



В настоящее время разработано множество моделей представления знаний. Имея обобщенное название, они различаются по идеям, лежащим в их основе, с точки зрения математической обоснованности. Типы моделей показаны на рисунке:

Первый подход, называемый эмпирическим, основан на изучении принципов организации человеческой памяти и моделировании механизмов решения задач человеком. На основе этого подхода в настоящее время разработаны и получили наибольшую известность следующие модели:

продукционные модели – модель основанная на правилах, позволяет представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие». Продукционная модель обладает тем недостатком, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций они начинают противоречить друг другу; сетевые модели (или семантические сети) – в инженерии знаний под ней подразумевается граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами. Обладает тем недостатком, что однозначного определения семантической сети в настоящее время отсутствует;

фреймовая модель – основывается на таком понятии как фрейм (англ. frame – рамка, каркас). Фрейм – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах. Слоты могут быть терминальными либо являться сами фреймами, т.о. образуя целую иерархическую сеть.

Условно в группу эмпирического подхода можно включить нейронные сети и генетические алгоритмы, относящиеся к бионическому (основано на предположении о том, что если в искусственной системе воспроизвести структуры и процессы человеческого мозга, то и результаты решения задач такой системой будут подобны результатам, получаемым человеком) направлению искусственного интеллекта. Особенностью моделей этого типа является широкое использование эвристик, что в каждом случае требует доказательства правильности получаемых решений.

Второй подход можно определить как теоретически обоснованный, гарантирующий правильность решений. Он в основном представлен моделями,

основанными на формальной логике (исчисление высказываний, исчисление предикатов), формальных грамматиках, комбинаторными моделями, в частности моделями конечных проективных геометрий, теории графов, тензорными и алгебраическими моделями. В рамках этого подхода до настоящего времени удавалось решать только сравнительно простые задачи из узкой предметной области.

ПРЕДСТАВЛЕНИЕ ЗНАНИЙ ФРЕЙМАМИ. ФРЕЙМОВАЯ МОДЕЛЬ ЗНАНИЙ.

Понятие фрейма.

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.

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.

В основе данной модели представления знаний лежит свойство концептуальных объектов иметь аналогии, которые позволяют строить иерархические структуры отношений типа “абстрактное-конкретное”.

Каждый фрейм следует рассматривать как сеть из нескольких вершин и отношений. На самом верхнем уровне фрейма представляется фиксированная информация о состоянии моделируемого объекта, которая является истинной вне зависимости от контекста рассмотрения объекта и соответствует имени фрейма.

Фрейм (англ. frame – рамка, каркас) – 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.

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.

Каждый фрейм, как показано на рисунке ниже, состоит из произвольного числа слотов, причем несколько из них обычно определяются самой системой для выполнения специфических функций, а остальные определяются пользователем.

В каждом слоте задается условие, которое должно выполняться при установлении соответствия между значениями (слот либо сам устанавливает соответствие, либо обычно это делает более мелкая составляющая фрейма). В одной системе различные фреймы могут иметь общие терминалы. Несколько терминалов одного фрейма обычно заранее определяются значениями по умолчанию, что позволяет представлять информацию общего характера при решении сходных задач. Среди слотов отдельного фрейма выделяют слоты, определяемые системой и определяемые пользователем. Примерами системных слотов могут служить : IS_A (указание на фрейм-родитель), слот указателей дочерних фреймов, слоты дат создания и изменения информации фрейма.

Определение. Фреймы, соответствующие описанию отдельных объектов, называются шаблонами, а фреймы верхнего уровня, используемые для представления этих шаблонов, называются фреймами классов.

3.2. Особенности фреймового представления знаний.

В качестве единицы представления знаний используется фрейм.

Иерархичность структуры единиц представления знаний.

Различие степени абстракции единиц знаний.

Возможность комбинации декларативных и процедурных знаний.

Основные свойства фреймов.

Базовый тип (базовый фрейм) – с его помощью запоминаются наиболее важные компоненты исследуемого объекта. На основании базовых фреймов строятся фреймы для новых состояний исследуемого объекта. При этом каждый фрейм содержит слот, оснащенный указателем подструктуры, который позволяет различным фреймам совместно использовать одинаковые части;

Процесс сопоставления – в ходе его проверяется правильность выбора фрейма. Вначале в соответствии с текущей целью делается попытка подтверждения релевантности некоторого базового фрейма(в т.ч. с помощью подфреймов), при подтверждении процесс сопоставления завершается. В противном случае для слота, в котором возникла ошибка, делается попытка присваивания надлежащего значения с учетом наложенных слотом ограничений. В случае неуспеха управление передается другому надлежащему фрейму из рассматриваемой системы, затем –соответствующему фрейму из другой фреймовой системы и т.д. пока не произойдет успешного сопоставления. В противном случае для данной фреймовой системы рассматриваемая поисковая задача решения не имеет и требуется либо переформулировать текущую цель, либо пересмотреть содержательную часть фреймовой модели, включая накладываемые слотами ограничения;

Иерархическая структура. Ее особенность заключается в том, что информация об атрибутах, которую содержит фрейм верхнего уровня, совместно используется всеми фреймами нижних уровней, связанных с ним;

Межфреймовые сети – образуются путем соединения фреймов, описывающих объекты с небольшими различиями, с использованием указателей различия;

Значение по умолчанию – под ним понимается значение слота, полученное путем распределения человеком-экспертом конкретных значений между терминальными слотами фрейма. Выводы, получаемые на основании значений по умолчанию,

называются выводами по умолчанию. С их помощью можно восполнить недостатки изначально заданной информации. Как правило, когда используется подобный способ вывода, предполагается наличие эффективно действующих межфреймовых сетей и демонов;

Отношения “абстрактное-конкретное” и “целое-часть”. Отношения “абстрактное-конкретное”(IS_A) характерны тем, что на верхних уровнях иерархии расположены более абстрактные объекты, а на нижних уровнях – более конкретные объекты, причем объекты нижних уровней наследуют атрибуты объектов верхних уровней. Отношение “целое-часть”(PART_OF)касается структурированных объектов и показывает, что объект нижнего уровня является частью объекта верхнего уровня. В отношениях этого типа нельзя использовать наследование атрибутов.

