**Интеллектуальная система** – «(в узком смысле) компьютерная программа». Это система, которая решает задачи (здесь задачи подразумеваются как, например, найдите ошибку, выбросите лишнее, запомнить и тд.) в предметной области, используя для этого формализованное представление содержания этой области в качестве данных для работы системы. «ИС решает задачи предметной области». ИС опирается на знания предметной области, правила, законы, соотношения, ограничения всякие. В БД это триггеры, формулы, замечания, проверки дополнительные.

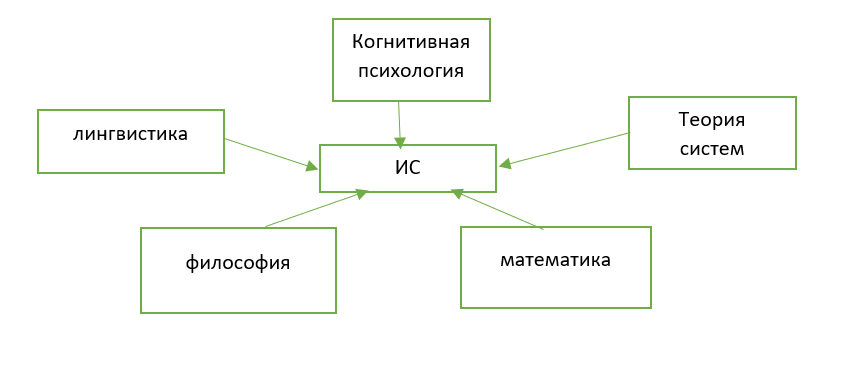
Для решения задач ИС применяют модели, моделирующие структуры памяти и процессы мышления. «Система ведет себя так, как ведет себя человек».

Для работы со знаниями применяются специальные форматы представления и операции обработки знаний. Операции над знаниями, основной операцией, если мыслить, как математик, это математическое рассуждение. В жизни с операциями сложнее, потому что могут быть операции понимания, запоминания (сохранить), ассоциирования знания с ситуацией.

Интеллектуальная система внутрь и наружу.

«наружу»

Формирование концепции ИС осуществляется с использованием понятий и конструктов областей знаний, которые занимаются изучением модели мышления и организации представления знания.



В философии есть два раздела: гносеология – теория познания и теория о том, как выглядят модели – антология.

Антология – система понятий, образующая каркас элементов, из которых собирается содержание предметной области, с помощью комбинаций которых создается описание содержания. Антология – это то, что рассказано на естественном языке.

Гносеология – как познавать, это механизм. Самый общий признак.

Математика: множества, операции над множествами, операции над элементами множеств и тд. Математика дает средства формализации.

Лингвистика – некая формализованная теория языка. Языка прежде всего естественного, разговорного. Язык выдает систему понятий, которыми апеллирует человек. И эти понятия уже можно каким-то образом интегрировать в какие-то структуры более-менее формализованные, применяемые в описаниях знаний и процессов обработки.

Описание знаний – это структурные элементы.

Процессы обработки – название операций, название действий.

Когнитивная психология: как мир познается? Когнитивная психология – психология мышления.

Теория систем занимается исследованием сложных систем, с разными моделями и свойствами поведения. То есть это не одна задача, которую надо решать, а какой-то агрегат, как сложная система. (пожар, как сложная система. Мы должны знать то, что знают пожарные, ДПС, страховщики, за что отвечает медицина, чиновники, электросети, гор. Газ, средства массовой информации и тд.).

Этапы развития концепции ИС

1. Этап универсальных ИС (УИС). «универсальность – значит умеет решать все». В основе подхода лежит применение языка математики (вычисление предикатов).

Механизм решения задач – это правило вывода, применяемое в вычислениях предикатов.

Правило modus ponens (MP): если A, то B.

Правило modus tolles (MT) – от противного: ¬A, A←B, ¬B.

Данный этап закончился неудачей потому, что

* невозможно формализовать содержание больших областей знаний из-за большого объема неявны знаний;
* переборность общих механизмов решения задач.

