

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕДАКТОРА ПРАВИЛ ДЛЯ
ОБОЛОЧКИ-ИНТЕРПРЕТАТОРА БАЗ ЗНАНИЙ ESWIN**

Выпускная квалификационная работа
обучающегося по направлению подготовки
09.03.03 «Прикладная информатика»
заочной формы обучения, группы 07001251
Черникова Максима Викторовича

Научный руководитель:
к.т.н. Путивцева Н.П.

В наши дни широко используются экспертные системы, помогающие при выборе решения для конкретной задачи.

При разработке экспертных систем с использованием оболочки ESWin, эксперт, а также инженер по знаниям сталкиваются с множеством проблем. К ним можно отнести выбор классов, признаков, но, пожалуй, самой трудной, а точнее трудоемкой, частью работы является составление правил и установление наибольшего соответствия тому или иному классу.

На данный момент не существует программных средств, способных в совершенстве решить существующую проблему. Разрабатываемая технология генераций правил для ESWin помогает решить эти проблемы, с помощью автоматического генерирования фреймов по заданным параметрам и признакам.

Объект исследования: информационное и программное сопровождение оболочки ESWin.

Предмет исследования: процесс разработки генератора базы правил.

Целью работы является повышение эффективности работы разработчиков экспертных систем (пользователей), за счет автоматизации процесса создания правил ЭС.

Для достижения цели необходимо решить следующие задачи:

- изучение форм представления знаний в ESWin;
- разработка структуры данных программы генерирующей правила;
- создание программного средства, автоматически генерирующего правила (фреймы) для дальнейшего их использования в системе с оболочкой ESWin;
- апробация на примере лабораторной работы в курсе предмета интеллектуальные информационные системы. И внедрение в учебный процесс;
- апробация программного средства на предприятии (на примере ООО «РЖД»).

Задача создания программного средства будет решаться с помощью продукта Borland C++ Builder . На данный выбор в основном повлияло то, что университет располагает лицензионной версией данного продукта.

Пояснительная записка состоит из введения, 3 разделов, заключения, списка используемой литературы и приложений.

Во введении рассматривается актуальность выбранной темы, выбираются предмет и объект исследования, ставится цель и формулируются задачи, приводится структура ВКР.

В первом разделе рассматриваются программные и аппаратные средства, широко распространенные в области интеллектуальных систем. Приводятся характеристики гибридных экспертных систем. Рассматривается экспертная оболочка продукционно-фреймового типа ESWin. Анализируются ее недостатки.

Во втором разделе приведена поэтапная разработка программы подготовки баз знаний.

В третьем разделе приведены примеры работы программы подготовки баз знаний и рассмотрены результаты внедрения.

Пояснительная записка к ВКР написана на 60 страницах, содержит 30 рисунков, 2 таблиц и 3 приложения.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

Экспертные системы (ЭС) - это яркое и быстро прогрессирующее направление в области искусственного интеллекта (ИИ). Причиной повышенного интереса, который ЭС вызывают к себе на протяжении всего своего существования, является возможность их применения к решению задач из самых различных областей человеческой деятельности. Пожалуй, не найдется такой проблемной области, в которой не было бы создано ни одной ЭС или по крайней мере такие попытки не предпринимались бы [9].

ЭС - это набор программ или программное обеспечение, которое выполняет функции эксперта при решении какой-либо задачи в области его компетенции. ЭС, как и эксперт-человек, в процессе своей работы оперирует со знаниями. Знания о предметной области, необходимые для работы ЭС, определенным образом формализованы и представлены в памяти ЭВМ в виде базы знаний, которая может изменяться и дополняться в процессе развития системы.

ЭС выдают советы, проводят анализ, выполняют классификацию, дают консультации и ставят диагноз. Они ориентированы на решение задач, обычно требующих проведения экспертизы человеком-специалистом. В отличие от машинных программ, использующий процедурный анализ, ЭС решают задачи в узкой предметной области (конкретной области экспертизы) на основе дедуктивных рассуждений. Такие системы часто оказываются способными найти решение задач, которые неструктурированы и плохо определены. Они справляются с отсутствием структурированности путем привлечения эвристик, т. е. правил, взятых «с потолка», что может быть полезным в тех системах, когда недостаток необходимых знаний или времени исключает возможность проведения полного анализа [20].

1.1 Описание программных и аппаратных средств использованных при изучении ИИС

В учебном процессе предмета ИИС были рассмотрены некоторые экспертные оболочки.

1.1.1 Интегрированная инструментальная среда GURU

В инструментальной среде построения ЭС GURU, разработанной фирмой Micro Data Base Systems, Inc., методы экспертных систем сочетаются с такими средствами обработки данных, как составление электронных ведомостей, управление базой данных и деловой графикой, и таким образом формируется уникальная среда для поддержки принятия решений и разработки прикладных интеллектуальных систем.

Система GURU легка в употреблении для новичков и в то же время является достаточно эффективной и гибкой системой для профессионалов - разработчиков.

В обычных «интегрированных» программных продуктах или несколько отдельных программ помещены в операционную среду, или несколько, второстепенных компонентов вкладываются в главный компонент (как, например, программа обработки электронных ведомостей или текстовый редактор). Трудности, с которыми сталкиваются при таких стилях «интеграции», хорошо известны. Первая трудность заключается в том, что пользователь вынужден переходить назад и вперед по отдельным программам и передавать данные между ними. Метод вложений заставляет пользователя выполнить всю обработку в пределах главного компонента, и в результате получаются относительно слабые вторичные компоненты.

Метод Интеграции, используемый в системе GURU, основывается на принципе синергизма. При этом под «синергизмом» здесь понимается

следующее. В системе GURU все средства всегда доступны. Многочисленные компоненты можно соединять по желанию в пределах одной операции, а это характеризует систему как гибкую и удобную в использовании. Например:

- в посылке любого правила ЭС можно делать прямые ссылки на поля базы данных, на ячейки электронных ведомостей, на статические переменные, на программные переменные и массивы;

- вывод любого правила ЭС может включать в себя операции управления базой данных, запросы на языке SQL (языке структурированных запросов), операции обработки электронных ведомостей, генерацию статистических данных, дистанционную передачу данных, выполнение процедур, генерацию деловой графики;

- поскольку ЭС обосновывает задачу, она может брать консультации у других ЭС, выполнять процедурные модели, просматривать базы данных, составлять электронные ведомости или проводить статистический анализ;

- любую ячейку электронной ведомости можно определить в терминах поиска в базе данных или в терминах всей программы, или в терминах консультации с ЭС [14,15].

Взаимодействовать с системой можно любым из четырех различных способов: с помощью меню, на ограниченном естественном языке, в режиме команд или через специально разработанные интерфейсы. Каждый тип интерфейса системы GURU предназначен для удовлетворения потребностей и вкусов различных классов пользователей. Всеми четырьмя интерфейсами можно пользоваться во время одного и того же сеанса взаимодействия с системой GURU.

Как и в большинстве оболочек, в GURU используется продукционная модель представления знаний в виде совокупности «If-then» правил с обратной стратегией вывода в качестве основной имеется возможность моделирования нечетких и неточных рассуждений. Кроме посылок и заключения в правила можно включать команды, которые будут выполняться перед проверкой условия, а также пояснительный текст для генерации объяснений. Правила также

включают необязательные параметры цены и приоритета, позволяющие управлять процессом выбора из совокупности, готовых, к выполнению правил очередного. С каждым правилом можно также связать число, определяющее, сколько раз это правило может выполняться в процессе консультации.

Правила, относящиеся к решению некоторой общей задачи, образуют базу знаний, или набор правил. В этот набор кроме правил включаются две специальные процедуры: инициализация и завершение, которые должны выполняться до и после выполнения правил. В набор правил также включаются описания переменных, участвующих в правилах, содержащие спецификации типа, точности и т. п.

По умолчанию в GURU принята стратегия обратного вывода, однако, можно использовать чисто прямой вывод, а также комбинировать его с обратным в рамках одного набора правил. Как стратегиями вывода, так и целевыми переменными можно управлять динамически в процессе консультации.

GURU обеспечивает мощные средства управления обработкой факторов уверенности, отражающих степень неточности и нечеткости выраженных в правилах эвристических знаний. Для предоставления такой нечеткости в GURU с каждым значением переменной может быть связан числовой коэффициент от 0 до 100. Система предоставляет разработчику выбор более чем из 30 различных формул, позволяющих управлять обработкой факторов уверенности во время вывода.

Полезными являются такие дополнительные средства управления логическим выводом, как установка степени «точности» вывода значения для некоторой переменной, изменение принятого по умолчанию порядка просмотра правил.

Эффективность машины логического вывода во многом зависит от того, как она осуществляет поиск в наборе правил, когда ищет правила, которые можно выполнять. В отличие от традиционного программного обеспечения, использующего принципы искусственного интеллекта, система GURU предоставляет расширенные средства управления настройкой, в частности

поддерживает до 50 различных стратегий поиска. Эффективность также зависит от количества и состава правил в наборе правил. Поскольку система GURU предоставляет разнообразные возможности создания наборов правил, то можно значительно сократить количество правил, необходимых для охвата всех знаний и опыта в конкретной проблемной области. Это приводит к ускорению процесса получения логических выводов, а также к упрощению управления этими правилами.

Интегрированная система GURU пытается превратить потенциальные преимущества ЭС в реальность, облегчить пользователю процесс создания ЭС, сделать его прямым, эффективным и естественным [6].

1.1.2 Система 1stCLASS

Система-оболочка 1stCLASS относится к классу систем индуктивного типа, так как исходит из принципа построения дерева поиска решений по множеству примеров. От пользователя не требуется изучения команд внутреннего языка оболочки, ему предоставляется серия из шести графических экранов-меню. Используя эти экраны, можно создать базу знаний (БЗ) прикладной ЭС и запустить режим консультации [8].

Имеется два способа формирования БЗ:

- основанный на примерах – задаются примеры, которые показывают, как эксперт принимает решение, затем система автоматически превращает эти примеры в сжатое правило (в виде дерева решений), формирующее базис для БЗ;
- основанный на правилах – строится графическая диаграмма дерева решений путем выбора факторов, значений и результатов (рекомендаций) из заранее составленного меню.

Используя серию меняющихся экранов, пользователь может быстро определять элементы БЗ, вводить примеры, строить правила и проверять

полученные БЗ. Затем можно из нескольких (до 1000) модульных БЗ построить одну комплексную БЗ.

Сформировав и отладив БЗ (т.е. проверив ее на полноту и выбрав подходящий метод обработки фактора неопределенности в виде коэффициентов правдоподобия), получаем прикладную консультирующую ЭС, которая рекомендует, как действовать в той или иной ситуации или решать некоторую проблему и способна заменить человека-эксперта в соответствующей проблемной области [16].

1stCLASS имеет также ряд других возможностей, которые позволяют решать достаточно эффективно более сложные задачи. В частности, система позволяет:

- взаимодействовать с электронными таблицами LOTUS 1-2-3 для обмена числовыми и логическими данными;
- посылать БЗ через средства телекоммуникации;
- автоматически генерировать журналы сеансов пользователя и структурированные доклады для дальнейшей обработки в файлах текстовых процессоров и баз данных;
- вызывать 1stCLASS из других программ и запускать внешние программы из 1stCLASS для ответов на вопросы ЭС и выполнения необходимых действий;
- взаимодействовать с аппаратурой, в том числе: считывать показания приборов, организовывать систему «автоматического библиотекаря» для подключения видеодисков и систем компакт-дисков, связываться с удаленными БД для получения информации, организовывать интерфейс со статическими программами;
- выполнять анализ «что если», используя различные методы генерации правил, различные наборы экспертных данных и различные ответы пользователя. Для каждого выданного системой совета можно графически

проследить логику поиска решения и обеспечить детальное объяснение для пользователя.

Система 1stCLASS получила широкое распространение в реализации прикладных ЭС - консультантов для ряда проблемных областей, например:

- медицинская диагностика;
- техническая диагностика;
- автоматические библиотеки;
- интерактивная подготовка и обучение;
- автоматизация программирования;
- разработка систем и др.

1.1.3 Язык искусственного интеллекта CLIPS

Название языка CLIPS – аббревиатура от C Language Integrated System. Язык был разработан в Центре космических исследований NASA (NASA's Johnson Space Center) в середине 1980-х годов и во многом сходен с языками, созданными на базе LISP и OPS5 [4,7]. Использование C в качестве языка реализации объясняется тем, что компилятор LISP не поддерживается частью распространенных платформ, а также сложностью интеграции LISP-кода в приложения, которые используют отличный от LISP язык программирования. Хотя в то время на рынке уже появились программные средства для задач искусственного интеллекта, разработанные на языке C, специалисты из NASA решили создать такой продукт самостоятельно. Разработанная ими система в настоящее время находится в открытом доступе (ее можно «скачать» по сети INTERNET), и отметим, что по своим возможностям она не уступает многим гораздо более дорогим коммерческим продуктам.

