ОПРЕДЕЛЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

*Искусственный интеллект* – это научное направление, в рамках которого ставятся и решаются задачи аппаратного и программного моделирования тех видов деятельности, которые традиционно считаются интеллектуальными, т.е. присущими только человеку.

В последнее время России искусственным интеллектом называют практические (программные) продукты этого направления

ОСНОВНЫЕ СКВОЗНЫЕ ЦИФРОВЫЕ ТЕХНОЛОГИИ ЦИФРОВОЙ ЭКОНОМИКИ

Основные меры государственной политики Российской Федерации по созданию необходимых условий для развития цифровой экономики Российской Федерации, в которой данные в цифровой форме являются ключевым фактором производства во всех сферах социально- экономической деятельности, что повышает конкурентоспособность страны, качество жизни граждан, обеспечивает экономический рост и национальный суверенитет.

Основными сквозными цифровыми технологиями, которые входят в рамки настоящей Программы, являются:

большие данные;

нейротехнологии и искусственный интеллект;

системы распределенного реестра;

квантовые технологии;

новые производственные технологии;

промышленный интернет;

компоненты робототехники и сенсорика;

технологии беспроводной связи;

технологии виртуальной и дополненной реальностей.

ПОЯВЛЕНИЕ ТЕРМИНА «ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ»

Термин «искусственный интеллект» появился в США в 1956 году, когда впервые заговорили о машинах. В 1964 году появилась первая экспертная система «Dendral». В 80-х гг. появился японский проект ЭВМ 5-го поколения, где предлагались: голосовой ввод, машины знаний и др.

В последнее время за рубежом часто используют термины «компьютерный интеллект» и «интеллектуальные вычисления».

Intellectual (способности человека) и intelligent (комп.)

ПРЕДЫСТОРИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Философия (с 426 г до н.э.)

Математика (с 800 г. н.э.)

Экономика (с 1776 г.)

Неврология (с 1861 г.)

Психология (с 1879)

Вычислительная техника (с 1940)

Теория управления и кибернетика (с 1948)

Лингвистика (с 1957)

Искусственный интеллект (1956)

Маккалок и Питс (искусственный нейрон) -1943

ЭТАПЫ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Ранний энтузиазм, большие ожидания (1952-1959)

Столкновение с реальностью

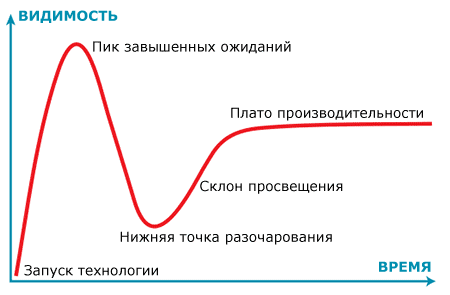
(1966-1973)

Системы, основанные на знаниях :могут ли они стать ключом к успеху

(1969-1979)

Превращение искусственного интеллекта в индустрию (с 1980 )

КРИВАЯ ГАРТНЕРА



ПРЕВРАЩЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ИНДУСТРИЮ (С 1980 …)

1981г. – японский проект ЭВМ V поколения – 10-летний план по разработке интеллектуальных компьютеров, работающих под управлением языка Prolog

В США сформирована корпорация МСС как научно-исследовательский консорциум, предназначенный для обеспечения конкурентоспособности американской промышленности

Амбициозные цели, поставленные перед специалистами ИИ в проектах МСС и ЭВМ V поколения, так и не были достигнуты.

В целом в индустрии ИИ произошел бурный рост, начиная с нескольких млн $ в 1980 г. и заканчивая миллиардами $ в 1988.

После этого наступил период, получивший название «зимы искусственного интеллекта», в течение которого пострадали многие компании, поскольку не сумели выполнить своих заманчивых обещаний.

ЭТАПЫ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Возвращение к нейронным сетям (с 1986)

Превращение ИИ в науку (с 1987)

Подходы, основанные на использовании скрытых марковских моделей

Технология анализа скрытых закономерностей в данных (data mining)

Признание важности теории вероятностей и теории решений для ИИ (байесовские сети)

Появление подхода, основанного на использовании интеллектуальных агентов (с 1995)

Представление знаний (онтологическая инженерия) (с 1990-х)

СОВРЕМЕННОЕ СОСТОЯНИЕ РАЗРАБОТОК   
В ОБЛАСТИ ИИ

Автономное планирование и составление расписаний

Ведение игр

Автономное управление

Диагностика

Планирование снабжения

Робототехника

Понимание естественного языка и решение задач

*Основные направления искусственного интеллекта:*

доказательство теорем

модели игр

распознавание образов

использование естественного языка

робототехника

экспертные системы

инженерия знаний

*Доказательство теорем.* Перекрывается с определенными областями математики. Связано с использованием исчисления высказываний.

*Модели игр.* Особое внимание уделяется шахматам.

*Распознавание образов.* Связано с распознаванием зрительных и слуховых образов.

*Использование естественного языка.* Большое внимание уделялось системам «вопрос-ответ» и системам автоматического перевода.

*Робототехника.* Имеет непосредственную практическую ценность.

*Экспертные системы.* Представляют большой раздел систем искусственного интеллекта, будут подробно рассматриваться в данном курсе.

*Инженерия знаний.* Эта область не является самостоятельной, но тесно связана, например, с экспертными системами.

*Два основных направления развития (пути создания) систем искусственного интеллекта:*

*Классическое*

Связан с попытками моделировать функции зрительного анализа, распознавания звуков речи, логических операций, переводов с одного языка на другой. Возникли затруднения при объяснении принципов работы человеческого мозга. Исследователи сосредоточили внимание на решении инженерных проблем и создании экспертных систем.

*Альтернативный путь создания ИИ*

Основан на построении сетей из нейроно-подобных элементов, осуществляющих параллельную обработку информации.

Оба направления стремятся выйти на создание прикладных систем, демонстрирующих преимущество этих систем перед традиционными, использующими только формальные методы.

АЛЬТЕРНАТИВНЫЙ ПУТЬ СОЗДАНИЯ ИИ

*Нейроинтеллект* (искусственные нейронные сети). Его техническое воплощение - многоуровневая, адаптивная, обучающаяся сеть из нейроно-подобных элементов (нейрокомпьютер).

Полагают, что ИНС позволят разрешить противоречие между возрастающими требованиями к быстродействию вычисления и техническим возможностями ЭВМ (аналогично живым организмам, которые при сравнительно небольшом быстродействии элементов нервной системы достигают высокой скорости распознавания образов).

О РАЗВИТИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В РФ

УКАЗ Президента РФ от 10.10.2019:

Утвердить прилагаемую Национальную стратегию развития ИИ на период до 2030 г.

Правительству Российской Федерации:

а) до 15 декабря 2019 г. обеспечить внесение изменений в национальную программу «Цифровая экономика РФ», в том числе разработать и утвердить федеральный проект «Искусственный интеллект»;

…

ОБЩИЕ ПОЛОЖЕНИЯ: Основные понятия (1)

а) Искусственный интеллект - комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая самообучение и поиск решений без заранее заданного алгоритма) и получать при выполнении конкретных задач результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека. Комплекс технологических решений включает в себя информационно-коммуникационную инфраструктуру, программное обеспечение (в том числе в котором используются методы машинного обучения), процессы и сервисы по обработке данных и поиску решений;

б) Технологии искусственного интеллекта - технологии, основанные на использовании искусственного интеллекта, включая компьютерное зрение, обработку естественного языка, распознавание и синтез речи, интеллектуальную поддержку принятия решений и перспективные методы искусственного интеллекта;

в) Перспективные методы искусственного интеллекта - методы, направленные на создание принципиально новой научно-технической продукции, в том числе в целях разработки универсального (сильного) искусственного интеллекта (автономное решение различных задач, автоматический дизайн физических объектов, автоматическое машинное обучение, алгоритмы решения задач на основе данных с частичной разметкой и (или) незначительных объемов данных, обработка информации на основе новых типов вычислительных систем, интерпретируемая обработка данных и другие методы);

Развитие искусственного интеллекта в России и в мире

6. Развитие информационных систем, помогающих человеку принимать решения, началось с появления в 1950-х годах экспертных систем, описывающих алгоритм действий по выбору решения в зависимости от конкретных условий. На смену экспертным системам пришло машинное обучение (?!), благодаря которому информационные системы самостоятельно формируют правила и находят решение на основе анализа зависимостей, используя исходные наборы данных (без предварительного составления человеком перечня возможных решений), что позволяет говорить о появлении искусственного интеллекта (?!).

8. Машинное обучение характеризуется рядом особенностей. Во-первых, для поиска вычислительной системой непредвзятого решения требуется ввести репрезентативный, релевантный и корректно размеченный набор данных.

Во-вторых, алгоритмы работы нейронных сетей крайне сложны для интерпретации и, следовательно, результаты их работы могут быть подвергнуты сомнению и отменены человеком.

Отсутствие понимания того, как искусственный интеллект достигает результатов, является одной из причин низкого уровня доверия к современным технологиям искусственного интеллекта и может стать препятствием для их развития.

ДВА СОВРЕМЕННЫХ НАПРАВЛЕНИЯ ИИ

Сейчас выделяют два направления ИИ:

1) связанное с данными – интеллектуальный анализ данных и машинное обучение;

2) связанное о знаниями (классическое)

Поскольку ИНС работают как «черный ящик» (выдают результат, но не объясняют его), появился термин «объяснимый искусственный интеллект», для реализации которого пытаются интегрировать ИНС и экспертные системы

КОДЕКС ЭТИКИ ИИ

26 октября 2021 г. в России принят «Кодекс этики ИИ».  Его разработал Альянс в сфере Искусственного интеллекта при поддержке аналитического центра при Правительстве РФ, Минэкономразвития России в соответствии с положениями Национальной стратегии развития искусственного интеллекта до 2030 года.

Кодекс распространяется на аспекты создания, внедрения и использования технологий искусственного интеллекта на всех этапах его жизненного цикла, которые в настоящее время не урегулированы законодательством Российской Федерации и/или актами технического регулирования. Главное в нём: гуманистический подход.

Утверждают, что Россия одна из первых в мире сформулировала пять рисков и угроз, которые сопровождают внедрение "цифры" в жизнь, - дискриминация, потеря приватности, потеря контроля над ИИ, причинение вреда человеку ошибками алгоритма, применение в неприемлемых целях. Все они включены в принятый "Кодекс этики искусственного интеллекта" как угрозы правам и свободам человека.

*В ответ на риски кодекс утвердил основные принципы внедрения ИИ – прозрачность, правдивость, ответственность, надежность, инклюзивность, беспристрастность, безопасность и конфиденциальность.*

ДАННЫЕ - ИНФОРМАЦИЯ – ЗНАНИЯ

Рассел Аккоф, один из классиков исследования операций, предложил следующую, вполне убедительную иерархию:

[данные – информация – знания – понимание – мудрость].

*Данные* по Р. Аккофу – это некоторые неупорядоченные символы, рассматриваемые безотносительно к какому-либо контексту.

*Информация* – это выделенная и упорядоченная часть сообщения, обработанная для использования, то есть отвечающая на вопрос: «Кто?, Что?, Где?, Когда?»

*Знание* – это выявленные тенденции или существенные связи между фактами и явлениями, представленные в информации.

*Понимание* – это осознание закономерностей, содержащихся в разрозненных знаниях, позволяющее ответить на вопрос: «Почему?

*Мудрость* – взвешенное, оцененное понимание закономерностей с точки зрения прошлого и будущего

На практике три последних позиции объединяют в понятии «знания»

ДАННЫЕ И ЗНАНИЯ:   
РАСШИРЕННОЕ ОПРЕДЕЛЕНИЕ ЗНАНИЙ

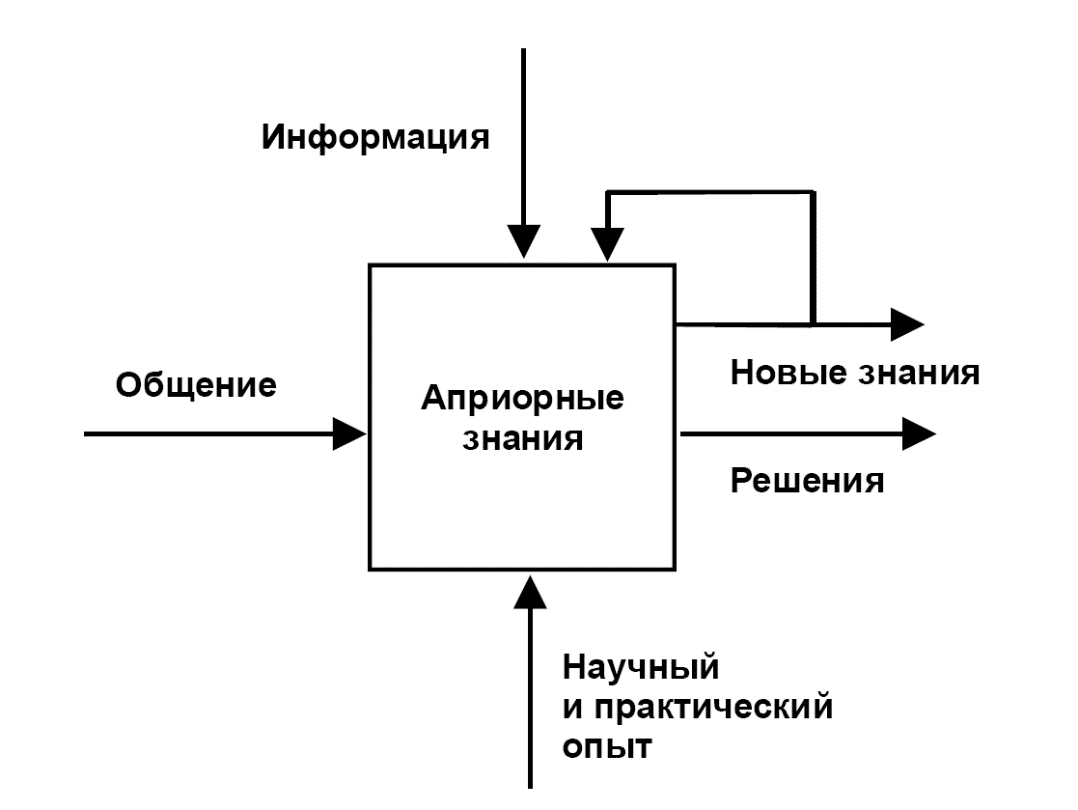
Общее определение знаний:

*Знания* – совокупность сведений, образующих целостное описание, соответствующее некоторому уровню осведомленности об описываемых вопросе, предмете, проблеме и т.п.

В области искусственного интеллекта в качестве рабочего берется следующее определение знаний:

*Знания* – это основные закономерности в предметной области, позволяющие человеку решать конкретные производственные, научные и другие задачи, то есть знания интерпретируются как факты, понятия, взаимосвязи, оценки, правила, эвристики, а также стратегии принятия решения в этой области (стратегические знания).

ЗНАНИЯ СПЕЦИАЛИСТА КАК «ЧЕРНЫЙ ЯЩИК»



КЛАССИФИКАЦИЯ ЗНАНИЙ

Вся совокупность знаний может быть разделена на следующие классы:

• эмпирические знания;

• теоретические знания;

• личностные знания;

• организационные знания;

• неявные знания (brain-знания);

• явные знания (e-знания).

К *неявным знаниям (tacit knowledge, или brain-знания)*, с учетом вышеизложенных определений, относятся опыт, мастерство, культура мышления, интуиция, хранящиеся в нейронных структурах головного мозга как результат генетической наследственности, образования и приобретенного жизненного опыта. Неявное знание – это способность человека к адаптации в меняющихся условиях. Неявное знание работает через эвристики – некие гибкие руководства к действию, формирующиеся посредством проб и ошибок в результате наблюдений и эволюции мыслительной деятельности мозга.

ОСНОВНЫЕ ВИДЫ ЗНАНИЙ В ОБЛАСТИ ИТ

Выделяют следующие *виды знаний:*

*Декларативные знания* – это знания, которые записываются в явном виде и используются также в явном виде (книги, учебники, докменты).

*Процедурные знания* – это знания, которые хранятся в виде процедур, программ и т.п.

*Эвристические знания* – это знания, накапливаемые в системе в процессе функционирования, а также заложенные в ней, но не имеющие статуса абсолютной истинности в данной предметной области. Их часто сравнивают с отображением в базе знаний неформализованного опыта решения задач.

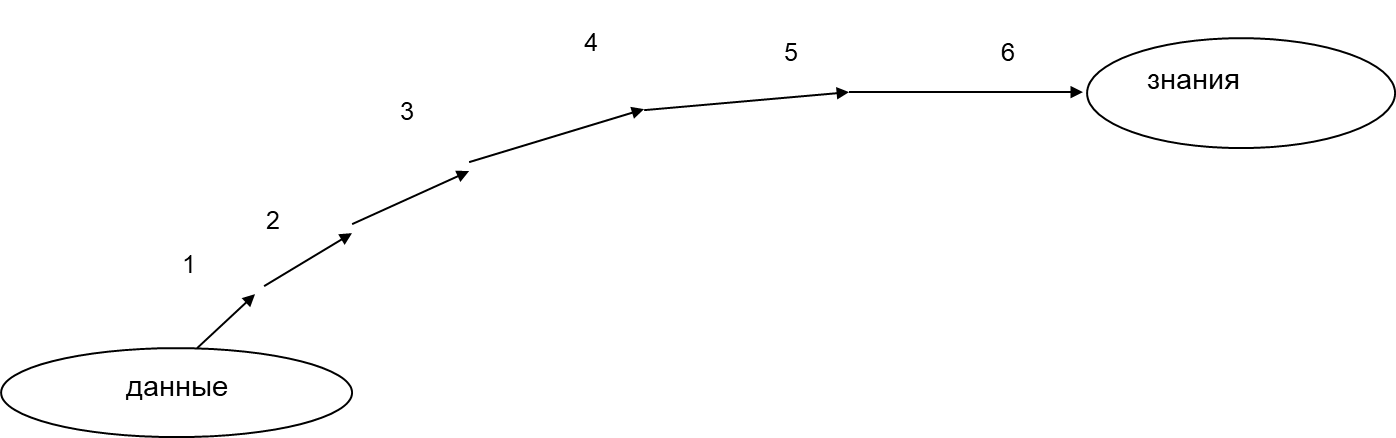
ДАННЫЕ И ЗНАНИЯ:   
ЗНАНИЯ КАК ОБОБЩЕННЫЕ, УСЛОЖНЕННЫЕ ДАННЫЕ

От противопоставления данных и знаний на начальных этапах сейчас перешли к представлению знаний как обобщенных, усложненных данных, обладающих по сравнению с элементарными данными рядом новых свойств.

Эти свойства соответствуют этапам перехода от файлового представления данных к базам данных иерархической, сетевой, реляционной структур с СУБД и далее к базам знаний и системам управления ими.

Иначе говоря, эти свойства соответствуют переходу от неявного представления знаний в традиционных программных комплексах к явному их представлению в системах, основанных на знаниях.

ДАННЫЕ И ЗНАНИЯ:   
ЗНАНИЯ КАК ОБОБЩЕННЫЕ, УСЛОЖНЕННЫЕ ДАННЫЕ



*1 – внутренняя интерпретируемость*

*2 – наличие внутренней структуры связей*

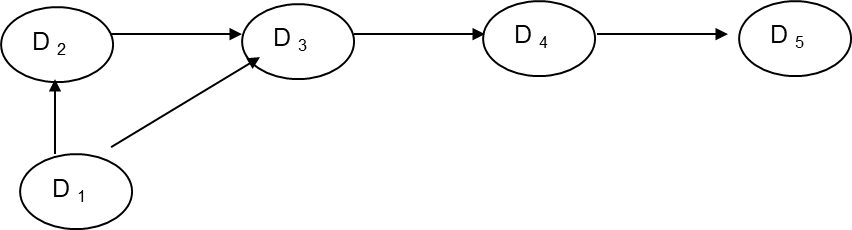
*3 – наличие внешней структуры связей*

*4 – шкалирование*

*5 – погружение в пространство с семантической метрикой*

*6 – наличие активности*

ОСНОВНЫЕ ФОРМЫ ДАННЫХ И ЗНАНИЙ:  
ФОРМЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ



D 1 – результаты наблюдения над объектами или данные в памяти;

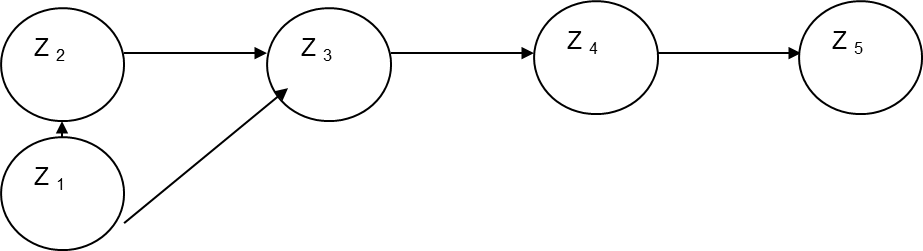
D 2 – фиксация данных на материальном носителе;

D 3 – модель данных;

D 4 – данные на языке описания данных;

D 5 – база данных на машинных носителях информации.

