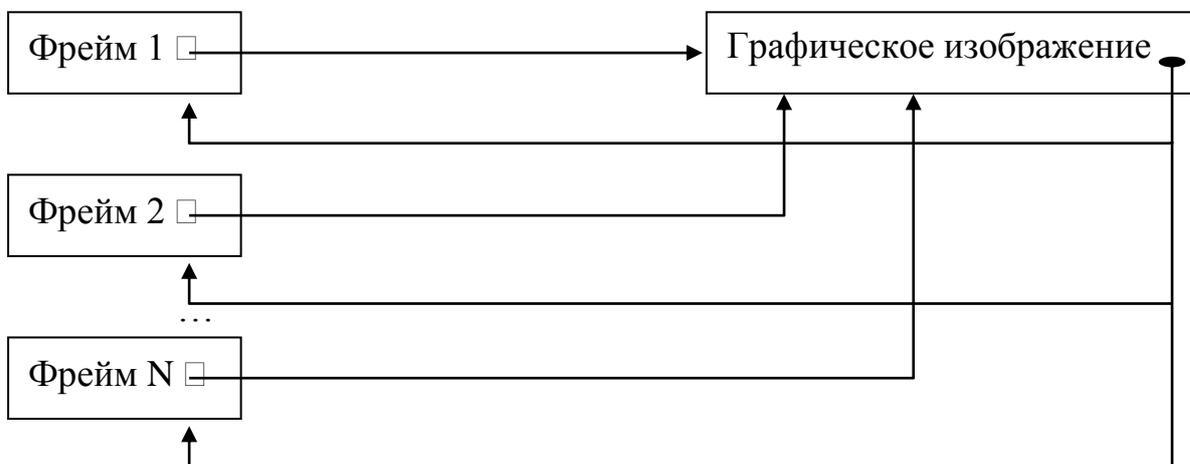


Филиппов А. Н.

ВИРТУАЛЬНОЕ СТРОКОВОЕ ПРОСТРАНСТВО ТЕХНОЛОГИЧЕСКИХ ДАННЫХ И ЗНАНИЙ



**Санкт-Петербург
2015**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Филиппов А. Н.

**ВИРТУАЛЬНОЕ СТРОКОВОЕ ПРОСТРАНСТВО
ТЕХНОЛОГИЧЕСКИХ ДАННЫХ И ЗНАНИЙ
Методы представления данных**

Учебное пособие

 **УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2015

УДК 658.512.011.56

А.Н. Филиппов. Виртуальное строковое пространство технологических данных и знаний /Учебное пособие// СПб: НИУ ИТМО, 2015. – 81 с.

Настоящее пособие предназначено для студентов специализации “Технологии приборостроения”. В пособии изложены наиболее важные темы, связанные с описанием методов организации данных и программирования алгоритмов с применением символьного представления объектов, как основы проектирования экспертных систем технологического назначения.

Адресовано студентам высших учебных заведений, обучающихся по направлению подготовки 230100 – Информатика и вычислительная техника, специальность 230100.68.17 - Интегрированные системы в проектировании и производстве.

Рекомендовано к печати Ученым советом факультета Точной механики и технологии 10-го марта 2015 г, протокол № 3.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

©Университет ИТМО, 2015

© А.Н. Филиппов, 2015

Содержание

Оглавление

Сокращения, принятые в пособии	4
1. Виртуальное строковое пространство технологических данных	7
1.1. Предпосылки представления информации в символьной форме	7
1.2. Определение виртуального строкового пространства технологических данных	7
2. Фреймы	12
2.1. Переходы как фреймы	14
2.2. Представление знаний в виде формул	15
2.3. Таблицы	17
2.4. Фреймы-терминалы	29
2.5. Фреймы-объяснения	31
2.6. Фреймы-запросы	31
2.7. Фреймы-продукции	34
2.8. Фреймы-документы	37
3. Возможные области применения ЭС при разработке САПР ТП механической обработки	39
3.1. Разумность разработки ЭС	39
3.2. Задачи, удовлетворяющие требованиям разумности разработки ЭС	40
3.3. Некоторые особенности применения ЭС в САПР ТП механической обработки	42
4. Методы работы с виртуальным строковым пространством технологических данных	47
4.1. Интерфейсный модуль	47
4.2. Информационно-поисковая система (ИПС), использующая методы экспертных систем (ЭС).	49
4.3. Общие принципы извлечения знаний в ЭС САПР ТП	60
4.4. Структура экспертной системы	65
4.5. Взаимоотношения ВСПТД и формата RUSMARC	66
5. Словарь метаданных	69
5.1 Структура словаря	72
5.2 Использование словаря модулями комплекса	74
6. Языки программирования и базы данных	75
6.1. Система управления записями	75
6.2. Интерфейс с языками программирования	75
6.3. Основные возможности системы управления записями BTRIEVE	76
ЛИТЕРАТУРА	78

Сокращения, принятые в пособии

ВСПТД – виртуальное строковое пространство технологических данных;
ГО – генератор отчетов;
ГСД – глобальный словарь данных;
ГЭС – гибридная экспертная система;
ДМВ – дедуктивная машина вывода;
ИИ – искусственный интеллект;
ИМ – интерфейсный модуль;
ИПС – информационно-поисковая система;
ЛСД – локальный словарь данных;
ЛПР – лицо, принимающее решение;
ТП – технологический процесс;
САПР ТП – система автоматизации проектирования технологических процессов;
СД – словарь данных;
СБПД – страничный буферный пул данных ;
СОЖ – смазывающая охлаждающая жидкость;
ФПЗ – формализм представления знаний;
ЭС – экспертная система;
ЯП – язык программирования.

Введение

Современный подход к созданию САПР технологических процессов (САПР ТП) предполагает использование новых информационных технологий, которые основываются на методах теории искусственного интеллекта (ИИ). Одним из важнейших направлений при разработке ИИ являются экспертные системы (ЭС).

При широком внедрении САПР ТП на предприятиях обычно сталкиваются с проблемами адаптации программного и информационного обеспечения. Эти проблемы обусловлены как традициями, так и особенностями производства на каждом конкретном предприятии. Кроме того, после того как был поднят пласт легкоалгоритмизируемых задач, перед разработчиками возникли задачи совсем другого уровня. Для них характерны следующие черты: они плохо формализованы, многие из них не обеспечены требуемым числом данных. В этих условиях возникла необходимость создания ЭС, которая является попыткой расширения области применения вычислительной техники и существенного увеличения ее возможностей как помощника человека в сфере его интеллектуальной деятельности. Однако, при этом должен быть рационально использован тот колоссальный багаж технологических знаний, который был реализован традиционными методами программирования и вполне успешно используется на предприятиях.

Таким образом, для эффективного применения ЭС в САПР ТП необходимо решить следующий ряд вопросов:

- задача организации такой структуры информации и системы программирования САПР ТП, которая позволила бы применять как традиционные методы программирования, так и методы новой информационной технологии, ядром которой является ЭС;

- задача наполнения базы знаний (БЗ) с помощью экспертов-технологов для адаптации системы к условиям предприятия без участия программистов;

- задача заимствования знаний, накопленных на различных предприятиях в процессе эксплуатации системы;

- решение вышеназванных задач, в свою очередь, требует привлечения специалистов в области инженерии знаний - дисциплины, потребность в которой возникла при разработке ЭС для обеспечения эффективного взаимодействия между специалистом в предметной области и системой в процессе формирования новых структур данных и базы знаний.

В рамках данного пособия невозможно охватить весь спектр задач, которые встают перед разработчиками САПР технологического назначения. В работе описаны инструментальные программные средства и методы построения структур данных, разработанные при создании САПР ТП "ТЕХКОМ" [18]. Предлагается базовый пакет программ и алгоритмов, который можно развивать путем подключения новых функций. Для применения данного пакета студенты должны овладеть основами программирования на языках C++ и PYTHON.

Учитывая многообразие задач, которые возникают перед разработчиками систем автоматизированного проектирования технологической подготовки производства САПР ТПП, встает вопрос выбора моделей представления баз данных (БД) и БЗ. Использование готовых инструментальных средств существенно сокращает время разработки системы. Однако, программное обеспечение подобного типа далеко не во всех случаях предоставляет возможности для обеспечения гибкости системы, отсутствуют условия для развития интерфейса, существенно ограничивается выбор формализма представления знаний (ФПЗ). Кроме того, эти факторы накладывают значительные ограничения на принцип эволюционного развития САПР ТП.

В работе [1,с.9-10,с.195] отмечаются недостатки, возникающие при организации работ по проектированию подсистем АСУП и САПР, которые можно объединить в следующие группы:

1. Отсутствие единого методического руководства разработкой математического обеспечения (МО) САПР, приводящее к нарушению системного принципа разработки МО.
2. Отсутствие обмена опытом, знаниями между технологами и математиками различных групп.
3. Трудность маневрирования, необходимость которого при разработке САПР возникает довольно часто.
4. Дублирование и параллельные разработки технологов, математиков и программистов в области создания типовых и стандартных программ обработки данных.
5. Невозможность сопоставления уровня квалификации технологов, математиков и программистов групп.

Все эти проблемы, как отмечается там же, могут быть решены лишь при едином методическом руководстве. При современном уровне разработок требуются усилия многих коллективов, внутренняя организация которых зачастую не зависит от желаний головного разработчика, и следовательно не всегда возможно единое методическое руководство. В этих условиях соглашение о едином способе доступа к информации призвано решить вопросы взаимодействия различных модулей в интегрированной системе.

Кроме того, для разработки ЭС необходимо иметь средство, которое обеспечивало бы прикладному программисту эффективную поддержку при работе с разнообразной слабоструктурированной информацией, т.е., программист пишет свою программу в предположении, что он располагает бесконечной оперативной памятью, хотя в действительности объем памяти ограничен. Для обеспечения эффективного взаимодействия различных модулей системы, исключения возможных рассогласований при трактовке одних и тех же понятий и ряда других сервисных функций необходим универсальный интерфейсный модуль и словарь метаданных (см. п.4. «Словарь метаданных»).

В данной работе рассматривается подход, основанный на специализированном представлении данных и знаний, позволяющий одновременно использовать программно-математический аппарат, имеющийся в существующих САПР ТП. Описан специальный способ организации БЗ и БД, определенный как виртуальное строковое пространство технологических данных (ВСПТД). В рамках данного представления определено множество типов фреймов, применение которых существенно упрощает описание и программирование сложных объектов. В качестве таковых могут выступать как непосредственно технологические объекты - деталь, инструмент и т.д., так и абстрактные объекты: формулы, анкеты ввода, тексты переходов и т.п. Вышеизложенные предложения получили свое развитие при разработке САПР ТП механической обработки "ТЕХКОМ" [18].

1. Виртуальное строковое пространство технологических данных

1.1. Предпосылки представления информации в символьной форме

Традиционно в языках программирования сильны возражения против использования связей между модулями в виде символьных имен. За этим взглядом стоит не только стремление к эффективности, но и убежденность в том, что имя всегда заменяет некоторый конкретный объект и имеет смысл лишь в связи с этим объектом. Этому подходу удобно следовать, пока мы остаемся в пределах математических задач, на базе которых и выросли традиционные языки программирования, но реальные ситуации, возникающие за пределами таких задач, демонстрируют необходимость более широкого использования имен, в частности, установления связи между независимо построенными программными модулями путем использования одинаковых имен. Кроме того, представление информации в символьном виде существенно расширяет возможности при создании баз знаний.

1.2. Определение виртуального строкового пространства технологических данных

1.2.1. Факты, цели, триплеты

Исходя из проведенных исследований и анализа предметной области введено понятие **фактов и целей** системы и представление их в виде триплетов, которое основывается на двухуровневом описании характеристик объектов. Под триплетом понимается специальная форма описания в символьном представлении по следующей схеме:

"объект - имя характеристики - отношение - значение – "комментарий".

Под *объектом* понимается некоторая целостность (т.е., обособленный фрагмент действительности или автономное образование), которая обладает

присущими только ей свойствами на фоне свойств среды и на фоне своей структуры (образующих объект частей и связей между ними). Любой объект, как реальность, существует только в определенной среде, как внутренняя граница в этой среде. И объект и среда - взаимообусловленные сущности, устанавливаемые в процессе перехода от одной к другой [2,с.8]. Технологические объекты как понятия уже рассматривались в работе [3]. Здесь только введено символическое представление этих понятий.

В процессе проектирования системе приходится оперировать со множеством различных данных. Одни данные вводятся в процессе диалога, например, описание заготовки и чертежа детали. Другие выбираются из базы данных, третьи получаются расчетным путем из первых. Возникает проблема такого представления данных, которое позволило бы безболезненно подключать новые модули к системе, а также расширять список участвующих в принятии решений правил.

Данные, известные системе в текущий момент, будем называть фактами. Каждый факт представляется в виде триплета $\Phi = \langle \text{Prefix}, \text{Name}, \text{Value} \rangle$, где Prefix - префикс, Name - имя параметра, Value - значение параметра. Здесь префикс обеспечивает контекстное понятие параметра, т.е. указывает на конкретный описываемый объект. Например, если L - имя параметра, указывающее в общепринятых обозначениях длину описываемого объекта, то выражение Z.L будет обозначать длину заготовки, при условии, что Z - это префикс, указывающий на объект "заготовка". Т.е., Z - это множество характеристик производственного объекта "заготовка", а Z.L - один из элементов множества Z. Третья составляющая факта "значение" всегда представляется в символической форме. При этом текстовое значение помещается в апострофы, чтобы отличить от числового. Вектор значений представляется списком через заданный разделитель. Такое представление фактов удобно для реализации правил вывода и сопутствующих объяснений. Кроме того, для обеспечения функций нечеткой логики [4] значение каждого факта может быть снабжено коэффициентом уверенности. Таким образом, формируется некоторое множество (пространство) фактов F, которое в процессе проектирования размещается в виртуальном строковом пространстве технологических данных (ВСПТД) [5]:

$$F = \cup \Phi_i, \text{ где } \Phi_i - \text{ триплет } i\text{-го факта.}$$

При взаимодействии управляющей структуры с различными модулями системы возникает необходимость получения новых фактов. Еще неизвестные системе факты будем называть целями системы. Каждая цель представляется в виде соответствующего триплета цели, у которого префикс и имя параметра организованы так, как и в триплете факта, а "значение" может быть представлено следующим способом:

Префикс.Имя=заявка;

Триплет такого типа будем называть "простой целью" (или целью). Задача системы - заменить заявку этого триплета на значение либо, используя информационно-поисковую систему (ИПС), либо расчетным или

логическим путем на основе уже имеющихся фактов, либо с помощью лица, принимающего решение (ЛПР). Получение фактов происходит как в жестко регламентированном режиме, так и с помощью управляющей структуры $\langle \text{Prefix, Name, Mode, Number} \rangle$, где Prefix - префикс, Name - имя параметра, Mode - способ достижения цели, а Number - идентификация исполняющей функции. С другой стороны, управляющая структура не всегда располагает информацией о том, какие именно факты нужны для реализации правил. Сведения об этом выдают непосредственно исполняющие правила в виде триплетов цели. Заявка может иметь в качестве своего значения ":" (двоеточие) либо ':текст', либо пустое значение. Если триплет имеет пустое значение, то это означает необязательность заполнения данного триплета. Такого рода триплет относится к триплетам с необязательной целью. Если заявка имеет значение ":", то триплет относится к разряду триплетов с обязательной целью и его заполнение обязательно в одном из трех указанных выше режимов в зависимости от контекста и уровня формализации. Если заявка триплета имеет вид ':текст', его значение будет выбираться из ВСПТД без участия ЛПР. При этом, если оно не будет найдено, то в качестве значения будет взят текст после двоеточия. Например, триплет цели (\$Z.L=:10;) перейдет в разряд триплет факта с видом (\$Z.L=10;). Это триплет с наложенным значением. Таким образом, в процессе проектирования появляется множество целей C.

$C = UC_i$, где C_i триплет i -той цели.

Кроме того, может быть два типа сложных целей:

1. Объектная (структурированная цель);
2. Составная цель.

Объектная цель запрашивает полную информацию по заданному объекту, определяемому префиксом. Например, по заявке $Z.=;$ будет получена вся информация по заготовке, имеющаяся в ВСПТД. То есть, множество наличных фактов с префиксом Z. Для объектной цели обязательного типа ($Z.=;$) по структуре заданного объекта определяется множество простых целей, каждая из которых трактуется как обязательная.

Составная цель может быть получена из некоторой фразы (фрейма^{1*}), содержащей факты, цели, и объяснения.

Например, фрейм

\$L.WOB=019; "притереть"; "поверхность" \$L.KE=;; "выдерживая требования" \$D.TT=;;

¹ Первое определение виртуального строкового пространства технологических данных (ВСПТД) дано в работе [19]. В дальнейшем в описание включается понятие "Знание" – ВСПТДЗ [18], однако, оставлена устоявшаяся аббревиатура.

* Фрейм предложен Марвином Минским [11] и представляет собой расширение фундаментального понятия списка ассоциаций. Эта структура, называемая фреймом (кадром) и представляющая собой совокупность слотов (гнезд), описывающих свойства некоторого объекта. Семантические сети представляют из себя граф, в котором вершинам соответствуют некоторые объекты, а дугам – семантические отношения между ними. Знания в виде фреймов - один из наиболее популярных способов представления семантических сетей.

содержит в себе факт, простые цели и объяснения. После заполнения целевых заявок она будет представлять собой текст перехода, готовый для передачи модулю документирования системы.

Задача системы заключается в переводе каждого триплета цели во множество фактов. В процессе проектирования триплеты группируются в *триплексные строки* в контексте "операция-переход" и помещаются в ВСПТД. В дальнейшем под *триплексной строкой* будет подразумеваться некоторое множество фактов и целей, объединённых одним ключом. А ВСПТД – это множество триплексных строк.

1.2.2. Представление графа в виде триплета

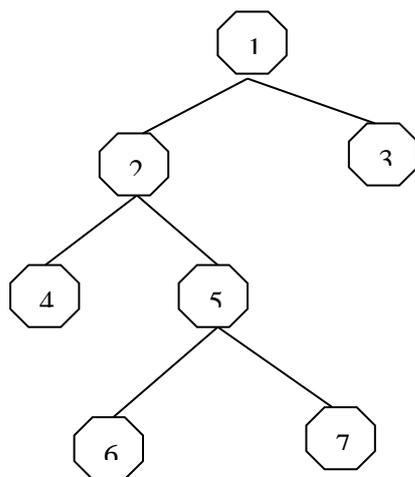


Рис. 1.1. Способ представления графа в виде триплета

В соответствии с принципами организации ВСПТД предложен следующий метод представления графов в триплексной строке. Считаем, что нам известны номера каждой вершины графа (рис. 1.1). Создаем множество пар вершин вида $\langle N_i, M_i \rangle$, где N_i – номер исходящей, а M_i – номер входящей вершины. Пусть L – описываемый объект (префикс). Тогда триплексное представление графа выглядит так:

$\$L.GRAPH = \langle N_1, M_1 \rangle, \langle N_2, M_2 \rangle, \dots, \langle N_i, M_i \rangle$;

Для представленного на рисунке графа триплет, будет иметь вид:

$\$L.GRAPH = \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 4 \rangle, \langle 2, 5 \rangle, \langle 5, 6 \rangle, \langle 5, 7 \rangle$;

На примере представления графа можно сделать вывод, что в виде триплетов можно представлять разнообразную информацию, определяемую в соответствующей предметной области.

1.2.3 Синтагмы²

Способ описания фактов в виде триплетов позволяет, в принципе, подключать описание любых отношений, вводя их через отношения имен. Например, если через префикс L - множество элементов обработки, а через имя PER - перпендикулярность, то описание перепендикулярности может выглядеть так:

$\$L.PER=(5,7),(8,10); (*)$

Это означает, что поверхность 5 перпендикулярна поверхности 7, а поверхность 8 перпендикулярна поверхности 10. Однако, при достаточно большом числе сочетаний отношений описываемых объектов этот способ не очень удобен. Поэтому эти же отношения описываются с помощью синтагм вида $A \ r \ B$, где A и B - некоторые объекты, а r - отношение между ними. Тогда отношения (*) будут выглядеть так:

$\$L5_PER_ \$L7; \$L8_PER_ \$L10; (**)$

При этом в синтагму допускается включение дополнительной информации в виде комментария, смысловое значение которой определяется из контекста отношения. Например,

$\$L5_NEPER_ \$L7 "<0.02";$

что означает неперпендикулярна поверхности 5 к поверхности 7 не более, чем 0.02.

Таким образом, в область фактов также включается множество синтагм.

В качестве первого способа представления технологических данных предложено описание в виде фрейма, в которой базовой компонентой (слотом) выступает триплет. В связи с этим следует обратить внимание на следующее: допускается применение префиксов без указания имен характеристик объекта, но недопускается применение имен характеристик без указания префиксов. То есть, L – это есть элемент обработки, но не длина.

1.2.4. Индексирование триплетов

В примере (**) приведен нумерованный список элементов обработки L1,L2 ...

² **Современный энциклопедический словарь:**

Синтагма (греч. syntagma, букв. - вместе построенное, соединенное),1) интонационно-смысловое единство, которое выражает в данном контексте и в данной ситуации одно понятие и может состоять из одного слова, группы слов и целого предложения.2) Сложный языковой знак, обычно двучленный, составленный из слов или морфем, соединенных определенным типом связи (напр., определительный).

При этом в словаре метаданных системы описывается один объект L. В том случае, когда возникает необходимость описывать несколько однотипных объектов, применяется индексирование объектов. При описании триплетов в ВСПТД принято следующее соглашение. Если объект индексирован (в данном контексте имеет номер), то префикс представляется в виде Pn, где P – базовый префикс, а n – номер объекта. Отсюда следует ограничение на описание префикса. Базовый префикс не должен заканчиваться на цифру. Рассмотрим ситуацию индексирования на примере таблицы (рис. 1.2) (Методы организации таблиц в ВСПТД подробно представлены в разделе 2.3.)

Предположим, что для работы выбраны две строки таблицы – первая и третья.

Обозначение V.OB	Конус Морзе хвостовика патрона V.NKH	Диаметр посадки втулки V.DPO	Дополнительный вылет патрона V.VI
6161-0021	3	38.0	199
6161-0022	4	45.0	222
6161-0023	4	55.0	237
6161-0024	4	65.0	257

Рис. 1.2. Патрон для нарезания резьбы плашкой

Тогда фрагмент ВСПТД с результатами поиска может выглядеть так:

... \$V1.DPO=38.8;\$V2.DPO=55.5; \$V2.NKH =3;\$V2.NKH =4;\$V1.OB='6161-0021'; \$V2.OB='6161-0023'; \$ V1.VI=199; \$V2.VI=237;...

Здесь индексы имеют значение 1 и 2 в соответствии с порядком вхождения найденных строк. С другой стороны, возможна ситуация, когда одной характеристике объекта приписывается несколько значений. В этом случае список значений определяется как вектор, значения которого записываются через заданный разделитель.

2. Фреймы

Термин *фрейм* (от англ. *frame* — "каркас" или "рамка") был предложен Марвином Минским, одним из пионеров ИИ, в 70-е годы для обозначения структуры знаний для восприятия пространственных сцен. Эта модель, как и семантическая сеть, имеет глубокое психологическое обоснование.

Фрейм — это абстрактный образ для представления стереотипа объекта, понятия или ситуации.

Интуитивно понятно, что под абстрактным образом понимается некоторая обобщенная и упрощенная модель или структура. Например, произнесение вслух слова "комната" порождает 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, мы получим уже чулан, а не комнату), но в нем есть "дырки" или "слоты"— это незаполненные значения некоторых атрибутов, например, количество окон, цвет стен, высота потолка, покрытие пола и др.

В теории фреймов такой образ комнаты называется фреймом комнаты. Фреймом также называется и формализованная модель для отображения образа.

