**Л 6. Сетевые и фреймовые модели представления знаний**

1. Семантические сети. Понятия, события и свойства. Виды семантических сетей. Семантические сети Куиллиана. Основной способ интерпретации семантической сети.
2. Фреймы. Слоты. Структура фрейма. Указатели наследования и типа данных. Значение слота. Присоединенные процедуры.
3. **Семантические сети**

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

**Основная идея подхода к представлению знаний, основанных на аппарате семантических сетей, состоит в том, чтобы рассматривать проблемную среду как совокупность сущностей (объектов) и (отношений) связей между ними.** Система знаний представляется семантической сетью. Сущности представляются при этом поименованными вершинами, а отношения - направленными поименованными ребрами.

Семантическая сеть наиболее близка к тому, как представляются знания в текстах на естественном языке. В ее основе лежит идея о том, что вся необходимая информация может быть описана как совокупность троек (**a** *r* **b**), где **а** и **b** - два объекта или понятия, а *r* двоичное отношение между ними.

Графически семантическая сеть представляется в виде помеченного ориентированного графа, в котором **вершинам** соответствуют **объекты** (понятия), а **дугам** - их **отношения**. Дуги помечаются именами соответствующих отношений.

**Понятия, события и свойства**

Используются три основных типа объектов: **понятия, события и свойства**.



**Виды семантических сетей**

Семантическая сеть является моделью широкого предназначения. Выделяют следующие виды семантических сетей:

**ситуационные** сети, которые описывают временные, пространственные и причинно-следственные (**каузальные -** зависимость между переменными или понятиями, при которой изменение в одной (или более) переменной или понятии влечет за собой изменение в другой (других)) отношения;

**целевые** сети, используемые в системах планирования и синтеза, которые описывают отношения "цель-средства" и "цель-подцель";

**классификационные** сети, использующие отношения "род-вид", "класс-подкласс";

**функциональные** сети, использующие отношения "аргумент-функция" и т.д.

Особенность семантической сети как модели представления знаний, которая может одновременно считаться и ее достоинством, и ее недостатком, заключается в невозможности в явном виде разделить БЗ и механизм логического вывода. Поэтому интерпретация семантической сети осуществляется только с помощью использующих ее процедур.

**Семантические сети Куиллиана**

При построении семантической сети отсутствуют ограничения на число связей и сложность сети. Семантическая сеть должна быть систематизирована для того, чтобы формализация оказалась возможной. Семантические сети TLC (Куиллиана) систематизируют функции отношений между понятиями с помощью следующих признаков:

• множество-подмножество (типы отношений «целое-часть»; род-вид» и т. д.);

• отношений «близости»;

• отношений «сходства-различия»;

• логических связей (И, ИЛИ, НЕ);

• количественных связей (больше, меньше, равно, ...);

• пространственных связей (далеко от, за, над, ...);

• временных связей (раньше, позже, в течение, ...);

• атрибутных связей (иметь свойство, иметь значение);

• лингвистических связей и др.

**Основной идеей семантической сети TLC** было описание значений класса, к которому принадлежит объект, его прототипа и установление связи со словами, отображающими свойства объекта.

Пример семантической сети приведен на рис. 6.1.