ОСНОВНЫЕ ФОРМЫ ДАННЫХ И ЗНАНИЙ:  
ФОРМЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ



Z1 – знания в памяти человека;

Z2 – материализованные знания (статьи, учебники);

Z3 – поле знаний (полуформализованное описание

Z1 и Z2);

Z4 – знания на языке представления знаний (модель

знаний), т.е. формализация Z3;

Z5 – БЗ в ЭВМ, т.е. на машинных носителях информации.

Реализация Z3 и есть построение модели предметной области.

ОТЛИЧИЕ ЗНАНИЙ ОТ ДАННЫХ:  
СВОЙСТВА, ОТСУТСТВУЮЩИЕ У ТРАДИЦИОННО ОРГАНИЗОВАННЫХ ДАННЫХ

*Внутренняя интерпретируемость.*

Вместе с традиционной информационной единицей элементом данных – в памяти ЭВМ можно хранить схему имен, связанных с этой единицей. Наличие схемы имен позволяет информационной системе «знать», что хранится в памяти, и уметь отвечать на запросы о содержимом БЗ.

*Рекурсивная структурируемость.*

Информационные единицы могут при необходимости расчленяться на более мелкие единицы и объединяться в более крупные.

*Взаимосвязь информационных единиц.*

Между информационными единицами возможно установление разнообразных отношений, отражающих семантику и прагматику связей, явлений и факторов.

ОТЛИЧИЕ ЗНАНИЙ ОТ ДАННЫХ:  
СВОЙСТВА, ОТСУТСТВУЮЩИЕ У ТРАДИЦИОННО ОРГАНИЗОВАННЫХ ДАННЫХ

*Возникновение семантического пространства.*

Знания не могут быть бессистемными, а должны быть взаимосвязанными и взаимозависимыми в общем для них семантическом пространстве. Структурирование знаний направлено на формирование семантического пространства.

*Активность знаний – определяющее свойство.*

С начала своего развития программирование опиралось на первичность процедур и вторичность данных, т.е. процедуры отражали способ решения задачи и активизировали необходимые данные, которые пассивно хранились. В БЗ знания являются активными, т.е. способ представления знаний, как правило, отражает способ решения задачи. Активность знаний обозначает, что мы можем получить знания, которые в явном виде в базе знаний не хранятся.

ДАННЫЕ И ЗНАНИЯ:  
ОТЛИЧИЕ БАЗЫ ДАННЫХ ОТ БАЗЫ ЗНАНИЙ

В функции БД, как правило, не входит анализ соответствующих элементов и взаимосвязей моделей с особенностями окружающей действительности.

Эта задача решается специалистами соответствующего профиля, а затем БД функционирует так, как будто ее содержимое тождественно реальному миру.

В обычных БД осуществляется хранение и поиск фактов о мире, а также некоторые операции объединения и отбора фактов, но не обеспечивается функция восприятия, необходимая для установления соответствия между состоянием внешнего мира и состоянием БД.

Область применения БД ограничивается сферой интеллектуальной деятельности, в которой функции восприятия и действия выполняются пользователями и обслуживающим персоналом БД.

*Задачи, решаемые совместно БД и ее разумным партнером, похожи на задачи, решаемые экспертной системой.*

ЭКСПЕРТНЫЕ (ИНТЕЛЛЕКТУАЛЬНЫЕ) СИСТЕМЫ  
(используемые термины)

*Экспертная система* – это интеллектуальная система, предназначенная для оказания консультационной помощи специалистам, работающим в некоторой предметной области.

*Интеллектуальная система* – это техническая или программная система, способная решать задачи, считающиеся творческими и принадлежащие конкретной предметной области, знания о которой хранятся в памяти *интеллектуальной системы.*

*Система* *баз знаний* – это интеллектуальная система, функционирование которой определяется совокупностью знаний о предметной области, в которой она используется.

ТИПЫ ЭКСПЕРТНЫХ СИСТЕМ

Существует *два типа экспертных систем:*

системы тиражирования знаний (для специалистов, чей профессиональный уровень не слишком высок);

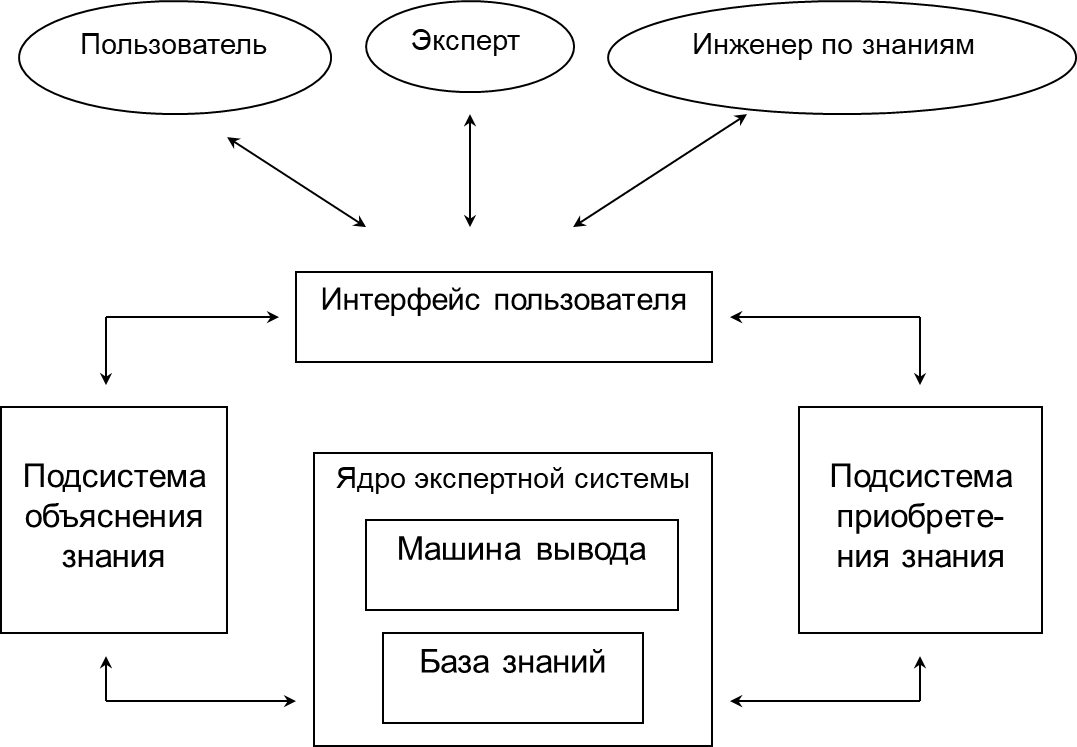
системы получения новых знаний (для специалистов высокой квалификации).

В БЗ систем тиражирования знаний хранятся знания, полученные от экспертов. Примером такой системы может являться АСДУ (автоматизированная система диспетчерского управления).

Особенностью экспертных систем получения новых знаний является наличие в них подсистемы объяснений, объясняющих, каким образом был получен тот или иной вывод.

Существуют системы третьего типа (нового поколения), например, система G2 (ее используют в системах реального времени для отслеживания показаний технических приборов).

АРХИТЕКТУРА ЭКСПЕРТНОЙ СИСТЕМЫ



Комментарии:

Инженер по знаниям (или инженер-когнитолог) – специалист, извлекающий знания для проектирования и заполнения базы знаний. Он же может быть разработчиком экспертной системы.

Ядро экспертной системы – база знаний и машина вывода. Последнюю считают аналогом СУБД и иногда называют Системой управления базой знаний (СУБЗ)

Подсистема приобретения знаний позволяет вводить в базу знаний новые понятия, которые ранее в ней отсутствовали

ИЗВЛЕЧЕНИЕ ЗНАНИЙ

Большинство разработчиков ЭС считают, что процесс извлечения знаний остается самым “узким” местом при построении промышленных ЭС. При этом часто приходится самостоятельно разрабатывать методы извлечения знаний , сталкиваясь со следующими трудностями:

организационные неувязки,

неудачный способ извлечения знаний, не совпадающий с их структурой в данной области,

неадекватная модель (язык) для представления знаний,

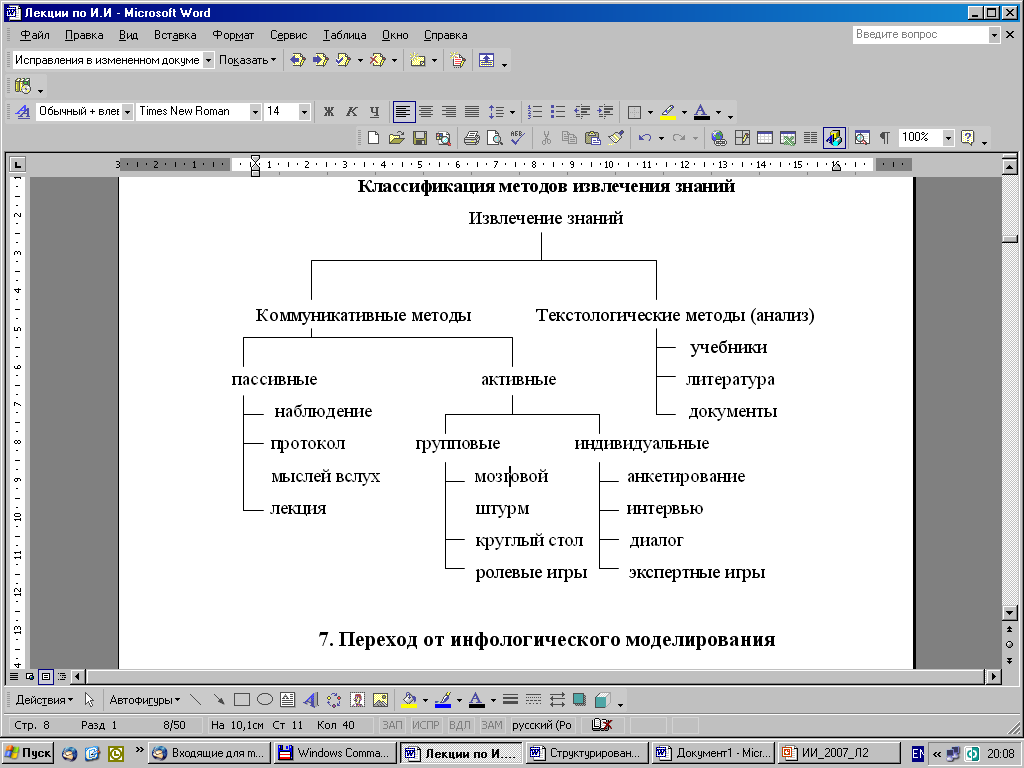
неумение наладить контакт с экспертом,

терминологический разнобой,

нарушение целостной картины знаний при извлечении фрагментов.

Метод извлечения знаний определяется инженером по знаниям в зависимости от конкретной ситуации и задачи.

КЛАССИФИКАЦИЯ МЕТОДОВ ИЗВЛЕЧЕНИЯ ЗНАНИЙ



ИЗВЛЕЧЕНИЕ ЗНАНИЙ

Методы извлечения знаний делят на две группы, в зависимости от источника знаний:

Коммуникативные методы

Текстологические методы:

Анализ учебников

Анализ литературы

Анализ документов

Извлечение знаний начинают с применения текстологических методов.

Цель их применения – сформировать тезаурус, т.е. освоить терминологию предметной области

Коммуникативные методы (основанные на общении с экспертом) делятся на:

Активные

Пассивные:

Наблюдение

Протокол мыслей вслух

Лекция

Часто извлечение знаний сравнивают с умением «сделать осознанными неосознаваемые экспертом знания», т.е. описать алгоритм принятия решений, которые эксперт часто принимает, «не задумываясь", на основе своего опыта, эрудиции и интуиции

Активные методы делятся на два типа

Индивидуальные:

Анкетирование

Интервью

Диалог

Групповые:

Мозговой штурм

Круглый стол

Ролевые игры

Экспертные игры

ГИБРИДНЫЕ ЭКСПЕРТНЫЕ СИСТЕМЫ

В общем случае представленный выше программный комплекс можно назвать гибридной экспертной системой.

Существует два типа гибридных экспертных систем:

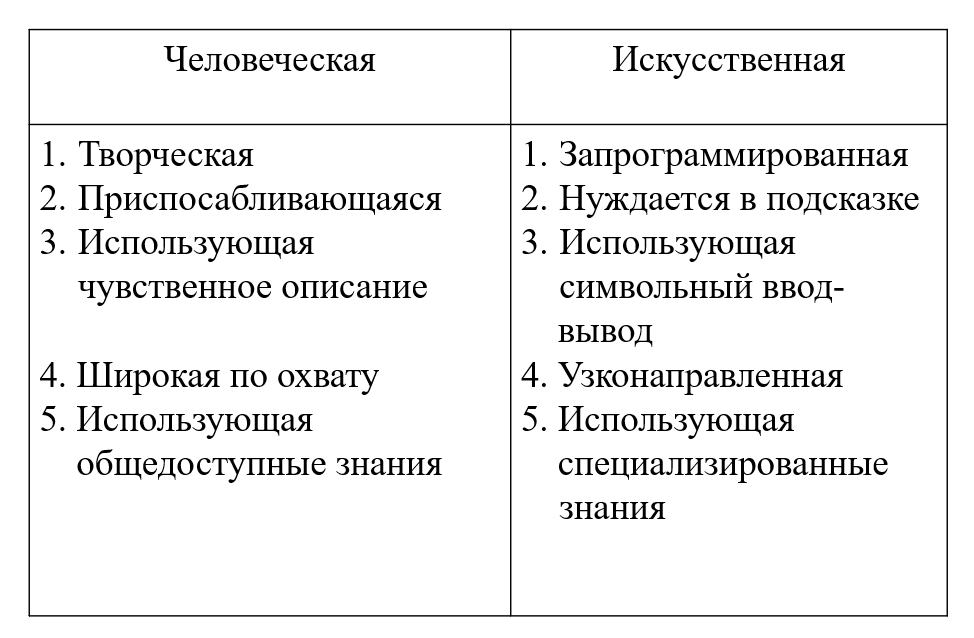
использующие разные модели представления знаний;

такие, которые кроме БД и БЗ, включают и прикладные программы конкретной предметной области.

ЧЕЛОВЕЧЕСКАЯ И ИСКУССТВЕННАЯ КОМПЕТЕНТНОСТЬ:  
 ДОСТОИНСТВА ИСКУССТВЕННОЙ КОМПЕТЕНТНОСТИ

|  |  |
| --- | --- |
| Человеческая | Искусственная |
| Непрочная  Труднопередаваемая  Непредсказуемая  Дорогая | Постоянная  Легкопередаваемая  Устойчивая  Приемлемая по затратам |

ЧЕЛОВЕЧЕСКАЯ И ИСКУССТВЕННАЯ КОМПЕТЕНТНОСТЬ:  
НЕДОСТАТКИ ИСКУССТВЕННОЙ КОМПЕТЕНТНОСТИ



ОСНОВНЫЕ ВИДЫ ДЕЯТЕЛЬНОСТИ   
ЭКСПЕРТНЫХ СИСТЕМ

Интерпретация (описание ситуации по информации, поступающей от датчика)

Прогноз (описание вероятностных последствий заданной ситуации)

Диагностика (выявление причин неправильного функционирования системы по результатам наблюдения)

Проектирование (построение конфигурации объектов при заданных ограничениях)

Планирование (определение последовательности действий)

Наблюдение (сравнение с ожидаемыми результатами)

Отладка (составление «рецептов» устранения неправильного функционирования системы)

Ремонт (выполнение последовательности действий, предписанных инструкцией)

Обучение (диагностика, отладка обучаемых)

Управление (поведение системы как целого)

ОТЛИЧИЕ ЭКСПЕРТНОЙ СИСТЕМЫ   
ОТ ТРАДИЦИОННЫХ КОМПЬЮТЕРНЫХ ПРОГРАММ

*Экспертная система* – это набор *компьютерных программ*, которые решают задачи в интересующей нас предметной области и содержат *компонент решения проблемы и компонент поддержки .*

Компонент поддержки позволяет пользователю работать с другой программой и может включать специальные отладочные средства для работы с программой. Этот компонент помогает разработчику тестировать программу, пользователю – взаимодействовать с экспертной системой, эксперту – модифицировать данные знания экспертной системы. Также компонент поддержки может включать различные средства ввода-вывода информации, в том числе, о ходе работы системы, и средства объяснения.

В архитектуре экспертной системы компонент решения проблемы образуют база знаний и машина вывода; компонент поддержки - подсистема объяснения, подсистема приобретения знаний, интерфейс пользователя.

*Под экспертной системой понимается программа, обладающая следующими свойствами:*

Компетентность, т.е. ЭС должна:

достигать экспертного уровня решений

быть «умной» (делать выводы, давать объяснения и др.)

адекватно реагировать на запросы

Символьное рассуждение, т.е. ЭС должна:

представлять знания в символьном виде

уметь переформулировать символьные знания

Глубина, т.е. ЭС должна:

уметь работать в предметной области, содержащей трудные задачи

уметь использовать сложные правила

Самосознание, т.е. ЭС должна:

уметь исследовать свои рассуждения

объяснять свои действия

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

Существуют *два класса инструментальных средств разработки экспертных систем:*

Пустые экспертные системы

Оболочки экспертных систем

В *пустых экспертных системах* БЗ не заполнена. При ее использовании специалисту необходимо спроектировать БЗ и внести ее в экспертную систему в соответствии с правилами. Ее особенностью является то, что она может быть эффективна лишь для одного класса задач, т.к. принятые в системе способы представления знаний и рассуждений ориентированы на этот класс задач.

*Оболочки экспертных систем* являются инструментальным средством для проектирования и создания экспертных систем. В ее состав входят: средства проектирования БЗ с различными формами представления знаний и выбора режима работы решения задач (этот класс инструментальных средств более универсальный)

СТАДИИ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ



ОБЛАСТИ ПРИМЕНЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ:

Военное дело

Технология

Инженерное дело

Информатика

Компьютерные системы

Космические системы

Математика

Медицина

Метеорология

Промышленность

Сельское хозяйство

Управление процессами

Физика

Химия

Электроника

Юриспруденция

ПРЕДШЕСТВЕННИКИ ЭКСПЕРТНЫХ СИСТЕМ:  
 ДОКУМЕНТАЛЬНЫЕ ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ

Информационные системы делят на фактографические (ориентированные на работу со структурированными данными - фактами) и документальные (ориентированные на работу со слабо структурированными или неструктурированными данными – текстами).

Первые системы автоматизированного информационного поиска документов появились еще в 60-х годах, развитые коммерческие информационно-поисковые системы, ориентированные на накопление и обработку текстовых документов, получили распространение лишь в конце 80-х – начале 90-х годов.

*Документальные информационно-поисковые системы* можно считать прообразом экспертных систем

ДОКУМЕНТАЛЬНЫЕ ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ (ИПС)

В фактографических ИС единичным элементом данных, имеющих отдельное смысловое значение, является запись, образуемая конечной совокупностью полей-атрибутов. В документальных ИС единичным элементом данных является неструктурированный на более мелкие элементы *документ*.

Основной задачей документальных ИС является накопление и предоставление пользователю документов, содержание, тематика, реквизиты и т.п. которых адекватны его информационным потребностям. Поэтому документальную ИС определяют как единое хранилище документов с инструментарием поиска и отбора необходимых документов. Поисковый характер документальных ИС исторически определил еще одно их название – *информационно-поисковые системы (ИПС).*

СПЕЦИФИКА ДОКУМЕНТАЛЬНЫХ ИПС

Поиск информации (данных) осуществляется и в фактографических ИС. Специфика документальных ИПС заключается в том, что они удовлетворяют информационные запросы пользователя, предоставляя ему документы, в которых содержится интересующая пользователя информация.

В зависимости от особенностей реализации хранилища документов и механизма поиска документальные ИПС можно разделить на две группы:

системы на основе индексирования;

семантически-навигационные системы.

СЕМАНТИЧЕСКИ-НАВИГАЦИОННЫЕ СИСТЕМЫ

В семантически-навигационных системах документы, помещаемые в хранилище, оснащаются специальными навигационными конструкциями, соответствующими смысловым связям (ссылкам) между различными документами или отдельными фрагментами одного документа. Такие конструкции реализуют некоторую *семантическую (смысловую) сеть* в базе документов. Способ и механизм выражения информационных потребностей в подобных системах заключаются в явной навигации (перенаправлении) пользователя по смысловым отсылкам между документами. Такой подход реализуется в гипертекстовых ИПС.

СИСТЕМЫ НА ОСНОВЕ ИНДЕКСИРОВАНИЯ

Процесс отображения документа в поисковое пространство называется индексированием.

В системах на основе индексирования исходные документы помещаются в базу без какого-либо дополнительного преобразования, но при этом смысловое содержание каждого документа отображается в некоторое поисковое пространство.

