствования, распределения и использования знаний в едином информационном пространстве, созданном на основе интеграции информационных моделей различных предметных областей.

Особенность и функции СУЗ

Отличительной **особенностью** подобных систем управления знаниями является **интеграция множеств разнородных знаний из различных предметных областей** для обеспечения необходимой адаптивности при управлении процессами извлечения знаний.

Основной **функцией** систем управления знаниями является **управление информационными потоками как совокупностью процессов систематической идентификации, приобретения, совершенствования, распределения и использования знаний в едином междисциплинарном пространстве.**

Функции СУЗ

1

- фиксация и хранение явных и неявных знаний;

2

- классификация и агрегация знаний;

3

- накопление, систематизация и обновление знаний;

Функции СУЗ

4

- распространение знаний, в соответствии с установленным регламентом;

5

- разведка знаний, семантический поиск и навигация по знаниям;

6

- поддержка процессов генерации, апробации и использования новых идей в решении задач управления знаниями;

7

- обеспечение поддержки процесса принятия решений

Архитектура СУЗ



Архитектура – это организационная структура системы, определяющая разбиение системы на части, связи между этими частями, механизмы взаимодействия и основные руководящие принципы для проектирования системы, а также множество значимых решений относительно принципов построения системы.

Состав архитектуры ИС

1

**структурные
элементы и
интерфейсы
связей между ними**

2

**крупномасштабная
организация
структурных
элементов и
определение
топологии их связей**

3

**важные
механизмы,
доступные всей
системе**

4

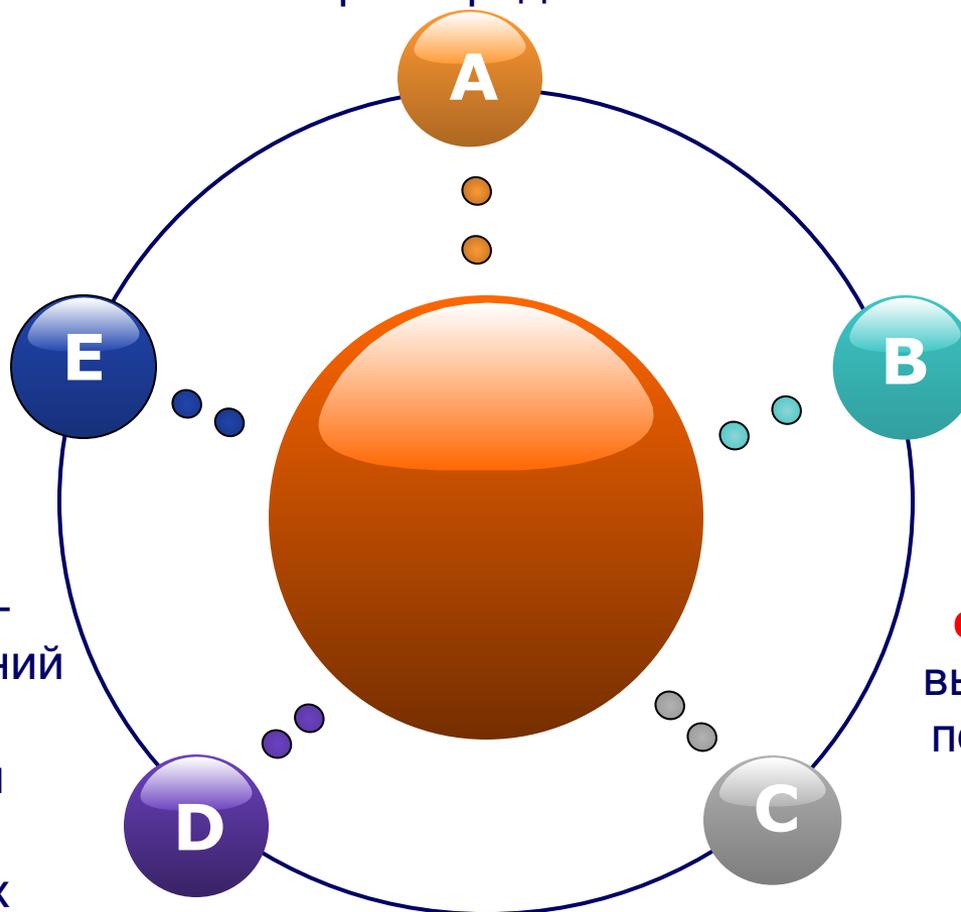
**архитектурный
стиль, который
управляет
организацией
элементов
системы**

Основные этапы накопления и обработки знаний в СУЗ

накопление – выявление неформализованных знаний из разнородных источников

обработка – преобразование знаний, создание унифицированных хранилищ данных

формализация – представление знаний на основе онтологических и семантических информационных моделей



классификация – идентификация предметной области

структуризация – выделение основных понятий, разработка структуры представления информации

Архитектура СУЗ



Ключевым модулем СУЗ является **поисковый модуль**, обеспечивающий оперативное извлечение и доставку элементов знаний по пользовательским запросам. Поиск извлекаемых знаний производится через **модуль доступа** к разнородным источникам знаний, *оценка релевантности* накопленных знаний происходит на основе оценки **модулем онтологии** их соответствия онтологической модели исследуемой предметной области. Для этого используется *комбинация синтаксического и семантического поиска* в экземплярах онтологии с учетом их семантических свойств и связей.

Архитектура СУЗ



Понятия онтологии, предназначенные для поддержки решения задач накопления и обработки знаний, должны быть связаны со значениями терминов предметной области.

Задачи классификации, структуризации, формализации и обработки знаний решают одноименные модули (рис).

Систему управления знаниями можно рассматривать как некоторую надстройку над интеллектуальной информационной системой (ИИС), развивающую ее функциональность.

Архитектура СУЗ



Поддержку семантического поиска релевантных знаний в подобных ИСУЗ предлагается реализовать на основе онтологических моделей. Усовершенствование поисковых запросов производится на основе анализа информации из информационных **энциклопедических справочных систем** различной функциональности и **баз прецедентов уже получивших эффективную реализацию** запросов. Разрабатываемая модифицированная архитектура ИСУЗ поддерживает реализацию двух подходов к повышению эффективности процедуры извлечения (накопления) знаний:

Повышение эффективности процедуры накопления знаний

1

извлечение
знаний на основе
онтологических
моделей и
энциклопедическ
их справочных
систем

2

извлечение
знаний на
основе case-
моделей и
прецедентов
поисковых
запросов



Модуль онтологии в качестве основы для проектирования семантических отношений использует *модели исследуемых предметных областей, построенных с помощью открытого стандарта конструкции онтологии OWL*.
Обобщающие тематические термины и синонимы модуль онтологии получает из энциклопедических справочных систем.

Архитектура СУЗ



Для некоторых источников знаний требуются **справочные системы** с расширенными функциями. Обновление модели рассматриваемой предметной области происходит на основе изменения и введения новых понятий и отношений, полученных с помощью дополнений и уточнений, которые могут добавляться модульным способом.

Архитектура СУЗ

Имея доступ к справочным и специальным знаниям для данного понятия или предметной области, модуль онтологии, получая первую редакцию поискового запроса, передает пользователю через модуль редактирования запроса на выбор ряд вариантов синонимов и уточнения исследуемого понятия предметной области, а также информацию о имеющихся междисциплинарных отношениях данного понятия с элементами знаний из других предметных областей. Подобная семантическая обработка позволяет усовершенствовать запрос на основе моделей онтологий предметных областей и энциклопедических справочных системам.

Архитектура СУЗ

С учетом неоднородности и распределенности источников знаний, для более полного накопления знаний, необходимым становится **разбиение измененного поискового запроса на подзапросы**, решение этой задачи производит **модуль редактирования запросов**. Задачу распределения полученных подзапросов по различным источникам знаний решает **модуль веб-сервиса**. **Модуль ранжирования** отвечает за сбор результатов подзапросов от различных источников знаний, оценивая их по семантическим критериям в соответствии с пользовательскими настройками и требованиями поискового запроса.

Архитектура СУЗ



Для повышения эффективности процедуры изменения поискового запроса помимо модуля онтологии будем использовать **case-модуль**, имеющий доступ хранилищам **эталонных case-моделей** и **прецедентов поисковых запросов**. К репозиторию прецедентов успешных поисковых запросов также имеют доступ **модули редактирования запросов, пользовательских настроек и веб-сервиса**. Это позволит проводить оперативное сравнение полученных изменений с имеющимися успешными прецедентами.

Архитектура СУЗ



Case- модуль на основе сравнительных данных от модуля редактирования позволит сформулировать успешный прецедент поискового запроса на основе эталонной case- модели. Использование эталонной структуры запроса, основанной на применении современных стандартов информационного моделирования (IDEF0, ARIS и т.п.), будет способствовать повышению качества результатов поискового запроса.



http://

@

www

internet

4. Проблемы компьютерного моделирования сложных систем

Понятие модели



Модель – представление объекта, системы или понятия в некоторой форме, отличной от реального существования, с целью изучения оригинала или воспроизведения его каких - либо свойств.

Модель может быть **точной копией** какого-либо объекта (хотя и в другом масштабе и из другого материала) или сохранять лишь часть значимых при постановке задачи его свойств. Во втором случае модель является **абстракцией** объекта .

Компьютерная модель



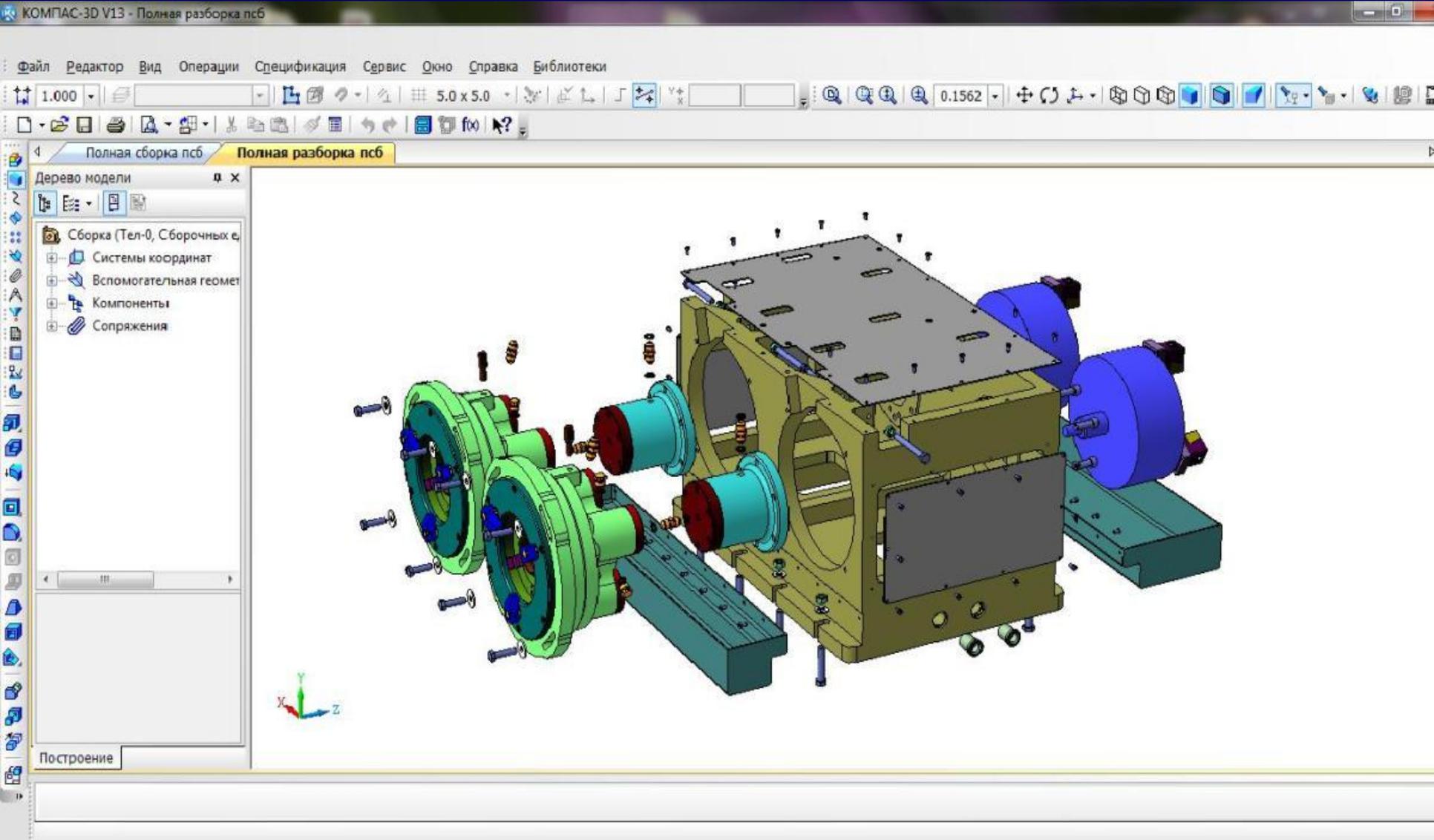
Компьютерная модель – компьютерная программа, работающая на отдельном компьютере, суперкомпьютере или множестве взаимодействующих компьютеров (вычислительных узлов), реализующая представление объекта, системы или понятия в форме, отличной от реальной, но приближенной к алгоритмическому описанию, включающей и набор данных, характеризующих свойства системы и динамику их изменения со временем.



Компьютерное моделирование – это метод решения задачи анализа или синтеза сложной системы на основе использования ее компьютерной модели.

Суть компьютерного моделирования заключена в получении количественных и качественных результатов на основе имеющейся модели

Копия



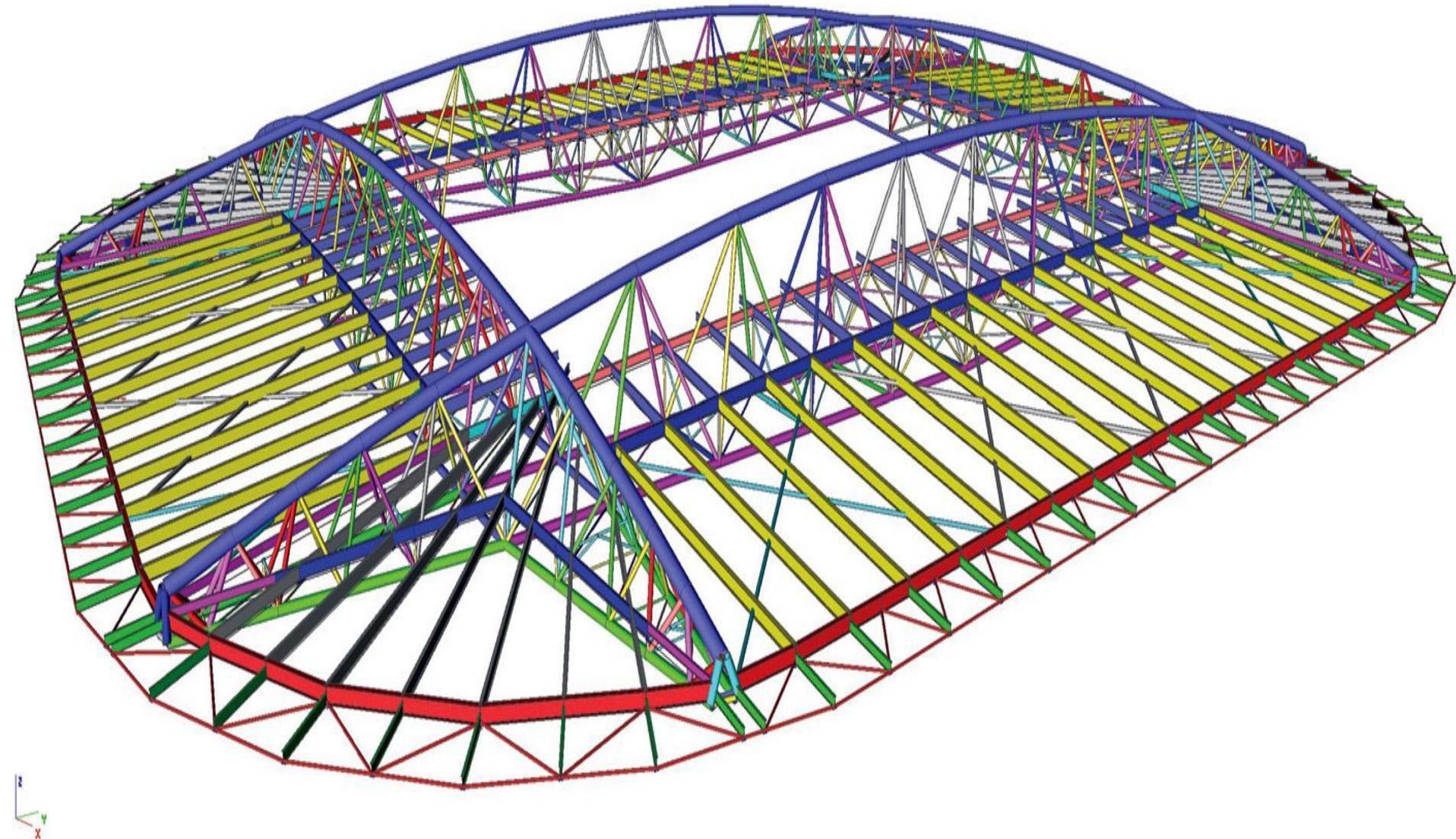
Абстракция



Абстракция



Абстракция



Методология компьютерного моделирования



Системный анализ (направление кибернетики, общая теория систем), в котором доминирующая роль отводится системным аналитикам.

В отличие от математического моделирования на ЭВМ, где методологической основой являются: исследование операций, теория математических моделей, теория принятия решений, теория игр и др.

Методология компьютерного моделирования



Центральной процедурой системного анализа является построение обобщенной модели, отражающей все факторы и взаимосвязи реальной системы.

Предметом компьютерного моделирования может быть любая сложная система, любой объект или процесс. **Компьютерная модель должна отражать** все свойства, основные факторы и взаимосвязи реальной сложной системы, критерии, ограничения.

Сложная система

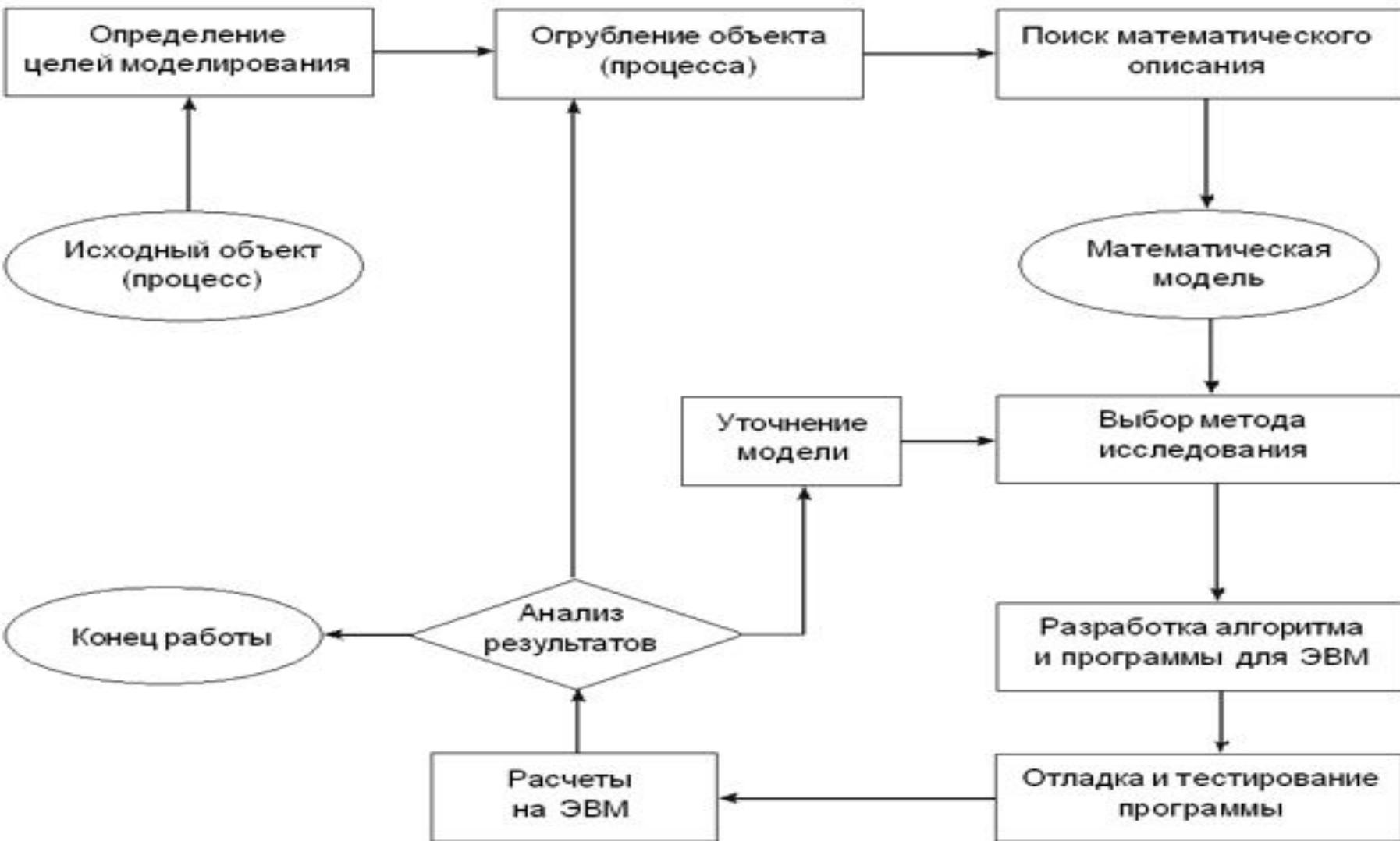


Сложная система — составной объект, части которого можно рассматривать как системы, закономерно объединенные в единое целое в соответствии с определенными принципами или связанными между собой заданными отношениями.

