

Лекция 3

Модели представления знаний в интеллектуальных системах

Данные - это отдельные факты, характеризующие объекты, процессы и явления в предметной области, а также их свойства.

При обработке на ЭВМ данные трансформируются, условно проходя следующие этапы:

- данные как результат измерений и наблюдений;
- данные на материальных носителях информации (таблицы, протоколы, справочники);
- модели (структуры) данных в виде диаграмм, графиков, функций;
- данные в компьютере на языке описания данных;
- базы данных на машинных носителях.



Знания связаны с данными, основываются на них, но представляют результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе выполнения какой-либо практической деятельности. Они получают эмпирическим путем.

Знания - это выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой области.

При обработке на ЭВМ знания трансформируются аналогично данным:

- знания в памяти человека как результат мышления;
- материальные носители знаний (учебники, методические пособия);
- **поле знаний** - условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих;
- знания, описанные на языках представления знаний (продукционные языки, семантические сети, фреймы);
- **базы знаний.**

- **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 *продукционных системах* знания представляются в виде совокупности специальных информационных единиц, имеющих следующую структуру:

Имя продукции: «сфера применения»

Предусловие

Условие ядра

Если А, то В

Постусловие.

Имя сферы указывает ту предметную область, к которой относятся знания, зафиксированные в данной продукции. Предусловия устанавливают на множестве правил из интересующей сферы некоторый порядок, приоритет их использования. Условия определяют возможность применения того или иного правила. Ядро продукции «Если А, то В» описывает преобразование, которое составляет суть продукционного правила, где А и В могут иметь разные значения. Постусловие говорит о том, что надо делать, когда данное продукционное правило сработало.

Например:

47: Интерпретация результатов тестирования

Использовать в первую очередь

Шкала «лжи» $L < 70$ T-баллов

Если (шкала ошибок F – шкала коррекции K) < -11 ,

то вывести сообщение: «Результаты
тестирования недостоверны»

Закончить интерпретацию результатов

Примеры простых продукций

- «Если сверкнет молния, то гремит гром».
- «Если в доме вспыхнул пожар, то вызывайте по телефону 01 пожарную команду».
- «Если в путеводителе указано, что в городе есть театр, то надо пойти туда».

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.*

Производственные 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.

Продукции выгодны для выражения знаний, которые могут принимать форму перехода между состояниями:

- ситуация → действие;
- посылка → заключение;
- причина → следствие.

Преимущества продукций

- модульность;
- наглядность;
- единообразиие 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.

Имеется большое число программных средств, реализующих продукционный подход (язык OPS 5; "оболочки" или "пустые" ЭС - EXSYS, ЭКСПЕРТ; инструментальные системы СПЭИС и др.), также промышленных ЭС на его основе (ФИАКР) и др.

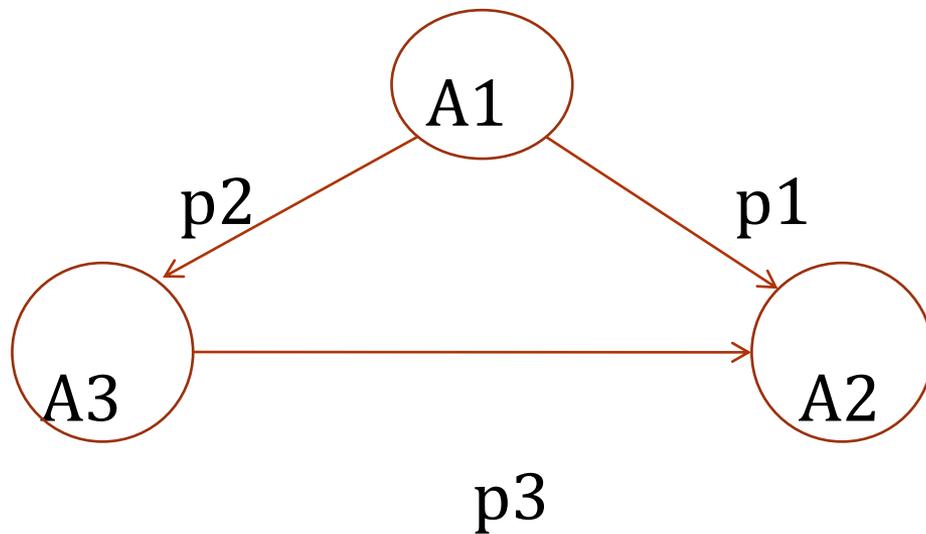
Семантическая сеть

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).



Основная идея моделирования при помощи семантических моделей заключается в том, что модель представляет данные о реальных объектах и связях между ними прямым способом, что облегчает доступ к знаниям. Начиная движение от некоторого понятия по дугам отношений, можно достичь других понятий.