Иначе говоря, каждый документ описывается набором ключевых слов (словосочетаний) – дескрипторов. Этот набор дескрипторов называется поисковым образом документа (ПОД). Для поиска документа пользователь формирует свой набор дескрипторов, формируя поисковый образ запроса (ПОЗ) к базе документов.

Документальная ИС на основе определенных критериев и способов ищет документы, для которых ПОД совпадают или близки ПОЗ, и выдает соответствующие документы (или их адреса – ссылки на них). Соответствие найденных документов запросу пользователя называется *релевантностью*.

ИНФОРМАЦИОННОЕ ОПОВЕЩЕНИЕ   
В ДОКУМЕНТАЛЬНЫХ ИС

Еше одна функция документальных ИС – задачи информационного оповещения по всем новым поступающим в систему документам, соответствующим заранее определенным информационным потребностям пользователя. Эти потребности отражаются в поисковое пространство в виде так называемых поисковых профилей пользователей. Информационно-поисковая система по мере поступления и индексирования новых документов сравнивает ПОД с поисковыми профилями пользователей и принимает решение о соответствующем оповещении.

ИНФОРМАЦИОННО-ПОИСКОВЫЕ ЯЗЫКИ

Поисковое пространство, отображающее ПОД и реализующее механизмы информационного поиска документов, так же, как и в СУБД фактографических систем, строится на основе языков документальных баз данных, называемых информационно-поисковыми языками (ИПЯ). По аналогии с языками баз данных фактографических систем в ИПЯ выделяют структурную и манипуляционную составляющие (в СУБД – языки описания данных и языки манипулирования данными).

КЛАССИФИКАЦИЯ ИПЯ

Структурная составляющая

в семантически-навигационных ИПС – гипертекстовые технологии;

в ИПС на основе индексирования – индексные указатели:

информационно-поисковые каталоги;

тезаурус;

генеральный указатель.

Поисковая (манипуляционная составляющая)

Дескрипторные языки

Семантические языки

предикатные

реляционные

СТРУКТУРНАЯ СОСТАВЛЯЮЩАЯ ИПЯ

Структурная составляющая ИПЯ семантически-навигационных систем реализуется в виде техники смысловых отсылок в текстах документов и специальном навигационном интерфейсе по ним и в настоящее время представлена гипертекстовыми технологиями.

Структурная составляющая ИПЯ документальных ИПС на основе индексирования реализуется индексными указателями в форме информационно-поисковых каталогов, тезаурусов и генеральных указателей.

ИНФОРМАЦИОННО-ПОИСКОВЫЕ КАТАЛОГИ

Информационно-поисковые каталоги представляют собой *классификационную систему знаний по определенной предметной области*. Смысловое содержание документа в информационно-поисковых каталогах отражается тем или иным *классом каталога*, а *индексирование* документов заключается в *присвоении* каждому документу специального кода (*индекса*) соответствующего по содержанию класса (классов) каталога и создания на этой основе специального индексного указателя (пример – УДК (универсальная десятичная классификация)).

ТЕЗАУРУС

Тезаурус представляет собой специальным образом организованную совокупность основных лексических единиц (понятий) предметной области и описание семантических отношений между лексическими единицами, не зависящими от контекста. Независимость от контекста обозначает обобщенность (абстрагированность) смысловых отношений, например, отношения «род-вид», «предмет-целое», «субъект-объект-средство-место-время\_действия». Так же, как и в информационно-поисковых каталогах, в информационно-поисковое пространство отражается не весь текст документа, а только выраженное средствами тезауруса смысловое содержание документа.

ГЕНЕРАЛЬНЫЙ УКАЗАТЕЛЬ

Генеральный указатель (глобальный словарь-индекс) – представляет собой перечисление всех слов (словоформ), имеющихся в документах хранилища, с указанием (отсылками) координатного местонахождения каждого слова (№ документа - № абзаца - № предложения - № слова). Индексирование нового документа производится через дополнение координатных отсылок тех словоформ генерального указателя, которые присутствуют в новом документе. Так как поисковое пространство в таких системах отражает полностью весь текст документа, то такие системы получили название полнотекстовых ИПС.

ДЕСКРИПТОРНЫЕ И СЕМАНТИЧЕСКИЕ  
ИНФОРМАЦИОННО-ПОИСКОВЫЕ ЯЗЫКИ

Поисковая (манипуляционная) составляющая ИПЯ реализуется дескрипторными и семантическими языками запросов.

В *дескрипторных языках* документы и запросы представляются наборами некоторых лексических единиц (слов, словосочетаний, терминов) – дескрипторов, не имеющих между собой связей (не имеющих грамматики). Таким образом, каждый документ или запрос представлен некоторым набором дескрипторов (ПОД или ПОЗ). Поиск осуществляется через сопоставление этих наборов.

*Семантические языки* содержат грамматические и семантические конструкции для описания смыслового содержания документов и запросов.

Семантические языки делятся на две большие группы:

предикатные языки

реляционные языки

ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ   
ПО В.М. ГЛУШКОВУ

Моделирование - выяснение (воспроизведение) свойств какого-либо объекта, процесса, явления с помощью другого объекта, процесса или явления - его модели

Информационное моделирование любого объекта - это «фиксация того или иного уровня познания этого объекта, позволяющая описывать не только его строение, но и предсказать (с той или иной степенью приближения) его поведение».

ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ:  
ВАРИАНТ КЛАССИФИКАЦИИ ИНФОРМАЦИОННЫХ МОДЕЛЕЙ

Типы информационных моделей:

вербальные

табличные

графические

ОСНОВНЫЕ НАПРАВЛЕНИЯ ИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ

Моделирование данных

Моделирование программ

Моделирование бизнес-процессов

Моделирование знаний?

МОДЕЛИРОВАНИЕ ДАННЫХ

Модели данных:

Инфологические (ER-модели)

Даталогические

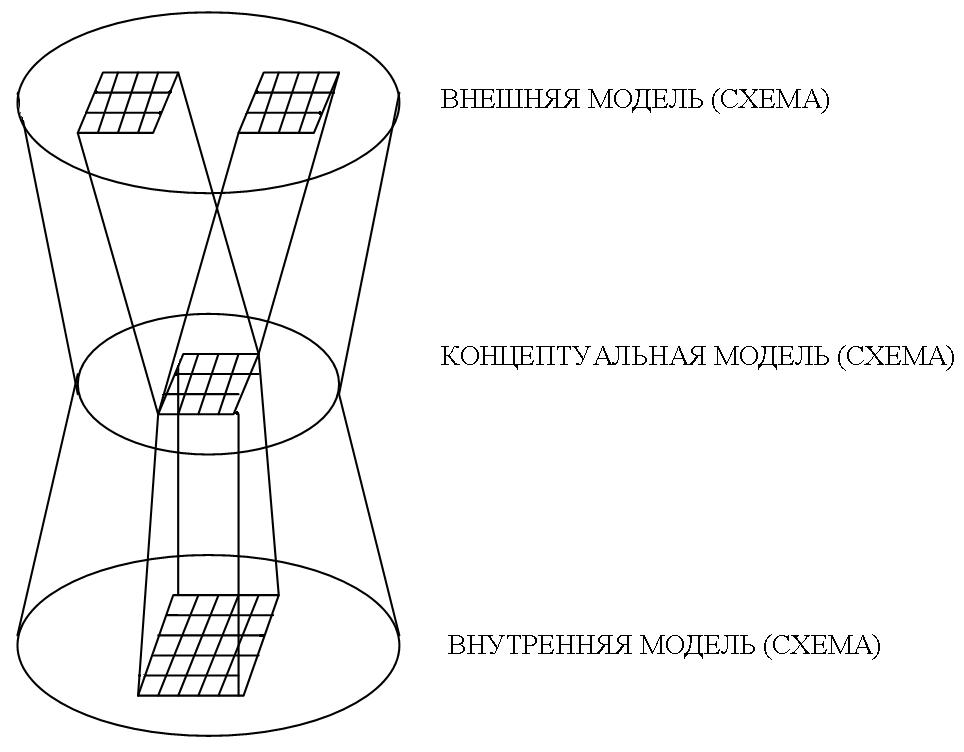
По уровням архитектуры систем баз данных: концептуальные, внутренние, внешние

По типу СУБД:

иерархические, сетевые, реляционные, объектные

ОБОБЩЕННАЯ СХЕМА ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

 ТРЕХУРОВНЕВАЯ ОРГАНИЗАЦИЯ БАЗ ДАННЫХ



ПРИМЕР ИНФОЛОГИЧЕСКОЙ МОДЕЛИ В НОТАЦИИ ЧЕНА



ПЕРЕХОД ОТ ИНФОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ   
К МОДЕЛЯМ ДАННЫХ И ЗНАНИЙ

Для построения модели знаний можно использовать, как базовую, инфологическую модель предметной области.

Представим:

*Инфологическую модель* в виде множества

{ *E, R }*, где E – множество объектов предметной области, R – множество отношений между объектами предметной области;

*Датологическую модель* в виде множества

*{ D, M }*, где D – множество описания данных,

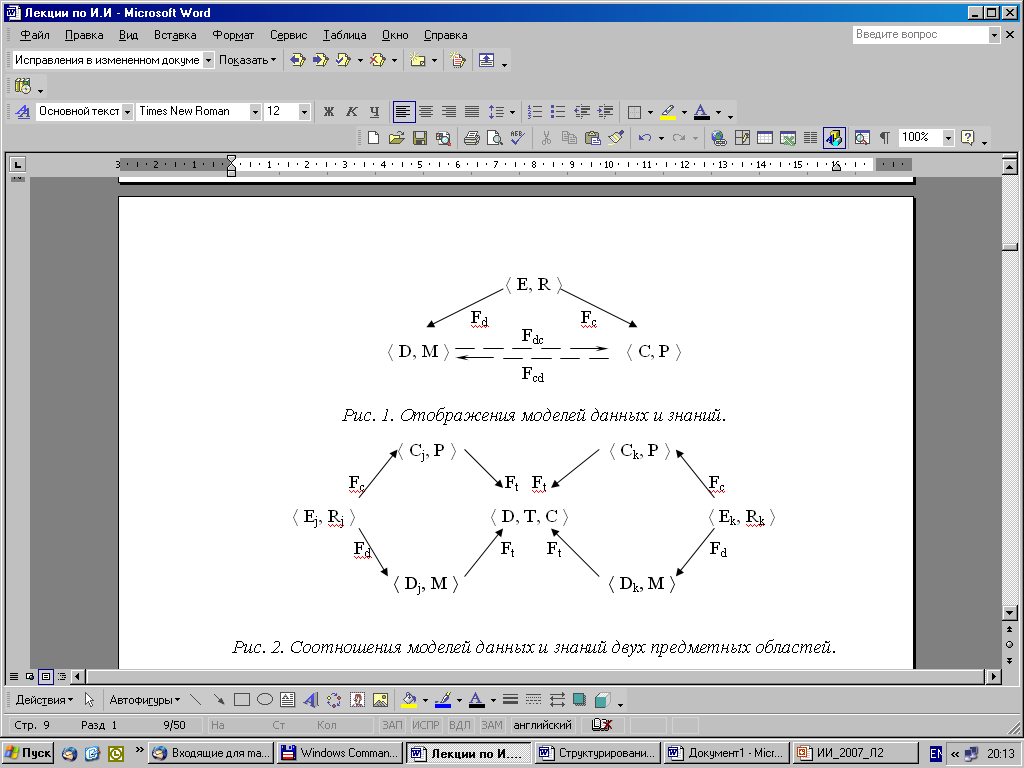
M – множество операторов манипулирования данными;

*Модель знаний* в виде множества *{ C, P },*

где C – описания описаний знаний, P – множество операторов манипулирования знаниями.

Для описания знаний часто используют термин «модель представления знаний»

ПЕРЕХОД ОТ ИНФОЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ   
К МОДЕЛЯМ ДАННЫХ И ЗНАНИЙ



Комментарии к рисункам:

Fd – отображение инфологической модели в модель данных; Fc - отображение инфологической модели в модель знаний; Fdc и Fcd – взаимные отображения моделй данных и знаний; Ft – отображение моделей данных и знаний в транзитную область.

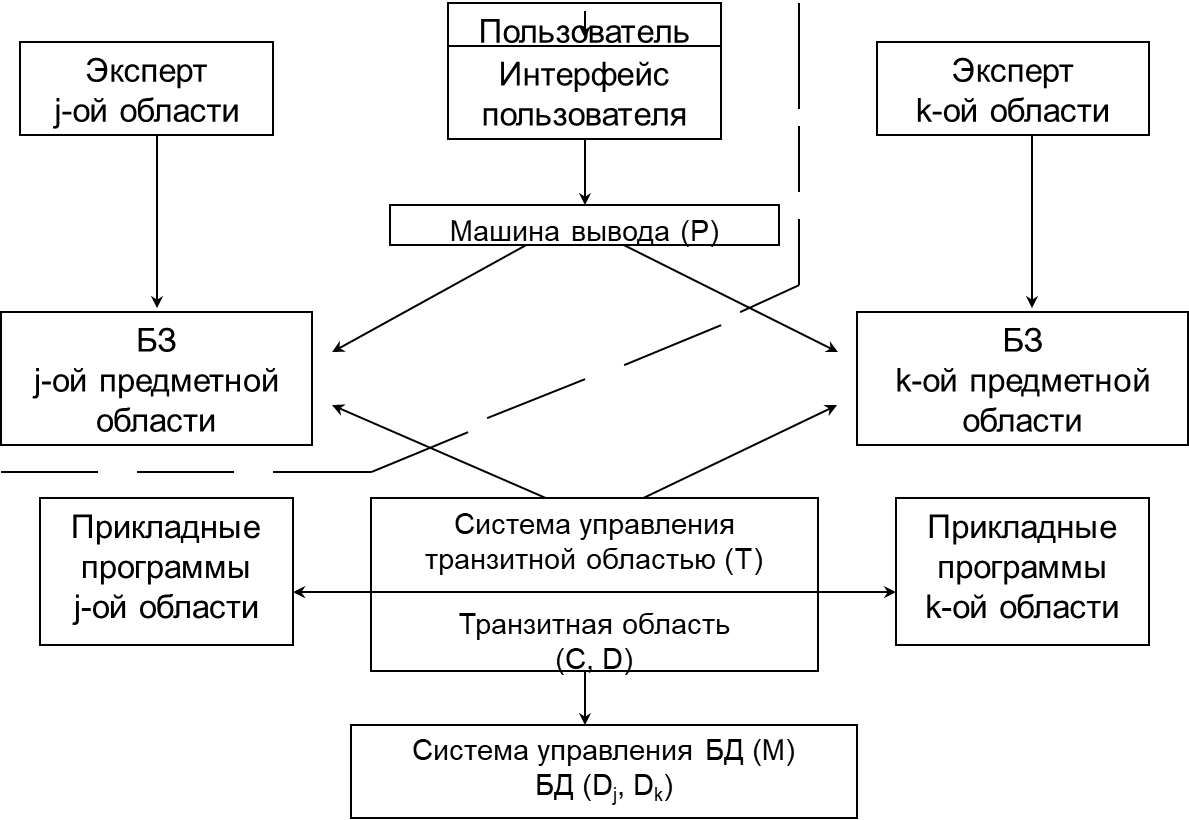
Индекс j относится к описаниям данных, знаний и инфологической модели j-й предметной области; индекс k - к описаниям данных, знаний и инфологической модели k-й предметной области.

{D, T, C} – гибридная модель данных и знаний;

Т – множество операторов преобразования данных и знаний.

Транзитная область – область для временного хранения данных и знаний.

АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА,   
ПОДДЕРЖИВАЮЩЕГО ОТОБРАЖЕНИЯ ДАННЫХ И ЗНАНИЙ   
ДЛЯ ДВУХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ



ГИБРИДНЫЕ ЭКСПЕРТНЫЕ СИСТЕМЫ

В общем случае представленный выше программный комплекс можно назвать гибридной экспертной системой.

Существует два типа гибридных экспертных систем:

использующие разные модели представления знаний;

такие, которые кроме БД и БЗ, включают и прикладные программы конкретной предметной области.

МОДЕЛИРОВАНИЕ ЗНАНИЙ

КЛАССИФИКАЦИЯ МОДЕЛЕЙ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ:

Логические

Логико-лингвистические

Продукционные

Фреймовые

Семантические сети

Онтологии (модели представления декларативных знаний)

Онтологии – базы знаний специального вида, которые могут «читаться» и пониматься, отчуждаться от разработчика и/или физически разделяться их пользователями. Это формально представленные знания на базе концептуализации (описания множества объектов и понятий, знаний о них и связей между ними)

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ:   
ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Логическая модель – это модель представления знаний, в основе которой лежит формальная система (например, исчисление предикатов).

Логико-лингвистическая модель – это модель, основанная на расширении формальной системы, в рамках которого вводятся процедуры изменения всей или части элементов формальной системы в зависимости от решаемых задач.

ЛОГИКА

Логика имеет дело с выявлением обоснованности утверждения, т.е. с методами, позволяющими доказать, можно ли данное заключение вывести, исходя из известных факторов.

Логика бывает:

монотонная

немонотонная

нечеткая

МОНОТОННАЯ, НЕМОНОТОННАЯ И НЕЧЕТКАЯ ЛОГИКА

Монотонная логика – это логика замкнутого мира, некоторая формальная система. Т.е., если на каком-либо шаге вывода получено утверждение, то оно действует и на последующих шагах.

Немонотонная логика – это логика открытого мира. Т.е. утверждение при поступлении в систему новой информации может измениться. Эта логика характерна для интеллектуальных систем, имеющих дело со сложными предметными областями.

Если экспертные системы реализуют монотонную логику, то их называют статическими, если немонотонную - динамическими.

Нечеткая логика – это логика, в которой используются не количественные, а качественные определения. Чаще всего это нечеткие определения лингвистической переменной «частота»: часто, редко, очень редко, никогда, всегда и т.п.

ЛОГИЧЕСКИЙ ВЫВОД И СТРАТЕГИЯ   
УПРАВЛЕНИЯ ВЫВОДОМ

Логический вывод – это последовательность рассуждений, приводящая к следствию с использованием аксиом и правил вывода.

Одной из основных формул логики является формула «modus ponens», которая выражается формулой:

«если А есть В, то С есть D».

Стратегия управления выводом – это совокупность правил, с помощью которых организуется выбор правил вывода в формальных системах или выбор продукции в системе продукций при поиске решения.

Стратегии вывода:

прямой вывод

обратный вывод

ИСЧИСЛЕНИЕ

Исчисление – это формальная система, задаваемая четверкой

(Т, В, А, Р), где

Т – множество базовых символов исчисления;

В – синтаксические правила, с помощью которых из элементов множества Т порождаются производные элементы;

А – множество априорно истинных элементов исчисления (аксиомы исчисления);

Р – множество семантических правил (т.е. правил вывода), с помощью которых из одних элементов системы порождаются другие.

ИСЧИСЛЕНИЕ ВЫСКАЗЫВАНИЙ

Исчисление высказываний – это формальная система, базовыми элементами которой являются высказывания, т.е. нерасчлененные предложения, относительно которых в каждый момент времени можно утверждать, что они либо абсолютно истинны, либо абсолютно ложны.

Исчисление высказываний изучает связи между высказываниями, которые задаются логическими связками (конъюнкция, дизъюнкция).

Исчисление высказываний является аксиоматической системой.

Для классического исчисления высказываний все аксиомы тождественно истинны, а правила вывода не меняют этого свойства.

ПРЕДИКАТ

Пропозициональная функция – это функция, областью значения которой служат высказывания.

Предикат – это пропозициональная функция, которая каждому упорядоченному набору

(a1, a2, … , an) элементов множества А ставит в соответствие некоторое высказывание

p (a1, a2, … , an) и принимает значение

истина (1) или ложь (0).

Если n = 0, то предикат оказывается высказыванием - p. Если n = 1, то предикат соответствует тому, что называется свойством – предикат 1-го порядка. Если n = 2, то это предикат 2-го порядка и т.д.

ИСЧИСЛЕНИЕ ПРЕДИКАТОВ

Неформально предикат определяют как специальный знак в исчислении предикатов, отражающий определенное отношение между конечным множеством сущностей аргументов. В качестве значений предиката на множестве означенных аргументов выступают два: истина и ложь.

Иначе говоря, неформально предикат определяют как логическую функцию, принимающую значения истина (1) или ложь (0).

Исчисление предикатов – это исчисление, где наряду с формулами исчисления высказываний используются формулы, в которые могут входить отношения (предикаты), связывающие между собой группы элементов исчисления и кванторы общности и существования.

КВАНТОРЫ

Квантор общности  – специальный указатель на то, что некоторое утверждение p, содержащее переменные, распространяется на все формулы, получаемые при подстановке вместо переменных, перечисленных в указателе, любых значений из области определения этих переменных.

Квантор существования  – специальный указатель на то, что некоторое утверждение p имеет место (истинно), при некоторых переменных, перечисленных в данном указателе, причем конкретные значения не указываются, а фиксируется лишь то, что они существуют. Переменные, перечисленные в указателе, называют связанными.