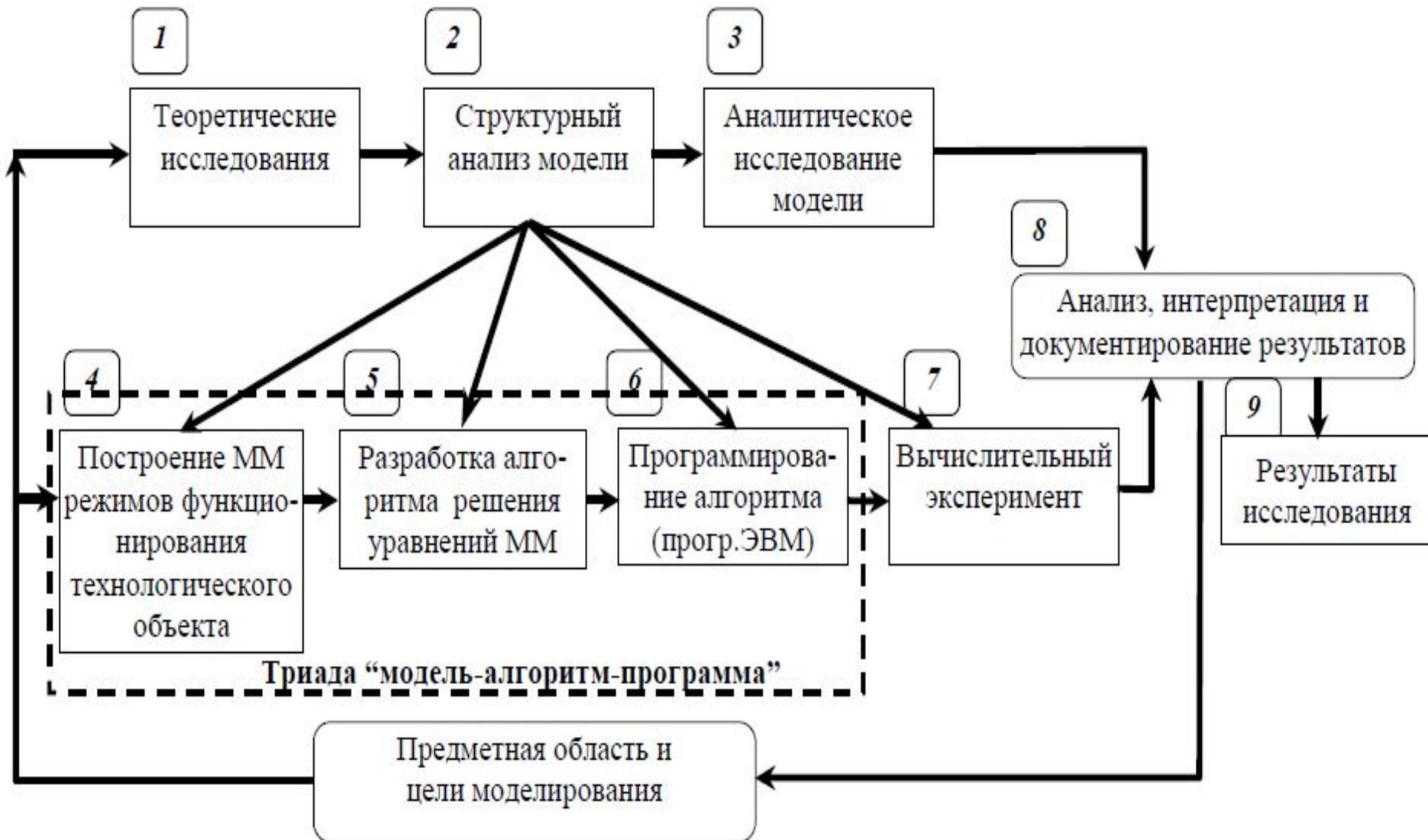
Циклический сценарий КМ



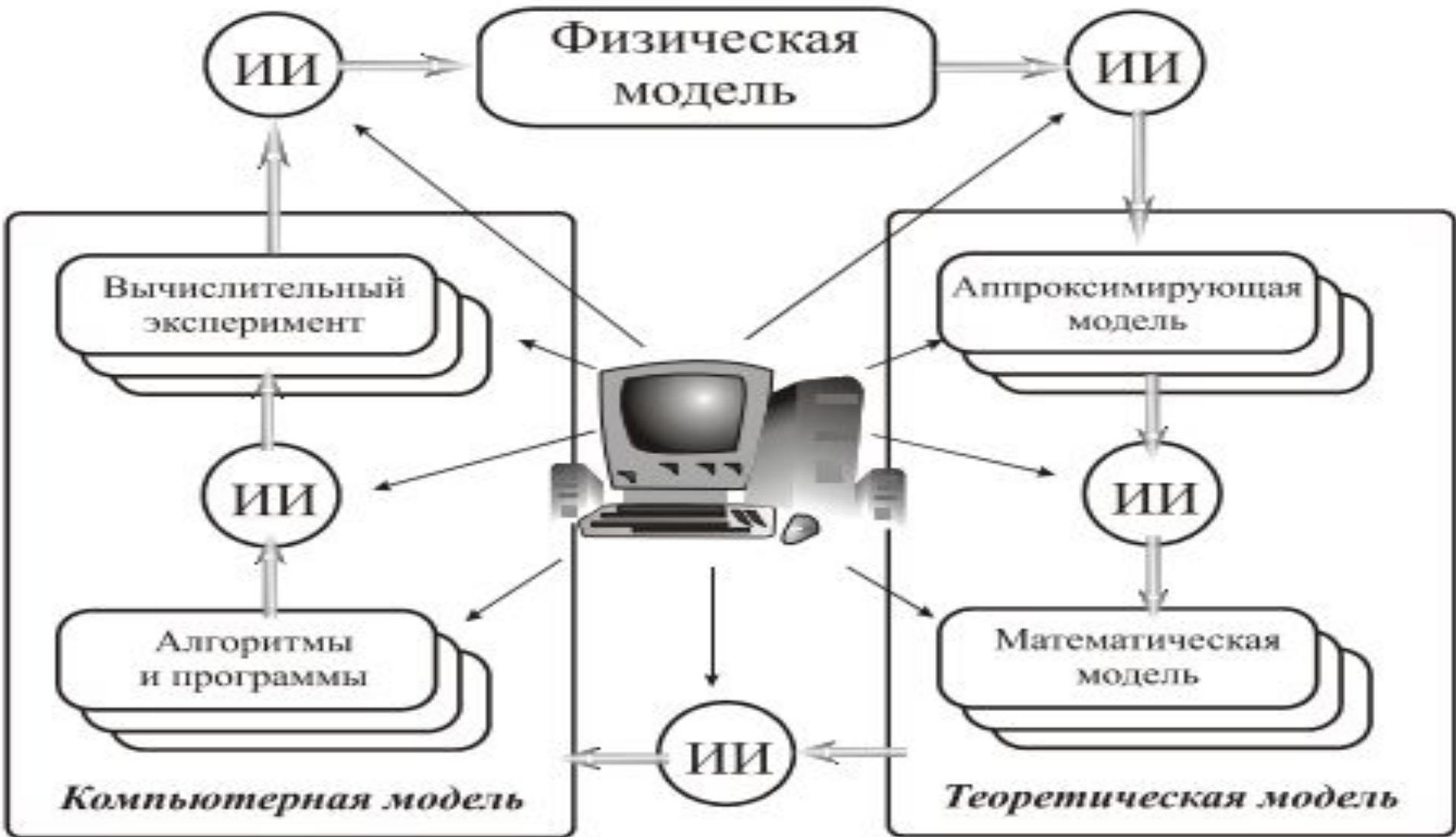
Схема процесса КМ



КМ как метод исследования



Проблемы компьютерного моделирования



Имитационное моделирование



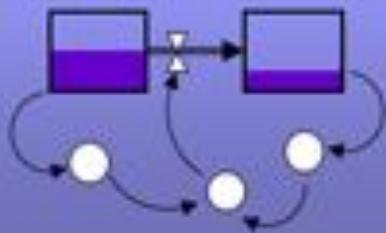
Имитационное моделирование – один из видов компьютерного моделирования, использующий методологию системного анализа, центральной процедурой которого является построение обобщенной модели, отражающей все факторы реальной системы, в качестве же методологии исследования выступает вычислительный эксперимент.

Имитационное моделирование сложных систем применяется в управлении слабоструктурированными системами.

Основные подходы ИМ

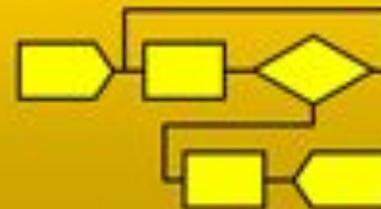
Системная динамика

Связанные переменные,
Накопители, Обратные связи



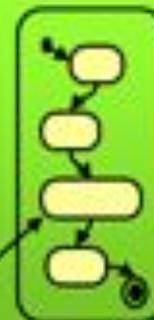
Дискретно-событийное

Заявки, Ресурсы, Процессы
(последовательности операций)



СИСТЕМА

Индивидуальные свойства
и правила поведения.
Прямое или косвенное
взаимодействие



Агентное моделирование

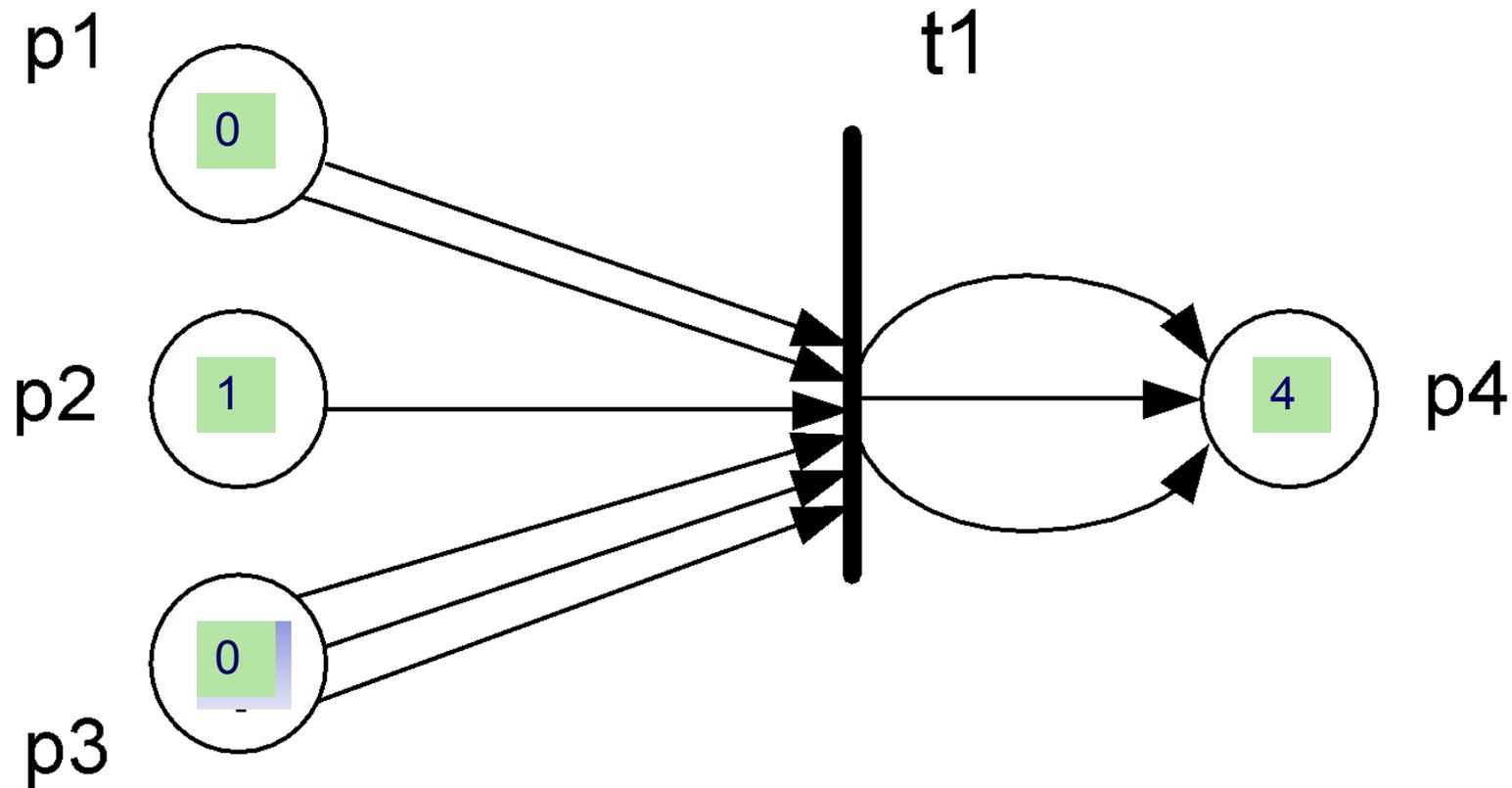
Сети Петри

- ❖ *Сеть Петри определяется как четверка $\langle P, T, I, O \rangle$, где P и T - конечные множества позиций и переходов, I и O - множества входных и выходных функций.*
- ❖ В сетях Петри вводятся объекты двух типов: **динамические**, которые изображаются метками (маркерами) внутри позиций, и **статические**, которые соответствуют вершинам сети Петри.
- ❖ **Маркировка** - распределение маркеров по позициям. Маркеры могут перемещаться в сети. Каждое изменение маркировки называют **событием**, причем каждое событие связано с определенным переходом. *События происходят мгновенно и одновременно при выполнении некоторых условий.*
- ❖ Каждому условию в сети Петри соответствует определенная **позиция**. Совершению события соответствует срабатывание перехода, при котором маркеры из входных позиций этого перехода перемещаются в выходные позиции. Последовательность событий образует моделируемый процесс.

ОСНОВЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

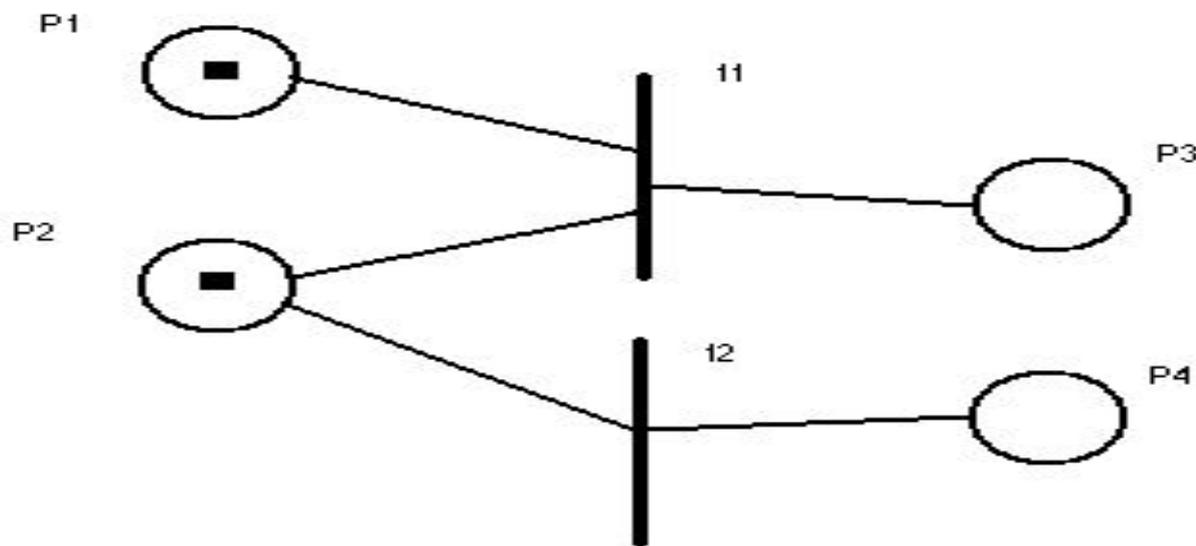
- ❖ Правила срабатывания переходов конкретизируют следующим образом: **переход срабатывает, если для каждой из его входных позиций выполняется условие $N_i \geq K_i$, где N_i - число маркеров в i -й входной позиции, K_i - число дуг, идущих от i -й позиции к переходу; при срабатывании перехода число маркеров в i -й входной позиции уменьшается на K_i , а в j -й выходной позиции увеличивается на M_j где M_j - число дуг, связывающих переход с j -й позицией.**
- ❖ На рисунке показан пример распределения маркеров по позициям. Для срабатывания перехода эту маркировку можно записать в виде (2, 1, 3, 1) или (2 1 3 1). После срабатывания перехода маркировка принимает вид (0,0,0,4).

Пример работы перехода



Конфликтная ситуация

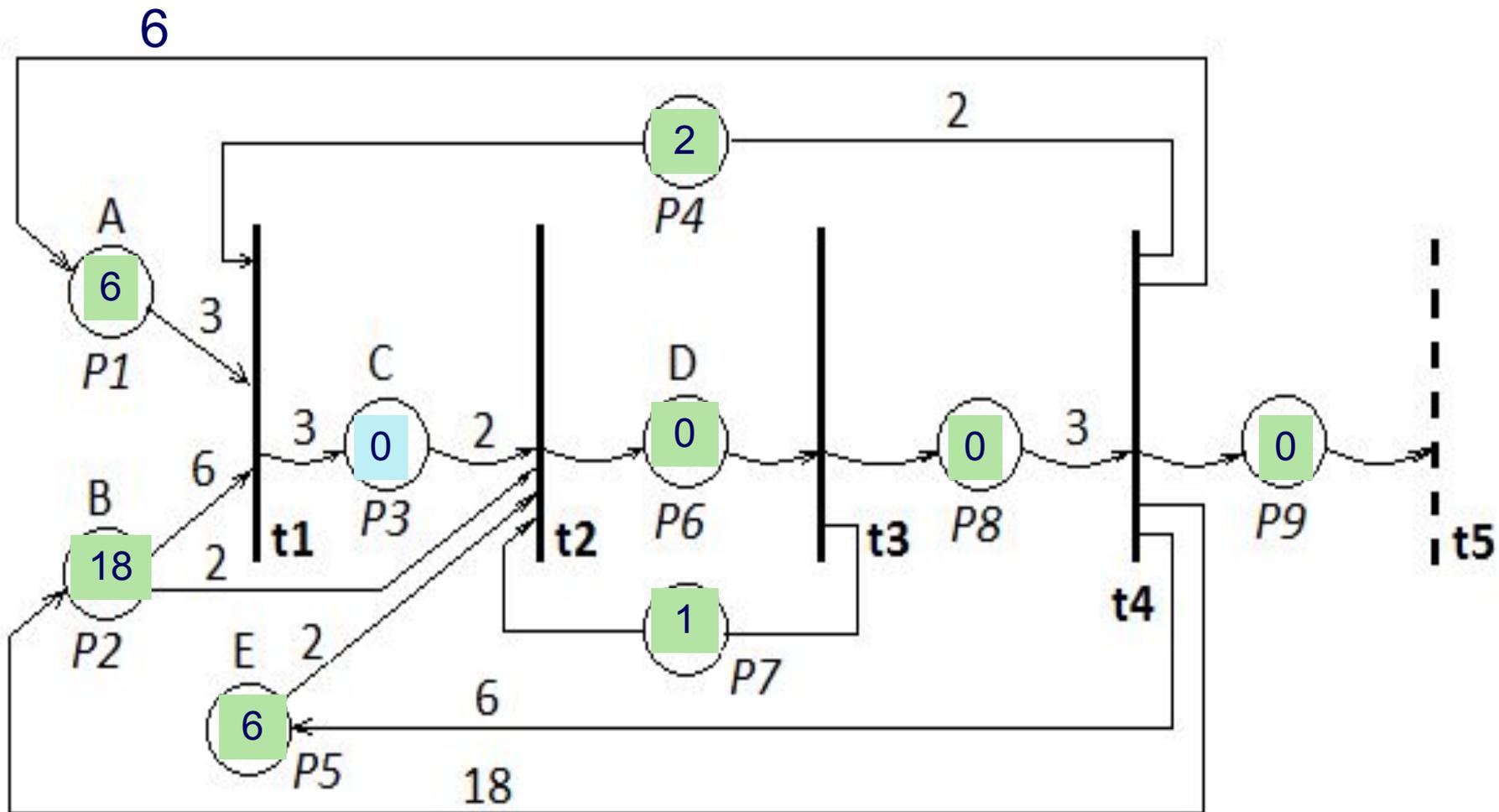
- На рисунке представлен фрагмент сети Петри, иллюстрирующий конфликтную ситуацию: маркер в позиции p_2 может запустить либо переход t_1 , либо переход t_2 . В стохастической сети предусматривается вероятностный выбор срабатывающего перехода в таких ситуациях.