Различают *фреймы-образцы* или *прототипы*, хранящиеся в базе знаний, и *фреймы-экземпляры*, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

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

Традиционно структура фрейма может быть представлена как список свойств:

(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)).

Существует несколько способов получения слотом значений во фрейме-экземпляре:

- по умолчанию от фрейма-образца (Default-значение);
- 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.

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 (A-Kind-Of = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т. е. переносятся, значения аналогичных слотов.

Рассматривая теорию фреймов, как один из основополагающих методов представления знаний, предложен способ представления технологических знаний в разных приложениях на основе единого механизма, позволяющего интегрировать данные при взаимодействии различных модулей системы. За основу в этом механизме взято единое представление элементарных фактов и целей (см. описание триплетов-фактов и триплетов-целей). Т.е., в качестве основной единицы представления фрейма - слота предложен триплет.

Основные группы представления знаний в виде фреймов следующие:

- фреймы-переходы и фреймы-операции;
- фреймы-формулы;
- фреймы-таблицы;
- фреймы-продукции;
- фреймы-терминалы;
- фреймы-документы;
- фреймы-объяснения;
- фреймы-запросы.

2.1. Переходы как фреймы

Рассматривая тексты переходов, как некоторые *целевые фразы*, в которых размещается переменная информация, представленная в виде триплетов цели, нетрудно заметить, что такое представление очень похоже на представление знаний в виде *фреймов*.

Знания о предметной области решаемых задач задаются на множестве фреймов, образующих систему фреймов $\{\Phi_{\text{ПО}}\}$. Каждому типу задач в $\{\Phi_{\text{ПО}}\}$ соответствует родовой фрейм (подсистема фреймов), отвечающих за решение задач данного типа. Каждый родовой фрейм посредством своих слотов определяет в общем, какие знания и умения необходимы для решения задач данного типа.

В такой интерпретации каждому возможному переходу соответствует некоторый фрейм. Например, один из фреймов *точение* выглядит так:

"точить "\$L.NM=:" выдерживая размеры "\$L.L=;; \$L.D=;;

Таким же образом можно рассматривать фреймы *сверление, фрезерование, шлифование* и т.п..

При вызове очередного фрейма из него извлекаются все цели, проверяется наличие соответствующих триплетов в ВСПТД, и при нахождении подставляются вместо целей. Факторизация оставшихся целей обеспечивается через управляющую структуру. При заведении фреймов в базу знаний слоты могут снабжаться дополнительной (*превентивной*) информацией, которая может быть полезна для функций заполнения данного слота.

Например, запись

\$E.TR=: <\$E.KI='11230421';>; означает, что для заполнения слота \$E.TR (типоразмер режущего инструмента) при обращении к ИПС ИНСТРУМЕНТ следует использовать конструктивный код, заданный в угловых скобках, с помощью которого осуществляется выход на таблицу, в которой будет производиться поиск инструмента. Кроме того, каждый слот может снабжаться, как и всякий триплет цели, значением по умолчанию. В качестве превентивной информации в фрейме может присутствовать ссылка

Многочисленные литературные источники по технологии машиностроения, которые здесь трудно было бы перечислить, указывают на то, что такого рода выражения содержатся почти в каждом виде работ, начиная от расчетов стойкости инструмента, режимов резания и кончая расчетом времени на личные надобности. Эти формулы носят как теоретический, так и эвристический характер. В процессе эксплуатации некоторые из них подвергаются корректировке, но будучи "защитными" в программы, не могут быть исправлены без участия программиста. В этих условиях формулы удобно хранить вне программ, обеспечивая возможность их корректировки силами эксперта в любой удобный для него момент. Известные прикладные пакеты программ для работы с электронными таблицами LOTUS-1-2-3, Суперкалк позволяют в графы таблиц вводить формулы, которые обычно связаны только со значением реквизитов из других граф этой же таблицы. В контексте применения ВСПТД разработано соответствующее представление формул, методы их хранения и способы реализации.

Фрейм-формула – арифметическое выражение, представленное в виде символической строки, в которой значения переменных заданы в виде триплетов целей и постоянных величин (заданных коэффициентов).

Синтаксис формулы соответствует синтаксису арифметического выражения в поисковом образе.

Например: Запись $X = \$A * \$Y ** \$K$ с

ответствует формуле $(X = \$A \cdot \$Y^{\$K})$, а запись

$\$K.FUN = SIN(\$E.FI) / COS(\$E.ALF)$

понятна без дополнительного объяснения.

Такое представление позволяет рассматривать формулу, как некоторую текстовую строку, которую можно хранить в базе знаний как и другие типы знаний.

Формула для расчета нормы расхода для пруткового материала с известным удельным весом:

$\$Z.RNR = '0.785 * \$M.D * \$Z.L * M.VH * \$M.KTOTe \$Z.KDZ * 10 ** 6'$;

Формула для расчета нормы расхода для плоских материалов с известным весом на 1 кв.м:

$$N = \frac{B \times L \times K_1 \times K_2}{n \times 10^6}$$

где N – норма расхода;

B – ширина листа;

L – длина листа;

K_1 – весовая характеристика материала;

K_2 – коэффициент отходов в заготовительном производстве;

n – количество деталей из заготовки.

Каждому из этих параметров присвоим имя реквизита, если имя реквизита не было объявлено ранее в словаре, то объявим его. Формат этих имен не должен быть текстовым, он может иметь один из следующих видов:

99...99 или 9(n) - формат целого поля, где n – количество цифр в реквизите.

9...9V9...9 или 9(n)V9(n) - формат вещественного поля, где V разделитель между целой и десятичной частью.

S9...9 или S9...9V9...9 - формат поля со знаком.

В рассматриваемом примере соответствие будет выглядеть следующим образом:

параметру N соответствует имя реквизита Z.NRM;

параметру B соответствует имя реквизита Z.B;

параметру L соответствует имя реквизита Z.L;

параметру K₁ соответствует имя реквизита M.VH;

параметру K₂ соответствует имя реквизита M.KTOT;

параметру n соответствует имя реквизита Z.KD;

Тогда

$$\$Z.NRM=\$Z.B*\$Z.L*\$M.VH*\$M.KTOT/(\$Z.KD*10**6);$$

Аналогично строится формула для расчета нормы расхода материала для пруткового материала с известным удельным весом:

$$\$Z.RNR='0.785*\$M.D*\$Z.L*M.VH*\$M.KTOTe\$Z.KDZ*10**6';$$

Принцип хранения формул в базе знаний соответствует общепринятому в системе методу хранения знаний, представляемых в строковой или табличной форме. Формулы разбиваются на группы, группы укладываются в виртуальные строки в ВСПТД.

2.3. Таблицы

Анализ состава технологической базы данных показал, что большую часть информации можно представить в виде сравнительно небольших таблиц, что дает возможность просматривать их в памяти ЭВМ за 2-5 обращений к диску. Принцип применения решающих таблиц имеет давнюю историю. Наиболее предпочтительным для задач САПР ТП является представление таблиц в виде фреймов. Представление таблиц как фактов является в ВСПТД предметом особого рассмотрения в связи с их разнообразной структурой и большим числом вхождений в базу технологических данных. Каждая таблица содержит несколько экземпляров строк информации одной структуры. При этом в структуре таблицы описывается список реквизитов с идентификацией также как и в триплете факта и с указанием формата значения. Такое представление позволяет передавать в раздел технологических данных ВСПТД сразу же в форме триплетов. Для обеспечения различия табличных значений и значений в ВСПТД в правилах вывода принято соглашение, по которому триплеты из области фактов снабжаются символом \$. При считывании тела таблицы в ВСПТД одновременно переписывается и ее структура. Другими словами, таблица представляет из себя виртуальную строку и является составной частью ВСПТД.

Таблица определяется как: $T=(IT,ST,FT,CT,HT,VT)$, где

T - таблица;

IT - имя таблицы;

ST - структура таблицы;

FT - фрейм-запрос;

CT – дополнительные характеристики таблицы;

HT - заголовок таблицы;

VT – тело таблицы.

Имя таблицы - это ключ строки в ВСПТД, в которой располагается таблица. Структура таблицы - представляет текстовую строку, в которой через точку с запятой перечисляются имена используемых реквизитов с описанием формата и названия граф при отображении. Признаком конца такой строки служат две литеры "косая черта", пробелы игнорируются. Например, описание

E.NUM 999; E.TR X(12); E.L 999V99; E.D 99 //

говорит, что каждая строка таблицы состоит из четырех реквизитов E.NUM, E.TR, E.L и E.D, расположенных в указанном порядке, причем реквизиты E.NUM и E.D - целые длиной 3 и 2 символа, E.TR - текстовая строка из 12 литер, а реквизит E.L - вещественное с тремя знаками до запятой и двумя - после запятой.

Фрейм-запрос - это правило поиска информации в таблице (см. п.3.6).

В дополнительных характеристиках таблицы содержится информация, общая для всех строк таблицы.

В заголовке указывается название таблицы.

Тело таблицы содержит непосредственно данные.

Большое количество таблиц в системе не позволяет разработчику предусмотреть все их разнообразие в процессе разработки. В связи с этим возникает задача создания такого программного аппарата, который позволяет эксперту-технологу самому создавать и включать в работу САПР ТП новые таблицы. Для этого предложен механизм, при котором технологу достаточно ответить на несколько простых вопросов. Например, если он хочет создать новую таблицу в базе данных режущего инструмента, то ему будет задано три группы вопросов. Первая группа вопросов касается имени таблицы, который представляет из себя конструктивно-технологический код инструмента. После формирования имени таблицы, в случае отсутствия ее в базе, технологу предлагается указать на экране дисплея из предложенного множества параметров, описывающих весь режущий инструмент (см. словарь), те реквизиты, которые он желает иметь в создаваемой таблице. При этом, ему предоставляется право включать в словарь новые параметры. После этих действий автоматически формируется форма таблицы для отображения ее на экране дисплея, и эксперт получает возможность вводить и корректировать данные. И, наконец, на основании третьей группы вопросов формируется фрейм-запрос (п.2.6), наличие которого позволяет включить таблицу в работу системы.

Все таблицы хранятся в табличных базах. Файл с табличной базой содержит в себе группу таблиц с различной структурой и представляет собой некоторое виртуальное пространство, построенное по принципу В-дерева.

2.3.1 Редактор табличных баз данных

В рамках проекта была разработана программа ведения табличных баз данных (ТБД) технологического назначения (tbluse.exe). Первые версии этой программы применялись еще под управлением ОС DOS.

В данной версии tbluse можно отметить следующие возможности: многозадачность, снятие ограничения по памяти, связь с внешним миром (импорт/экспорт), эргономичность, использование технологии Drag & Drop при создании и реструктуризации таблиц с применением словаря метаданных, расширение типов данных для совместимости с другими системами.

2.3.2. Особенности интерфейса TblUse

Программа tbluse предоставляет более удобный интерфейс пользователю, чем ее DOS-прототип, а так же ряд дополнительных возможностей.

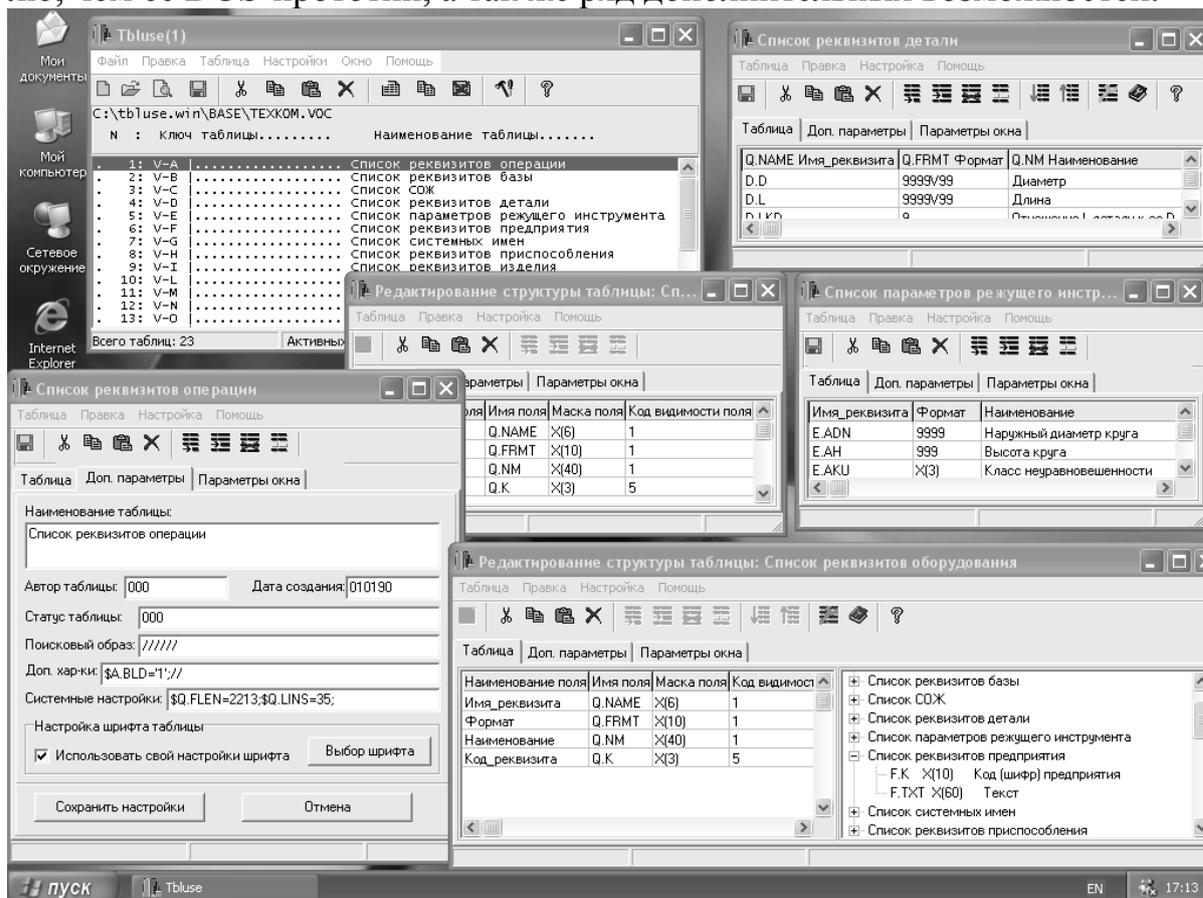


Рис. 2.3. Общий вид редактора таблиц TblUse с демонстрацией своих возможностей

Вместе с тем программа не утратила своей полной совместимости со старыми версиями этого редактора (рис. 2.3).

Среди достоинств данного редактора хочется выделить удобство интерфейса, а так же такую особенность свойственную Windows-совместимыми системами, как многозадачность, то есть одновременная работа с несколькими таблицами одновременно.

Основное окно программы представляет собою список доступных таблиц в текущей открытой одной или двух базах данных, и, при желании, пользователь может открыть любую таблицу в отдельном окне программы простым щелчком мыши. Одновременно возможно просматривать две табличных базы.

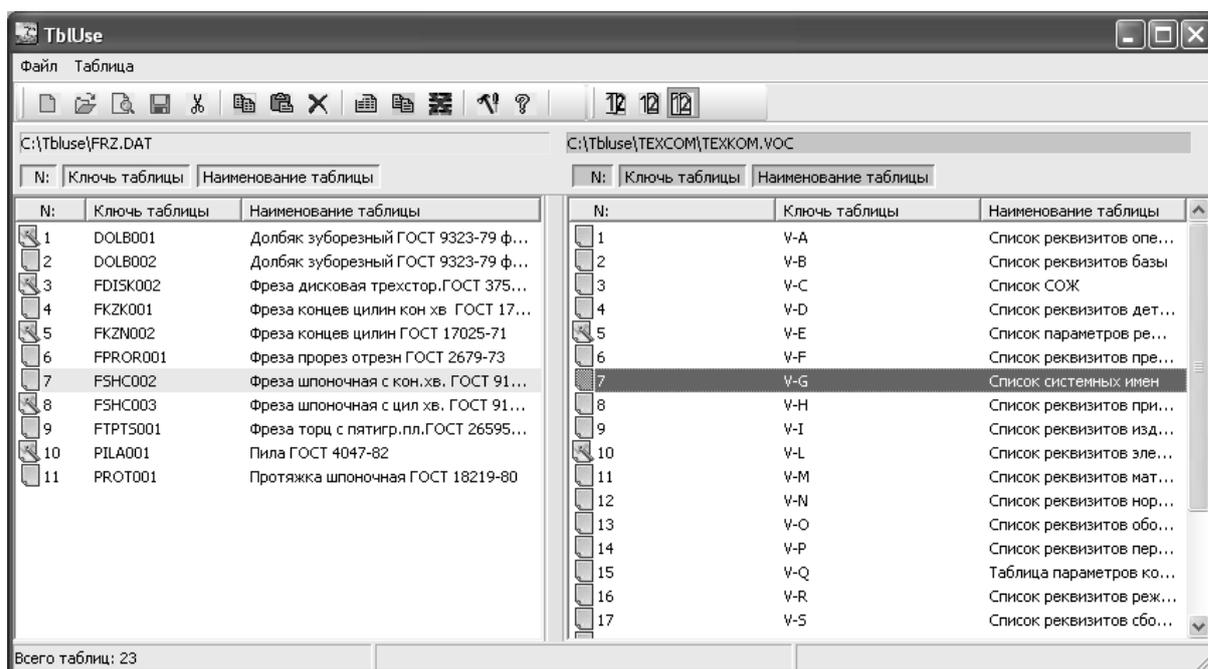


Рис. 2.4. Фрагменты списка доступных таблиц в базах данных

Работы с текущей базой данных осуществляются с помощью меню в виде пиктограмм, или контекстном меню, всплывающим при нажатии правой кнопки мыши. Возможны следующие операции над таблицами в текущей базе данных:

СОЗДАНИЕ НОВОЙ ТАБЛИЦЫ – создает пустую таблицу для дальнейшего ее заполнения;

УДАЛЕНИЕ ТАБЛИЦЫ – физически удаляет таблицу из базы данных;

КОПИРОВАНИЕ ТАБЛИЦЫ В ДРУГУЮ ТАБЛИЦУ – создание новой таблицы на основе данных копируемой таблицы;

ВЫГРУЗКА ТАБЛИЦЫ В ТЕКСТОВЫЙ ФАЙЛ В ВИДЕ ТЕКСТА – выгружает информацию, содержащуюся в таблице в текстовый файл;

ЗАГРУЗКА ТАБЛИЦЫ ИЗ ТЕКСТОВОГО ФАЙЛА СПЕЦИАЛЬНОГО ФОРМАТА – загружает в таблицу информацию из текстового файла;

ВЫГРУЗКА ТАБЛИЦЫ В MS EXCEL - выгружает информацию, содержащуюся в таблице на активную страницу MS Excel;
ЗАГРУЗКА ТАБЛИЦЫ ИЗ MS EXCEL - загружает в таблицу информацию из активной страницы MS Excel;
ОТКРЫТИЕ СОДЕРЖИМОГО ТАБЛИЦЫ В НОВОМ ОКНЕ – открытия окна для редактирования содержимого таблицы.

2.3.3. Редактирование таблицы

Этот режим служит для корректировки данных, поискового образа и дополнительных характеристик, которые содержат таблицы.

Для того, чтобы начать редактирование таблицы, необходимо выбрать таблицу из списка нажатием мыши или клавиши **ENTER**. После этого, если эта таблица не защищена паролем на экране, появится новое окно с содержимым таблицы, в противном случае доступ к таблице, частичный или полный будет определяться знанием Вами пароля. Под частичным доступом понимается доступ в виде

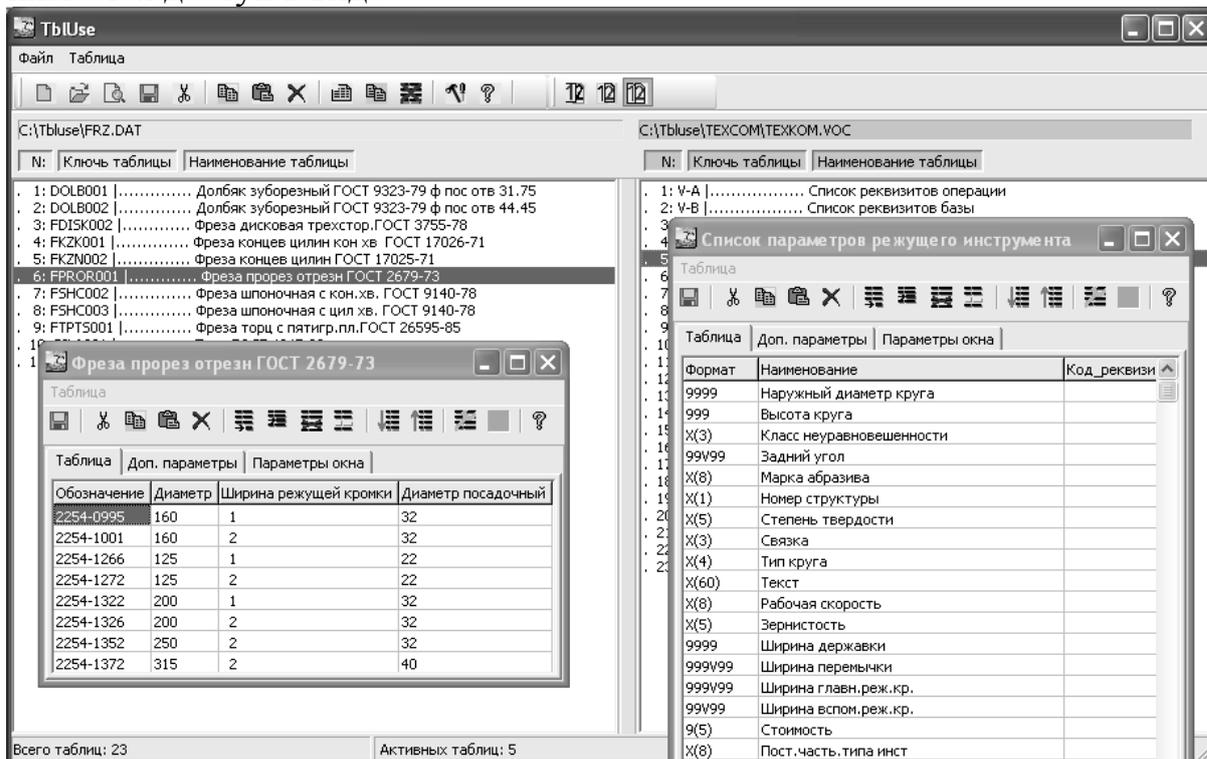


Рис. 2.5. Общий вид таблиц, открытых для редактирования

Такая таблица уже готова к редактированию ее содержимого: для этого достаточно будет просто нажать на интересующую ячейку и внести нужные данные.

Для других изменений, достаточно будет воспользоваться меню в виде пиктограмм, или всплывающим контекстным меню, где доступны следующие функции:

РАБОТА С БУФЕРОМ ОБМЕНА ДАННЫХ – Tbluse эффективно использует системный буфер обмена Windows, где Вы сможете обмениваться данными с другими приложениями Windows.

ДОБАВЛЕНИЕ, УДАЛЕНИЕ, ВСТАВКА, ОЧИСТКА СТРОКИ – откуда можно управлять количественной характеристикой строк в таблице.

СОРТИРОВКА ДАННЫХ ПО ВОЗРАСТАНИЮ, ПО УБЫВАНИЮ – сортирует данные в таблице по выбранному столбцу (рис. 2.6).