ПРИМЕРЫ ИСЧИСЛЕНИЯ ВЫСКАЗЫВАНИЙ И ПРЕДИКАТОВ

Является (Смит, специалист по ЭВМ)

Является (Смит, оптимист)

Является (Х, специалист по ЭВМ)

Написал (Смит, программа)  работает (программа)  отладить (Смит, программа, вечер)  передать (программа, программист, следующий день). Это означает, что если Смит написал программу и она не работает, то ему следует отладить программу или передать программисту на следующий день.

 (x) (ИТ-специалист (x)  программист (x)). Т.е. все ИТ-специалисты являются программистами.

 (x) (ИТ-специалист (x)  оптимист). Т.е. некоторые ИТ-специалисты являются оптимистами.

ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ ПРЕДИКАТОВ   
ДЛЯ ОПИСАНИЯ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА   
В ИССЛЕДОВАНИЯХ ЭНЕРГЕТИКИ

ТЭК – топливно-энергетический комплекс

ТЭК включает отраслевые системы:

ЭЭС – электроэнергетическая система

ГСС – газоснабжающая система

НСС – нефтеснабжающая система

ТСС – теплоснабжающая система

Для исследований ТЭК в целом необходимо выполнить сначала исследования отраслевых систем энергетики

ПРЕДИКАТ, ОПИСЫВАЮЩИЙ СХЕМУ КОМПЛЕКСНОГО ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

Т (Р (D)), где

Т – комплексный вычислительный эксперимент,

Р – программный комплекс для исследования ТЭК,

D – исходные данные для Р. Детальнее:

Т (Р (S1 (P1 (D1))) & (S2 (P2 (D2))) & … & (Sn (Pn (Dn)))),

где Si – i-ая система энергетики, Рi – программный комплекс для ее исследования, Di – исходные данные для Рi.

СМЫСЛ ПРЕДИКАТОВ: Т (Р (D))   
Т (Р (S1 (P1 (D1))) & (S2 (P2 (D2))) & … & (Sn (Pn (Dn)))),

Результаты комплексного вычислительного эксперимента будут получены, если

Есть данные об отраслевых системах энергетики, т.е. Di – «истина»;

Разработаны программные комплексы для исследования систем энергетики, т.е. Рi – «истина»;

Проведены вычислительные эксперименты для исследования систем энергетики, т.е. Si – «истина», в результате которых получены исходные данные для исследования ТЭК, т.е. D – «истина»;

Имеется программный комплекс исследований ТЭК; Р – «истина».

Проведен вычислительный эксперимент для исследования ТЭК, т.е. Т – «истина».

ЛОГИКО-ЛИНГВИСТИЧЕСКИЕ МОДЕЛИ

Продукционная модель – это модель представления знаний, в основе которой лежит продукция, т.е. правила типа «если … , то … ».

Семантическая сеть (сетевая модель знаний) – это модель представления знаний, в основе которой лежат семантические сети, в вершинах которых лежат информационные единицы, а дуги характеризуют отношения и связи между ними.

Фреймовая модель – это модель представления знаний, в основе которой лежат фреймы. Фрейм состоит из конечного числа слотов (или составных ячеек), каждый из которых имеет имя и значение. Последнее может быть ссылкой на другие слоты или фреймы.

ПРОДУКЦИОННЫЕ ЭКСПЕРТНЫЕ СИСТЕМЫ

*Продукционные системы состоят из трех компонентов:*

БЗ, содержащая правила продукции

БД, которая отображает текущее состояние некоторой задачи (содержит факты, используемые в правилах-продукциях)

Управляющая структура, решающая, какое из правил продукции требуется применить первым

ПРИМЕР: ПРАВИЛА ДЛЯ УСТАНОВЛЕНИЯ   
ПРОДОЛЖИТЕЛЬНОСТИ ОТПУСКА

ЕСЛИ «служащий работает в управлении» ТО «он старший управляющий или управляющий или клерк»

ЕСЛИ «служащий – старший управляющий и дипломированный специалист» ТО «продолжительность отпуска 8 недель»

ЕСЛИ ««служащий – старший управляющий, но не имеет диплома» ТО «продолжительность отпуска 6 недель»

ЕСЛИ «служащий – управляющий и дипломированный специалист» ТО «продолжительность отпуска 5 недель»

ЕСЛИ «служащий – управляющий, но недипломированный специалист» ТО «продолжительность отпуска 4 недели»

ЕСЛИ «служащий – клерк, но имеет стаж работы»

ТО «продолжительность отпуска 3 недели»

ЕСЛИ «служащий – клерк, и не имеет стажа работы»

ТО «продолжительность отпуска 2 недели»

ГРАФИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ПРИМЕРА: ПРАВИЛА ДЛЯ УСТАНОВЛЕНИЯ ПРОДОЛЖИТЕЛЬНОСТИ ОТПУСКА



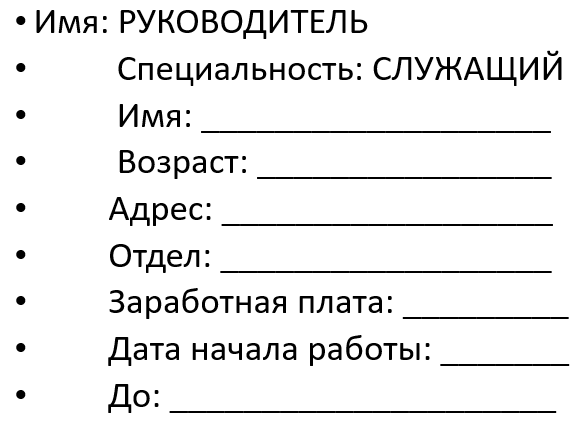
СЕМАНТИЧЕСКАЯ СЕТЬ

Базовым функциональным элементом семантической сети служит структура из двух компонентов: узлов и связывающих их дуг. Каждый узел представляет собой некоторое понятие, а дуга – отношения между ними.

Наибольшее сходство с семантическими сетями имеют модели Чена, или ER-модели (модель «сущность-связь»)

ФРЕЙМОВЫЕ МОДЕЛИ

Скелетный фрейм для понятия «руководитель»



ФРЕЙМОВЫЕ МОДЕЛИ

*Конкретизация фрейма для общего понятия «руководитель»*

Имя: РУКОВОДИТЕЛЬ

Специальность: СЛУЖАЩИЙ

Имя: агрегат (Ф.И.О.)

Возраст: агрегат (годы)

Адрес: АДРЕС

Отдел: диапазон (производство, администрация)

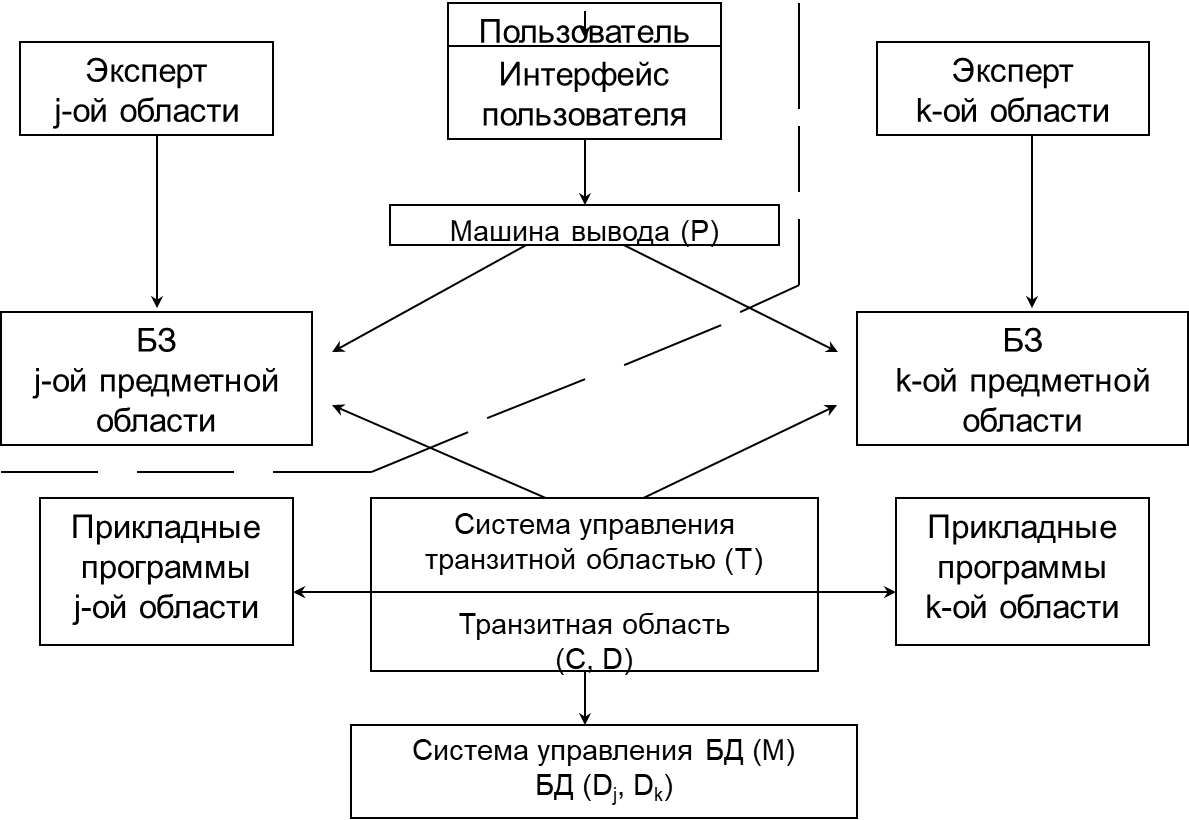
Дата начала работы: агрегат (месяц, год)

До: агрегат (по умолчанию – текущая дата)

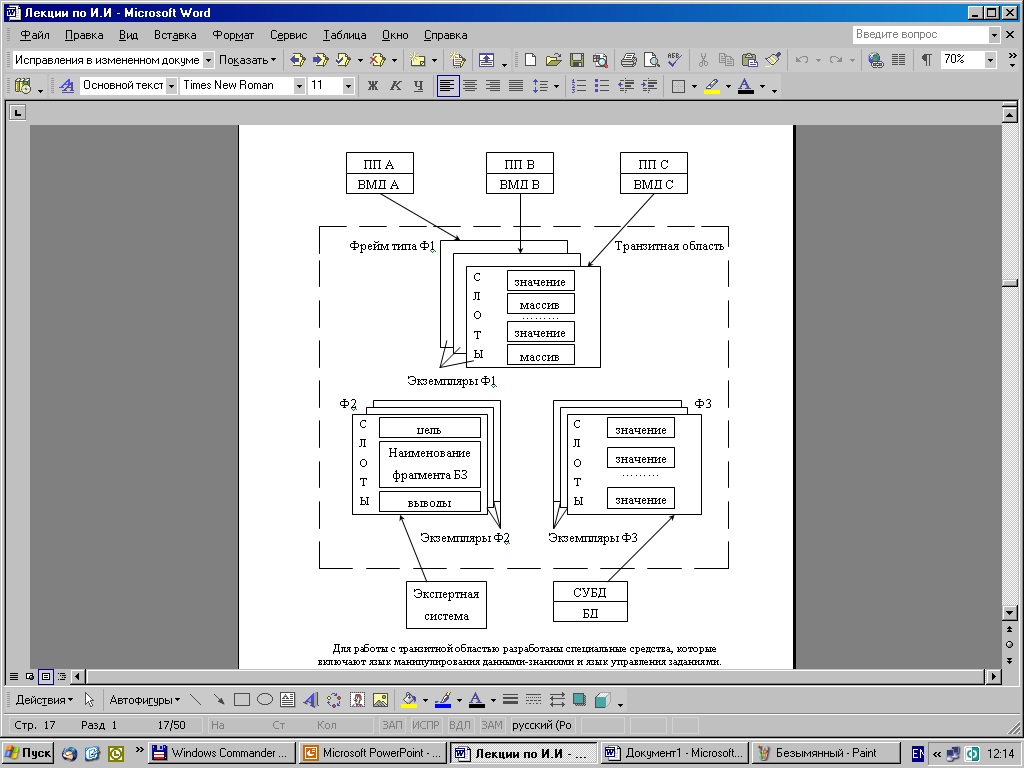
ФРЕЙМ (с англ. «каркас», «рамка»)

Фрейм является наиболее сложной структурой, позволяющей широко использовать вложенность составляющих его структур. Фреймовые структуры могут связывать правила (продукции), данные, описывающие состояние объектов и вычислительные процедуры. В зависимости от выбора стратегии реализации могут быть определены разные типы и разная степень вложенности фреймов.

АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА,   
ПОДДЕРЖИВАЮЩЕГО ОТОБРАЖЕНИЯ ДАННЫХ И ЗНАНИЙ   
ДЛЯ ДВУХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ



ПРИМЕР ИСПОЛЬЗОВАНИЯ ФРЕЙМОВОЙ МОДЕЛИ ДЛЯ РЕАЛИЗАЦИИ ТРАНЗИТНОЙ ОБЛАСТИ



КОММЕНТАРИИ К СХЕМЕ

Фреймы типа Ф1 предназначены для представления внешних моделей данных прикладных программ;

Фреймы типа Ф2 – для представления фрагментов баз знаний экспертных систем;

Фреймы типа Ф3 раскрывают значения слотов в экземплярах Ф1 и Ф2

Для работы с транзитной областью были разработаны специальные средства, которые включают язык манипулирования данными-знаниями и язык управления заданиями.

Макрооператоры языка манипулирования данными-знаниями выполняют необходимый минимум функций: ввод, удаление, копирование и редактирование фреймов

Язык управления заданиями позволяет описать вычислительную цепочку: перечислить этапы вычислительного эксперимента, каждый из которых может быть описан во фрейме типа Ф1

Используя в качестве предикатов обозначения вложенных фреймов, которые, в свою очередь, могут быть фреймами продукций, процедур, данных, можно с их помощью описывать стратегии вывода в интеллектуальной системе.

УПРАВЛЕНИЕ ЗНАНИЯМИ

Понимание растущей роли человеческих ресурсов (human resource), организационного управления (organizational

behaviour) и информационных технологий (information technologies) в повышении эффективности современных

компаний явилось основной причиной того, что со второй половины 90-х годов специалисты различных научных направлений стали активно исследовать и обсуждать

проблематику управления знаниями (knowledge management).

В настоящее время знания, интеллектуальный капитал, нематериальные активы, интеллектуальная собственность рассматриваются как новый источник богатства, как важный фактор приобретения конкурентных преимуществ.

ДВА ПОДХОДА К УПРАВЛЕНИЮ ЗНАНИЯМИ

Классический, когда СУЗ строится на основе комбинирования существующих,

уже зарекомендовавших себя технологий для поддержки различных подпроцессов работы со знанием. Речь идет о стандартных и широко используемых IT-технологиях, таких, как E-mail, доски объявлений,

дискуссионные форумы, общие каталоги документов, порталы, метаданные, а

также о специфических технологиях, тяготеющих к инструментарию искусственного интеллекта, таких, как автоматическая классификация,

автоматическое аннотирование документов, распознавание образов и речи и

т.п.

Семантический - основан на использовании взаимосвязанного набора методов и технологий по работе со смыслом, семантикой данных, информацией и

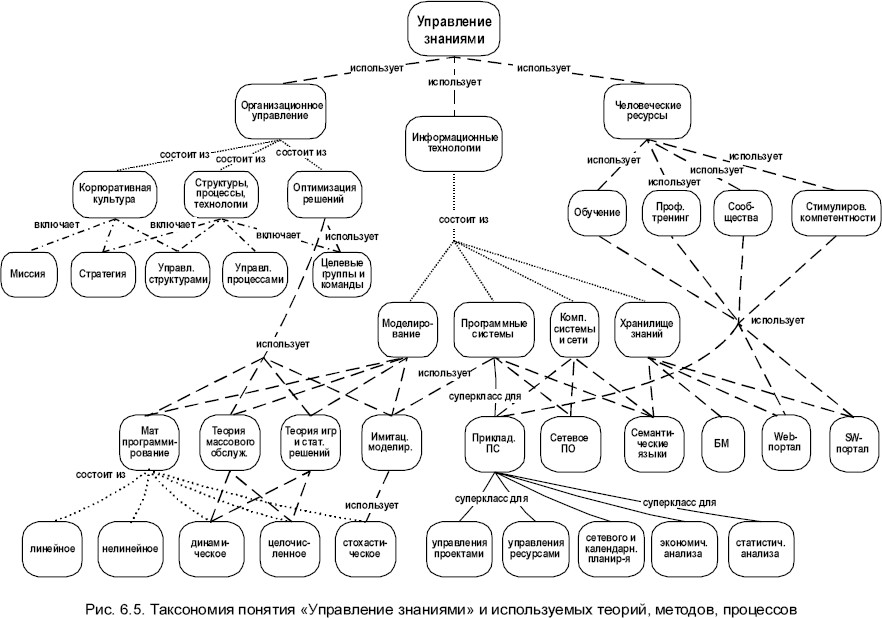
знаниями. В их числе онтологии предметных областей, технологии их

построения и сопровождения, семантические метаданные, семантический

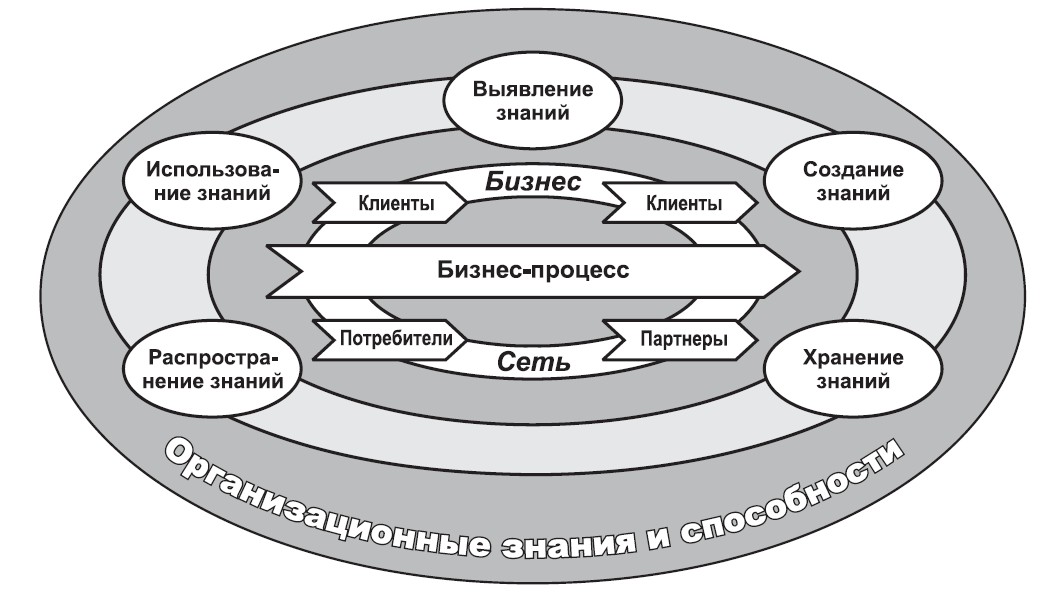
поиск, системы логического вывода, семантическое профилирование знаний экспертов, семантические порталы и сети и т.п., и все это с соответствующей

технологической поддержкой в части языков описания, моделей, программных

инструментов и систем.



ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА ЗНАНИЙ



ОНТОЛОГИЧЕСКИЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

(второй подход

к управлению знаниями)

Основные определения

Определение термина «онтология» зависит от контекста и целей его использования

Онтоло́гия (от греч. οντοσ — сущее, то, что существует и λογος —

учение, наука) — раздел философии, изучающий проблемы бытия, наука

о бытии

Онтология – это формальная спецификация концептуализации, которая имеет место в некотором контексте предметной области (Gruber, 1993).

Концептуализация представляет собой описание понятий, а также всю информацию, имеющую отношение к понятиям (свойства, отношения, ограничения, аксиомы, утверждения), необходимую для описания и решения задач в избранной предметной области.

Концептуальная модель - система концептов и отношений предметной области.

Концепт - понятие, отражающее некоторый конкретный или абстрактный объект реального мира

Формально онтология состоит из понятий (терминов, организованных в таксономию), их описаний и правил вывода.