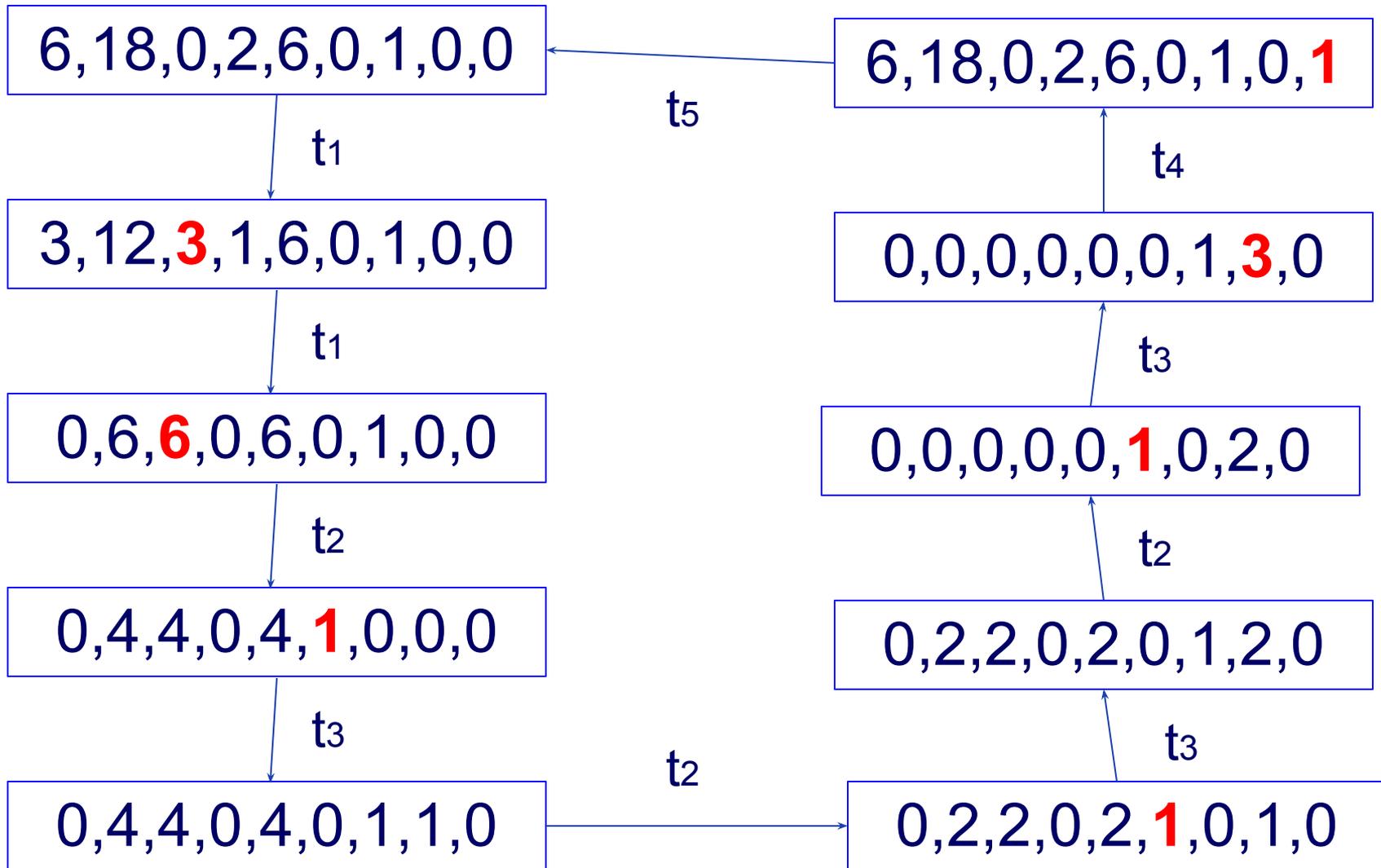
Задание для моделирования

На заводе **С** производится изделие из **1-го** комплектующего завода **А** и **2-х** комплектующих завода **В**. Потом, полученное изделие дорабатывается на заводе **Д** с использованием **1-го** комплектующего завода **Е** и **1-го** комплектующего с завода **В**, причем, завод **С** имеет возможность **одновременной сборки 3-х изделий**, а завод **Д** **2-х**. Сборка на заводах возможна только при условии готовности сборочных цехов. Изделие, собранное на заводе **Д** необходимо проверить на исправность. Задание: построить сети Петри, провести анализ достижимости, создать модель на основе двудольного ориентированного графа.

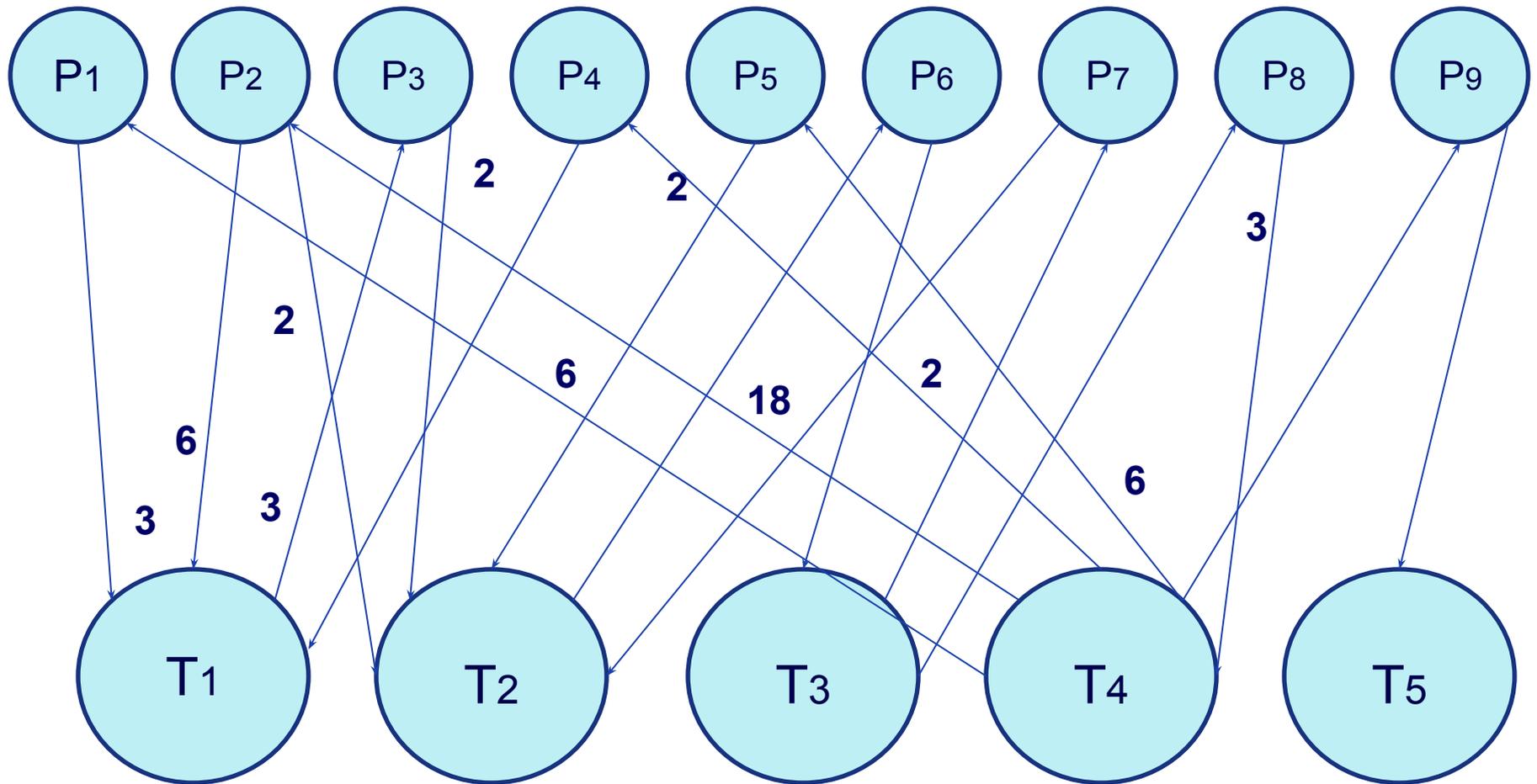
Пример работы сети Петри



Построение графа достижимости



Двудольный ориентированный граф



Модель распределения ресурсов

271

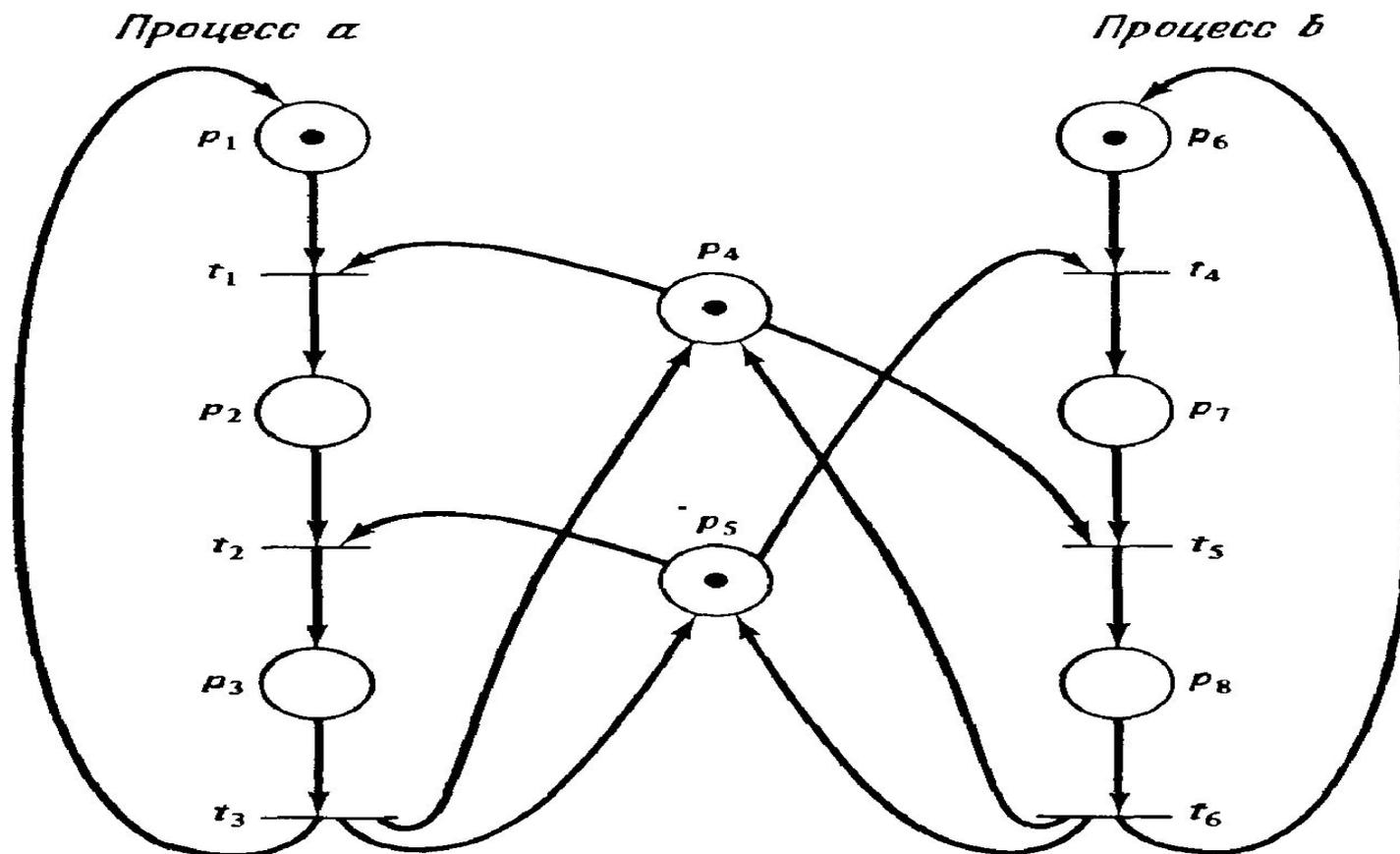


Рис. 4.6. Распределение ресурсов для случая двух процессов (α и β) и двух ресурсов (q (моделируется p_4) и r (моделируется p_5)).

Модель развития кризисной ситуации

272

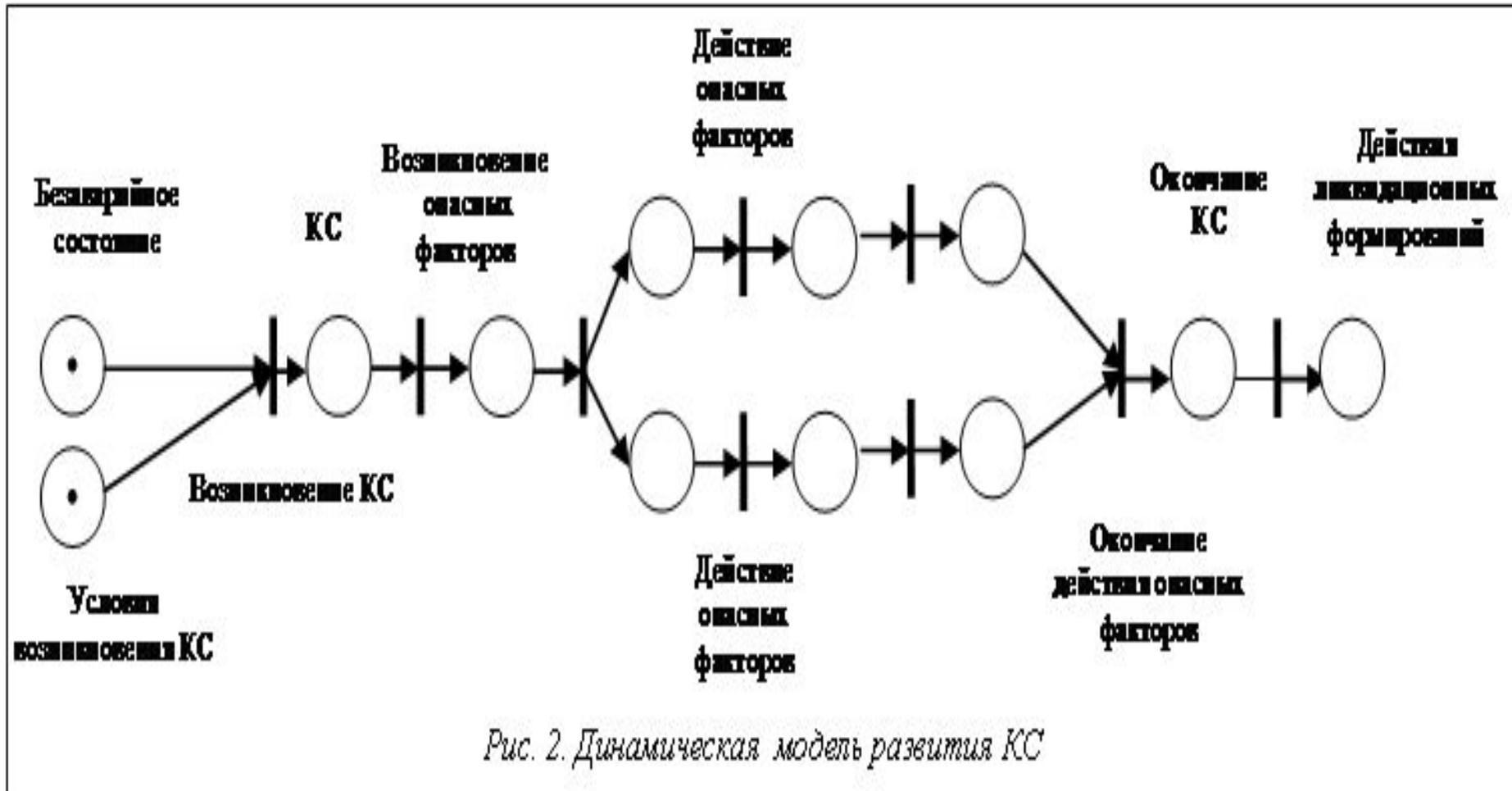
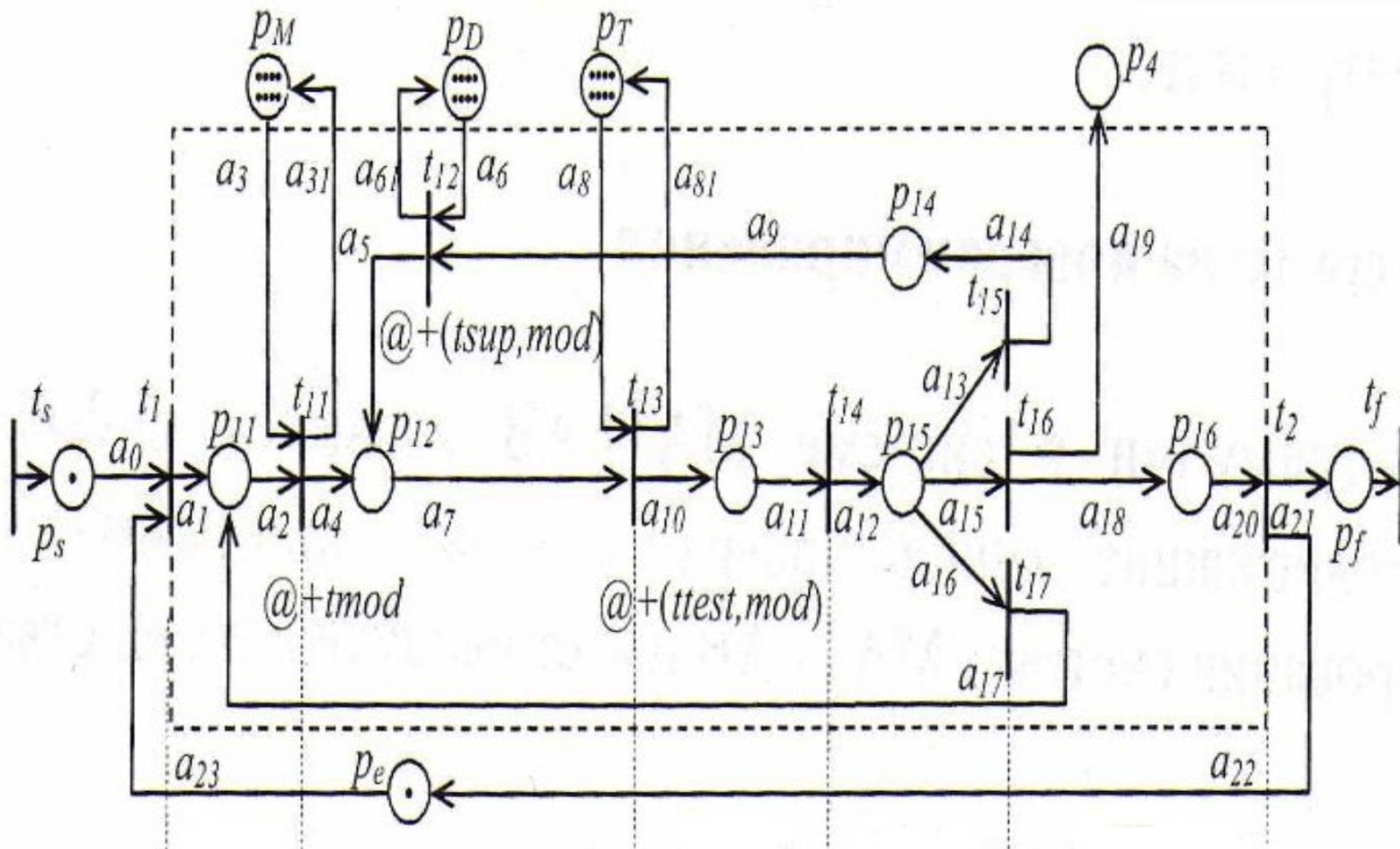