Первая версия представляла собой, по сути, интерпретатор порождающих правил. Процедурный язык и объектно-ориентированное расширение CLIPS –

Object-Oriented Language (COOL) были включены в этот программный продукт только в 1990-х годах. Существующая в настоящее время версия может эксплуатироваться на платформах UNIX, DOS, Windows и Macintosh.

CLIPS включает в себя язык представления порождающих правил и язык описания процедур.

Основными компонентами языка описания правил являются база фактов (fact base) и база правил (rule base). База фактов представляет исходное описание задачи. База правил содержит операторы, которые преобразуют состояния проблемы, приводя его к решению – целевому состоянию.

Машина логического вывода CLIPS сопоставляет факты из базы фактов и правила из базы правил и выясняет, какие из правил можно активизировать. Это выполняется циклически, причем каждый цикл (так называемый продукционный цикл или цикл распознавания действия) состоит из трех основных фаз:

- сопоставление фактов и правил;
- выбор правила, подлежащего активизации;
- выполнение действий, предписанных активным («зажженным») правилом [6].

1.2. Характеристики гибридных экспертных систем, основанных на продукционно-фреймовой модели представления знаний

Представление знаний – вопрос, возникающий в когнитологии (науке о мышлении), в информатике и в искусственном интеллекте. В когнитологии он связан с тем, как люди хранят и обрабатывают информацию. В информатике – основная цель – подбор представления конкретных и обобщенных знаний, сведений и фактов для накопления и осмысленной обработки информации в ЭВМ [10].

Существует несколько моделей представления знаний.

Продукционная модель, или модель, основанная на правилах, позволяет представить знания в виде предложений типа: Если (условие), то (действие).

Под, условием понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под действием - действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы) [2].

При использовании продукционной модели база знаний состоит из набора правил, Программа, управляющая перебором правил, называется машиной вывода. Чаще всего вывод бывает прямой (от данных к поиску цели) или обратный (от цели для ее подтверждения - к данным). Данные - это исходные факты, на основании которых запускается машина вывода - программа, перебирающая правила из базы.

Семантические сети

Термин семантическая означает смысловая, а сама семантика - это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т.е. наука, определяющая смысл знаков.

Семантическая сеть- это ориентированный граф, вершины которого - понятия, а дуги - отношения между ними.

Фрейм (англ. frame - каркас или рамка) предложен М.Минским в 70-е гг. Как структура знаний для восприятия пространственных сцен. Эта модель, как и семантическая сеть, имеет глубокое психологическое обоснование [18].

Под фреймом понимается абстрактный образ или ситуация. В психологии и философии известно понятие абстрактного образа. Например, слово «комната» вызывает у слушающих образ комнаты: «жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью, площадью 6-20 м²». Из этого описания ничего нельзя убрать (например, убрав окна мы получим уже чулан, а не комнату), но в нем есть «дырки», или «слоты», - это незаполненные значения

некоторых атрибутов - количество окон, цвет стен, высота потолка, покрытие пола и т.д. [5].

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

1.3. Анализ ESWin как типичного представителя продукционно-фреймового класса ЭС

ESWin v.1.0 - программная оболочка для работы с продукционно-фреймовыми экспертными системами с возможностью использования лингвистических переменных. Описываемая программная оболочка предназначена для решения задач методом обратного логического вывода на основе интерпретации правил-продукций с использованием фреймов как структур данных, включающих в себя в частности лингвистические переменные.

БАЗА ЗНАНИЙ

База знаний состоит из набора фреймов и правил-продукций. Формат внешнего представления базы знаний (в текстовом файле) выглядит следующим образом:

```
TITLE = <название экспертной системы>
```

```
COMPANY = <название предприятия>
```

```
FRAME // фрейм
```

```
<описание фрейма>
```

```
ENDF
```

```
.
```

```
.
```

```
FRAME // фрейм
```

```
<описание фрейма>
```

```

ENDF
RULE                // правило-продукция
  <описание условий правила>
DO
  <описание заключений правила>
ENDR
.
.
RULE                // правило-продукция
  <описание условий правила>
DO
  <описание заключений правила>
ENDR

```

База знаний состоит из двух частей: постоянной и переменной. Переменная часть базы знаний называется базой данных и состоит из фактов, полученных в результате логического вывода. Факты в базе данных не являются постоянными. Их количество и значение зависит от процесса и результатов логического вывода.

До начала работы с экспертной оболочкой база знаний находится в текстовом файле. В файле с расширением *.klb (KnowLedge Base) хранятся фреймы и правила-продукции (база знаний). При начале работы с программной оболочкой наличие данного файла обязательно. Этот файл создается пользователем с помощью специального редактора или вручную. В файле с расширением *.dtb (DaTa Base) хранятся факты, полученные в процессе логического вывода (база данных). При начале работы с программной оболочкой наличие данного файла необязательно. Файл с базой данных создается программной оболочкой в процессе логического вывода. Первые части имен этих двух файлов совпадают.

При работе с программной оболочкой (после загрузки в оперативную

память баз) фреймы и правила-продукции, находившиеся в файле с расширением *.klb, остаются неизменными. Факты, находившиеся в файле с расширением *.dtb, могут изменяться в процессе логического вывода (появляться, удаляться или менять свое значение в результате срабатывания правил-продукций или диалога с пользователем).

Фреймы используются в базе знаний для описания объектов, событий, ситуаций, прочих понятий и взаимосвязей между ними. Фрейм - это структура данных, состоящая из слотов (полей).

Фрейм может принадлежать к одному из трех типов фреймов: фрейм-класс (тип описывается зарезервированным словом «класс»), фрейм-шаблон (тип описывается зарезервированным словом «шаблон»), фрейм-экземпляр (тип описывается зарезервированным словом «экземпляр»). В базе знаний содержатся фреймы-классы и фреймы-шаблоны. При создании базы знаний тип фрейма-класса можно не описывать, этот тип фрейма понимается по умолчанию. Явно следует описывать только тип фрейма-шаблона.

В базе данных хранятся только фреймы-экземпляры. Так как для хранения фреймов-экземпляров используется специальный файл с расширением *.dtb, явно их тип в этом файле также можно не описывать. (Описание типов фреймов-классов и фреймов-экземпляров используется по преимуществу во внутреннем представлении базы знаний и базы данных).

**ИМЯ ФРЕЙМА, ФРЕЙМА-РОДИТЕЛЯ, ФРЕЙМА-ВЛАДЕЛЬЦА,
СЛОТА**

Имена фрейма, фрейма-родителя, фрейма-владельца, слота - это последовательность символов (русские и/или латинские буквы, цифры, пробелы, знаки подчеркивания).

Тип слота может принадлежать к одному из трех типов: символьный, численный, лингвистический. Описание типа слота определяет тип возможных значений слота. Обязательным является описание типов слотов численного (описывается зарезервированным словом «численный») и лингвистического. Слот без описания типа понимается как символьный по умолчанию.

Вопрос слота - любая последовательность символов. Вопрос слота не является обязательным. В таком случае, в процессе логического вывода, при возникновении необходимости задать вопрос пользователю, касающийся определения значения данного слота, пользователю будет предложена формулировка: «Выберите значение» или «Введите значение».

Значение слота - любая последовательность символов. Значения слота разделяются точками с запятыми. Список значений слота не обязателен, он может отсутствовать, в таком случае пустые круглые скобки необязательны. Во фрейме-экземпляре у каждого слота может быть только единственное значение, во фреймах-классах и фреймах-шаблонах число значений слотов не ограничено.

С помощью специальных слотов `parent` и `owner` фреймы могут объединяться в деревья. Кроме того, между фреймами могут существовать и произвольные связи через обычные слоты (значением слота в этом случае является имя другого фрейма).

1.4. Анализ недостатков ESWin

Программное средство ESWin является одним из наиболее доступных представителей продукционно-фреймового класса ЭС. Но как и другие представители данного класса, ESWin обладает значительным недостатком-предварительное написание фреймов, установленной формы. Это является одним из наиболее трудоемких этапов разработки ЭС. А точнее одним из занимающих наибольший промежуток времени. На рисунке 1.1. показано, каким образом, в основном, разрабатываются экспертные системы.

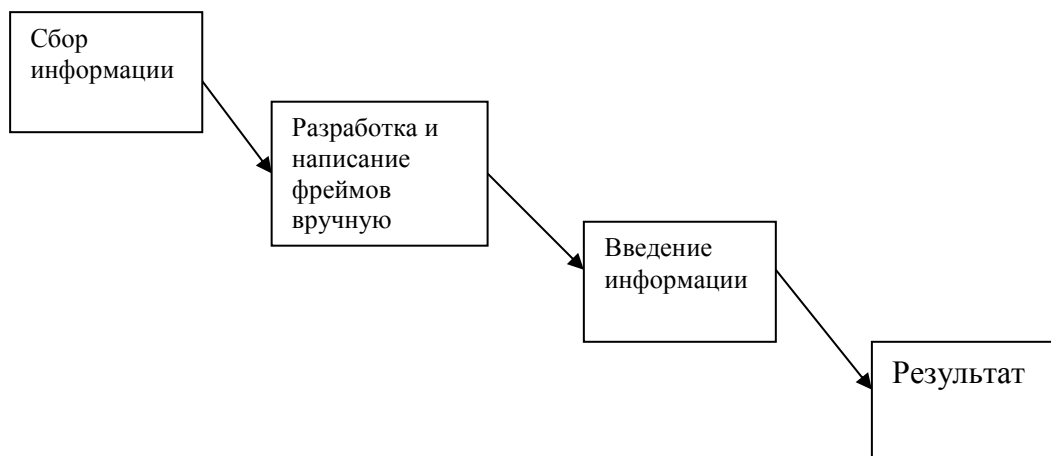


Рисунок 1.1 - Схема разработки

Из рисунка видно, что вторым этапом разработки системы является разработка правил, осуществляемая вручную, т.е. занимающая огромное количество времени и сил.

1.5 Постановка задачи

1.5.1 Цель и задачи

Целью работы является повышение эффективности работы разработчиков экспертных систем (пользователей), за счет автоматизации процесса создания правил ЭС.

Для достижения цели необходимо решить следующие задачи:

- изучение форм представления знаний в ESWin;
- разработка структуры данных программы генерирующей правила;
- создание программного средства, автоматически генерирующего правила (фреймы) для дальнейшего их использования в системе с оболочкой ESWin;

- апробация на примере лабораторной работы в курсе предмета интеллектуальные информационные системы;
- апробация и внедрение программного средства на предприятие. На примере ООО «РЖД».

1.5.2 Требования к функциональным характеристикам, техническому и информационному обеспечению

Программный продукт изначально разрабатывается для работы опытных пользователей и программистов. Так как генерация правил производится однократно для решения одной конкретной задачи, а затем сгенерированные правила используются многократно. Генератор правил ESWin будет иметь статический вид. Все преобразования будут производиться непосредственно в коде. В этом случае не загромождается интерфейс программного средства. При запуске программного средства нам представляется окно с полем, в котором выводится количество правил, полосой загрузки и двумя кнопками:

- «Сгенерировать»- производится запуск программы;
- «Результат»- открывается файл со сгенерированными правилами.

Программа должна генерировать правила путем полного их перебора по заданным параметрам.

Техническое обеспечение - это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Выделим основные критерии выбора технических устройств:

- стоимость устройства;
- скорость работы.

Разработка и тестирование экспертной системы будет проводится на компьютере со следующими характеристиками:

- процессор: Intel® Core 2 Duo T7300 (2.1 GHz, 800 MHz 4MBL2 Cache, 4Гб DDR2);
- жесткий диск: Hitachi 160 Гб.

Разработанная система может эксплуатироваться на любом типе современных компьютеров.

Информационное обеспечение - это создание информационных условий функционирования системы, обеспечение необходимой информацией, включение в систему средств поиска, получения, хранения, накопления, передачи, обработки информации.

Информационным обеспечением будет являться эксперт, который будет подготавливать программу к генерации правил.

1.5.3 Выбор средств создания «Генератора правил ESWin»

Для реализации поставленной перед нами задачи, было выбрано мощное и эффективное ПС: Borland C++ Builder.

C++Builder - программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке C++.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов, разработанных в Delphi, можно использовать и в C++ Builder без модификации, но, к сожалению, обратное утверждение не верно.

C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает

программирование благодаря встроенному WYSIWYG - редактору интерфейса и пр.

C++ Builder первоначально создавалась только для платформы Microsoft Windows. Поздние версии, содержащие Кроссплатформенную компонентную библиотеку Borland, поддерживают и Windows и Linux.

