

image not found or type unknown



Фреймовая модель представления знаний – была предложена М.Минским в 1979 г. как структура знаний для восприятия пространственных сцен. Эта модель, как и семантическая сеть, имеет глубокое психологическое обоснование.

В психологии и философии известно понятие абстрактного образа. Например, слово «комната» вызывает у слушающих образ комнаты: «жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью, площадью 6-20 м²». Из этого описания ничего нельзя убрать (например, убрав окна, получим уже чулан, а не комнату), но в нем есть «дырки», или «слоты», — это незаполненные значения некоторых атрибутов — количество окон, цвет стен, высота потолка, покрытие пола и др. В этой теории такой абстрактный образ называется фреймом.

Фреймом называется также и формализованная модель для отображения образа.

В качестве идентификатора фрейму присваивается имя фрейма. Это имя должно быть единственным во всей фреймовой системе.

Фрейм имеет определенную внутреннюю структуру, состоящую из множества элементов, называемых *слотами*, которым также присваиваются имена. За слотами следуют *шпации*, в которые помещают данные, представляющие текущие значения слотов. Каждый слот в свою очередь представляется определенной структурой данных. В значение слота подставляется конкретная информация, относящаяся к объекту, описываемому этим фреймом.

Структуру фрейма можно представить так:

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).

Ту же запись представим в виде таблицы, дополнив двумя столбцами.

Таблица 5.1 – Структура фрейма

Имя слота	значение слота	способ получения значения	присоединённая процедура
------------------	-----------------------	----------------------------------	---------------------------------

В таблице дополнительные столбцы предназначены для описания способа получения слотом его значения и возможного присоединения к тому или иному слоту специальных процедур, что допускается в теории фреймов. В качестве значения слота может выступать имя другого фрейма; так образуют сети фреймов.

Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных.

Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире:

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

Важнейшим свойством теории фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО-связям (A-Kind-Of = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся, значения аналогичных слотов.

Значением слота может быть практически что угодно: числа, формулы, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов. В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет реализовывать во фреймовых представлениях «принцип матрешки». Связи между фреймами задаются значениями специального слота с именем «Связь».

В общем случае структура данных фрейма может содержать более широкий набор информации, в который входят следующие атрибуты.

Имя фрейма. Оно служит для идентификации фрейма в системе и должно быть уникальным. Фрейм представляет собой совокупность слотов, число которых может быть произвольным. 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 (**IS-A**), слот-указатель дочерних фреймов, слот для ввода имени пользователя, слот для ввода даты определения фрейма, слот для ввода даты изменения фрейма и т.д.

Имя слота. Оно должно быть уникальным в пределах фрейма. Обычно имя слота представляет собой идентификатор, который наделен определенной семантикой. В качестве имени слота может выступать произвольный текст. Например, <Имя слота> = Главный герой романа Ф.М. Достоевского «Идиот», <Значение слота> = Князь Мышкин. Имена системных слотов обычно зарезервированы, в различных системах они могут иметь различные значения. Примеры имен системных слотов: *IS-A*, *HASPART*, *RELATIONS* и т.д. Системные слоты служат для редактирования базы знаний и управления выводом во фреймовой системе.

Указатели наследования. Они показывают, какую информацию об атрибутах слотов из фрейма верхнего уровня наследуют слоты с аналогичными именами в данном фрейме. Указатели наследования характерны для фреймовых систем иерархического типа, основанных на отношениях типа «абстрактное — конкретное». В конкретных системах указатели наследования могут быть организованы различными способами и иметь разные обозначения:

U (Unique) — значение слота не наследуется;

S (Same) — значение слота наследуется;

R (Range) — значения слота должны находиться в пределах интервала значений, указанных в одноименном слоте родительского фрейма;

O (Override) — при отсутствии значения в текущем слоте оно наследуется из фрейма верхнего уровня, однако в случае определения значения текущего слота оно может быть уникальным. Этот тип указателя выполняет одновременно функции указателей U и S.

Указатель типа данных. Он показывает тип значения слота. Наиболее употребляемые типы: *frame* — указатель на фрейм; *real* — вещественное число; *integer* — целое число; *boolean* — логический тип; *text* — фрагмент текста; *list* — список; *table* — таблица; *expression* — выражение; *lisp* — связанная процедура и т.д.