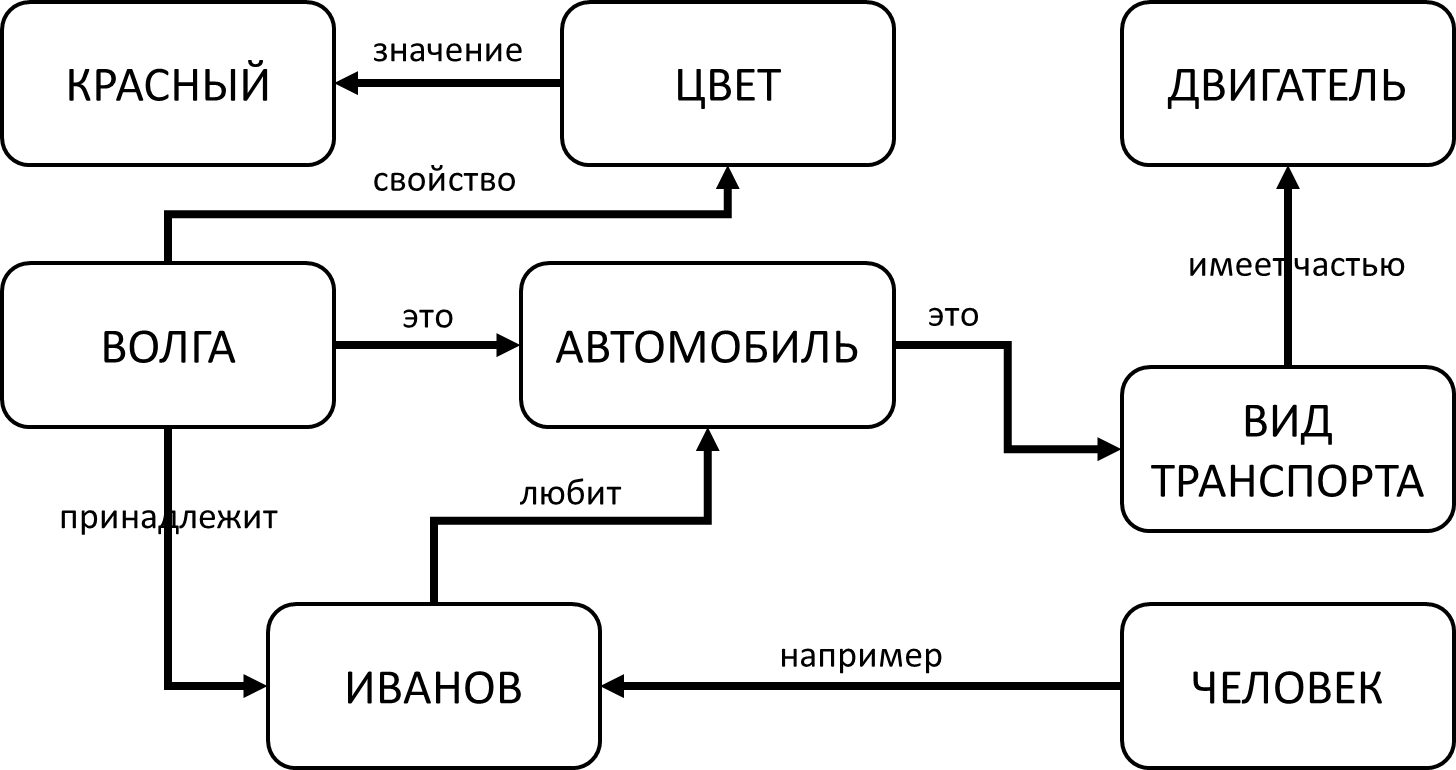


Рис. 6.1. Семантическая сеть

Для реализации семантических сетей существуют специальные языки: **NET**, язык **SIMER+MIR** и др.

**PROSPECTOR, TORUS** — известные экспертные системы, использующие семантические сети в качестве модели представления знаний.

**Основной способ интерпретации семантической сети.**

Основным способом интерпретации семантической сети является **способ сопоставления частей сетевой структуры**. Он основан на построении подсети (подграфа), соответствующей задаваемому вопросу, и сопоставлении ее с общей сетью, имеющейся в БЗ. Запросная подсеть накладывается на имеющийся в базе знаний фрагмент.

Для поиска отношений между концептуальными объектами используется другой способ перекрестного поиска. Согласно этому способу ответ на вопрос выводится путем обнаружения в имеющейся сети узла, в котором пересекаются дуги, исходящие из различных узлов запросной подсети.

Систематизация отношений семантической сети является сложной задачей, зависящей от специфики знаний предметной области. Общезначимые отношения, присутствующие во многих предметных областях, являются основой концепции семантических сетей. Семантические сети получили широкое распространение в системах распознавания речи и экспертных системах. Необходимость структуризации семантических сетей привела к появлению концепции фреймов.