В 2003 Borland выпустила C++ BuilderX (CBX), написанный при помощи той же инфраструктуры, что и JBuilder, который при этом был мало похож на C++ Builder или Delphi. Этот продукт предназначался для разработки больших программ для крупных предприятий, но коммерческого успеха не достиг. В конце 2004 года Borland объявила, что продолжит развитие классического C++ Builder и объединит его со средой разработки Delphi, прекратив, таким образом, разработку C++ BuilderX.

Спустя примерно год после этого объявления, Borland выпустила Borland Developer Studio 2006, который включал в себя Borland C++ Builder 2006, предлагавший улучшенное управление конфигурацией и отладкой. Borland Developer Studio 2006 - единственный полноценный комплект, содержащий Delphi, C++Builder и C#Builder.

В 2007 CodeGear выпустила C++ Builder 2007, в котором реализовала полную поддержку API Microsoft Windows Vista, увеличила полноту соответствия стандарту ANSI C++, увеличила скорость компиляции и сборки до 500 %, включила поддержку MSBuild, архитектур баз данных DBX4 и «VCL для Web», поддерживающий AJAX. Поддержка API Microsoft Windows Vista включила в себя приложения, изначально оформленные в стиле Vista, и естественную поддержку VCL для Aero и Vista Desktop. CodeGear RAD Studio 2007 содержит C++ Builder 2007 и Delphi. Также в 2007 CodeGear «воскресила» марку «Turbo» и выпустила две «Turbo» версии C++ Builder: Turbo C++ Professional и Turbo C++ Explorer (бесплатный), основанных на Borland C++ Builder 2006.

2 РАЗРАБОТКА ПРОГРАММЫ ПОДГОТОВКИ БАЗ ЗНАНИЙ ДЛЯ ЭКСПЕРТНЫХ СИСТЕМ ESWin

2.1 Разработка структуры данных

В вычислительной технике структура данных - это программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных. Для добавления, поиска, изменения и удаления, данных структура данных предоставляет некоторый набор функций, составляющих интерфейс структуры данных. Структура данных часто является реализацией какого-либо абстрактного типа данных [2].

При разработке программного обеспечения большую роль играет проектирование хранилища данных, и представление всех данных в виде множества связанных структур данных. Хорошо спроектированное хранилище данных оптимизирует использование ресурсов (таких как время выполнения операций, используемый объём оперативной памяти, число обращений к дисковым накопителям), требуемых для выполнения наиболее критичных операций.

Система поддержки принятия решений должна состоять из двух основных частей генератора правил и оболочки экспертной системы (рисунок 2.1).

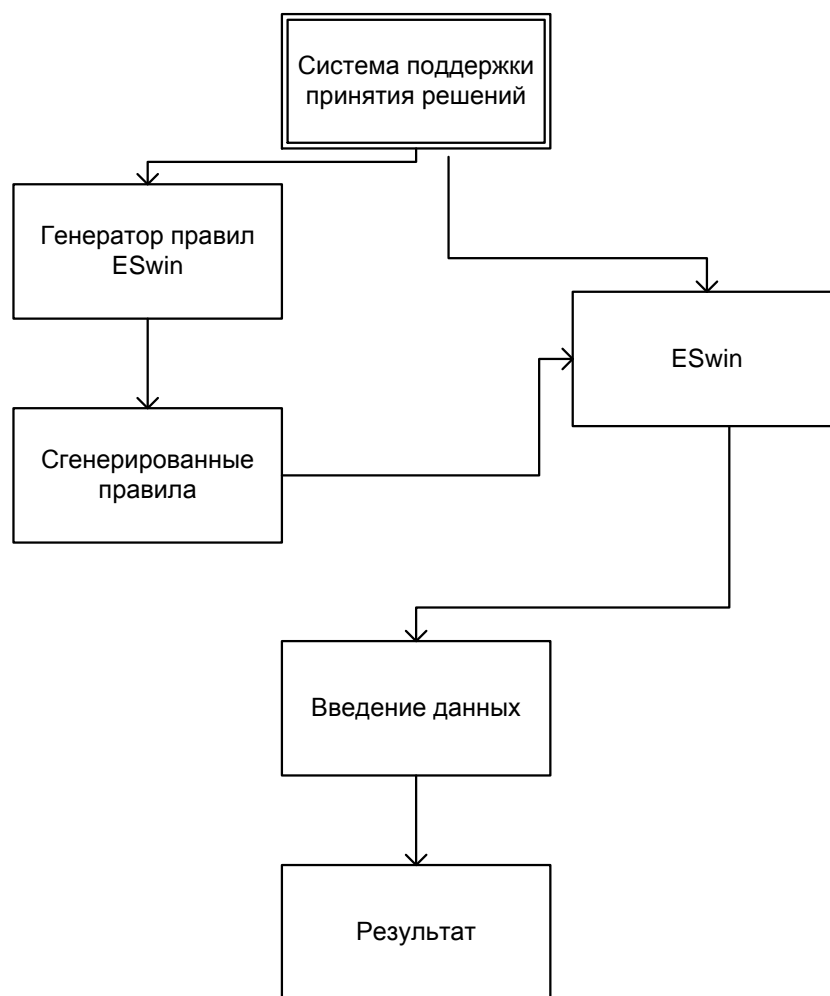


Рисунок 2.1 - Схема работы системы поддержки принятия решений

Как было сказано выше, программный продукт генератор правил ESWin обладает открытым кодом. В программном коде существует два хранилища данных, представленные матрицами. Первая матрица, размерностью $A \times B$, где A - наибольшее число параметров одного признака, а B - количество признаков, заполняется параметрами из которых в последствии формируются правила, методом перебора.

Вторая матрица, размерностью $(B+1) \times C$, где C - количество классов, заполняется классами с соответствующими им признаками.

Далее рассмотрены заполненные матрицы на примере ЭС по выбору модема. На рисунке 2.2 приведен пример заполненной матрицы параметров.

```

AnsiString mass[8][4]={
    {"ADSL", "VDSL", "SDSL", "SHDSL"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
};

```

Рисунок 2.2 - Матрица параметров

На рисунке 2.3 представлен пример заполненной матрицы классов.

```

int kol=15;
AnsiString modem[15][9]={
    {"Dynamix UM-S", "SHDSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Да"},
    {"NSGate qBRIDGE-105", "SDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
    {"ZyxeI P-660RU EE", "ADSL", "Да", "Нет", "Да", "Нет", "Да", "Нет", "Да"},
    {"NetGear DG834", "ADSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Да"},
    {"TRENDnet Tw100-BRM504", "ADSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Нет"},
    {"Planet GRT-101", "SHDSL", "Да", "Нет", "Да", "Нет", "Да", "Нет", "Да"},
    {"D-link DSL-2520U", "ADSL", "Да", "Нет", "Да", "Да", "Да", "Нет", "Да"},
    {"ASUS AM604", "ADSL", "Да", "Да", "Да", "Нет", "Да", "Нет", "Да"},
    {"Planet GRT-501", "SHDSL", "Да", "Нет", "Да", "Да", "Да", "Нет", "Да"},
    {"ADC PairGain Megabit Modem 300S", "SDSL", "Нет", "Нет", "Нет", "Нет", "Да", "Нет", "Нет"},
    {"Aviv SDSL-165", "SDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
    {"Planet VC-200M", "VDSL", "Да", "Да", "Нет", "Нет", "Да", "Нет", "Да"},
    {"ZyxeI Prestige 841 EE", "VDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
    {"D-link DEV-301M", "VDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Да", "Нет"},
    {"Planet VC-102S", "VDSL", "Да", "Нет", "Нет", "Нет", "Нет", "Да", "Да"},
};

```

Рисунок 2.3 - Матрица классов

2.2 Разработка алгоритмов генерации баз знаний

Генерация правил в программе производится путем полного перебора всех возможных вариантов, следующим образом. Из матрицы параметров выбирается сочетание параметров таким образом, что сочетания не повторяются.

Затем после формирования параметров и получения класса он сопоставляется с данными внесенными в матрицу классов. И затем подбирается наиболее подходящий. Следующим шагом создается непосредственно правило в заданном виде, со всеми подобранными параметрами.

Далее приведен алгоритм генерации правил, рассмотренный на примере выбора модема.

Первым шагом из матрицы параметров, по каждому из пунктов, выбирается первое значение (рисунок 2.4).

```

AnsiString mass[8][4]={
    {"ADSL", "VDSL", "SDSL", "SHDSL"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
};

```

Рисунок 2.4 - Первый выбор значений из матрицы

Далее выбранное сочетание значений матрицы: «ADSL, Да, Да, Да, Да, Да, Да, Да», сопоставляется с матрицей приведенной на рисунке 2.5. Как видно, выбранным параметрам соответствует четвертое значение NetGear DG834.

```

int kol=15;
AnsiString modem[15][9]={
    {"dynamix UM-S", "SHDSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Да"},
    {"NSGate qBRIDGE-105", "SDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
    {"Zyxel P-660RU EE", "ADSL", "Да", "Нет", "Да", "Нет", "Да", "Нет", "Да"},
    {"NetGear DG834", "ADSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Да"},
    {"TRENDnet TW100-BRM504", "ADSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Нет"},
    {"Planet GRT-101", "SHDSL", "Да", "Нет", "Да", "Нет", "Да", "Нет", "Да"},
    {"D-link DSL-2520U", "ADSL", "Да", "Нет", "Да", "Да", "Да", "Нет", "Да"},
    {"ASUS AM604", "ADSL", "Да", "Да", "Да", "Нет", "Да", "Нет", "Да"},
    {"Planet GRT-501", "SHDSL", "Да", "Нет", "Да", "Да", "Да", "Нет", "Да"},
    {"ADC PairGain Megabit Modem 300S", "SDSL", "Нет", "Нет", "Нет", "Нет", "Да", "Нет", "Нет"},
    {"Aviv SDSL-16S", "SDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
    {"Planet VC-200M", "VDSL", "Да", "Да", "Нет", "Нет", "Да", "Нет", "Да"},
    {"Zyxel Prestige 841 EE", "VDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Да", "Нет"},
    {"D-link DEV-301M", "VDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Да", "Нет"},
    {"Planet VC-102S", "VDSL", "Да", "Нет", "Нет", "Нет", "Нет", "Да", "Да"},
};

```

Рисунок 2.5 - Соответствие значения

Затем в соответствии с правилами построения модели формируется правило.

Из полей Label, расположенных непосредственно на основной форме, берутся статические данные, общие для каждого правила (рисунок 2.6).

```
Rule:
=(Исходные данные.Тип модема;
=(Исходные данные.Наличие маршрутизатора;
=(Исходные данные.Наличие коммутатора;
=(Исходные данные.Наличие DHCP-сервера;
=(Исходные данные.Поддержка VPN;
=(Исходные данные.Поддержка WEB-интерфейса;
=(Исходные данные.Наличие встроенного сплиттера;
=(Исходные данные.Адаптация для России;
Do
=(Цель.Выбрать модем; Под ваш выбор подходит;
EndR
```

Рисунок 2.6 - Статическая информация для составления правил

А затем к статической информации добавляется динамическая, которая берется из проведенной выборки.

После чего получается сформированное правило, приведенное на рисунке 2.7.

```
Rule 1
=(Исходные данные.Тип модема; ADSL)
=(Исходные данные.Наличие маршрутизатора; Да)
=(Исходные данные.Наличие коммутатора; Да)
=(Исходные данные.Наличие DHCP-сервера; Да)
=(Исходные данные.Поддержка VPN; Да)
=(Исходные данные.Поддержка WEB-интерфейса; Да)
=(Исходные данные.Наличие встроенного сплиттера; Да)
=(Исходные данные.Адаптация для России; Да)
Do
=(Цель.Выбрать модем; Под ваш выбор подходит NetGear DG834)
EndR
```

Рисунок 2.7 - Сформированное правило 1

Сгенерированное правило заносится в файл rule.txt. После чего начинается генерация второго правила.

Для создания второго правила матрицы параметров, по каждому из пунктов, за исключением последнего, выбирается первое значение, для последнего же, выбирается 2-е значение (рисунок 2.8).

```
TForm1 *Form1;
AnsiString mass[8][4]={
    {"ADSL", "VDSL", "SDSL", "SHDSL"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
};
```

Рисунок 2.8 - Второй выбор значений из матрицы

Далее из матрицы классов выбирается соответствующее значение, и генерируется правило 2 (рисунок 2.9).

```
Rule 2
=(Исходные данные.Тип модема; ADSL)
=(Исходные данные.Наличие маршрутизатора; Да)
=(Исходные данные.Наличие коммутатора; Да)
=(Исходные данные.Наличие DHCP-сервера; Да)
=(Исходные данные.Поддержка VPN; Да)
=(Исходные данные.Поддержка WEB-интерфейса; Да)
=(Исходные данные.Наличие встроенного сплиттера; Да)
=(Исходные данные.Адаптация для России; Нет)
Do
=(Цель.Выбрать модем; Под ваш выбор подходит TRENDnet Tw100-BRM504)
EndR
```

Рисунок 2.9 - Сформированное правило 2

Далее все правила формируются аналогичным образом. Путем полного перебора.