Значение слота. Оно должно соответствовать указанному типу данных и условию наследования.

Демоны. Демоном называется процедура, автоматически запускаемая при выполнении некоторого условия. Демоны 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 типа *IF-ADDED* запускается при попытке изменения значения слота. Демон *IF-REMOVED* запускается при попытке удаления значения слота. Возможны также другие типы демонов. Демон является разновидностью связанной процедуры.

Присоединенная процедура. В качестве значения слота может использоваться процедура, называемая служебной в языке Лисп или методом в языках объектно-ориентированного программирования. Присоединенная процедура запускается по сообщению, переданному из другого фрейма. Демоны и присоединенные процедуры являются процедурными знаниями, объединенными вместе с декларативными в единую систему. Эти процедурные знания являются средствами управления выводом во фреймовых системах, причем с их помощью можно реализовать любой механизм вывода. Представление таких знаний и заполнение ими интеллектуальных систем — весьма нелегкое дело, которое требует дополнительных затрат труда и времени разработчиков. Поэтому проектирование фреймовых систем выполняется, как правило, специалистами, имеющими высокий уровень квалификации в области искусственного интеллекта.

Часть специалистов по системам искусственного интеллекта считают, что нет необходимости выделять фреймовые модели представления знаний, так как в них объединены все основные особенности моделей остальных типов.

Приведем несколько примеров фреймовых описаний.

Пример 1. Фрейм, описывающий человека.

Фрейм:	Человек
Имя слота:	Значение слота
Класс:	Животное
Структурный элемент:	Голова, шея, руки, ...
Рост:	30 ? 220 см
Масса:	1 ? 200 кг
Хвост:	Нет
Язык:	Русский, английский, ...
Связь:	Обезьяна

Пример 2. Описание с помощью фреймов понятия письменного отчета.

В виде семантической сети «отчет» можно представить следующим образом.

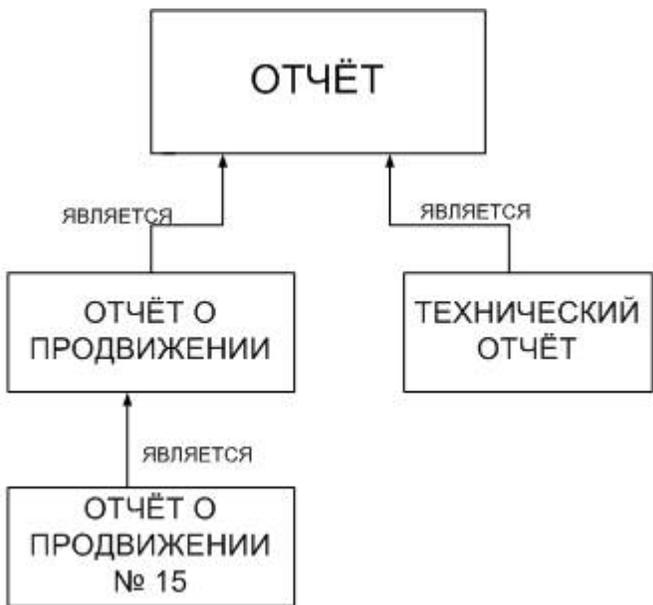


Рис. 5.1. Семантическая сеть понятия «отчет»

Приведем представление понятия отчёта с помощью фреймов.