Семантическая сеть как модель наиболее часто используется для представления декларативных знаний. С помощью этой модели реализуются такие свойства системы знаний, как интерпретируемость и связность, в том числе по отношениям IS-A и PART-OF. За счет этих свойств семантическая сеть позволяет снизить объем хранимых данных, обеспечивает вывод умозаключений по ассоциативным связям.

Одной из первых известных моделей, основанных на семантической сети, является **TLC-модель** (Teachable Language Comprehender – доступный механизм понимания языка), разработанная Куиллианом в 1968 году.

Модель использовалась для представления семантических отношений между концептами (словами) с целью описания структуры долговременной памяти человека в психологии.

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

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

Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс - элемент класса;
- свойство - значение;
- пример элемента класса.

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:

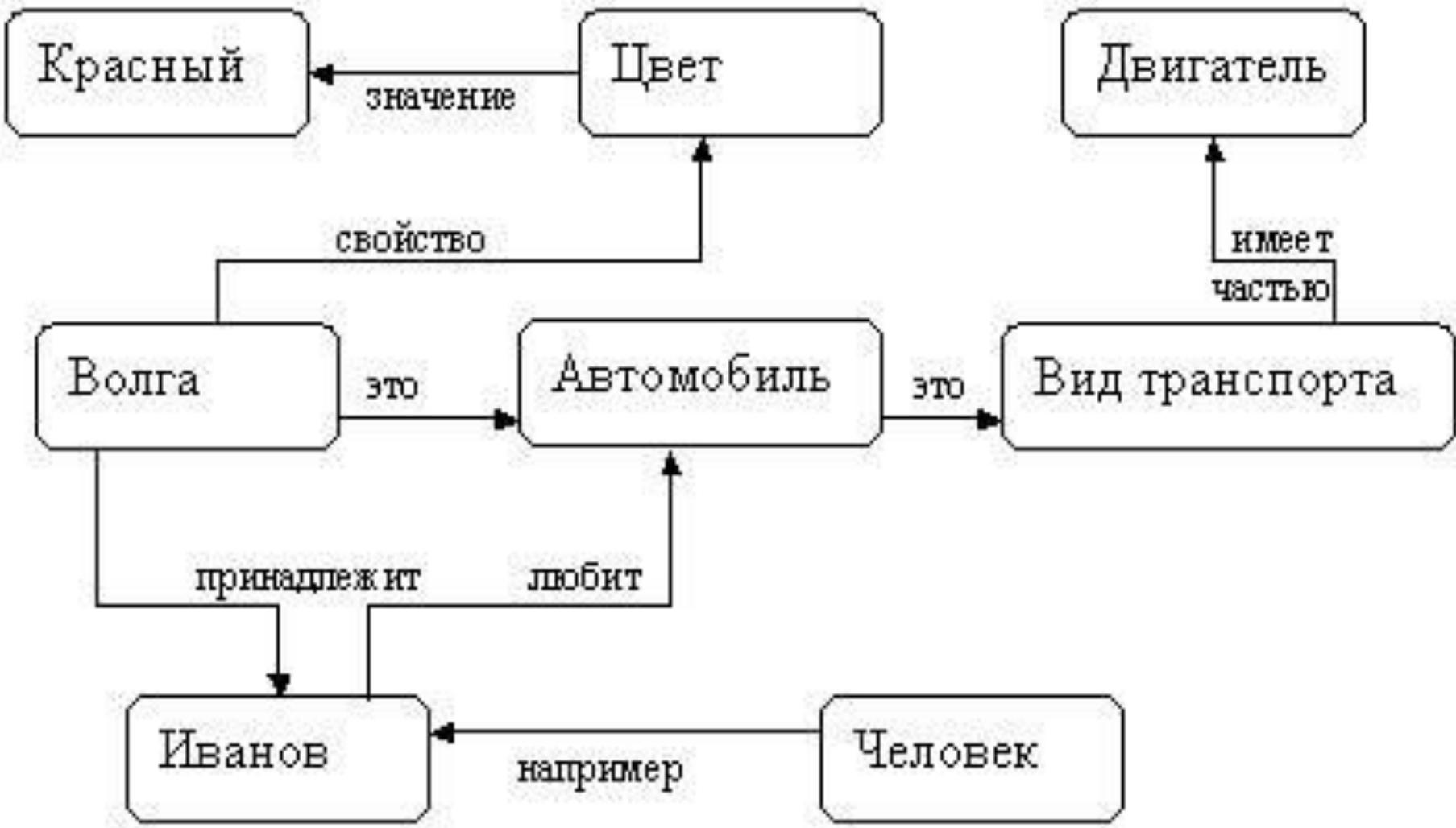
- связи типа "часть-целое" ("класс-подкласс", "элемент-множество" и т.п.);
- функциональные связи (определяемые обычно глаголами "производит", "влияет" ...);
- количественные (больше, меньше, равно...);
- пространственные (далеко от, близко от, за, под, над...);
- временные (раньше, позже, в течение...);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

Различают **экстенциональные** и **интенциональные** семантические сети.