Рис. 2. Динамическая модель развития КС

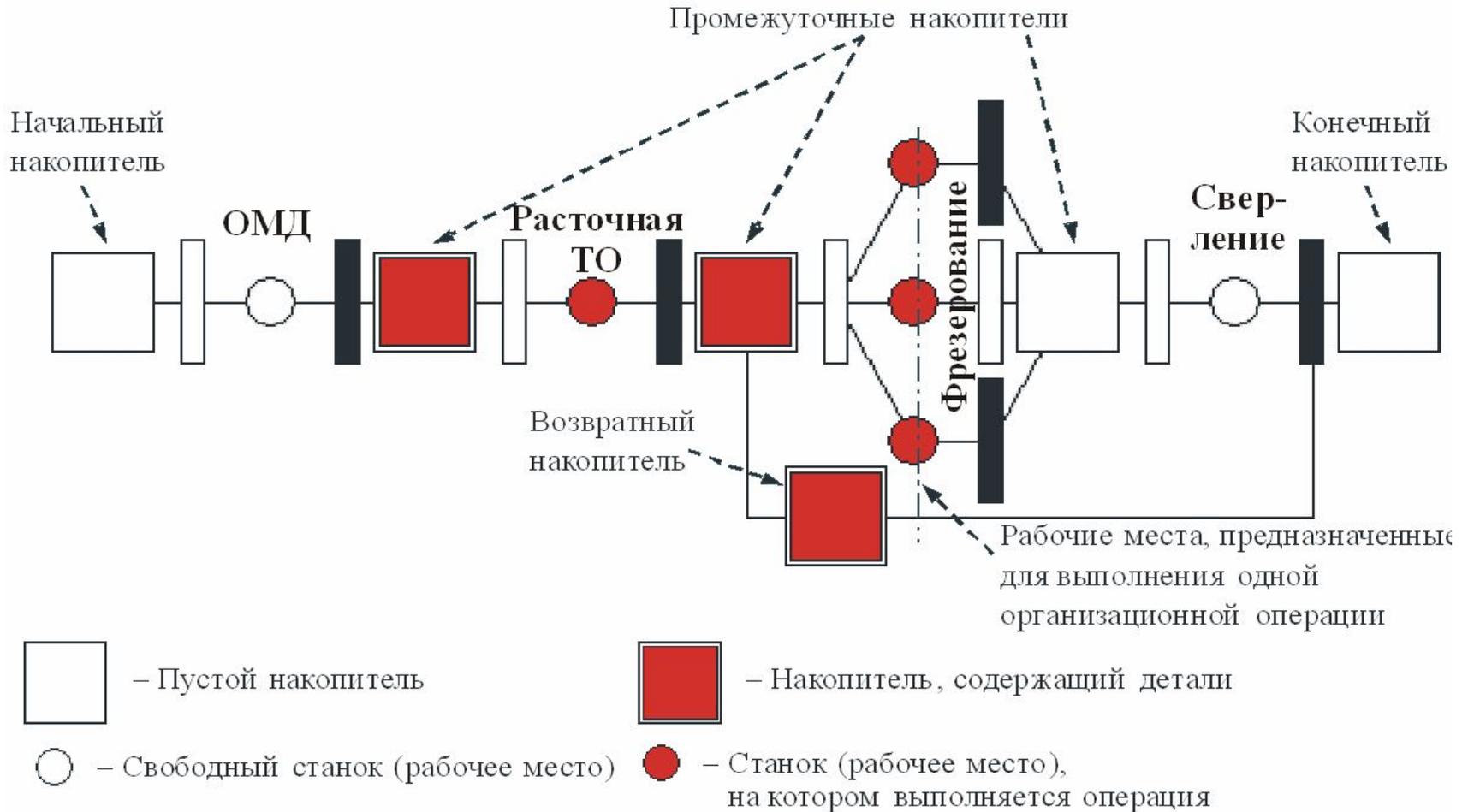
Математическое моделирование на основе сетей Петри

273



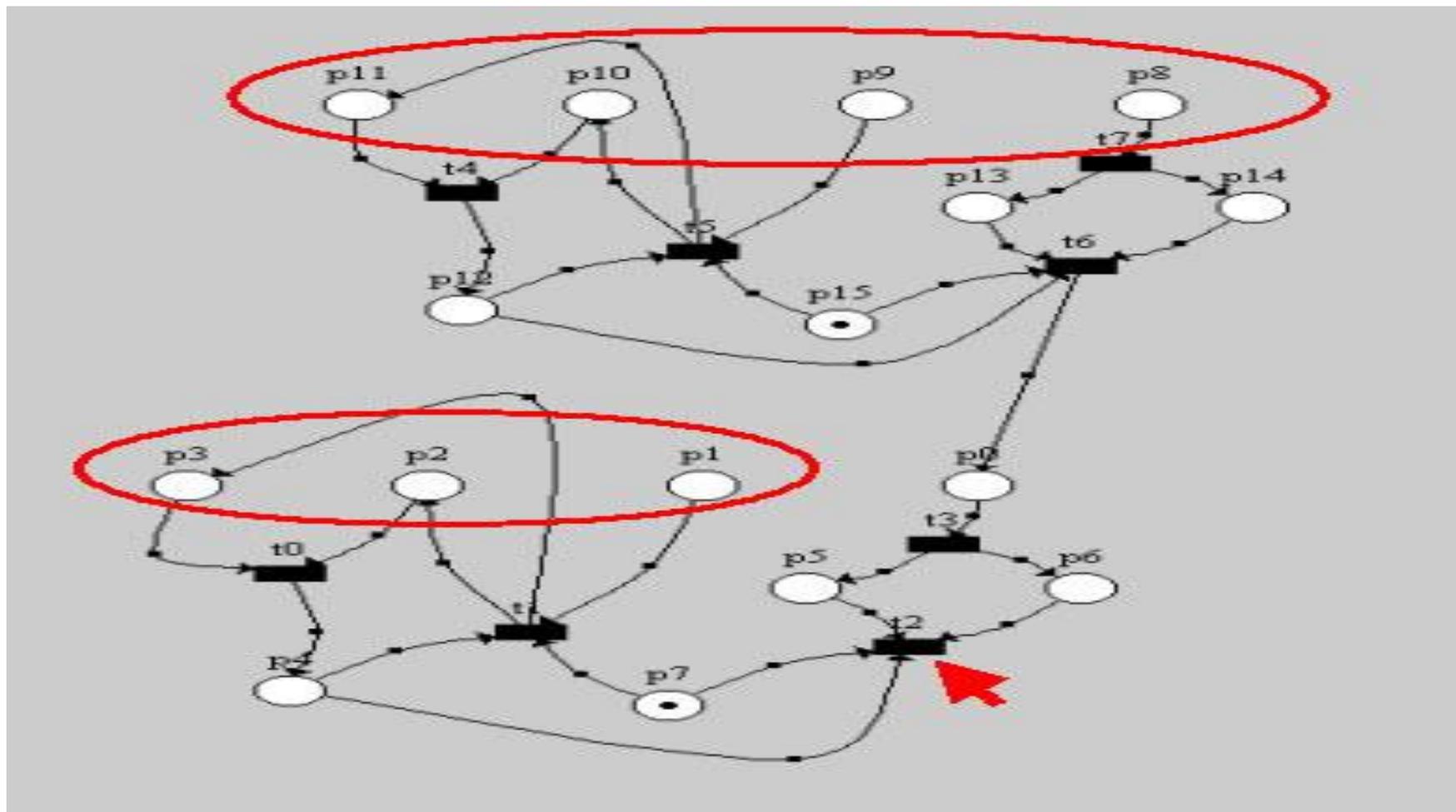
Модель производственной линии

274



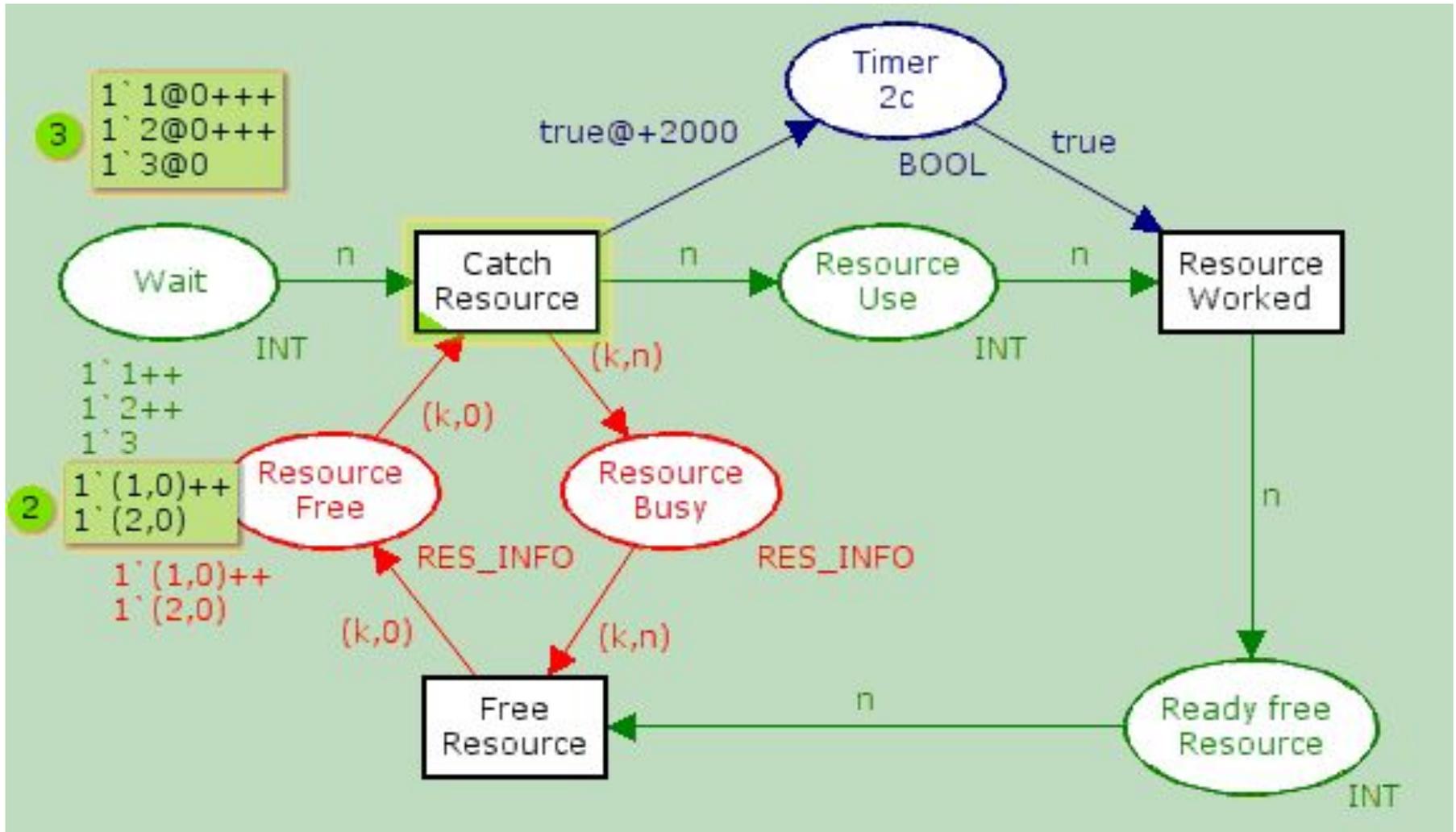
Моделирование открытия переходов при соблюдении условий (защита информации)

275



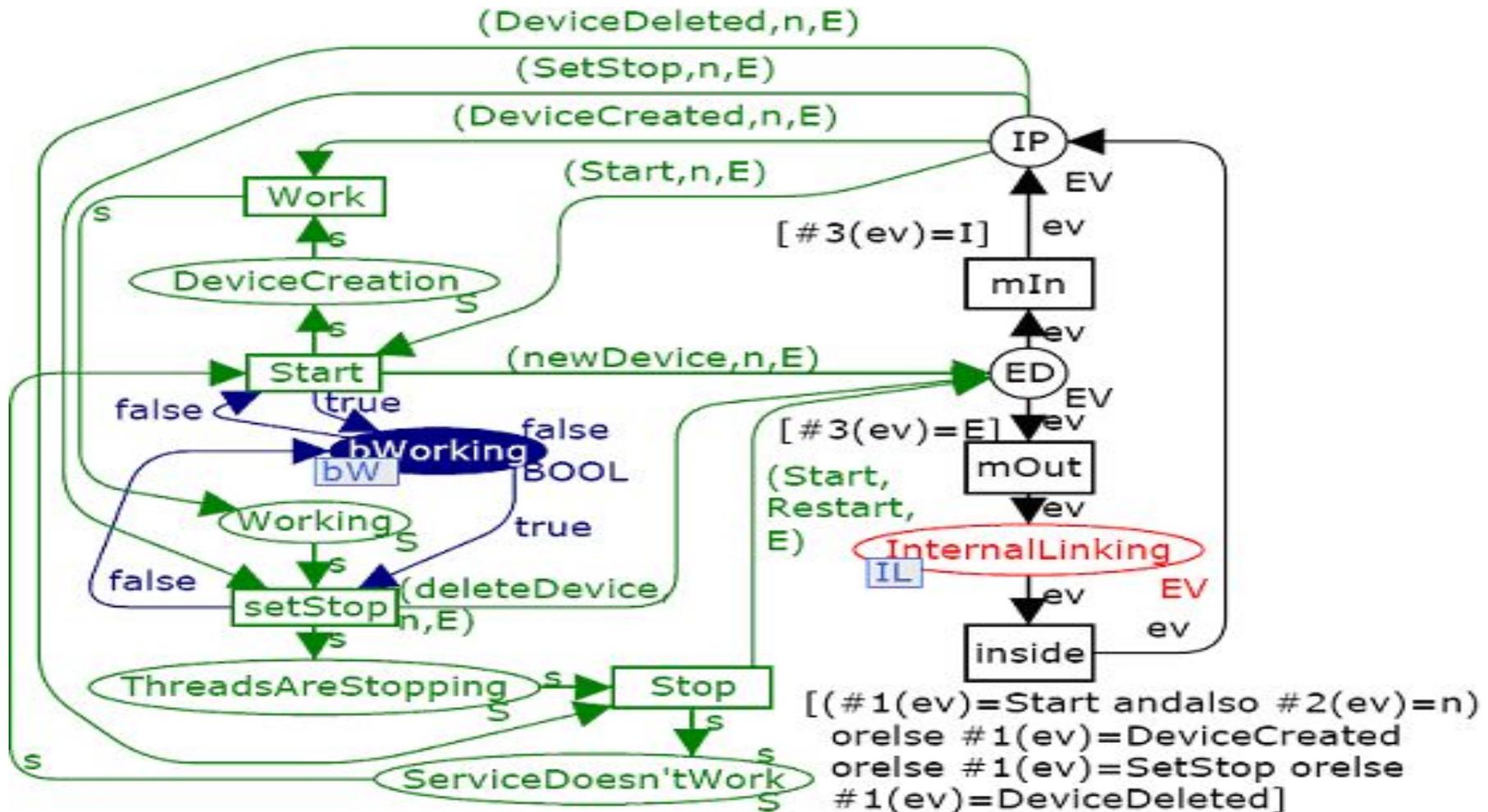
Моделирование программно-аппаратных систем

276



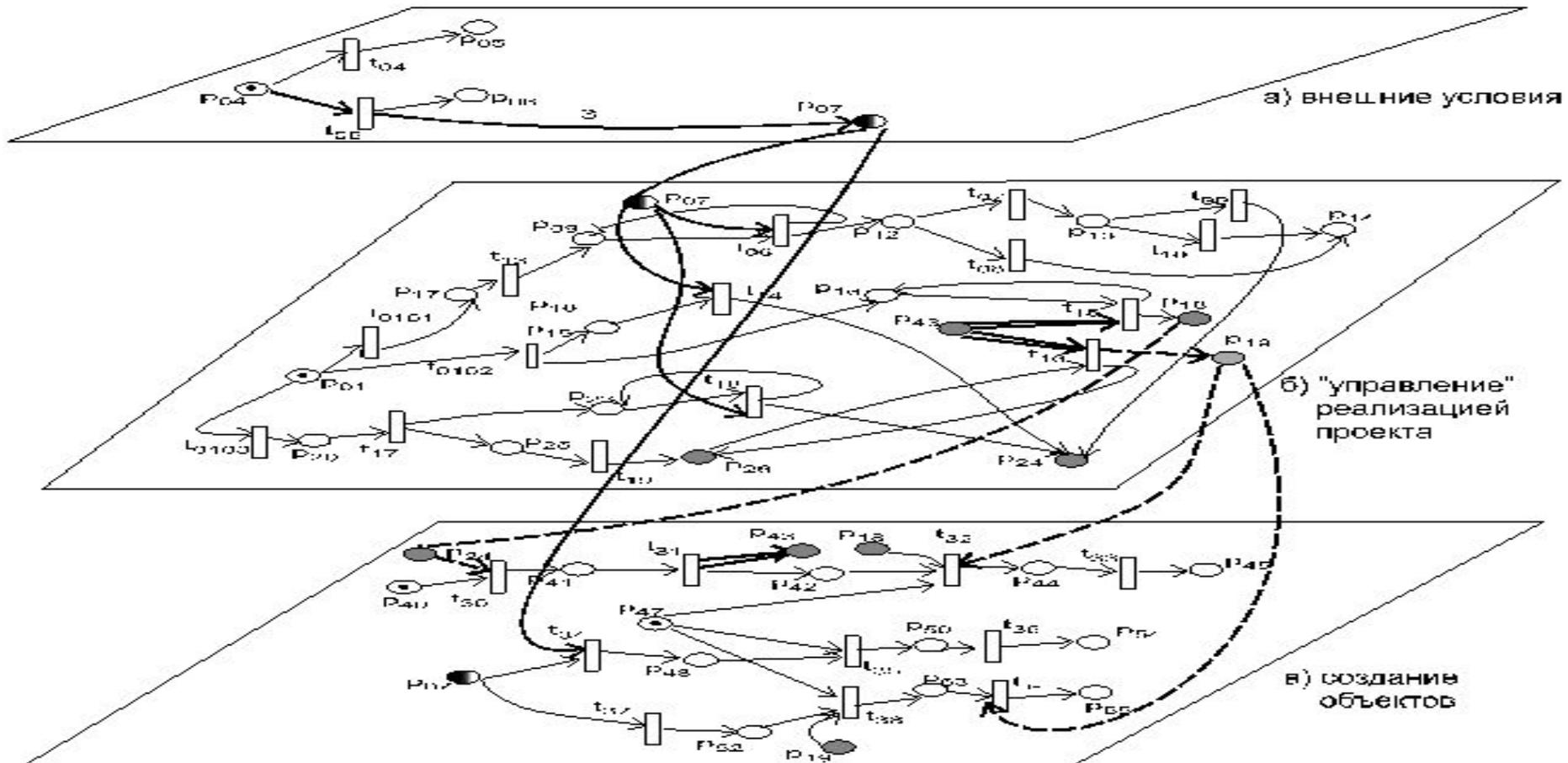
Анализ многопоточного программного обеспечения

277



Экономические модели

278



Условные обозначения:

- условия, □ — события, ⊙ "наличие" условия
- — внешние условия верхнего уровня, ● — "внутренние" условия управления проектом



http://

@

www

internet

5. Развитие технологий проектирования систем

Системное мышление

Системное мышление опирается на **связность, взаимоотношения и контекст.**

Главными свойствами системы являются **свойства целого, которыми не обладает ни одна из частей. Систему нельзя понять с помощью анализа!**

Свойства частей могут быть выведены только из организации целого, поэтому **системное мышление строится на принципах организации.**

Системное мышление – контекстуально,
т.е. противоположно аналитическому мышлению.

Системное мышление



Квантовая теория заставила физиков принять тот факт, что твердые материальные объекты классической физики на субатомном уровне разлагаются на волноподобные вероятностные паттерны. Более того, эти паттерны представляют не вероятности объектов, а вероятности взаимосвязей. Субатомные частицы как изолированные сущности бессмысленны, они могут быть понятны лишь как взаимосвязи. Т.е. природа не демонстрирует нам никаких изолированных строительных блоков.

Целое определяет поведение частей.

Гештальт-психология

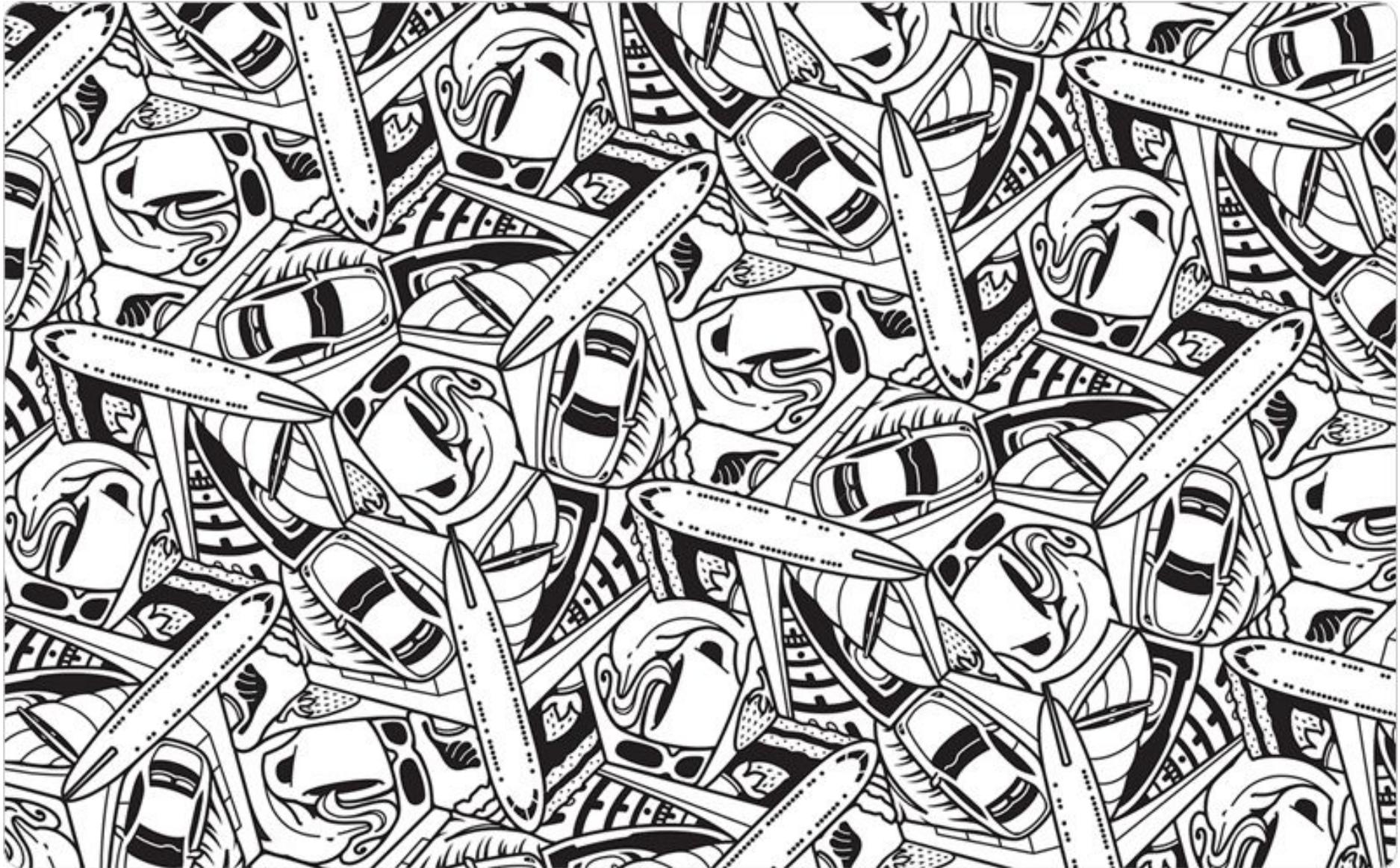


Христиан фон Эренфельс впервые использовал термин гештальт (gestalt) для обозначения **нередуцируемого перцептуального паттерна.**

Эренфельс утверждал, что в гештальте целое превышает сумму своих частей.