1. **Фреймы**

Фреймовая модель представления знаний основана на теории фреймов М. Минского, которая представляет собой психологическую модель памяти человека и его сознания.

**Фрейм** - это абстрактный образ для представления некоего стереотипа восприятия (объекта, понятия или ситуации) путем сопоставления факта с конкретными элементами и значениями в рамках, определенных для объекта, в структуре БЗ. Под абстрактным образом понимается некоторая обобщенная и упрощенная модель или структура. Фреймом также называется и формализованная модель для отображения образа.

**Фреймовая МПЗ** базируется так же на понятии функционального программирования - способа составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программ на части является введение имени для функции и задание для этого имени выражения, вычисляющего значение функции, а единственным правилом композиции - оператор суперпозиции других функций. Оно следует из такого раздела математики, как лямбда-исчисление, созданное американским ученым А. Чёрчем.

Фрейм имеет имя, служащее для идентификации описываемого им понятия. Различают фреймы-образцы, или прототипы, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма отображает все многообразие знаний о мире через:

• фреймы-структуры, которые применяются для обозначения объектов и понятий (заем, залог, вексель);

• фреймы-роли (менеджер, кассир, клиент);

• фреймы-сценарии (банкротство, собрание акционеров, и т. д.);

• фреймы-ситуации (тревога, авария, рабочий режим устройства и др.).

**Слоты**

**Фрейм** - структура, состоящая из имени фрейма, слотов (имена и значения) и присоединенных процедур, связанных с фреймом или со слотами. Основные структурные элементы фрейма определяются с помощью слотов. Текущие значения слотов помещаются в **шпации**. Число слотов в каждом фрейме устанавливается проектировщиком системы, при этом часть слотов определяется самой системой для выполнения специфических функций (**системные слоты**), примерами которых являются: **слот-указатель родителя данного фрейма (АКО),** слот-указатель дочерних фреймов, слот для определения даты создания фрейма и т. д.

Слот имеет уникальное в пределах фрейма имя; в качестве имени слота может выступать любой произвольный текст. Имена системных слотов обычно зарезервированы, в различных системах они могут иметь различные значения. Системные слоты могут иметь имена: АКО, RELATIONS и т. д. Системные слоты служат для редактирования базы знаний и управления выводом во фреймовой системе.

Слот может содержать не только конкретное значение, но также имя процедуры, позволяющей вычислить это значение по заданному алгоритму. Такие процедуры называются присоединенными или связанными процедурами. Вызов связанной процедуры осуществляется при обращении к слоту, в котором она помещена. Например, слот «возраст» может одержать имя процедуры, которая вычисляет возраст по дате рождения, записанной в другом слоте, и текущей дате.

В слоте могут содержаться данные сложных типов: массивы, списки, множества, фреймы и т. д. Значение слота может представлять собой арифметическое значение, фрагмент текста и т. д.

Формально структура фрейма может быть представлена как список свойств:

(<имя\_фрейма>

<имя\_слота1> (<значение\_слота1>) <тип\_значения\_слота1>

<имя\_слота2> (<значение\_слота2>) <тип\_значения\_слота2>

. . . . . . . . . . .

<имя\_слота N> (<значение\_слота N>) <тип\_значения\_слота N>)

Зададим, например, фрейм для объекта «Преподаватель»:

(Преподаватель

ФИО (Петров И. П.) строка\_символов

Уч\_степень (канд) строка\_символов

Уч\_звание (доцент) строка\_символов

Должность (доцент) строка\_символов

Возраст age(дата\_рожд) процедура

Адрес Дом\_адрес фрейм)

**Структура фрейма**

Фрейм имеет структуру, представленную в табл. 6.1.

Таблица 6.1

Структура фрейма

|  |  |  |
| --- | --- | --- |
| Имя фрейма | | |
| Имя слота | Значение слота | Значение типа данных слота |

Совокупность данных предметной области может быть представлена множеством взаимосвязанных фреймов, образующих единую фреймовую систему, в которой объединяются декларативные и процедурные знания, такая система имеет, как правило, иерархическую структуру, в которой фреймы соединены друг с другом с помощью родовидовых связей. На верхнем уровне находится фрейм, содержащий наиболее общую информацию, истинную для всех фреймов.

