

Значения – это значения из какого-нибудь множества, которые могут принимать свойства.

Семантические сети могут использоваться для контроля за энергопотоками, для технического диагностирования технологического процесса. Пример семантической сети представлен на рис. 1.

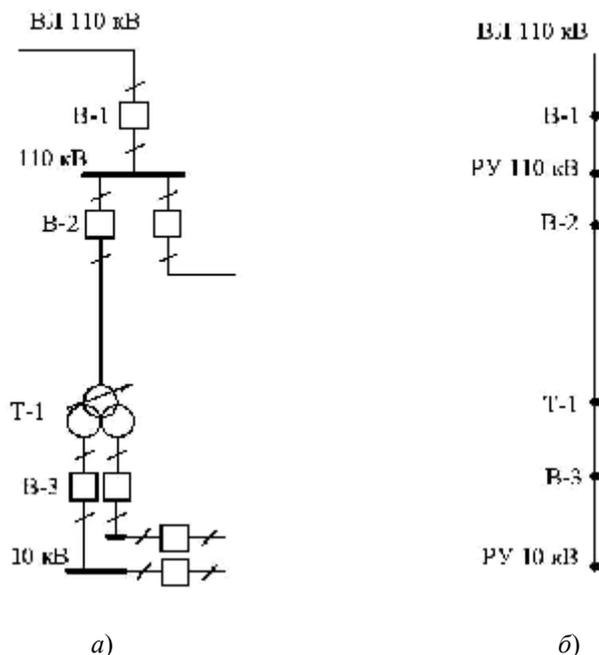


Рис. 1. Фрагмент схемы электроснабжения (а), представленный в виде семантической сети (б)

Хотя семантические сети могут быть очень полезны в представлении знаний, они имеют ограничения, такие, как отсутствие стандартов наименований направленных линий. Дополнительной проблемой является наименование узлов.

Для представления определенного знания в семантических сетях должны быть строго определены имена направленных линий и узлов. Естественно, такие же проблемы могут возникать и в языках программирования.

Еще одна проблема – стремительный рост числа просматриваемых узлов, особенно если ответ на запрос отрицательный. Вот почему для того, чтобы получить отрицательный результат на запрос, многие или все направленные линии сети должны быть найдены [2].

Реляционные базы знаний и данных

Реляционная база данных (БД) – БД, в которой данные организованы в виде таблиц, каждая из которых отображает *отношение (relation)* между объектами.

В реляционной модели данных информация хранится в одной или нескольких связанных таблицах. Отдельная таблица обычно представляет совокупность (группу) реальных либо некоторых абстрактных объектов, либо событий одного типа. Каждая запись в таблице идентифицирует один объект группы. Таблица состоит из строк и столбцов, называемых *записями* и *полями* соответственно. Свойства таблицы:

1) каждый элемент таблицы представляет собой один элемент данных, т. е. группа значений в одном столбце одной строки недопустима;

2) все столбцы в таблице однородные. Это означает, что элементы столбца имеют одинаковую природу. Столбцам присвоены имена;

- 3) в таблице нет двух одинаковых строк;
 4) порядок размещения строк и столбцов в таблице может быть произвольным.

Таблицы, обладающие такими свойствами, являются точным прообразом математического двумерного множества – отношения. Но эти два понятия не эквивалентны. Отношение – это абстрактный математический объект, а таблица – это конкретное изображение этого абстрактного объекта. В отношении строки и столбцы могут быть неупорядочены, в таблице же строки упорядочены сверху вниз, а столбцы – слева направо. Строки в таблице могут повторяться, а в отношении – нет.

В реляционной модели каждая строка таблицы уникальна. Это обеспечивается применением ключей, которые содержат одно или несколько полей таблицы. Ключи хранятся в упорядоченном виде, обеспечивающем прямой доступ к записям таблицы во время поиска. Связь между таблицами осуществляется посредством значений одного или нескольких совпадающих полей (преимущественно ключевых).