Ниже в блок-схеме приведен полный алгоритм генерации правил (рисунок 2.10).

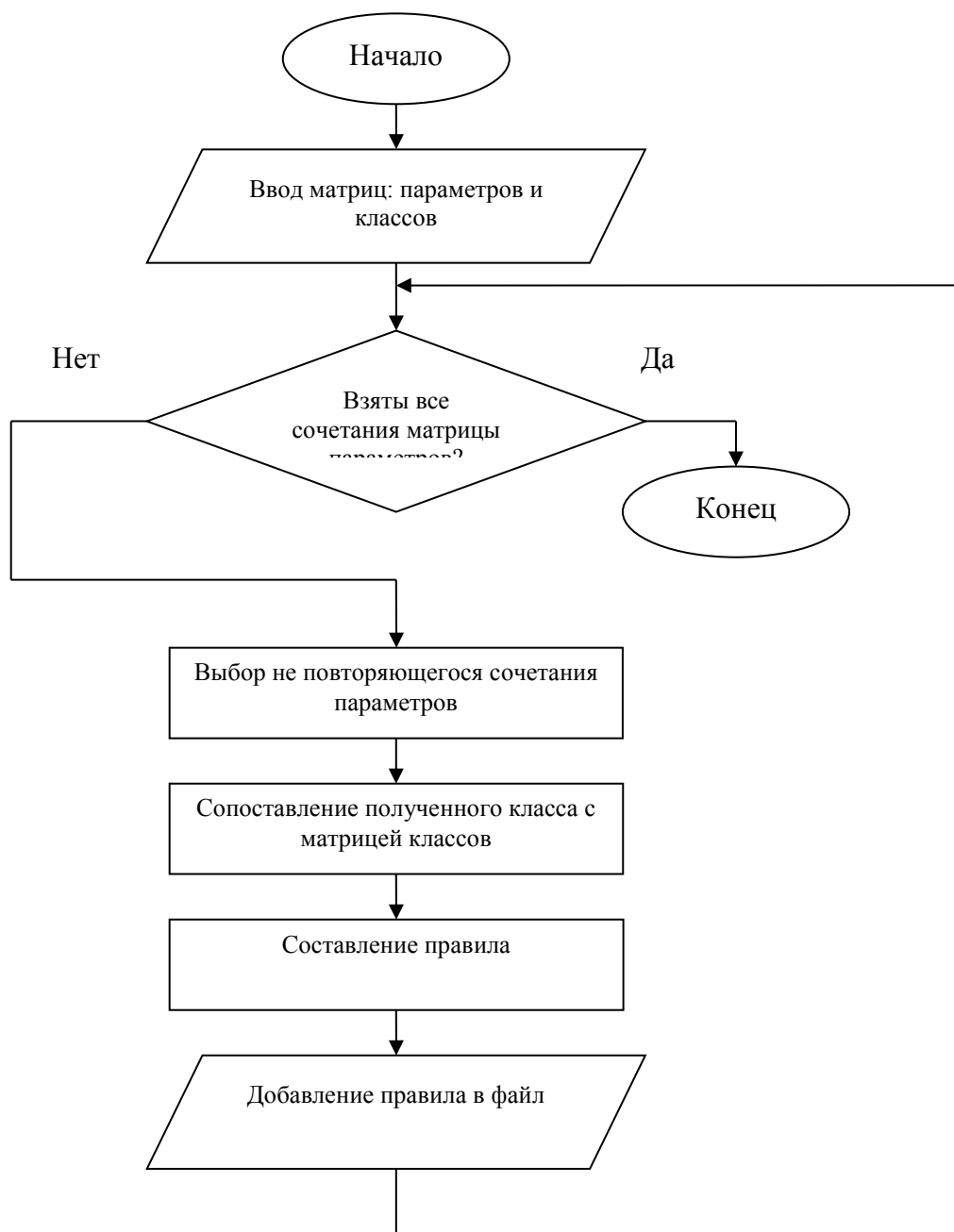


Рисунок 2.10 - Алгоритм генерации

2.3 Разработка интерфейса

Интерфейс пользователя - это та часть программы, которая находится на виду пользователя.

Как было выяснено выше, основной частью пользователей будут являться более или менее опытные программисты. В связи с этим было принято решение о создании программы со статическими данными. Т.е. изменение всех данных будет производиться непосредственно в коде программы.

Консоль приложения будет включать в себя (рисунок 2.11):

- окно в котором выводятся данные о создаваемом в настоящий момент правиле;
- полоса отображения выполнения процесса;
- кнопка начала генерации;
- кнопка просмотра результатов.

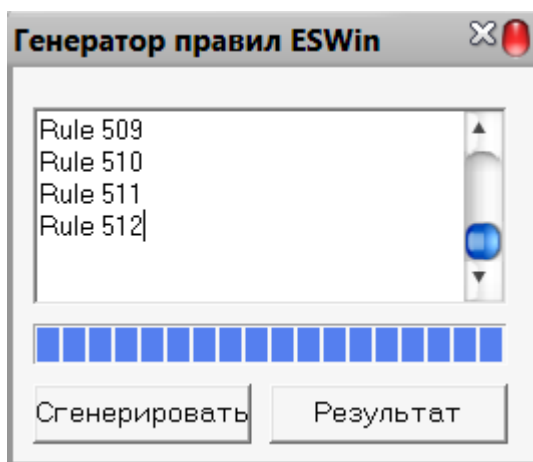


Рисунок 2.11 - Интерфейс пользователя

2.4 Разработка программной документации

Под программной документацией, в данном случае, понимается лабораторная работа по дисциплине «Интеллектуальные информационные системы» для студентов направления подготовки Прикладная информатика (Приложение А).

3 АПРОБАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ НА ПРИМЕРАХ И ЕЁ ВНЕДРЕНИЕ

3.1 Апробация на примере лабораторной работы в курсе ИИС

В ходе выполнения лабораторной работы №4 возникает необходимость разработки и создания экспертной системы с использованием ESwin.

Целью лабораторной работ является освоение технологии и методики построения экспертных систем на примере разработки учебной экспертной системы. Студент выступает в роли одновременно эксперта и инженера по знаниям.

Студент при выполнении данной работы может столкнуться с рядом сложностей. К ним можно отнести выбор классов, признаков, но, пожалуй, самой трудной, а точнее трудоемкой, частью работы является составление правил и установление наибольшего соответствия тому или иному классу.

В ходе работы вручную предлагается написать правила в определенном формате:

```
RULE                // правило-продукция
  <описание условий правила>
DO
  <описание заключений правила>
ENDR .
```

И затем сохранить их в текстовом документе с расширением *.klb, который уже непосредственно используется ESwin. Для создания достойной экспертной системы необходимо разработать как минимум 200 не повторяющихся правил. На это может уйти несколько часов.

Генератор правил ESwin предлагает проделать ту же работу с наименьшими затратами.

Для демонстрационного выполнения лабораторной работы была выбрана тема «Выбор модема».

Возьмем, к примеру, определенный перечень модемов:

- «Dynamix UM-S»;
- «NSGate qBRIDGE-105»;
- «Zyxel P-660RU EE»;
- «NetGear DG834»;
- «TRENDnet TW100-BRM504»;
- «Planet GRT-101»;
- «D-link DSL-2520U»;
- «ASUS AM604»;
- «Planet GRT-501»;
- «ADC PairGain Megabit Modem 300S»;
- «Aviv SDSL-16S»;
- «Planet VC-200M»;
- «Zyxel Prestige 841 EE»;
- «D-link DEV-301M»;
- «Planet VC-102S».

Основными параметрами их отличий являются:

- тип модема: ADSL, VDSL, SDSL, HDSL;
- наличие маршрутизатора;
- наличие коммутатора;
- наличие DHCP –сервера;
- поддержка VPN;
- поддержка WEB-интерфейса;
- наличие встроенного сплитера;
- адаптация для России.

Исходя из этих данных можно начать заполнение программы.

Первым шагом создается двумерный массив, размерностью $A*B$, где A - наибольшее число параметров одного признака, а B - количество признаков, заполняется параметрами из которых в последствии формируются правила, методом перебора.

Т.е. в нашем случае B - это количество параметров различающих модемы (8), A - количество типов модемов (4), этот признак обладает наибольшим числом параметров.

После создания и заполнения массива получим матрицу представленную на рисунке 3.1.

```
AnsiString mass[8][4]={
    {"ADSL", "VDSL", "SDSL", "SHDSL"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
    {"Да", "Нет", "0", "0"},
};
```

Рисунок 3.1 - Матрица параметров на примере

Следующим шагом заполняется второй двумерный массив, размерностью $(B+1)*C$, где C - количество классов, заполняется классами с соответствующими им признаками.

Как сказано выше B - это количество параметров различающих модемы, их 8, а прибавляющаяся единица непосредственно название модема.

C - это количество рассматриваемых в примере модемов.

После создания и заполнения второго массива получим матрицу представленную на рисунке 3.2.

```

AnsiString modem[15][9]={
  {"Dynamix UM-S", "SHDSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Да"},
  {"NSGate qBRIDGE-105", "SDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
  {"Zyxel P-660RU EE", "ADSL", "Да", "Нет", "Да", "Нет", "Да", "Нет", "Да"},
  {"NetGear DG834", "ADSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Да"},
  {"TRENDnet TW100-BRM504", "ADSL", "Да", "Да", "Да", "Да", "Да", "Нет", "Нет"},
  {"Planet GRT-101", "SHDSL", "Да", "Нет", "Да", "Нет", "Да", "Нет", "Да"},
  {"D-link DSL-2520U", "ADSL", "Да", "Нет", "Да", "Да", "Да", "Нет", "Да"},
  {"ASUS AM604", "ADSL", "Да", "Да", "Да", "Нет", "Да", "Нет", "Да"},
  {"Planet GRT-501", "SHDSL", "Да", "Нет", "Да", "Да", "Да", "Нет", "Да"},
  {"ADC PairGain Megabit Modem 300S", "SDSL", "Нет", "Нет", "Нет", "Нет", "Да", "Нет", "Нет"},
  {"Aviv SDSL-16S", "SDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет", "Нет"},
  {"Planet VC-200M", "VDSL", "Да", "Да", "Нет", "Нет", "Да", "Нет", "Да"},
  {"Zykel Prestige 841 EE", "VDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Да", "Нет"},
  {"D-link DEV-301M", "VDSL", "Нет", "Нет", "Нет", "Нет", "Нет", "Да", "Нет"},
  {"Planet VC-102S", "VDSL", "Да", "Нет", "Нет", "Нет", "Нет", "Да", "Да"},
};

```

Рисунок 3.2 - Матрица классов на примере

В поля label, расположенные на форме и в последующем скрытые, заносим информацию о параметрах отличающих модемы в таком виде, в котором они в последующем будут заноситься в файл, а также вспомогательную информацию для создания правил (рисунок 3.3).

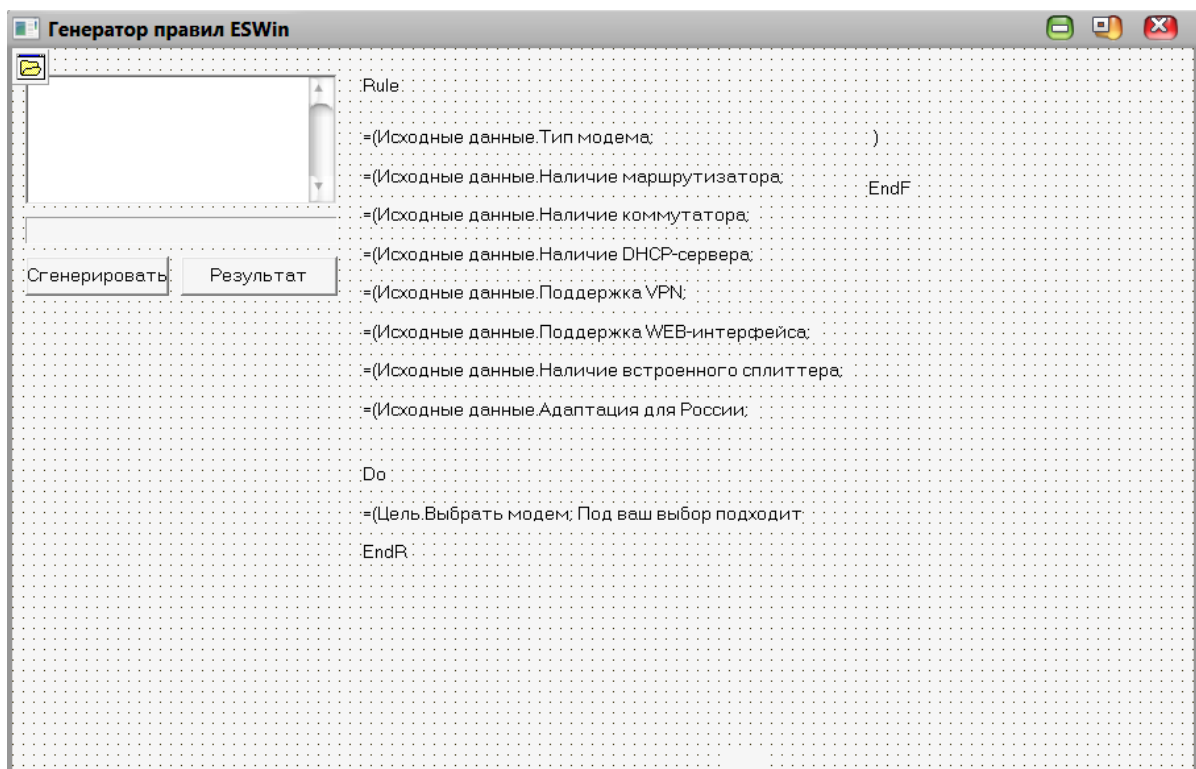


Рисунок 3.3 - Заполненные поля label

По желанию все выше перечисленные данные можно вписать непосредственно в программный код.