Таксономия (от греч. taxis - расположение, строй, порядок и nómos - закон), теория классификации

и систематизации сложноорганизованных областей действительности

Определение онтологии

соответствующее спецификации FIPA (Foundation for Intelligent Physical Agents)

Онтология – это подробная *спецификация* структуры определенной *проблемной области*;

Онтология включает в себя *словарь* (т.е. список логических констант и предикатных символов) для описания предметной области и *набор* логических

высказываний, формулирующих существующие в данной

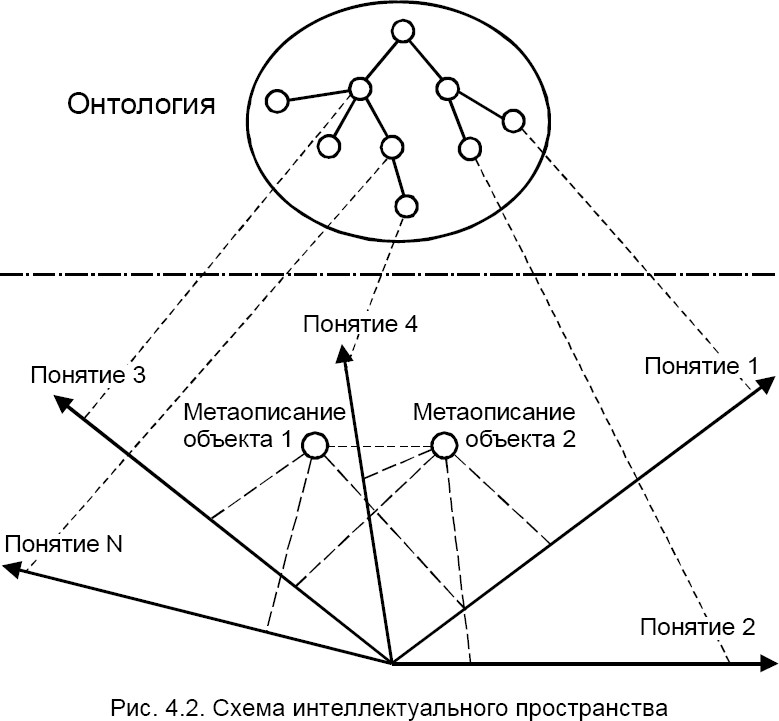
*проблемной области ограничения* и определяющих

интерпретацию словаря;

Онтология предлагает словарь для представления и обмена *знаниями* по интересующей проблеме и *набор связей и свойств*, которые определены между

имеющимися в ее словаре неделимыми *сущностями*.

ОТОБРАЖЕНИЕ ПОНЯТИЙ ПРЕДМЕТНОЙ ОБЛАСТИ В ПОНЯТИЯ ОНТОЛОГИИ



АСПЕКТЫ ИНТЕРПРЕТАЦИИ ТЕРМИНА “ОНТОЛОГИЯ”

(ГАВРИЛОВА Т.А., ХОРОШЕВСКИЙ В.Ф.)

Философское понятие (какие свойства являются общими для всего

сущего?)

Неформальная концептуальная система (базис для

определенной БЗ)

Формальный взгляд на семантику (формализация семантических

структур)

Спецификация концептуализации (в рамках ИИ-сообщества)

Представление концептуальной системы через логическую теорию (характеризуемую формальными свойствами или только ее назначением)

Словарь (используемый логической теорией)

Спецификация логической теории (метауровневая)

Онтология (в информатике) —

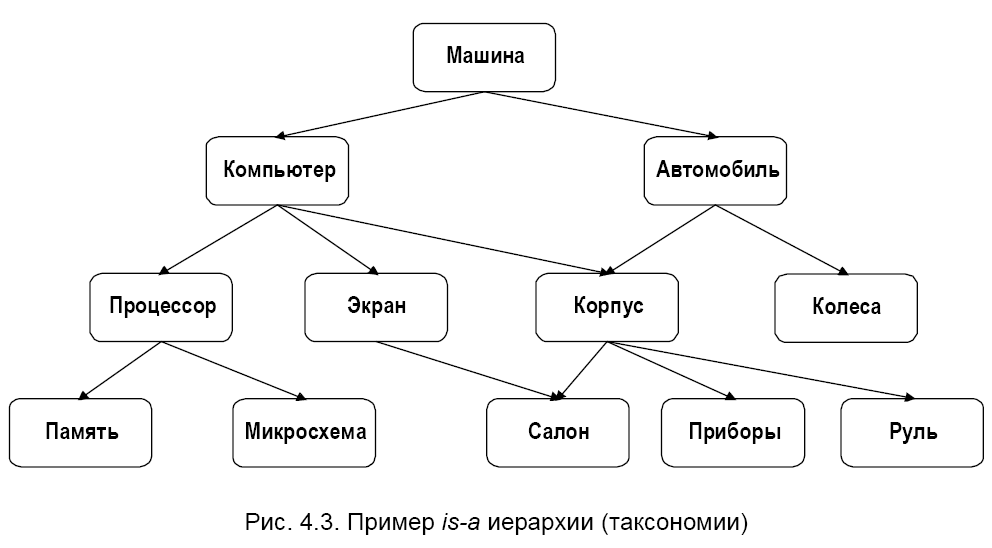
это попытка всеобъемлющей и детальной формализации некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из иерархической структуры данных, содержащей все релевантные классы объектов, их связи и правила (ограничения), принятые в этой области

Онтологии – *базы знаний специального вида,*

*которые могут «читаться» и пониматься,*

*отчуждаться от разработчика и/или физически разделяться их пользователями. Это формально представленные знания на базе концептуализации (описания множества объектов и понятий, знаний о них и связей*

*между ними) (Гаврилова, Хорошевский)*



ПРИМЕР ОНТОЛОГИИ

МОДЕЛЬ ОНТОЛОГИИ

Формально онтология определяется как

O = <X,R,F>, где

X - конечное множество понятий (концептов)

предметной области,

R - конечное множество отношений между понятиями,

F - конечное множество функций интерпретации, заданных на концептах и/или отношениях.

При R=0 и F=0 онтология трансформируется в простой словарь. Пример – индексы поисковых машин

МОДЕЛЬ ОНТОЛОГИЧЕСКОЙ СИСТЕМЫ

Формально определяется как:

Z=<O, P, M> , где

O – онтология верхнего уровня (метаонтология)

(содержит общие понятия и отношения, не зависящие от предметной

области - «объект», «свойство», «значение» и т.п.);

P – множество предметных онтологий и онтологий задач предметной области (с учетом предпочтений пользователя);

M – модель машины вывода данной онтологической

системы (например, для изменения критериев релевантности поиска

или критериев формирования репозитория).

Модель онтологической системы позволяет описывать взаимосвязь онтологий разных уровней

КЛАССИФИКАЦИЯ ОНТОЛОГИЙ

метаонтология – содержит общие понятия и отношения, не зависящие от предметной

области («объект», «свойство», «значение» и т.п.);

предметная онтология – содержит понятия, описывающие конкретную предметную область и отношения, семантически значимые для данной предметной области;

онтология задач (и методов) – содержит в качестве понятий типы решаемых задач, а отношения специфицируют декомпозицию задач на подзадачи;

прикладная онтология (онтология приложения) – описывает концепты, зависящие как от

предметной области, так и от задач

сетевая онтология – используется для описания конечных результатов действий,

выполняемых объектами предметной области или задачи.

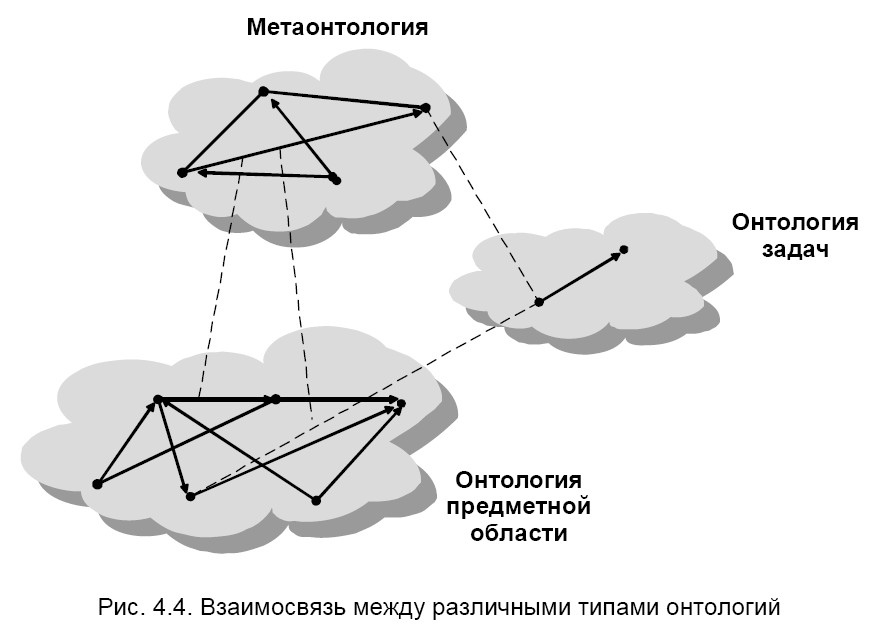
Другие классификации:

по уровню детализации

по “природе” предметных областей

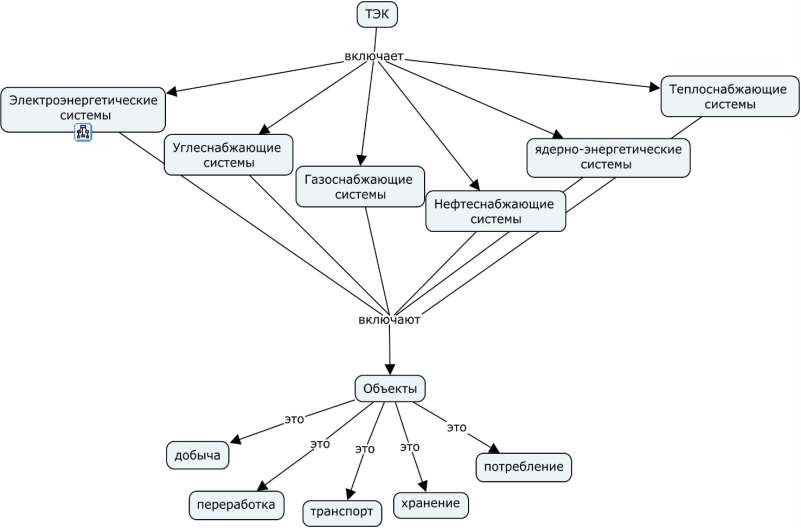
по степени разработки и сопровождения

ВЗАИМОСВЯЗЬ МЕЖДУ РАЗЛИЧНЫМИ ТИПАМИ ОНТОЛОГИЙ

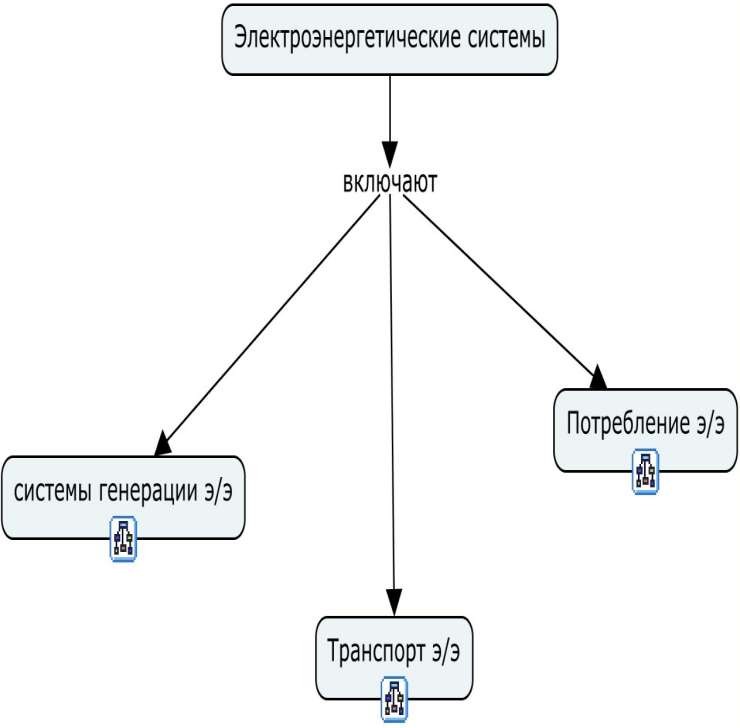


ПРИМЕР: Переход от метаонтологии топливно-

энергетического комплекса (ТЭК) к метаонтологии ЭЭС

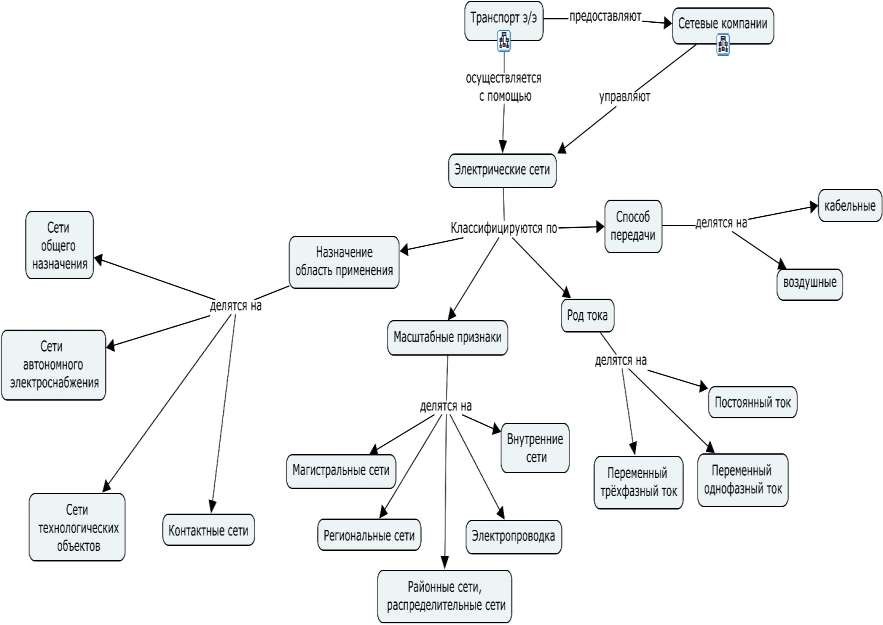


ПРИМЕР: Переход от метаонтологии электро-энергетических систем к онтологии транспорта э/э



ПРИМЕР: Онтология транспорта электроэнергии

УРОВНИ ФОРМАЛИЗАЦИИ ОНТОЛОГИИ



Неформализованный. Слабо описанная на естественном языке онтология

Неформально-структурированный. Онтология

описана ограниченным набором лексики естественного языка в

структурированном виде

Полуформальный. Онтология описана на специальном

формально определенном языке

Строго формальный. Для описания используются тщательно определенные термины с формальной семантикой, теоремы и доказательства для таких свойств, как надежность и

полнота онтологии

СПЕКТР ВЫРАЗИТЕЛЬНОСТИ ОНТОЛОГИЙ

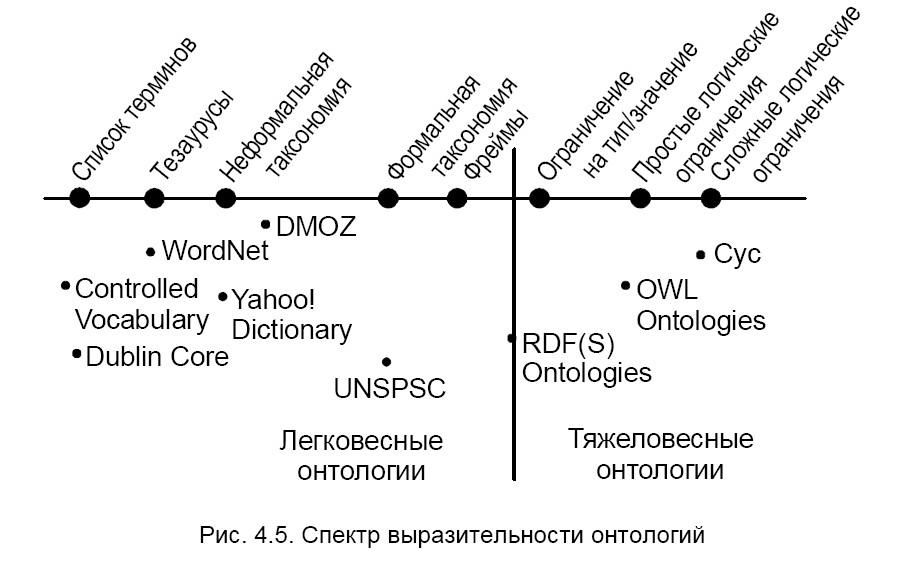
ЯЗЫКИ ПРЕДСТАВЛЕНИЯ ОНТОЛОГИЙ

традиционные языки спецификации онтологий (LOOM, OKBC, OCML, Flogic, LBase)

специальные языки спецификации онтологий (Ontolingua, CycL, SHOE)

языки основанные на Web-стандартах (UPML, DAML, OIL, XODL, XML, RDF, RDFS, OWL)

Выбор языка зависит от целей разработки онтологии



ЛОГИЧЕСКАЯ ЗАВИСИМОСТЬ ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ОНТОЛОГИЙ

OWL (WEB ONTOLOGY LANGUAGE) – ЯЗЫК WEB-ОНТОЛОГИЙ

Основное назначение OWL – для использования

приложениями, которые должны не только представлять информацию человеку, но и обрабатывать ее.

OWL спроектирован для хранения и отображения большого количества разнородной информации и знаний, а также для поддержки семантики и связей между этими знаниями.

OWL предназначен для описания классов и отношений между ними, которые присущи для Web-документов и приложений

ЖИЗНЕННЫЙ ЦИКЛ ОНТОЛОГИЙ

Планирование (задачи, ресурсы, сроки)

Разработка

Спецификация – определяет цели создания, предполагаемое использование и потенциальных пользователей

Концептуализация – обеспечивает структурирование предметных знаний

Формализация – трансформирует концептуальную модель в формальную или вычислительную

Реализация – вычислительная модель программируется на соответствующем языке

представления знаний

Проверка - включает в себя идентификацию и редактирование неопределенных терминов; выявление конфликтов и несогласованностей; проверку онтологии на

полноту; идентификацию семантических различий между двумя терминами в различных онтологиях; синтаксический анализ; анализ таксономии; семантическую оценку

Поддержка (одновременно с разработкой) – приобретение знаний, оценка онтологии,

интеграция при построении новой онтологии с использованием существующих

ЖИЗНЕННЫЙ ЦИКЛ ОНТОЛОГИИ



ОСНОВНЫЕ ЭТАПЫ РАЗРАБОТКИ ОНТОЛОГИЙ

определение классов в онтологии (понятий в рассматриваемой предметной области);

определение слотов (свойств и атрибутов каждого понятия);

описание допускаемых значений этих слотов и ограничений, наложенных на слоты.

отображение иерархии классов (подкласс – надкласс);

определение других отношений между классами.

КРИТЕРИИ РАЗРАБОТКИ ОНТОЛОГИЙ

(Gruber)

Ясность- онтология должна эффективно описывать смысл определяемых терминов.

Согласованность- онтология должна быть согласованной, определения и

выводимые утверждения должны быть непротиворечивыми.

Расширяемость- онтология должна предусматривать возможность

определения новых терминов без необходимости пересмотра

существующих определений.

Минимум влияния кодирования- концептуализация должна быть

специфицирована на уровне знаний вне зависимости от конкретного

способа кодирования.

Минимальное онтологическое соглашение- для онтологии должен быть достигнут минимальный объем словаря, достаточный для описания знаний, которые предполагается совместно использовать.

СРЕДСТВА ПОСТРОЕНИЯ ОНТОЛОГИЙ

Свободно доступные специализированные программные

средства:

системы Protégé, CMapTools, FaCТ.