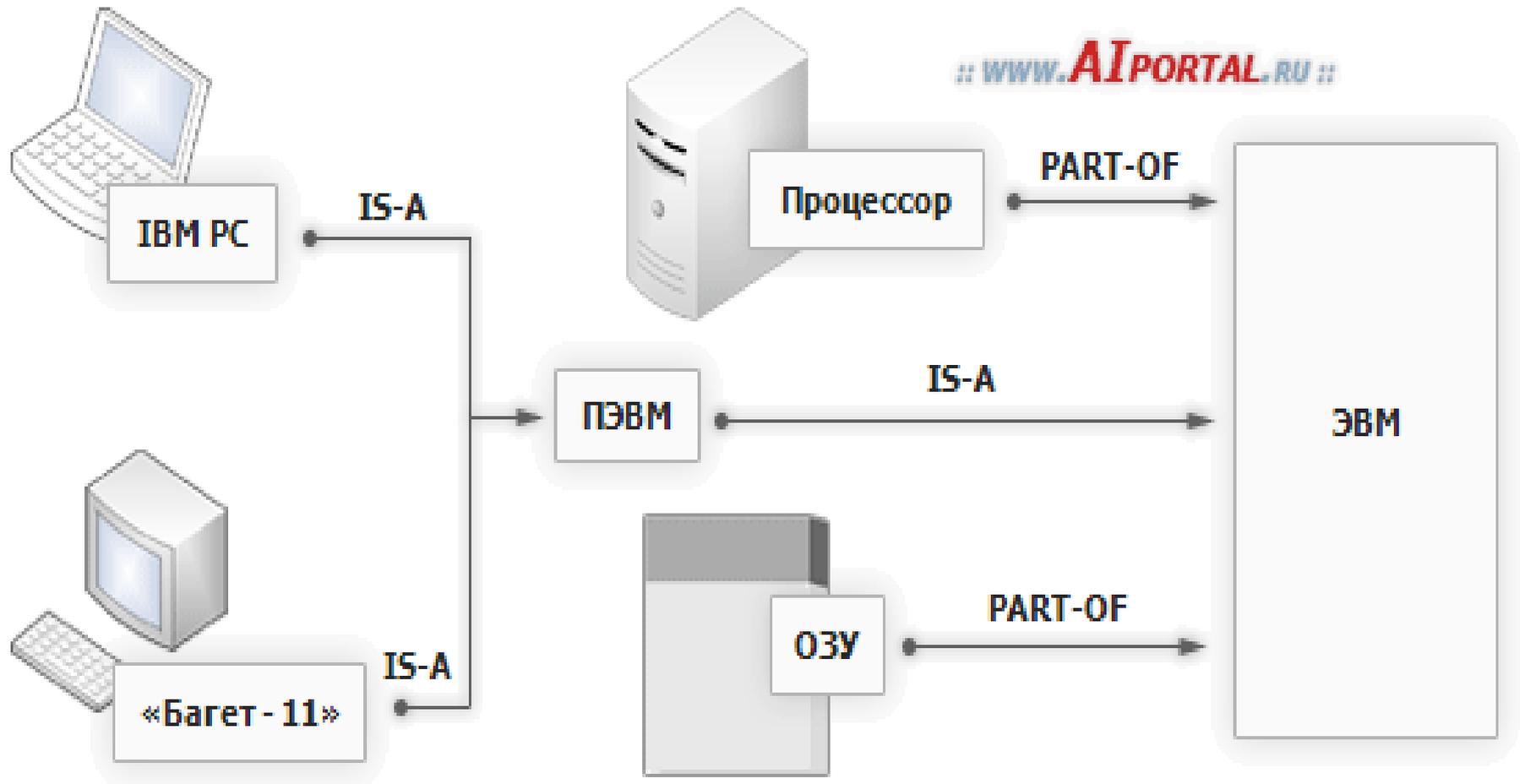
Экстенциональная семантическая сеть описывает конкретные отношения данной ситуации.

Интенциональная – имена классов объектов, а не индивидуальные имена объектов. Связи в интенциональной сети отражают те отношения, которые всегда присущи объектам данного класса.