Затем редактируется сам код генерации, с подведением под заданные параметры генерации.

В Приложении Б приведена часть кода программы генерации правил для ЭС выбор модема.

Затем, после компиляции, запускаем программу, которая в процессе своей работы генерирует файл «rule» содержащий правила. Загружаем его в ESwin. И отвечаем на задаваемые вопросы.

Ниже на скриншотах показана работа экспертной системы работающей на основании сгенерированных выше правил.

Первым шагом загружаем базу знаний (рисунок 3.4).

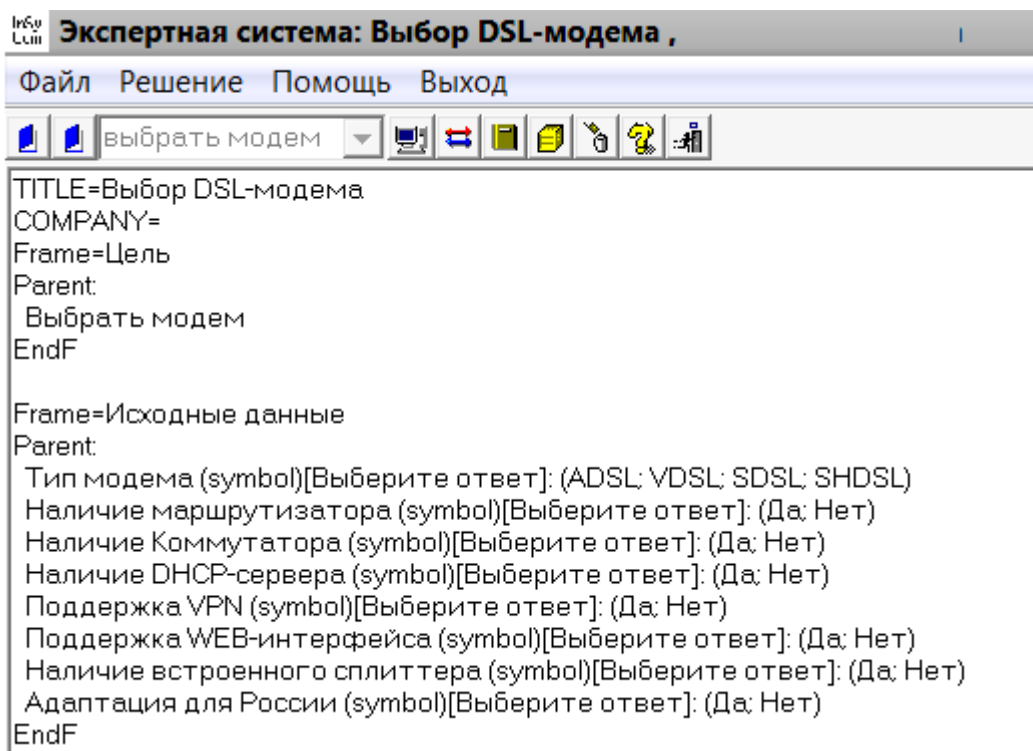


Рисунок 3.4 - Загрузка базы знаний

Далее отвечаем на задаваемые экспертной системой вопросы (Рисунок 3.5-3.7).

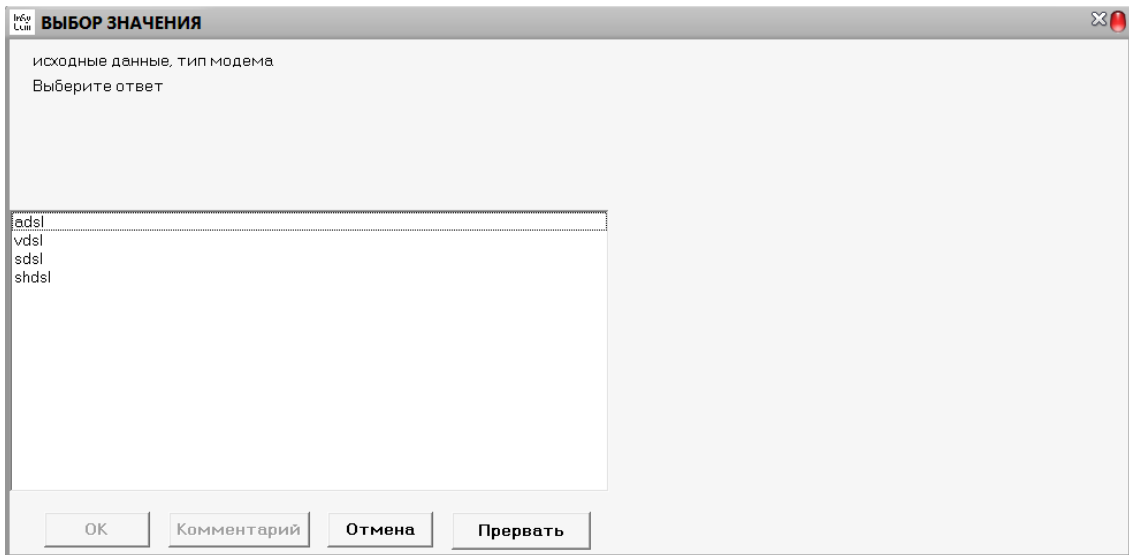


Рисунок 3.5 - Выбор значения 1

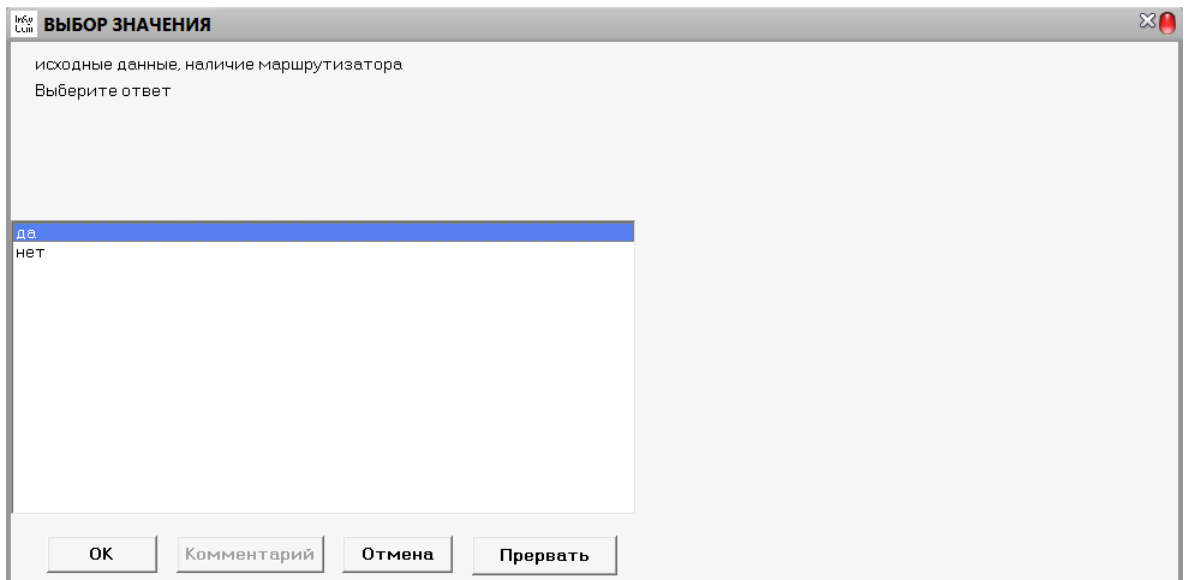


Рисунок 3.6 - Выбор значения 2

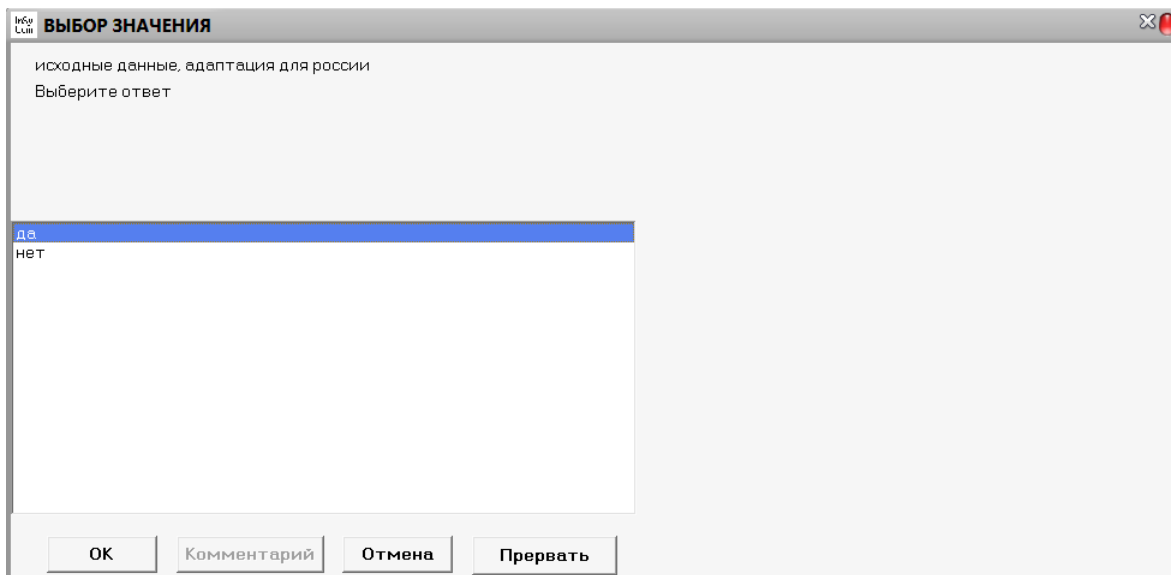


Рисунок 3.7 - Выбор значения 8

После ввода последнего значения, экспертная система подбирает наиболее подходящий вариант и выводит результаты на дисплей.

При тестировании программы по всем значениям выбирались первые ответы. В результате под данные параметры должен подходить модем: Zyxel p-660ru ee.

Как видно на рисунке 3.8, экспертная система выдает необходимый результат в должной форме.

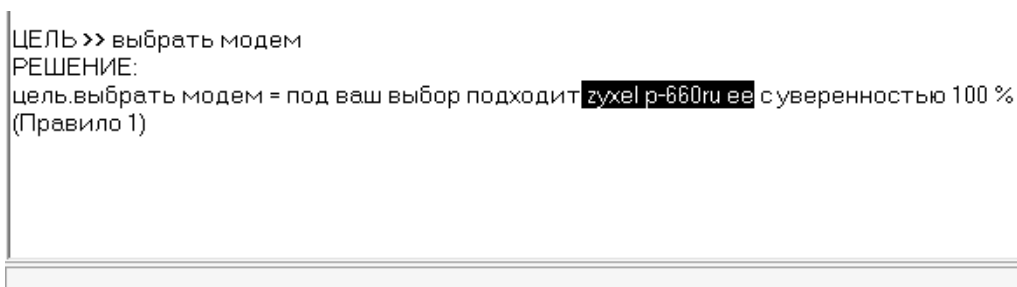


Рисунок 3.8 - Результат работы ЭС

Затем была проведена апробация с различными комбинациями значений. Как показали испытания, программа работает стабильно с правилами, сгенерированными разработанным ПС.

Как видно из выше рассмотренного, программа работает стабильно и точно, т.е апробацию можно считать успешной.

3.2 Апробация на примере выбора кабеля в ООО «РЖД»

Практика была пройдена в ООО «РЖД». В ходе прохождения практики было принято решение об автоматизации выбора кабеля, по его параметрам.