В реляционной БД взаимоотношение между объектами (сущностями) определяет связи между таблицами. Поддерживаются связи четырех типов: «один к одному», «один ко многим», «много к одному» и «многие ко многим».

«Один к одному». Связь «один к одному», что означает, что каждой записи из первой таблицы соответствует одна и только одна запись из другой таблицы. Между двумя ключевыми полями может существовать только связь «один к одному».

«Один ко многим» и «много к одному». Связь «один ко многим» означает, что каждой записи из первой таблицы может соответствовать одна либо много записей из другой таблицы.

Подчеркнем, что связь устанавливается на основе значений совпадающих полей, а не их наименований. Таким образом, если связь устанавливается между ключевым полем одной таблицы и неключевым полем второй таблицы, то это будет связь типа «один ко многим».

«Много ко многим». Эта связь возникает между двумя таблицами, когда:

- каждой записи из первой таблицы может соответствовать одна либо много записей из другой таблицы;
- одной записи из второй таблицы может соответствовать одна либо много записей из первой таблицы [2].

Реляционный принцип организации БД можно использовать при учете различных приборов контроля, при записи паспортных данных различного оборудования, при составлении планов ППО и ППР и т. д. Пример реляционной БД представлен в табл. 1.

Таблица 1

Пример реляционной базы данных

№ п/п	Наименование оборудования	$P_{\text{сум}}$, кВт	Процент от $P_{\text{уст. сумм}}$	$W_{\text{год}}$, тыс. кВт · ч	Процент от $W_{\text{год}}$
1	Подготовительный цех	20226,5	35,60	11676,2	22,59
2	Каландровый цех	8850,6	15,58	6716	13,00
3	Цех вулканизации	2392,5	4,21	5791,6	11,21
4	Сборочный цех	10298,5	18,12	6114,0	11,83
5	Цех ПВХ	12742,4	22,43	20335,1	39,35
6	Цех ТПС	2052,7	3,61	669,3	1,30
7	Административно-бытовой корпус	257,6	0,45	374,4	0,72
	<i>Итого по заводу</i>	56821	100	51677	100

Фреймовые модели

Фрейм – минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса.

Теорию фреймов опубликовал в 1975 г. М. Минский. Эта теория относится к психологическим понятиям и касается способов понимания того, как мы воспринимаем (видим, слышим) явления, процессы, объекты и т. п.

В основе теории фреймов лежит восприятие фактов посредством полученной извне информации о некотором явлении в сопоставлении с уже имеющимися данными, накопленными опытным путем или полученными в результате вычислений. Когда человек попадает в новую ситуацию, он вызывает из своей памяти основную структуру, называемую фреймом. *Фрейм* (рамка) – это единица представления знаний, запомненная

в прошлом, детали которой могут быть изменены согласно текущей ситуации.

Формально над фреймом обычно понимают структуру следующего вида:

$$f[\langle v_1, q_1 \rangle, \langle v_2, q_2 \rangle, \dots, \langle v_n, q_n \rangle],$$

где f – имя фрейма; пара $\langle v_i, q_i \rangle$ – i -й слот, где v_i – имя слота и q_i – его значение.

Каждый элемент фрейма – *слот* (от англ. *slot* – щель) – содержит определенное свойство объекта или понятия, в слоты вносятся значения, характеризующие данные свойства [3]. Слоты могут быть пустыми и заполняться в процессе активизации фрейма в соответствии с определенными условиями.

Фрейм состоит из произвольного числа слотов, среди которых имеются системные слоты и слоты, определяемые пользователем. Каждый слот характеризуется определенной структурой и уникальным именем внутри данного фрейма. В качестве системных могут быть определены, например, следующие слоты: указатель фрейма-родителя, указатель прямого дочернего фрейма, пользователь фрейма, дата определения фрейма

и его последней модификации, а также некоторые другие. Системные слоты используются при редактировании БД и управлении выводом.

