

# **Модели представления знаний в интеллектуальных системах**

# ОТЛИЧИЯ ЗНАНИЙ ОТ ДАННЫХ

*Данными* называют информацию фактического характера, описывающую объекты, процессы и явления предметной области, а также их свойства.

*Знания* описывают не только отдельные факты, но и взаимосвязи между ними, поэтому знания иногда называют структурированными данными

# Определения

- «**Знания** — это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области».
- «**Знания** — это хорошо структурированные данные или данные о данных, или метаданные».
- «**Знания** — формализованная информация, на которую ссылаются или используют в процессе логического вывода»

# Признаки классификаций знаний

1. ***По природе:*** декларативные, процедурные
2. ***Специальные языки для описания знаний (языки описания знаний):*** языки процедурного типа, языки декларативного типа
3. ***По способу приобретения знания:*** факты, эвристика
4. ***По типу представления знания:*** факты, правила, метазнания

# Факторы выбора модели данных

- **Однородность** представления приводит к упрощению механизма управления знаниями.
- **Простота** понимания важна для пользователей интеллектуальных систем и экспертов, чьи знания закладываются в ИИС.

- **Представление знаний** - структурирование знаний с целью формализации процессов решения задач в определенной проблемной области.
  
- **Модель представления знаний** - формализм, предназначенный для отображения статических и динамических свойств предметной области.
  
- В искусственном интеллекте основными моделями представления знаний являются:
  - **продукционные системы,**
  - **семантические сети,**
  - **фреймы,**
  - **формальные логические модели.**

# Продукционные правила

В этой модели знания представляются в виде предложений типа:

***Если (условие), то (действие).***

Под ***условием*** понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под ***действием*** - действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

Любое продукционное правило, содержащееся в БЗ, состоит из двух частей: антецедента и консеквента.

*Антецедент* – это посылка правила (условная часть), состоит из элементарных предложений, соединенных логическими связками И, ИЛИ.

*Консеквент* (заключение) включает одно или несколько предложений, которые выражают либо некоторый факт, либо указание на определенное действие, подлежащее исполнению.

Продукционные правила записываются в виде

АНТЕЦЕНДЕНТ → КОНСЕКВЕНТ



# Пример продукционных правил

## Правило 1:

**Если** топливо поступает в двигатель и двигатель вращается,  
**то** проблема в свечах зажигания.

## Правило 2:

**Если** двигатель не вращается и фары не горят,  
**то** проблема в аккумуляторе или проводке.

## Правило 3:

**Если** двигатель не вращается и фары горят,  
**то** проблема в стартере.

## Правило 4:

**Если** в баке есть топливо и топливо поступает в карбюратор,  
**то** топливо поступает в двигатель.

В общем случае продукционная система  
включает следующие компоненты:

- *базу данных, содержащую множество фактов;*
- *базу правил, содержащую набор продукций;*
- *интерпретатор (механизм логического вывода) или правила работы с продукциями.*

Продукционные системы делят на два типа — с прямыми и обратными выводами.

- При прямом выводе рассуждение ведется от данных к гипотезам.
- При обратном производится поиск доказательства или опровержения некоторой гипотезы (от цели к данным).

Часто используются комбинации прямой и обратной цепи рассуждений.

Например, имеется фрагмент базы знаний из двух правил:

71: Если "отдых - летом" и "человек - активный", то "ехать в горы",

72: Если "любит солнце", то "отдых летом",

Пусть в систему поступили данные :

"человек активный" и "любит солнце"

# ПРЯМОЙ ВЫВОД

Необходимо, исходя из данных, получить ответ.

## 1-й проход.

- Шаг 1. Правило П1, не работает (не хватает данных "**отдых - летом**").
- Шаг 2. Правило П2, работает, в базу поступает факт "**отдых - летом**".

## 2-й проход.

- Шаг 3. Правило П3, срабатывает, активируется цель "**ехать в горы**", которая и выступает как совет, который дает ЭС.

# Обратный вывод

Необходимо подтвердить выбранную цель при помощи имеющихся правил и данных.

## 1-й проход.

- Шаг 1. Цель - **"ехать в горы"**:

пробуем П1 - данных, **"отдых - летом"** нет, они становятся новой целью, и ищется правило, где она в правой части.

- Шаг 2. Цель **"отдых - летом"**:

правило П2 подтверждает цель и активирует ее.

## 2-й проход.

- Шаг 3. Пробуем П1, подтверждается искомая цель.

# Преимущества продукций

- модульность;
- наглядность;
- единообразие структуры (основные компоненты продукционной системы могут применяться для построения интеллектуальных систем с различной проблемной ориентацией);
- естественность (вывод заключения в продукционной системе во многом аналогичен процессу рассуждений эксперта);
- легкость внесения дополнений и простота механизма логического вывода;
- гибкость родовидовой иерархии понятий, которая поддерживается только как связи между правилами (изменение правила влечет за собой изменение в иерархии).

# Недостатки продукций

- отличие от структур знаний, свойственных человеку;
- этот процесс трудно поддается управлению;
- сложно представить родовидовую иерархию понятий.
- неясность взаимных отношений правил;
- сложность оценки целостного образа знаний;
- низкая эффективность обработки знаний.



Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

*Определение 1.* Смешанные продукционные правила представляют собой модифицированную модель нечетких продукционных правил [8, 9].

$$R_k : IF \ x_1 = A_{k1} \ AND \ x_2 = \tilde{A}_{k2} \ AND \ \dots \ AND \ x_n = \tilde{A}_{kn}([x_{k1min}, x_{k1max}]), \ THEN \ y = B_k \ [Rung_k],$$

где  $x_i$  – входные переменные,  $A_i, i = 1, \dots, n$ ;  $A_i$  – четкие значения переменных;  $\tilde{A}_i = \{x_i, \mu_{\tilde{A}_i}(x_i)\}$  – нечеткие значения переменных,  $\mu_{\tilde{A}_i}(x_i)$  – функция принадлежности;  $\tilde{A}_i([x_{imin}, x_{imax}])$  – значения переменных представлены в виде интервала, где  $x_{imin}, x_{imax}$  – минимальное и максимальное значение  $i$ -й переменной; которые могут принимать только числовые значения;  $y$  – выходная переменная,  $B$  – четкое и/или нечеткое значение выходной переменной;  $Rung$  – значимость (оценка) правила,  $Rung \in \{1, 2, \dots, c\}$ ,  $c$  – количество СПП,  $c \in [0, 1]$ .

*Определение 2.* Смешанные продукционные правила называются избыточными, если два правила полностью дублируют друг друга.

Таким образом, БЗ будет являться избыточной, если будут выполняться следующие условия:

1) Отсутствуют полные дубликаты СПП, т.е. в БЗ отсутствуют правила с одинаковыми предусловиями и заключениями. Для наглядности правил представим их в виде ориентированного графа, где вершинам соответствуют входные и выходные переменные, а ребрам – связи между переменными (рисунок 1):

$$R_1: IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n} ([x_{1nmin}, x_{1nmax}]), THEN y = B_1.$$

$$R_2: IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n} ([x_{1nmin}, x_{1nmax}]), THEN y = B_1.$$

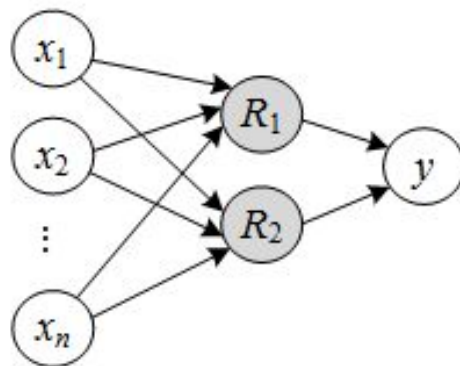


Рисунок 1. Полные дубликаты СПП

Для решения данной проблемы необходимо удалить все СПП дублирования, кроме одного.

2) Отсутствуют неполные дубликаты (рисунок 2):

$$R_1: \text{IF } x_1 = A_{11} \text{ AND } x_2 = \tilde{A}_{12}, \text{ THEN } y = B_1.$$

$$R_2: x_2 = \tilde{A}_{12} \text{ AND } \dots \text{ AND } x_n = \tilde{A}_{2n} ([x_{2n\min}, x_{2n\max}]), \text{ THEN } y = B_1.$$

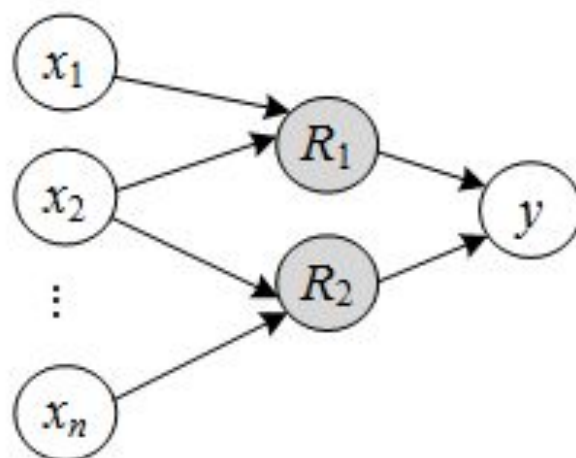


Рисунок 2. Неполные дубликаты СПП



3) Отсутствует включение предусловий (inclusive duplicates), т.е. в БЗ не существует правил с одинаковыми заключениями, таких, что предусловие некоторого правила является подмножеством предусловия другого правила (рисунок 3):

$$R_1 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12}, THEN y = B_1,$$

$$R_2 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{2n}([x_{2nmin}, x_{2nmax}]), THEN y = B_1$$

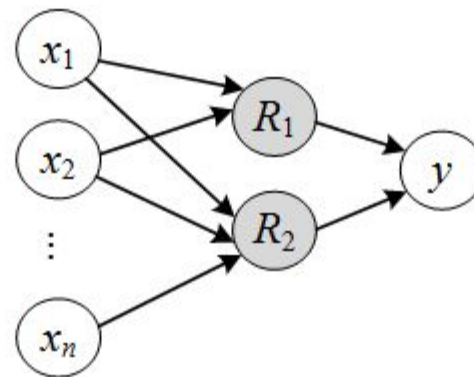


Рисунок 3. Включение предусловий

Отметим, что единого решения для решения данных проблем, а именно исправления условий 2 и 3 не существует, поэтому конкретное решение о способе улучшения БЗ и исправлении структурных ошибок принимает эксперт в каждом конкретном случае.

*Определение 3.* Смешанные продукционные правила называются циклическими, если цепочка правил в наборе циклических правил образует цикл (рисунок 4), т.е. последовательность правил, в которой предусловия всех правил, кроме последнего, содержатся в заключение следующего правила в последовательности, а предусловия правила содержатся в заключение первого правила последовательности.

$$R_1 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12}, THEN y = B_1,$$

$$R_2 : IF x_3 = B_1 AND x_4 = \tilde{A}_{23}, THEN y = B_2,$$

$$R_3 : IF x_5 = B_2 AND x_6 = \tilde{A}_{32}, THEN y = \tilde{A}_{12}.$$

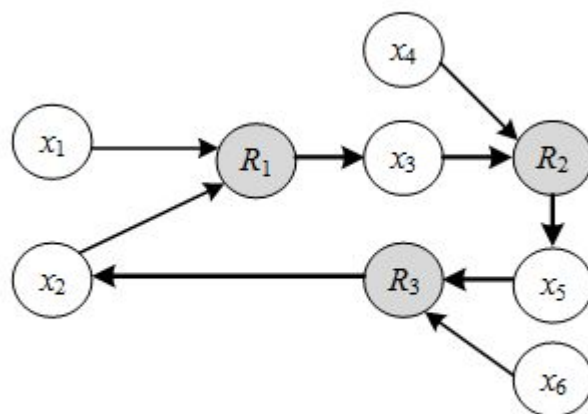


Рисунок 4. Циклические СПП

Таким образом, в БЗ отсутствуют циклические правила, если выполняются следующие условия :

1) Отсутствует прямой цикл, т.е. в БЗ отсутствуют правила, в которых в предусловии и заключении присутствуют термы одной лингвистической переменной (ЛП).