1. Этап пакета прикладных программ (ППП) – это библиотека алгоритмов решения конкретных задач в заданной предметной области, снабженная механизмом подбора нахождения алгоритма наилучшей для решаемой задачи.
2. Этап экспертных систем 1970г. MYCIN – система диагностики назначения лечения вирусного заболевания крови. Экспертная система «заменяет эксперта».

Экспертная система – это ИС в узкой предметной области, которая решает задачи в этой области, используя ограниченное множество знаний и методы их обработки, применяемые специалистами.

Экспертная система ничего нового не предлагает то, что уже известно предметной области и то, как специалист принимает решение, она берет оттуда, там все расписано. Не надо никаких универсальных механизмов обработки и общих схем.

Каждая экспертная система своя в каждой области знаний. Нет универсальных моделей, единой модели представления знаний и нет единого языка. Она должна быть маленькой.

Недостатком является сложность обобщения и сравнения систем, которые свои для каждой области знаний.

1. Этап семантических информационных систем. Данный этап связан с появлением информационных ресурсов иерархической структуры. Такие ресурсы развивают концепцию SW – Semantic Web, реализованную для решения задач навигации и поиска нужно наделять ресурсы наборами свойств. В рамках этого подхода используются XML (RDF, OWL), которые позволяют создавать иерархические семантические структуры (деревья), листья которых содержат информационные объекты, а вершины – различные отношения и связывания между ресурсами.

Они соответствуют иерархическим БД сложных семантических объектов, позволяющих моделировать знания.

Иерархический формат представления знаний является унифицированным.

**Понятие задачи. Задачи и этапы их решений.**

Задачи – это описание ситуации с указанием целей задачи в заданной ситуации. Задача – это всегда конкретное указание на исходные данные.

Зная <исходные данные> найти <цели задачи>

Исходные данные – это обычно набор требований к значениям параметров. Например: возраст > 20; время: 2000

Этапы решения задач:

1. Идентификация области знаний
2. Идентификация начальных данных в базе знаний
3. Извлечение фрагмента базы знаний
4. Нахождение последовательности применений знаний, позволяющей найти решение задачи
5. Применение этой найденной последовательности в конкретных условиях
6. Проверка найденного решения

Программа, решающая задачи предметной области, называется ИС, если она автоматизирует выполнение первых четырех этапов решения.

Архитектура ИС.

1. Простая архитектура ИС включает в себя:

* СО (система общения) – способ общения с внешней средой.
* МПЗ (механизм приобретения знаний)
* МВ (механизм вывода)
* БД (база данных)
* БЗ (база знаний)

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

1. Модель многомерной ИС

|  |
| --- |
| Поверхностный **А** |
| Алгоритмический **В** |
| Когнитивный **С** |

Компоненты ИС, которые нужно создавать, связаны с характеристиками знаний, обрабатываемых в этих компонентах. Примером является – измерение:

* Уровня знаний

|  |
| --- |
| Неструктурированный **1** |
| Слабо структурированный **2** |
| Полностью структурированный **3** |

* Структурированность представления знаний

Изображение выглядит как диаграмма, План, Технический чертеж, Прямоугольник

Автоматически созданное описание

Каждая компонента архитектуры ИС имеет собственную память. Обычно организованную в форме иерархии. В каждой компоненте имеется собственная база знаний. Знания о том, как работать со знаниями в этом формате или с этими свойствами. Это знания о том, как работать в этой компоненте.

Есть области, в которых реализуются процессы обработки знаний. Процессы обработки знаний группируются в потоки. Поток – это сложная структура, состоящая в перемещении знаний между компонентами, а перемещать можно между соседними по одной координате, при этом в каждой компоненте связанности в системе выполняется некоторое преобразование знаков, то есть какая-то обработка производится.

Управление целями – это такая организация управления задач, при которой вся система поддерживает нахождение в определенном состоянии «(в состоянии устойчивости)» со способностью осуществления интеллектуальной деятельности. «6 компонентов обычно достаточно».

Еще одним компонентом ИС является процессы обработки знаний компонентов. Обычно они моделируются диаграммами операций (морфизмов) преобразования структур знаний.