Одна из особенностей систем организации знаний в форме фреймов – это возможность задавать в слотах указатели на другие фреймы (табл. 2). В данном случае значения «подшипник № 307» во фрейме «насос ЦН-104» отсылает программу

к области памяти компьютера, в которой хранятся значения о подшипнике № 307: диаметр внутреннего кольца, число тел качения и др. При разработке программных средств используют фреймы знаний и фреймы данных. Фреймы знаний описывают общие понятия. Фреймы данных содержат некоторые фактические данные.

Фрейм можно представить в виде таблицы (отношения), строки которой соответствуют слотам фрейма, а столбцы – атрибутам (характеристикам) объекта. Название таблицы является именем фрейма, и оно уникально. Каждый слот содержит следующие атрибуты: имя слота, указатель наследования, тип данных, значение слота, демон.

Тип данных определяет, что слот либо имеет численное значение, либо является именем другого фрейма. К типам данных относятся: FRAME (указатель); INTEGER (целый); REAL (действительный); BOOL (булев); LISP (присоединенная процедура); TEXT (текст); LIST (список); TABLE (таблица); EXPRESSION (выражение) и др.

Значение слота соответствует указанному типу данных этого слота; кроме того, должно выполняться условие наследования. Значениями слотов могут быть имена других фреймов, что обеспечивает связь (вложенность) между фреймами.

Демон – автоматически запускаемая процедура при обращении к слоту и выполнении некоторого условия. Например, демон типа 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 может использоваться в качестве значения слота и запускаться по сообщению, переданному из другого фрейма. Когда мы говорим, что фреймы как модели представления знаний объединяют декларативные и процедурные знания, то считаем демоны и присоединенные процедуры процедурными знаниями [2].

Таблица 2

Фреймовая модель хранения данных

Насос ЦН-104		Параметры подшипника № 307	
ЦН-104	Центробежный насос	d	35 мм
Электродвигатель	$2p = 2$ $2p = 37$	D	80 мм
Зав. номер насоса	187375		Z_m
Подшипник э/д	№ 307	r _ш	7,145 мм
Подшипник насоса	нет		$R_{вн}$
Суммарная наработка	12000	F _ш	25,57 Гц
Обороты	2850		–
Место установки	Исх. № 2	–	–

Представление знаний в виде эталонов

В техническом диагностировании без **эталонов** не обойтись. Чтобы распознать предаварийную ситуацию или отказ, машина должна сопоставить поступившие признаки с хранящимися в ее памяти моделями и эталонами. Для диагностирования ЭМ, применяют эталоны исправных и изношенных механизмов (рис. 2–4).

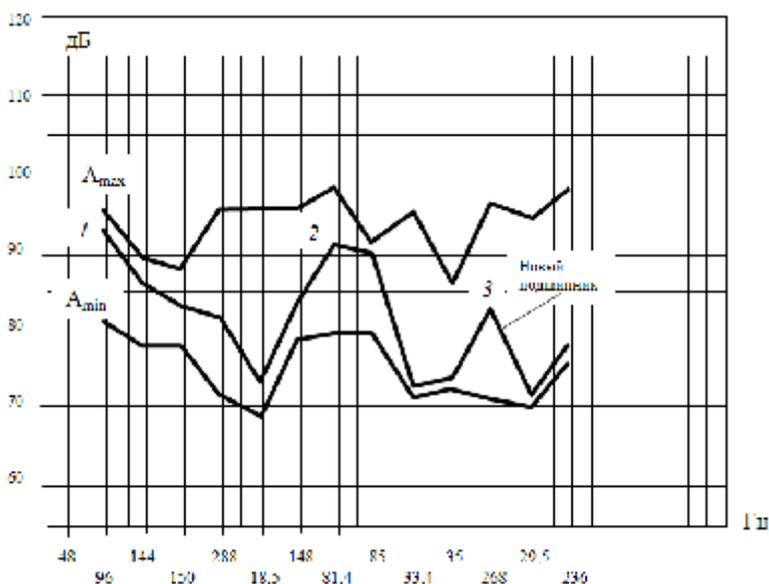


Рис. 2. Виброакустические характеристики подшипников качения с различными дефектами: 1 – овальность; 2 – дефекты тел качения; 3 – дефекты внутреннего кольца

Виброакустические характеристики подшипников качения строятся следующим образом. Берем выборку из 6–10 подшипников качения, новых и полностью изношенных. Измеряем вибрацию на информативных частотах. Для каждого подшипника на информативной частоте берем минимальное значение вибрации, т. е. делаем его условно идеальным. Так получается виброакустическая эталонная характеристика условно идеального подшипника A_{\min} . Точно так же берем максимальные значения вибрации на информативных частотах и строим A_{\max} – виброакустическую эталонную характеристику изношенного подшипника.