Гештальт-психологи видели в существовании нередуцируемых целых ключевой аспект восприятия. Они утверждали, что живые организмы воспринимают вещи не как изолированные элементы, а как интегрированные перцептуальные паттерны – значимые организационные целостности.

Пример паттерна



Экология



Экология обогатила зарождающееся системное мышление, введя два новых понятия – **сообщество и сеть**. Схематично можно изобразить экосистему в виде сети с несколькими узлами. Каждый узел представляет собой организм. Т.е. каждый узел при увеличении сам окажется сетью. При этом в природе не существует «над» и «под», не существует иерархий. Существуют лишь сети, вложенные в другие сети.

Пример, пчелы и муравьи не могут выжить в изоляции, но в больших количествах ведут себя как клетки живого организма.

Особенности системного мышления

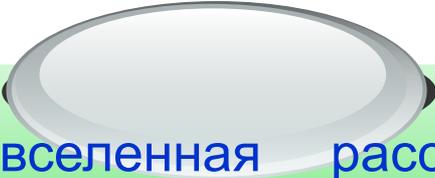
Системы представляют собой интегрированные целостности, чьи свойства не могут быть сведены к свойствам их более мелких частей.

Различные системные уровни различаются уровнями сложности. На каждом уровне наблюдаемые явления отличаются свойствами, которых нет на более низких уровнях.

Системное мышление – контекстуальное мышление. То, что мы называем частью, вообще не существует, – это всего лишь паттерн в неделимой паутине отношений.

В системном мышлении метафора здания по отношению к знанию сменяется метафорой сети. Т.е. знания формируют сеть понятий и моделей, в которой отсутствуют основы.

Понятие динамической «паутины»



Материальная вселенная рассматривается как **динамическая паутина взаимосвязанных событий**. Ни одно свойство любой части этой паутины не является фундаментальным; все они вытекают из свойств других частей, и общая согласованность их взаимосвязей определяет структуру всей паутины. Применительно к науке в целом это означает, что **физика не может более рассматриваться как самый фундаментальный уровень науки**. Различные явления могут принадлежать к различным системным уровням, но ни один из них не может быть фундаментальнее остальных.

Переход к эпистемиологической науке



Вернер Гейзенберг: «То, что мы наблюдаем, не есть природа как таковая, но природа в свете наших вопросов».

Таким образом, **системное мышление включает переход от объективной к эпистемологической науке; к структуре, в которой эпистемология – описание процесса познания – становится составной частью научных теорий.**

Все критерии системного мышления взаимозависимы. Природа рассматривается как взаимосвязанная паутина отношений, в которой идентификация паттернов как «объектов» зависит от наблюдателя и процесса познания.

Приблизительность науки

- Поскольку все природные явления в конечном счете взаимосвязаны, то чтобы объяснить любое из них, придется понять и все остальные.
- Все научные понятия и теории ограничены и приблизительны. Наука никогда не сможет обеспечить полного и окончательного понимания.
- Ученые никогда не имеют дела с истиной в смысле точного соответствия между описанием и описываемым объектом. В науке мы всегда ограничиваемся приблизительными описаниями реальности.

Процессуальное мышление

Все системные понятия можно рассматривать как контекстуальное мышление. Помимо этого, системное мышление – это всегда процессуальное мышление.

Процессуальный аспект в системном мышлении был впервые выделен австрийским биологом Людвигом фон Берталанфи в 30 гг прошлого века. (теория открытых систем)

Афоризм Гераклита: «Всё течет».

Альфред Норт Уайтхед, Клод Бернар, Уолтер Кэннон (концепция гомеостаза – саморегулирующийся механизм, который позволяет организмам поддерживать себя в состоянии динамического баланса, в то время как их переменные колеблются в допустимых пределах.

Метаболизм живой клетки не может быть описан механистической наукой.

Тектология

За 30 лет до Берталанфи русский исследователь **Александр Богданов** разработал системную теорию **тектологии** (от греческого *tekton* – строитель) – науку о структурах. Тектология стала первой наукой, в которой сформулированы принципы организации, действующие в живых и неживых системах. Богданов определял организационную форму как «совокупность связей среди системных элементов». Он использовал термины «комплекс» и «система» как синонимы, и, по сути, первым описал механизм обратной связи на примере парового двигателя и назвал его «**биорегулятором**».

Три типа систем по Богданову

1

- организованные комплексы, где целое превышает сумму своих частей;

2

- неорганизованные комплексы, где целое меньше суммы своих частей;

3

- нейтральные комплексы, где организующая и дезорганизующая деятельности нейтрализуют друг друга;

Общая теория систем

Ньютоновская механика была наукой сил и траекторий, а эволюционное мышление – мышление, основанное на переменных, росте и развитии, требующее новой науки о сложных системах. Первой формулировкой этой новой науки стала классическая термодинамика с ее знаменитым **вторым законом – законом рассеивания энергии (Сади Карно). Любая изолированная или закрытая система будет спонтанно развиваться в направлении постоянно нарастающего беспорядка.** Для математической формулировки данного закона было введено понятие **энтропии** как меры беспорядка. («стрела времени»)

Общая теория систем

В отличие от закрытых систем, находящихся в состоянии теплового баланса, открытые системы далеки от равновесия и поддерживают себя в «устойчивом состоянии», которое характеризуется непрерывным потоком и изменениями. Данное состояние динамического равновесия **Берталанфи** назвал – «**текущее равновесие**». Т.е. в открытых системах энтропия (беспорядок) может снижаться. И второй закон термодинамики здесь не применим. Автор термодинамики для открытых систем – **Илья Пригожин (Теория самоорганизации диссипативных структур)**



http://

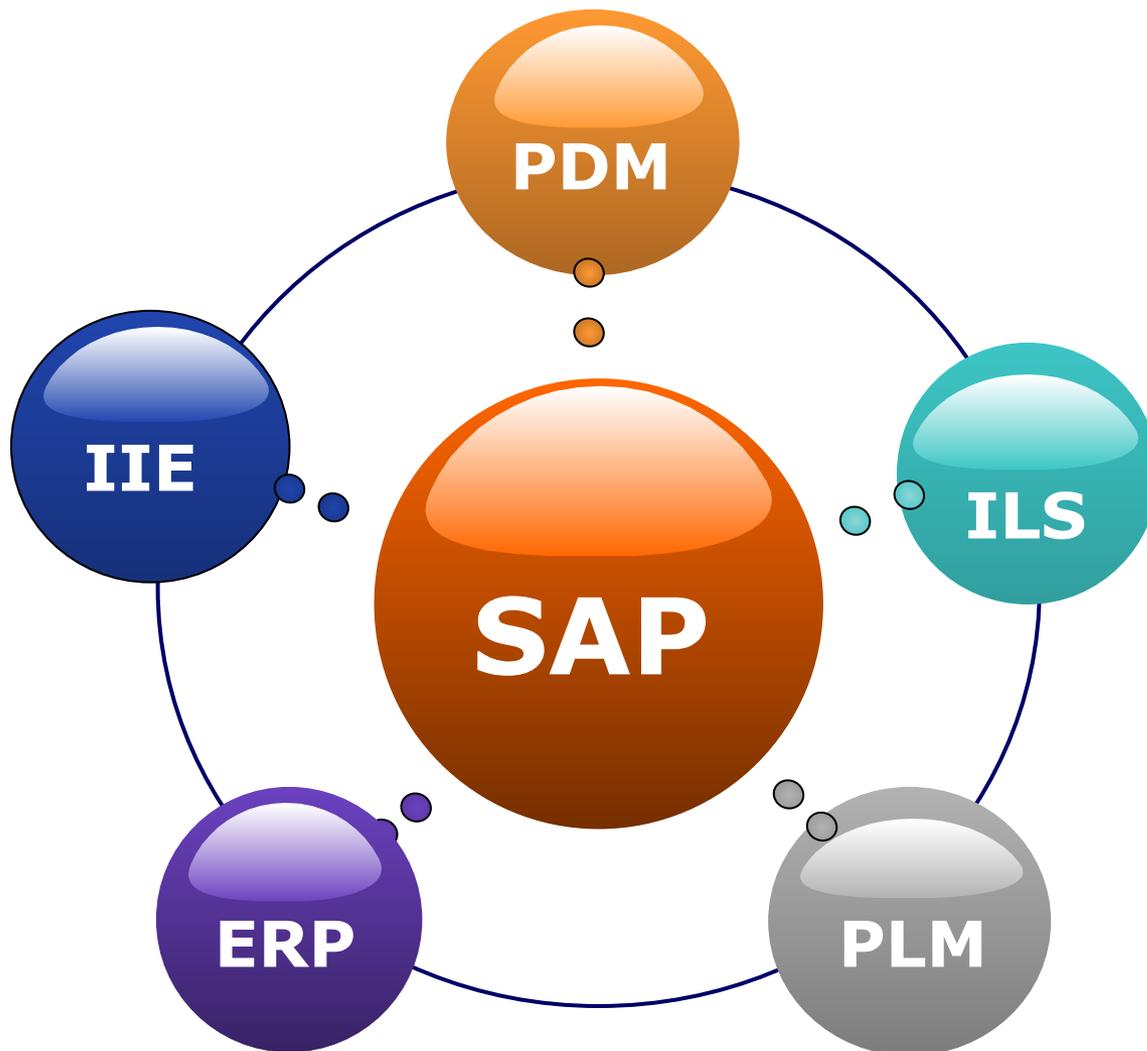


www

internet

6. Направления развития систем поддержки жизненного цикла наукоёмкой продукции

CALS- технологии (ИПИ)



SAP PLM – решение

296

Основная терминология и общие принципы SAP ERP

- ❖ **Основы mySAP ERP.** Технологии SAP ERP предназначены для контроля и описания взаимодействия основных бизнес-процессов в областях управления: заготовкой и планированием материалов; данными жизненного цикла; выполнением производства; складами и запасами; заказами клиентов; основными средствами и сервисным обслуживанием; программами и проектами; человеческим капиталом; внутренней и внешней финансовой отчетностью, а также – реализации бизнес- информации и аналитики и стратегического планирования на предприятии.

Компоненты SAP NetWeaver

297

SAP EP (enterprise portal)

**SAP BW (business information
warehouse)**

SAP XI (exchange infrastructure)

**SAP Web AS
(application server)**

SAP PLM – решение

Для поддержки ²⁹⁸ архитектуры корпоративных сервисов (**enterprise services architecture – ESA**) в SAP применяется прикладная и интеграционная платформа **SAP NetWeaver**, построенная на основе технологии веб-сервисов. Необходимые функции для отраслевых решений предоставляют **четыре компонента SAP NetWeaver**:

- ◆ 1) **SAP EP (enterprise portal)** – интеграция трудовых ресурсов (организация многоканального доступа);
- ◆ 2) **SAP BW (business information warehouse)** – интеграция информации (бизнес-информация и аналитика, управление знаниями);
- ◆ 3) **SAP XI (exchange infrastructure)** – интеграция процессов (управление бизнес-процессами, инфраструктура обмена);
- ◆ 4) **SAP Web AS (application server)** – прикладная платформа (на основе языков Java и Abap).

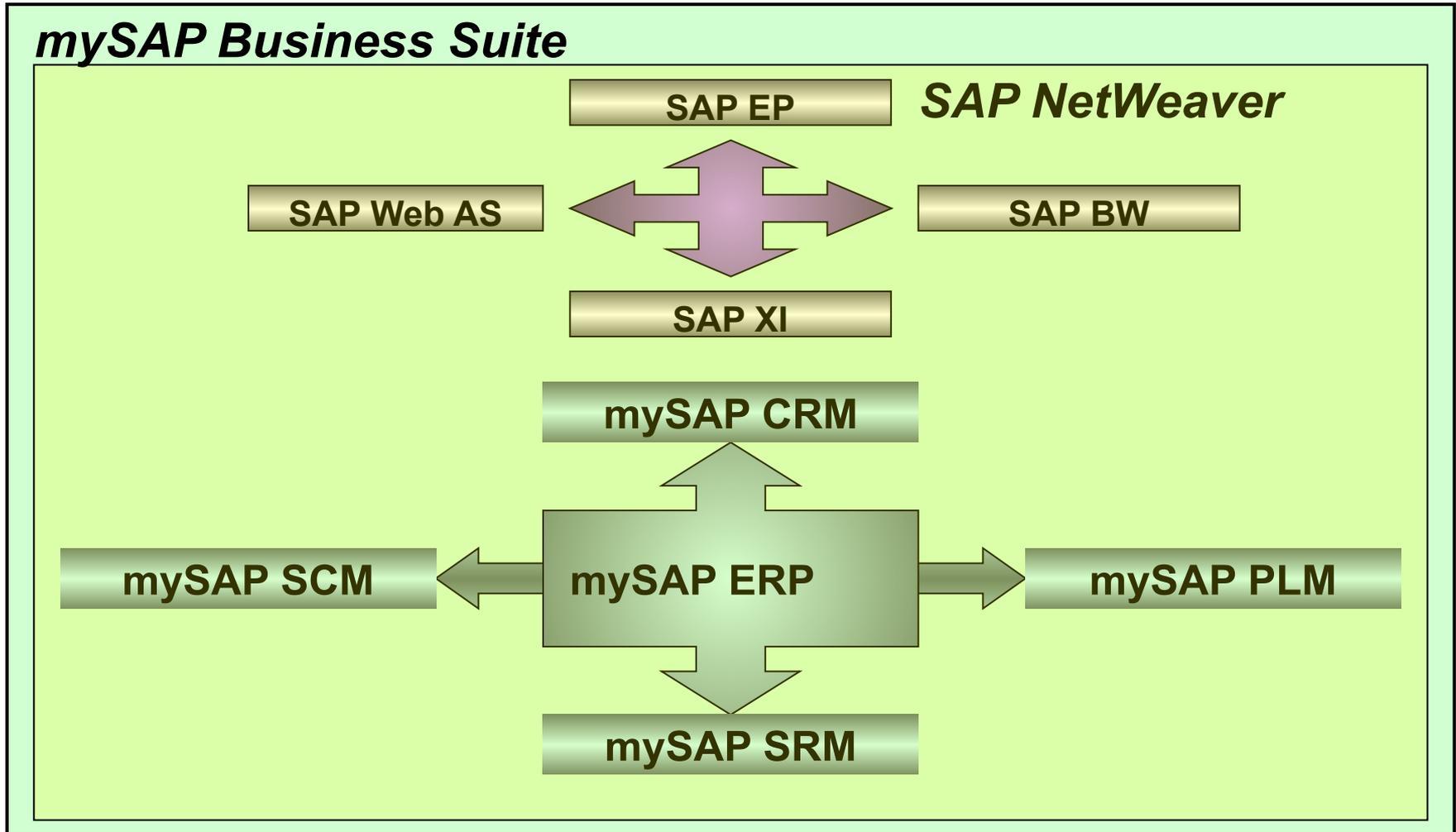
SAP PLM – решение

299

- ❖ Однако, не все решения SAP основаны на SAP NetWeaver. Существуют другие продукты, например, **SAP Business One**, который связан с системной средой SAP посредством XML. Данный компонент реализован на языке C++ и может быть установлен в различных ОС MS Windows. SAP Business One содержит в себе функции из различных областей управления бизнес-процессами (финансы; управление клиентами, закупками, складами и т.д.).
- ❖ Современный этап развития систем SAP начался с системы **SAP R/3**, уже имеющей двухуровневую архитектуру (SAP-базис и SAP-приложение), которая вошла компонентом в представленную в 2003 году систему mySAP ERP – пакет решений в составе mySAP Business Suite (рис.):

mySAP Business Suite

300



mySAP ERP

301

mySAP ERP (enterprise resource planning) – планирование ресурсов предприятия

- ◆ **mySAP CRM (customer relationship management) – управление связями с клиентами;**
- ◆ **mySAP PLM (product lifecycle management) – управление жизненным циклом продукта;**
- ◆ **mySAP SRM (supplier relationship management) – управление отношениями с поставщиками;**
- ◆ **mySAP SCM (supply chain management) – управление логистической цепочкой.**

SAP PLM – решение

Организационные уровни

302

- ❖ **Организационные уровни.** В приложениях SAP **организационные единицы** отражают структуру предприятия. Организационными элементами являются самостоятельные объекты (заводы, склады, пункты продаж, места возникновения прибыли). **Мандант (client)** – виртуальное предприятие (группа предприятий), которое представляет собой единицу верхнего уровня иерархии всех организационных элементов. В баланс данного предприятия включается центральный организационный элемент финансовой отчетности – **балансовая единица (company code)**. В рамках виртуального предприятия балансовых единиц может быть несколько, каждая из которых будет отображать конкретное **предприятие (enterprise)**, **компанию (company)** и **дочернюю компанию (subsidiary)**. Для планирования **производства (factory)** центральной организационной единицей логистики является **завод (plant)**.

SAP PLM – решение

Организационные уровни

- ❖ Одна балансовая единица (company code) может иметь в подчинении несколько заводов, присваиваемых ей в настройке. Условиями продаж клиенту управляет элемент – **сбытовая организация (sales organization)**, имеющая нижним уровнем своей иерархии для представления линеек товаров **отделы (departments)**, **сектора (divisions)**, **бизнес сферы (business areas)** объединенные организационным элементом – **сектор (division)**. При описании возможных многочисленных **складов (warehouses)** используется элемент – **склады (storage locations)** (таблица). Все организационные единицы присваиваются одному или нескольким приложениям системы.

SAP PLM – решение

Организационные уровни

304

Организационные уровни	Терминология SAP
Предприятие (enterprise)	Мандант (client)
Компания (company), дочерняя компания (subsidiary)	Балансовая единица (company code)
Производство (factory)	Завод (plant)
Сбытовая организация (sales organization)	Сбытовая организация (sales organization)
Отделы (departments), сектора (divisions), бизнес сферы (business areas)	Сектор (division)
Склады (warehouses)	Склады (storage locations)

SAP PLM – решение

Организационные уровни

305

- ❖ Организационную структуру, для которой осуществляется ведение и перерасчет затрат и выручки называют **контроллинговой единицей (controlling area)** – отдельная единица учета затрат. **Рынок сбыта (sales area)** – совокупность **сбытовой организации (sales organization)**, **каналов сбыта (distributional channels)** и **секторов (divisions)**. Каждая страна для заводов находящихся на ее территории имеет одну **закупочную организацию (purchasing organization)**, которая выполняет закупки для всех заводов в стране и их проводку в **балансовой единице (company code)** этой страны.