Изображение выглядит как круг, диаграмма, снимок экрана, Шрифт

Автоматически созданное описание

ИД – исходные данные. F1 – анализ исходных данных на предмет, а нужно ли решать задачу, представляется ли задача актуальной для этих исходных данных. Дальше получается какое-то заключение. Попали в S2, обработали, получили. Дальше F2 – решение задачи. Установление возможности решения, проверили можем ли решать. Решаем, а дальше извлечение ответа.

Компоненты берутся из измерений. Компоненты архитектуры определяют соседство, то есть по одному свойству на одно значение меняется.

**Знания и их свойства. Достоинства/недостатки ИС.**

1. Знания – это виды ресурсов, применяемых для решения задач. С помощью знаний отображается содержание предметных областей. Знания – это какие-то законы, свойства, понятия, которые есть в предметной области. Определение понятия знания можно – это системный подход.

Можно перечислить свойства:

* Содержание и форма (обычно форма соответствует содержанию с точки зрения эргономичности).

Форма – формат представления знаний ассоциируется с понятием структуры.

Универсальные конструкции позволяют представлять знания в виде алгебраического выражения, в котором знание собирается как композиция других знаний.

((a ○ b) ○ (c ○ a)) ○ r

Изображение выглядит как линия, диаграмма, дизайн

Автоматически созданное описание

Содержание знаний – представление о возможности его использования. В качестве представления содержания применяется предикат вложения.

* Отделимость и связанность.

Каждое знание допускает независимое самостоятельное существование.

Знание допускает связывание в системы знаний и совместную обработку процессов решения задач. Как правило получаются серии знаний.

* Область определения и область значений

Область определения – ситуация, в которой знание может быть применено.

Область значений – результат применения, который выражается в изменении некоторых параметров или свойств содержания области знаний или базы знаний.

* Метазнания

Изображение выглядит как диаграмма, текст, Шрифт, круг

Автоматически созданное описание

Знания можно отнести к XML структуре. Знания относятся к предметной области. Это знания о предметной области. Метазнания – это знание о мире знаний.

* Неточность знаний

1. *Недостоверность*, когда знания имеют вероятностную природу. Исходные данные недостоверные, с какой-то вероятностью это проявляется. Вероятность возникает как в исходных данных, так и в самом правиле. В исходных данных, потому что у нас есть ошибки в измерениях, а в правилах из-за того, что мы не можем перечислить все параметры, которые на закон влияют.
2. *Нечетность* – знания, которые оперируют нечеткими понятиями. Нечеткое понятие – это свойство, которое присутствует всегда. Вот вероятность – это либо есть, либо нет, то есть есть какая-то частотность, а нечетность – это свойство присутствует всегда, но в разной степени.
3. *Многозначность* – ситуация, когда в одних и тех же условиях применяются разные знания, иногда противоречащие друг другу.

**Достоинства/недостатки ИС**

Достоинства:

1. Высокий уровень решения задач, так как в ИС используется знания и опыт квалифицированных специалистов.
2. Невысокая стоимость
3. Устойчивость решений
4. Универсальность условий применения

Недостатки:

1. Ограниченность. Ограниченность заключается в принятии ограниченного разнообразия знаний как правило в узкой предметной области
2. Есть способность к обобщению. «Обобщение – способ сформулировать уникальный закон»
3. Неспособность к абстрагированию и творчеству
4. Методы решения задач основаны на понятии следствия из имеющихся знаний и поэтому существенно опираются на перебор имеющихся знаний.

**Модель представления знаний**

Исторически при построении прикладных ИС применяются близкие (не значит совпадающие) по свойствам модели представления знаний.