Структура данных фрейма.

Имя фрейма – присваиваемый фрейму идентификатор, для заданной фреймовой системы имя фрейма должно быть уникально;

Имя слота – присваиваемый слоту идентификатор. Слот должен иметь уникальное имя во фрейме, к которому он принадлежит;

Указатели наследования – с их помощью определяется, какую информацию об атрибутах слотов фрейма верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня. Типичные указатели наследования : U (Unique, уникальный) – слот наследуется, но данные в каждом фрейме могут принимать любые значения; S (Same, такой же)–наследование тех же значений данных; R (Range) –значения слотов фрейма нижнего уровня должны находиться в пределах, указанных значениями слотов фрейма верхнего уровня; O –выполняет одновременно функции указателей U и S, при отсутствии указаний работает как S;

Указание типа данных: FRAME (указатель на другой фрейм), INTEGER (целый), REAL (действительный), BOOL (булевский), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение);

Значение слота – должно совпадать с указанным типом данных этого слота. Кроме того, должно выполняться условие наследования;

Демон – особая разновидность присоединенной процедуры, которая запускается при выполнении некоторого условия, определяемого значением соответствующего слота. Пример: демон IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено, IF-ADDED запускается при подстановке в слот значения, IF-REMOVED – при стирании значения слота;

Присоединенная процедура является программой процедурного типа, которая является значением слота и запускается по сообщению, переданному из другого фрейма. Демоны и присоединенные процедуры являются процедурными знаниями из представляемых фреймовой моделью.

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.

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).

ПРИМЕРЫ фреймовых моделей представления знаний

Пример №1: фреймовой модели иерархического типа представлен на рисунке ниже:

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.

Формально фрейм – это тип данных вида:

N – имя объекта;

S1 – множество слотов, содержащих факты, определяющие декларативную семантику фрейма;

S2 – множество слотов, обеспечивающих связи с другими фреймами (каузальные, семантические и т. д.);

S3 – множество слотов, обеспечивающих преобразования, определяющие процедурную семантику фрейма.

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. Пример № 2: М.А. Чошанов предлагает использовать проблемные фреймы, основные элементы которых и их условные обозначения приведены на рисунке.

Пример № 3: Фреймовая модель. (Фрейм – рамка, остов, скелет, минимальное описание явления). Фрейм в технологии обучения – это единица представления знаний, заполненная в прошлом, детали которой при необходимости могут быть изменены согласно ситуации.

Обычно фрейм состоит из нескольких ячеек (слотов), каждый из которых имеет свое назначение. При помощи фреймовой модели можно «сжимать», структурировать и систематизировать информацию в виде таблиц, матриц. Примером данной модели может служить фрейм книги, представленный на

рисунке:

ВВЕДЕНИЕ

ЗАКЛЮЧЕНИЕ

Наименование: КНИГА

Атрибуты

ПЕРЕПЛЕТ

ОГЛАВЛЕНИЕ

ТИТУЛЬНЫЙ ЛИСТ

ТЕКСТ

РАЗДЕЛЫ

АННОТАЦИЯ

ИЛЛЮСТРАЦИИ

ПРИЛОЖЕНИЕ

ОБЪЕМ

ФОРМАТ

...

Пример№ 4: Описание с помощью фреймов понятия письменного отчета. В виде семантической сети «отчет» можно представить следующим образом.

Приведем представление понятия отчёта с помощью фреймов.

ЗАКЛЮЧЕНИЕ

Главная особенность применения языка представления знаний фреймами – простота написания программ для решения интеллектуальных проблем. Высокая универсальность языка позволяет с помощью присоединенных процедур эффективно реализовывать любую программу управления выводом, но это требует определенной квалификации разработчика. Это позволяет квалифицировать язык представления знаний фреймами как язык, ориентированный на специалистов в области искусственного интеллекта.

Поскольку большинство систем, ориентированных на решение сложных проблем, содержат в качестве составляющей продукционную систему, то в ряде случаев допускается использование продукционных правил в качестве типа данных фрейма.

Весьма серьезной и сложной задачей является обнаружение семантических противоречий во фреймовой системе. Поиск эффективных научных подходов для ее решения связан с исследованием по машинному обучению.

Применение модели знаний на основе фреймов и фреймовых сетей при реализации инструментальных средств позволяет описывать достаточно широкий класс технологических объектов. Используемая унифицированная структура фреймов и небольшое число стандартных правил обработки позволяют описывать более сложные правила обработки и делают систему гибкой в построении различных объектно-ориентированных структур знаний. Построенная модель информационной структуры и соответствующая модель выбора удовлетворяют поставленным требованиям, характерным для новых информационных технологий, а именно: проблемно-независимое представление информации, обеспечение создания и корректировки базы знаний, данных и фактов на информационном уровне, поддержка выбора на качественном и количественном уровне, возможность описания на информационном уровне различных правил выбора, достаточно простой перевод модели в машинное представление.