SAP PLM – решение

Основные данные

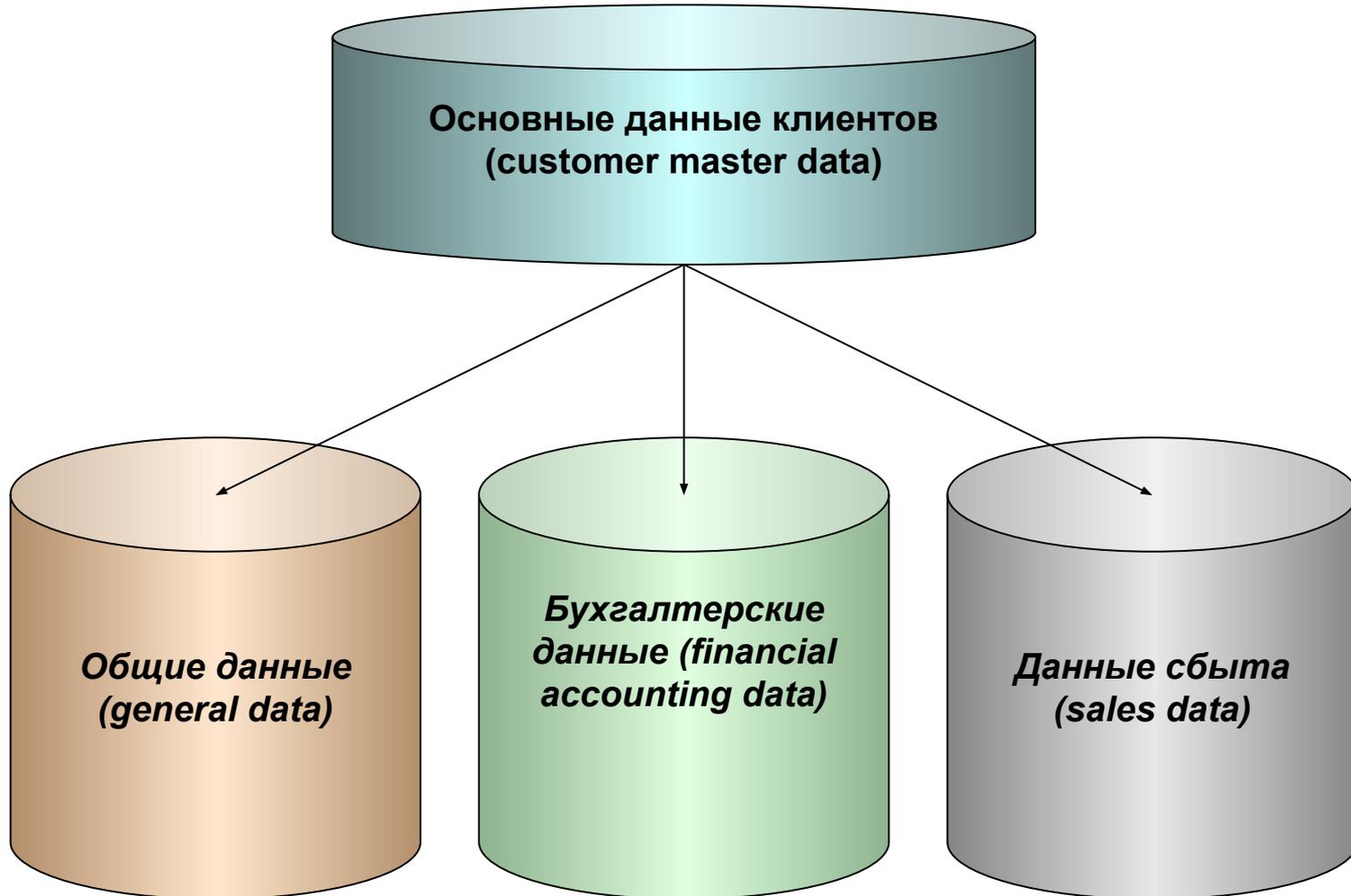
306

- ❖ **Основные данные.** Постоянно используемые в SAP-системе для нескольких бизнес-процессов данные называются **основными (master data)**, например, **клиенты (customers), материалы (materials) и поставщики (vendors)**. Основные данные клиентов (**customer master data**) включают в себя информацию описывающую отношения между компанией и ее клиентом (запросы клиента, поставки, платежные документы и т.д.) и организованы в трёх уровнях (рис):
- ❖ в уровне **манданта (client) – общие данные (general data)**;
- ❖ в уровне **балансовой единицы (company code) – бухгалтерские данные (financial accounting data)**;
- ❖ в уровне **рынка сбыта (sales area) – данные сбыта (sales data)**.

SAP PLM – решение

Основные данные

307



SAP PLM – решение

Основные данные

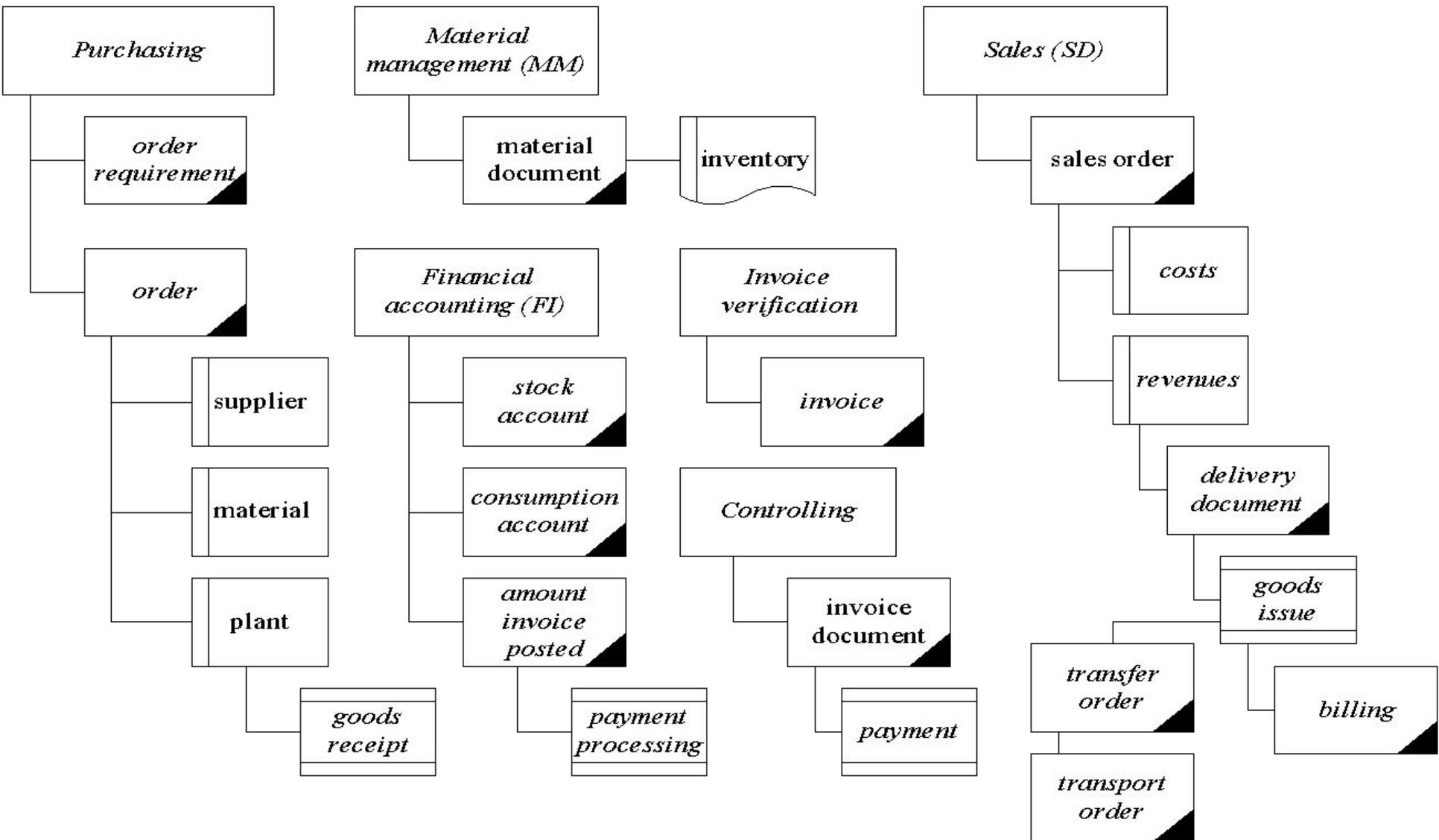
308

Прикладные программы, используемые в SAP для организации бизнес-процессов:

- ❖ создания **заказа клиента (customer order)**;
- ❖ проводки **входящего платежа (incoming payment)**;
- ❖ утверждения **уведомления об отсутствии (leave request)**,
- ❖ называются **транзакциями (transactions)**. Запись данных, создаваемая при выполнении каждой транзакции – **документом (document)**. Документ содержит в себе всю необходимую заданную информацию из **основных данных (master data)** и **организационных элементов (organizational elements)**.

SAP PLM – решение

Основные данные



SAP PLM – решение

Основные данные

310

При формировании потребности в материалах в приложении **Закупки** (*Purchasing*) можно автоматически создать **заявку** (*order requirement*). Создавая **заказ** (*order*), необходимо задать обязательные данные (**поставщик** (*supplier*), **материал** (*material*), **завод** (*plant*) и другую информацию, соответствующую **закупочной организации** (*purchasing organization*)).

SAP PLM – решение

Основные данные

311

С момента **поступления материала (goods receipt)** выполняется ряд обязательных действий:

- ❖ сравнивается с заказом количество поступившего материала;
- ❖ создается **документ материала (material document)** для обновления **данных о запасе (inventory)** в приложении **Управление материалами (Material management (MM))**;
- ❖ в приложении **Финансы (Financial accounting (FI))** создаются документы проведения материала по **счету запасов (stock account)** или **счету расхода (consumption account)** и **проводки суммы счета (amount invoice posted)** по счету поступления материала (дебет/debit) и счету кредитора (кредит/credit);
- ❖ **счет-фактура (invoice)** поступает в приложение **Контроль счетов (Invoice verification)**, где проверяется правильность расчетов и других данных.
- ❖ в **Финансах (FI)** выполняется **обработка платежей (payment processing)**, с помощью которой принимаются решения о способах оплаты и банковских расчетах.

SAP PLM – решение

Основные данные

❖ **Заказ клиента (sales order)** является основой процесса сбыта в приложении **Продажи (Sales (SD))**. При продаже изделий со склада заказ клиента создается для материала в запасе. В данном случае, **затраты (costs)** и **выручка (revenues)** рассчитываются автоматически и создается **документ исходящей поставки (delivery document)**. **Фактурирование (billing)** поставки выполняется после **отпуска материала (goods issue)** со склада и его проводки. Можно также создать **транспортный заказ (transfer order)** на основе **заказа на перемещение запаса (transport order)**. В процессе фактурирования в приложении SD создается **документ счета (invoice document)**, а выручка переносится в приложение **Контролинг (Controlling)**. После клиентского **платежа (payment)** и его получения поставщиком в FI выполняется проводка поступления платежа.

SAP PLM – решение

Основные данные

313

- ◆ Предоставление же услуг является прямым процессом **создания услуги (services generation process)**, который также связан с заказом клиента – **носителем затрат (cost bearer)**. Т.е. затраты и выручку можно провести отдельно в приложении SD. В данном случае отсутствуют этапы транспортировки и поставки. Позиция **заказа клиента (носителя затрат)** проводится для всех транзакций **создания услуг (service generation)**. С помощью **позаказного расчета затрат (order settlement)** после предоставления услуги данные о затратах и выручке переносятся в приложение **Учет результатов (profitability analysis (CO-PA))**.

SAP PLM – решение

Основные данные

- ❖ **Принципы анализа и составления отчетов.** При выполнении приложениями транзакций обрабатываемые данные обновляются в **информационной системе логистики (ИСЛ) (Logistic information system (LIS))**. Данная система отвечает за агрегирование и сохранение информации, анализируемой потом в **информационной системе сбыта (ИСС) (Sales information system (SIS))**. Использование агрегированных данных позволяет сократить время реакции системы и повысить качество генерируемых отчетов.
- ❖ Анализ данных из оперативных и бизнес-приложений, а также внешних источников SAP-систем проводит **SAP BW**. Процессы сбора данных контролируют **инструментальные средства администратора (ИСАдм) (Administrator workbench (AWB))**.

SAP PLM – решение

Основные данные

315

- ❖ **Оперативную аналитическую обработку OLAP (On-Line Analytical Processing)** из больших объемов оперативных и исторических данных также осуществляет **SAP BW**. **Технологии OLAP** были разработаны для анализа данных в системах баз данных с целью поддержки принятия решений и ориентированы, главным образом, на обработку нерегламентированных интерактивных запросов. Основной целью анализа является количественная и качественная оценка достигнутых результатов и/или динамики деятельности компании. Используемые для этого методы сводятся к генерации различного рода выборок, формированию агрегированных данных, трансформациям способов представления данных. Возможности для всестороннего анализа пользователям предоставляет компонент **Business Explorer (BEx)**.

SAP PLM – решение

Компоненты SAP PLM

316

- ❖ При разработке конструкторской документации данные создаются с помощью систем CAD и переносятся в систему управления предприятием при помощи интерфейса PLM. Система управления документами (*document management system (DMS)*) позволяет сохранить первичные документы в защищенных областях SAP и создавать соединения с другими объектами. Браузер структуры продукта позволяет на одном экране просмотреть всю информацию об изделии (*основная запись материала (material master), спецификации (bills of material), технологические карты (routings) и документы (documents)*). За внесение изменений в данные в зависимости от даты, серийного номера или других параметров, определяемых пользователем, отвечает служба изменений. Управление конфигурацией вводит и переносит данные по изделию путем тиражирования.

SAP PLM – решение

Компоненты SAP PLM

317

- ❖ **Система управления документами** интегрирует внешние файлы в mySAP ERP, позволяя выбирать их формат. Требуемым объектом является **инфо-запись (info record)** документа, управляющая обработкой первичных данных. Инфо-запись документа связана с другими объектами, например, материалами и оборудованием, получающими доступ к первичной информации, которая хранится в различных защищенных областях доступных через концепцию полномочий для просмотра и обработки из полей инфо-записи документа.

SAP PLM – решение

Компоненты SAP PLM

318

Документ содержит специфичные данные и состоит из **инфо-записи** и ее **первичного документа**, представленного в электронном или бумажном виде. Инфо-записи имеют свои версии и классификацию и позволяют управлять файлами первичных документов, контролируя ход их обработки. Первичные документы могут храниться в защищенной области. Отображение и редактирование инфо-записи производится с помощью средства просмотра – языка **ECL** или посредством интеграции с **Microsoft Office**. Внешние системы САПР подключаются к SAP при помощи **интерфейса PLM**, а системы архивации – с помощью интерфейса **ArchiveLink**. Преимущества инфо-записи в SAP ERP проиллюстрированы на рисунке.

Функциональные возможности инфо-записи документа в SAP ERP

319

Инфо-запись документа

Функции

Классификация

Управление
версиями

Управление
статусами

Защищенные
области

Интерфейсы к внешним системам

Интерфейс PLM

ArchiveLink

Интерфейс DMS

Интеграция с ECC (ERP Central Component)

Соединение
объектов

Просмотр и обработка первичных документов

Средство просмотра ECL

Интеграция с Microsoft Office

SAP PLM – решение

Компоненты SAP PLM

320

При необходимости организации доступа к информации первичных документов из других объектов SAP можно **создать соединение между инфо-записью и этими объектами**, в числе которых могут быть:

- ◆ материал (material);
- ◆ оборудование (equipment);
- ◆ техническое место (function location);
- ◆ заявка (purchase requisition);
- ◆ документ (document);
- ◆ заказ на поставку (purchase order);
- ◆ номер изменения (change number);

SAP PLM – решение

Компоненты SAP PLM

321

- ❖ операция сетевого графика (network operation);
- ❖ производственный заказ (production order);
- ❖ клиент (customer);
- ❖ позиция торгового документа (sales document item) и т.д.

Также с инфо-записью документа можно соединять спецификации, документацию и информацию с экрана. Соединение объектов может выполняться либо на стороне объекта, либо на стороне документа. Соединения можно использовать для обращения к основной записи материала.

SAP PLM – решение

Приложения PLM и классификация

322

- ❖ Интерфейс PLM даёт возможность подключать к mySAP ERP множество внешних систем для обеспечения обмена данными. В числе таких систем чаще всего присутствуют: **системы CAD; классификационные системы (classification systems); географические информационные системы (GIS) и офисные приложения.**
- ❖ При помощи интерфейса PLM организован двунаправленный обмен данными между mySAP ERP и внешними системами. Перенос данных может осуществляться с использованием **графического интерфейса пользователя (ГИП) (Grafical user interface (GUI)).**

SAP PLM – решение

Приложения PLM и классификация

323

- ❖ В этом случае экраны SAP применяются в качестве диалоговых окон. В противном случае, если GUI не используется, данные вводятся в диалоговых окнах **BAPI (Business API)** — программный интерфейс для доступа к методам бизнес объектов SAP **R/3** системы, который может быть реализован как метод класса, так и как **RFC (Request for Comments)** вызов. **RFC** – запрос комментариев из серии пронумерованных информационных документов **Интернета**, содержащих технические спецификации и стандарты, широко применяемые в глобальной сети. Интерфейс PLM базируется на стандартной библиотеке SAP RFC или других технологиях, например, SAP Java Connector.

SAP PLM – решение

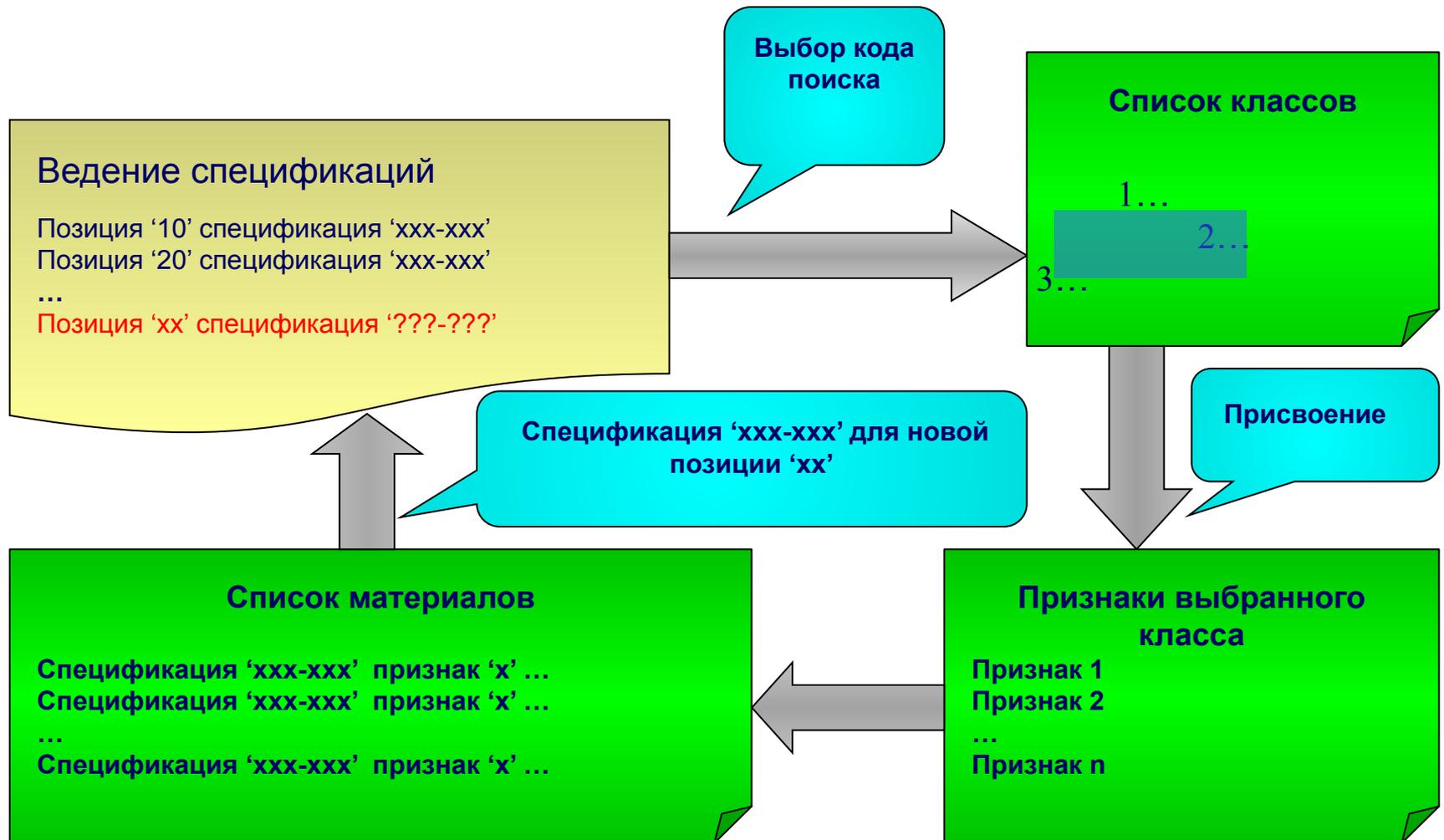
Приложения PLM и классификация

324

- ❖ Для упрощения поиска объектов предусмотрена **классификация**, позволяющая определять не только точный номер объекта, но и его **атрибуты и параметры**. Организована возможность поиска похожих объектов в пределах установленных ограничений на точность критериев сравнения. Новая основная запись материала создается как **компонент спецификации (component for the BOM – bill of material)**. Для определения места позиции данного нового материала выполняется поиск по соответствующим **классам материалов**. При обнаружении необходимого, на экран выводятся **признаки классов (characteristics of the classes)**. Процесс поиска запускается после присвоения признаков. Результатом поиска является **список материалов (list of materials)**, содержащий значения признаков, совпадающих с заданными критериями выбора. После чего, найденный соответствующий материал копируется в спецификацию (рис.).

Цикл определения спецификации для новой позиции

325



SAP PLM – решение

Приложения PLM и классификация

326

- ❖ В SAP PLM создается **система классов**, где определяются **атрибуты (признаки)**, которые наиболее полно и корректно описывают продукт. Далее эти признаки создаются с указанием **специфичных значений признаков** и присваиваются не материалу, а целому **классу**. Классам присваиваются соответствующие **объекты**. Документ может включаться в другой **класс** как **основная запись материала**. **Классификация выполняет следующие основные функции:**
 - 1) создание признаков и их допустимых значений;
 - 2) ведение классов и присвоение им соответствующих признаков;
 - 3) создание объектов и присвоение их определенным классам;
 - 4) поиск объектов.