The Protégé Ontology Editor and Knowledge Acquisition System

– система Protégé

[http://protégé.stanford.edu](http://protégé.stanford.edu/)

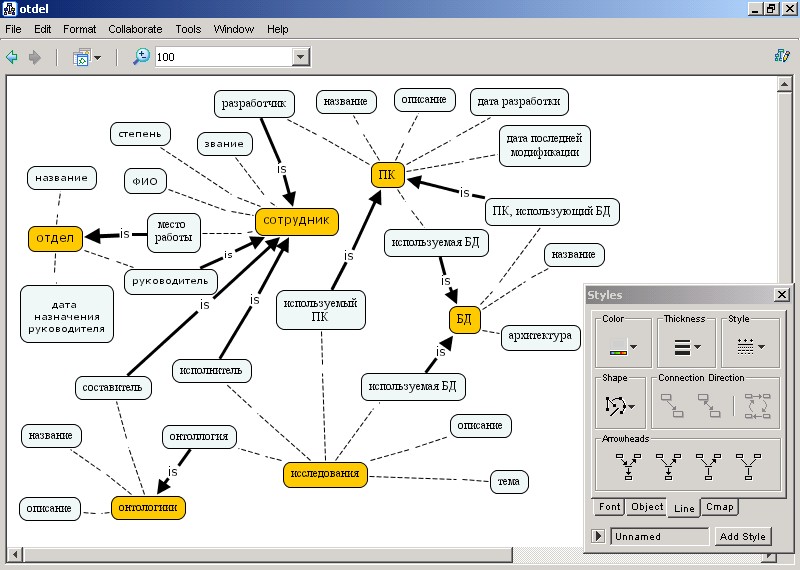
<http://prot%C3%A9g%C3%A9.stanford.edu/> - IHMC CmapTools – система CmapTools [http://cmap.ihmc.us <http://cmap.ihmc.us/>](http://cmap.ihmc.us/)

The FaCT System, [http://www.cs.man.ac.uk/~horrocks/FaCT/](http://www.cs.man.ac.uk/%7Ehorrocks/FaCT/) - система FaCT (поддерживает основные операции

логического вывода)

<http://www.daml.org/tools/>- обзор инструментальных средств ( в т.ч. система FaCT) 86

ПРИМЕР ГРАФИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ОНТОЛОГИИ В СИСТЕМЕ CmapTOOLS



ОСНОВНЫЕ НАПРАВЛЕНИЯ ПРИМЕНЕНИЯ ОНТОЛОГИЙ

Формализованное представление знаний

организация общей и предметной терминологии

описание предметных областей для совместного использования

человеком и компьютером

индексирование, фильтрация и классификация формализованных знаний

приобретение, структурирование знаний и формирование новых

знаний

Семантическая интеграция информационных ресурсов (Формирование единого сетевого пространства знаний )

Обеспечение возможности поиска нужной

информации (индексирование, фильтрация и классификация

формализованных знаний )

ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ОНТОЛОГИЙ

Порталы

Системы автоматизированного сбора информации

Управление корпоративными информационными ресурсами

Обучающие системы

Мультимедийные энциклопедии и коллекции

Взаимодействие электронных бытовых устройств

КОЛЛЕКЦИИ ОНТОЛОГИЙ

(KA)2 (Knowledge Annotation Initiative of the Knowledge Acquisition Community) – Аннотация знаний сообществом приобретения знаний (разрабатываются онтологии организаций, проекта, личности, направления исследований, публикаций, событий, исследовательских продуктов, тсследовательских групп) Инструментарий – Ontobroker

SHOE (Simple HTML Ontology Extensions)) – осуществляет аннотацию Web-

страниц семантической информацией

CYC – ориентация на создание мультиагентной базы знаний всех общих

понятий, включая семантическую структуру терминов и связей между ними

TOVE (Toronto Virtual Enterprise Project)- предметно ориентирована на

представление модели корпорации

Plinius – предназначена для полуавтоматического извлечения знаний в области химии

НАПРАВЛЕНИЯ РАЗВИТИЯ ОНТОЛОГИЙ

Совершенствование способов доступа к знаниям

Развитие средств разработки онтологий

Совершенствование методов автоматического вывода на онтологиях

Развитие средств представления

Развитие средств взаимодействия онтологий

Под эволюцией понимаются медленные, постепенные

количественные и качественные изменения объекта.

При этом каждое новое состояние объекта, как правило, имеет по сравнению с предыдущим более высокий уровень развития и организации.

Эволюция приводит к формированию адаптации (приспособлений) организмов к условиям их

существования.

В науке под адаптацией понимают процесс накопления и использования информации в системе, направленный на достижение ее (системы) оптимального состояния, при

первоначальной неопределенности и изменяющихся

внешних условиях.

В связи с развитием теории эволюции эти идеи все более используются при моделировании мышления и поведения человека, создании систем принятия решений в

неопределенных и нечетких условиях, решении оптимизационных задач большой размерности.

Для описания концепций популяционной генетики Ф. Хедрик

предлагает использовать различные модели. Они

представляют собой вербальное, графическое или математическое описание событий. Преимущество моделей перед простым описанием состоит в том, что они охватывают как сам процесс, так и его составные части.

Из многочисленных теорий эволюционного развития авторы планируют использовать и рассматривать шесть моделей (видов) эволюции.

МОДЕЛЬ ЭВОЛЮЦИИ ДАРВИНА



Это условная структура, реализующая процесс, посредством

которого особи некоторой популяции, имеющие более высокое функциональное значение, получают большую

возможность для воспроизведения потомков, чем «слабые» особи. Такой механизм часто называют *методом «выживания сильнейших».*

Эволюция по Дарвину состоит из следующих положений:

в природе все подвержено неопределенной

наследственной изменчивости, производится потомство,

отличающееся по многим признакам;

все организмы в природе размножаются в геометрической

прогрессии, но численность всех организмов в среднем остается более или менее постоянной, она колеблется около средней величины;

основой отбора является метод «выживания сильнейших».

*Популяция.* Пусть существует популяция особей (например, табун лошадей) на некоторой территории обитания (степь). Табун (попу ляция) насчитывает 50 взрослых лошадей (особей).

*Наследственность.* Каждый год в табуне (популяции) рождаются

жеребята (новое поколение, потомство).

*Изменчивость.* Все родившиеся жеребята (потомки) отличаются друг от друга и от родителей цветом, размерами, ростом, физиче скими данными (скорость, выносливость и т.д.).

*Отбор.* Наибольшие шансы выжить и закрепиться в табуне (по

пуляции) имеют физически более мощные жеребята (способность к

самозащите, возможность победить соперников, способность к

воспроизводству), обладающие большей скоростью (возможность спастись от хищников), выносливостью (способность преодолевать большие расстояния в поисках пищи). В то же время в ходе эволюции могут появляться особи с так

называемыми «полезными» отклонениями от доминирующего в табуне

(популяции) «стандарта» (генотипа), которые позволяют их носителям успешнее выживать в существующих условиях.

*Эволюционная смена форм.* Именно такие особи, как правило, выживают в результате схемы эволюции Дарвина и они постепенно, поколение за

поколением становятся преобладающим (доминирующим) видом в данной популяции.

МОДЕЛЬ ЭВОЛЮЦИИ ЭЙГЕНА И ШУСТЕРА ( *МОДЕЛЬ ГИПЕРЦИКЛОВ)*

МОДЕЛЬ ЭВОЛЮЦИИ ЭЙГЕНА И ШУСТЕРА



Моделирует условную стадию эволюции. М. Эйген на основе гиперциклов описал добиологическую фазу эволюции, в ходе которой происходят процессы отбора, выражающие свойства вещества в особых системах реакций. Они известны как каталитические циклы.

М. Эйген отмечал, что в далеких от равновесия биохимических

системах каталитические реакции объединяются, формируя

сложные сети, в которых могут содержаться и замкнутые циклы.

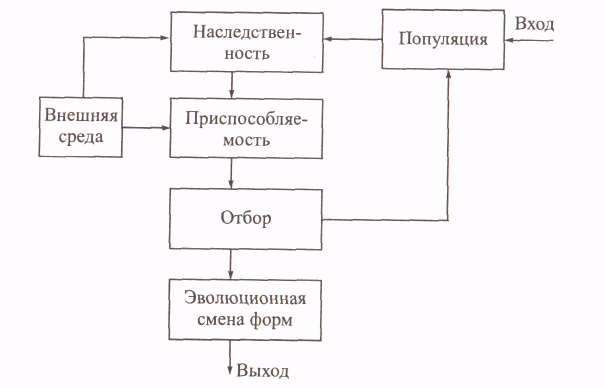
М. Эйген установил, что в условиях достаточного времени и

непрерывного потока энергии каталитические циклы сцепляются, образуя замкнутые циклы, в которых ферменты, созданные в одном цикле, являются катализатором в последующем цикле.

Он назвал гиперциклами те петли, в которых каждый узел

представляет собой каталитический цикл.

МОДЕЛЬ ЭВОЛЮЦИИ ЛАМАРКА



Основана на предположении, что характеристики, приобретенные

особью (организмом) в течение жизни, наследуются его потомками.

Эти изменения, как утверждал Ж. Ламарк, вызываются прямым влиянием внешней среды, упражнением органов и наследованием приобретенных при жизни признаков.

Он объясняет одну из особенностей эволюции органического мира приспособляемостью. Прогрессивную эволюцию, появление форм, более сложных и совершенных, он объяснял «законом градаций» — стремлением живых существ усложнять свою структуру.

Согласно Ж. Ламарку виды эволюционируют, приспособляясь и усложняясь, потому что у них существуют свойства —

приспосабливаться и усложняться.

Причины направленных изменений объясняются различно, но их можно свести к двум:

направленное влияние внешней среды;

способность самого организма.

Рассмотрим схему эволюции на уже знакомом нам сообществе особей (табун

лошадей) из предыдущего примера.

1-2. *Популяция + Наследственность. Как.* видно из схемы, первые два пункта у нас остались неизменными. Так же существует табун лошадей на некоторой территории обитания. Каждый год в табуне появляется потомство.

*3. Внешняя среда.* Происходит воздействие внешних факторов в ви де сильной засухи в степях, где обитает наш табун. Трава в степи выгорела от засухи и табун в поисках пищи вынужден мигриро вать на соседние территории, являющиеся предгорьями. На этих территориях более мягкий климат и трава все еще сохранилась.

*4. Приспособляемость.* В условиях изменившейся среды обитания

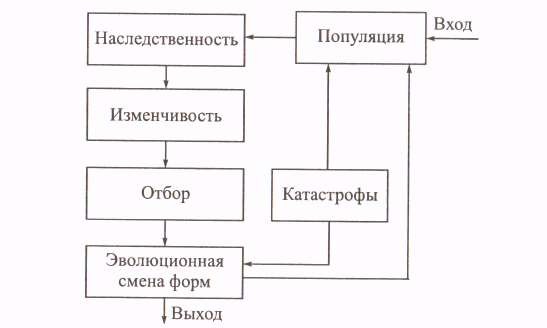
(предгорья) популяции (табуна) более предпочтительные шансы на добычу корма и, следовательно, на выживание имеют коротко ногие, невысокие особи, которым гораздо удобнее передвигаться в условиях пересеченной гористой местности.

5-6. *Отбор + Эволюционная смена форм.* Соответственно особи именно такого типа будут иметь преобладающие шансы на выживание и, следовательно, на закрепление своего фенотипа в популяции. Таким образом, если условия внешней среды останутся неизменными достаточно долго и данный табун продолжит обитать в данной местности, то после нескольких поколений указанный тип станет доминирующим в данной популяции. В свою очередь, высокие длинноногие лошади, скорее всего, обречены на вымирание в изменившихся условиях обитания.

Данная модель не получила применения в биологии.

Авторы считают, что ее применение является полезным при решении технических задач, когда популяция имеет сходимость в области локального оптимума

МОДЕЛЬ ЭВОЛЮЦИИ ДЕ ФРИЗА



В ее основе лежит моделирование социальных и географических катастроф, приводящих к резкому изменению видов и популяций.

Эволюция, таким образом, представляет собой

последовательность скачков в развитии популяции без предварительного накопления количественных изменений в эволюционных процессах.

Такой механизм эволюции иногда называют эволюцией катастроф. Он проявляется, ориентировочно, один раз в несколько тысяч поколений.

Основная идея его состоит во внесении глобальных изменений в генофонд на момент катастрофы.

Пусть существует известный нам табун лошадей из предыдущих

примеров. В ходе эволюции данной популяции происходит постепенная смена поколений, причем происходящие изменения генотипа

популяции носят регулярный постепенный характер.

Однако на некотором шаге эволюции данная популяция случайным

образом подвергается катастрофическому воздействию внешней среды,

которое приводит к значительному сокращению (или вымиранию)

популяции и вызывает кардинальное изменение генотипа.

В нашем примере таким фактором может быть большая засуха, в результате которой были уничтожены кормовые угодья. Вследствие случившегося голода выживает лишь незначительное количество особей из рассматриваемого табуна.

Эти пережившие катастрофу особи составят впоследствии новую

популяцию (табун) особей, с новыми качествами (рост, цвет и т. д.),

которая заменит старую популяцию, и процесс эволюции будет

продолжаться.

МОДЕЛЬ ЭВОЛЮЦИИ ПОППЕРА



Это условная структура, реализующая иерархическую систему

гибких механизмов управления, в которых мутация

интерпретируется как метод случайных проб и ошибок, а отбор

— как один из способов управления с помощью устранения ошибок при взаимодействии с внешней средой.

К. Поппер интерпретировал эволюцию Дарвина в виде триады: дедуктивизм - отбор –устранение ошибок.

Эволюция Поппера излагается в виде двенадцати тезисов. Основными

из них являются:

проблемы эволюции всегда решаются методом проб и ошибок;

устранение ошибок может осуществляться либо путем полного

устранения неудачных форм, либо в виде эволюции механизмов управления;

популяция использует тот механизм управления, который выработался в процессе эволюции;

популяция является пробным решением, анализируемым в процессе

эволюции, выбирающим окружающую среду и преобразующим ее;

эволюционная последовательность событий представляется в виде

последовательности

*F\* —*> TS —> ЕЕ* —> *fz,*

где *F\ • -* исходная проблема, *TS —* пробные решения, *ЕЕ —*

устранение ошибок, *fz —* новая проблема.

В отличие от эволюции Дарвина, где существует одна проблема -«выживание сильнейших», в

эволюции Поппера существуют и другие проблемы: воспроизводство, избавление от лишнего потомства и т. п.

Согласно К. Попперу естественные системы

исследуют окружающую среду и активно получают из нее информацию.

Процесс выбора лучшей индивидуальности в данной эволюции может являться процессом отбора (селекции), а отбор из некоторого множества

случайных событий не обязан быть случайным.

МОДЕЛЬ НЕЙТРАЛЬНОЙ ЭВОЛЮЦИИ КИМУРЫ



*М. Кимура* предложил модель нейтральной

эволюции с нейтральным отбором. По его теории на генетическую изменчивость исходно влияют

мутации, обуславливающие изменчивость и

генетический дрейф, исключающий ее в отсутствии дифференцирующего отбора.

Теория нейтральности предполагает, что большая часть молекулярных вариантов имеет равную

приспособленность друг относительно друга.

Изменчивость здесь поддерживается балансирующим отбором.

Пусть задана популяция, состоящая из «больших» и

«маленьких» особей. Тогда эволюция заключается в реализации последовательностей поколений.

Процесс реализации поколения состоит из двух шагов. На первом шаге дублируются все особи:

большие имеют два больших потомка, маленькие — имеют два маленьких потомка.

На втором шаге из популяции случайным образом удаляется ровно половина особей с равной вероятностью для больших и маленьких.

Рассматриваемый процесс всегда сходится к одному из поглощающих состояний (все особи большие или все маленькие).

Механизм работы модели нейтральной эволюции Кимуры можно пояснить на примере знакомого нам табуна лошадей.

Пусть существует некий табун лошадей (популяция). В этом табуне имеется две большие группы особей (лошадей), отличающиеся друг от друга окраской (вороные и гнедые).

Тогда согласно данной модели в результате

эволюции по прошествии некоторого определенного числа поколений в табуне останется только одна доминирующая группа особей (лошадей)

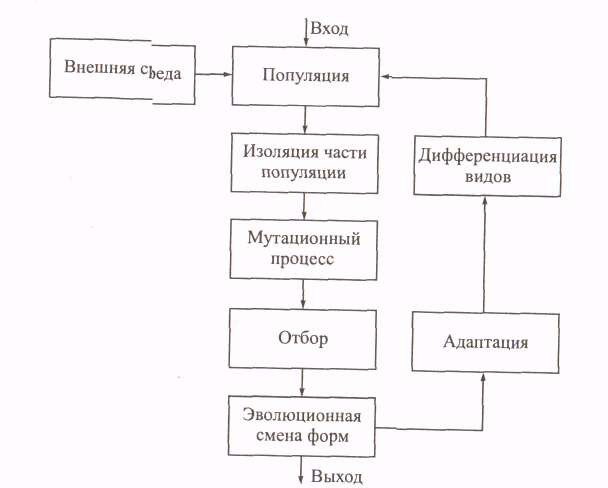
одинаковой окраски.

*Модель синтетической эволюции,* описанная Н. Дубининым, представляет интеграцию различных моделей эволюции, в том числе Ч. Дарвина, Ж. Ламарка и Г. де Фриза.

Процессы, происходящие при реализации синтетической теории эволюции по Н. Дубинину, включают случайные, периодические и скачкообразные колебания, а также колебания численности отдельных популяций, обусловленные процессом миграции.

МОДЕЛЬ СИНТЕТИЧЕСКОЙ ЭВОЛЮЦИИ ДУБИНИНА

МОДЕЛЬ СИНТЕТИЧЕСКОЙ ЭВОЛЮЦИИ ДУБИНИНА



Отметим, что признание единства факторов

эволюции в виде наследственности, изменчивости и естественного отбора не исключает существование разных форм эволюции.

Н. Дубинин выделяет четыре основные формы

осуществления внутренне единого эволюционного процесса:

микроэволюция (процессы внутривидовой эволюции);

эволюция на основе фазы нарастающего эволюционного

усовершенствования;

эволюция на основе переломных моментов;

эволюция на основе интеграционных особенностей в

организации естественных систем.

Эволюционный процесс связан с двумя типами

адаптации. Один тип адаптации основан на выработке приспособлений к условиям внешней среды, в которых вид существует в настоящее время. Другой тип адаптации связан с выработкой таких особенностей в структуре, которые должны обеспечить его

соревнование с другими видами во времени.

Основная задача синтетической теории эволюции — определение природы противоречий или постепенной эволюции, т.е. разных форм противоречий между наследственностью и постоянно меняющимися

потребностями в приспособлениях.

Кардинальное положение синтетической теории эволюции —

признание стохастичности процессов мутаций и больших

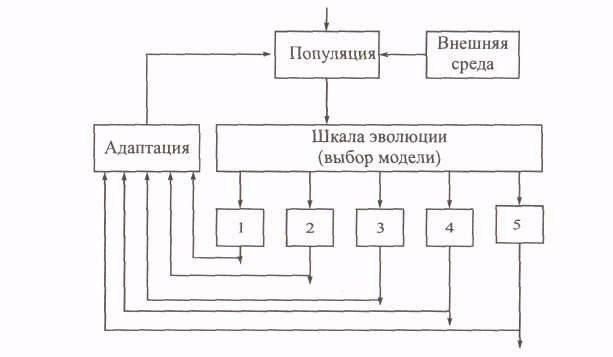
резервов рекомбинационной изменчивости. Условия внешней среды — не только факторы исключения неприспособленных особей, но и особенности, формирующие синтетическую

теорию эволюции.

В этой связи авторы считают важным объединение всех видов и моделей эволюции. Далее приведена условная упрощенная интегрированную схема эволюции.

Блоки 1-5 соответствуют схемам моделей эволюции Дарвина, Ламарка, де Фриза, Поппера и Кимуры *соответственно.* Основным этапом в каждой модели эволюции является анализ популяции, ее преобразование тем или иным

способом и эволюционная смена форм.



*Основой* для возникновения генетических алгоритмов

послужили модель биологической эволюции и методы

случайного поиска. Л. Растригин отмечал, что случайный

поиск возник как реализация простейшей модели эволюции, когда случайные мутации моделировались случайными

шагами оптимального решения, а отбор — «устранением»

неудачных вариантов.

*Эволюционный поиск* с точки зрения преобразования

информации — это последовательное преобразование одного конечного нечеткого множества промежуточных решений в другое.

Само преобразование можно назвать алгоритмом поиска, или генетическим алгоритмом. Генетические алгоритмы — это не просто случайный поиск. Они эффективно используют

информацию, накопленную в процессе эволюции.

Цель генетических алгоритмов состоит в том, чтобы:

абстрактно и формально объяснять адаптацию процессов в

естественной системе и интеллектуальной

исследовательской системе;

моделировать естественные эволюционные процессы для эффективного решения оптимизационных задач науки и техники.

В настоящее время используется новая парадигма решений оптимизационных задач на основе генетических алгоритмов и их различных модификаций. Генетические алгоритмы

осуществляют поиск баланса между эффективностью и качеством решений за счет «выживания сильнейших альтернативных решений» в неопределенных и нечетких условиях.

Генетические алгоритмы отличаются от других оптимизационных и поисковых процедур

следующим:

работают в основном не с параметрами задачи, а с

закодированным множеством параметров;

осуществляют поиск не путем улучшения одного решения,

а путем использования сразу нескольких альтернатив на

заданном множестве решений;

используют *целевую функцию,* а не ее различные

приращения для оценки качества принятия решений;

применяют не детерминированные, а вероятностные

правила анализа оптимизационных задач.

Для работы генетических алгоритмов выбирают множество

натуральных параметров оптимизационной проблемы и

кодируют их в последовательность конечной длины в некотором

алфавите.

Они работают до тех пор, пока не будет выполнено заданное

число генераций (итераций алгоритма) или на некоторой

генерации будет получено решение определенного качества, или когда найден локальный оптимум, т. е. возникла

преждевременная сходимость и алгоритм не может найти выход

из этого состояния.

В отличие от других методов оптимизации эти алгоритмы, как

правило, анализируют различные области пространства решений одновременно и поэтому они более приспособлены к

нахождению новых областей с лучшими значениями целевой

функции.

При решении практических задач с использованием генетических алгоритмов обычно выполняют четыре предварительных этапа:

выбор способа представления решения;

разработка операторов случайных изменений;

определение способов «выживания» решений;

создание начальной популяции альтернативных решений.

*На первом этапе* для представления решения в формальном виде требуется такая структура, которая позволит кодировать любое возможное решение и производить его оценку.

Математически доказано, что не существует

идеальной структуры представления, так что для создания хорошей структуры требуется анализ, перебор и эвристические подходы.

Возможный вариант представления должен

позволять проведение различных перестановок в альтернативных решениях.

Для оценки решений необходимо определить способ вычисления целевой функции.