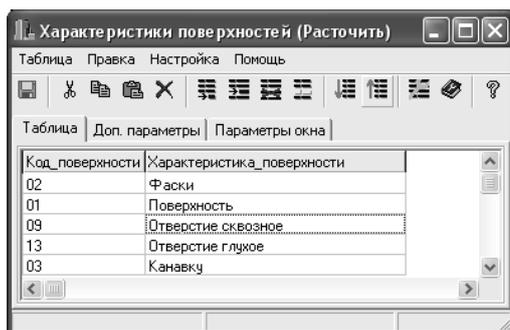
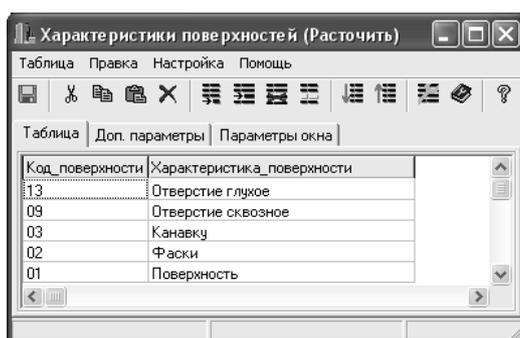
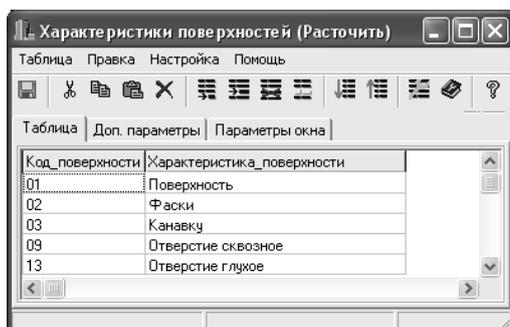


Рис. 2.6. Демонстрация возможностей модуля сортировки данных в таблице

2.3.4. Редактирование структуры таблицы

Пример редактирования структуры таблицы с описанием полей показан на рис. 2.7.

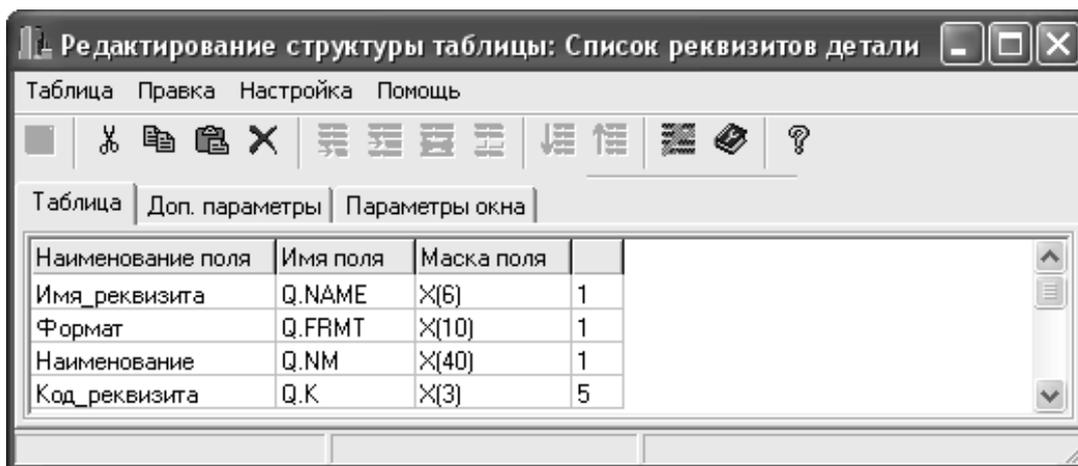


Рис. 2.7. Общий вид редактора, при редактировании таблицы

Структура таблицы может состоять из n-ого количества граф. При описании формата каждой графы (поля) таблицы запрашивается значение четырех реквизитов (рис. 2.8.).

Наименование Поля	Формат Поля	Название графы В таблице	Видимость поля (Код)
			1

Рис. 2.8. Описание полей таблицы

- *наименование поля;*
- *формат поля;*
- *название графы в таблице;*
- *код видимости.*

Наименование поля

В рамках теории ВСПТД имена полей (граф) задаются в соответствии с правилами формирования имен триплетов (см. п. 1.2.1. Факты, цели, триплеты). Здесь необходимо указать имя реквизита длиной до 30-ти символов. В системе принято следующее правило формирования имени реквизита: P.NAME, где

P – префикс, указывающий на принадлежность к группе имен (к объекту);
NAME – имя длиной до 18-ти символов.

Формат поля

В формате поля указывается размер и тип поля (см. 4.3.2).

Название графы в таблице

Здесь указывается наименование реквизита, оно может быть отличным от

наименования указанного в словарной базе. Не следует путать с именем реквизита, которое участвует в формировании имени триплета.

Код видимости

Код видимости каждой графы таблицы может принимать значение от 0 до 7. Информацию о кодах видимости можно получить после активизации F6:

- 0 - видимо в любом режиме в первую очередь (для первой графы);
- 1 - видимо в любом режиме в порядке очереди;
- 2 - видимо в режиме редактирования (код 0), скрыто при просмотре;
- 3 - видимо в режиме редактирования (код 1), скрыто при просмотре;
- 4 - видимо в режиме просмотра (код 0), скрыто при редактировании;
- 5 - видимо в режиме просмотра (код 1), скрыто при редактировании;
- 6 - видимо, но недоступно для редактирования;
- 7 - 0+6 (для первой графы).

Коды 0 и 7 применяются только для 1-ой графы. Код 0 имеет смысл использовать в широких таблицах для ключевых реквизитов, когда возникает необходимость просмотра справа налево и наоборот. При этом графа с кодом 0 всегда будет присутствовать на экране.

Не допускается одновременное использование кодов 0, 6 и 7.

По умолчанию задается код видимости 1.

2.3.5. Реструктуризация таблицы

В режиме реструктуризации таблицы можно изменить формат любой графы, удалить графу, поменять местами разные столбцы. Добавлена возможность: физическая перемена столбцов местами одним перетаскиванием столбца мышью с места на место (рис. 2.9).

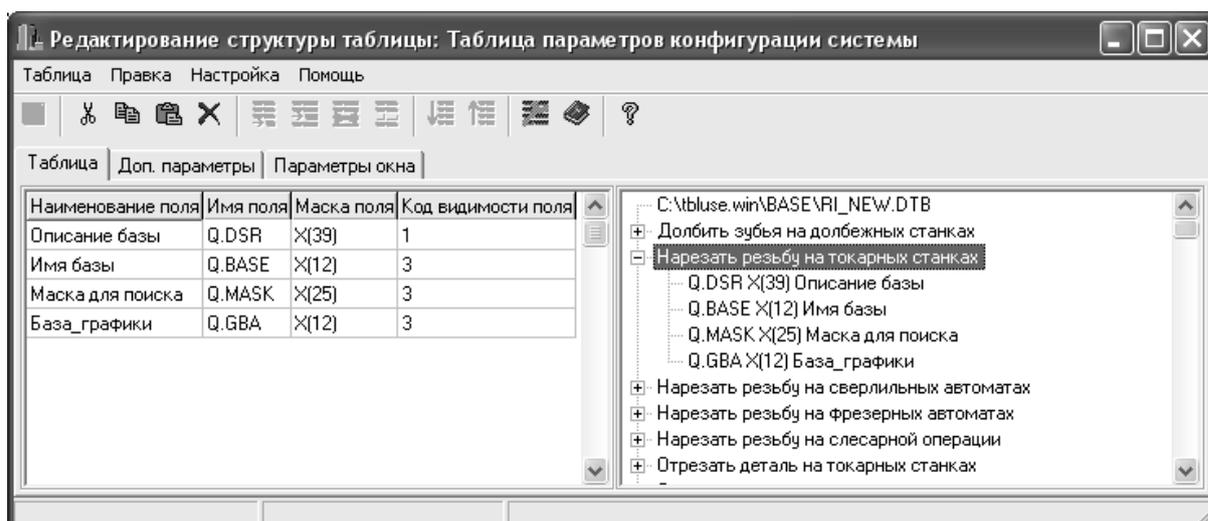


Рис. 2.9. Редактирование таблицы с использованием словарной базы данных

Главным преимуществом работы со словарной базой данных стало использование в качестве словаря не только базу данных, строго предназначенную для этого, но и любую другую таблицу, где система просто создаст поля в редактируемой таблице по аналогии с “донорской” таблицей.

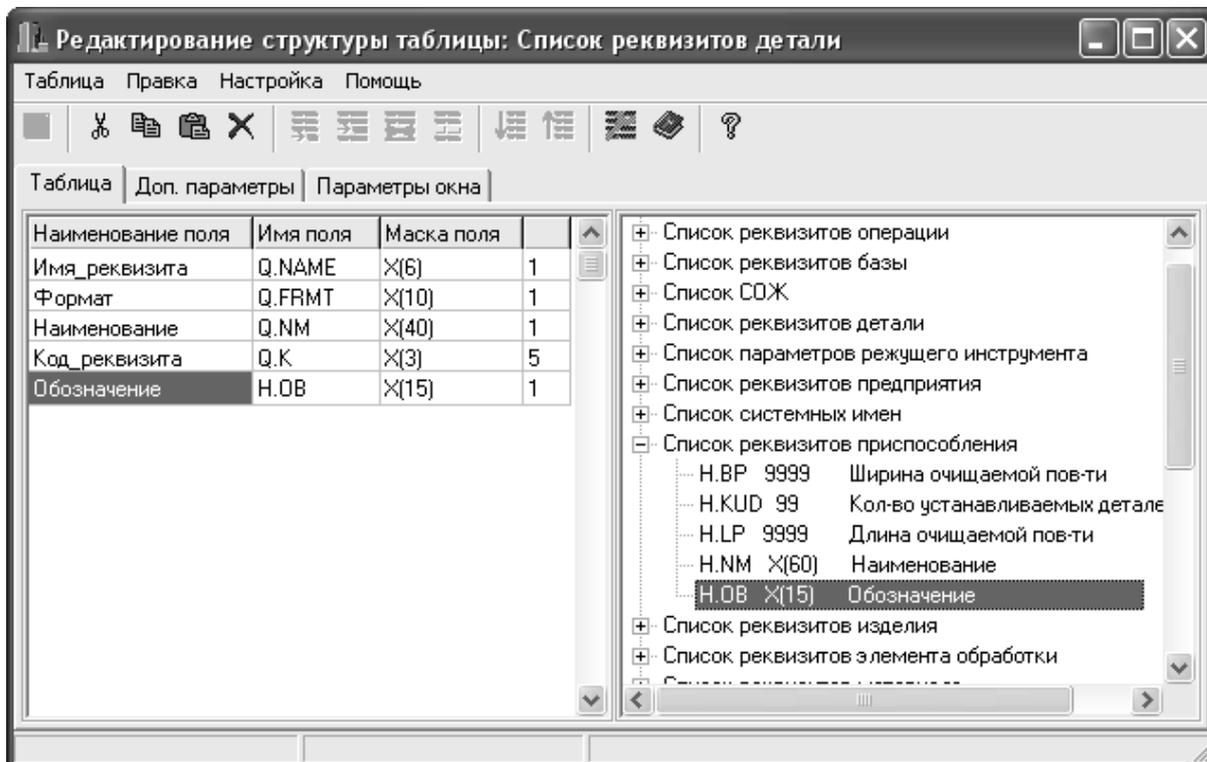


Рис. 2.10. Редактирование таблицы с использованием обычной базы данных в качестве словарной

2.3.6. Поисковый образ

Для того чтобы в режиме проектирования выводилась таблица с конкретными данными, необходимыми в той или иной ситуации, надо установить правило ограничения вывода таблицы на экран (поисковый образ). В описание поискового образа входят:

- имена внешних данных (со знаком \$);
- имена данных используемых в таблице (без знака \$);
- логические функции;
- операции сортировки.

2.3.6.1. Логические функции

- & - “И”
- ! - “ИЛИ”
- > - “Больше”
- >= - “Больше или равно”

- < - “Меньше“
- <= - “Меньше или равно“
- = - “Равно“
- != - “Не равно“

2.3.6.2. Знаки сортировки

- + - по возрастанию
- - по убыванию

2.3.6.3. Правило составления поискового образа

Каждый поисковый образ и операция сортировки должны обязательно заканчиваться символами “ // “. Если операция сортировки отсутствует, то признак конца операции сортировки “ // “ все равно обязательно прибавляется к символам окончания поискового образа.

Рассмотрим правило составления поискового образа на примере выбора штангенциркуля из базы данных.

$$\$L.IR \leq W.L \& \$W.CD \geq W.CD // W.L + W.CD - //$$

где:

$\$L.IR$ – измеряемый размер (внешнее данное). Данный параметр выбирается из проектируемого ТП или вводится через анкету.

$W.L$ – максимальный размер измеряемый штангенциркулем (табличное данное).

$W.CD$ – цена деления штангенциркуля (табличное данное).

$\$W.CD$ – необходимая цена деления при проектировании ТП (внешнее данное). Данный параметр выбирается из ТП или вводится через анкету.

Читается это правило следующим образом:

Измеряемый размер должен быть **меньше либо равен** максимальному размеру, который может измерить штангенциркуль, **и** цена деления необходимая при проектировании ТП должна быть **больше или равна** цене деления штангенциркуля (правило поискового образа закончилось) взять максимальные размеры, измеряемые штангенциркулем **по возрастанию**, цену деления штангенциркуля – по убыванию (правило сортировки закончено).

В режиме проектирования ТП в соответствии с этим правилом на экран будет выведена не вся таблица типоразмеров штангенциркулей, а лишь ее часть. Это позволяет пользователю ускорить процесс выбора из таблиц. Более подробное описание поискового образа можно посмотреть в разделе 3.6.

2.3.7. Дополнительные характеристики

Каждая таблица может быть снабжена дополнительной информацией, которая вводится в режиме редактирования заголовка или в виде анкеты в момент редактирования тела таблицы.

В дополнительных характеристиках вводятся общие данные для таблицы, например: наименование материала, ГОСТ на материал и т.д. Если необходимо, то вводится правило вывода типоразмеров таблицы в выходной документ ТП (\$W.FTD).

Рассмотрим пример правила вывода в выходной документ ТП типоразмеров для штангенциркуля. Оно выглядит следующим образом:

```
$W.FTD='$W.NM $W.OB $W.IST';
```

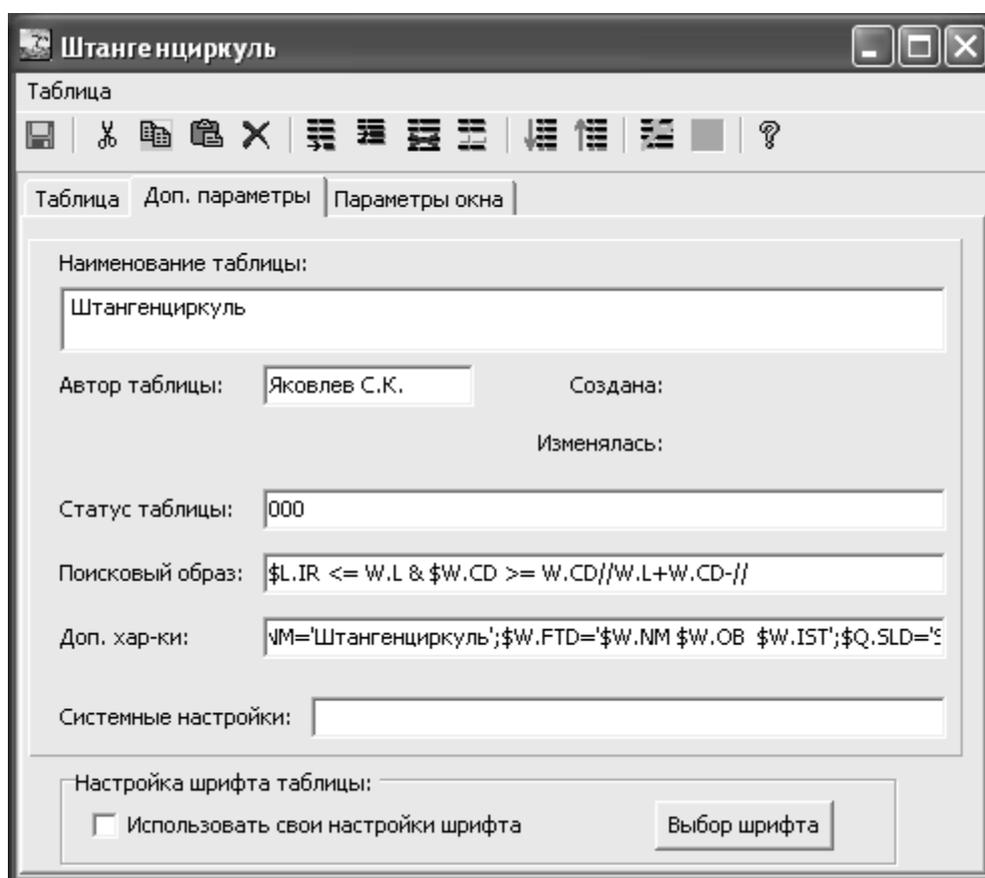


Рис. 2.11. Редактирование дополнительных параметров таблицы

Это означает, что после проектирования ТП и подготовки данных будет сформирована строка:

```
Штангенциркуль ШЦ-I-125-0,1 ГОСТ 166-80  
W.NM W.OB W.IST
```

Именно в такой последовательности и будет записан штангенциркуль в выходном документе ТП.

2.3.8. Режим создания новой таблицы

Для того, чтобы создать таблицу необходимо присвоить ей имя. Имя таблицы является символьным ключом и в пределах одного файла должно быть уникальным. Система всегда следит за тем, чтобы пользователь случайно не записал новую таблицу на место уже существующей.

2.3.9. Вывод таблицы в файл

В этом режиме предоставляется возможность выгрузить таблицу в табличный файл, который будет записан в текущую директорию.

Табличный файл – это копия одной таблицы из табличной базы, в качестве имени берется имя файла табличной базы до расширения, а в качестве расширения - свой порядковый номер в табличной базе.

Табличный файл служит для обеспечения сохранности базы и для передачи данных другим системам.

Структура табличного файла приведена ниже:

```
* Ключ таблицы
stangencirk
* Заголовок таблицы
Штангенциркуль

* Скрытые поля
111
* Поискový образ
$.IR <= W.L & $.CD >= W.CD//W.L+W.CD-//
* Формат таблицы
W.OB=X(15)"Обозначение";W.L=9999"Длина";W.CD=9V9999"Цена
деления";//
*5 Дополнительные характеристики
$.IST='ГОСТ 166-80';$.NM='Штангенциркуль';$.FTD='$.NM
$.OB
$.IR $.IST';$.SLD='SHTANG2.SLD';///
*6 Внутренние характеристики
$.FLEN=680;$.LINS=15;//
* Тело таблицы
ШЦ-I-125-0,1 1250.1
ШЦ-II-160-0,05 1600.05
.....
.....
.....
ШЦ-III-315-0,1 3150.10
ШЦ-III-400-0,1 4000.10
```

Строки, которые начинаются с символа «*», не входят в состав таблицы, но являются разделителями между группами табличной информации. Последовательность и присутствие этой информации в табличном файле строго обязательна.

Строка «* Формат таблицы» содержит описание последовательности граф таблицы и их размеры. После строки «* Тело таблицы» выводятся данные таблицы, согласно размерам своих полей.

2.3.10. Ввод табличных файлов в базу

При вводе табличных файлов в базу необходимо помнить, что структура табличного файла должна полностью соответствовать требованиям, предъявленным выше (см. «**Вывод таблицы в файл**»).

При выходе на режим «**Загрузить из Файла**» сканируется текущий директорий и предоставляется выход в другие директории. При этом будут представлены файлы, у которых имя соответствует базе, а расширение формируется из внутренних номеров таблиц.

Для того чтобы ввести таблицу из файла необходимо подвести курсор к строке содержащей данный файл и активизировать **ENTER**. При нормальном завершении выдается сообщение вида:

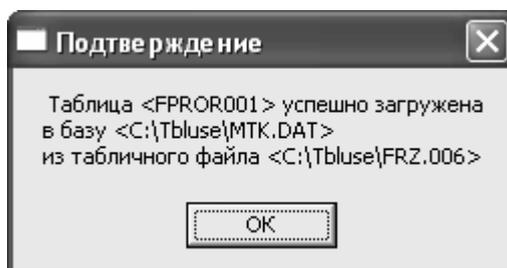


Рис. 2.12. Сообщение о загрузке таблицы из текстового файла.

Если в базе таблица уже присутствует, то запрашивается подтверждение на перезапись. Если табличный файл организован неправильно, выдается сообщение о типе ошибки, а табличный файл в базу не записывается.

2.4. Фреймы-терминалы

В процессе работы большинства систем автоматизации имеются режимы по вводу информации. Как правило, диалоговые окна программируются в процессе создания таких систем. Здесь предлагается представление диалоговых окон в виде фреймов-терминалов, которые в данном контексте будем называть *фрейм-анкетами* или анкетами.

Анкета рассматривается как набор целей в виде имен реквизитов для отображения и ввода значений в ВСПТД, которые должны быть обязательно введены в словарь системы. Каждое имя реквизита должно начинаться со

знака \$. Например, анкете «**Оборудование**» (рис. 2.13) соответствует строка, в которой перечислены со знаком \$ следующие имена:

\$O.C\$O.U\$O.MD|1K62\$O.K\$O.NM\$O.SPU

Все эти имена реквизитов должны быть прописаны в словаре системы, где каждое имя имеет наименование и размер для ввода и триплексная строка для представления информации (см. п.4. «Словарь метаданных»). Например:

O.C - Цех, размер поля для ввода информации 999.

O.U – Участок, размер поля для ввода информации 999.

O.MD – Модель, размер поля для ввода информации X(20), и т.д.

Перед именами реквизитов могут использоваться следующие знаки:

\$\$ - поле не редактируется в пределах анкеты, т.е. в режиме проектирования в этом поле будет выведена какая-то информация, которую нельзя редактировать.

\$: - поле для обязательного заполнения, т.е. в режиме проектирования курсор не выйдет из этого поля пока оно не будет заполнено.

Рис. 2.13. Пример анкеты

Если после имени реквизита указать символ ‘|’ и после него ввести какую-то информацию, то в режиме проектирования после вызова анкеты эта информация будет заполняться в данное поле, как информация по умолчанию. Например, в рассматриваемой строке параметров окон анкеты указано:

\$O.MD|1K62

В режиме проектирования после вызова анкеты «Оборудование» в поле «Модель» будет введено 1K62, при условии, что значение в данное поле ранее не вводилось.

Кроме того, для отображения анкеты подаются параметры представления на экране и наименование анкеты.

\$Q.ZAGA='Оборудование'; \$Q.W=600; \$Q.X=100; \$Q.Y=200; \$Q.H=400;

Здесь: \$Q.ZAGA-заголовок, \$Q.W – ширина, \$Q.X - координата а,

$\$Q.Y$ - координата b , $\$Q.H$ – высота в пикселях. Подробное описание представлено в разделе “Функции формирования анкеты”.

2.5. Фреймы-объяснения

В процессе проектирования система должна обеспечивать объяснения каждого шага проектирования. Объяснения могут иметь статический или динамический характер. Объяснения статического характера формируют подсказку относительно текущего реквизита в кадре информации или о кадре в целом. Процедура объяснений состоит в следующем: ЛПР подводит курсор к интересующему реквизиту и нажимает на клавишу помощи 'HELP'. На дисплее появляется объяснение, относящееся к текущему реквизиту. Второе нажатие на клавишу 'HELP' вызывает объяснение, относящееся ко всему экрану. При этом, если объяснение не исчерпывается одним экраном, может быть вызвана очередная порция объяснений. Таким образом организуется направленный граф статических объяснений. Такого типа подсказки сейчас активно используются в современных программных системах.