2) Отсутствует не прямой цикл, т.е. в БЗ отсутствуют правила, при которых ЛП определяют ЛП, от которых они зависят.

*Определение 4.* Смешанные продукционные правила называются противоречивыми, если они имеют одинаковые предусловия (одинаковые значения входных переменных), но различные заключения (различные значения выходных переменных) (рисунок 5):

$$R_1 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n}([x_{1nmin}, x_{1nmax}]), THEN y = B_1,$$

$$R_2 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n}([x_{1nmin}, x_{1nmax}]), THEN y = B_2.$$

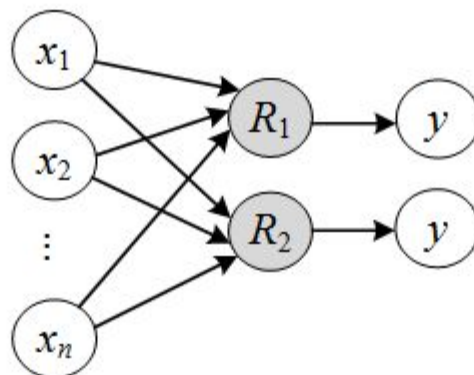


Рисунок 5. Противоречивые СПП

Из рисунка 5 видно, что правила  $R_1$  и  $R_2$  являются противоречивыми, поскольку одно и то же предусловие ( $A_{11}$  и  $\tilde{A}_{12}, \dots, \tilde{A}_{1n}([x_{1nmin}, x_{1nmax}])$ ) активизирует разные выходные переменные (термы, значения  $B_1$  и  $B_2$ ).



*Определение 5.* База знаний является неполной, если в ней отсутствуют СПП, приводящие к определенным значениям целевых параметров [11].

В качестве ошибки неполноты могут выступать изолированные входные и выходные переменные, т.е. отсутствуют правила, ведущие к ним (рисунок 6).

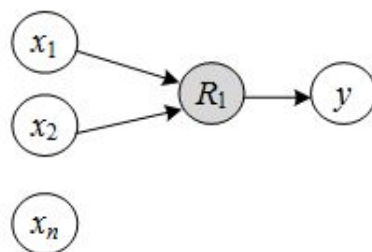


Рисунок 6. Изолированные переменные

Из рисунка видно, что переменная  $x_n$  является изолированной. Для решения данной проблемы эксперту предлагается выяснить следующее: если переменная является важной входной либо выходной переменной, то необходимо добавить правило, иначе переменная может быть удалена из БЗ.

*Определение 6.* База знаний является связной, если для любой пары смежных СПП значение выходной ЛПП тоже смежное, т.е. предусловия этих правил отличаются только одним из  $n$  подусловием и в этих подусловий используется одна и та же ЛПП с различными значениями:

$$R_1: \text{IF } x_1 = A_{11} \text{ AND } x_2 = \tilde{A}_{12} \text{ AND } \dots \text{ AND } x_n = \tilde{A}_{1n} ([x_{1n\min}, x_{1n\max}]), \text{ THEN } y = B_1.$$

$$R_2: \text{IF } x_1 = A_{21} \text{ AND } x_2 = \tilde{A}_{12} \text{ AND } \dots \text{ AND } x_n = \tilde{A}_{1n} ([x_{1n\min}, x_{1n\max}]), \text{ THEN } y = B_2.$$

Приведенные СПП являются смежными, поскольку значения  $A_{11}$  и  $A_{21}$  входной переменной  $x_1$  смежные.

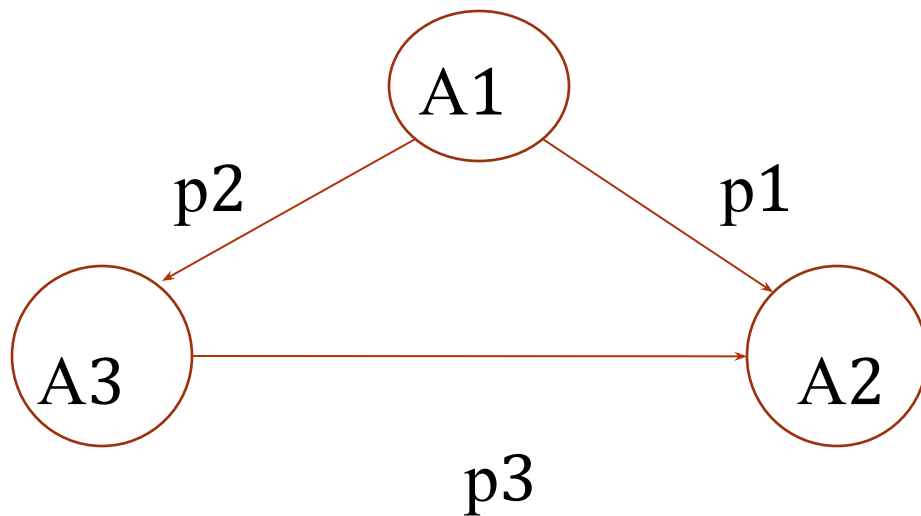
# Семантическая сеть

Термин *семантическая* означает смысловая, а сама семантика - это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т.е. наука, определяющая смысл знаков.