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



Фрагмент описания вычислительной техники

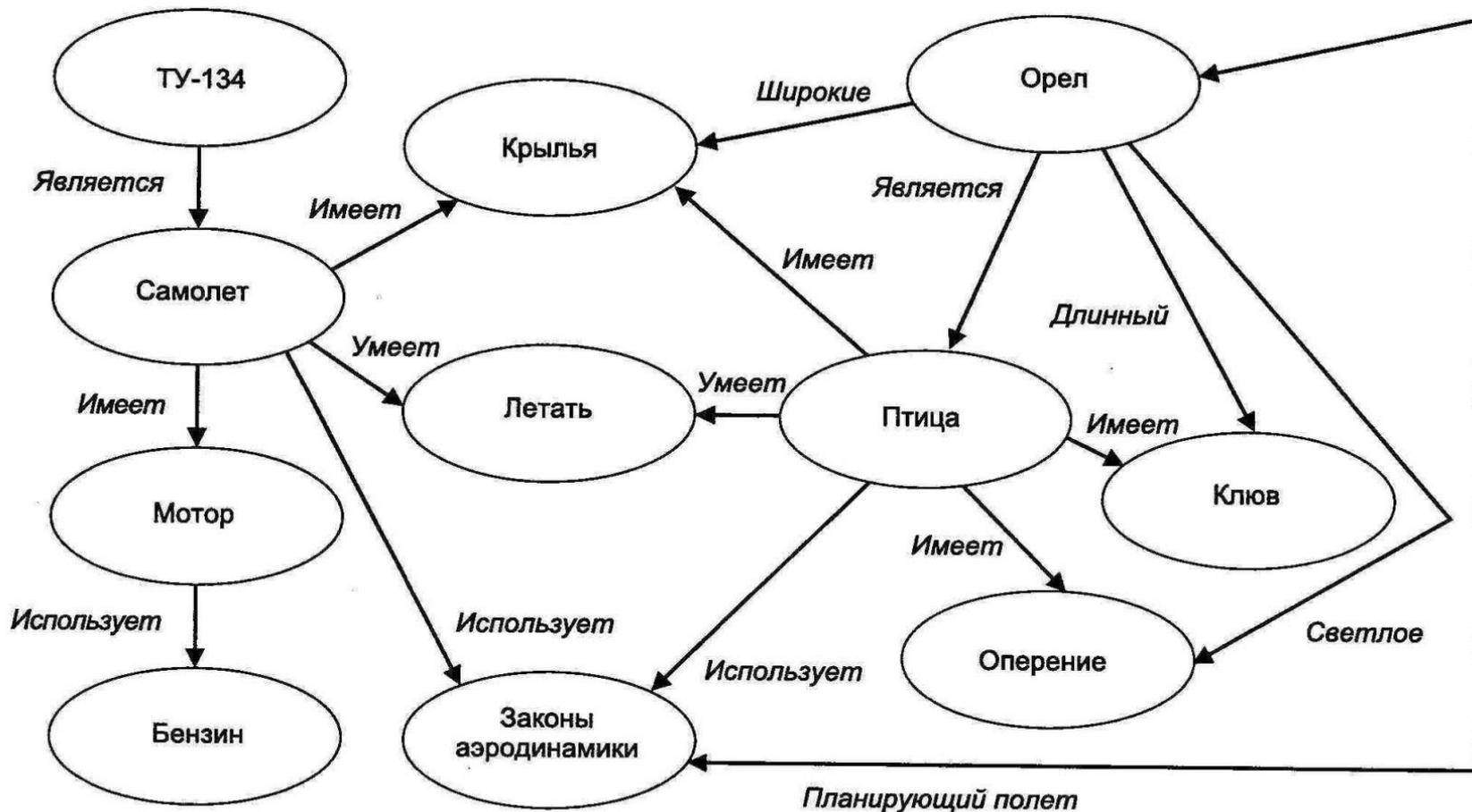


Преимущество этой модели - в соответствии современным представлениям об организации долговременной памяти человека.