Кабель (нидерл. kabel) - один или несколько изолированных друг от друга проводников (жил), заключённых в оболочку.

Материал из которых изготавливаются проводники обладает огромным разнообразием.

Проводники в кабелях изготавливаются из следующих материалов [1]:

1.Для передачи электрической энергии и сигналов:

- алюминий;
- медь;
- серебро;
- золото;
- сплавов различных металлов.

2.Для передачи оптических сигналов:

- стекло;
- пластмассы.

3.Для рассеивания тепла:

- нихрома;
- константана.

Оболочка кабеля обладает огромным разнообразием.

Оболочка кабеля может состоять из одного и более герметизирующих и армирующих слоёв, в качестве этих слоёв могут применяться различные материалы: ткань, пластмассы, металл, резина и проч. Кабели для передачи

электрических сигналов могут быть снабжены экраном из металлической сетки, листового металла (фольги) или полимерной плёнки с тонким металлическим покрытием.

Оболочка кабелей, прокладываемых открыто, должна быть не распространяющей горение [1].

Кабельные изделия должны подразделяться по показателям пожарной безопасности на следующие типы исполнения:

- кабельные изделия, не распространяющие горение при одиночной прокладке (без исполнения);
- кабельные изделия, не распространяющие горение при групповой прокладке (исполнение - нг);
- кабельные изделия, не распространяющие горение при групповой прокладке, с пониженным дымо- и газовыделением (исполнение - нг-LS);
- кабельные изделия, не распространяющие горение при групповой прокладке и не выделяющие коррозионно-активных газообразных продуктов при горении и тлении (исполнение - нг-HF);
- кабельные изделия огнестойкие, не распространяющие горение при групповой прокладке, с пониженным дымо- и газовыделением (исполнение - нг-FRLS);
- кабельные изделия огнестойкие, не распространяющие горение при групповой прокладке и не выделяющие коррозионно-активных газообразных продуктов при горении и тлении (исполнение -нг-FRHF);
- кабельные изделия, не распространяющие горение при групповой прокладке, с пониженным дымо- и газовыделением и с низкой токсичностью продуктов горения (исполнение - нг-LSLTx);
- кабельные изделия, не распространяющие горение при групповой прокладке, не выделяющие коррозионно-активные газообразные продукты при горении и тлении и с низкой токсичностью продуктов горения (исполнение - нг-HFLTx) [11].

На сегодняшний день в России выпускается более 20 тыс. типоразмеров кабеля.

По области применения их можно условно разделить на несколько групп:

- для передачи электрической энергии (силовые кабели);
- для проводной связи и сигнализации (кабели связи);
- для управления (кабели управления);
- для передачи энергии и сигналов:
 - на радиочастотах - (радиочастотные кабели);
 - в оптическом диапазоне - (оптические кабели).

Также кабели разделяют по:

- типу и наличию изоляции;
- типу и наличию экрана;
- по количеству жил;
- по материалу, из которого изготовлены провода;
- по гибкости:
 - для подвижного соединения;
 - для не подвижного соединения.

Стандарт ISO 11801 2002 детально описывает классификацию кабелей.

Для передачи большой мощности применяются высоковольтные силовые кабели. Современные кабели производятся с изоляцией из сшитого полиэтилена и используются в сетях различного класса напряжения (до 500 кВ). Применение сшитого полиэтилена обеспечивает высокие диэлектрические свойства изоляции, высокие механические свойства, более высокие по сравнению с бумажно - масляной изоляцией термические режимы, надёжность и долговечность кабелей.

Также кабели обладают огромным спектром производителей. А зачастую марка производителя является синонимом уровня качества и долговечности.

Основные Российские производители:

- «АББ Москабель»;

- «Инкаб»;
- «Иркутсккабель»;
- «Кавказкабель»;
- «Камский кабель»;
- «Кирскабель»;
- «Людиновокабель»;
- «Псковкабель»;
- «Подольсккабель»;
- «Режевской кабельный завод»;
- «Рыбинский кабельный завод»;
- «Самарская кабельная компания»;
- «Саранская кабельная компания»;
- «Севкабель-Холдинг»;
- «Сибкабель»;
- «Спецкабель»;
- «Уралкабель»;
- «Уфимкабель»;
- «Электон Кировский кабельный завод»;
- «Электрокабель, Кольчугинский завод»;
- «Эликс-Кабель»;
- «Энергорегионкомплект».

Основные иностранные производители:

- «Helukabel», Германия;
- «Lapp Kabel», Германия;
- «Sudkabel», Германия;
- «nkt cables», Германия;
- «Novkabel», Сербия;
- «Brugg Kabel AG», Швейцария;
- «Prysmian cables», Финляндия;

- «REKA cables», Финляндия [17].

Как видно из вышесказанного кабель имеет огромное разнообразие, и всего от одной буквы в названии кабеля может зависеть качество, сроки поставки, а самое главное стоимость. А именно эти 3 параметра являются ключом успешной торговли.

Ракой объем разновидностей сложно удержать в голове, а тем более собрать в единую маркировку кабеля.

3.2.1 Описание основных свойств ИС для предприятия и выбранного комплекса задач

ИС внедряемая в деятельность ООО «РЖД» должна быть:

- простой в эксплуатации;
- обладать понятным интерфейсом;
- иметь не высокую стоимость, т.к организация является мелкой и не может позволить себе больших затрат на разработку и внедрение ПО.

ИС должна иметь перечень основных существующих кабелей. В параметрах выбора кабеля должно присутствовать все необходимое. Система должна выдавать точные и четкие решения.

В комплекс поставленных задач входит:

- изучение типов кабельной продукции;
- изучение видов кабельной продукции;
- изучение производителей кабельной продукции;
- Изучение маркировки и аббревиатур кабельной продукции;
- выбор оболочки, системы поддержки принятия решений;
- разработка ПО для автоматического создания правил для системы поддержки принятия решений основанной на продукционно-фреймовой модели;

- создание правил;
- внедрение ПО;
- составление руководства пользования системой;
- обучение персонала.

3.2.2 Экономическая сущность комплекса задач

В случае выполнения поставленных задач скорость приема и рассмотрения заявок возрастает в разы. Из этого следует возрастание скорости выставления счетов, с уже установленными срокам поставки. А как следствие увеличение конкурентоспособности. Что непосредственно укрепит положение организации на рынке.

Также высокая скорость обслуживания и профессиональный подход к рассмотрению заявок укрепляет авторитет организации в глазах клиентов. А это, как правило, увеличивает спрос на продукцию организаций. В этом случае обороты, а следовательно и прибыль организации увеличивается.

При постоянной работе в таком режиме нарабатывается определенная клиентская база. Клиенты, которые работают только с одной организацией, чувствуют себя более уверенно при совершении сделок, и даже при не большем увеличении цен не стремятся расторгнуть контракт. В следствии этого также увеличивается прибыль организации.

3.2.3 Определение места проектируемой задачи в комплексе задач

Разрабатываемая программа будет использоваться менеджерами при подборе конкретного товара, а точнее кабеля. При введении необходимых параметров программа выводит на экран необходимую марку кабеля.

Без применения программных средств, процесс подбора кабеля занимает длительный срок, а, как правило, это сокращает конкурентоспособность на рынке.

3.2.4. Анализ существующих разработок для автоматизации комплекса задач

Cable 1.1

Существует программа Cable 1.1. Она выполняет не совсем те функции которые нужны для поставленных целей.

Программа предназначена для подбора сечения проводника кабеля или провода. Исходными данными могут быть либо мощность, которую должен выдержать проводник, либо сила тока, проходящего через поперечное сечение проводника. Род тока также можно задавать: переменный однофазный, переменный трехфазный, постоянный. Напряжение жестко привязано, и его можно только выбрать из ниспадающего списка. Тип проводника: медный или алюминиевый. Также выбирается способ прокладки кабеля. На рисунке 3.10. приведен пример работы программы.

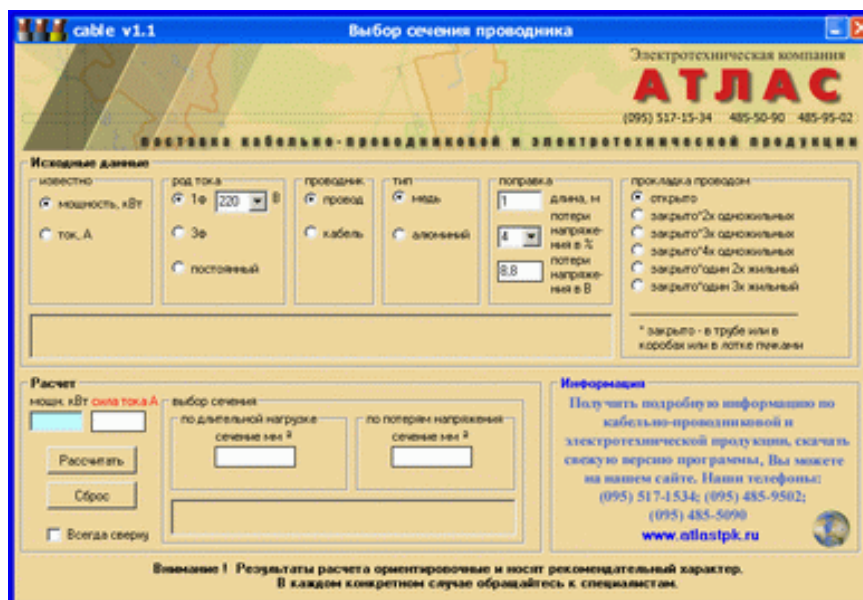


Рисунок 3.10 - Работа программы Cable 1.1

CalcSec

Приложение CalcSec подбирает кабели и провода по нагрузке и, наоборот, рассчитывает нагрузку в зависимости от различных факторов (рисунок 3.11):

- марка кабеля;
- материал изоляции;
- максимально допустимая температура проводящей жилы;
- металлическая оболочка;
- число жил;
- число кабелей в фидере;
- температура окружающей среды;
- способ прокладки;
- режим нагрузки;
- сечение.

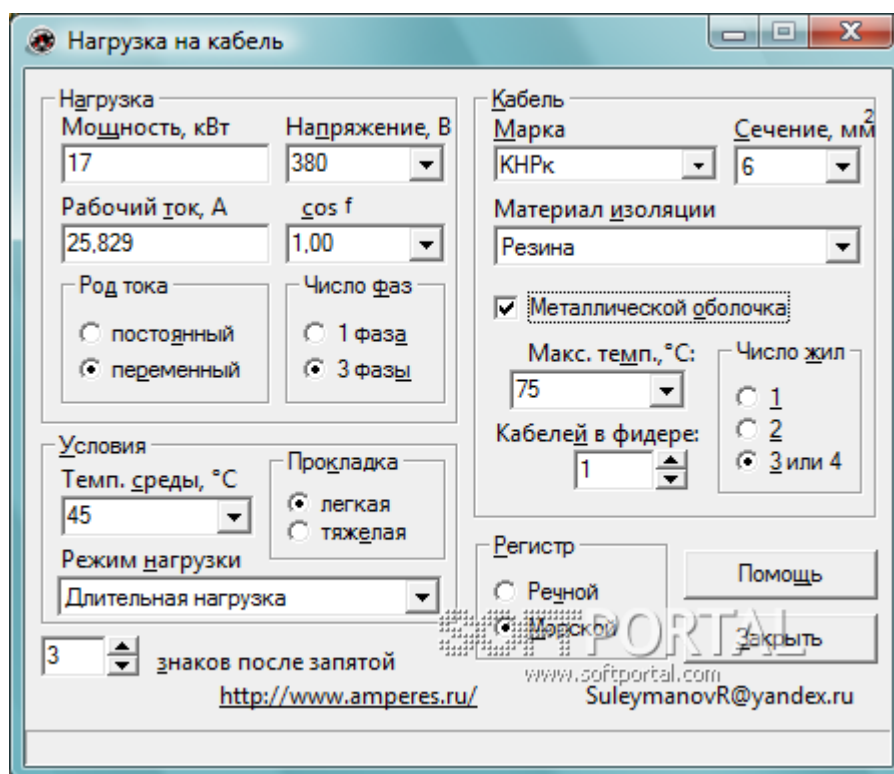


Рисунок 3.11 - Работа программы CalcSec

Нагрузка может задаваться как токовая, так и по мощности.

Программа работает так, что, вводя значение в любой из элементов управления, все другие параметры автоматически пересчитываются. Задав значение на панели нагрузки, происходит расчет минимально допустимого сечения для жилы кабеля или провода. Если выбирается сечение из списка на панели «Кабель», то рассчитывается максимально допустимая нагрузка в фидере - величина тока и мощность.

В зависимости от марки кабеля определяется максимально допустимая температура проводящей жилы, требуемая для расчета нагрузок или сечения жилы кабеля.