1. Продукционные модели (модели правил)
2. Модели, основанные на семантических сетях
3. Логические модели (знания представляются в виде предложений логики первого порядка)

Существуют и другие подходы представления знаний, удобные в определенных случаях

**Атомарные продукционные системы**

Пусть задано множество A = {a1…an}, элементы которого называются атомами. Каждый из атомов может принимать одно из двух значений:

* Истинно
* Ложно

Продукциями называется выражения вида:

Где t1…tk+1 – атомы из A

Содержательно, продукция означает, что если t1…tk объявлены истинными, то истинным объявляется tk+1

Атомы t1…tk – посылки; tk+1 – заключение продукции

Если k = 0, то продукция называется аксиомой

Аксиома содержит атом, являющийся истинным по определению

Атомарной продукционной системой называется всякое конечное множество P = {}

1. Аксиома P обычно представляет начальные данные решаемых задач
2. Постановка задач:
   1. a-? (aA) – требуется установить истинность атома
   2. x-? (xA) – x – неизвестное, решением является множество атомов истинных для данной системы
3. Механизм вывода. Механизм решения задач

Возможны две основные схемы организации постановления задач

1. Прямой вывод
2. Обратный вывод

**Прямой вывод**

1. Пусть задана система атомарных продукций P (там все накидано и аксиомы, как правил это исходные данные, истинность которых известна и просто продукции). Составим два списка

* L1 – содержит атомы, являющиеся заключениями аксиом
* L2 – все остальные продукции

1. Рассмотрим список L2 – до тех пор, пока не встретится первая по порядку продукция, все посылки которой содержатся в L1, а заключения не содержатся в L1.
2. Если такой продукции нет, то механизм вывода завершает свою работу. Если продукция найдена, то добавляем ее заключение в L1.
3. Повторяем действие 2.

Механизм прямого вывода в атомарных продукционных системах всегда завершает свою работу. Это так, потому что на каждом шаге к L1 добавляется новый атом и множество атомов конечно, то количество итераций при работе механизма вывод ограниченно.

После этого задача a – ? получает ответ зависящий от содержимого L1. Если a L1, то a – истинно и выводимо.

Если a ∉ L1, то атом – не выводимый.

Ответом на задачу X – ? является весь список L1 (утверждение об истинности всех атомов из L1).

**Задача** (спросит на экзе)

Проверить, что если существует последовательность применений продукций, позволяющая в конце доказать истинность некоторого атома, то механизм прямого вывода объявляет атом истинным и добавляет в L1 обязательно. «(тут есть тонкость, что, если возьмем какой-то вывод, он может быть не единственным, там разный порядок применения продукций можно доказывать одно и то же. Например, есть какая-то итоговая продукция, которая зависит от двух атомов, мы ее можем доказывать с помощью двух продукций…)»

Замечание

Приведенная схема прямого вывода крайне неэффективна и как правило не применяется, поскольку допускает более эффективную реализацию, эффективную с точки зрения времени.

A = {a1…an} – продукционная система

P = {} – продукции

Представим продукционную систему в виде двудольного графа с двумя множествами вершин A⋃P

Если задана , то ей соответствует фрагмент графа вида:

Изображение выглядит как диаграмма, линия, зарисовка, График

Автоматически созданное описание

* 1. Вершины-атомы. Разметим вершины-атомы соответствующей аксиомы символом T, а каждой вершине продукции припишем количество посылок этой продукции k.
  2. Уменьшим разметки вершин продукций на количество атомов, которые объявлены истинными на шаге 1. (если у продукции один атом объявлен доказанным, то уменьшаем на 1).

Изображение выглядит как линия, диаграмма, График

Автоматически созданное описание

* 1. Составим список L – продукций, размеченных 0 (0 означает, что все посылки доказаны), а заключения не объявлены истинными (то есть L – все продукции, которые можно исполнять).
  2. Возьмем первый элемент списка L, разметим вершину атома заключения выбранной продукции символом T, то есть объявим его истинным. Для всех продукций, у которых aik+1 является посылкой, уменьшаем значение счетчика на 1.

Изображение выглядит как линия, диаграмма, Шрифт, дизайн

Автоматически созданное описание

Из рассматриваемых продукций добавим в L такие продукции, для которых число не доказываемых посылок равно нулю, а заключение еще не доказано, то есть заключение не размечено символом T и тд.

**Механизм обратного вывода**

Прямой вывод плох тем, что если база знаний большая и рассчитана на решение самых разных задач, то прямой вывод решает все задачи одновременно, он не думает о том, что он доказывает, он строит все следствия, которые может построить из исходных данных, то есть доказывает истинность любых атомов, для которых это можно сделать.