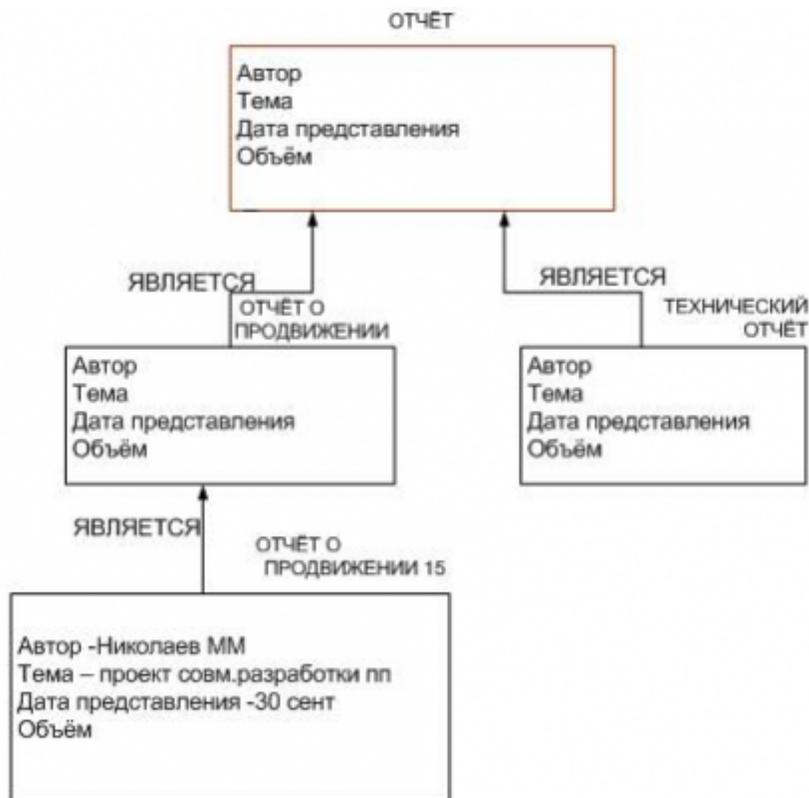


Рис. 5.2. Представление понятия «отчет» в виде фреймов

Пример 3. Спецификации фреймов на продукционно-фреймовом языке описания знаний PILOT/2.

Фреймы John и Mary — экземпляры фрейма-прототипа Person.

[Person **is_a prototype**;

Name **string, if_changed** ask_why();

Age **int, restr_by** >=0;

Sex **string, restr_by** (=="male" || == "female"),

by_default "male";

Children {**frame**}];

[John **is_a** Person; **if_deleted** bury();

Name = "Johnson";

Age = 32;

Children = {Ann, Tom}];

[Mary **is_a** Person;

without Age;

Name = "Smirnova";

Sex = "female";

Children = **empty**];

Основным преимуществом **фреймов** как **модели представления знаний** является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность. Наиболее ярко достоинства фреймовых систем представления знаний проявляются в том случае, если родовидовые связи изменяются нечасто и предметная область насчитывает немного исключений. Во фреймовых системах данные о родовидовых связях хранятся явно, как и знания других типов. Значения слотов представляются в системе в единственном экземпляре, поскольку включаются только в один фрейм, описывающий наиболее понятия из всех тех, которые содержит слот с данным именем. Такое свойство систем фреймов обеспечивает экономное размещение базы знаний в памяти компьютера. Еще одно достоинство фреймов состоит в том, что значение любого слота

может быть вычислено с помощью соответствующих процедур или найдено эвристическими методами. То есть фреймы позволяют манипулировать как декларативными, так и процедурными знаниями.

К недостаткам фреймовых систем относят их относительно высокую сложность, что проявляется в снижении скорости работы механизма вывода и увеличения трудоемкости внесения изменений в родовую иерархию. Поэтому при разработке фреймовых систем уделяют наглядным способам отображения и эффективным средствам редактирования фреймовых структур.

Специальные языки представления знаний в сетях фреймов FRL (Frame Representation Language), KRL (Knowledge Representation Language), **фреймовая** оболочка Карра, PILOT/2 и другие программные средства позволяют эффективно строить промышленные системы.

В последние годы термин «фреймовый» часто заменяют термином «объектно-ориентированный». Этот подход является развитием **фреймового представления**. Шаблон фрейма можно рассматривать как класс, экземпляр фрейма — как объект. Языки объектно-ориентированного программирования (ООП) предоставляют средства создания классов и объектов, а также средства для описания процедур обработки объектов (методы). Языки ООП, не содержащие средств реализации присоединенных процедур, не позволяют организовать гибкий механизм логического вывода, поэтому разработанные на них программы либо представляют собой объектно-ориентированные базы данных, либо требуют интеграции с другими средствами обработки знаний (например, с языком PROLOG).

Объектно-ориентированная методология представления знаний реализована в системах G2, RTWorks.