Если марка кабеля не определена, то максимально допустимая температура жилы кабеля определяется исходя из материала кабеля или провода.

Температура, максимально допустимая для данного кабеля и данного материала изоляции.

Для каждой марки кабеля установлена своя максимальная температура жилы. В случае если марка кабеля не определена, температура может быть установлена по материалу изоляции. Температуру можно также задать, выбрав марку кабеля, либо просто вписав значение в соответствующее поле ввода на панели «Кабель».

Наличие вокруг материала изоляции кабеля металлической оболочки увеличивает допустимую кратковременную или повторно-кратковременную нагрузку.

Чем больше жил (одна, две, три или четыре), тем хуже теплоотдача от проводников кабеля окружающей среде и, следовательно, уменьшается допустимая нагрузка.

Увеличение количества кабелей распределяет нагрузку между всеми кабелями. Допустимая нагрузка для данного фидера при прочих одинаковых условиях увеличивается в n раз, где n – число кабелей в фидере.

Если температура окружающей среды отлична от приведенной в правилах, то допустимые нагрузки пересчитываются в соответствии с вновь введенным значением температуры.

Условия прокладки кабелей могут быть отнесены к одному из двух приведенных ниже случаев.

Легкие условия прокладки:

- прокладка не более шести кабелей одним пучком или в один ряд с плотным прилеганием друг к другу;
- прокладка кабелей в два ряда, независимо от количества кабелей в ряду, при условии, что между группой или пучком из шести кабелей имеется свободное пространство для циркуляции воздуха.

В тех случаях, когда кабели или провода установлены в цепях повторно-кратковременной или кратковременной работы нагрузки могут быть увеличены в зависимости от сечения кабеля (провода), режима нагрузки и наличия металлической оболочки.

На панели «Нагрузка» рассчитывается мощность и величина тока в зависимости друг от друга, а также различных параметров: напряжения, коэффициента мощности, рода тока и фазности.

При работе с программой следует задать, по каким правилам будет вестись расчет нагрузок, в зависимости от того под какой регистр подпадает судно, кабели и провода, на котором вы желаете рассчитать. В целом, токовые нагрузки не сильно будут различаться в зависимости от «Правил...», по которым они будут рассчитываться [19].

Но эта программа также не позволяет определить марку кабеля в зависимости от его параметров.

Для целей определенных выше нам необходимо создать собственную экспертную систему.

3.2.5 Выбор и обоснование стратегии автоматизации комплекса задач

Как и в любом процессе автоматизации, начальный этап стратегии будет представлен изучением предметной области:

- изучение типов кабельной продукции;
- изучение видов кабельной продукции;
- изучение производителей кабельной продукции;
- изучение маркировки и аббревиатур кабельной продукции.

Затем будут выбраны средства, с помощью которых будет реализовываться поставленная цель: выбор оболочки системы поддержки принятия решений.

И, наконец, завершающим этапом автоматизации будет являться:

- разработка ПО для автоматического создания правил для системы поддержки принятия решений основанной на продукционно-фреймовой модели;
- создание правил;

- внедрение ПО.

Также, после осуществления внедрения системы, может возникнуть потребность в оказании сопутствующих услуг:

- составление руководства пользования системой;
- обучения персонала.

Модель комплекса задач представлена на рисунке 3.12.

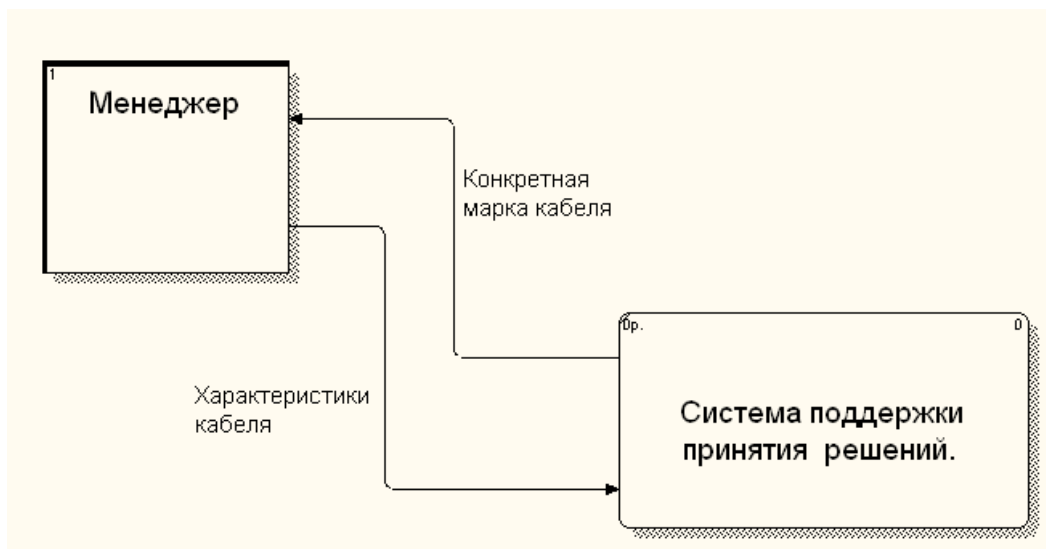


Рисунок 3.11 - Модель комплекса задач

Как было сказано выше, система поддержки принятия решений помогает решить проблему выбора кабеля по введенным параметрам.

Это и представлено на рисунке 3.11.

3.2.6 Создание экспертной системы выбор кабеля

Экспертная система «Выбор кабеля» создается аналогично системе выбор модема, описанной ранее. Она создается на основании знаний полученных в процессе изучения кабельной продукции.

Для того чтобы создать экспертную систему по выбору кабеля необходимо понять расшифровку аббревиатур таких марок как: ВВГ, АВВГ,

ВВГнг, АВВГнг, ВВГнг-LS, АВВГнг-LS, ВБбШв, АВБбШв, ВБбШнг, АВБбШнг, ВБбШнг-LS, АВБбШнг-LS, достаточно знать:

– Присутствие буквы А в маркировке указывает на алюминиевую жилу. В случае отсутствия данной буквы в маркировке указывает на медную жилу;

– При отсутствии буквы А в маркировке, первая буква В, в противном случае вторая, указывает на материал внутренней изоляции поливинилхлорид (в дальнейшем ПВХ);

– При отсутствии буквы А в маркировке, вторая буква В, в противном случае третья, указывает на материал внешней изоляции (оболочка) ПВХ;

– Присутствие буквы Г в маркировке указывает на отсутствие защитного покрова (от сокращения «голый»);

– нг - сокращение слов « не поддерживающий горения»;

– LS - сокращение от американского Low Smoke - низкое газодымовыделение;

– Бб указывает на наличие двойной брони из стальных лент;

– Шв - наружный покров из ПВХ шланга [1].

Вся необходимая информация добавляется в программу генерации правил после чего генерируем правила, для выбора кабеля. Представленный на рисунке 3.12.


```

Выбор Кабеля.klb - Блокнот
Файл  Правка  Формат  Вид  Справка
TITLE=Выбор Кабеля
COMPANY=ООО'
Frame=Цель
Parent:
  Выбрать Кабель
EndF

Frame=Исходные данные
Parent:
  наружный покров из ПВХ шланга (symbol)[Выберите ответ]: (да; нет)
  материал (symbol)[Выберите ответ]: (медь; алюминий)
  материал внутренней изоляции ПВХ (symbol)[Выберите ответ]: (да; нет)
  материал внешней изоляции ПВХ (symbol)[Выберите ответ]: (да; нет)
  отсутствие защитного покрова (symbol)[Выберите ответ]: (да; нет)
  не поддерживающий горения (symbol)[Выберите ответ]: (да; нет)
  низкое газо-дымовыделение (symbol)[Выберите ответ]: (да; нет)
EndF

Rule 1
=(Исходные данные.наружный покров из ПВХ шланга; нет)
=(Исходные данные.материал; Алюминий)
=(Исходные данные.материал внутренней изоляции ПВХ; да)
=(Исходные данные.материал внешней изоляции ПВХ; да)
=(Исходные данные.отсутствие защитного покрова; да)
=(Исходные данные.не поддерживающий горения; да)
=(Исходные данные.низкое газо-дымовыделение; да)
Do
=(Цель.Выбрать Кабель; Под ваш выбор подходит АВВГнг-LS)
EndR

Rule 2
=(Исходные данные.наружный покров из ПВХ шланга; нет)
=(Исходные данные.материал; Алюминий)
=(Исходные данные.материал внутренней изоляции ПВХ; да)
=(Исходные данные.материал внешней изоляции ПВХ; да)
=(Исходные данные.отсутствие защитного покрова; да)
=(Исходные данные.не поддерживающий горения; да)
=(Исходные данные.низкое газо-дымовыделение; нет)
Do
=(Цель.Выбрать Кабель; Под ваш выбор подходит АВВГнг)
EndR

Rule 3
=(Исходные данные.наружный покров из ПВХ шланга; нет)
=(Исходные данные.материал; Медь)
=(Исходные данные.материал внутренней изоляции ПВХ; да)
=(Исходные данные.материал внешней изоляции ПВХ; да)

```

Рисунок 3.12 - Модель комплекса задач

Затем как, и в первом примере, файл с правилами загружается в ESWin. Далее даются ответы на поставленные вопросы. Работа в ESWin представлена на рисунках 3.13.-3.15.