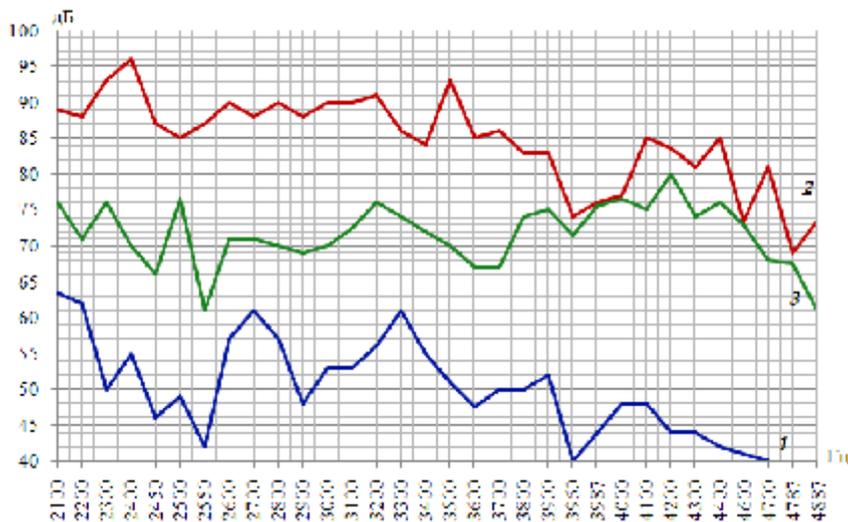


Рис. 3. Виброакустические характеристики силовых трансформаторов (ГЭЦ-3, г. Минск):

1 – огибающая минимальная; 2 – огибающая максимальная;

3 – характеристика трансформатора Т-7, фаза С

Минимальная и максимальная огибающие для трансформатора строятся подобным образом, как и для подшипников качения.

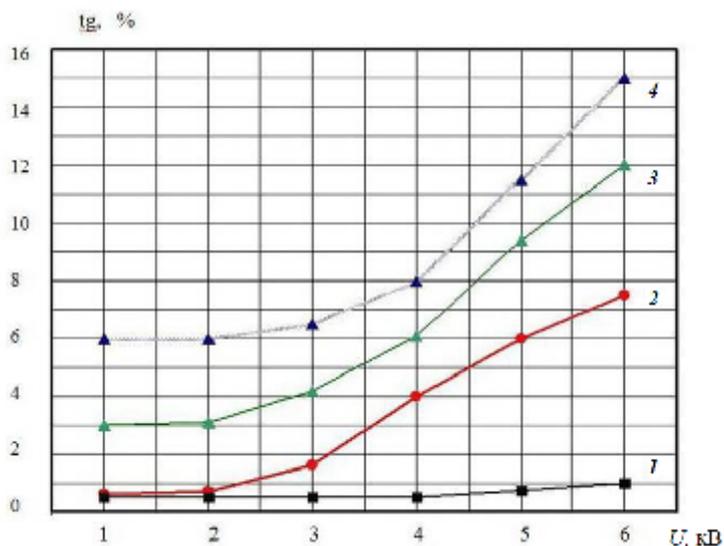


Рис. 4. Зависимость $\text{tg} \delta$ изоляции статорной обмотки электродвигателей от приложенного напряжения: 1 – эталонная характеристика двигателя в идеальном техническом состоянии; 2, 3 – эталон с начальным износом; 4 – эталонная характеристика двигателя в предаварийном состоянии

Основное достоинство представления знаний в виде эталонов заключается в том, что процесс получения знаний о предметной области автоматизируется. Задача эксперта состоит только в составлении математической модели объекта диагностирования с помощью компьютера.