В основе механизма обратного вывода лежит конструкция дерева обратного вывода.

X – ? – решать все задачи вида a – ?, где a A

Атом постановки задачи припишем вершине корня дерева

Изображение выглядит как диаграмма, линия

Автоматически созданное описание

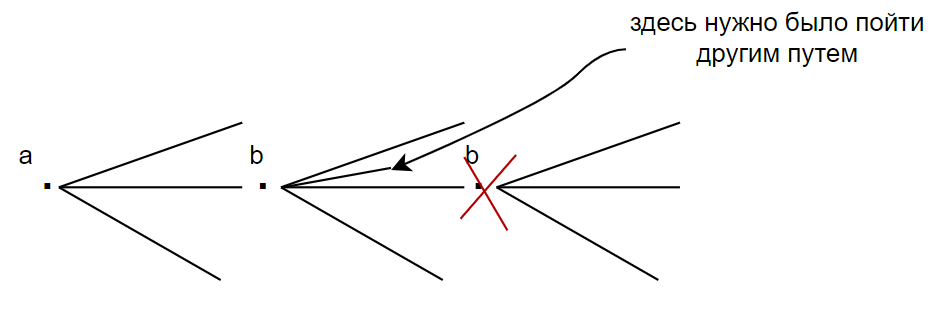
Потомками корневой вершины являются вершины, размеченные продукциями, для которых атом постановки задачи является заключением. Все продукции мы извлекаем и у продукций общее свойство: a – заключение для каждого из них.

Потомками каждой вершины первого яруса являются вершины, размеченные атомами посылками соответствующей продукции, каждая вершина первого яруса, размечена продукцией, а потомками этой вершины являются вершины, размеченные атомами посылок. Из этого следует, что у каждой вершины размеченной продукции потомков столько, сколько посылок и поскольку это дерево, если у двух продукций есть одинаковые посылки, то они представляются двумя разными вершинами, размеченными одинаково.

На третьем ярусе будут продукции (дерево и/или)

Для каждой вершины второго яруса постоим дерево обратного вывода для атома, приписанного этой вершине. Построение дерева заканчивается в вершинах, размеченных атомами, при выполнении одного из трех условий:

1. Атом, приписанный вершине, является аксиомой
2. Атом, приписанный вершине, не является заключением никакой продукции. То есть он не представлен как заключение аксиомы и нет других продукций, в которых этот атом является заключением. «То есть этот атом может быть посылкой, а заключением нет. Как правило аксиомы – исходные данные и когда есть традиционная база знаний, то обычно отдельно продукции и отдельно аксиомы. И когда исходные данные тасуются, тогда по отношению к другим продукциям может оказаться, что посылки и заключения могут быть только как аксиомы, но использоваться в качестве посылок в других продукциях».
3. Атом, приписанный вершине, встречается среди разметок вершин на пути из корня дерева в эту вершину



Дерево обратного вывода является конечным, так как оно содержит конечное множество вершин, которые можно оценить сверху.

Каждая вершина дерева имеет не больше чем max(m, n) потомков, поэтому при переходе с яруса на ярус количество вершин возрастает не более, чем в max(m, n) раз, поэтому, если глубина дерева (глубина – это число ярусов *l*, глубина – максимальная длина пути) *l* с ярусами 0…*l*, то количество вершин такого дерева не превосходит

Значение *l* < 2n, где n – количество атомов в системе.

Почему длина дерева не превосходит 2n? По правилу птичьих гнезд. Вершина продукции, вершины атомов чередуются. Если считать с нулевого яруса, то их будет не более чем 2n+1. Ярусы, размеченные атомами 0, 2, 4, 6… они будут встречаться n+1 раз, и n – размеченные продукциями. Больше не нужно, потому что по правилу птичьих гнезд у нас n разных атомов и n + 1 вершина, размеченная атомами, хотя бы одна точно повторится.

Проверить истинность атомов в корне дерева обратного вывода можно, если выполнить процедуру разметки вершин дерева как выводимых и невыводимых.

Разметим листья дерева как выводимые и невыводимые по следующим правилам:

1. Вершина, размеченная аксиомой – выводимая
2. Остальные – невыводимые