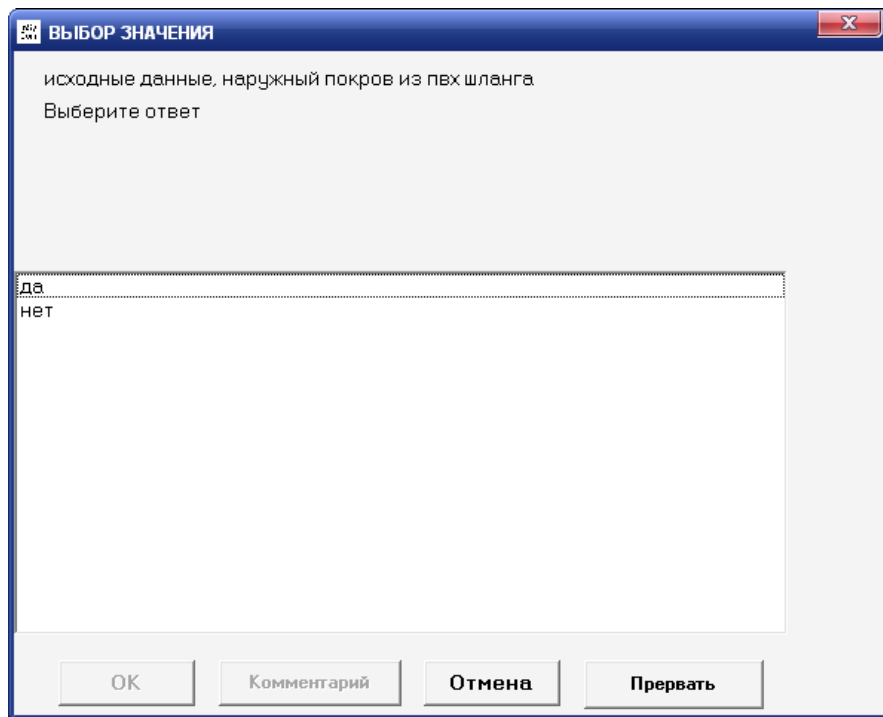


Рисунок 3.13 - Выбор значения 1

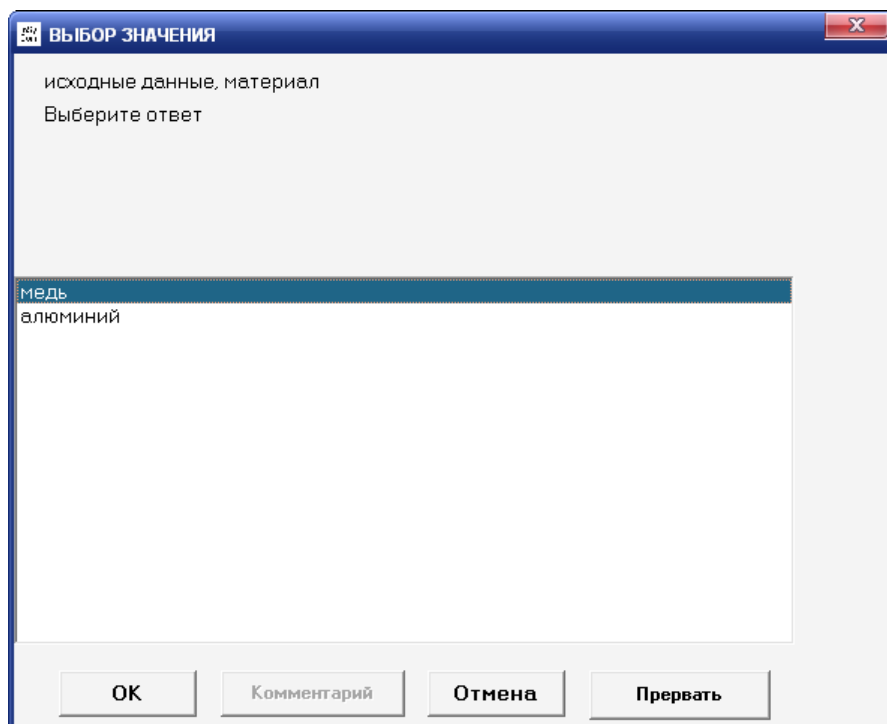


Рисунок 3.14 - Выбор значения 2

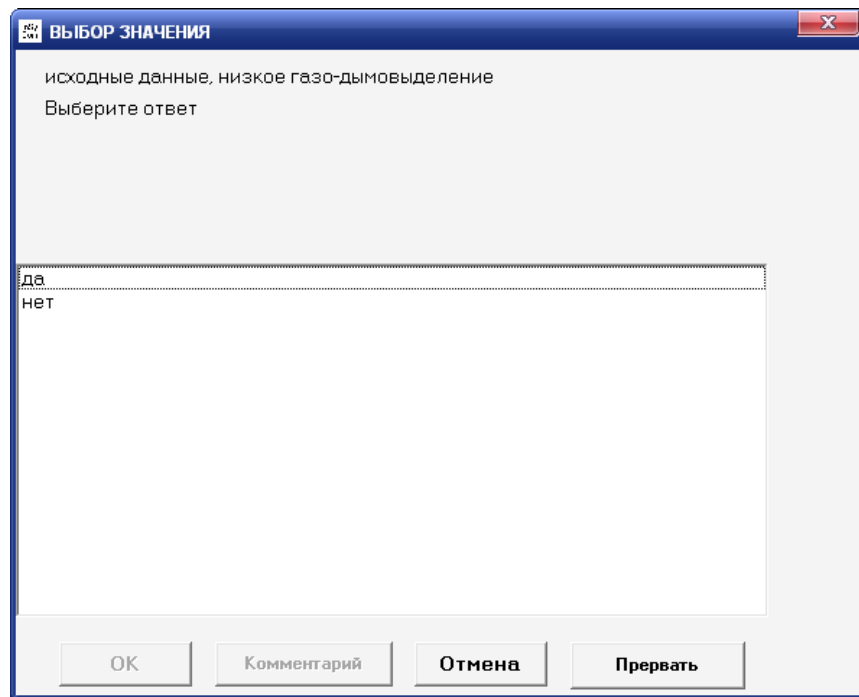


Рисунок 3.15 - Выбор значения б

В результате экспертная система выдает результат, соответствующий ответам, даваемым на поставленные вопросы (рисунок 3.16).

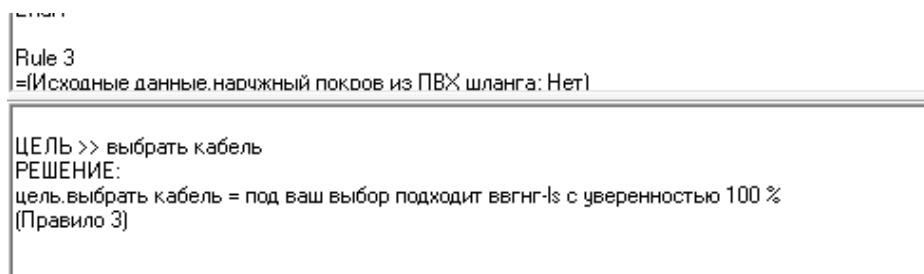


Рисунок 3.16 - Результат выбора кабеля

В данном случае был выбран кабель ВВГнг-LS, соответствующий правилу 3 (рисунок 3.17).

```

Rule 3
=(Исходные данные.наружный покров из ПВХ шланга; Нет)
=(Исходные данные.Материал; Медь)
=(Исходные данные.материал внутренней изоляции ПВХ; Да)
=(Исходные данные.материал внешней изоляции ПВХ; Да)
=(Исходные данные.отсутствие защитного покрова; Да)
=(Исходные данные.не поддерживающий горения; Да)
=(Исходные данные.низкое газо-дымовыделение; Да)
Do
=(Цель.Выбрать Кабель; Под ваш выбор подходит ВВГнг-LS)
EndR

```

Рисунок 3.17 - Правило 3

3.3 РЕЗУЛЬТАТЫ ВНЕДРЕНИЯ СИСТЕМЫ

3.3.1 Внедрение в учебный процесс

В процессе разработки ПС и подготовки программной документации, была разработана лабораторная работа «Автоматизированное создание правил с последующим использованием для разработки экспертных систем на основе ESWin». В данной работе используется программный продукт «Генератор правил ESwin» для создания экспертной системы. Лабораторная работа была внедрена в учебный процесс студентов 4 курса Института инженерных технологий и естественных наук по направлению подготовки «Прикладная информатика».

Цель лабораторной работы: изучение программного продукта «Генератор правил ESwin», освоение технологии генерации правил с последующим созданием экспертной системы на основании сгенерированных правил.

Работа включает в себя:

- теоретическую часть, включающую в себя:
 - структуру данных программного продукта;

- алгоритм генерации правил.
- выполнение лабораторной работы, состоящую из:
 - описаний,
 - схем,
 - скриншотов.

Разработанная лабораторная работа является довольно простым и понятным учебным пособием (Приложение Б).

Увеличение скорости набора правил достигнутое за счет автоматической генерации правил можно проследить в таблице 3.1.

Таблица 3.1 - Скорость набора правил

Количество правил	Ручной набор правил	Генерация
До 20	15-20 мин	15-20 мин
20-100	0.9-1 мин/правило	0.3-0.35 мин/правило
100-500	1-1.2 мин/правило	0.15-0.2 мин/правило
500-1000	1.2-1.5 мин/правило	0.1-0.05 мин/правило
>1000	Проблематично	0.05-0.02 мин/правило

Из таблицы видно, что создание маленького количества правил не требует автоматизации процесса. Но с увеличением числа правил скорость их создания в автоматическом режиме превосходит в разы, скорость ручного ввода. В таблице приведены скорости внесения данных без разработки правил.

3.3.2 Внедрение на предприятии

Во время прохождения преддипломной практики в ООО «РЖД» была внедрена экспертная система по выбору кабеля, созданная на основании «Генератора правил», разработанного в ходе выполнения ВКР.

С помощью данной системы менеджер данной организации может подобрать маркировку необходимого кабеля по его параметрам, за считанные секунды.

До внедрения системы поддержки принятия решений подбор кабельной продукции занимал значительный промежуток рабочего времени. Он осуществлялся с помощью бумажных каталогов. А т.к. существует огромное разнообразие кабельной продукции, не всегда можно было подобрать необходимый товар. На рисунке 3.18 вы можете увидеть диаграммы распределения рабочего времени менеджера до и после внедрения системы.

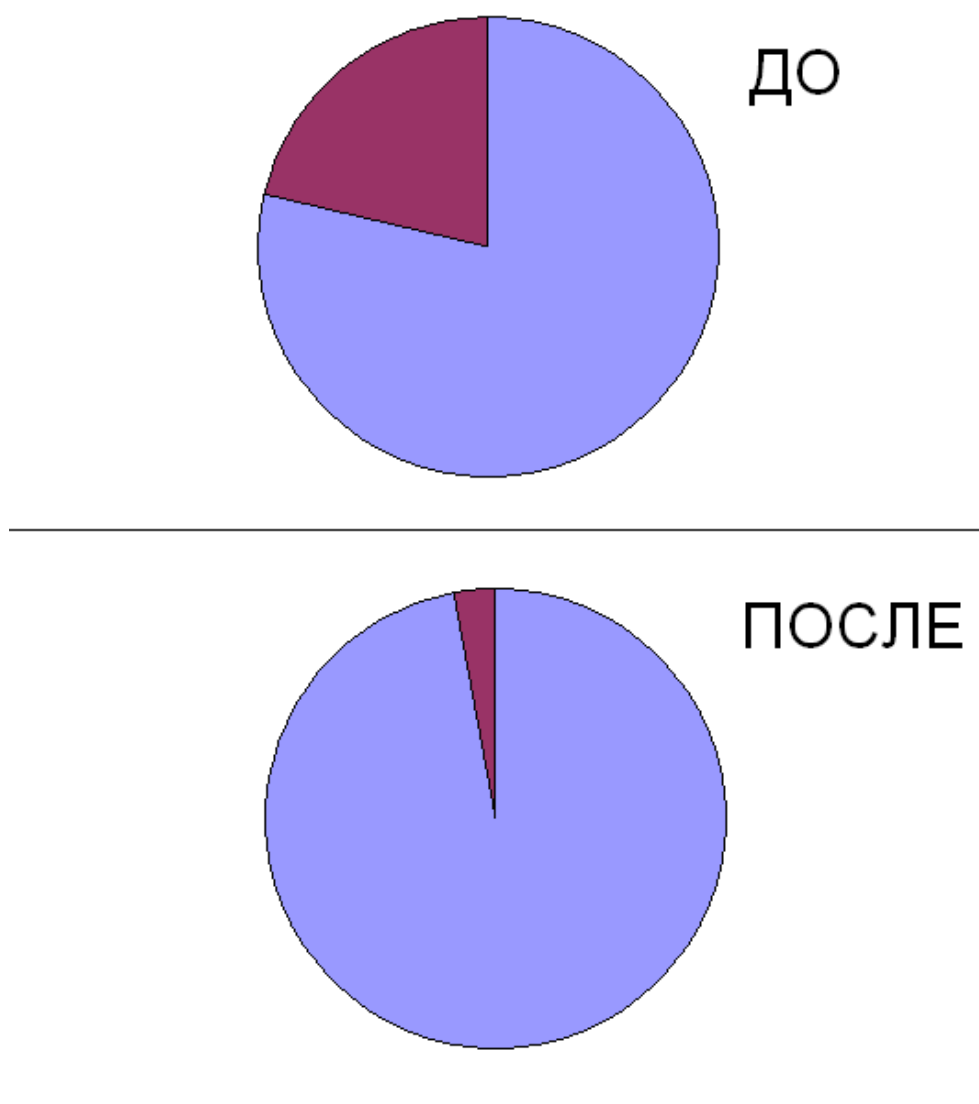


Рисунок 3.18 - Диаграммы распределения времени

3.3.3 Оценка экономической эффективности

В данном проекте можно рассмотреть два вида экономической эффективности:

– к первому мы можем отнести эффективность, относящуюся к прямому пользователю, т.е. непосредственно программисту - разработчику экспертных систем. Сюда можно приписать снижение времени и трудозатрат в разы, при создании экспертных систем.

– ко второму виду можно отнести косвенных пользователей ПС, т.е. пользователей уже готовых экспертных систем. Разумеется, при снижении трудозатрат стоимость конечного продукта сокращается, что выгодно для пользователей. Также «приятным фактом» для конечного пользователя является увеличение скорости разработки ЭС.

В таблице 3.2. представлен SWOT анализ программного средства.

Таблица 3.2 - SWOT – матрица

Сильные стороны	Возможности		Угрозы.	Итого
	1. Расширение круга потребителей	2. Совершенствование разработки	1. Появление новых конкурентов	
1. Низкая стоимость разработки	++	0	++	+4
2. Низкое число конкурентов	++	+	+	+4
Итого	+5	+1	+3	+8
Слабые стороны				
1. Направленность на квалифицированных пользователей	-	0	--	-3
Итого	-1	0	-2	-3
Общий итог	+4	+1	+1	+6

Проанализировав полученную SWOT - матрицу, можно сделать следующие выводы:

1. обе сильные стороны являются довольно важными. Они плотно укрепляют продукт на рынке;
2. слабая сторона является довольно опасной;
3. из рассмотренных возможностей более реальной является расширение круга потребителей;
4. появление конкурентов возможно, но данная вероятность значительно снижается в случае разработки новых версий.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы, были выполнены все цели и задачи. Создан программный продукт, способный автоматически генерировать правила по заданным параметрам.

Система была апробирована и внедрена на предприятие ООО «РЖД», основным видом деятельности которого является торговля электрооборудованием и кабельной продукцией.

После проведения анализа деятельности предприятия была обоснована необходимость внедрения ЭС в процесс подбора кабеля по его параметрам. Это значительно снизило трудозатраты менеджеров при подборе продукции

Также разработанная программа была апробирована и внедрена в учебный процесс студентов 4 курса по направлению подготовки Прикладная информатика в виде лабораторной работы по курсу ИИС.

Была проведена оценка экономической эффективности, в ходе которой было выяснено, что разработка может иметь дальнейшие версии за счет низкой себестоимости.