**Указатели наследования и типа данных. Значение слота.**

**Важнейшим свойством теории фреймов** является заимствование из теории семантических сетей - так называемое наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО -связям. Слот связи указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т.е. переносятся, значения аналогичных слотов.

**Указатели наследования** показывают, какую информацию об атрибутах слотов из фрейма верхнего уровня наследуют слоты с одинаковыми именами в данном фрейме. Указатели наследования характерны для фреймовых систем иерархического типа. Наследование происходит по АКО-связям (A-Kind-Of). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда наследуются значения одинаковых слотов.

В конкретных системах указатели наследования могут быть организованы различными способами: U (Unique) - значение слота не наследуется; S (Same) - значение слота наследуется; R (Range) - значение слота должно находиться в пределах интервала значений, указанных в одноименном слоте родительского фрейма; О (Override) - при отсутствии значения в текущем слоте оно наследуется из фрейма верхнего уровня].

Например, в сети фреймов на рис. 6.2 понятие «ученик» наследует свойства фреймов «ребенок» и «человек», которые находятся на более высоком уровне иерархии. На вопрос «любят ли ученики сладкое?» следует ответ «да», так как этим свойством обладают все дети, что указано во фрейме «ребенок». Наследование свойств может быть частичным, так как возраст для учеников не наследуется из фрейма «ребенок», поскольку указан явно в своем собственном фрейме.

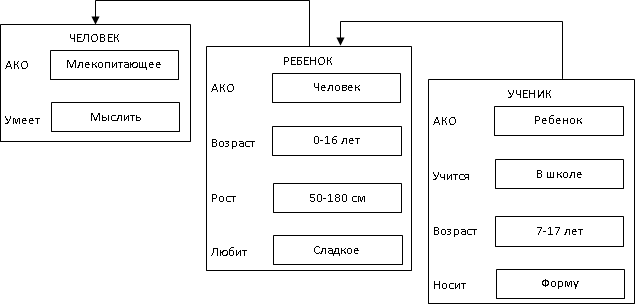


Рис. 6.2. Фреймовая модель, описывающая понятие «ученик»

**Указатель типа данных** показывает тип значения слота. Наиболее употребляемые типы: frame - указатель фрейма; real, integer, Boolean, text, table, expression, lisp - вещественный, целый, логический текстовый тип, таблица, выражение, связанная процедура и т. д.

Значение слота должно соответствовать указанному типу данных и условию наследования. Существует несколько способов получения слотом значений во фрейме-экземпляре:

• по умолчанию от фрейма-образца (значение по умолчанию);

• через наследование свойств от фрейма, указанного в слоте АКО;

• по формуле, указанной в слоте;

• через присоединенную процедуру;

• явно из диалога с пользователем;

• из базы данных.

**Присоединенные процедуры**

Во фреймах различают два типа присоединенных процедур: **процедуры-демоны и процедуры-слуги.**

Демоны (процедуры) автоматически запускаются при обращении к соответствующему слоту. Типы демонов связаны с условием запуска процедуры. Демон с условием IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено. Демон IF-ADDED запускается при попытке изменения слота. Демон IF-REMOVED запускается при попытке удаления значения слота. Возможны также другие типы демонов. Процедуры-слуги запускаются явно.

В качестве значения слота может быть присоединенная процедура, называемая **служебной** в языке LISP или **методом** в языках объектно- ориентированного программирования. Присоединенная процедура запускается по сообщению, передаваемому из другого фрейма. Демоны и присоединенные процедуры являются процедурными знаниями, объединенными с декларативными в единую систему.

Фреймовые системы в последние годы реализуют на объектно-ориентированных языках, но в этих языках нет средств для реализации присоединенных процедур, поэтому требуется интеграция объектно-ориентированного языка с другими средствами обработки знаний (например, с языком PROLOG). Существуют также специализированные языки, такие как семейство KL(1) представления знаний на основе фреймовой модели.

Основным преимуществом фреймов как модели представления знаний является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность. Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language).