*На втором этапе* достаточно сложным является выбор случайного оператора (или операторов)

для генерации потомков. Существует огромное число таких операторов. Существуют два основных типа размножения: половое и бесполое. При половом размножении два родителя

обмениваются генетическим материалом, который используется при создании потомка. Бесполое размножение — это фактически клонирование, при котором происходят различные мутации при передаче информации от родителя к потомку. Модели этих типов размножения играют важную роль в генетических алгоритмах.

В общем случае можно применить модели размножения, которые не существуют в природе. Например, использовать материал от трех или более родителей, проводить голосование при выборе родителей и т. п. При решении технических задач нет смысла слепо копировать законы природы и ограничиваться только ими.

Успех генетических алгоритмов во многом зависит от того, как взаимодействуют между собой схема представления, методы случайных изменений и способ определения целевой функции. Поэтому для определенного класса задач целесообразно использовать направленные методы.

В качестве примера рассмотрим два способа представления перестановок при решении оптимизационных задач. В первом случае будем использовать одного родителя (альтернативное решение) и получать одного потомка. Во втором случае используем двух родителей, случайно выберем точку перестановки и для образования потомка возьмем первый сегмент у первого

родителя, а второй сегмент — у второго. Первый метод похож на бесполое размножение, а второй

— на половое размножение. Стоит отметить, что если первый метод всегда генерирует реальное решение, то второй может генерировать недопустимые решения. При этом требуется

«восстанавливать» допустимые решения перед их оценкой.

*На третьем из рассматриваемых этапов* задаются правила выживания решений для создания потомства. Существует множество способов проведения селекции альтернативных решений. Простейшее правило — это «выживание сильнейших», т. е. остаются только лучшие решения с точки зрения заданной целевой функции, а все остальные

устраняются. Такое правило часто оказывается малоэффективным при решении сложных технических проблем. Иногда лучшие решения могут происходить от худших, а не только от самых лучших. Тем не менее, логично использовать принцип: «*Чем «лучше» решение, тем больше вероятность его выживания»*

•

*На последнем предварительном этапе* создается начальная

популяция. При неполноте исходных данных о проблеме решения могут случайным образом выбираться из всего множества альтернатив. Это реализуется генерацией случайных внутрихромосомных перестановок, каждая из которых представляет собой определенное решение. При создании начальной популяции рекомендуется использовать знания о решаемой задаче. Например, эти знания могут быть получены из опыта разработчика, существующих стандартов и библиотек алгоритмов решения задач данного класса.

*Эффективность генетического алгоритма —* степень

реализации запланированных действий алгоритма и достижение требуемых значений целевой функции. Эффективность во

многом определяется структурой и составом начальной

популяции. При создании начального множества решений происходит формирование популяции на основе четырех основных принципов:

«одеяло» — генерируется полная популяция, включающая все возможные решения в некоторой заданной области;

«дробовик» — подразумевает случайный выбор альтернатив из всей

области решений данной задачи.

«фокусировка» — реализует случайный выбор допустимых альтернатив из заданной области решений данной задачи.

«комбинирование» — состоит в различных совместных реализациях первых трех принципов.

Отметим, что популяция обязательно является конечным

множеством.

В каждой генерации генетического алгоритма хромосомы являются результатом применения некоторых генетических операторов.

*Оператор —* это языковая конструкция, представляющая

один шаг из последовательности действий или набора

описаний алгоритма.

Генетический алгоритм состоит из набора генетических

операторов.

*Генетический оператор* по аналогии с оператором алгоритма - средство отображения одного множества на другое. Другими словами, это конструкция,

представляющая один шаг из последовательности действий

генетического алгоритма.

Оператор репродукции (селекции)

Оператор кроссинговера

Оператор мутации

Оператор инверсии

*Оператор репродукции (селекция) —* это процесс, посредством которого хромосомы

(альтернативные решения), имеющие более высокое значение целевой функции (с

«лучшими» признаками), получают большую возможность для воспроизводства (репродукции) потомков, чем «худшие» хромосомы. Элементы, выбранные для репродукции, обмениваются генетическим материалом, создавая аналогичных или различных потомков.

Существует большое число видов операторов репродукции. К ним относятся

следующие.

*Селекция на основе рулетки* — это простой и широко используемый в простом генетическом алгоритме метод. При его реализации каждому элементу в популяции соответствует зона на колесе рулетки, пропорционально соразмерная с величиной целевой функции. Причем

элемент с большим значением целевой функции имеет большую вероятность для выбора.

*Селекция на основе заданной шкалы.* Здесь популяция предварительно сортируется от

«лучшей» к «худшей» на основе заданного критерия. Каждому элементу назначается определенное число и тогда селекция выполняется согласно этому числу.

*Элитная селекция.* В этом случае выбираются лучшие (элитные) элементы на основе сравнения значений целевой функции. Далее они вступают в различные преобразования,

после которых снова выбираются элитные элементы. Процесс продолжается аналогично до тех пор, пока продолжают появляться элитные элементы.

*Турнирная селекция.* При этом некоторое число элементов (согласно размеру «турнира») выбирается случайно или направленно из популяции, и лучшие элементы в этой группе на основе заданного турнира определяются для дальнейшего эволюционного поиска.

*Оператор кроссинговера —* это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков. Существует огромное число операторов кроссинговера, так как их структура в основном и

определяет эффективность генетических алгоритмов.

*Простой (одноточечный) оператор кроссинговера.* Перед началом работы одноточечного оператора кроссинговера определяется так называемая точка оператора кроссинговера, или разрезающая точка оператора кроссинговера, которая обычно определяется случайно.

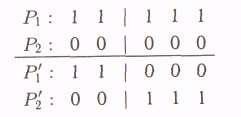
Эта точка определяет место в двух хромосомах, где они должны

быть «разрезаны».

Например, пусть популяция *Р* состоит из хромосом *Р1* и *P2,* которые выступают в качестве родителей, *Р = {P1, P2}.* Пусть первый и второй родители имеют вид *Р1 :* 11111, *Р2 :* 00000.

Выберем точку оператора кроссинговера между вторым и третьим генами в *Р1, Р2.*

Тогда, меняя элементы после точки оператора кроссинговера между двумя родителями, можно создать два новых потомка. В нашем примере получим:



Одноточечный оператор кроссинговера выполняется в три этапа:

Две хромосомы *А = а1,а2,...* , аL, и *В* = *а1’,а2’,*... *,aL’*

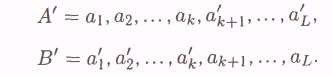
выбираются

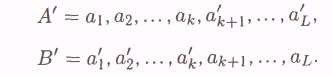
случайно из текущей популяции.

Число *k* выбирается из {1,2, ...,L— 1} также

случайно. Здесь L — длина хромосомы, *k —* точка оператора кроссинговера (номер, значение или код гена, после которого выполняется разрез хромосомы).

Две новые хромосомы формируются из А и В путем перестановок элементов согласно правилу:





После применения оператора кроссинговера имеем две старые

хромосомы и всегда получаем две новые хромосомы.

Схематически простой оператор кроссинговера показывает

преобразование двух хромосом и частичный обмен

информацией между ними, использующий точку разрыва,

выбранную случайно.

В каждой хромосоме определяются две точки оператора кроссинговера, и хромосомы обмениваются участками, расположенными между двумя точками оператора кроссинговера.

Точки оператора кроссинговера в двухточечном операторе кроссинговера также определяются случайно.

Существует большое количество модификаций двухточечного оператора кроссинговера.

Развитием двухточечного оператора кроссинговера является

многоточечный или *N*-точечный оператор кроссинговера. Многоточечный оператор кроссинговера выполняется аналогично двухточечному, хотя большое число «разрезающих» точек может привести к потере «хороших» родительских свойств.

Выделяют также:

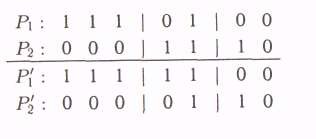
упорядоченный оператор кроссинговера (ОК);

частично-соответствующий ОК;

циклический ОК;

универсальный ОК;

жадный ОК



Здесь точки оператора кроссинговера делят хромосому на ряд

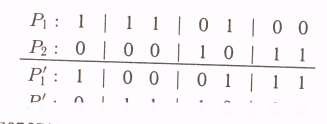
строительных блоков (в данном случае 4).

Потомок *Р1’* образуется из нечетных блоков родителя *Р1 (1 и 3)* и

четных блоков родителя *Р2 (2 и 4).*

Потомок *Р2'* образуется соответственно из нечетных блоков

родителя *Р2* и четных блоков родителя *P1.*



*Оператор мутации —* языковая конструкция, позволяющая на

основе преобразования родительской хромосомы (или ее части)

создавать хромосому потомка.

Оператор мутации обычно состоит из двух этапов:

1. В хромосоме *А = (а1, а2, а3,* aL-2, aL-1, aL*)* определяются

случайным образом две позиции (например, *а2* и aL-1).

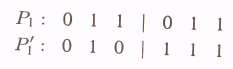
2. Гены, соответствующие выбранным позициям, переставляются,

и формируется новая хромосома

*А' = (а1, аL-1, a3, ••• ,аL-2, a2, aL)*

Простейшим оператором мутации является одноточечный. При

его реализации случайно выбирают ген в родительской хромосоме и, обменивая его на рядом расположенный ген, получают хромосому потомка



При реализации *двухточечного оператора мутации* случайным или направленным образом

выбираются две точки разреза. Затем производится перестановка генов между собой, расположенных

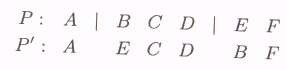
справа от точек разреза.

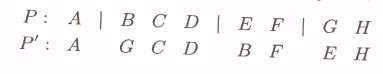
Развитием двухточечного оператора мутации

является многоточечный (или n-точечный) оператор мутации. В этом случае происходит

последовательный обмен генов, расположенных правее точек разреза друг с другом в порядке их расположения. Ген, расположенный правее

последней точки разреза, переходит на место первого.





*Оператор инверсии -* это языковая конструкция, позволяющая на основе инвертирования родительской хромосомы (или ее части) создавать хромосому потомка. При его реализации случайным образом определяется одна или несколько точек разреза

(инверсии), внутри которых элементы инвертируются.

Генетический оператор инверсии состоит из следующих шагов.

Хромосома *В = b1, b2, •••, bL* выбирается случайным образом

из текущей популяции.

Два числа *у1’* и *у2’* выбираются случайным образом из

множества *{0, 1,2,..., L + 1},* причем считается, что *у1*= min *{y1’, у2’}*

и *у2 = max {y1’, у2’} .*

Новая хромосома формируется из *В* путем инверсии сегмента, который лежит справа от позиции *у1* и слева от позиции *у2* в хромосоме *В.* Тогда после применения оператора инверсии

получаем:







Транслокации

Транспозиции

Сегрегации

Удаления

Вставки

Редукции

Рекомбинации

*Оператор транслокации —* это языковая

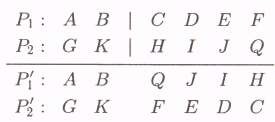
конструкция, позволяющая на основе скрещивания и инвертирования из пары родительских хромосом (или их частей) создавать две хромосомы потомков.

Другими словами, он представляет собой

комбинацию операторов кроссинговера и инверсии.

В процессе его реализации случайным образом

производится один разрез в каждой хромосоме. При формировании потомка *Р1’* берется левая часть до разрыва из родителя *Р1* и инверсия правой части до разрыва из *Р2.* При создании Р2' берется левая часть P2 и инверсия правой части *Р1,* например:



*Оператор транспозиции* — языковая конструкция, позволяющая на основе преобразования и инвертирования выделяемой части родительской хромосомы создавать хромосому потомка.

Например, возьмем три точки разреза в хромосоме

P1 : A’BC’DEF’GH

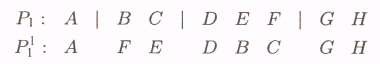
Точки разреза выбираются случайным или направленным

образом. В родительской хромосоме *Р1* строительный блок *DEF* инвертируется и вставляется в точку разреза между генами А и В. В результате получаем хромосому-потомок

Р1’ : A FE DBC GH

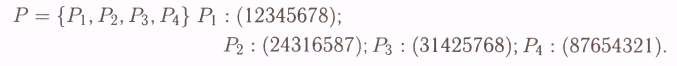
Отметим, что существует большое количество модификаций

оператора транспозиции.



*Оператор сегрегации - э*то языковая конструкция, позволяющая на основе выбора строительных блоков из хромосом родителей (или их частей) создавать хромосомы потомков.

Приведем один из примеров его реализации. Отметим, что оператор сегрегации, как правило, реализуется на некотором наборе хромосом. Пусть имеется популяция *Р,* состоящая из четырех родительских хромосом:



В каждой хромосоме выделим строительные блоки. Выделим по два строительных блока: в Р1 — 23 и 67, в *Р2*

*—* 1 и 4, в Р3 — 768 и 425 и в Р4 *—* 54 и 87.

Тогда, например, потомок *Р1’* можно сформировать, взяв первые строительные блоки из каждой родительской хромосомы популяции. Так, вариант *Р1’* будет

представлен последовательностью (23176854) и является

реальным решением, а вариант *Р2'* (67442587) является нереальным (недопустимым).

Очевидно, что оператор сегрегации можно реализовать различными способами в зависимости от выборки строительный блоков или генов из хромосом.

*Оператор удаления - э*то языковая конструкция,

позволяющая на основе удаления строительных блоков из хромосом родителей (или их частей) создавать хромосомы потомков.

При его реализации направленным или случайным образом

определяется точка или точки разреза.

Далее производится пробное удаление генов или их строительных блоков с вычислением изменения значения целевой функции. Элементы, расположенные справа от точки оператора удаления или между двумя точками, удаляются из хромосомы.

При этом производится преобразование потомка таким образом, чтобы соответствующее альтернативное решение оставалось реальным.

*Оператор вставки -* языковая конструкция, позволяющая на основе вставки строительных блоков в хромосомы родителей создавать хромосомы потомков.

При его реализации направленным или случайным образом создается хромосома (донор), состоящая из строительных блоков, которые желательно разместить в другие хромосомы популяции.

После этого направленным или случайным образом определяется хромосома для реализации оператора вставки. В ней находится точка или точки разреза.

Затем анализируются другие хромосомы в популяции для определения альтернативных вставок.

Далее производится пробная вставка строительных блоков с вычислением изменения значения целевой функции и получением реальных решений.

Новые строительные блоки вставляются в хромосому справа от точки

оператора вставки или между его двумя точками.

Отметим, что оператор удаления и оператор вставки могут изменять размер хромосом. Для сохранения постоянного размера хромосом эти операторы можно применять совместно.

*Оператор редукции* — языковая конструкция, позволяющая на

основе анализа популяции после одной или нескольких

поколений генетического алгоритма уменьшать ее размер до

заданной величины.

Рассмотрим способы реализации оператора редукции. Он выполняется для устранения неудачных решений. В некоторых генетических алгоритмах, в частности, в простом генетическом алгоритме, этот оператор применяется для сохранения

постоянного размера начальной популяции.

Основная проблема здесь — нахождение компромисса между

разнообразием генетического материала и качеством решений.

Сначала формируют репродукционную группу из всех решений, образовавшихся в популяции *Nt*, затем выполняют отбор

решений в следующую популяцию.



где

*Nt+1* - численность новой популяции,

*Nt* - численность популяции на предыдущем шаге (поколении) *t,*

*NOK, NOM, N*ОИ*, NOT, NOTР, Noc, Noy, NOB —* потомки,

полученные в результате применения операторов: кроссинговера, мутации, инверсии, транслокации, транспозиции, сегрегации, удаления, вставки.

Отметим, что оператор редукции может

применяться после каждого оператора или после всех в одной генерации генетического алгоритма.

Оператор рекомбинации — языковая конструкция, которая определяет, как

новая генерация хромосом будет построена из родителей и потомков. Другими словами, оператор рекомбинации — это технология анализа и преобразования популяции при переходе из одной генерации в другую.

Существует много путей выполнения рекомбинации. Один из них состоит из перемещения родителей в потомки после реализации

каждогогенетического оператора.

Другой путь заключается в перемещении

некоторой части популяции после каждой генерации.

Часто в генетических алгоритмах задается параметр *W(P),* который управляет этим процессом.

Так, *Np(1* - *W(P))* элементов в популяции *Р,* выбранных случайно, могут

«выжить» в следующей генерации. Здесь *Np --* размер популяции. Величина *W(P) =* 0 означает, что целая предыдущая популяция перемещается в новую популяцию в каждой генерации.

При дальнейшей реализации алгоритма лучшие или отобранные элементы

из родителей и потомков будут выбираться для формирования новой

популяции.

В инженерных задачах используются различные механизмы и модели

рекомбинации:

Ml — вытеснение (*crowding factor*). Этот механизм определя

ет способ и порядок замены родительских хромосом из генерации *t* хромосомами потомками после генерации *t+* 1. Механизм реализован таким образом, что стремится удалять «похожие» хромосомы из популяции и оставлять отличающиеся.

М2 — разделение (*sharing*). Этот механизм вводит зависимость

значения целевой функции хромосомы от их распределения в популяции и поисковом пространстве. Это позволяет копиям родительских хромосом или близких к ним не появляться в популяциях.

МЗ — введение идентификаторов (*tagging*). Специальным хромо сомам присваиваются метки. Операторы генетических алгоритмов применяются только к помеченным хромосомам.

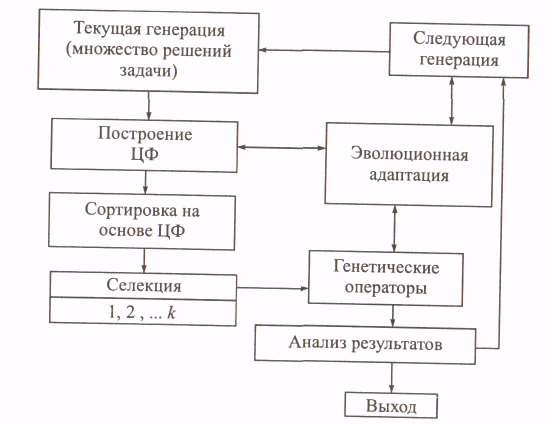
Отметим, что оператор редукции является частным случаем оператора

рекомбинации.

ПРОСТОЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

*Простой генетический алгоритм* случайно генерирует популяцию хромосом (альтернативных упорядоченных и неупорядоченных решений). Затем производится копирование хромосом, перестановка их частей и генерация новых хромосом (решений) на основе трех операторов: репродукции, кроссинговера и мутации.

БАЗОВАЯ СТРУКТУРА ГЕНЕТИЧЕСКОГО АЛГОРИТМА



БАЗОВАЯ СТРУКТУРА ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Блок эволюционной адаптации -- специальный блок, который на основе обратных связей управляет процессом эволюционного поиска. Согласно данной схеме на первом этапе случайным, направленным или комбинированным методом получают некоторое подмножество решений рассматриваемой задачи.

Эти решения образуют текущую генерацию или

популяцию исследуемых решений на шаге

*t (t =* 0,1,... , Т).

Далее вводится или вычисляется значение целевой функции. Вычисление целевой функции является сложной задачей, причем от точности значений целевой функции зависит качество будущих решений.

*Качество —* степень, с которой совокупность присущих решению задачи характеристик удовлетворяет заданным требованиям.

БАЗОВАЯ СТРУКТУРА ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Отметим, что для каждой оптимизационной задачи желательно строить новую целевую функцию. При построении целевой функции необходимо использовать знания о конкретной задаче

На основе целевой функции производятся

ранжирование и сортировка популяции решений.

Затем в результате различных методов селекции в популяции подбираются родительские хромосомы для применения различных генетических операторов.

После реализации всех операторов получается новое подмножество решений *Р'.* Оно объединяется с первоначальным подмножеством решений.

Получается новое множество *РГА = PUP'.*

БАЗОВАЯ СТРУКТУРА ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Используя значение целевой функции, производится анализ *РГА*. Все элементы в *РГА* (решения задачи), значения целевой функции которых хуже заданного порога, являются с нашей точки зрения неперспективными решениями и удаляются из *РГА.*

Получается новое множество *Р‘ГА* , причем | *Р‘ГА* | = |*Р*|.

Если данное условие не выполняется, например, |*Р‘ГА*

| < |*Р*|, то в *Р‘ГА* включается элемент с лучшими характеристиками из отброшенных. Множество *Р‘ГА* объявляется новой текущей популяцией решений и далее процесс может повторяться на основе блока *эволюционной адаптации* итерационно до получения подмножества или одного оптимального решения.

НЕСТАНДАРТНЫЕ АРХИТЕКТУРЫ

ЭВОЛЮЦИОННОГО ПОИСКА

В настоящее время известно много нестандартных архитектур эволюционного поиска, позволяющих в большинстве случаев частично решать проблему предварительной сходимости алгоритмов. Это методы миграции, метагенетической оптимизации, стохастически- итерационные генетические и поисковые, методы «прерывистого равновесия», объединения генетического поиска и моделирования отжига и др.

Определения:

*Процедура --* установленный порядок выполнения

какой-либо деятельности или процесса.

*Миграция —* процедура установления порядка обмена хромосом между популяциями.