Более наглядными являются языки, опирающиеся на сетевую модель представления знаний. В основе такой модели лежит идея о том, что любые знания можно представить в виде совокупности объектов (понятий) и связей (отношений) между ними.

*Семантическая сеть* описывает знания в виде ориентированного графа. В качестве *вершин* сети выступают понятия, факты, объекты, события и т. п., а в качестве *дуг* сети — отношения, которыми вершины связаны между собой. Семантическая сеть является представлением структуры памяти человека.

Например, «*программист сел за компьютер и отладил программу*». Объектами являются: программист (A1), компьютер (A2), программа (A3). Объекты связаны отношениями: сел за компьютер (p1), отладил (p2), загружена в компьютер программа (p3).



*Понятиями* обычно выступают абстрактные или конкретные объекты, а *отношения* - это связи типа:

- "это" ("is"),
- "имеет частью" ("has part"),
- "принадлежит",
- "любит".

## Классификация семантических сетей:

Можно предложить несколько классификаций семантических сетей, связанных с типами отношений между понятиями.

*По количеству типов отношений:*

Однородные (с единственным типом отношений).

Неоднородные (с различными типами отношений).

*По типам отношений:*

Бинарные (в которых отношения связывают два объекта).

N-арные (в которых есть специальные отношения, связывающие более двух понятий).

Самыми распространенными являются следующие типы отношений:

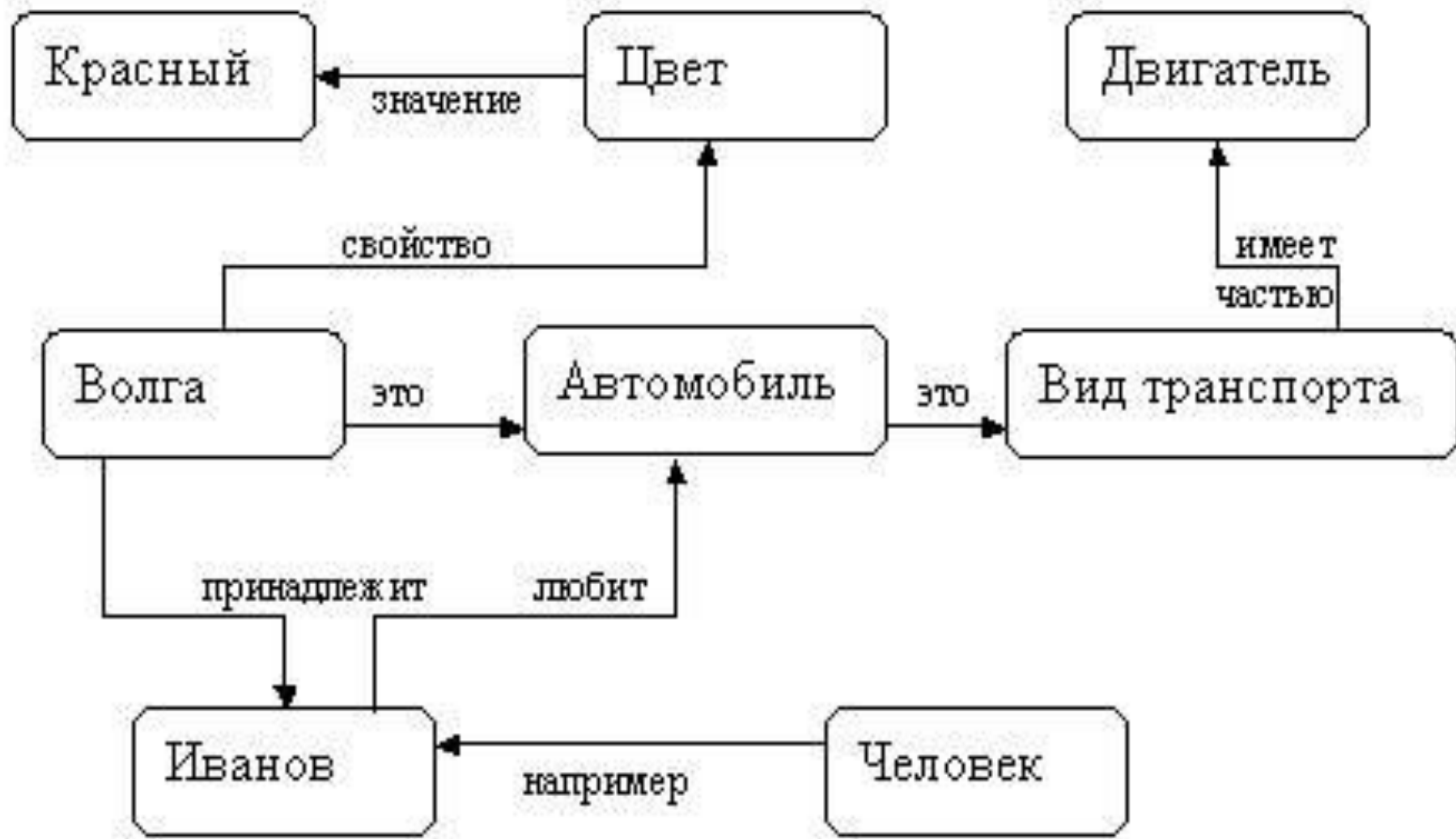
- **быть** элементом класса, то есть объект входит в состав данного класса (ВАЗ 2106 является автомобилем);
- **иметь** свойства, то есть задаются свойства объектов (жираф имеет длинную шею);
- **иметь** значение, то есть задается значение свойств объектов (человек может иметь двух братьев);
- **является** следствием, то есть отражается причинно-следственная связь (астеническое состояние является следствием перенесенного простудного заболевания).