Объяснения динамического характера привязываются не к конкретному реквизиту или кадру, а к текущему действию. Объяснения даются на вопросы "как?" и "почему?". Поскольку текущее действие реализуется некоторым правилом вывода, то на вопрос технолога "как?" на экране дисплея может быть напечатан текст соответствующего правила вывода. В некоторых случаях текст правила вывода ввиду его сложности может быть заменен вербальным объяснением. На повторный вопрос "как" встречаются различные в зависимости от контекста пути объяснения действий системы. Например, при назначении типоразмера инструмента печатается текст таблицы, из которой выбран инструмент, а при формировании конструктивного кода инструмента обеспечивается и объяснение цепочки правил, с помощью которых формировался код. Для реализации такой возможности в процессе выполнения правил в памяти ЭВМ создается два списка: список номеров выполненных правил и список номеров невыполненных правил. Правило считается выполненным, если код завершения условия истина, иначе - невыполненным. Таким образом, формируется граф динамических объяснений, на основании которого могут быть аргументированы действия системы. Простейшее представление объяснений в системе реализуется как атрибут правила вывода - всякое правило снабжается ссылкой на семантическую цепочку объяснений, отвечающих на вопросы "как?" и "почему?".

2.6. Фреймы-запросы

При работе с табличной информацией возникла необходимость создания простого и эффективного метода обработки таблиц на основании параметров из ВСПТД. Для этих целей разработан механизм включения в

таблицу фреймов-запросов, представляющих из себя некоторую булеву формулу, в которой в качестве слотов выступают наименования параметров из ВСПТД.

Таким образом, фрейм-запрос работает с выбранной таблицей и областью параметров ВСПТД. Собственно фрейм-запрос есть правильно устроенная логическая формула из констант (чисел и текстовых строк в апострофах), имен реквизитов и имен параметров фрейма-запроса, соединенных знаком арифметических и логических операций и отношений. Для группировки используются круглые скобки. Вся формула заканчивается двумя литерами "косая черта", за которыми следуют список упорядочения и завершающие две литеры "косая черта". Кроме того, после некоторых параметров во фрейме-запросе может следовать двоеточие, за которым идет константа - значение данного параметра по умолчанию. Отсутствие значения после двоеточия означает обязательность данного параметра (обязательная цель) и, в случае отсутствия в ВСПТД, его значение будет запрошено в диалоговом режиме. Пример:

`E.NUM > 3 И $E.TR @ E.TR И E.L < $E.L:10//E.D+//`

Заданный здесь фрейм-запрос отбирает те строки таблицы, в которых значение реквизита E.NUM больше 3, текстовый параметр \$E.TR включается в строку реквизита E.TR и реквизит из таблицы E.L меньше параметра \$E.L, значение которого принимается равным 10, если его нет в ВСПТД. (Параметры из ВСПТД помечаются символом \$ "знак доллара").

2.6.1. Состав операций в описании фрейма-запроса

В описание фрейма-запроса входят следующие операции:

1) Логические операции:

>-больше

>= (=>)-больше или равно

<-меньше

=< (<=)-меньше или равно

=-равно

2) Арифметические операции

+ - сложение

--вычитание

* - умножение

/ - деление

** - возведение в степень

3) Логические операции

AND (И) (&)-конъюнкция

OR (ИЛИ)-дизъюнкция

NOT (¬) (НЕ)-отрицание

4) Одноместные операции::

ABSENT (ABS) (НЕТ) - проверяет отсутствие указанного параметра в области параметров

PRESENT (PRE) (ЕСТЬ) - проверяет присутствие указанного параметра в области параметров

Пример:

ABS(\$L) AND PRE(\$D)

НЕТ(\$L) И ЕСТЬ(\$D)

5) Тригонометрические операции:

SIN - синус

COS - косинус

TG - тангенс

CTG - котангенс

6) операции со строками

1. Операция: **высечения подстроки**.

name[n,t], где name - имя реквизита, n - номер символа, с которого высекают, t - количество высекаемых символов.

Пусть нам задан параметр \$D='BCD'. Требуется выбрать строку, в которой первые три символа реквизита D совпадают с параметром \$D. Эта операция выглядит так:

(D[1,3]=\$D)//

Операция высечения подстроки длиной в t символов из строки, начиная с символа n (нумерация символов в строке начинается с 1) - это выделение t символов подряд начиная с символа n.

2. Операция: **вхождение подстроки неопределенной длины**

Эта операция представляет собой в некотором смысле расширение операции `высечения подстроки` и проверяет вхождение подстроки неопределенной длины в заданную строку. Имеется два варианта `вхождения`:

Вариант 1. Вхождение подстроки, начиная с первой позиции заданной строки. Операция `@`.

Вариант 2. Вхождение подстроки, начиная с любой позиции заданной строки. Операция `@@`.

НАПРИМЕР:

\$STR='ABC' или \$STR='ABCD'; \$STR1='CDE'; STRING='ABCDEFGH'.

Проверка включения подстроки \$STR в строку STRING с первой позиции выглядит так:

\$STR @ STRING,

а проверка вхождения с любой позиции подстроки \$STR1:

\$STR1 @@ STRING

3. Операция: **наложенное значение**

В некоторых ситуациях в случае отсутствия какого-либо входного параметра в фрейме можно задавать конкретное значение соответствующего параметра, которое пишется после формального параметра. Впереди наложенного значения ставится символ `:` (двоеточие).

наложенное значение::= \langle значение \rangle

Если при этом наложенное значение пусто (двоеточие присутствует), а указанный параметр отсутствует в области параметров, то обработка фрейм-запроса прекращается с сообщением:

`отсутствует обязательный параметр` \$имя параметра`

Такой режим работы позволяет подключать отсутствующие обязательные параметры в диалоговом режиме.

4. Операция: *сортировка*

Если мы задаем фрейм-запрос по вышеописанным правилам, то будет выбрана первая строка, удовлетворяющая фрейм-запрос. Но в таблице может быть несколько решений. Для выбора более точного (ближайшего) значения после конца фрейм-запроса задаются параметры сортировки обычно по тем реквизитам, которые выбираются по отношениям:

\langle , \rangle , \leq , \geq

Параметры сортировки::= \langle ИМЯ ПАРАМЕТРА \rangle \langle ЗНАК СОРТИРОВКИ \rangle ;

; \langle ИМЯ ПАРАМЕТРА \rangle \langle ЗНАК СОРТИРОВКИ \rangle КОНЕЦ СОРТИРОВКИ

Знаки сортировки::= + - по возрастанию

- - по убыванию

обычно знаку \langle соответствует знак '-'

знаку \rangle соответствует знак '+'

конец параметров сортировки: '//'

Например, фрейм-запрос с сортировкой может выглядеть так:

((A>\$A) AND (B<\$B))//A+;B-//

Комментарий

После каждого параметра в фрейме-запросе в кавычках можно указать информацию, не участвующую в поиске, но которая может быть использована для объяснения реализуемого алгоритма.

Например: D<=\$S.DR "РАЗМЕР ДЕРЖАВКИ"

2.7. Фреймы-продукции

Фреймы-продукции, по сути, представляют естественное развитие фреймов-запросов.

Знания представляются в виде правил вывода (продукций). Интерпретация правил осуществляется с помощью дедуктивной машины вывода (ДМВ). Место ДМВ в составе экспертной системы показано на рисунке 2.14.

Машина вывода (интерпретатор правил) – программа, имитирующая логический вывод эксперта, пользующегося данной продукционной базой знаний для интерпретации поступивших в систему данных, и выполняет две функции:

- (1) Просмотр существующих фактов (данных) из рабочей памяти (базы данных) и правил из базы знаний и добавление (по мере возможности) в рабочую память новых фактов.

- (2) Определение порядка просмотра и применения правил. Этот механизм управляет процессом консультации, сохраняя для пользователя информации о полученных заключениях, и запрашивает у него информацию, если для срабатывания очередного правила в рабочей памяти оказывается недостаточно данных.



Рис. 2.14. ДМВ в составе ЭС.

Продукции представляют собой простейшие правила следующего вида:

ЕСЛИ УСЛОВИЕ ТО ДЕЙСТВИЕ КОНЕЦ

Применен формальный язык правил вывода, обеспечивающий основные потребности инженера технологических знаний. Условия продукции включают в себя обычные функции сравнения текстовых и числовых величин, арифметические выражения и тригонометрические функции, а также две функции "ЕСТЬ" и "НЕТ", проверяющие соответственно наличие (отсутствие) указанного параметра в ВСПТД. Правила вывода разбиваются на группы по выполняемым функциям. Каждой группе присваивается постоянный номер, который используется управляющей структурой при определении последовательности их исполнения. Например, при формировании шифра инструмента в 1-ю группу включаются правила, определяющие вид инструмента (резец, сверло, зенкер и т.п.). После определения вида инструмента вызывается группа правил, которая выводит оставшиеся составляющие шифра соответствующего вида инструмента.

Условие

Если ДМВ находит несколько правил с истинным условием, то выбор результата определяется отдельно задаваемым критерием, который может изменяться в процессе работы.

При проверке условия может встретиться триплет с обязательным значением, которого в ВСПТД не оказалось. ДМВ сообщает имя отсутствующе-

го триплета, номер правила и код остановки. Управляющая структура принимает решение о способе получения значения, помещает его в ВСПТД и в редуцированную область и снова передает управление ДМВ. В зависимости от результата (получено или не получено значение) передается код продолжения работы ДМВ:

- продолжить с текущего правила;
- продолжить со следующего правила;
- прекратить работу;
- начать работу сначала.

Действие

Если результат отработки условия - истина, то ДМВ выполняет действие. В результате исполнения действия могут быть следующие варианты:

- а) формирование фрейма;
- б) обращение к решающей таблице;
- в) формирование целевой функции, не исполняемой ДМВ;

Результат помещается в список (вектор) возможных ответов. Каждый триплет снабжается коэффициентом достоверности, количеством используемых в условии параметров и номером породившего его правила. Левая часть действия представлена переменной, имя и формат которой имеются в списке локальных переменных данной группы.

Операция присваивания

$I:=V$, где I - имя параметра, V - выражение. Выражение может содержать в себе арифметические действия (см. п. 2.6.1)

ДМВ вычисляет выражение, результат преобразует в символьное представление и формирует триплет, который помещается в очередной элемент вектора ответов.

Пример продукции

ЕСЛИ вид обработки **СВЕРЛИТЬ ИЛИ** рассверлить **И НЕТ** технических требований **ТО** выбрать сверло общего назначения.

Приведенная продукция будет записана в ВСПТД в виде строки:

ЕСЛИ (($\$L.WOB = 25$ **ИЛИ** $\$L.WOB = 27$) **И** **НЕТ**($\$A.TT$)) **ТО** $\$E.KTS$
[1,2] := '21'

При этом возможные значения параметра $\$L.WOB$:

- (a) 025 – рассверлить;
- (b) 027 – сверлить.

Возможные значения параметра $\$E.KTS$:

(a) 21 – сверло общего назначения.

Префиксы и имена триплетов в словаре метаданных могут представлять тот или иной объект:

- (a) L – элемент обработки;
- (b) WOB – вид обработки;
- (c) E – инструмент;
- (d) KTS – шифр инструмента.

2.8. Фреймы-документы

При разработке САПР ТП выделяются следующие опорные точки интеграции: обработка текстов, табличная обработка, традиционные функции БД, среди которых подготовка печатных сводок иногда образует отдельный модуль, деловая графика и др. Именно эти функции несут в себе отдельные модули в целой группе программных продуктов, сохраняющих внутреннюю связь и преемственность.

Проблема подготовки печатных сводок или генератора отчетов (ГО) появилась вместе с первыми печатающими устройствами, и в этой области имеется много различных решений, среди которых выделяются три направления:

- разработка ГО с помощью языков высокого уровня: C++, XML, Паскаль и др.;

- сопряжение с базами данных;

- табличные генераторы матриц и отчетов;

Поскольку информация от всех предметных модулей укладывается в форме триплетов в ВСПТД, то возникла необходимость разработки генератора отчетов, работающего над ВСПТД. Здесь нет жестко структурированных записей с одной стороны, а с другой стороны - каждый реквизит в строке ВСПТД снабжается именем, и строки ВСПТД, в свою очередь, имеют четко выраженный контекст (имеется в виду представление ключей строк ВСПТД в соответствии со структурой технологического процесса). Этот факт позволил создать довольно простую и компактную систему генерации документов. Макет документа (фрейм) формируется с помощью редактора текстов, а каждое поле значений снабжается именем соответствующего реквизита (цель) из строки ВСПТД. В простейшем случае задача сводится к формированию макета и указанию в программе печати последовательности адресов (ключей) строк, из которых выбирается информация. Опыт внедрения автоматизированной системы проектирования технологических процессов показал, что на различных предприятиях предъявляются неодинаковые требования к форме одних и тех же документов. Переработка программ печати обычно связана с определенными ресурсами столь дефицитного времени программиста. Возможность рассматривать документ как некоторый фрейм, позволила проводить его

корректировку силами технолога, владеющего минимальными навыками работы на ЭВМ.

Схема формирования фрейма такова: "ввод бланка документа в ЭВМ с помощью текстового редактора - разметка окон документа в соответствии с обозначениями соответствующих реквизитов и контекстом".

Например: распечатка параметров обрабатываемой поверхности в текущем переходе (контекст - L3).

Параметры обрабатываемой поверхности:

Длина Диаметр Наименование

Фрагмент фрейма-документа с явно заданными слотами может выглядеть, например, так:

Параметры обрабатываемой поверхности:

Длина Диаметр Наименование

L3 \$L.L \$L.D \$L.NM

Для формирования такого простого документа печати достаточно создать соответствующий файл, где мнемоническое обозначение реквизитов \$L.L, \$L.D и \$L.NM соответствует наименованию окон документа. В тех случаях, когда обозначения реквизитов не помещаются в соответствующую графу документа, они перечисляются на отдельной строке, а в соответствующих графах остается символ \$.

Пример фрагмента фрейма-документа с неявно заданными слотами

Параметры обрабатываемой поверхности:

Длина Диаметр Наименование

\$L.L \$L.D \$L.NM

Сложный документ может быть составлен из нескольких фреймов. Задача программиста заключается в определении последовательности их применения.

На описание слотов в документе распространяются те же правила, которые предложены для триплетов-целей. В частности, можно указывать значение по умолчанию или обязательность указанного параметра. Например,

а) значение по умолчанию: \$T.FMGT:'Иванов' - если не задана фамилия главного технолога, то напечатать 'Иванов'.

б) обязательный параметр: \$E.V0: - если не задан обязательный параметр скорость резания, то опросить в диалоговом режиме.

Наглядный пример применения генератора отчетов можно рассмотреть на примере формирования наименования инструмента. Пусть область ВСПТД содержит информацию о резце:

...\$E.BD=20; \$E.HD=25; \$E.IST='ГОСТ 18877-73'; \$E.KRZ='отогнутый';
\$E.NM='резец'; \$E.MMR='BK8'; \$E.TIP='проходной';...

Для распечатки наименования инструмента в заданной форме в макете документа могут быть предложены следующие варианты:

1. \$E.NM: резец;

2. \$E.NM: \$E.TIP \$E.KRZ \$E.IST
резец проходной отогнутый ГОСТ 18877-73;
3. \$E.NM: \$E.TIP \$E.MMR \$E.IST
резец проходной BK8 ГОСТ 18877-73;
4. \$E.NM: \$E.BD "x" \$E.HD\$E.IST
резец 20x25 ГОСТ 18877-73;
5. \$E.NM: \$E.TIP \$E.MMR\$E.BDx\$E.HD\$E.IST
резец проходной BK8 20x25 ГОСТ 18877-73.

В этом примере предложено пять вариантов формирования наименования резца. Эксперт-технолог может сам принять решение о наиболее подходящем для его предприятия варианте, указав последовательность параметров в макете документа.

Другой метод формирования документов заключается в подключении управляющей структуры для формирования значения реквизитов по заданному условию. Дальнейшее развитие формирования отчетов связано с организацией связи между ВСПТД и приложением MS Word (см. Применение виртуального строкового пространства в САПР ТП. Методическое пособие). Решена задача формирования документов из данных, представленных в ВСПТД, на основе формата OpenDocument³[20].

3. Возможные области применения ЭС при разработке САПР ТП механической обработки

3.1. Разумность разработки ЭС

До настоящего времени свойства, характеризующие целесообразность разработки ЭС для той или иной проблемы, в общем виде отсутствуют.

В литературе по ЭС имеется ряд методических рекомендаций по вопросу их применения для различных приложений. Ключевые факторы разумности разработки ЭС - это характер, сложность, широта постановки задачи.

Краткое содержание этих рекомендаций.

Характер. ЭС разумно применять к задачам, которые сводятся к символическим рассуждениям и являются эвристическими по своей природе.

Сложность. Задача не должна быть слишком легкой, т.е. ее трудно реализовать в рамках традиционного программирования.

Широта постановки задачи. Задача должна быть достаточно узкой, чтобы с ней можно было справиться, и достаточно широкой, чтобы представлять практический интерес.

³ Российская Федерация входит в ряд стран, в которых формат OpenDocument принят в качестве государственного стандарта. Стандарт ГОСТ Р ИСО/МЭК 26300-2010 вступил в действие 1 июля 2011 года.

Кроме того, определены классы задач, которые целесообразно выполнять в области машиностроения. К ним относятся: интерпретация данных, прогнозирование, обучение, диагностика, проектирование, планирование.

Отметим две последние, как наиболее отвечающие характеру задач САПР ТП механической обработки.

Проектирование - разработка документов для технологических процессов, удовлетворяющих стандартам (ГОСТ, ОСТ, СП и т.п.) и другим требованиям производства.

Планирование - составление плана действий (например, технологический маршрут), при этом должны учитываться состояние ресурсов, возможности оборудования, противоречивость данных и т.п.

3.2. Задачи, удовлетворяющие требованиям разумности разработки ЭС

Применение, в условиях проектирования технологических процессов, методов ЭС определяется тем, что перед непосредственным написанием технологического процесса (маршрута, операционных карт и т.п.) технолог должен проанализировать (подвергнуть экспертизе):

1. Технические требования (технические условия) на деталь, как правило, написанные на чертеже детали и учитывающие:

- взаимное расположение поверхностей;
- профили поверхностей;
- требования к термической обработке (химико-термической обработке) и т.п.

2. Геометрию детали, что определяет возможные способы получения поверхностей детали и, самое важное, последовательность их обработки с учетом количества обращений к соответствующим поверхностям для их обработки. Кроме того, сопоставление технических требований к детали и ее геометрии позволяет провести анализ размерных цепей с точки зрения их технологической реализации и возможности их пересчета из технологических соображений.

Задача определения технологических баз является одной из основных и наиболее сложных проблем в процессе проектирования. При решении задачи выбора технологических баз, необходимо учитывать большое количество факторов, влияющих на точность взаимного расположения поверхностей детали и детали относительно средств обработки. К этим факторам относятся размерные связи, технические требования, вытекающие из служебного назначения деталей, геометрическая форма отдельных поверхностей, форма и качество заготовки, возможностей средств изготовления и т.п. Как отмечается в работе, трудность определения технологических баз вызывается в значительной мере не только сложностью задачи, но и слабо развитыми методическими положениями. Рекомендации по решению этой задачи могут быть применены при использовании методов ЭС.

Анализ технологических потребностей для изготовления детали приводит к экспертной оценке возможностей станочного парка (оборудования) на методы обработки, реализуемые на том или ином оборудовании, что, в сущности, является одним из основных принципов формирования поверхностей при ранее определенной последовательности их обработки.

Кроме того, анализ возможностей обработки поверхностей (детали) приводит к решению формирования промежуточных элементов детали, впоследствии удаляемых (ложные центры, промежуточные дополнительные площадки для наклонного сверления и т.д.).

Следующей задачей, решение которой требует применения новых методов, является выбор заготовки. Как отмечалось в работе, стоимость расходуемого материала из-за неоптимального выбора заготовки зачастую значительно превышает стоимость обработки детали. Указывая на преимущества принятия решения при выборе заготовки с помощью ЭВМ, отмечается, что в настоящее время не имеется универсальных решений этой задачи. Трудность выбора заготовки с помощью ЭВМ обусловлена многообразием различных факторов.

Анализ технических требований и геометрии детали, количество и расположение обрабатываемых поверхностей приводит к обоснованному решению по методу (и месту) получения заготовки. Так, например, торсионные валы будут работать лучше, если они будут изготовлены из поковки (штамповки) с продольным расположением волокон материала заготовки.

Комплексный анализ технических требований, геометрии детали и возможностей оборудования дает возможность обоснования выбора инструмента или комплекта инструментов. Так, например, если при обработке отверстия требуется получить высокую точность при оценке цилиндрического отверстия, то перед окончательной обработкой отверстия разверткой (инструмент) может потребоваться расточить отверстие резцом, т.к. резец дает хорошее направление оси отверстия, развертка дает хорошую цилиндричность и чистоту поверхности.

Анализ вариантов использования различных режущих инструментов для равноценной обработки поверхности ведет к оптимизации количества режущих инструментов данного типоразмера для их хранения и оптимизации инструментальной наладки.

Анализ метрологических характеристик измерительного инструмента также позволяет оптимизировать и выбор измерительной оснастки (универсальной и специальной), и технологическую подготовку производства с точки зрения выдачи заказов на изготовление и т.д.

Все вышесказанное о режущих и измерительных инструментах в равной степени относится к вспомогательным инструментам и приспособлениям.

Выбор смазывающе-охлаждающей жидкости (СОЖ) оказывает существенное влияние на характеристики обработки деталей с точки зрения режимов резания и качества поверхности после обработки. Выбор СОЖ требует тщательного анализа обработки поверхности (грубая, чистовая и т.п.), требований к

антикоррозийности СОЖ и ряда других факторов. Например, сульфозфрезол (СОЖ) используется как при развертывании, так и, чаще всего, при зубонарезании.

Кроме того, на выбор СОЖ влияет и обрабатываемый материал. Например, при токарной обработке (грубая, черновая обработка) латуни не рекомендуется использовать эмульсию, а при обработке бронзы эмульсия рекомендуется к применению.

Таким образом, автоматизацию проектирования следующих технологических задач целесообразно реализовать с использованием методов ЭС:

- выбор заготовок;
- выбор технологических баз и способов крепления заготовок;
- выбор оборудования;
- выбор приспособлений и других видов оснастки;
- формирование конструктивно-технологического кода или шифра и поиск инструмента;
- расчет припусков и назначение промежуточной точности и чистоты обработки;
- назначение смазочно-охлаждающей жидкости (СОЖ).

Многообразие вариантов при решении вышеперечисленных задач является одной из главных характеристик, подтверждающих целесообразность их реализации с помощью ЭС.

3.3. Некоторые особенности применения ЭС в САПР ТП механической обработки

Первые попытки применения ЭС в САПР показали, что разработка и внедрение систем такого типа оказалось очень трудоемким и неэффективным занятием. И, после первой эйфории, у многих разработчиков, и прежде всего руководителей, появилось скептическое отношение к таким экспериментам. О том, что от применения ЭС не следует ожидать существенного прогресса в кратчайшие сроки сразу во всех применениях САПР, отмечается в работе. С другой стороны, при построении ЭС проектирования процесс накопления знаний упрощается, так как большинство законов проектирования уже представлены в формализованном виде, то есть высок удельный вес знаний первого рода. Эти два вроде бы противоречащих друг другу утверждения на самом деле как раз и указывают на целесообразность применения ЭС САПР ТП механической обработки и выявляют требования, предъявляемые к ЭС в САПР ТП:

-подключение ЭС к промышленным САПР ТП не должно влиять отрицательно на их рабочие характеристики;

-следует активно использовать технологические знания, которые были реализованы традиционными методами программирования и вполне успешно используются на предприятиях.

Эти положения будут подробно изложены в следующих главах.

По результатам экспериментальных исследований привлечения пользователей к процессу создания САПР повышает уверенность в эффективности системы.

Анализируя историю развития программирования можно заметить, что проблема создания информационно-программного продукта с новой идеологией, в общем-то, повторяется. Аналогичная ситуация возникла, в свое время, с системами управления базами данных (СУБД). На заре развития СУБД в начале 70-х годов многим казалось, что их внедрение позволит быстро решить все проблемы, связанные с программированием инструментальных средств и ведением информационных баз. Дело, однако, кончилось тем, что многие разработчики на первом этапе попросту отказались от СУБД, переходя на файловые системы. Затем стали определять разумное место для СУБД с учетом особенностей задачи, возможностей СУБД и используемой ЭВМ. И если с этих позиций подходить к применению ЭС в САПР ТП механической обработки, то остается определить роль и место ЭС в разрабатываемой системе.