SAP PLM – решение

Приложения PLM и классификация

327

- ❖ Таким образом, классификация осуществляет **присвоение объектов классу и присвоение признаков в классе**. Классификация производится либо непосредственно для объекта, либо в соответствующих транзакциях системы классификации.
- ❖ При описании комплексных изделий в компонентах: **сбыт (sales) и производство (product)** выполняются задачи по **конфигурированию (configuration)**. Изделие изготавливаемое в различных вариантах называется **конфигурируемым материалом (configurable material)**, который включает все возможные свойства (признаки) продукта для их дальнейшего отбора по заданным критериям в производимый объект.

SAP PLM – решение

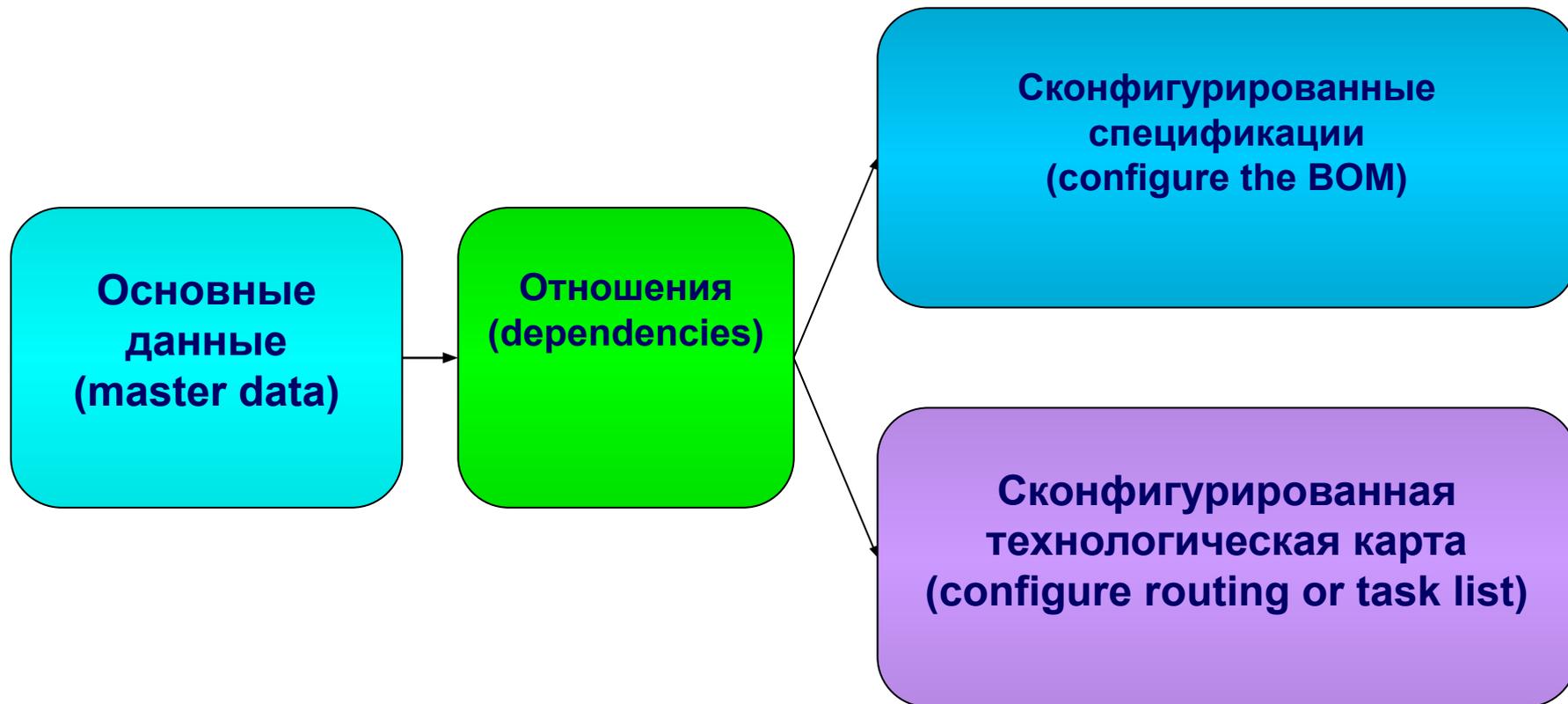
Приложения PLM и классификация

328

- ❖ Все компоненты, которые могут понадобиться для производства конкретного варианта материала содержатся в *спецификации конфигурируемого материала (BOM of the configurable material)*, для создания *конфигурации спецификации (configure the BOM)* (выбора компонентов для конечного определенного варианта изделия) применяется *описание отношений (dependencies)* (рис.).
- ❖ *Технологическая карта (routing)* или *список задач (task list)* конфигурируемого материала – это перечень всех операций, которые могут оказаться необходимыми для производства определенного варианта изделия. Для выбора необходимых операций также используется *описание отношений (dependencies)* (рис.).

Конфигурируемые основные данные

329



SAP PLM – решение

Браузер структуры изделия

- ❖ **Графический браузер³³⁰ структуры изделия (*graphical product structure browser*)** – основное навигационное информационное средство в компоненте *mySAP PLM*, которое представляет собой древовидную структуру и отражает функциональные связи между объектами. В браузере любой объект может быть вызван и изменен. Все изменения отображаются сразу после подтверждения с помощью кнопки **«обновить» (*refresh*)**. Ограничение вывода на экран подробных данных осуществляется при помощи **фильтров (*filters*)** в соответствии с заданными требованиями. За просмотр в браузере **первичных документов (*originals*)** отвечает инструмент ***Enterprise Application Integration (EAI)***.

- ❖ **Инструментальные средства инжиниринга (ИСИ) (*Engineering workbench (EWB)*)** позволяют не только создавать, но и вести спецификации и технологические карты, формируя отношения между их позициями в ***рабочем списке (worklist)***. Рабочий список содержит выбранные для обработки ИСИ объекты, которые копируются в него из базы данных. После завершения обработки выбранных объектов (создания новых, изменения или удаления) рабочий список сохраняется.

- ◆ Для организации возможности обработки некоторых данных одновременно несколькими пользователями в ИСИ реализуется **логика блокирования (lock logic)**. Пользователь может деблокировать позицию или операцию без прерывания обработки других объектов, если другому пользователю необходимо отредактировать эти данные. Информация о способах связи с нужным пользователем выводится на экран **автоматически**.

- ❖ **Служба изменений (Engineering change management (ECM))** – централизованная логистическая функция для изменения основных данных. Проводимые изменения группируются, контролируются и документируются при необходимости. ECM может применяться для сохранения истории версий различных объектов (спецификаций, технологических карт). Кроме того, определяется **область действия (effectivity) изменения (change)**, активность которой позволяет изменению вступить в силу. Данный параметр может определяться моментом времени.

- ◆ **Изменения вступающие в силу автоматически в определенной области действия, становятся активными во всех сегментах логистической цепочки, например, для заказов клиента (sales orders), планирования потребности материалов (ППМ) (material requirement planning (MRP)) и управления производством (shop floor), но только в том случае, если для этих областей установлен ключ деблокирования (key is set).**

Таким образом, основные функции службы изменений можно представить следующим списком:

- ◆ группировка связанных между собой изменений и их присвоение номеру изменения;
- ◆ контроль и документирование изменений;
- ◆ сохранение нескольких статусов изменений для одного объекта;
- ◆ планирование и реализация определенной области действия;
- ◆ интеграция цепочки логистических процессов.

SAP PLM – решение

Служба изменений

336

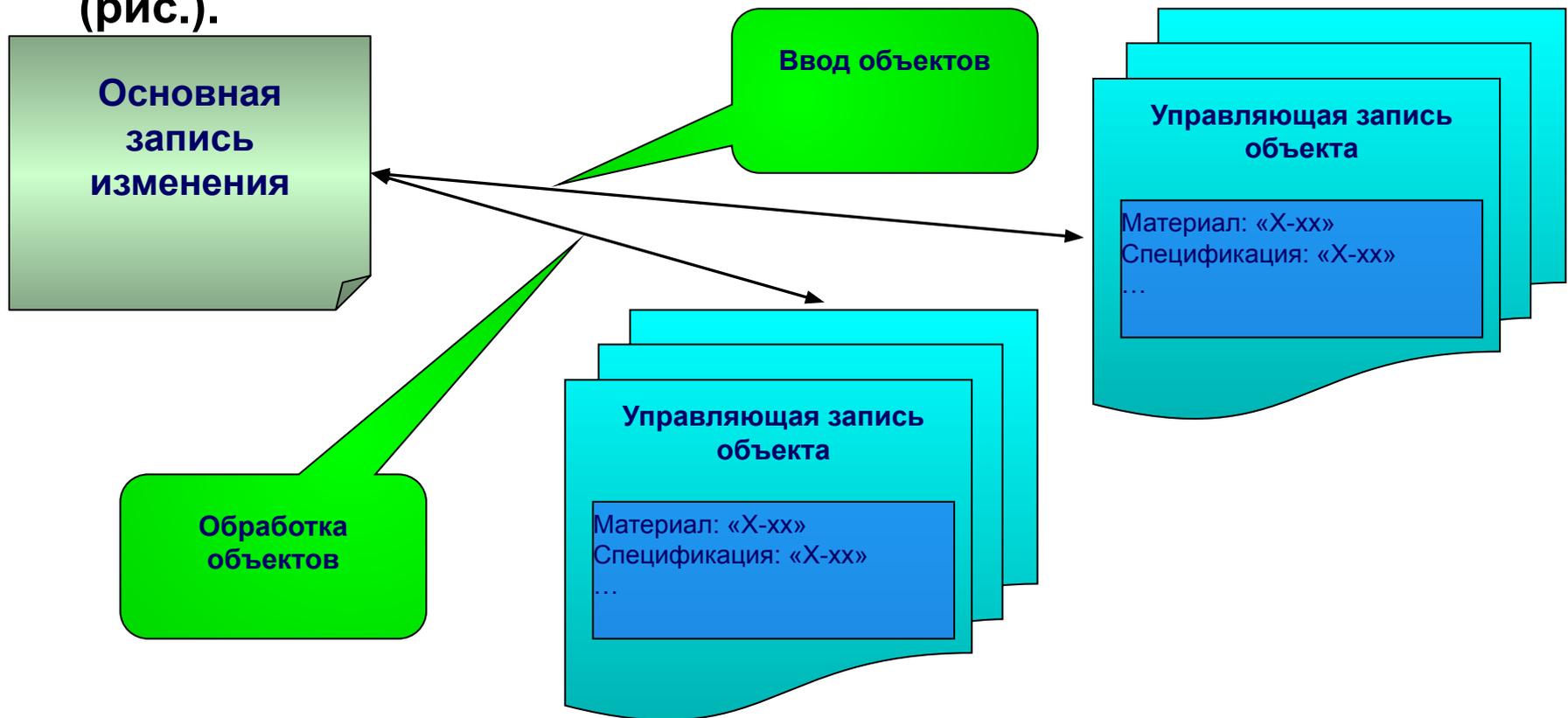
Общие данные об изменении (краткое описание, область действия, статус) вводятся в его **заголовок (*header*)**. Выбор соответствующих видов объектов позволяет определить допустимость применения к ним данного номера изменения. Сами объекты, подлежащие изменению, их спецификации и технологические карты указываются в **управляющей записи объекта (*object management records*)**. Номеру изменения можно также присвоить **связанные документы (*accompanying documents*)** в форме инфо-записей документов. Поиск номеров изменения производится при помощи **классификации**. Точные сроки начала действия изменения для определенных объектов позволяют контролировать **альтернативные даты (*alternative dates*)**.

SAP PLM – решение

Служба изменений

337

Информация о причине изменения или области его действия заносится в **заголовок основной записи изменения**. Там же указываются изменяемые или создаваемые объекты (рис.).



SAP PLM – решение

Служба изменений

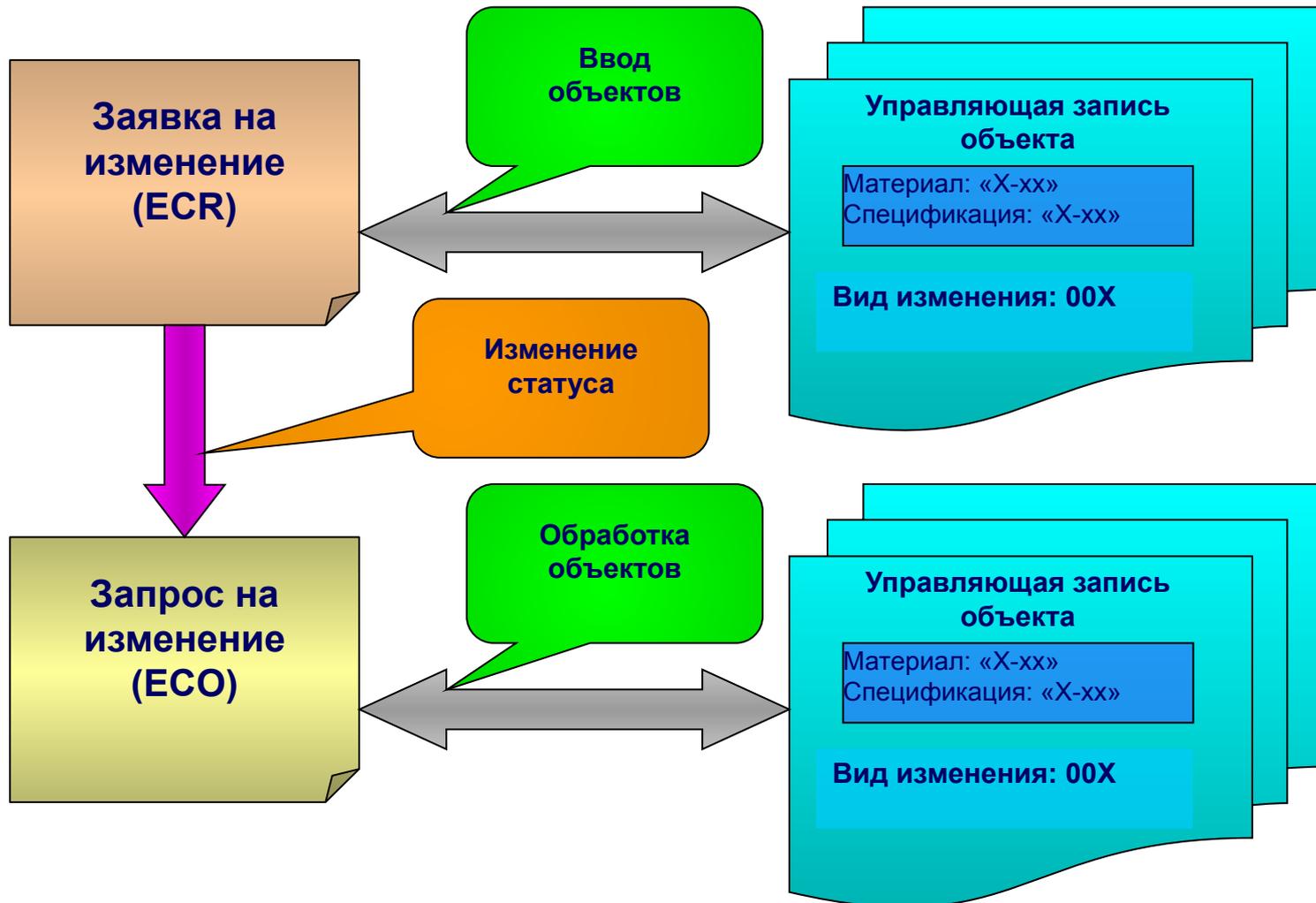
338

Данные об изменяемом объекте и его статусе также вводятся в заголовок **заявки на изменение (engineering change request (ECR))**. Пользователь сможет начать изменять объекты с момента преобразования заявки на изменение (ECR) в **запрос на изменение (engineering change order (ECO))** (рис.).

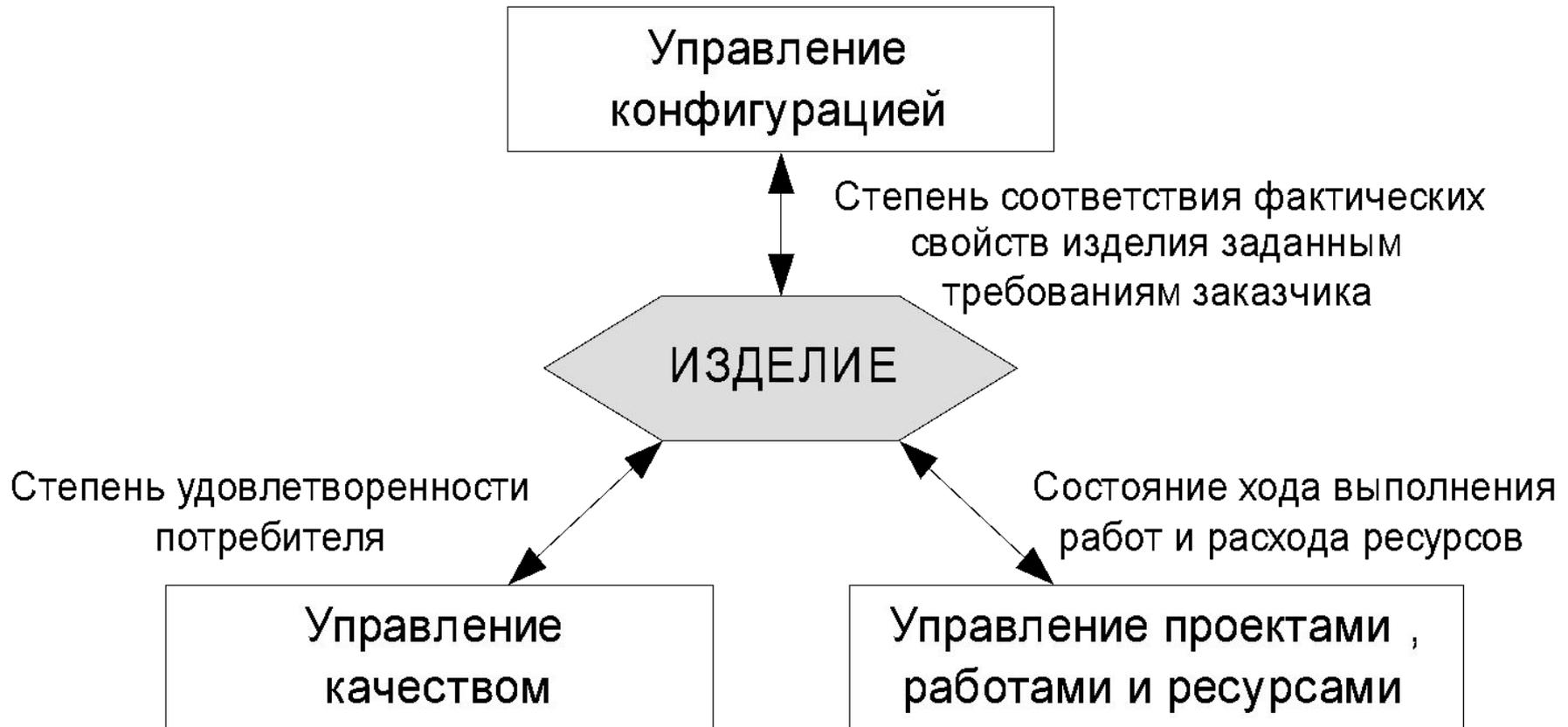
Заявка на изменение (ECR) и запрос на изменения (ECO) отличаются от основной записи изменения сетью статусов, отображающих процессы запроса, проверки и реализации изменений. За координацию информационных потоков по всем видам операций отвечает **система управления потоками операций (Workflow management system)**, что позволяет пользователям получать оперативное представление правильной информации.