*Микроэволюция --* это создание одной хромосомы и реализация на

ее основе эволюционного поиска.

*Макроэволюция —* это создание одной популяции и реализация на

ней эволюционного поиска.

*Метаэволюция* — это создание множества популяций и реализация на нем эволюционного поиска.

АРХИТЕКТУРА ЭВОЛЮЦИОННОГО ПОИСКА

С МИГРАЦИЕЙ

Здесь предварительно выполняется этап *метаэволюции,* т. е. создается не одна популяция, а некоторое множество.

Эволюционный поиск осуществляется путем объединения хромосом из различных популяций. Блоки 1-3 в этой схеме представляют собой простой генетический алгоритм. Заметим, что в каждом блоке выполняются различные операторы репродукции.

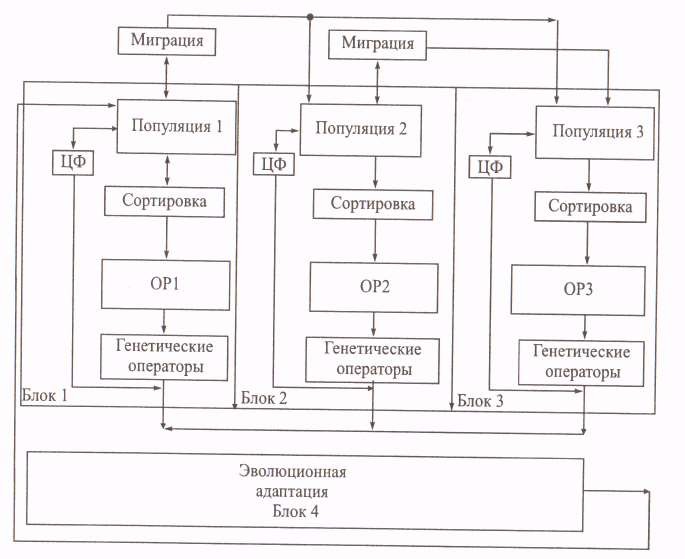
В первом блоке — ОР1 на основе рулетки.

Во втором блоке используется ОР2 на основе заданной шкалы. В третьем блоке — ОРЗ на основе элитной селекции. В блок миграции каждый раз отправляется лучший представитель из популяции.

Связь между блоками 1-3 осуществляется на основе следующих путей: (1-2, 2-3), (1-3).

В блоке 4 осуществляется установление баланса на размер популяции на основе эволюционной адаптации и управления всем процессом эволюционного поиска.

СХЕМА ЭВОЛЮЦИОННОГО ПОИСКА НА ОСНОВЕ МИГРАЦИИ

****

МЕТАГЕНЕТИЧЕСКИЙ

ОПТИМИЗАЦИОННЫЙ ПРОЦЕСС

Основным является первый блок, в котором осуществляется реализация генетического алгоритма, генерация новых решений, определение значения целевой функции и использование предыдущих решений для генерации лучших результатов.

Второй блок позволяет использовать «историю» предыдущих решений для генерации лучшего множества параметров.

В третьем блоке генерируется новое множество оптимизационных

параметров.

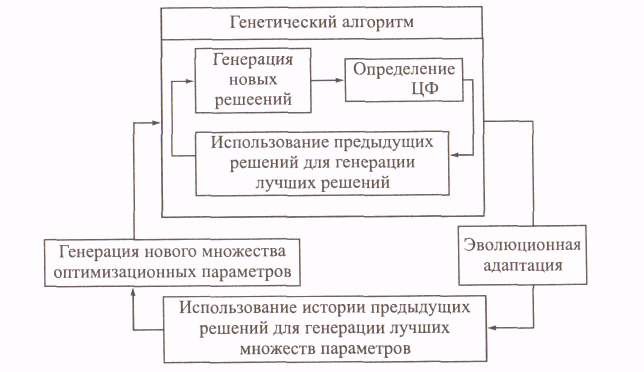
Используя метагенетический оптимизационный процесс, можно случайным, направленным или случайно-направленным способом генерировать начальные популяции, моделировать их и выполнять генетический алгоритм.

Можно случайно выбирать родителей из популяции с произвольной или заданной вероятностью, причем вероятность выполнения каждого оператора может определяться пропорционально значению его целевой функции.

Окончательное множество параметров выбирается после моделирования из конечной популяции на основе блока эволюционной адаптации. Для каждой оптимизационной задачи желательно строить конкретный метагенетический алгоритм. Такое построение зависит от наличия вычислительных ресурсов и времени, заданного на получение окончательного решения.

СХЕМА РЕАЛИЗАЦИИ ПРОЦЕССА

МЕТАГЕНЕТИЧЕСКОЙ ОПТИМИЗАЦИИ



УПРОЩЕННАЯ МОДЕЛЬ ЭВОЛЮЦИОННОГО

ПОИСКА ПРИ СОВМЕСТНОМ ИСПОЛЬЗОВАНИИ

МОДЕЛЕЙ ЭВОЛЮЦИИ ДАРВИНА И ЛАМАРКА

Первый элемент эволюции Ламарка предусматривает случай, когда *лицо, принимающее решение (ЛПР)* применяет ее ко всей популяции (шкала =1). Этот элемент собирает полное множество номеров хромосом в популяции и для них вызывает алгоритм Ламарка.

Второй элемент используется, если процентное соотношение в популяции больше 0. Тогда число хромосом для процесса вычисляется по параметру процентного соотношения и размеру популяции и соответственно обрабатывается набор номеров хромосом в популяции.

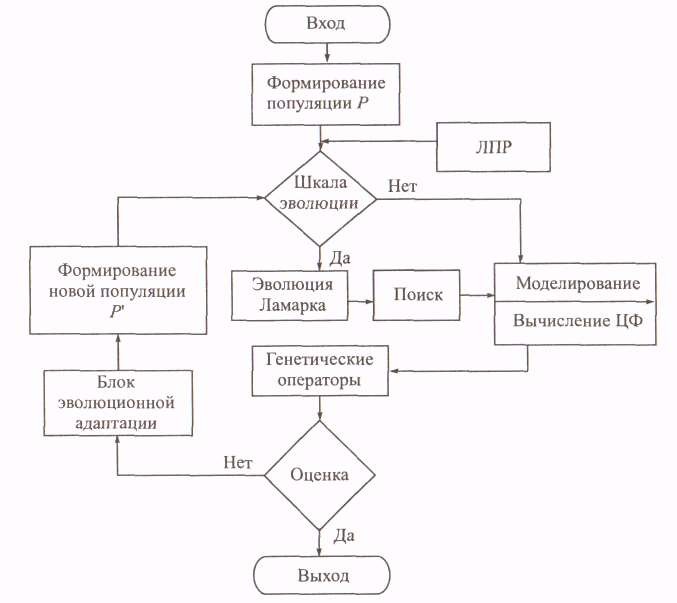
Второй элемент эволюции Ламарка вызывает предикат для выбора множества номеров хромосом, лежащих в пределах от 1 до размера популяции. Для выполнения этого предикат многократно повторяется до тех пор,пока не будет набран требуемый размер. Здесь используется стандартный оператор репродукции для выбора хромосом.

Если большой процент популяции подвергается эволюции Ламарка, то этот процесс может быть медленным, так как выбор хромосом для даннойэволюции требует дополнительного времени.

Если ЛПР не желает использовать эволюцию Ламарка, то шкалу

выбираем равную 0.

СТРУКТУРНАЯ СХЕМА СОВМЕСТНОГО АЛГОРИТМА ЭВОЛЮЦИЙ ДАРВИНА И ЛАМАРКА



ЗАДАЧА ПРИНЯТИЯ РЕШЕНИЙ

*Задачей принятия решений* называют кортеж *Z = {N, Q*} , где *N —* множество вариантов решений задачи; Q — принцип оптимальности, дающий представление о качестве вариантов, в простейшем случае — правило предпочтения вариантов.

Решением задачи *А* называют множество *Non,* входящее в *N,* полученное на основе принципа оптимальности. Задачу *Z* решают следующим образом. Составляют множество *N,* если это возможно, т. е. определяют варианты, а затем решают задачу выбора.

Стандартная система поддержки принятия решения состоит из следующих

основных блоков:

генерация возможных альтернатив решений (сценариев);

оценки решений (построения целевой функции));

согласование решений, анализ динамики развития ситуации;

выбор решения (группы решений), сценария;

оценка соответствия принятых решений заданным целям.

Сравнивая эту систему со структурой генетических алгоритмов, можно заметить много общего. Следовательно, предложенные концепции, принципы, архитектуры и генетические алгоритмы могут быть использованы в системах поддержки принятия решения.

КОМБИНИРОВАННАЯ СХЕМА ПРИНЯТИЯ РЕШЕНИЯ НА ОСНОВЕ БЛОКА ЭВОЛЮЦИОННОЙ АДАПТАЦИИ



КОММЕНТАРИЙ К КОМБИНИРОВАННОЙ СХЕМЕ ПРИНЯТИЯ РЕШЕНИЯ НА ОСНОВЕ БЛОКА ЭВОЛЮЦИОННОЙ АДАПТАЦИИ

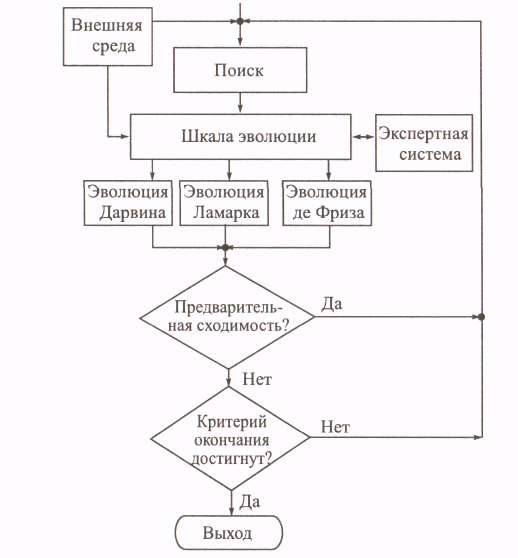
ПГА — это простой генетический алгоритм, реализующий модель эволюции Дарвина. Блок эволюции Дарвина и эволюции Ламарка

— это модель их совместной реализации. Компенсатор регулирует динамически изменяемый размер популяции решений, расширяя, сужая или оставляя ее постоянной. После реализации эволюции Дарвина и Ламарка компенсатор во взаимодействии с внешней средой реализует принципы эволюционного поиска, при этом лучшие хромосомы выбираются для смешивания популяций и выхода из локальных оптимумов.

Компенсатор, редуктор и фильтр позволяют повысить эффективность реализации эволюции и скорость принятия решений. В инженерных задачах большой размерности процесс решения усложняется, но выполнение параллельных генетических алгоритмов снижает временную сложность.

Внешняя среда в конечном итоге определяет эволюцию, но не в простой форме связи между наследственной изменчивостью организмов и средой, как в эволюции Ламарка, а в более сложной форме.

БАЗОВАЯ СТРУКТУРА ГЕНЕТИЧЕСКОГО ПОИСКА НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ЭВОЛЮЦИИ ДАРВИНА, ЛАМАРКА И ДЕ ФРИЗА

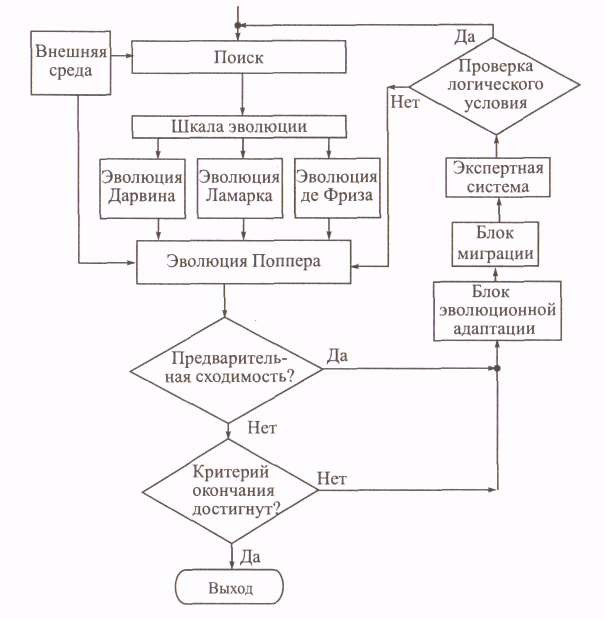


КОММЕНТАРИЙ К СХЕМЕ

Здесь на основе шкалы эволюции при взаимодействии с внешней средой вырабатываются сигналы: 0 — на выполнение эволюции Дарвина; 1 — на реализацию эволюции Ламарка; 0,5 — на реализацию эволюции де Фриза.

Экспертная система позволяет сформулировать набор правил построения целевой функции и окончания поиска. Повторение эволюционного поиска возможно при предварительной сходимости алгоритма или при достижении заданного значения целевой функции. Особенностью данной схемы является использование поисковых стратегий, описанных выше.

МОДИФИКАЦИЯ БАЗОВОЙ СТРУКТУРЫ



КОММЕНТАРИЙ К СХЕМЕ

В отличие от базовой структуры, шкала эволюции, взаимодействуя только с внешней средой, вырабатывает сигналы на выбор эволюции Дарвина (0), Ламарка (1), де Фриза (0,5). После этого выполняется модель эволюции Поппера, реализующая один из видов эвристического поиска в виде метода проб и ошибок.

В цепи обратной связи добавлены блоки адаптации и миграции. Они позволяют производить построение порядка из хаоса, установление баланса в системе, выбор параметров для управления эволюционным поиском с целью получения оптимальных и квазиоптимальных решений.

ПОВЫШЕНИЕ КАЧЕСТВА ПРИНЯТИЯ РЕШЕНИЙ

Для повышения качества принятия решений используют, кроме описанных, технологии поиска, связанные с распараллеливанием генетических алгоритмов, структурированием популяции, миграцией хромосом и популяций и т. п.

Обычно используются два подхода структурирования популяции.

После конструирования популяции она случайным или заданным образом разбивается на несколько подпопуляций. Модели эволюции Дарвина, Ламарка, де Фриза, Поппера и Кимуры или любая их модификация применяются внутри каждой подпопуляций. При этом возможно случайное или направленное перемещение хромосом между любыми или заданными популяциями.

После инициализации популяции для каждой хромосомы определяется ее пространственное местоположение в популяции. Далее, если необходимо, выполняется разбиение популяции. Выбор хромосом для использования генетических операторов зависит от «близости» решений в пространстве (значение целевой функции). Возможно большое количество комбинаций и модификаций этих подходов с учетом блока эволюционной адаптации, причем эффективность их устанавливается в основном эмпирически.

УКРУПНЕННАЯ СХЕМА ПАРАЛЛЕЛЬНОГО

ЭВОЛЮЦИОННОГО ПОИСКА ПРИ РАЗБИЕНИИ ПОПУЛЯЦИИ НА ДВЕ ПОДПОПУЛЯЦИИ



УКРУПНЕННАЯ СХЕМА ПАРАЛЛЕЛЬНОГО

ЭВОЛЮЦИОННОГО ПОИСКА ПРИ РАЗБИЕНИИ ПОПУЛЯЦИИ НА ДВЕ ПОДПОПУЛЯЦИИ

Здесь в блоках генетических операторов выполняются операторы кроссинговера, мутации, инверсии, сегрегации, транслокации, удаления и вставки.

В блоке редукции производится удаление хромосом со значением целевой функции ниже средней.

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ, ИСПОЛЬЗУЮЩИЕ

ИДЕИ ПОСТРОЕНИЯ ФРАКТАЛОВ

В 1980 г. Б. Мандельброт указал на фрактальную геометрию природы.

Согласно Б. Мандельброту, эволюционирующие системы имеют фрактальную природу и наличие направленного нестихийного отбора и самосогласованную эволюцию.

*Фрактальные объекты* самоподобны, т.е. их вид не претерпевает существенных изменений при изменении масштабов их деятельности. Множества, имеющие такую структуру, считаются обладающими геометрической (масштабной) универсальностью.

Преобразования, создающие такие структуры, — это процессы с обратной связью с большим числом итераций, когда одна и та же операция выполняется снова и снова, аналогично эволюционным процессам. Здесь результат одной итерации является начальным условием для другой и требуется нелинейная зависимость между результатом и реальным значением.

Такие множества объектов называются *фрактальными множествами.* Основными примерами таких множеств являются множество Кантора и ковер Серпинского. Эти множества обладают геометрической инвариантностью и называются «множества средних третей».

МНОЖЕСТВО КАНТОРА

Рассмотрим простой вариант построения *множества Кантора.* Пусть на вещественной оси задан отрезок единичной длины [0,1]. Он делится на три равные части и средняя часть является открытым интервалом (1/3, 2/3 вырезается), как показано на рисунке.

Аналогичные действия выполняются с каждым из оставшихся отрезков. Получаем последовательность отрезков убывающей длины. На первом этапе — это один отрезок, на втором — два, на третьем — 4 и т.д., на *k-м — 2k.* При *k —*> оо, имеем множество точек, называемое множеством Кантора. Суммарная длина всех вырезанных отрезков равна 1.

Множество Кантора, промежуточное между точками *d*

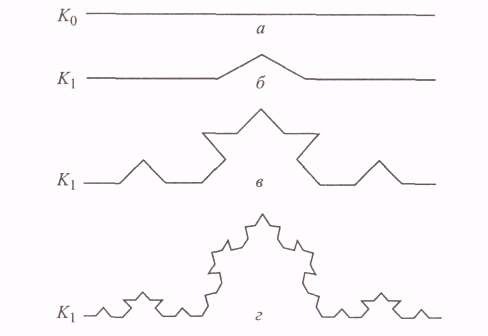
*= 0* и *d =* 1, является *фракталом.*

ПРИМЕР ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА



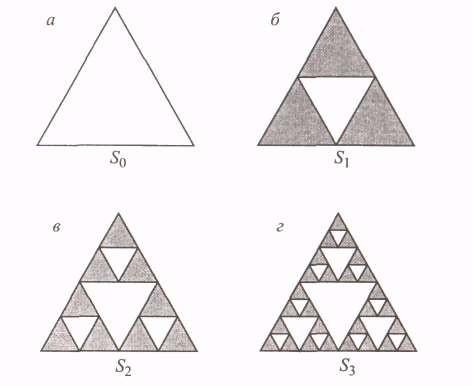
ПРОЦЕСС ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА

«СНЕЖИНКИ» (КРИВАЯ КОХА)



ОБОБЩЕНИЕ МНОЖЕСТВА КАНТОРА НА СЛУЧАЙ ПЛОСКИХ ФИГУР ПРИВОДИТ

К *«КОВРУ СЕРПИНСКОГО».*



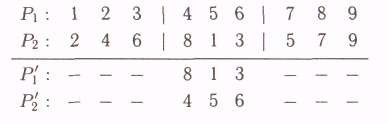
ОПЕРАТОР КРОССИНГОВЕРА НА ОСНОВЕ ИДЕЙ

ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА

Пусть заданы две родительские хромосомы *Р1, Р2.* Согласно идее построения множества Кантора в *Р1, Р2* определим две точки разрыва, которые делят хромосому на три равные части (1/3 + 1/3 + 1/3). Если при делении длины хромосомы на 3 ответ не является целым, то берутся ближайшие целые, например, при *L = 7* получим три части *L1=2, L2=2, Lз = 3 (L1 + L2 + Lз = 7).*

Вторая часть хромосомы *Р2* помещается в хромосому- потомок *Р’1 ,* а вторая часть хромосомы *Р1* помещается в хромосому- потомок *Р’2.*

Дальнейшее заполнение потомка *Р’1* выполняется из *Р1* слева направо, исключая повторяющиеся и вырезанные гены. Пустые позиции заполняются генами из *Р2 .* Аналогичная процедура выполняется и для построения потомка *Р’2 .*



ОПЕРАТОР МУТАЦИИ НА ОСНОВЕ ИДЕЙ

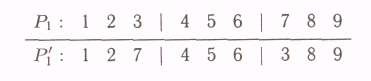
ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА

Он заключается в перестановке генов, находящихся за точками разреза.

Например, пусть задана родительская хромосома *Р1.*

Определим две точки разрыва в *Р1,* которые делят хромосому на три равные или близкие к ним части

ПРИМЕР ОПЕРАТОРА МУТАЦИИ НА ОСНОВЕ ИДЕЙ ПОСТРОЕНИЯ МНОЖЕСТВА КАНТОРА

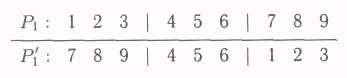


МОДИФИКАЦИЯ ОПЕРАТОРА МУТАЦИИ

МНОЖЕСТВА КАНТОРА

Модификацией оператора мутации множества Кантора является процедура, когда первая часть родительской хромосомы меняется с третьей

ПРИМЕР МОДИФИКАЦИИ ОПЕРАТОРА МУТАЦИИ МНОЖЕСТВА КАНТОРА



ВЫВОД

На основе множества Кантора и ковра Серпинского можно строить любой генетический оператор.

При использовании последовательного поиска выполняется перебор точек разрыва для нахождения хромосомы с оптимальным значением целевой функции.