Рис. 3.1. а) ЭС не включена в систему б) ЭС включена в систему

Технологические данные можно классифицировать по двум типам: статические и динамические. Статические данные, как правило, описывают конкретные объекты. Динамические данные представляются в процедурном виде. А поскольку они, в свою очередь, могут содержать статические данные, и, будучи параметризованными, во время работы использовать другие статические данные, то здесь прослеживается связь между статическими и динамическими данными, или, в терминологии ЭС, между данными и знаниями. Отсюда вытекает требование, исходя из которого, каждый предметный модуль системы должен иметь равные возможности получения, как данных из БД, так и знаний из БЗ. Принцип равных возможностей при работе с БД и с БЗ из прикладных программ обеспечивает эволюционный метод развития САПР ТП с применением ЭС (рис. 3.1.).

Задача проектирования технологических процессов сводится к выполнению ряда независимых модулей, каждый из которых при этом может использовать результаты других. Их взаимодействие осуществляется с помощью универсального интерфейсного модуля (ИМ) через виртуальное строковое про-

странство технологических данных (ВСПТД), в котором накапливаются и обновляются сведения по данному технологическому процессу.

Знания предложено представлять в виде правил вывода (продукций) и фреймов. Интерпретация правил осуществляется с помощью дедуктивной машины вывода (ДМВ). Исходя из такого представления предложена глобальная структура САПР ТП и на основе проведенного анализа сформулированы требования, предъявляемые к организации информационного обеспечения и инструментальных средств программирования при разработке гибридной ЭС (ГЭС) САПР ТП, основными из которых являются следующие:

- как прикладные программы, так и ДМВ работают с одним и тем же пространством данных, связь с которым обеспечивает специальный интерфейсный модуль (ИМ);

- любая прикладная программа имеет возможность в случае необходимости через СУБЗ и ДМВ обращаться к БЗ;

- любая прикладная программа имеет возможность через СУБД обращаться к БД.

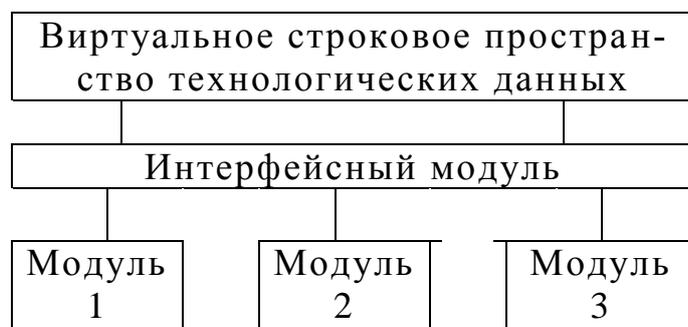


Рис. 3.2 Глобальная архитектура ГЭС САПР ТП механической обработки.

Каждый из модулей имеет один из вариантов структуры, представленной на рис. 3.1.

Гибридные ЭС представляют программный комплекс, агрегирующий стандартные пакеты прикладных программ (например, математическую статистику, линейное программирование или системы управления базами данных) и средства манипулирования знаниями. Это может быть интеллектуальная надстройка над ППП (пакетами прикладных программ) или интегрированная среда для решения сложной задачи с элементами экспертных знаний⁴.

Результаты настоящей работы показывают на возможность создания ГЭС САПР ТП механической обработки с использованием уже существующих разработок. На основании проведенных исследований

⁴ Несмотря на внешнюю привлекательность гибридного подхода следует отметить, что разработка таких систем является задачей на порядок более сложной, чем разработка автономной ЭС. Стыковка не просто разных пакетов а различных методологий (что происходит в гибридных системах) порождает целый комплекс теоретических и практических трудностей.

разработан ряд модулей при создании ГЭС. Построена глобальная структура, обеспечивающая последовательное независимое исполнение различных модулей системы с использованием ИМ, работающего над ВСПТД. Вся оперативная информация записывается с помощью ИМ в виртуальные строки в триплексной форме. Эта информация может быть получена как в диалоговом режиме, так и при работе с БЗ и БД. Задача системы заключается в переводе каждого триплета цели во множество фактов. В процессе проектирования триплеты и синтагмы группируются в строки в контексте "операция-переход" и помещаются в ВСПТД. Таким образом, вся информация при проектировании технологического процесса укладывается в триплексные строки.

При проектировании операций с применением ЭС предложен подход, который включает в себя следующие этапы:

- формирование базы знаний в виде фреймов-переходов;
- формирование базы знаний в виде правил вывода,

обеспечивающих автоматический выбор переходов при работе в режиме индивидуального проектирования.

Будем считать, что база знаний в виде фреймов-переходов уже готова. В этих условиях разрабатываются правила вывода, с помощью которых на основе данных о чертеже детали, параметров оборудования и заготовки, технических требований и, возможно, других данных, решаются задачи выбора переходов на операцию и последовательности обработки поверхностей и выбора переходов на операцию.

Проектирование переходов осуществляется по одному из следующих способов:

-технолог полностью описывает текущий переход, включая описание всех параметров перехода;

-исходя из параметров оборудования и вида обработки, система автоматически редуцирует область базы знаний, и предлагает технологу выбрать окончательный вариант фрейма-перехода для включения его в операцию;

-на основе правил вывода система автоматически выбирает группу фреймов-переходов на операцию.

При этом спроектированная операция может быть включена в базу готовых решений.

Средства управления табличными данными и знаниями, а также объяснения действий системы в диалоговом режиме статического и динамического характера, позволяют оперативно работать с базой данных (знаний) и, при необходимости, настраивать систему на желаемый результат с учетом знания промежуточных результатов. В режиме корректировки табличной базы данных подключается встроенный редактор базы знаний, обладающий синтаксическим анализатором с высокоуровневой системой диагностики, позволяющий менять тексты фрейма-запроса и структуры

таблицы с автоматическим анализом и контролем синтаксиса и некоторых семантических отношений фрейма-запроса и структуры.

База данных может корректироваться под условия конкретного предприятия, производственного подразделения силами самих пользователей. Однако, если корректировку табличной базы данных пользователь может осуществлять практически без предварительной подготовки, то включение новых способов обработки таблиц с учетом текущего состояния пространства технологических данных требует некоторых навыков.

3.4. Параметризованная информация о 3D-моделях

В работе [19] описаны инструментальные средства, позволяющие получать параметризованную информацию о 3D-моделях, созданных в различных САД системах и представлять ее в виде триплексных строк, например, для детали, представленной на рис. 3.3:

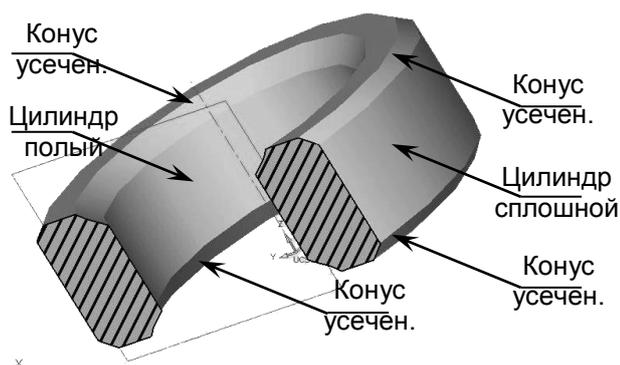


Рис. 3.3. Пример детали для представления в триплетной форме

может быть создано следующее параметрическое представление:

Вал (1 ед.):

```
$CIM_CYL.RAD=30;  
$CIM_CYL.LENGTH=20;  
$CIM_CYL.TYPE=1;  
$CIM_CYL.LINK=0;
```

Отверстие (1 ед.):

```
$CIM_CYL1.RAD=20;  
$CIM_CYL1.LENGTH=20;  
$CIM_CYL1.TYPE=0;  
$CIM_CYL1.LINK=0;
```

Фаска (4 ед.):

```
$CIM_CHM.LEN=3;
```

```
$CIM_CHM.LINK=1;  
$CIM_CHM.ANGLE=45;
```

Полученные триплексные строки будут являться входными параметрами созданной экспертной системы наряду с данными, полученными от пользователя. Результатом работы системы также могут быть параметризованные данные о 3D-модели (модифицированной или созданной), которые будут переданы обратно в САД-систему с помощью генератора моделей, описанного в той же диссертации.

4. Методы работы с виртуальным строковым пространством технологических данных

Виртуальное строковое пространство (ВСП) создано для облегчения разработчику работы с большими массивами данных с записями как фиксированной, так и переменной длины. Операции взаимодействия с ВСП представляют из себя набор процедур, с помощью которых организуется соответствие между вводимыми пользователем данными и уникальным ключом, который пользователь присваивает данным в момент занесения их в ВСП. Выполнив процедуру связывания данных с ключом, пользователь может в любой момент времени обратиться к этим данным, изменить их и занести обратно в ВСП. ВСП организовано так, что в памяти хранится только некоторая управляющая информация о ВСП и та его часть, которая наиболее интенсивно используется при работе. Это достигается тем, что ВСП применяет страничный буферный пул данных (СБПД), который основан на алгоритме Double Ring Link (Двухсвязное списочное кольцо), позволяющий следить за частотой использования данных и сортировке их по этому признаку (частоте). При обращении к данным по ключу происходит обращение к СБПД и, если искомым данным нет в наиболее часто используемой области, то самые редко используемые из них заменяются на искомые. Те данные, которых нет в области СБПД, записываются в файл и при необходимости считываются из него в область СБПД.

В качестве одного из вариантов развития системы выбрана система управления записями ВTRIEVE (см. 6.1. Система управления записями ВTRIEVE), обеспечивающая работу в сети и применения языка запросов SQL.

4.1. Интерфейсный модуль

Для решения задачи взаимодействия различных модулей системы предложена методика технологии программирования, учитывающая потребности предметных модулей.

- Для связи процедур обработки вводятся следующие соглашения:
- каждая строка ВСПТД идентифицируется уникальным ключом;
 - каждый триплет строки ВСПТД может принадлежать двум и только двум процедурам обработки (порождающей триплет и потребляющей его);
 - каждому триплету из виртуальной строки ставится в соответствие параметр состояния этого триплета, принимающий одно из predetermined состояний этого триплета: значение триплета пусто, триплет имеет значение, триплет отсутствует в строке; параметры состояния выдают процедуры обработки, работающие с соответствующим триплетом, управляющему модулю верхнего уровня;
 - процедура обработки может быть запущена, если она обеспечена входными данными.
 - процедура обработки может завершить работу, изменив состояние хотя бы одного из своих входных или выходных триплетов.

Интерфейсный модуль представляет собой основное связующее звено при обработке текстовой информации между предметными программами и ЭС. Символьная обработка информации в строке становится критичной по времени при достаточно длинных строках. Для эффективной работы ИМ предложен **двоично-последовательный метод поиска**. Триплеты укладываются в строку в лексикографическом порядке имен. Алгоритм поиска триплета по имени следующий: проверяется сначала первый триплет в строке. Если искомый триплет "меньше" (в смысле лексикографического упорядочивания), то поиск прекращается, иначе, если "больше", то последовательно ищется последний триплет в строке (поиск триплета с конца). Если триплет "меньше", то строка делится пополам и процедура повторяется.

Представление технологической информации в САПР ТП механической обработки заготовок.

Система может получать данные (и их сочетания) с помощью следующих методов:

- анкетный ввод данных;
- поиск данных;
- поиск данных с пересчетом;
- расчет данных;
- выбор данных из единичной таблицы.

Данные о технологическом процессе размещены в файле с именем VSP<регистрационный номер>.trp, где "регистрационный номер" - это номер технологического процесса в общем каталоге системы. Например:

vsp0001.trp - информация о техпроцессе № 1 по каталогу.

Файл с информацией по технологическому процессу имеет древовидную структуру. Каждая вершина этого дерева описывается с помощью уникальных ключей. В вершинах находятся данные в виде триплетов.

Структура ключей следующая:

- NNNNFFFWWW00 - Ключ хранения информации о переходе, где:

NNNN - номер операции в техпроцессе,
FFF -номер перехода в данной операции,
WWW - код вида обработки в соответствии с Базой данных переходов,
— NNNNFFFWWW02 - Ключ хранения текста перехода после введения переменной информации.
— 000000000000 - Ключ хранения общей информации о технологическом процессе;
— NNNN00000000 - Ключ хранения сведений об операции, где:
NNNN - номер операции в техпроцессе, например 001000000000 - по десятой операции.



Рисунок 4.1 — Представление данных в ВСПТД

4.2. Информационно-поисковая система (ИПС), использующая методы экспертных систем (ЭС).

Ниже описывается пример построения ИПС с применением ВСПТД в САПР ТП ТЕХКОМ.

ИПС – информационно-поисковая система предназначена для поиска необходимой информации при создании технологических переходов. При этом ИПС действует по заданной схеме, используя методы ЭС, взаимодействуя с несколькими структурами БД и БЗ.

ЭС – это специализированные программные системы (ПС) полезные для специалиста конкретной предметной области, их нужно снабжать множеством высококачественных специальных знаний об этой проблемной области. Современная ЭС рассматривается как помощник или партнер при решении многих задач конкретной предметной области, усиливающих и расширяющих профессиональные возможности конкретного пользователя.

Хочется, чтобы такая система или подобные программные пакеты, чаще всего основывающиеся на прямом переборе записей БД, могли использовать все преимущества ЭС. Они предоставляют пользователю множество разнообразной, иногда излишней информации. Пользователь должен сам искать то, что ему необходимо, а если он задаст много конкретной информации, тем самым, сузив границы поиска, и при малейшей неточности в написании или построении запроса, желаемый результат не всегда получается, потому что поиск ведется прямым сравнением. Если ЭС имеет специальные знания, с помощью которых можно манипулировать с цепочками выводов, то открывается возможность эти знания использовать для проверки точности, устойчивости и правдоподобия решений и можно даже построить доводы, оправдывающие или объясняющие процесс рассуждения. ИПС на основе ЭС традиционно применяется для решения задач.

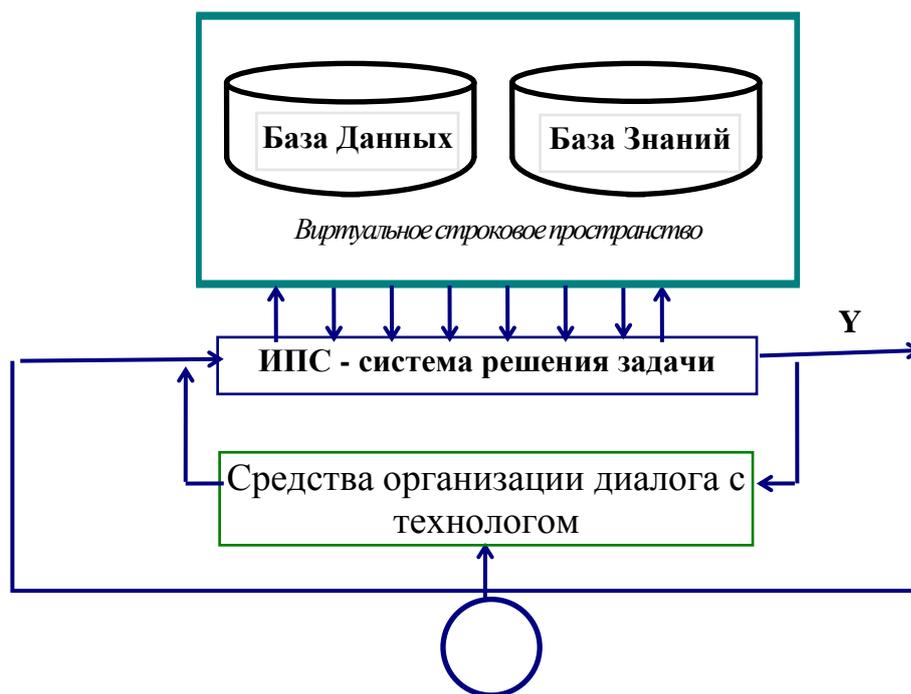


Рис. 4.1. Схема применения ИПС на основе ЭС.

4.2.1. Формализованное представление выбора инструмента

Пусть P - множество свойств, описывающих обрабатываемую поверхность ($p_i \in P$); O - множество характеристик станка ($o_i \in O$); Z - множество характеристик заготовки ($z_i \in Z$); T - множество технических требований ($t_i \in T$); E - множество характеристик инструмента, при этом E_1 - множество конструктив-

но-технологических характеристик ($E1 \subset E$), а элемент $e_i \in E1$, и определяет конкретный тип инструмента. И,

наконец, пусть W - вид обработки.

Тогда $M = P \cup O \cup Z \cup T \cup W$. Требуется сформировать группу правил ρ_i , позволяющих на основе данных из множества M получить некоторое e_i , совместное со множеством M , т.е., найти такой конструктивно- технологический шифр инструмента, с помощью которого можно было бы обработать заданную поверхность, удовлетворяя при этом свойствам множества M . Это означает, что $\rho_i(M) = e_i. \implies \cup \rho_i(M) = E1$

4.2.2. Структура базы данных

БД ИПС реализована на основе табличного представления. Вся база разбивается на таблицы типоразмеров. Каждая таблица состоит из заголовка таблицы и тела таблицы.

Заголовок таблицы содержит:

- 1) имя таблицы (ключ),
- 2) номер таблицы,
- 3) длину таблицы,
- 4) полное наименование таблицы,
- 5) структуру таблицы,
- 6) фрейм-запрос таблицы,
- 7) дату корректировки таблицы,
- 8) код чертежа,
- 9) источник информации (рис. 3).



Рис. 4.3. Структура заголовка таблицы в ИПС "Режущий инструмент"

Длина имени таблицы определяется при ориентации на конкретную предметную область и является постоянной для всей группы. Имя состоит из алфавитно-цифровой информации и задается в соответствии с правилами фасетной структуры имени таблицы, определяемыми в каждой конкретной группе таблиц.

В качестве примера рассмотрим общую структуру имени таблицы в ИПС "Режущий инструмент", представленную на рис. 4.4.

Наименование реквизита	Имя реквизита
Код источника информации	E.IST
Конструктивно-технологический шифр	E.KTS
Код группы материала режущей части инструмента	M.KRM
Признак группы обрабатываемого материала	M.PGM
Признак группы оборудования	O.PGO

Рис. 4.4. Состав реквизитов в имени таблицы типоразмеров

Наиболее сложную и существенную часть имени таблицы составляет конструктивно-технологический шифр инструмента. Тело таблицы формируется автоматически в соответствии со структурой.

4.2.3. Поиск инструмента

Ниже приводятся два примера поиска инструмента –

- с применением фрейма-гипотезы на поиск
- с применением правил вывода

4.2.3.1. Поиск инструмента с применением фрейма-гипотезы

Метод поиска инструмента состоит из двух этапов:

- 1) формирование имени таблицы с помощью ЭС;
- 2) поиск инструмента в таблице;

Вводим понятие *фрейма-гипотезы*. В контексте данной задачи термин «гипотеза»^{5*} рассматривается в более узком смысле этого слова. На основе этой гипотезы создается база сценариев ИПС, с помощью которой редуцируется пространство поиска.

В основе базы сценариев ИПС заложен метод с применением *фрейма-гипотезы* на поиск, рабочее название которого **“Таблица правил”**. В этой таблице отражены правила формирования классификационных диалогов

* 1) Гипотеза – научное предположение, выдвигаемое для объяснения какого-либо явления и требующее проверки на опыте и теоретического обоснования, для того чтобы стать достоверной научной теорией.

2) Гипотеза - условие применения правовой нормы.

ИПС при ведении баз данных типоразмеров и при составлении гипотезы на поиск (в режиме поиска при проектировании).

Формирование фрейма-гипотезы на поиск осуществляется на основе таблицы правил (рис. 4.), которая создается технологом до использования ИПС по особому правилу, либо с помощью программы формирования правил различного назначения.

В 1-й графе В.ТR- параметр, хранящий имя (или часть имени) искомой таблицы.

Значение второго столбца первой строки таблицы отвечает за основной сценарий формирования гипотезы на поиск, а результат присваивается параметру В.ТR. Следующие строки таблицы отвечают за выбор в соответствии с фасетной структурой, определенной в первой строке. Операция “ВЫБРАТЬ” содержит в качестве параметра имя таблицы, из которой выбирается очередное значение. Имя таблицы в общем случае конкатенируется с

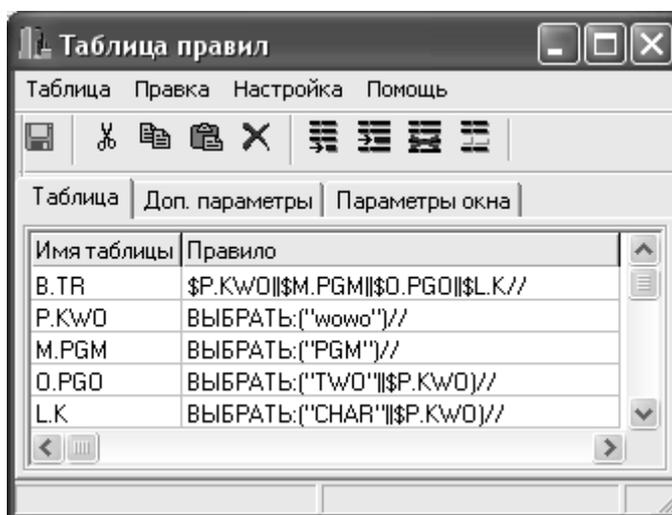


Рис. 4.5. Таблица правил

текстовой строкой, заданной в кавычках со значением параметра, который начинается с символа '\$'.Здесь “||”–символы, указывающие на конкатенацию(в имя таблицы не входят).

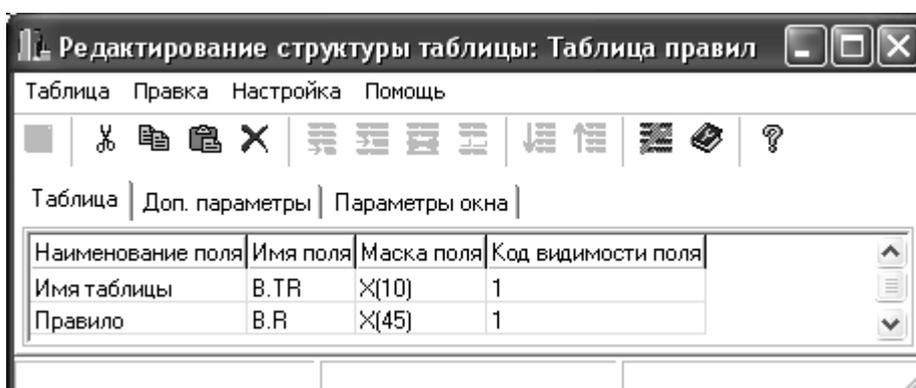


Рис. 4.6. Структура таблицы правил

Пример работы таблицы правил по выбору «расточить фаску на токарных станках»: Таблица со значением имени в триплете V.TR представляет собой правило формирования выходной строки для поиска. Расшифровывая ее, мы получаем интерфейс для пользователя. Сначала пользователь выбирает значение P.KWO из таблицы (рис. 4.6) “wowo” («Виды обработки»), и оно автоматически заносится в триплет \$V.TR=026;.

Здесь символ ‘\$’ - означает, что результат в начале диалога берется из ВСПТД, а в конце диалога кладется в ВСПТД.

