

Глава 1. Классификация интеллектуальных информационных систем

1.1. Особенности и признаки интеллектуальности информационных систем

Принятие решений относительно действий или поведения в той или иной ситуации любыми субъектами (людьми, роботами, сложными системами управления) осуществляется на основе информационных процессов. Информационный процесс реализует отношение объекта и субъекта (рис.1.1) и представляет собой восприятие субъектом объективной реальности в виде данных, переработку этих данных в соответствии с целевой установкой и имеющимися знаниями о зависимостях фактов в информацию. На основе полученной информации происходит обновление знания субъекта, выработка решения по возможному изменению состояния объекта и целевой установки субъекта. Причем изменение состояния объекта во многих случаях может быть представлено в форме данных.

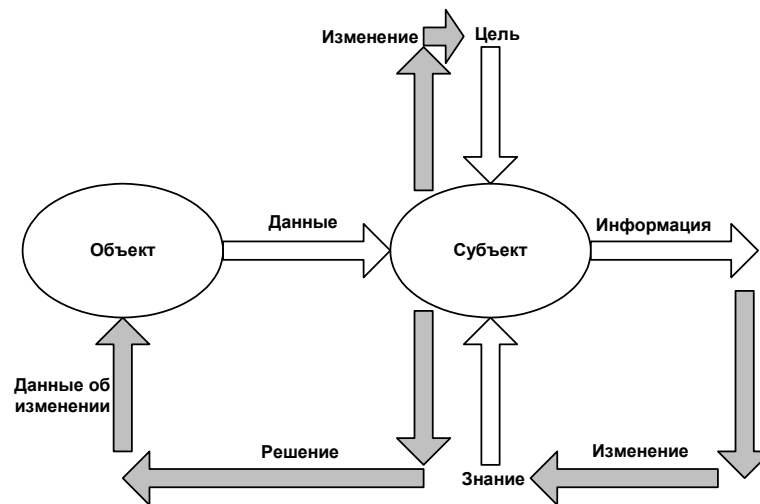


Рис . 1.1. Информационный процесс

Таким образом, информационный процесс рассматривается в трех аспектах:

- Синтаксический аспект – отображение объективной реальности в какой-либо среде и на каком-либо языке, которое представляет собой данные.
- Семантический аспект – понимание, осмысливание и интерпретация данных на основе знаний субъекта, которые отражают зависимости, закономерности взаимодействия объектов.
- Прагматический аспект – оценка полезности полученного нового знания (приращения знания) субъекта в соответствии с целевой установкой для принятия решения, то есть получение информации в узком смысле.

В широком смысле под информацией понимают все три аспекта отражения информационного процесса.

Любая компьютерная информационная система (ИС), реализующая информационный процесс, выполняет следующие функции: воспринимает вводимые пользователем информационные запросы (цели решения задачи) и необходимые исходные данные, обрабатывает введенные и хранимые в системе данные в соответствии с известным алгоритмом и формирует требуемую выходную информацию. С точки зрения реализации перечисленных функций ИС можно рассматривать как фабрику, производящую информацию, в которой заказом является информационный запрос, сырьем - исходные данные, продуктом - требуемая информация, а инструментом (оборудованием) - знание, с помощью которого данные преобразуются в информацию.

Знания субъектов информационного процесса могут представляться в различной форме. У людей знания представляются либо в недокументированной (неявной, непосредственно в голове) форме, либо в документированной (явной, книжной) форме. Причем документированная текстовая форма представления знаний в виде учебников, положений, инструкций и т.д. мало приспособлена для быстрого извлечения необходимых знаний при обосновании конкретных решений. Неявное знание экспертов вообще трудно доступно для использования в решении задач другими специалистами.

Компьютерные информационные системы, выступающие в роли субъектов информационного процесса, призваны упростить процесс использования знания в решении задач принятия решений. Для этого знания должны структурироваться и запоминаться для последующего многократного использования.

Знание имеет двоякую природу: фактуальную и операционную.

- Фактуальное знание представляет известные сведения об объектах отражаемой реальности и накапливается в обычных базах данных.
- Операционное знание отражает зависимости и отношения между объектами, которые позволяют интерпретировать данные или извлекать из них информацию. Операционное знание представляется либо в алгоритмической форме, либо в декларативной форме в виде специальных баз структурированных знаний.

Часто фактуальное знание называют экстенциональным (детализированным), а операционное знание - интенциональным (обобщенным).