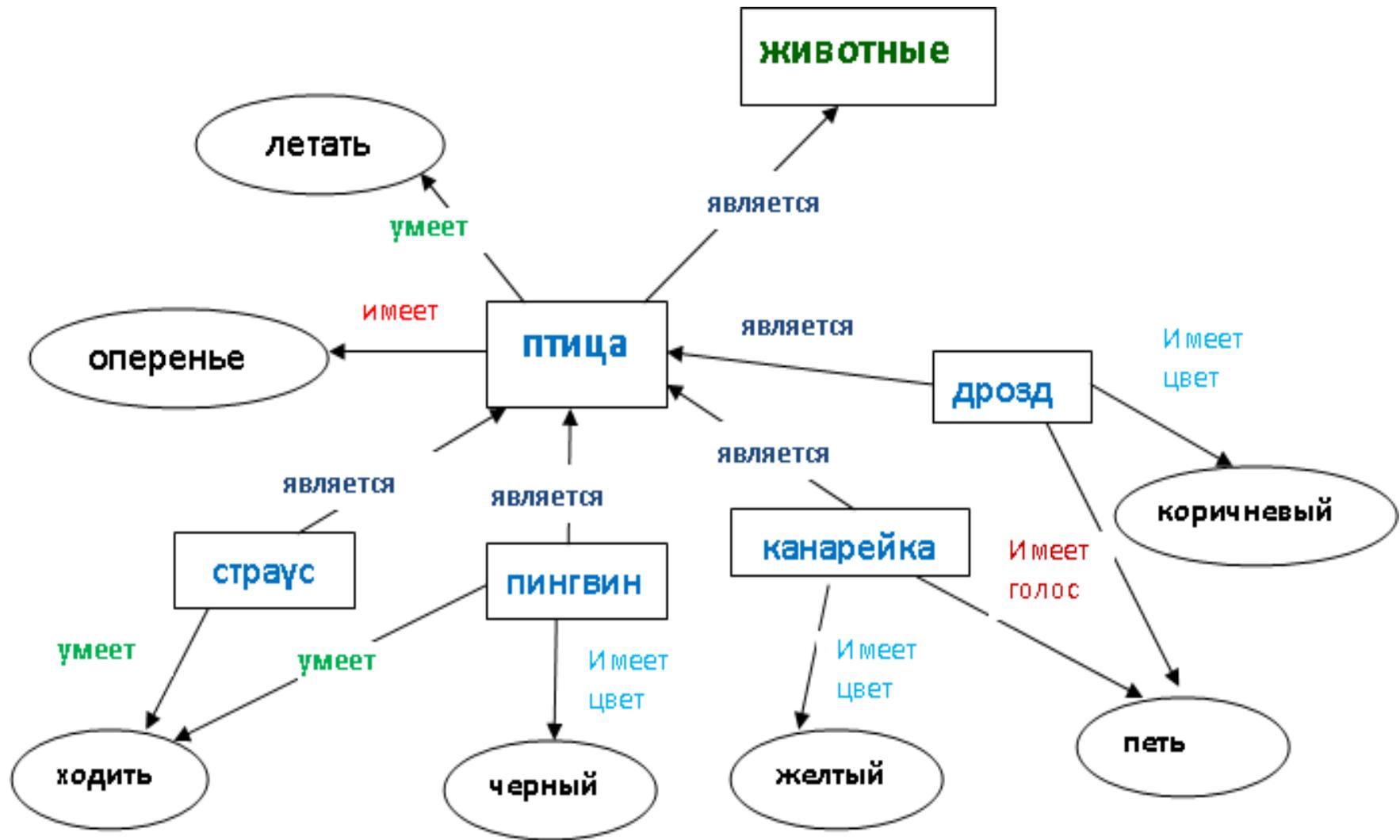
Недостаток модели - сложность поиска вывода на семантической сети.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, который соответствует некоторой подсети, соответствующей поставленному вопросу.

Семантическая сеть, показывающая взаимоотношения птиц и самолета



Фрагмент семантической сети, описывающей птиц



Для реализации семантических сетей существуют специальные сетевые языки, например NET и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний - PROSPECTOR, CASNBT, TORUS .

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 – слотов, 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
слота #1**

Имя фрейма

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

**Имя
слота #N**

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

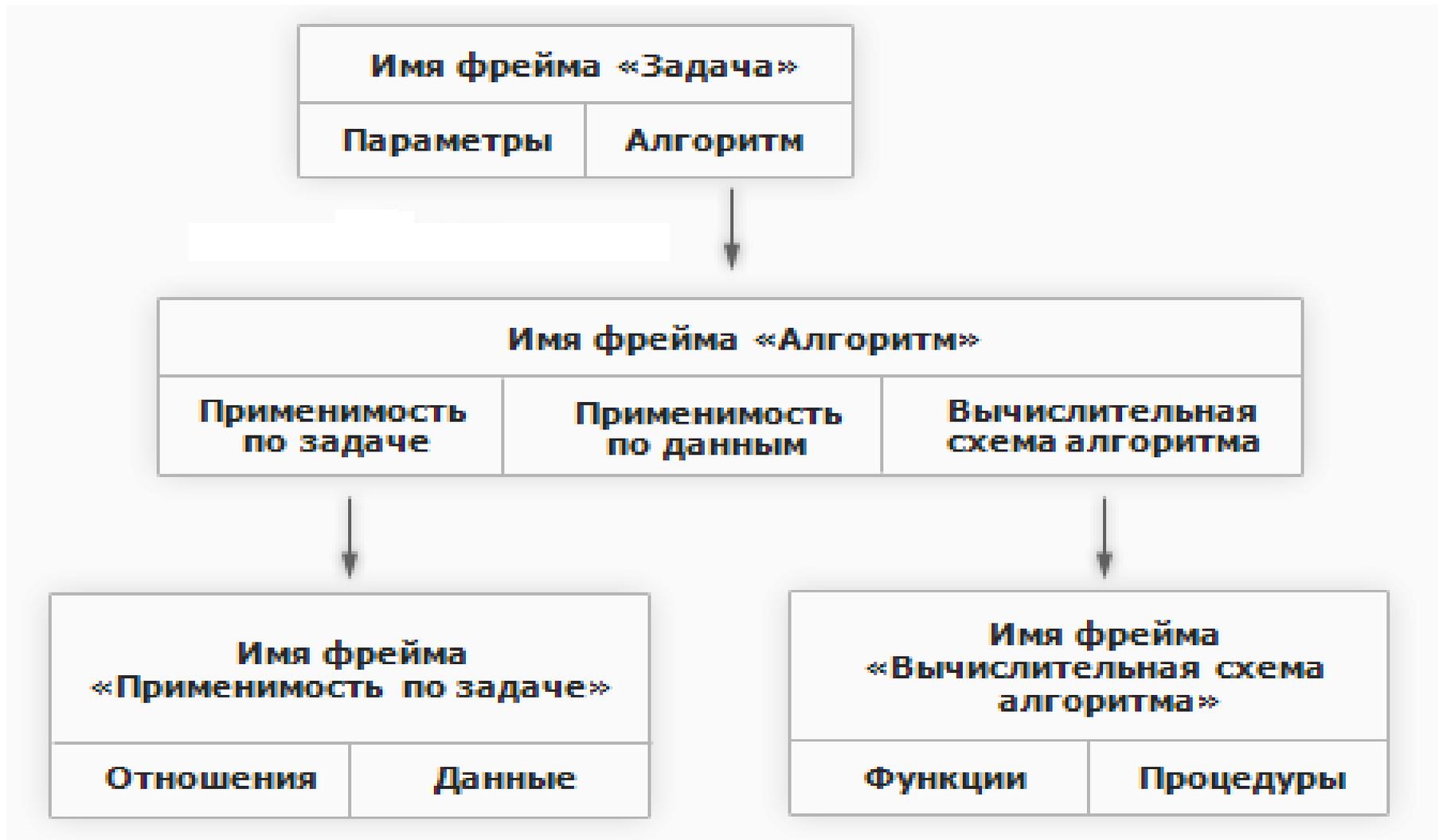
- **имя фрейма** – это идентификатор, присваиваемый фрейму. Фрейм должен иметь имя, единственное в данной фреймовой модели (уникальное имя);
- **имя слота** – это идентификатор, присваиваемый слоту. Слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой 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.

