

**Российский государственный социальный
университет**

Итоговый контроль

по дисциплине «Интеллектуальные информационные системы»

ФИО студента	Хиоара Валерия Ивановна
Направление подготовки	
Группа	

Москва

Введение

Проблема представления знаний в компьютерных системах - одна из основных проблем в области искусственного интеллекта. Решение этой проблемы позволит специалистам, не обученным программированию, непосредственно на языке “деловой прозы” в диалоговом режиме работать с ЭВМ и с ее помощью формировать необходимые решения. Таким образом, решение проблемы представления знаний в компьютерных системах позволит существенно усилить интеллектуальную творческую деятельность человека за счет ЭВМ.

Актуальность данной темы заключается в том, что содержимое памяти ЭВМ не равносильно человеческому знанию, которое является гораздо более сложным феноменом, но может служить удобной для коммуникации моделью этого знания. Этот принцип моделирования профессиональных знаний лежит в основе экспертных систем.

Каждая из интеллектуальных систем соотносится с определенной частью реального мира — сферой деятельности человека, выделенной и описанной в соответствии с некоторыми целями и называемой предметной областью. Описание предметной области представляет собой совокупность сведений:

- а) обо всех предметах — объектах, процессах и явлениях, выделенных с точки зрения рассматриваемой деятельности;
- б) об отношениях между выделенными предметами и/или их частями;
- в) обо всех проявившихся и возможных взаимодействиях между предметами, их частями и отношениями, возникших в результате осуществления деятельности человека.

Сетевые модели

В основе этой модели лежит конструкция, которая называется семантической сетью, которая может быть описана на языке теории множеств:

$$H = \langle I, C_1, C_2, \dots, C_n, \Gamma \rangle$$

I – множество информационных единиц

C_1, C_2, \dots, C_n – множество типов связей между I (информационными единицами)

Γ – множество отображений между информационными единицами

В зависимости от типов связей, между информационными единицами, которые используются в данной модели, различают следующие типы сетей:

1. Классифицирующие сети – применяют отношение структуризации;
2. Функциональные сети – функциональные отношения между информационной единицей. Примером является вычисление, поэтому они могут выступать еще как вычислительные;
3. Сценарии.

Часто используются казуальные отношения между информационными единицами (причинно-следственные отношения). Кроме того, могут встречаться отношения следующих типов:

1. средство – результат;
2. орудие – действие.

Если в сетевой модели применяют отношения всех типов, то такую сеть называют семантической.

Пример представления декларативных знаний с помощью сетевой модели: «Слева от станка расположен приемный бункер, расстояние до него = 2м. Справа от станка находится бункер готовой продукции, находящийся рядом со станком (0м.) Робот перемещается прямо параллельно станку и бункерам на расстоянии 1м.»

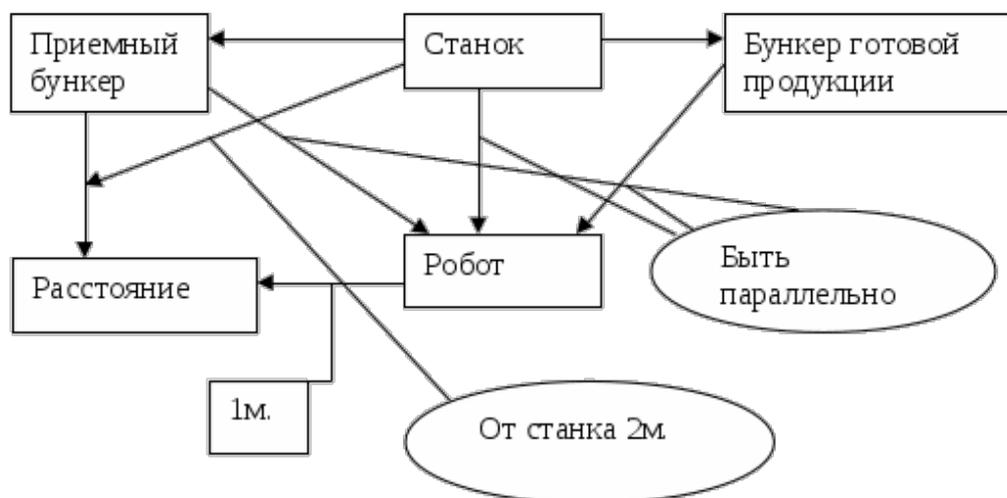


Рисунок 1 – Расположение станка по отношению других объектов.

Преимущество сетевых моделей:

1. Наглядность;
2. Понятность;
3. Удобство для представления в компьютере.

Недостатки сетевых моделей:

1. По мере роста сложности модели теряется её наглядность;
2. Область применения сетевых моделей ограничена.