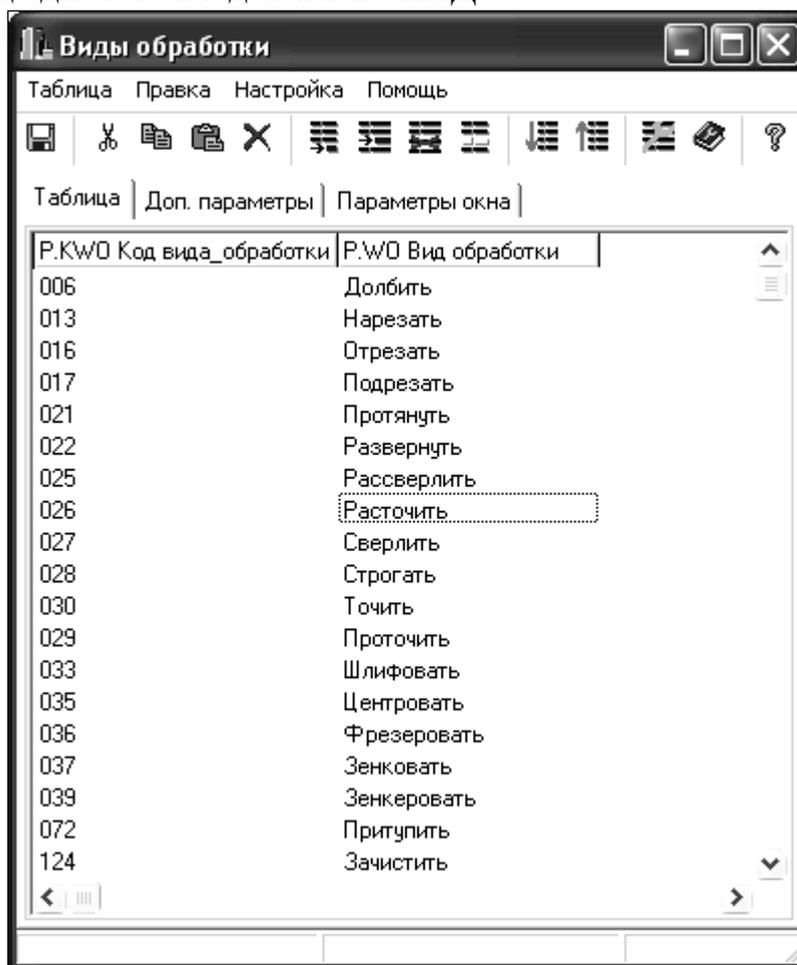


Рис. 4.7. Таблица видов обработки

Затем выбор идет за значением M.PGM, соответственно по ключу PGM из таблицы (рис. 4.8) «признак группы обрабатываемого материала», и занесением в V.TR в соответствующие позиции, теперь оно равно «02600»;

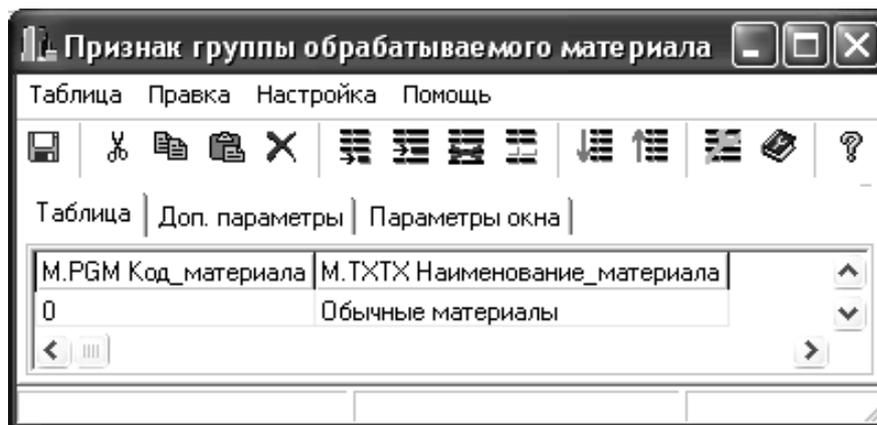


Рис. 4.8. Таблица признаков групп обрабатываемого материала

В любой момент времени технолог может прервать формирование гипотезы и дать команду на поиск. Следует иметь в виду, что чем меньше условий накладывается на формирование гипотезы, тем шире результаты поиска.

Следующий выбор O.PGO, оно состоит из нескольких ключей. Из уже выбранного \$P.KWO и нового TWO. Следовательно, таблица (рис. 4.8) будет иметь ключ вида TWO026 «виды группы оборудования (расточивание)». ИПС как бы уже отсеяла выбор для конкретного вида обработки «расточивание». Значение B.TR=0260002;

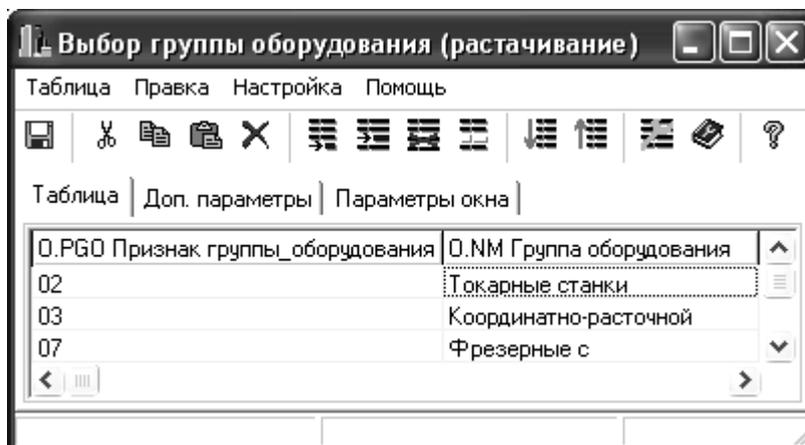


Рис. 4.9. Таблица видов групп оборудования (расточивание)

Выбор L.K, также состоит из нескольких значений: выбранного значения \$P.KWO и нового CHAR. Следовательно, таблица (рис. 4.9) будет иметь ключ вида CHAR026 «Характеристики поверхности (расточить)». B.TR примет окончательное значение "026000202";

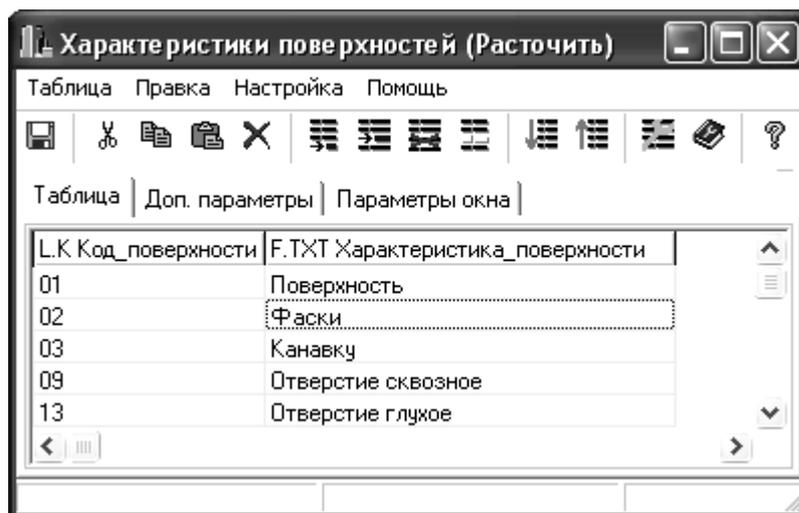


Рис. 4.10. Таблица характеристик поверхностей (расточить)

Сформировав имя таблицы (\$B.TR='026000202';), открываем таблицу (рис. 4.11) с таким номером и предоставляем выбор технологю возможных инструментов.



Рис. 4.11. Состав таблицы «расточить фаску на токарных станках»

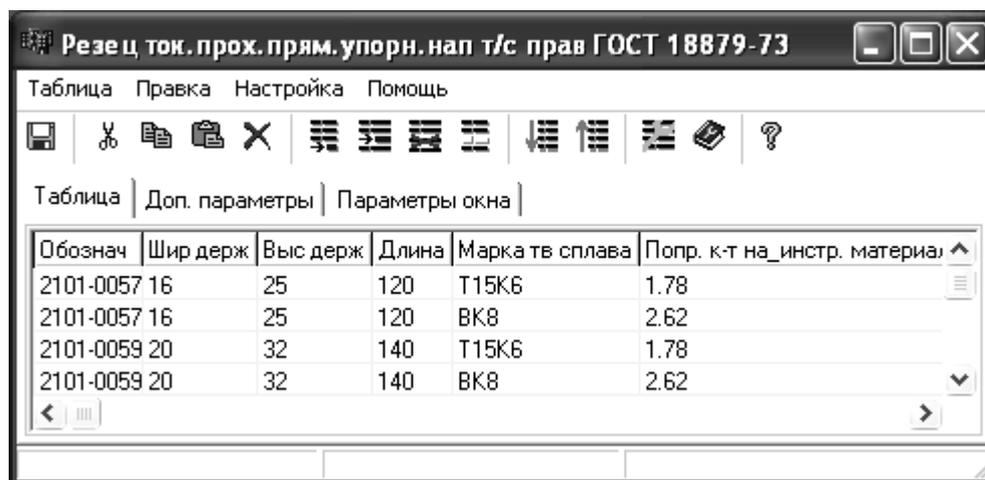
Дальнейший выбор переходит к другой базе данных "rzc.dat" для конкретных инструментов «резцов» (рис. 4.10), где осуществляется поиск типоразмера инструмента по правилу (фрейму-запросу), выбираемому из заголовка таблицы.

Фрейм-запрос выглядит следующим образом (пример):

$E.D=\$L.D$ И $E.LR>=\$L.L$ И $E.VI>=\$E.VI$ И $(E.TR1[1,4]=\$TR[1,4]$ ИЛИ $E.TR2[1,4]=\$TR[1,4])//E.LR+;E.VI-//$

Это правило означает, что надо найти строки таблицы, в которых диаметр инструмента равен диаметру обрабатываемого элемента, длина режущей части не меньше глубины отверстия, вылет инструмента не меньше заданного и первая часть типоразмера совпадает с заданной, а также отсорти-

ровать найденные результаты по возрастанию параметра E.LR и убыванию параметра E.VI.



Обознач	Шир держ	Выс держ	Длина	Марка тв сплава	Попр. к-т на_инстр. материал
2101-0057 16	25	120	120	T15K6	1.78
2101-0057 16	25	120	120	BK8	2.62
2101-0059 20	32	140	140	T15K6	1.78
2101-0059 20	32	140	140	BK8	2.62

Рис. 4.12. Таблица найденных результатов

Если после поиска ИПС нашла несколько вариантов, то выбор окончательного результата должен, прежде всего, определяться отдельно задаваемым критерием, характеризующим данную задачу.

Так работает программа поиска инструмента. Соответственно, можно таким же способом построить БЗ по СОЖ, по оборудованию, и другим необходимым для технолога ИПС. При этом программа универсальна и не требует вмешательства программиста.

4.2.3.1. Поиск инструмента с применением правил вывода

При автоматизированном назначении инструмента конструктивно-технологический шифр (КТШИ) предложено выбирать с помощью ЭС. Автоматизированное формирование КТШИ с помощью информационно-логических таблиц было предложено в работе [14,с.35]. Однако, многообразие вариантов затрудняет их адаптацию к условиям производства. Задача назначения инструмента состоит из следующих этапов:

1.Выбор конструктивного кода инструмента с учетом требований инструментальной наладки.

2.Поиск типоразмера инструмента исходя из параметров обрабатываемых элементов, оборудования и других технологических сведений.

Следует иметь ввиду, что если второй этап решается с помощью ИПС табличного представления, в которой правила поиска “привязываются” к таблицам и задаются в момент настройки экспертами-технологами, а выход на таблицу обеспечивается через конструктивный код, то первый этап представляет из себя довольно трудную задачу, при решении которой, в

зависимости от предъявляемых требований, должен учитываться ряд дополнительных факторов, среди которых могут быть следующие:

1. Обеспечение минимальной стоимости;
2. Обеспечение заданной производительности;
3. Минимизация состава инструментов;
4. Обеспечение заданного уровня надежности.

Выбор конструктивного кода на первом этапе определяется с применением ЭС на основе производционных правил. Для обработки каждой поверхности может быть предложено несколько типов инструментов. В результате формируется матрица $\{K_{ij}\}$, где K_{ij} - код j -го инструмента, обрабатывающего i -ю поверхность. По кодам инструмента определяется матрица стоимости C_{ij} .

Из модуля расчета режимов резания получается матрица $\{T_{ij}\}$, где T_{ij} - время обработки i -ой поверхности j -м инструментом. С использованием алгоритмов оптимизации определяется состав инструментов. По алгоритму расчета уровня надежности определяется оценка надежности выбранной группы инструментов [19]. Наиболее сложную и существенную часть имени таблицы составляет конструктивно-технологический шифр инструмента (КТШИ).

Аналогично строится конструктивно-технологический шифр и для других инструментов, таких как сверло, фреза и т.п..

Тело таблицы формируется в соответствии со структурой. При этом в режиме просмотра автоматически генерируется форма таблицы (фрейм-терминал) для отображения на экране дисплея.

Предлагается метод поиска необходимого типоразмера инструмента, состоящий из двух этапов:

- формирование имени таблицы с помощью ЭС;
- поиск инструмента в таблице.

Возникает задача на основе накопленной в процессе проектирования ТП информации (описание поверхностей, заготовки, технические требования, параметры оборудования и т.п.) получить КТШИ.

Для этих целей накапливается база знаний в виде правил вывода. Правила вывода разбиваются на группы по выполняемым функциям. Управляющая структура определяет последовательность их исполнения. При формировании шифра инструмента в 1-ю группу включаются правила, определяющие вид инструмента (резец, сверло, зенкер и т.п.). После определения вида инструмента вызывается группа правил, которая выводит оставшиеся составляющие шифра соответствующего вида инструмента.

Если ДМВ находит несколько правил с истинным условием, то выбор результата определяется отдельно задаваемым критерием, который может изменяться в процессе работы.

Ниже приводится фрагмент базы знаний и ВСПТД при формировании конструктивного шифра инструмента.

Пусть в некоторый момент проектирования в ВСПТД находится некоторое множество фактов, часть которых здесь представлена.

...\$L.D=3.5;\$L.L=10;\$P.SE='221440';\$L.KW=12;\$E.NST=1;\$E.KRM=1;\$O.GRO='20001';\$E.VI=35;\$E.NI=1;\$L.WOB=27;\$M.PGM=3;...

Здесь префикс L указывает на принадлежность к обрабатываемой поверхности, префикс E - на инструмент, O - на оборудование. Далее имена параметров в ВСПТД имеют следующий смысл:

D - диаметр; L - длина; SE - шифр поверхности; KW - качество; NST - номер стандарта. TT - код технических требований; NI - количество; WOB - вид обработки; KRM - код материала режущей части; KTS - код инструмента.

(*)

Возможные значения параметра \$L.TT:

03 - некруглость поверхности;

04 - нецилиндричность поверхности;

18 - непараллельность поверхности;

34 - радиальное биение поверхности;

37 - полное радиальное биение поверхности;

58 - полное торцевое биение поверхности;

Возможные значения параметра \$L.WOB

025 - рассверлить;

027 - сверлить.

Далее приводится группа выполненных правил:

ПРАВИЛО 2. ЕСЛИ ((\$L.WOB=25 ИЛИ \$L.WOB=27) И НЕТ(\$L.TT)) ТО \$E.KTS[1,2]='21'... "Сверло общего назначения";.

ПРАВИЛО 7. ЕСЛИ ((\$E.KTS='41' ИЛИ \$E.KTS='21') И \$E.NI=1 И \$L.KW >= 12 И \$L.KW <= 13) ТО \$E.KTS[3,1]='1'... "Конструкция рабочей части: спиральное обычное";.

ПРАВИЛО 11. ЕСЛИ (\$E.KTS='21' И (\$O.GRO NOT='10000' И \$O.GRO NOT='10001')) ТО \$E.KTS[4,1]='0' ... "Правое";.

ПРАВИЛО 13. ЕСЛИ (\$E.NST > 2) ТО \$E.KTS[5,1]='5'... "Средняя серия";.

ПРАВИЛО 14. ЕСЛИ (\$E.KTS[1,2]='21' ИЛИ \$E.KTS[1,2]='41') ТО \$E.KTS[6,1]='1'... "Цилиндрическое";.

Итак, сформирован код инструмента E.KTS='211051', а после конкатенации с параметрами E.IST, E.KRM, M.PGM в соответствии с табл.4.3., получается имя таблицы '121105113' из ИПС "Режущий инструмент", в которой осуществляется поиск типоразмера инструмента по правилу (фрейму-запросу), соответствующему выбранной таблице. Синтаксис условия этого правила соответствует вышеописанным, а действием при успешном выполнении будет запись реквизитов найденной строки в форме триплетов в ВСПТД. Для найденной таблицы правило выглядит следующим образом:

((E.D=\$L.D И E.LR>=\$L.L И E.VI >= \$E.VI: И (E.TR1[1,4]=\$TR[1,4] ИЛИ E.TR2[1,4]=\$TR[1,4])//E.LR+;E.VI-//

Вербально это правило означает следующее: найти строки таблицы, в которых диаметр инструмента равен диаметру обрабатываемого элемента, длина режущей части не меньше глубины отверстия, вылет инструмента не меньше заданного и первая часть типоразмера совпадает с заданной.

СВЕРЛО СПИРАЛЬНОЕ С ЦИЛИНДРИЧЕСКИМ ХВОСТОВИКОМ ГОСТ.10902-77

Номер строки (NUM)	Типоразмер правый (E.TR1)	Типоразмер Левый (E.TR2)	Диаметр Инструмента (E.D)	Длина реж. Части (E.LR)	Вылет инстр. (E.VI)
1	2300-0155	2300-7516	3.0	33.0	35
2	2300-0156	2300-7518	3.1	36.0	40
3	2300-0157	2300-7522	3.15	36.0	40
4	2300-0158	2300-7524	3.20	36.0	40
5	2300-0159	2300-7526	3.30	36.0	40
6	2300-0160	2300-7528	3.35	36.0	40

Рис. 4.13. Фрагмент найденной таблицы инструментов

Выражение E.LR+;E.VI-// - означает: сортировать найденные результаты по возрастанию параметра E.LR и убыванию параметра E.VI. При наличии нескольких строк удовлетворяющих условию выбирается первая (с учетом сортировки) или указанная технологом.

Вообще говоря, здесь можно применить тот же метод выбора окончательного решения с учетом требуемых критериев оптимизации.

4.3. Общие принципы извлечения знаний в ЭС САПР ТП

С выявлением знаний обычно связано два типа источника знаний: прямые или знания 1-го рода и опосредованные или знания 2-го рода. Прямой тип – это знания полученные от эксперта, специалиста или из экспертной среды. Опосредованные – знания, полученные из книг, документов, инструкций, справочников.

ЭС в процессе приобретения знаний 1-го рода, разделяются на две фазы:
Построение прототипа базы знаний БЗ;

Уточнение знаний, для достижения желаемого уровня работы системы.

Задача построения прототипа БЗ решается с помощью текстовых и интеллектуальных редакторов, уточнение прототипа БЗ - решается с помощью автоматического тестирования знаний. Другие средства приобретения знаний (системы индуктивного вывода, системы понимания текстов и т.д.) находятся на стадии тестирования и экспериментальных исследований.

Наиболее популярным методом извлечения знаний, по данным работ отечественных разработчиков ЭС, для экспертов оказалось структурированное интервью. Как уже отмечалось, в САПР ТП механической обработки вы-

сок удельный вес знаний первого рода. Это, в частности, является определяющей причиной для выбора указанного метода.

В результате исследования технологических информационных структур и с учетом анализа рекомендаций, предложен процесс приобретения технологических знаний, который мы можем разделить на 3 этапа:

- построение скелетной структуры БЗ;
- построение прототипа БЗ;
- уточнение БЗ.

Скелетная структура редактора БЗ строится в виде графа, где вершинами являются таблицы (домены), которые заполняются ведущими экспертами-технологами и инженером по знаниям.

Процесс извлечения знаний осуществляется экспертами-технологами в диалоговом режиме. Он представляет собой обход этого графа по дугам, которыми являются выбираемые строки таблиц. В процессе последовательных ответов на вопросы системы формируется знания, которые можно разделить два уровня:

- определение группы, к которой относится данное "знание";
- само "знание", представляющее фрейм или правило вывода.

При возникновении необходимости ввести новое правило или фрейм, минуя обход графа по дугам, то в этих случаях предоставляется возможность свободного ввода очередного знания. При этом автоматически осуществляется синтаксический и, где это возможно, семантический контроль и анализ. При корректировке базы фреймов в случае удаления или изменения ссылок осуществляется навигация по всей базе, проверяющая корректность ссылок.

Требования к расширяемости редактора БЗ не должно ограничиваться простым подключением новых процедур. Необходимо, чтобы старые модули успешно работали в новом окружении, тогда как прикладная семантика обрабатываемых данных частично изменяется. Реальный шаг на пути к решению этой проблемы основан на интенсивном применении демонов.

При формировании базы знаний технологического назначения, в результате исследований и с учетом разработанных инструментальных средств доступа нижнего уровня предложен единый механизм хранения знаний, который распространяется на все знания, представимые в виде строчной информации.

В качестве базового метода выбран пакет программ, работающий с ВСПТД.

Знания могут быть нескольких типов: продукции, фреймы, формулы и т.д. (см.п.2). С учётом возможного распределения баз знаний по разным персональным ЭВМ и применения их в локальных подсистемах знания разных типов располагаются физически в разных виртуальных пространствах.

При вводе и редактировании знаний одного типа предложен следующий принцип:

- знания разбиваются на группы;

-по специальному сценарию, имеющему древовидную структуру, вычисляется код группы;

-каждая группа размещается в отдельной виртуальной строке с ключом, соответствующим коду группы.

Таким образом, редактор БЗ выполняет две основные функции:

-формирование кода группы;

-редактирование группы.

Каждая группа правил, располагается на одной виртуальной строке. Фраза содержит подробное пояснение относительно назначения данной группы и участвует в блоке объяснений при ответе на вопрос "почему?". Описание списка параметров представлено ниже. После ввода или редактирования очередной порции осуществляется синтаксический анализ. Формирование знаний внутри группы существенно зависит от выбранного формализма. Например, выяснилось, что ввод фреймов-переходов можно осуществлять по такому же принципу, как и формирование кода группы. При тщательной проработке скелетной структуры наполнение БЗ фреймами-переходами проводится без участия инженера по знаниям. Далее предлагается фрагмент формирования фрейма-перехода с применением редактора БЗ (рис. 4.14.1 - 4.14.4).

Шаг 1.Выбираем вид обработки “Накатать” (рис. 4.14.1.).

Получили фрейм:

“Накатать”.

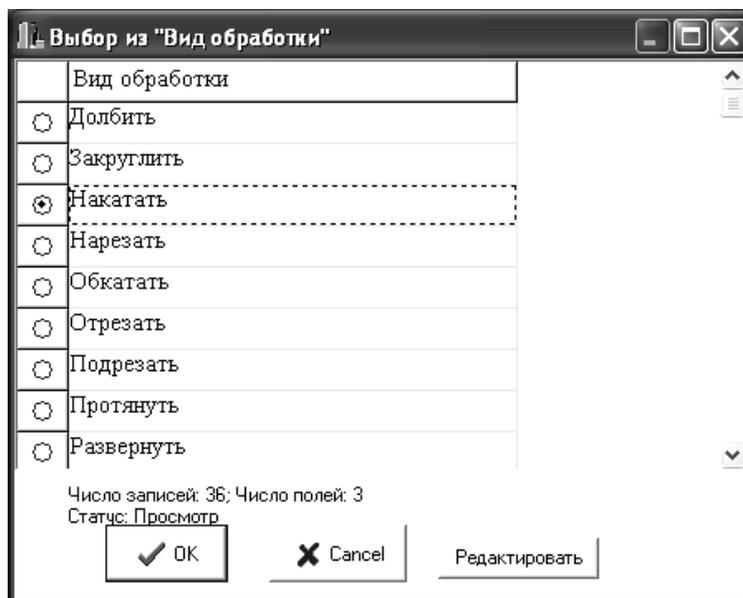


Рис. 4.14.1.Выбираем вид обработки “Накатать”

Шаг 2 Выбор характеристик поверхности “шлицы” (рис. 4.14.2.)