Более детально можно выделить следующие отношения:

- связи типа "часть-целое" ("класс-подкласс", "элемент-множество" и т.п.);
- функциональные связи (определяемые обычно глаголами "производит", "влияет" ...);
- количественные (больше, меньше, равно...);
- пространственные (далеко от, близко от, за, под, над...);
- временные (раньше, позже, в течение...);
- атрибутивные связи (*иметь свойство*, *иметь значение*...);
- логические связи (и, или, не);
- отношения «близости»;
- отношения «сходства-различия»;
- отношения «причина-следствие» и др.



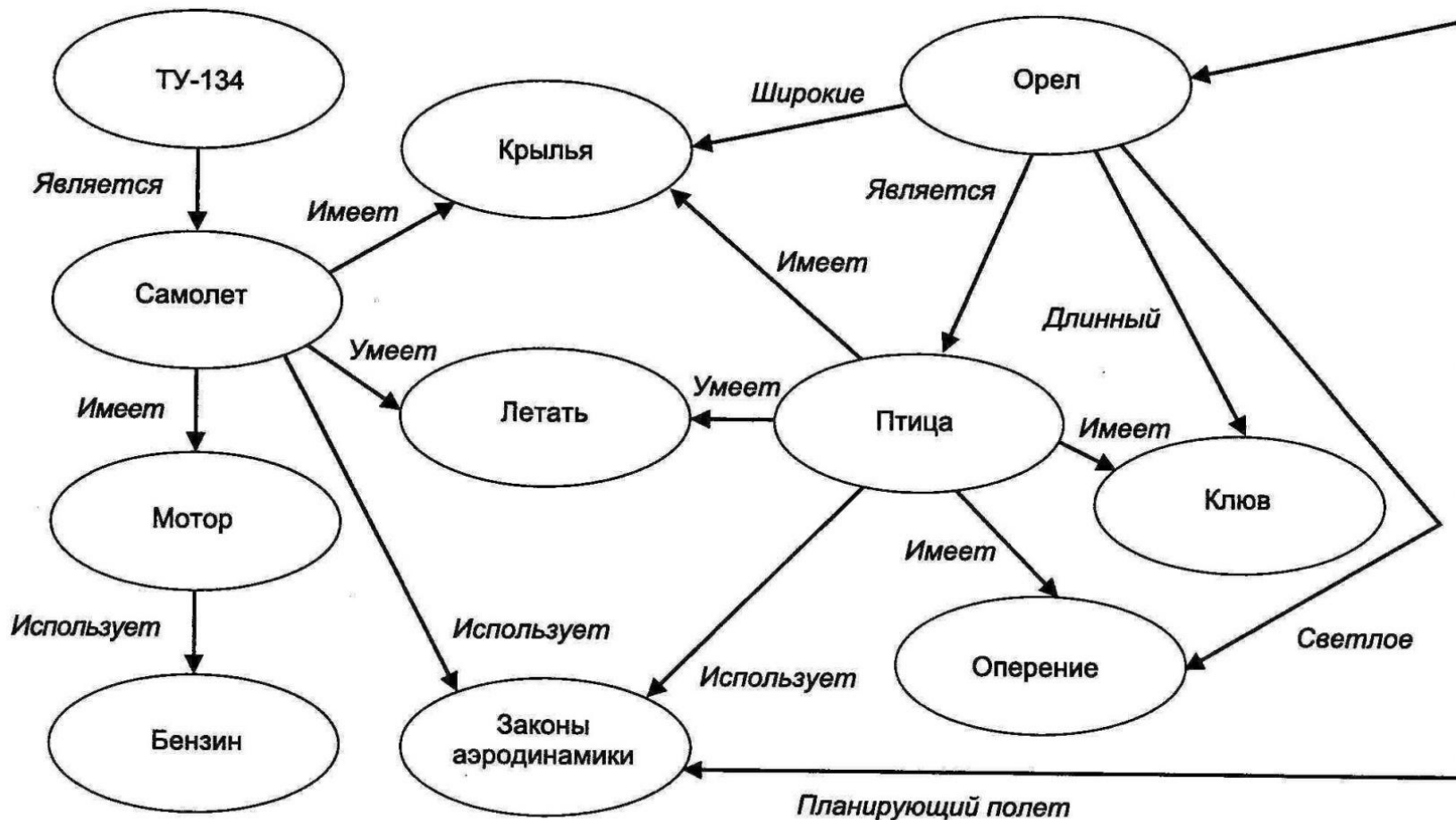
Пример семантической сети. В качестве вершин понятия: Человек, Иванов, Волга. Автомобиль, Вид транспорта. Двигатель.