SAP PLM – решение Служба изменений

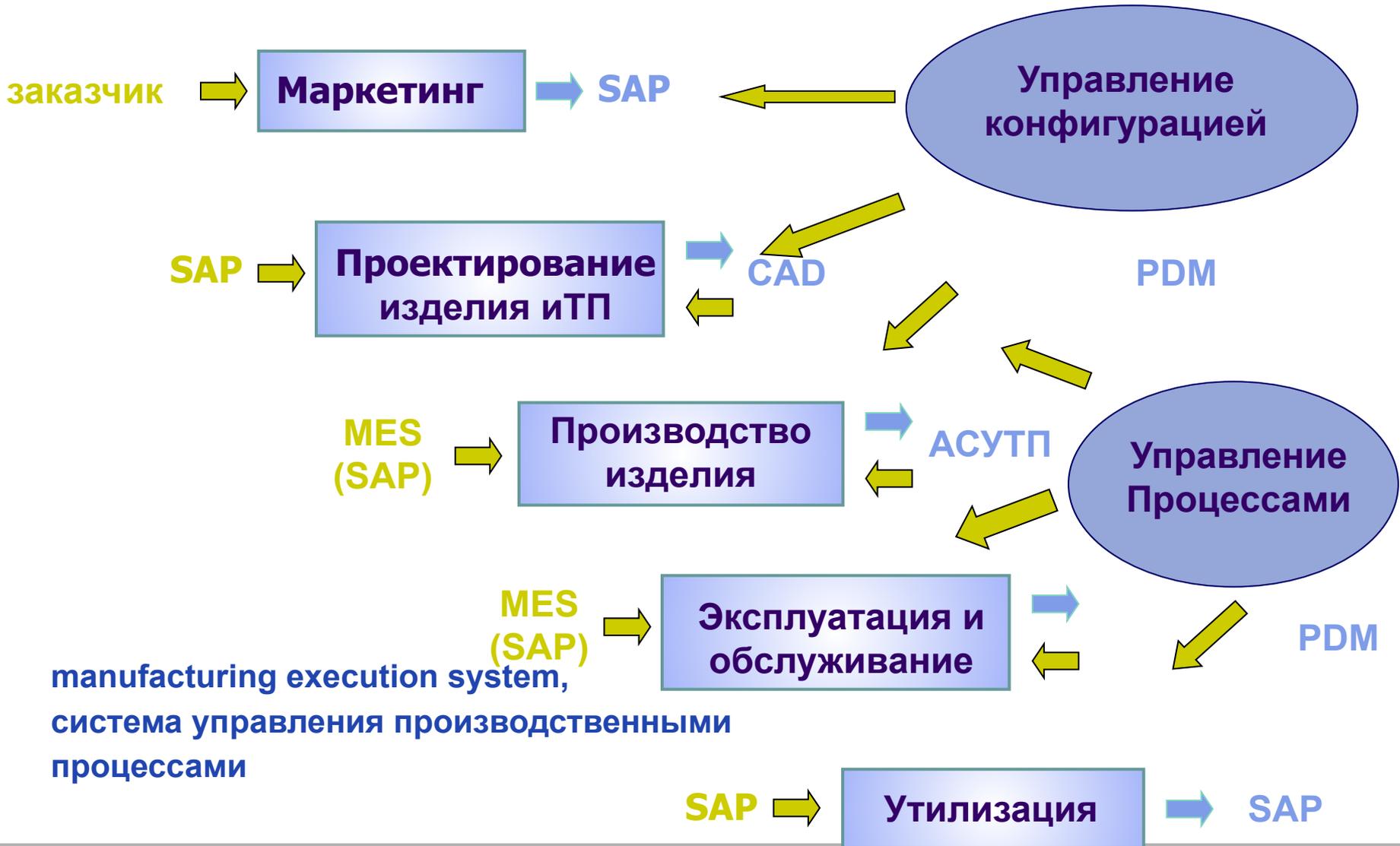
339



CALS технологии



Технологии CALS на этапах жизненного цикла продукции

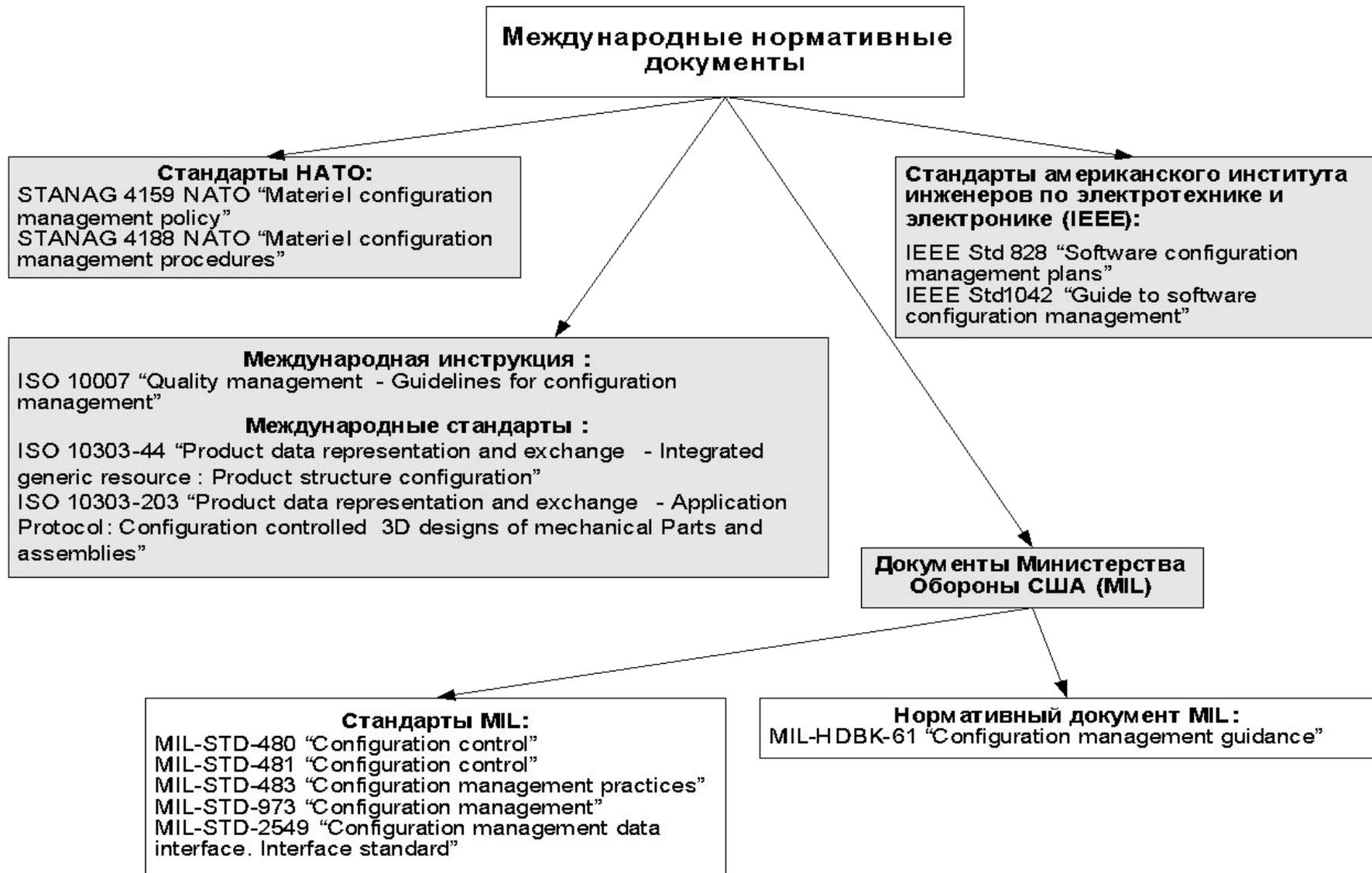


Управление конфигурацией (Configuration Management, CM)

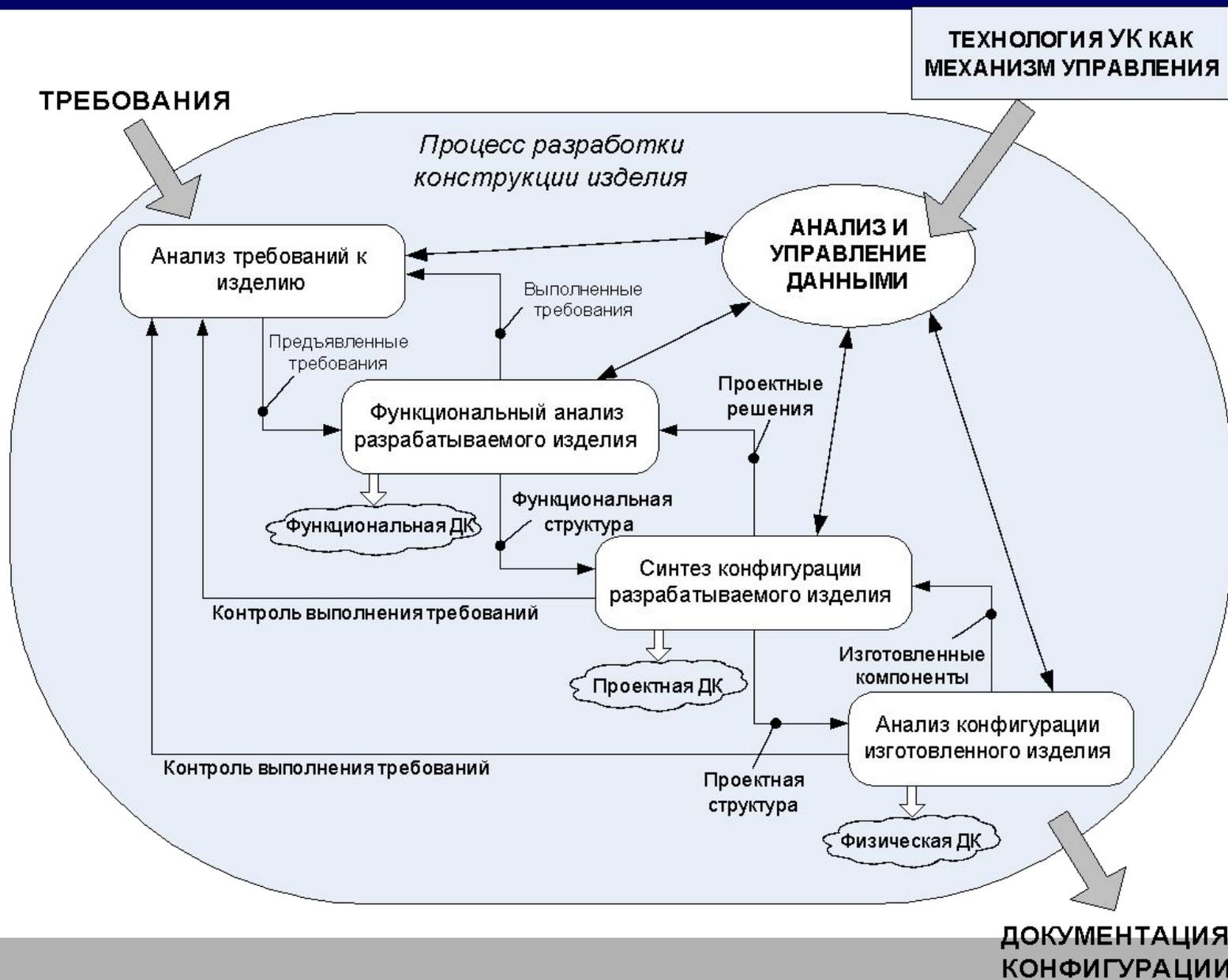
УК - управленческая технология, направленная на установление и поддержание соответствия эксплуатационных, функциональных и физических атрибутов (свойств, характеристик) изделия заданным требованиям (в том числе требованиям заказчика), в процессе создания и преобразования информационных моделей этого изделия в течение его ЖЦ.

Технология УК применима только к изделиям, имеющим достаточно сложную функциональную структуру. Из нее могут быть выделены **объекты конфигурации (ОК)**, выполняющие в составе конечного изделия четко определенные функции и обладающие значимым набором характеристик, сопоставимых с подмножеством требований, предъявляемых к конечному изделию.

Нормативная база управления конфигурацией

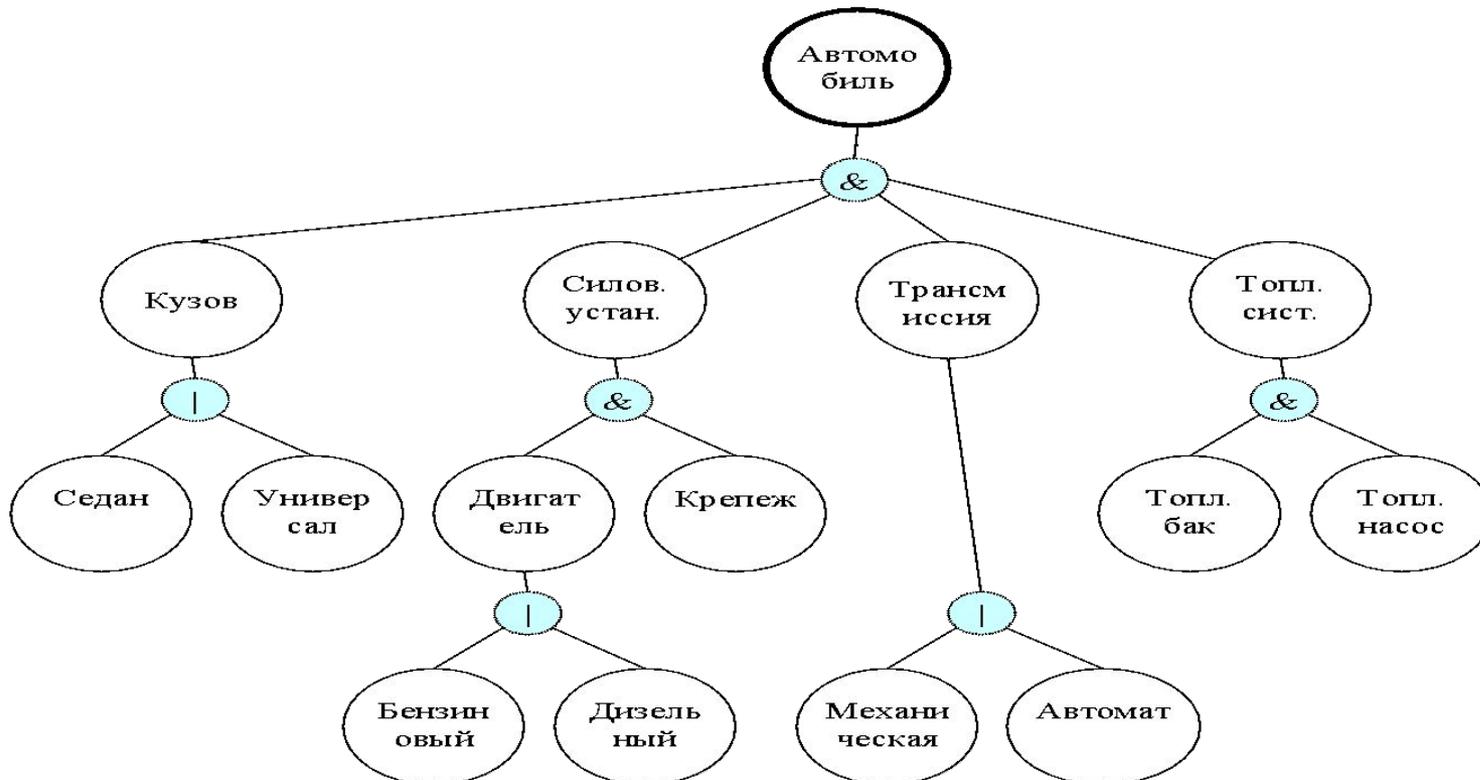


Этапы управления Конфигурацией



Онтологии предметной области и задачи

- Функциональная конфигурация (ФК) полностью соответствует определению онтологии предметной области.
- Она представляет собой формализованное описание объектов определенного типа и знаний относительно их свойств.
- Множество ОК представляется в виде И-ИЛИ графа, определяющего структуру конфигурации изделия



Применение метода программирования в ограничениях

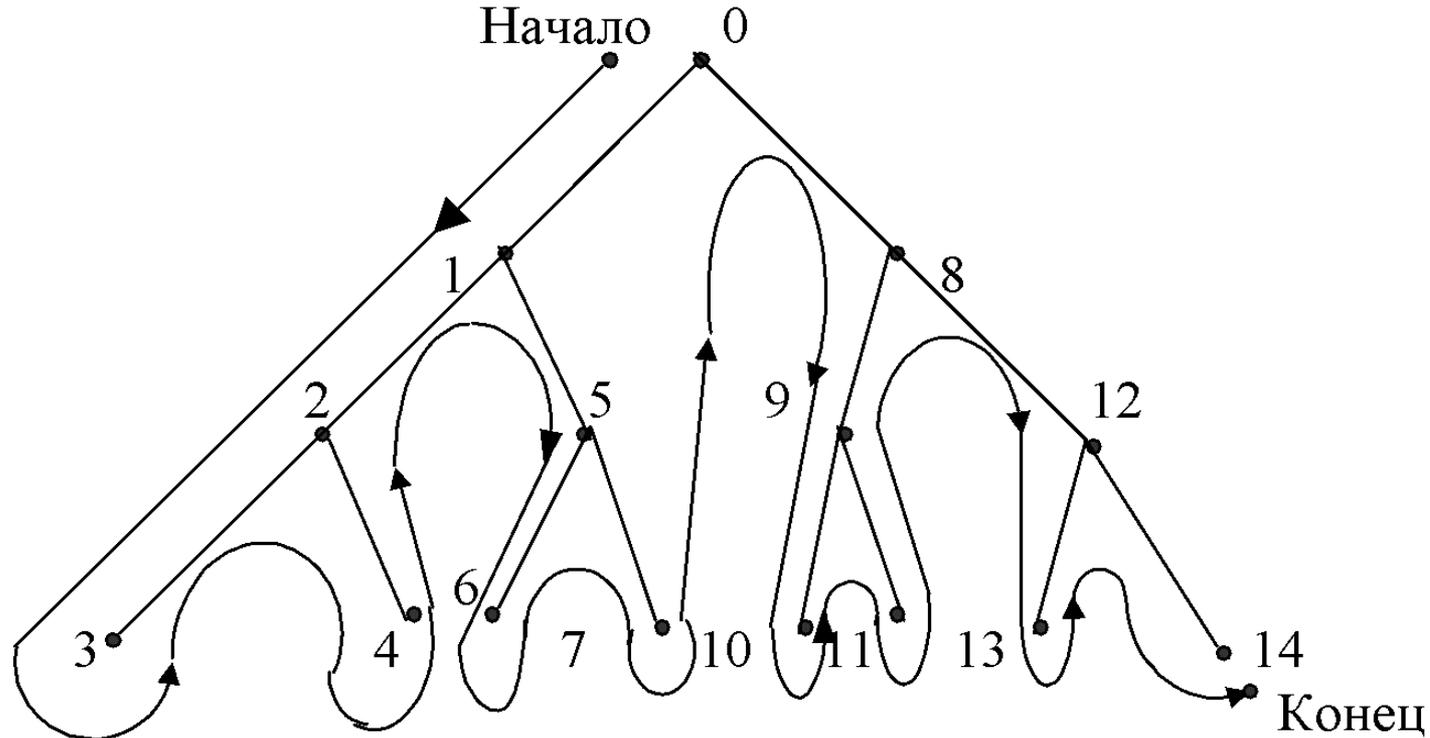
- ❖ Метод программирования в ограничениях применяется для поиска конфигурации, наиболее точно удовлетворяющей требованиям.
- ❖ Метод программирования в ограничениях, называемый также методом удовлетворения (распространения) ограничений («**Constraint Satisfaction Problem**» или **CSP**) является одним из подходов искусственного интеллекта, применяемых для решения различных сложных и нестандартных задач, в том числе подобных комбинаторных.
- ❖ Программирование в ограничениях требует только описания задачи, но не заставляет разработчика определять алгоритм ее решения.
- ❖ Задача и модель представляются в этом случае как неупорядоченная совокупность отношений, которые соответствуют связям, существующим между переменными задачи.
- ❖ Эти отношения, называемые общим термином "ограничения", могут иметь вид уравнений, неравенств, логических выражений и т. п.

Процедура полного перебора

- ❖ Определяем n вложенных циклов (где n - количество ОК подлежащих определению)
- ❖ Для каждого из циклов определяем диапазон перебора, соответствующий числу компонентов из справочника, соответствующих текущему ОК
- ❖ Внутри циклов:
 - Поместить в текущую конфигурацию n компонентов с индексами $i_1 \dots i_n$
 - Проверить ограничения и требования
 - Если все условия для данной конфигурации выполняются, то добавить текущую конфигурацию в список пригодных конфигураций

Поиск с возвратами

- ◆ Данный метод предполагает перебор всех возможных комбинаций путем обхода дерева вариантов компоновки с возвратами при не выполнении условий



Другие методы поиска решений

- ❖ **Метод ветвей и границ** является вариацией полного перебора с отсеком подмножеств допустимых решений, заведомо не содержащих оптимальных решений. Применение данного метода позволяет ускорить процесс решения за счет ограничения пространства поиска
- ❖ **Жадный алгоритм** – начиная с пустой конфигурации, происходит локальное улучшение. ОК наполняются компонентами с наиболее предпочтительными параметрами. Процесс продолжается пока не будет найдено приемлемое решение.
- ❖ **Генетические алгоритмы** – предполагает описание процедуры мутации и скрещивания особей, а также определение функции выживаемости особи и стратегии управления популяцией.
- ❖ **Алгоритмы случайного поиска** – поиск новых конфигураций происходит путем случайного выбора направления поиска (подбираемого ОК).
- ❖ **Метод покоординатного спуска** – предполагает перемещение в пространстве состояний конфигурации путем последовательной замены компонентов в отдельных ОК.
- ❖ **Комбинированные методы**

Техническое обслуживание и ремонт оборудования (ТОРО)

Комплекс операций или операция по поддержанию работоспособности или исправности изделия при использовании по назначению, ожидании, хранении и транспортировке.

- ❖ Устранение отказов оборудования;
- ❖ Инспекция в определенном объеме с определенной периодичностью;
- ❖ Плановая замена деталей и ремонт оборудования по состоянию, наработке;
- ❖ Планирование ремонтов и обслуживания.

Виды ТОРО

- ❖ **Обслуживание «по событию»** (например, устранение поломки оборудования);
- ❖ **Регламентное обслуживание** (в паспорте производителя описано в каком режиме и какое обслуживание необходимо выполнять для поддержания работоспособности оборудования);
- ❖ **«По состоянию»** (на основании оценки делается прогноз, когда оборудование надо выводить в ремонт)

Онтология проблемной области ТОРО

Справочник изделий

Справочник работ