Получили фрейм:

”Накатать шлицы” \$L.OB.

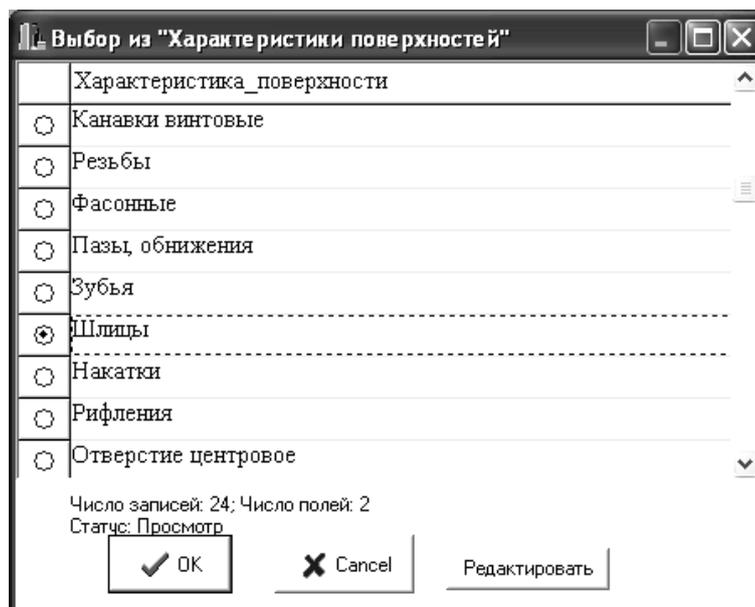


Рис. 4.14.2. Выбор характеристик поверхности “шлицы”

Шаг 3 выбор условий: “Выдерживая технические требования” (рис. 4.14.3.)

Получили фрейм:

”Накатать шлицы” \$L.OB “Выдерживая технические требования”.

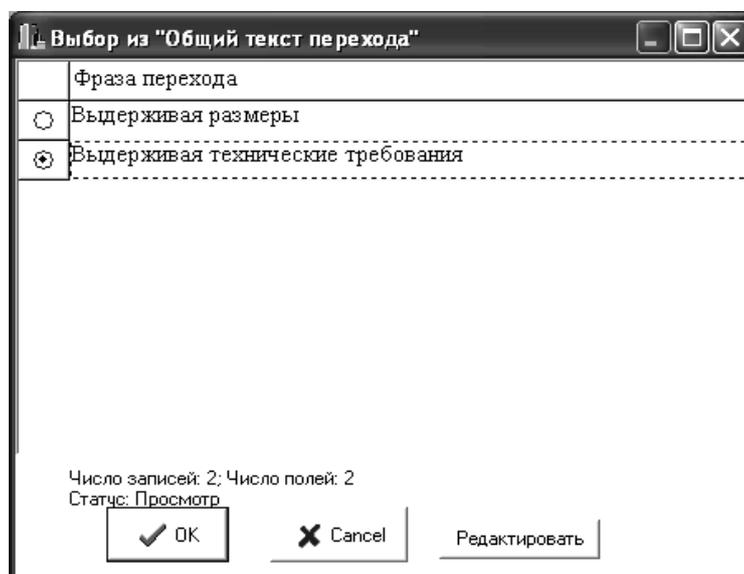


Рис. 4.14.3. Выбор условий: “Выдерживая технические требования”

Шаг 4 выбор параметров элемента обработки (рис. 4.14.4.).

Получили фрейм :

”Накатать шлицы” \$L.OB “Выдерживая технические требования” <ПС \$L.BETA \$L.DN \$L.KOL \$L.L >.

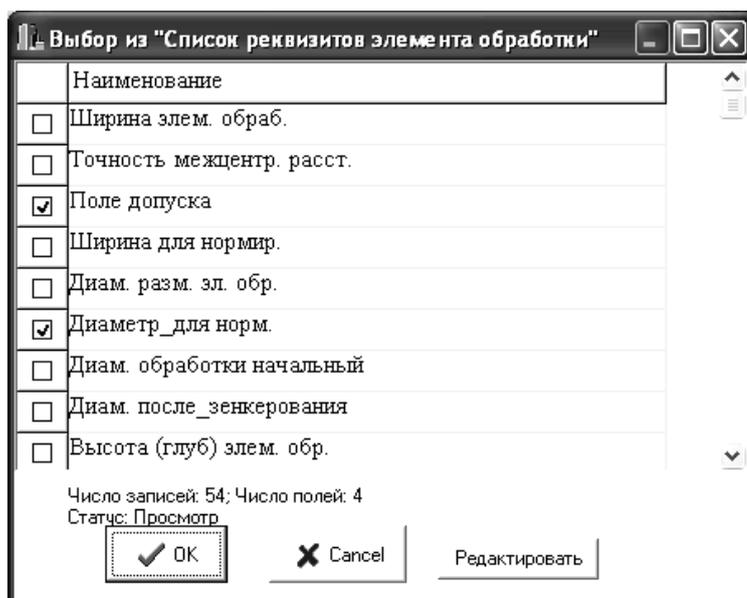


Рис. 4.14.4. Выбор параметров элемента обработки

Ввод знаний, представимых в виде продукций, значительно сложнее, и здесь большая роль по-прежнему остается за инженером по знаниям, который при организации структурированного интервью с экспертом ориентируется, прежде всего, на опосредованные знания. Задача когнитолога⁶ сводится к двум основным позициям:

- извлечение знаний от экспертов на понятийном уровне;
- формализация знаний.

Для извлечения знаний в виде продукций в САПР ТП механической обработки предложена следующая стратегия. Рассматриваемая предметная область представляет собой набор некоторых объектов, среди которых можно выделить два основных класса: статические объекты, описывающие конкретную предметную область (оборудование, инструмент, заготовка и т.п.); динамические объекты, описывающие некоторые отношения между характеристиками статических объектов. Вместе эти множества назовем семантическими.

Как уже отмечалось, вся производственная система представляет собой множество производственных объектов, каждый из которых обладает некоторым набором характеристик. Для адекватного обеспечения взаимодействия технолога с ЭВМ доступ к большинству этих характеристик осуществляется на понятийном уровне. Для этих целей используется интерфейс понятий. Кроме конкретных производственных объектов в процессе эксплуатации система накапливает множество семантических объектов (фреймы, правила и т.п.), каждый из которых в той или иной мере может влиять на формирование новых знаний.

Пусть конечная цель представлена в виде некоторого множества фактов Ф. Требуется выяснить, на основании каких данных и правил выводится это

⁶ Когнитолог - инженер по знаниям

множество. В предположении, что в нашем распоряжении имеются опосредованные знания или высококвалифицированный эксперт, хорошо понимающий проблему в целом, осуществляется редуцирование области понятий, которые могут воздействовать на каждый факт из множества Φ . Т.е., крупными "мазками" определяются те объекты, часть характеристик которых может влиять на получение данной цели. Таким образом, для каждого факта Φ_i определяется множество объектов Q_i . Каждый объект описывается кортежем признаков P (см. описание словаря). Стратегия извлечения знаний заключается в том, что выявленные на очередном шаге понятия становятся базисом для последующего интервью.

После такого редуцирования понятий осуществляется структурированный опрос эксперта-технолога, на основании протокола которого и будут формироваться правила. Для каждой пары факт-объект определяются те характеристики объекта, которые могут участвовать в выводе. При этом на решение могут влиять следующие состояния характеристик объекта:

- отсутствие характеристики;
- присутствие характеристики;
- присутствие и значение характеристики;
- отношения объекта с другими объектами.

Объединение одноместных (только по одному отношению) и двухместных операций на основе указанных вариантов можно рассматривать как приближенную модель правила для данной предметной области, которая и будет служить прототипом, а в идеальном случае и некоторой продукцией, участвующей в выводе.

Каждому выводимому факту ставится в соответствие некоторое подмножество из множества понятий I в предположении, что характеристики этих объектов могут участвовать в выводе.

4.4. Структура экспертной системы

Структура экспертной системы представлена на рис. 4.15. Как видно из рисунка ключевой частью экспертной системы является дедуктивная машина вывода (ДМВ). ДМВ – программа, которая осуществляет логический вывод из предварительно построенной базы фактов и правил в соответствии с законами формальной логики. Факты хранятся в контексте задачи, а правила – в базе знаний. Также ДМВ связана с различными базами данных, из них система получает недостающие факты и осуществляет поиск параметров для представления пользователю. Стоит отметить, что деление на базы данных и базы знаний весьма условно. Технологические данные можно классифицировать по двум типам: статические и динамические. Статические данные, как правило, описывают конкретные объекты. Динамические данные представляются в

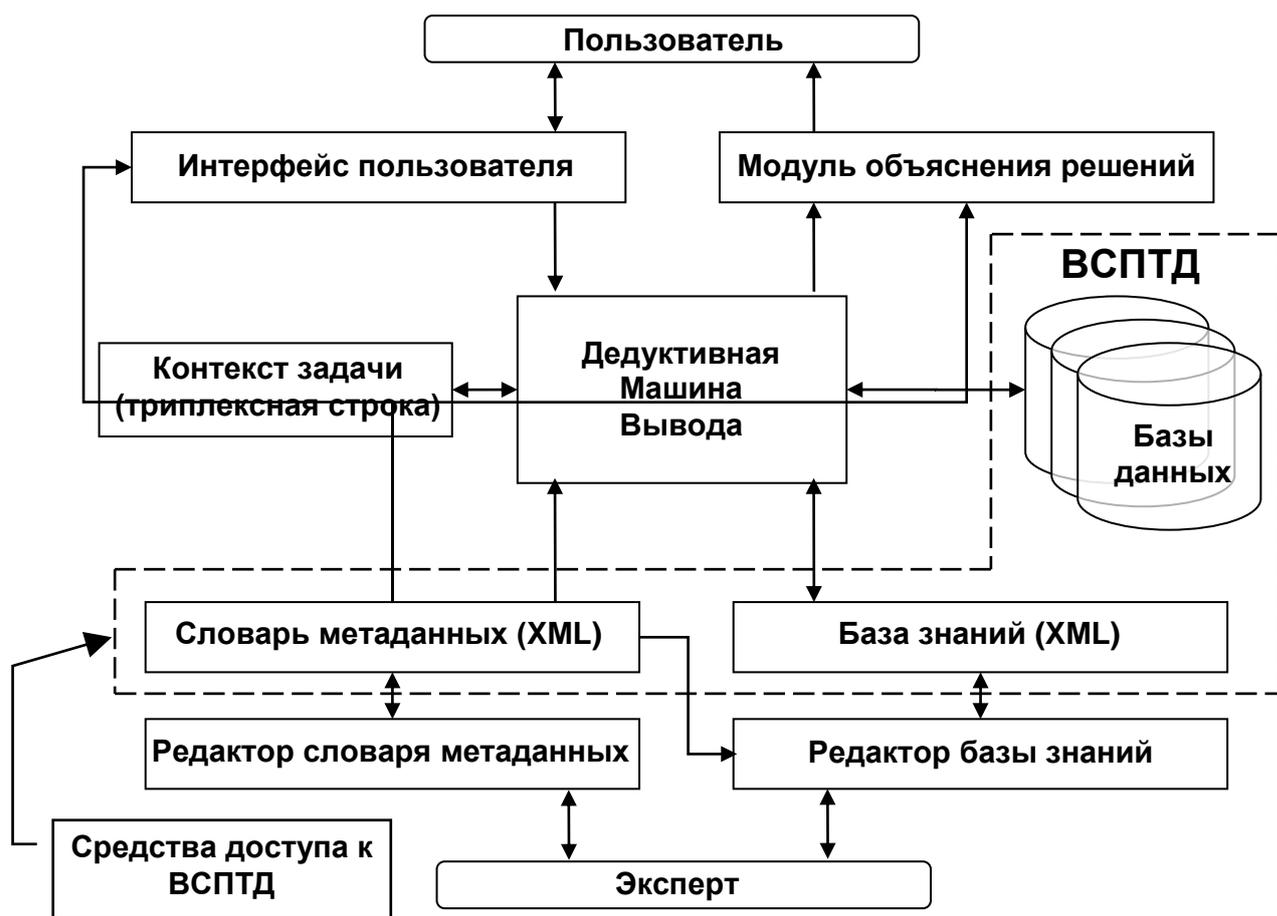


Рис. 4.15. Структура экспертной системы

в процедурном виде. А поскольку они, в свою очередь, могут содержать статические данные, и, будучи параметризованными, во время работы использовать другие статические данные, то здесь прослеживается связь между статическими и динамическими данными, или, в терминологии ЭС, между данными и знаниями. Отсюда вытекает требование, исходя из которого, каждый предметный модуль системы должен иметь равные возможности получения, как данных из базы данных, так и знаний из БЗ. Принцип равных возможностей при работе с БД и БЗ из прикладных программ обеспечивает эволюционный принцип развития САПР ТП с применением ЭС.

4.5. Взаимоотношения ВСПТД и формата RUSMARC

Эффективное обеспечение информацией экспертных систем предполагает возможность взаимодействия с коммуникативными форматами, широко представленными в ИНТЕРНЕТ. Направления, в которых проводились работы по развитию формата ВСПТД, были связаны также и с получением информации, представленной в *Российском коммуникативном формате RUSMARC*. Российский коммуникативный формат не является совершенно новой, отдельно стоящей разработкой. - это российская версия Международ-

ного коммуникативного формата UNIMARC, в трактовке и категориях действующих в России ГОСТов и Правил каталогизации, базирующаяся на выборе наиболее общих схем представления элементов данных. С практической точки зрения это означает, что любая запись, переданная в Российском коммуникативном формате, должна адекватно восприниматься любым программным обеспечением, о котором заявлено, что данное программное обеспечение работает с форматом UNIMARC.

4.5.1. Структура формата RUSMARC

Российский коммуникативный формат, как и UNIMARC, реализует библиографическую запись в машиночитаемой форме в структуре международного стандарта ISO 2709. Каждая библиографическая запись, подготовленная для обмена, должна содержать (рис. 4.16):

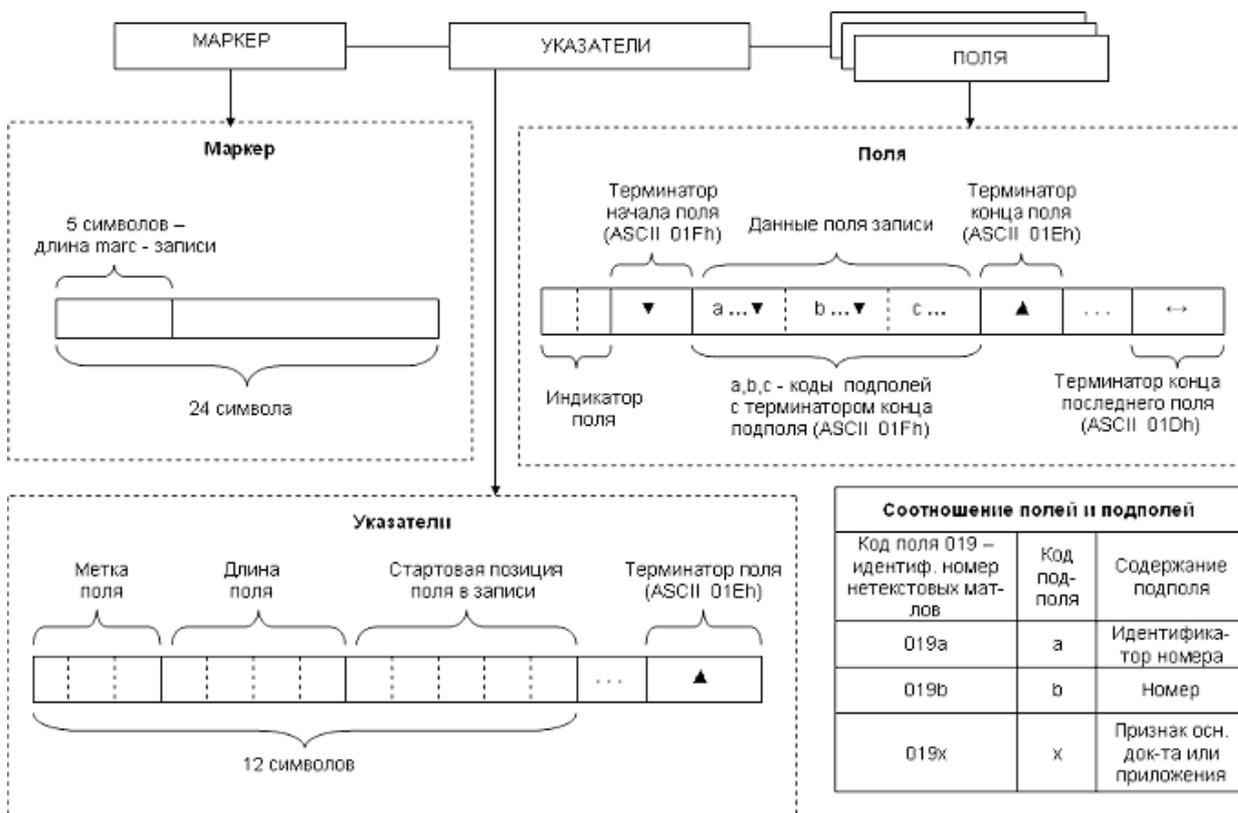


Рис. 4.16. Структура MARC – записи

- **МАРКЕР ЗАПИСИ**, состоящий из первых 24-х символов элемент записи, в котором определяются ее основные характеристики;
- **УКАЗАТЕЛЬ**, состоящий из 3-х цифровой метки для каждого поля данных, длины поля и позиции начального символа, относящегося к первому полю данных;

- ПОЛЯ ДАННЫХ переменной длины, отделенные друг от друга разделителем поля;

Общая структура:

МАРКЕР ЗАПИСИ	УКАЗАТЕЛЬ	ПЕРЕМЕННЫЕ ПОЛЯ	%
---------------	-----------	-----------------	---

% - Разделитель записи

В дальнейшем устанавливается, что данным в полях необязательно предшествуют индикаторы, и они необязательно подразделяются на подполя. Российский коммуникативный формат, как и UNIMARC, использует следующие положения, установленные Международным стандартом ISO 2709. Маркер записи: ISO 2709 предписывает, что каждая запись начинается с 24-х символьного МАРКЕРА ЗАПИСИ. Он содержит данные, относящиеся к структуре записи, определения которых даются в стандарте ISO 2709, а также некоторые элементы данных, выделенные для особого применения. Эти элементы данных, определяемые при применении, связаны с типом записи, ее библиографическим уровнем и положением в иерархии уровней, степенью полноты записи и использованием правил ISBD или других, основанных на ISBD, правил подготовки записи. Элементы данных в МАРКЕРЕ ЗАПИСИ предназначены, прежде всего, для обработки записи и только косвенно для идентификации самих библиографических материалов.

Указатели: За маркером записи следуют указатели. Каждая статья указателя состоит из трех частей: трехзначной метки, четырехзначного числа, указывающего длину поля данных, и пятизначного числа, указывающего начальную символьную позицию. Другие символы в статье указателя не допускаются. Более подробное описание формата RUSMARC представлено в различных источниках [11,12, 13].

Для работы с форматом MARC применяются сложные и дорогие программные системы, приобретение которых не всегда целесообразно. В связи с этим была поставлена задача перевода данных из формата MARC в представление ВСПТД (конвертирование). В рамках этой проблемы были выработаны следующие соглашения:

- имена подполей представляются в виде nnnx, где **nnn** – код поля (в формате MARC), а **x** – код подполя;
- имена индикаторов представляются в виде Innn;
- в качестве префиксов в триplete применяется номер повторяемости поля;
- повторяемые подполя представляются как вектор значений с задан-

ным разделителем;

- маркеру MARC-записи присваивается код поля 999.

Рассмотрим пример библиографической записи:

Митрофанов С.П., Куликов Д.Д., Миляев О.Н., Падун Б.С. Технологическая подготовка гибких производственных систем. / Под общей редакцией С.П. Митрофанова. Л.: Машиностроение, 1987. – 352 с.

Представление записи в формате RUSMARC:

200: 1 \$a Технологическая подготовка гибких производственных систем. \$g Под общей редакцией С.П. Митрофанова

210: \$aЛ.\$c Машиностроение \$d1987

215:: \$a352 с.

700: 1\$a Митрофанов \$bС.П.

700: 1\$a Куликов \$bД.Д.

700: 1\$a Миляев \$bО.Н.

700: 1\$a Падун \$bБ.С.

Представление записи в формате ВСПТД:

\$0.200a='Технологическая подготовка гибких производственных систем'; \$0.200g='Под общей редакцией С.П. Митрофанова'; \$0.210a='Л.'; \$0.210c='Машиностроение'; \$0.210d=1987; \$0.215a='352 с'; \$0.700a='Митрофанов'; \$0.700b='С.П.'; \$1.700a='Куликов'; \$1.700b='Д.Д.'; \$2.700a='Миляев' \$2.700b='О.Н.'; \$3.700a='Падун'; \$3.700b='Б.С.'; \$0.I200a='1'; \$0.I700='1'; \$1.I700='1'; \$2.I700='1'; \$3.I700='1';

5. Словарь метаданных

Словарь метаданных - наиболее универсальное средство, обеспечивающее документирование, хранение и контроль описания данных." При использовании словаря данных может возникнуть опасная иллюзия достижения абсолютного контроля метаданных. Необходимо, однако, чтобы дисциплина контроля нашла поддержку у руководства, была принята и использовалась проектировщиками на регулярной основе, была признана и соблюдалась программистами" [8].

Для принятия решений на всех уровнях проектирования требуется полная, точная и своевременная информация. Одно из таких средств - системы словарей-справочников данных (СССД) [9] или метаданных. С помощью СССРД осуществляется контроль за данными, некоторыми элементами синтаксиса и семантики БЗ, обеспечивается лингвистическое обеспечение системы при ведении диалога и формировании отчетных документов. Следующей принципиально важной функцией словаря является обеспечение интерфейса с пользователем на понятийном уровне. Все элементы предметной области (фреймы, правила и т.п.) имеют общий механизм интерфейса понятий. Это позволяет избавить пользователя от необходимости работы с кодируемой информацией, обеспечивает формирование некоторых видов фреймов в удобной для него форме.