Продукционные модели представления знаний

Наиболее распространенным методом представления знаний являются правила продукций, или продукционные правила. Идея этого метода принадлежит американскому логик Е. Посту (1943). Этот метод широко используется в разработке информационных систем.

Термин «продукция» принадлежит Е. Посту. В качестве продукции Пост понимал запись «если A , то B », которая трактовалась как оператор замены цепочки A цепочкой B в некотором входном слове. То есть под продукцией понимались подстановки, которые использовались в дальнейшем при описании различных уточнений понятия алгоритма. Идея продукционных правил используется в языках логического программирования (Пролог) и в современных СУБД.

Продукционная модель состоит из множества правил продукций $P_i A \rightarrow B$ (если A , то B) и множества фактов, накапливающихся по мере функционирования экспертной системы в рабочей области базы знаний [1]. База знаний состоит из:

- 1) конечного набора правил $\Pi = \{P_1, \dots, P_m\}$;
- 2) конечного набора фактов-условий $A = (a_1, \dots, a_n)$;
- 3) набора фактов-заключений $S = (S_1, \dots, S_k)$ таких, что условие применимости любого из правил Π состоит в одновременном наличии фактов a_{i1}, \dots, a_{is} .

Тогда любая продукция P_i из Π имеет вид:

$$P_i: a_{i1} \& a_{i2} \& \dots \& a_{is} \rightarrow S_m,$$

где S_m – новый факт, выведенный из фактов условий.

Существует два типа выполнения продукций: прямой и обратный. В первом случае поиск идет от левых частей продукции, т. е. проверяется условие A и актуализируются те продукции, для которых A имеет место; во втором – от изначально заданных B , по которым отыскивается необходимое для установления истинности B значение A . Два этих метода называют восходящим и нисходящим методами реализации системы продукций.

Популярность продукционных правил объясняется несколькими причинами:

1. Большинство человеческих знаний можно представить в виде правил продукций.
2. Модульность продукций позволяет добавлять в систему новые продукции без изменения прежних.
3. Продукции могут реализовать любые алгоритмы и, следовательно, любые процедурные знания.
4. Параллелизм и асинхронность продукций делают их удобной моделью вычислений, отвечающей подобным требованиям новых поколений компьютеров.

Недостаток правил продукций состоит в том, что при большом их количестве становится трудоемкой проверка непротиворечивости системы продукций, например, при добавлении новых правил.

Пример продукции: «Если на частоте 96 Гц текущее значение вибрации находится в пределах $90 \text{ дБ} < A_{\text{тек}} < 95 \text{ дБ}$, то подшипник качения имеет повышенное нарушение овальности» (рис. 2).

Таблица принятия решений (таблица решений) является табличной формой представления множества продукционных правил. Таблицы решений применяются с 1960-х гг. в различных областях, например, в задачах автоматизации проектирования технологических процессов.

Таблицу решений выгодно составлять в том случае, если множество продукционных правил содержит одинаковые условия (посылки), на основе знаний которых принимаются заключения [2].

Заключение

1. Для построения интеллектуальных систем диагностирования оборудования наиболее эффективными являются продукции, фреймовые модели, эталоны.

2. Для диагностирования технологического процесса наиболее эффективными являются семантические сети.

Литература

1. Лорьяр, Ж.-Л. Системы искусственного интеллекта : пер. с франц. / Ж.-Л. Лорьяр. – М. : Мир, 1991. – 568 с.
2. Змитрович, А. И. Базы данных и знаний : учеб. пособие / А. И. Змитрович, В. В. Апанасович, В. В. Скакун. – Минск : Издат. центр БГУ, 2007. – 364 с. : ил.
3. Поспелов, Г. С. Искусственный интеллект – основа новой информационной технологии / Г. С. Поспелов. – М. : Наука, 1988. – 280 с.

Получено 20.04.2011 г.