Пример фрейма РУКОВОДИТЕЛЬ

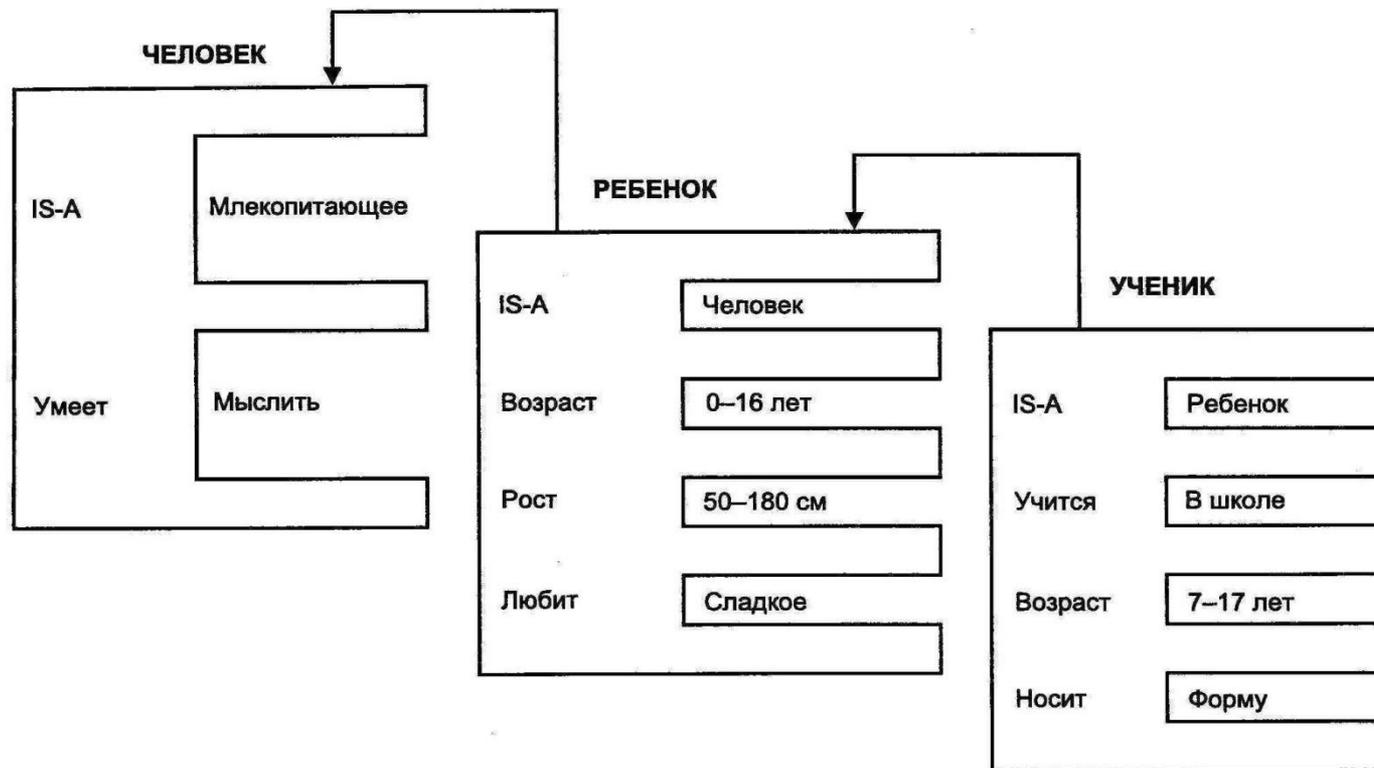
Имя слота	Значение слота	Тип значения слота
Имя	Иванов И.И.	Строка символов
Дата_рождения	01.01.1965	Дата
Возраст	Age(дата, дата_рождения)	Процедура
Специальность	Юрист	Строка символов
Отдел	Отдел кадров	Строка символов
Зарплата	50000	Число
Адрес	Дом_адрес	Фрейм