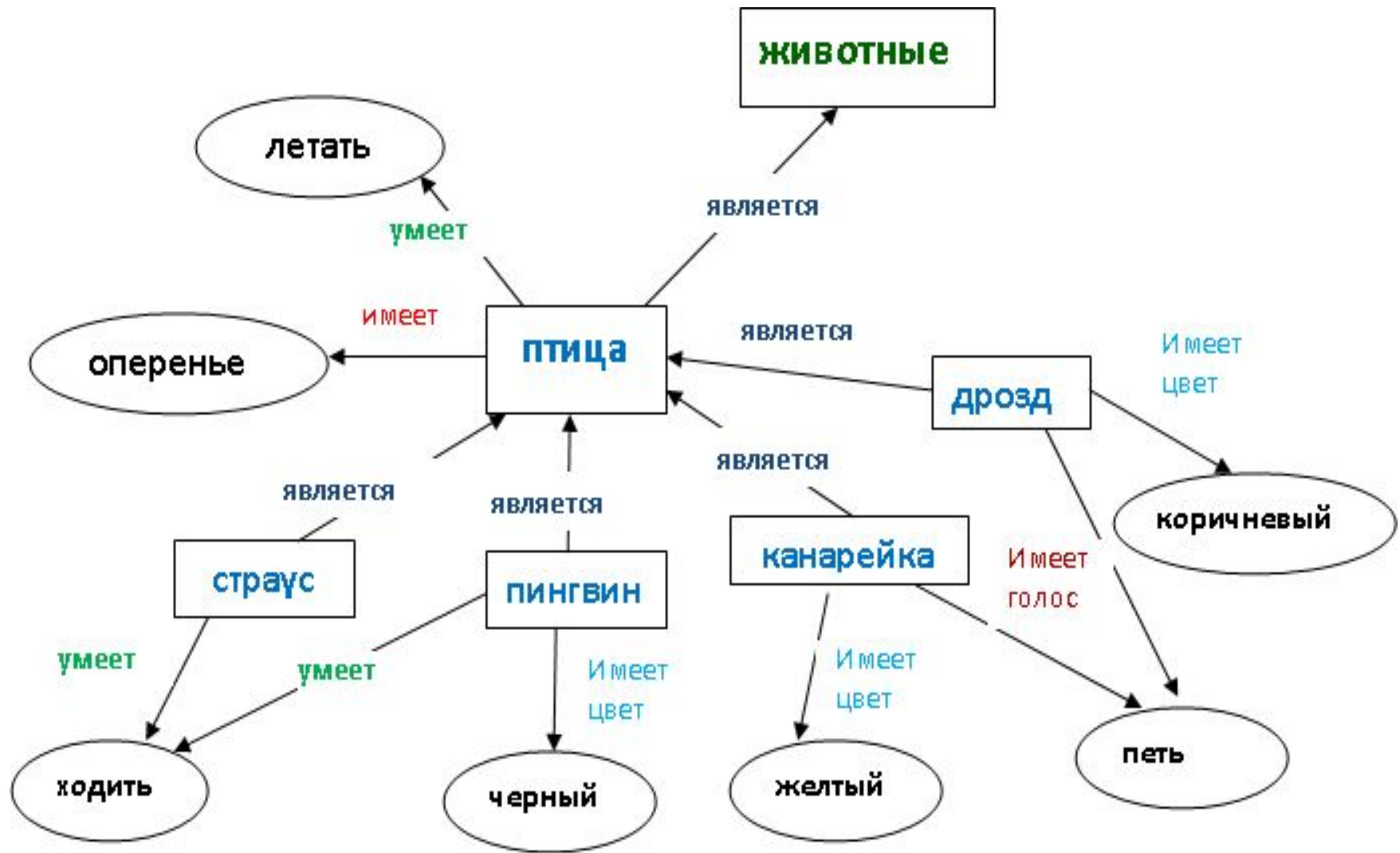
Преимущество этой модели - в соответствии современным представлениям об организации долговременной памяти человека.

Недостаток модели - сложность поиска вывода на семантической сети.

# Семантическая сеть, показывающая взаимоотношения птиц и самолета



# Фрагмент семантической сети, описывающей птиц



# Фреймовая модель

Фреймовая модель основана на концепции Марвина Мински (Marvin Minsky) – профессора Массачусетского технологического института, основателя лаборатории искусственного интеллекта, автора ряда фундаментальных работ. Фреймовая модель представляет собой систематизированную психологическую модель памяти человека и его сознания.

**Фреймом** (англ. frame – рамка, каркас) называется структура данных для представления некоторого концептуального объекта.

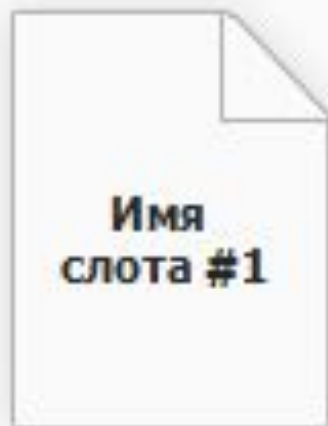
Информация, относящаяся к фрейму, содержится в составляющих его слотах.

**Слот** (англ. slot – щель, прорезь) может быть терминальным (листом иерархии) или представлять собой фрейм нижнего уровня.

Фрейм имеет имя, служащее для идентификации описываемого им понятия, и содержит ряд описаний – слов, с помощью которых определяются основные структурные элементы этого понятия. Слот может содержать не только конкретное значение, но и имя процедуры, вычисляющей это значение по заданному алгоритму.

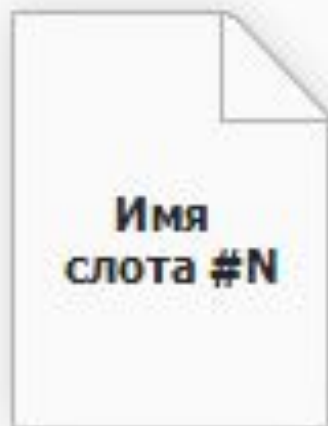
Фреймовые системы подразделяются на *статические* и *динамические*. Динамические допускают изменение фреймов в процессе решения задачи.