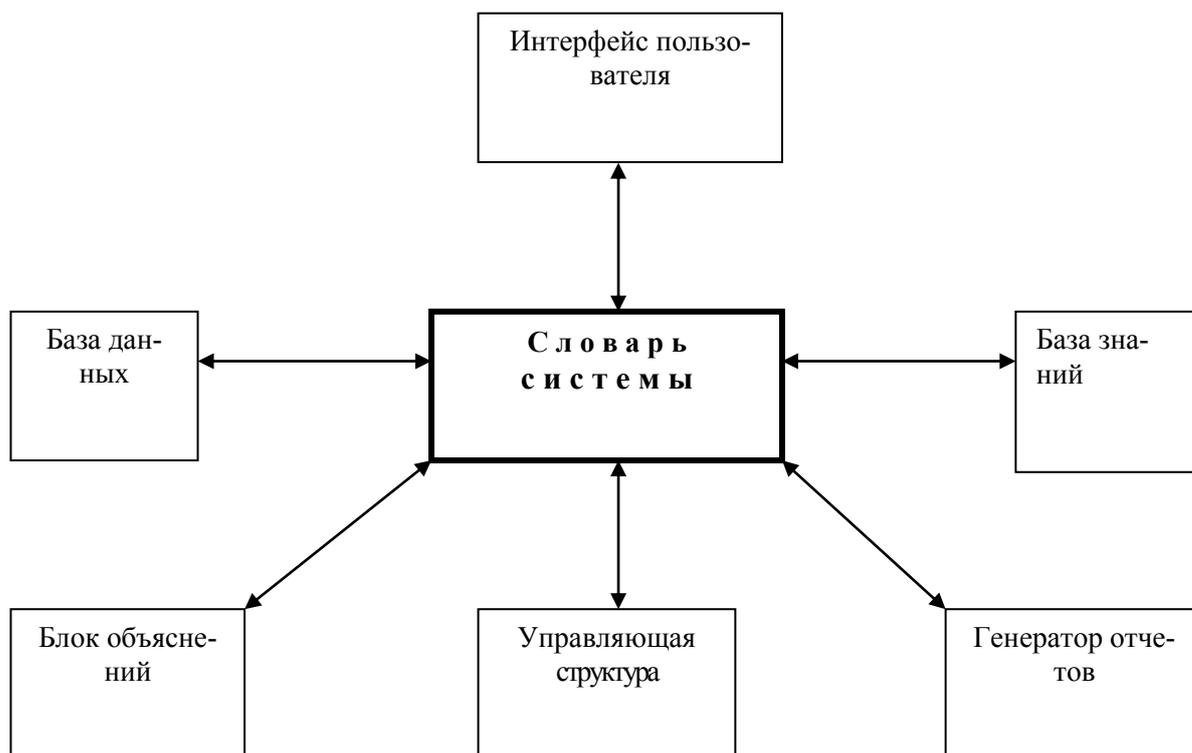


Рис. 5.1. Место словаря в ЭС САПР ТП

Кроме того, требование открытости ЭС предполагает и открытый доступ пользователя к метаданным. По этим причинам при работе с ЭС уделяется особое внимание функции ведения словарей данных (СД). О каждом описываемом в СД реквизите содержится, как минимум, следующая информация: мнемоническое обозначение, полное наименование, формат и способ получения реквизита. В процессе функционирования ЭС из соображений эффективности весь словарь или большую его часть целесообразно держать в оперативной памяти. Исследования показали, что при разработке САПР ТП из различных модулей в работу включается свыше 1000 наименований параметров, что предъявляет существенные требования к эффективной организации словаря в оперативной памяти. При этом возникла необходимость быстрой адаптации словаря как при подключении в систему, так и при исключении из нее различных подсистем. Исходя из этих положений разработана методика ведения словаря данных (СД) САПР ТП, которая основывается на представлении данных в виде триплетов.

Словарь данных разбивается на две группы:

- локальный словарь данных (ЛСД);
- глобальный словарь данных (ГСД);

ЛСД используется в описании конкретного предметного модуля и/или объекта, подробно описывает каждый параметр и применяется как при работе в автономном режиме, так и при включении модуля в систему. ГСД представляет собой сжатое описание словаря системы, обеспечивает межмодульную связь, контроль типов данных и используется только при функционировании системы. Каждый ЛСД имеет свое имя, которое

соответствует префиксу описываемого объекта (см. описание триплетов), и представляет из себя таблицу. Здесь имя реквизита взято из общепринятого соглашения при описании триплетов и представляет вторую часть (без префикса) имени триплета.

ЛСД формируется в диалоговом режиме с участием эксперта-технолога и инженера по знаниям (лиц, принимающих решение - ЛПР) информационного обеспечения. ГСД формируется автоматически в процессе формирования ЛСД.

Процесс формирования ЛСД и ГСД изображен на рис. 5.2.

Т.о., ГСД можно представить как объединение нескольких ЛСД. Пусть P - вектор префиксов, N - вектор имен, R - матрица кодов способов получения реквизитов.

Тогда ГСД строится по следующей схеме:

ГСД = (P, N, V, R) , где

$$V_{ij} = \begin{cases} >0, & \text{если пара } (P_i, N_j) \text{ семантически допустима,} \\ 0, & \text{если пара } (P_i, N_j) \text{ семантически недопустима} \end{cases}$$

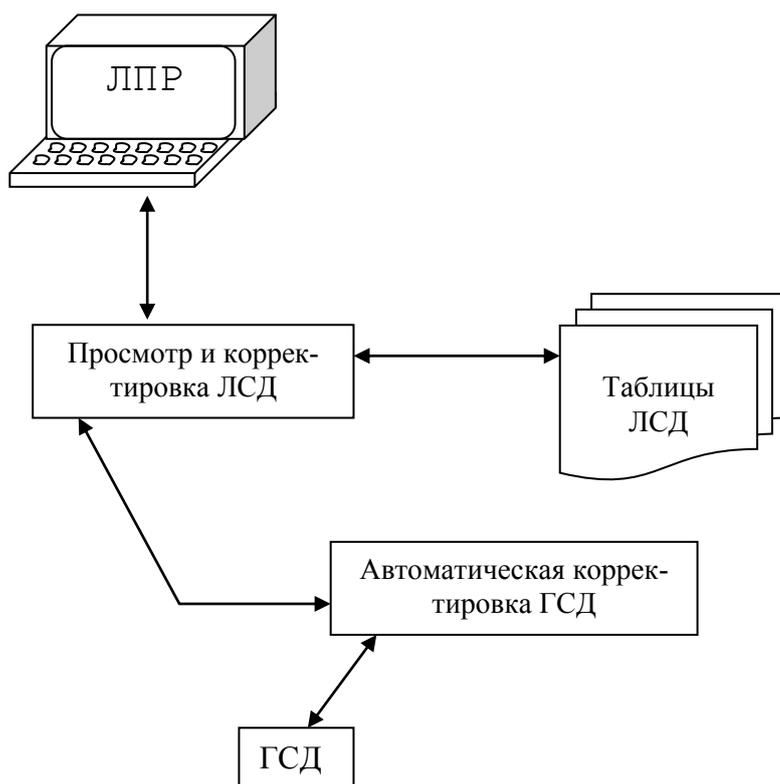


Рис. 5.2. Принцип формирования ЛСД и ГСД

Если значение $V_{ij} > 0$, то оно определяет код формата соответствующего реквизита. С помощью матрицы R - определяется код способа получения реквизита. Эта матрица служит для организации работы управляющей структуры.

Кроме основных ЛСД в словарь могут быть включены вспомогательные. Например, таблица аффиксов, которая применяется после морфологического синтеза словоформ для формирования фраз на естественном языке. В некоторых ЛСД используются усеченные парадигмы, позволяющие формировать окончания слов в соответствии с правилами русского языка.

В процессе функционирования ЭС САПР ТП механической обработки словарь оказывает существенное влияние на работу различных блоков системы (рис. 5.1.).

Роль словаря актуальна при организации интерфейса пользователя, блока объяснений, генератора отчетов, формировании базы знаний и базы данных, организации управляющей структуры. Ниже описывается связь словаря с генератором отчетов. Во всех остальных случаях схема работы со словарем в основном аналогична.

5.1 Структура словаря

Структурно словарь представляет собой табличный файл, создаваемый пользователем при помощи средств аппарата работы с табличной базой данных. Он представляет собой набор таблиц с именами типа V-<префикс>, где <префикс> это произвольная буква латинского алфавита, общая для всех имен таблицы (например, E - режущий инструмент, N - нормирование и т.д.). Имя однозначно идентифицирует некое полученное значение в системе, где в процессе вычисления значение имени может меняться. Например: \$E.NM="режущий инструмент".

Сами таблицы имеют одинаковую структуру для первых трех граф:

- имя поля;
- формат поля;
- наименование поля.

• Имя поля

Имя поля формируется в соответствии с правилами организации триплетов (см. 2.2.1). Применяется при обмене табличных данных с триплексными строками.

• Формат поля

Форматы поля, разрешенные для объявления, следующие:
999999 или 9(6) -целое число;
999V999 или 9(3)V9(3) или 999V9(3) -действительное без знака
S999V999 или S9(3)V9(3) -действительное со знаком;
XXXX или X(4) –текст.

Примечание: в новой версии произошел уход от описания поля таблицы через жесткий формат данных в ячейке. Теперь формат данных в ячейке пред-

ставляется в виде маски поля, где можно задавать любую гибридную форму данных в ячейке, так же появилась возможность вставлять обязательные символы в строку. Теперь разрешены следующие форматы для описаний данных:

! – Наличие символа «!» означает, что в строке все недостающие символы предваряются пробелами, а отсутствие его – пробелы размещаются в конце;

> - Символ «>» означает, что все последующие символы должны быть в верхнем регистре до конца поля, либо до символа «<>»;

< - Символ «<» означает, что все последующие символы должны быть в нижнем регистре до конца поля, либо до символа «>>»;

<> - Символы «<>» обозначают, что анализ регистра не производится;

\ - Символ «\» указывает на обязательный символ идущий за ним;

L – Указывает, что в данной позиции должна быть обязательно буква;

l – Указывает, что в данной позиции должна быть буква, или ничего;

A – Указывает, что в данной позиции должна быть обязательно либо буква, либо цифра;

a – Указывает, что в данной позиции должна быть либо буква, либо цифра, либо ничего;

C – Указывает, что в данной позиции должен быть обязательно любой символ;

c – Указывает, что в данной позиции должен быть любой символ, или ничего;

0 – Указывает, что в данной позиции должна быть обязательно цифра;

9 – Указывает, что в данной позиции должна быть цифра, или ничего;

- Указывает, что в данной позиции может быть цифра, знак «+», или знак «-», или ничего;

: - Используются для разделения часов, минут и секунд;

/ - Используется для разделения месяцев, дней и годов в датах;

_ - Символ «_» означает автоматическую вставку пробела в текст;

S(N) – Означает, что символ S будет повторен N-раз.

Например, маска в виде: Q.MASK="!(999\) 000-00-00;1;_"; будет в поле таблицы представлен следующий шаблон для ввода: “(____) ____-__-__”, что будет удобно для записи, например, номера телефона: “(812) 555-12-12”.

- **Наименование поля**

Наименование поля - это какое-либо предметное (технологическое) понятие. Например: “Диаметр”, “Наименование операции” и т.п. Не следует путать с именем поля, отвечающего за представление данных в ВСПТД.

- **Другие графы**

При формировании любой словарь может быть снабжен дополнительными графами, которые призваны выполнять дополнительные функции в про-

ектируемой системе. Например, для организации работы управляющего модуля организуется графа, содержащая номер/код функции, обеспечивающей методы работы с данным полем.

5.2 Использование словаря модулями комплекса

Словарь является первичным образованием, которое необходимо для функционирования комплекса.

- **Словарь и табличная база данных**

Аппарат работы с базами данных взаимодействует со словарем при контроле имен данных и их форматов, используемых в качестве заголовков колонок таблиц. При работе с табличной базой можно отключить этот контроль, но лучше, чтобы все имена были объявлены в словаре ради соблюдения дисциплины имен.

- **Словарь и база данных переходов**

При ведении базы данных переходов все именованные данные, участвующие в текстах переходов должны быть обязательно объявлены в словаре. В противном случае модуль ввода переменной информации в текст перехода работать не будет.

- **Словарь и модуль расчетных задач**

Модуль расчетных задач использует словарь при составлении анкеты запроса данных по составу формулы.

- **Словарь и модуль формирования анкет**

Модуль формирования анкетного запроса данных использует при своей работе словарь, поэтому все данные, объявленные для запроса в анкеты, должны быть обязательно объявлены и в словаре.

- **Словарь и информационно-поисковые задачи**

Информационно-поисковые задачи используют словарь при составлении гипотезы на поиск и при ограничении таблицы через поисковый образ. Таким образом, данные, которые участвуют при работе информационно-поисковых задач, должны быть обязательно указаны в словаре.

6. Языки программирования и базы данных

6.1. Система управления записями

Технологическая подготовка производства (ТПП) включает в себя разработку различной технологической и плановой документации, проектирование и изготовление технологической оснастки и т.д. Использование ЭВМ обуславливает разработку и широкое внедрение новых методов автоматизированной подготовки технологических процессов, что является важным резервом снижения себестоимости изготовления деталей, сокращения сроков технологической подготовки производства, повышения качества технологии.

При создании САПР ТП для предприятий, сталкиваются с некоторыми проблемами, такими как: адаптация, традиции, особенность производства для каждого конкретного предприятия. При этом создается множество различных программ, которые обеспечивают решения узкого круга задач, в частности, информационно-поисковые системы (ИПС).

В настоящее время рынок программных продуктов располагает достаточно большим набором программных средств для разработки информационно-поисковых систем (ИПС). Широкое распространение имеют такие программы как FoxPro, Dbase III, Oracle, Informix, MS SQL и др. В них используются реляционные базы данных, которые задаются в виде двухмерных таблиц. Требования, которые предъявляются к записям при работе с ВСПТД, выражаются, прежде всего, в наличии возможности доступа к записям переменной длины с заданным ключом. Таким требованиям отвечает система управления записями (СУЗ) BTRIEVE. Создателем этой системы является фирма NOVELL, а развитием в настоящее время занимается фирма Pervasive Software. В частности, ею разрабатываются новые версии BTRIEVE теперь уже с названием Pervasive SQL. Из названия следует, что данная версия СУЗ поддерживает стандартный язык запросов SQL.

6.2. Интерфейс с языками программирования

СУЗ Btrieve⁷ предлагает готовую систему управления записями для прикладных программ так, что Вам не придется писать свои собственные программы. Доступны интерфейсы со многими популярными языками, включая следующие.

QuickBasic фирмы Microsoft, IBM BASIC и компилятор BASIC

⁷ СУЗ Btrieve поставлялась фирмой Novell в составе сетевой операционной системы NetWare. Имеется опыт программирования ВСПТД с применением современных реляционных СУБД. В таких случаях виртуальная строка представляется в реляционной таблице как поле переменной длины. Остальные поля таблицы отвечают за первичный ключ. Вопросы доступа к виртуальным строкам решаются средствами применяемой СУБД. СУЗ Btrieve здесь представлена как исторически первое инструментальное средство для применения системы ВСПТД.

Pascal фирмы IBM (или Microsoft), Turbo Pascal, Turbo BASIC и BORLAND C, BUILDER C++, Cobol фирмы Microsoft, C фирмы Microsoft, Lattice C.

В результате усовершенствования технических приемов и структур в записях ВСПТД можно игнорировать физические структуры файлов и сконцентрироваться на их логической структуре.

6.3. Основные возможности системы управления записями BTRIEVE

- Многоключевой доступ к записям;
- Реляционный доступ к файлам;
- Организация записей переменной длины (до 64К) ;
- Автоматическая поддержка всех ключей;
- Поддержка дублируемых, модифицируемых, сегментных, нулевых и убывающих ключей;
- Поддержка при удалении и дополнении дополнительных индексов;
- Поддержка 13 типов расширенных ключей;
- Разделяемые буфера;
- Кэш буфера ввода/вывода;
- Неограниченные файловые размеры (до 4 миллиардов байтов);
- Возможности сетевого доступа к файлам.

Следует отметить, что в системе программирования ВСПТД не применяются в явном виде функции BTRIEVE. Пакет функций разработан таким образом, чтобы минимизировать затраты на перепрограммирование при возможном переходе на другой метод доступа. В идеальном случае системы проектирования должны иметь готовность к работе с уже используемыми на предприятиях базами данных с минимальными издержками по адаптации.

Одна из основных задач при разработке ЭС САПР ТП - реализация дружественного интерфейса с технологом. Эффективность решения этой задачи во многом определяется применяемыми инструментальными средствами, и прежде всего языками программирования, к которым предъявляются следующие требования:

- обеспечение преемственности при развитии системы;
- обеспечение жизненного цикла программ;
- обеспечение высокой эффективности инструментальных средств.

Требование по обеспечению преемственности определяет необходимость использования языка программирования C++, так как на нем реализована значительная часть алгоритмов, исходные тексты которых находятся в распоряжении разработчиков, что позволяет минимизировать их адаптацию к новым условиям и соответствует принципу эволюционного развития ЭС САПР ТП.

С другой стороны, новые предметные модули целесообразно развивать с помощью более мощных языков. В качестве такого языка выбран Си, который в наибольшей степени удовлетворяет задачам САПР. Кроме того, использование его широким кругом разработчиков, позволяет обеспечить жизненный цикл программ, т.к. поддержание программ в рабочем состоянии часто осуществляется не автором, а кем-то другим. Дальнейшее развитие системы ВСПТД ориентировано на языки программирования C++ Builder 10.0, PYTHON. В настоящее время завершены работы по представлению ВСПТД (триплексных строк) в XML-формате, что существенно расширяет возможности системы при взаимодействии с внешней средой. Идея представления любого объекта, участвующего в проектировании, в символьной форме получило новое развитие применительно к Web-технологии представления ВСПТД. Реализованы программные средства конвертирования технологических данных ВСПТД в web-ориентированную систему управления базами данных (СУБД) [16]. Реализован web-модуль на базе web-сервера Apache 1.3.27 и СУБД MySQL с применением серверных скриптов PHP. При разработке системы были использованы языки программирования Borland C++ 5, C++ Builder 6.0, PHP 4.0..

Выводы. Единство языка программирования, методов доступа к базам данных и других инструментальных средств является одним из важнейших условий развития больших систем. Однако, многообразие внешнего мира заставляет решать задачи в разнообразных языковых средах. В такой ситуации возможность взаимодействия с системой через единый интерфейсный модуль, в частности, установления связи между независимо построенными программными модулями путем использования одинаковых имен (см п.1.1), является важным условием для успешного решения поставленной задачи.

Исследования в области БД, позволили создать интерфейсы доступа к другим БД, не меняя методы и свойства программного класса взаимодействия с базой данных.

К примеру, был создан модуль доступа к БД MySQL на уровне прямых интерфейсов с базой данных. Все изменения касаются только внутренней реализации модуля доступа к БД, никак не затаргивая элементы уровня ВСПТД. Сами таблицы в БД хранятся как записи переменной длины.

Организован подход, ориентированный на реляционные БД.

Так реализован конвертер из таблиц в формате ВСПТД в обычные реляционные таблицы других БД. На данный момент опробован конвертер в MySQL. Конвертирование в другие базы данных возможно с применением универсального подхода – MS ADO, позволяющим работать, а следовательно и конвертировать таблицы разных баз данных. Здесь используются такие сервисы как ODBC и OLE Provider.

Система программирования ВСПТД уже нашла свое применение в ряде проектов на кафедре технологии приборостроения.

ЛИТЕРАТУРА

1. Яблочников Е.И., Маслов Ю.В. Автоматизация в приборостроении [Текст]: учеб. пособие // СПб. : ИТМО, 2003.
2. Ступаченко А.А. САПР технологических операций [Текст]. - Л.: Машиностроение, 1988. 234с.
3. Тамм Б.Г., Кюттнер Р.А. Реализация специализированных инструментальных систем программирования для САПР ТП [Текст] // сб. Автоматизированное проектирование и производство в машиностроении. / Под общ. ред. д-ра техн. наук Ю.М. Соломенцева и д-ра техн. наук В.Г. Митрофанова, - М.: Машиностроение, 1986. с.230-244
4. Заде Л. Понятие лингвистической переменной и его применение к понятию приближенных решений [Текст]. - М.: Мир, 1976.
5. Филиппов А.Н. Автоматизированная система проектирования технологических процессов с использованием подхода экспертных систем [Текст]// Актуальные проблемы современного программирования Сборник ЛИИАН, - Л.: 1989. с.113-122
6. Бруевич Н.Г., Боброва И.В., Челищев Б.Е. Математические основы автоматизации проектирования процессов механической обработки деталей [Текст] // Изв. АН СССР. Техническая кибернетика. 1978. N % 1. с.134-148
7. Томпсон Б., Томпсон У., Анатомия экспертной системы. [Текст] / /Реальность и прогнозы искусственного интеллекта. - М.: Мир, 1987.- 206с
8. Инмон У., Фридман Л. Методология экспертной оценки проектных решений для систем с базами данных. [Текст] пер.с англ. - М.: Финансы и статистика, 1986. с.183
9. Леонг-Хонг Б., Плагман Б. Системы словарей-справочников данных. Администрирование, реализация, использование [Текст]. -М.: Финансы и статистика, 1986. с.311
10. Минский М. Фреймы для представления знаний [Текст]. - М.: - Энергия, 1979. с.49-51
11. Российский коммуникативный формат представления библиографических записей в машиночитаемой форме, <http://www.nlr.ru/rba/rusmarc/content.htm>.
12. Цветкова И.Б. и др. Российский коммуникативный формат представления библиографических записей в машиночитаемой форме [Текст]. — СПб: Изд-во РНБ, 1998г., 356с.
13. Американский коммуникативный формат представления библиографических записей. СПб: Изд-во РНБ, 1990г., 673с.
14. Цветков В.Д. Системно-структурное моделирование и автоматизация проектирования технологических процессов [Текст]. — Минск: Наука и техника, 1979. с.81-85
15. Филиппов А. Н. «Разработка и исследование методов экспертных систем в САПР ТП механической обработки» [Текст]. - Диссертация на соискание ученой степени кандидата технических наук. Л., 1991 г., 160 с.

16. Сисюков А. Н. Разработка и применение специализированных экспертных систем для САПР технологических процессов механической обработки заготовок – Диссертация на соискание ученой степени кандидата технических наук. СПб, 2007. 150 с.
17. Афанасьев М. Я. Разработка и исследование многоагентной системы для решения задач технологической подготовки производства — Диссертация на соискание ученой степени кандидата технических наук. СПб., 2012., 153с.
18. «Техком-SERVICE Plus», V 3.5. — Описание применения. СПб: НТЦ Техком, 1993. 205 с.
19. Филиппов А.Н. Разработка критериев выбора окончательного решения при поиске инструмента в экспертной системе технологического назначения // XIV заочная научная конференции Research Journal of International // Международный Научно-Исследовательский Журнал / под ред. Пачурина Г.В. д-р техн. наук, проф., Федоровой Е.А. д-р техн. наук, проф., Герасимовой Л.Г., д-р техн. наук, Курасова В.С., д-р техн. наук. - Екатеринбург: ООО «Евро-принт», 2013. - № №4 (11). - Часть 1. - С. 121-124. - ISSN 2303-9868. - URL: <http://research-journal.org/wp-content/uploads/2011/10/4-11-1.pdf>.
20. Голубев Я.Ф. Проектирование модуля заполнения операционных карт на основе стандарта OpenDocument — Магистерская диссертация, СПб., 2011,

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ТЕХНОЛОГИИ ПРИБОРОСТРОЕНИЯ

Кафедра технологии приборостроения относится к числу ведущих кафедр института со дня его основания в 1931 году. Тогда она называлась кафедрой механической технологии и возглавлялась известным ученым в области разработки инструмента профессором Александром Павловичем Знаменским. Позже она была переименована в кафедру технологии приборостроения.

За время своего существования кафедра выпустила из стен института более тысячи квалифицированных инженеров, более сотни кандидатов и докторов наук. В разные годы ее возглавляли известные ученые и педагоги профессора Николай Павлович Соболев и Сергей Петрович Митрофанов.

Кафедра имеет выдающиеся научные достижения. Заслуженным деятелем науки и техники РСФСР, профессором С.П. Митрофановым были разработаны научные основы группового производства, за что он был удостоен Ленинской премии СССР. Методы группового производства с успехом применяются в промышленности и постоянно развиваются его учениками. Заслуженным деятелем науки и техники РСФСР, Заслуженным изобретателем СССР Юрием Григорьевичем Шнейдером разработаны метод и инструментарий нанесения регулярного микрорельефа на функциональной поверхности.

В настоящее время кафедра осуществляет выпуск бакалавров и магистров по направлениям 200100 "Приборостроение" и 230100 "Информатика и вычислительная техника".

Филиппов А. Н.

**ВИРТУАЛЬНОЕ СТРОКОВОЕ ПРОСТРАНСТВО
ТЕХНОЛОГИЧЕСКИХ ДАННЫХ И ЗНАНИЙ
Методы представления данных**

Учебное пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Тираж 50 экз.

Отпечатано на ризографе

**Редакционно-издательский отдел НИУ
Университета ИТМО**
197101, Санкт-Петербург, Кронверкский пр., 49