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



Пример иерархии фреймов



Фрагмент фрейма, описывающий гостиничный номер



Пример фреймов для базы знаний о птицах

name: bird
isa: animal
properties: flies feathers
default:

name: animal
isa: animate
properties: eats skin
default:

name: canary
isa: bird
properties: color(yellow) sound(sing)
default: size(small)

name: tweety
isa: canary
properties:
default: color(white)

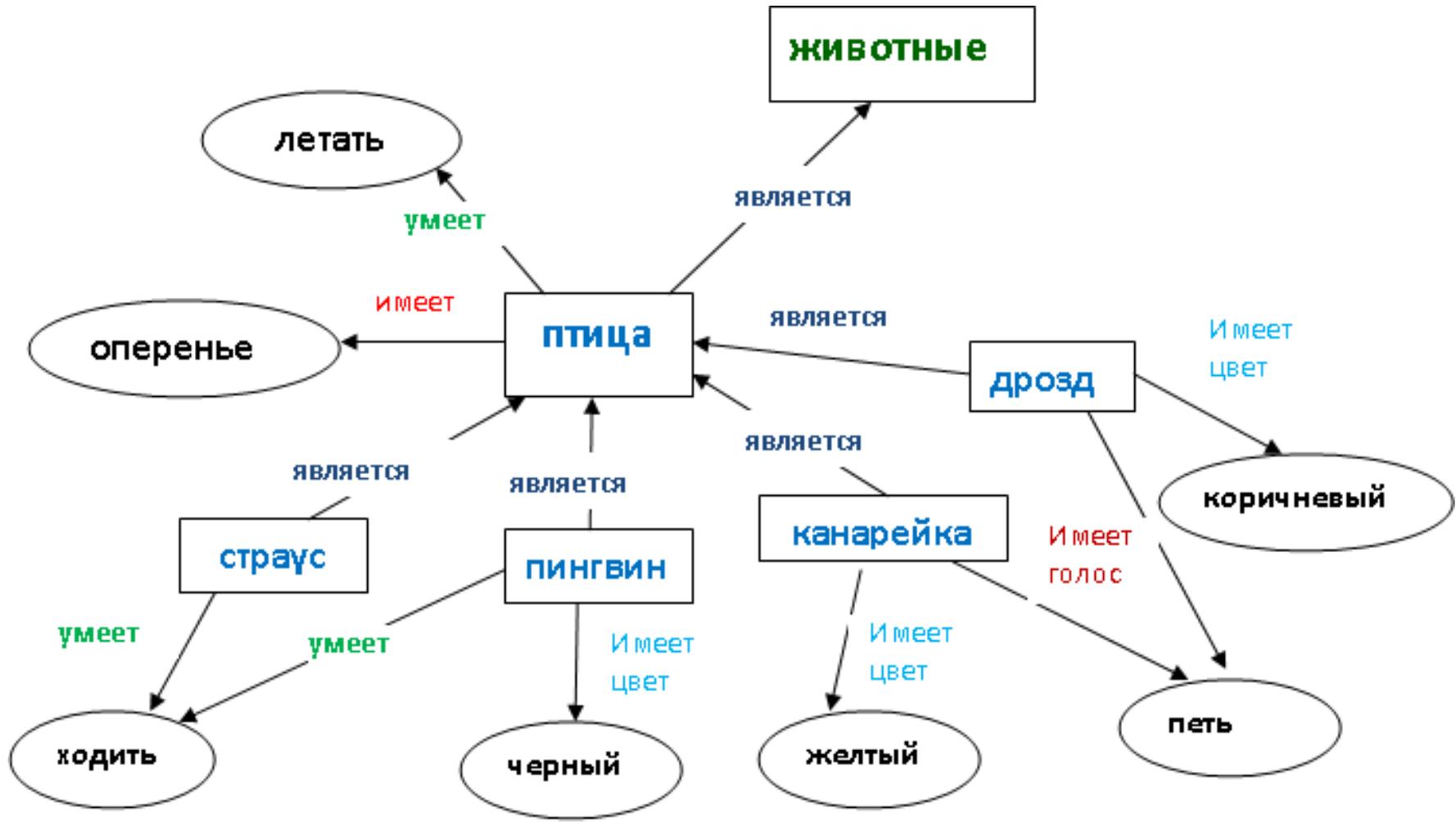
Фреймы подразделяются на:

- **фрейм-экземпляр** – конкретная реализация фрейма, описывающая текущее состояние в предметной области;
- **фрейм-образец** – шаблон для описания объектов или допустимых ситуаций предметной области;
- **фрейм-класс** – фрейм верхнего уровня для представления совокупности фреймов образцов.

- 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.

Реализация моделей данных на языке ПРОЛОГ

Фрагмент семантической сети, описывающей птиц



Обозначения:

Объекты- птица, страус, пингвин, канарейка, дрозд.

1) Отношение «является» - связывает классы различных уровней иерархии.

2) Отношение «имеет свойство» - описывает свойство объектов:

- имеет,
- имеет цвет,
- имеет голос,
- умеет.

Опишем предикаты:

1) Отношение «является».

является(объект, класс) – указывает, что объект является подтипом класса.

Например:

является(канарейка, птица).

является(страус, птица).

является(пингвин, птица).

является(дрозд, птица).

Является(птица, животные).

2) Отношение «**имеет свойство**» - описывает свойство объектов: **имеет, имеет цвет, имеет голос, умеет.**

имеет_свойство(объект, свойство, значение) –
описывает свойство объекта,
где объект и значение – узлы сети,
свойство- имя объединяющих их связи.

Например:

имеет_свойство(канарейка, цвет, желтый).

имеет_свойство(канарейка, голос, пение).

имеет_свойство(птица, покров, оперенье).

Можно рекурсивно определить, обладает ли некий объект семантической сети указанным свойством.

Например, свойство «летать» относится к классу «птица» и всем его подклассам. Исключения размещаются на отдельном уровне исключений.

Введем предикат и опишем правило:

**иметь_свойства(Объект, Свойство, Значение):-
имеет_свойство(Объект, Свойство, Значение).**

**иметь_свойства(Объект, Свойство, Значение):-
является(Объект, Класс),
иметь_свойства(Класс, Свойство, Значение).**

Задания для самостоятельного решения

- 1) Описать на языке Пролог фрагмент семантической сети о птицах, проверить работу программы.
- 2) Описать на языке Пролог семантическую сеть, описывающую самолет и птиц . Проверить работу программы.
- 3) Привести свой пример семантической сети (графическое представление, программная реализация)

Фреймы

Семантические сети можно разбивать на части, добавляя к описаниям узлов дополнительную информацию, получая фреймовую структуру.

Представим пример с птицами в виде фрейма.

Каждый фрейм- это набор отношений семантической сети.

1-ый слот – содержит имя узла,

2-ой слот - определяет отношение наследования между узлом и его родителями,

3-ий слот- содержит список свойств, описывающих этот узел,

4-ый слот может содержать сведения исключения (в виде списка).

Пример фреймовой структуры

ИМЯ:	птица
является:	животное
свойства:	летать, оперенье
ОТЛИЧИЯ:	

ИМЯ:	канарейка
является:	птица
свойства:	цвет желтый, имеет голос
ОТЛИЧИЯ:	маленький размер

Введем предикат **фрейм**, который связывает имена слотов, списки свойств и принимаемые по умолчанию значения.

**фрейм(имя (птица),
является(животное),
[летать, оперенье],
[]).**

**фрейм(имя (канарейка),
является(птица),
[цвет_желтый, имеет_голос],
[маленький_размер]).**

Для определения всех свойств объекта, опишем рекурсивную процедуру.

определить(Свойство, Объект):-

**фрейм(имя(Объект),_,Список_свойств,_),
запомнить(Свойство, Список_свойств).**

определить(Свойство, Объект):-

**фрейм(имя(Объект),_,_,Список_исключений),
запомнить(Свойство, Список_исключений).**

определить(Свойство, Объект):-

**фрейм(имя(Объект),является(Класс),_,_),
определить(Свойство, Класс).**

Задания для самостоятельного решения

- 1) Дополнить фреймовое описание о птицах. Описать на языке Пролог фреймовое представление о птицах, проверить работу программы.
- 2) Привести свой пример фрейма (графическое представление, программная реализация)

Логическая модель

Логическая модель основана на системе исчисления предикатов первого порядка.

Высказывание – это предложение, смысл которого можно выразить значениями истина и ложь.

Исчисление высказываний позволяет формализовать лишь малую часть множества рассуждений, т.к. этот аппарат не позволяет учитывать внутреннюю структуру высказывания, которая существует в естественных языках.