# Каждый фрейм, состоит из произвольного числа слотов



Имя фрейма

Указатель наследования	Указатель атрибута	Значение слота	Демон
------------------------	--------------------	----------------	-------



Указатель наследования	Указатель атрибута	Значение слота	Демон
------------------------	--------------------	----------------	-------

- **имя фрейма** – это идентификатор, присваиваемый фрейму. Фрейм должен иметь имя, единственное в данной фреймовой модели (уникальное имя);
- **имя слота** – это идентификатор, присваиваемый слоту. Слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота, но в некоторых случаях оно может иметь специфический смысл;
- **указатель наследования** – только для фреймовых моделей иерархического типа; они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня;
- **указатель атрибутов** – указатель типа данных слота. К таким типам относятся: `FRAME` (указатель), `INTEGER` (целое), `REAL` (вещественное), `BOOL` (булево), `LISP` (присоединенная процедура), `TEXT` (текст), `LIST` (список), `TABLE` (таблица), `EXPRESSION` (выражение) и другие;
- **значение слота** – значение, соответствующее типу данных слота и удовлетворяющее условиям наследования;
- **демон** – процедура, автоматически запускаемая при выполнении некоторого условия. Демоны запускаются при обращении к конкретному слоту фреймовой модели.



Фрейм можно представить:

а) в виде цепочки:

Фрейм: <слот 1>, <слот 2>, ... , <слот N>.

или более подробно, как список свойств:

Имя ФРЕЙМА: <имя 1-го слота: значение 1-го слота>, <имя 2-го слота: значение 2-го слота>, <...>, <имя N-го слота: значение N-го слота>.

б) в виде таблицы:

<b>Имя фрейма</b>	
Имя слота 1	Значение слота 1
Имя слота 2	Значение слота 2
Имя слота N	Значение слота N

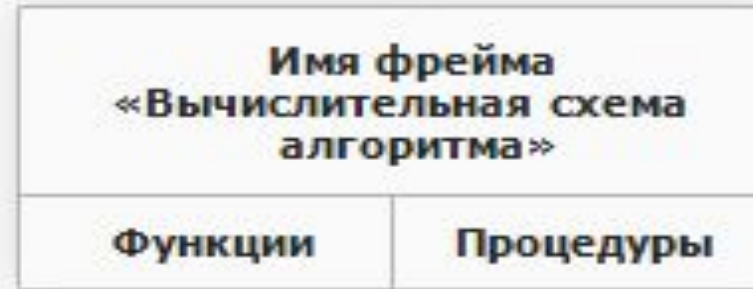
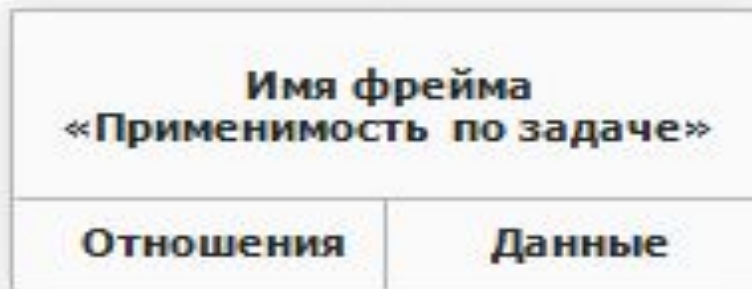
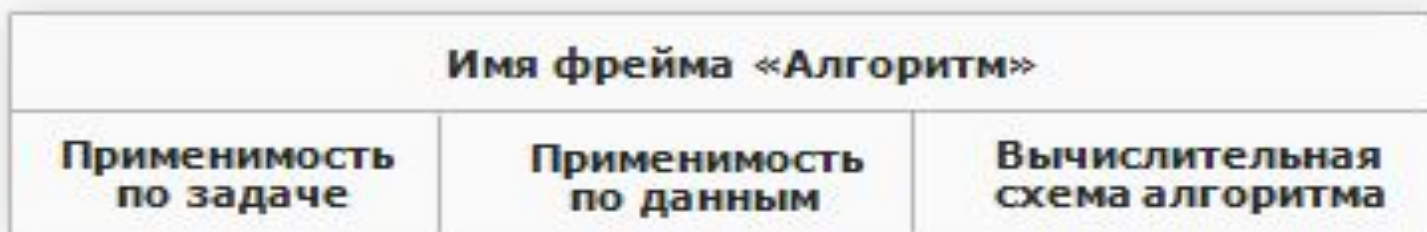
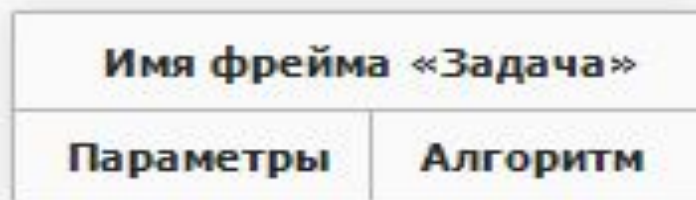
# Пример фрейма

## РУКОВОДИТЕЛЬ

Имя слота	Значение слота	Тип значения слота
Имя	Иванов И.И.	Строка символов
Дата_рождения	01.01.1965	Дата
Возраст	Age(дата, дата_рождения)	Процедура
Специальность	Юрист	Строка символов
Отдел	Отдел кадров	Строка символов
Зарплата	50000	Число
Адрес	Дом_адрес	Фрейм

Фреймы образуют иерархию. Иерархия во фреймовых моделях порождает единую многоуровневую структуру, описывающую либо объект, если слоты описывают только свойства объекта, либо ситуацию или процесс, если отдельные слоты являются именами процедур, присоединенных к фрейму и вызываемых при его актуализации.

# Пример фреймовой модели иерархического типа



# Фрагмент фрейма, описывающий гостиничный номер



**Типы фреймов.** Различают фреймы-прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных.

*Фрейм-прототип (протофрейм) – это фрейм, содержащий знания о самом понятии.*

Например, фрейм понятия «Битва» можно изобразить следующим образом:

<b>Битва</b>	
Субъект (кто?)	X1
Объект (с кем?)	X2
Место (где?)	X3
Время (когда?)	X4
Результат	X5

В этом фрейме указаны имена слотов (субъект, объект и т. д.), но вместо их значений стоят переменные (*X1, X2* и т. д.).