IDMS - база данных, основанная на сетевой модели

Сокращение от Integrated Database Management System , в первую очередь сетевая модель (CODASYL) система управления базами данных для мэйнфреймов . Впервые он был разработан в В.Ф. Goodrich и позже проданный компанией Cullinane Database Systems (переименованной в Cullinet в 1983 году). С 1989 года продукт принадлежит Computer Associates (ныне CA Technologies), которая переименовала его в Advantage CA-IDMS, а позже просто в CA IDMS .

IDMS организует свои базы данных в виде серии файлов. Эти файлы отображаются и предварительно форматируются в так называемые области . Области разделены на страницы, соответствующие физическим блокам на диске. Записи базы данных хранятся в этих блоках.

Администратор базы данных выделяет фиксированное количество страниц в файле для каждой области. Затем администратор базы данных определяет, какие записи должны храниться в каждой области, и подробности того, как они должны храниться.

IDMS перемежает специальные страницы распределения пространства по всей базе данных. Эти страницы используются для отслеживания свободного места, доступного на каждой странице в базе данных. Чтобы снизить требования к вводу-выводу, свободное пространство отслеживается для всех страниц только тогда, когда свободное пространство для области падает ниже 30%.

Для хранения записей в базе данных IDMS доступны четыре метода: прямой, последовательный, CALC и VIA. Версия Fujitsu / ICL IDMSX расширяет это с помощью еще двух методов: Page Direct и Random.

В прямом режиме ключ целевой базы данных указывается пользователем и сохраняется как можно ближе к этому ключу БД, при этом фактический ключ БД, на котором хранится запись, возвращается прикладной программе.

Последовательное размещение (не путать с индексированным последовательным) просто помещает каждую новую запись в конец области. Этот вариант используется редко.

CALC использует алгоритм хеширования, чтобы решить, где разместить запись; затем хеш-ключ обеспечивает эффективное извлечение записи. Вся область CALC предварительно отформатирована, каждая с заголовком, состоящим из специальной записи «владельца» CALC. Алгоритм хеширования определяет номер страницы (по которому можно определить

адрес физического диска), а затем запись сохраняется на этой странице или как можно ближе к ней и связывается с записью заголовка на этой странице с помощью CALC. задавать. Записи CALC связаны с записью владельца CALC страницы с помощью единого списка ссылок (указателей). Владелец CALC, расположенный в заголовке страницы, таким образом, владеет набором всех записей, которые нацелены на его конкретную страницу (независимо от того, хранятся ли записи на этой странице или, в случае переполнения, на другой странице).

CALC обеспечивает чрезвычайно эффективное хранение и извлечение: IDMS может извлекать запись CALC в операциях ввода-вывода 1.1. Однако этот метод плохо справляется с изменениями значения первичного ключа, и требуется дорогостоящая реорганизация, если количество страниц необходимо увеличить. Обходной путь - расширить область, а затем запустить прикладную программу, которая последовательно сканирует область для каждой записи CALC, а затем использует команду MODIFY для обновления каждой записи. Это приводит к тому, что каждая запись CALC подключается к набору CALC для правильной целевой страницы, рассчитанной для нового диапазона страниц области. Обратной стороной этого метода является то, что исчезающе мало записей CALC теперь будет на их целевых страницах, а навигация по набору CALC каждой страницы, вероятно, потребует множества операций ввода-вывода. В результате рекомендуется использовать этот обходной путь только в экстремальных обстоятельствах, так как производительность снизится.

Размещение VIA пытается сохранить запись рядом с ее владельцем в определенном наборе. Обычно записи группируются на той же физической странице, что и владелец. Это приводит к эффективной навигации, когда доступ к записи осуществляется путем следования установленной взаимосвязи. (VIA позволяет хранить записи в другой области IDMS, так что они могут храниться отдельно от владельца, но при этом оставаться кластеризованными вместе для повышения эффективности. В IDMSX они также могут быть смещены от владельца на заданное количество страниц).

Page Direct (только IDMSX) похож на режим Direct, однако указывается номер целевой страницы базы данных, и запись подключается к цепочке CALC для этой страницы.

Случайно (только IDMSX) назначает номер целевой страницы экземпляру записи, когда она сохраняется с использованием алгоритма CALC (при этом либо используется ключ в записи, либо, в случае случайного выбора без ключа, используется дата & время хранения в качестве начального числа для алгоритма CALC).

Наборы обычно хранятся в виде связанных списков с использованием ключа базы данных в качестве указателя. Каждая запись включает прямую ссылку на следующую запись; разработчик базы данных может выбрать, включать ли указатели владельцев и предыдущие указатели (если они не указаны, навигация в этих направлениях будет медленнее).

Некоторые версии IDMS впоследствии включали возможность определять индексы: либо индексы записей, позволяющие находить записи, зная вторичный ключ, либо индексы набора, позволяющие извлекать элементы набора по значению ключа .

Записи прямого и случайного размещения страниц IDMSX обычно используются вместе с индексами записей, как описано выше. Сами индексы подчиняются правилам размещения, либо Direct (что на самом деле означает «CALC с использованием идентификатора индекса в качестве ключа») или CALC. Википедия site:wiki5.ru