**Фрейм-экземпляр** (экзофрейм) - это фрейм, содержащий конкретное описание понятия или знания.

В примере, наверное, основными для фрейма "Битва" можно считать слоты с именами "субъект" и "объект". Битву осуществляет субъект  $X1$  с объектом  $X2$  в месте  $X3$  во время  $X4$ , при этом получается результат  $X5$ . Подставляя вместо всех переменных конкретные значения, получим конкретный факт-описание:

<b>Куликовская битва</b>	
Субъект (кто?)	Князь Дмитрий
Объект (с кем?)	Хан Мамай
Место (где?)	Куликово поле
Время (когда?)	Утром в сентябре 1380 года
Результат	Победа князя Дмитрия

Фрейм является *простым*, если он не содержит в себе других фреймов. *Сложный (составной)* фрейм содержит в себе два и более фрейма, и по существу представляет сеть фреймов.

Фрейм позволяет отобразить все многообразие знаний о мире через:

- фреймы – *структуры*, использующиеся для обозначения объектов и понятий (заем, залог, вексель);
- фреймы – *роли* (менеджер, кассир, клиент);
- фреймы – *сценарии* (банкротство, собрание акционеров, празднование именин);
- фреймы – *ситуации* (тревога, авария, рабочий режим устройства) и др.

***Фреймовая модель представления знаний*** – это модель, в которой структура знаний предметной области формализуется в виде совокупности взаимосвязанных фреймов, описывающих объекты, а свойства этих объектов и факты, относящиеся к ним, описываются в структурных элементах фрейма.



- Состав фреймов и слотов в каждой конкретной фреймовой модели может быть разным, однако в рамках одной системы целесообразно единое представление для устранения лишнего усложнения.
- Разнотипные объекты или объекты, соответствующие концепции «множественности миров», заключающейся, к примеру, в том, что лошадь – животное бескрылое для одного (реального) мира и одновременно крылатое (Пегас в мифическом мире) для другого, могут описываться отличающимися друг от друга фреймами.
- В целом фреймовая модель допускает представление всех свойств декларативных и процедурных знаний. Глубина вложенности слотов во фрейме (число уровней) зависит от предметной области и языка, реализующего модель.

# Логическая модель представления знаний

*Логическая модель* представляет собой формальную систему — некоторое логическое исчисление как правило, исчисление предикатов первого порядка, когда предметная область или задача описывается в виде набора аксиом.

Все знания о предметной области описываются в виде формул этого исчисления или правил вывода. Описание в виде формул дает возможность представить декларативные знания, а правила вывода — процедурные знания.

*Предикат* в исчислении предикатов специальный знак, отражающий определенное отношение между конечным множеством сущностей - аргументов. Предикат первого порядка имеет два состояния - истина и ложь.

Рассмотрим в качестве примера знание: "Когда температура в печи достигает  $120^\circ$  и прошло менее 30 минут с момента включения печи, давление не может превосходить критическое. Если с момента включения печи прошло более 30 мин, то необходимо открыть вентиль №2".

Логическая модель представления этого знания имеет вид:

$$P(p = 120) \wedge T(t < 30) \rightarrow (D < D_{кр});$$

$$P(p = 120) \wedge T(t > 30) \Rightarrow F(\text{№}2).$$

В этой записи использованы символы логического следования ( $\Rightarrow$ ).

$P(p = 120)$  — предикат, становящийся истинным, когда температура достигает  $120^\circ$ ;

$T(t < 30)$  — предикат, остающийся истинным в течение 30 мин с начала процесса;

$T(t > 30)$  — предикат, становящийся истинным по истечении 30 мин с начала процесса;

$D < D_{кр}$  — утверждение о том, что давление ниже критического;

$F(\text{№}2)$  — команда открыть вентиль №2.

логические связки И ( $\wedge$ ),  
импликации ( $\rightarrow$ )  
логического следования ( $\Rightarrow$ ).



Рассмотрим в качестве примера знание: "Когда температура в печи достигает  $120^\circ$  и прошло менее 30 минут с момента включения печи, давление не может превосходить критическое. Если с момента включения печи прошло более 30 мин, то необходимо открыть вентиль №2".

Логическая модель представления этого знания имеет вид:

$$P(p = 120) \wedge T(t < 30) \rightarrow (D < D_{кр});$$

$$P(p = 120) \wedge T(t > 30) \Rightarrow F(\text{№2}).$$

В этой записи использованы символы логического следования ( $\Rightarrow$ ).

$P(p = 120)$  — предикат, становящийся истинным, когда температура достигает  $120^\circ$ ;

$T(t < 30)$  — предикат, остающийся истинным в течение 30 мин с начала процесса;

$T(t > 30)$  — предикат, становящийся истинным по истечении 30 мин с начала процесса;

$D < D_{кр}$  — утверждение о том, что давление ниже критического;

$F(\text{№2})$  — команда открыть вентиль №2.

логические связки И ( $\wedge$ ),  
импликации ( $\rightarrow$ )  
логического следования ( $\Rightarrow$ ).

Языки представления знаний логического типа широко использовались на ранних стадиях развития интеллектуальных систем, но вскоре были вытеснены (или, во всяком случае, сильно потеснены) языками других типов. Объясняется это громоздкостью записей, опирающихся на классические логические исчисления. При формировании таких записей легко допустить ошибки, а поиск их очень сложен. Отсутствие наглядности, удобочитаемости (особенно для тех, чья деятельность не связана с точными науками) затрудняло распространение языков такого типа.

Исчисление предикатов 1-го порядка в промышленных экспертных системах практически не используется. Эта логическая модель применима в основном в исследовательских «игрушечных» системах, так как предъявляет очень высокие требования и ограничения к предметной области.