Информационный процесс с помощью компьютерной информационной системы сводится к адекватному соединению операционного и фактуального знаний и в различных типах ИС выполняется по-разному. Самый простой путь их соединения заключается в рамках одной прикладной программы:

Программа = Алгоритм (Правила преобразования данных + Управляющая структура) + Структура данных

Таким образом, операционное знание (алгоритм) и фактуальное знание (структура данных) неотделимы друг от друга. Однако если в ходе эксплуатации ИС выяснится потребность в модификации одного из двух компонентов программы, то возникнет необходимость ее переписывания. Это объясняется тем, что полным знанием проблемной области обладает только разработчик ИС, а программа служит “недумающим исполнителем” знания разработчика. Конечный же пользователь вследствие процедурности и машинной ориентированности представления знаний понимает

лишь внешнюю сторону процесса обработки данных и никак не может на него влиять.

Следствием перечисленных недостатков является плохая жизнеспособность ИС или неадаптивность к изменениям информационных потребностей. Кроме того, в силу детерминированности алгоритмов решаемых задач ИС не способна к формированию у пользователя знания о действиях в не полностью определенных ситуациях.

В системах, основанных на обработке баз данных (СБД - Data Base Systems), происходит отделение фактуального и операционного знаний друг от друга. Первое организуется в виде базы данных, второе - в виде программ. Причем программа может автоматически генерироваться по запросу пользователя (например, реализация SQL или QBE запросов). В качестве посредника между программой и базой данных выступает программный инструмент доступа к данным - система управления базой данных (СУБД):

СБД = Программа \Leftrightarrow СУБД \Leftrightarrow База данных

Концепция независимости программ от данных позволяет повысить гибкость ИС по выполнению произвольных информационных запросов. Однако эта гибкость в силу процедурности представления операционного знания имеет четко определенные границы. Для формулирования информационного запроса пользователь должен ясно представлять себе структуру базы данных и до определенной степени алгоритм решения задачи. Следовательно, пользователь должен достаточно хорошо разбираться в проблемной области, в логической структуре базы данных и алгоритме программы. Концептуальная схема базы данных выступает в основном только в роли промежуточного звена в процессе отображения логической структуры данных на структуру данных прикладной программы.

Общие недостатки традиционных информационных систем, к которым относятся системы первых двух типов, заключаются в слабой адаптивности к изменениям в предметной области и информационным потребностям пользователей, в невозможности решать плохо формализуемые задачи, с которыми управленческие работники постоянно имеют дело. Перечисленные недостатки

устраняются в интеллектуальных информационных системах (ИИС).

Анализ структуры программы показывает возможность выделения из программы операционного знания (правил преобразования данных) в так называемую базу знаний, которая в декларативной форме хранит общие для различных задач единицы знаний. При этом управляющая структура приобретает характер универсального механизма решения задач (механизма вывода), который связывает единицы знаний в исполняемые цепочки (генерируемые алгоритмы) в зависимости от конкретной постановки задачи (сформулированной в запросе цели и исходных условий). Такие ИС становятся системами, основанными на обработке знаний (СБЗ - Knowledge Base (Based) Systems):

СБЗ = База знаний \Leftrightarrow Управляющая структура \Leftrightarrow База данных
(Механизм вывода)

Следующим шагом в развитии интеллектуальных информационных систем является выделение в самостоятельную подсистему или репозиторий метазнания, описывающего структуру операционного и фактуального знания и отражающего модель проблемной области. В таких системах и программы, и структуры данных генерируются или komponуются из единиц знаний, описанных в репозитории, каждый раз при изменении модели проблемной области. Будем называть ИИС, обрабатывающие метазнание, системами, основанными на моделях (СБМ - Model Based Systems):

СБМ = Репозиторий \Leftrightarrow Механизм \Rightarrow Информационная
(Онтология - генерации система
Модель проблемной или
области) конфигурации

Для интеллектуальных информационных систем, ориентированных на генерацию алгоритмов решения задач, характерны следующие признаки:

- развитые коммуникативные способности,
- умение решать сложные плохо формализуемые задачи,

- способность к самообучению,
- адаптивность.

Коммуникативные способности ИИС характеризуют способ взаимодействия (интерфейса) конечного пользователя с системой, в частности, возможность формулирования произвольного запроса в диалоге с ИИС на языке, максимально приближенном к естественному.

Сложные плохо формализуемые задачи - это задачи, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, для которой могут быть характерны неопределенность и динамичность исходных данных и знаний.

Способность к самообучению - это возможность автоматического извлечения знаний для решения задач из накопленного опыта конкретных ситуаций.

Адаптивность - способность к развитию системы в соответствии с объективными изменениями модели проблемной области.

В различных ИИС перечисленные признаки интеллектуальности развиты в неодинаковой степени и редко, когда все четыре признака реализуются одновременно. Условно каждому из признаков интеллектуальности соответствует свой класс ИИС (рис. 1.2):

- Системы с интеллектуальным интерфейсом;
- Экспертные системы;
- Самообучающиеся системы;
- Адаптивные системы.

Все четыре признака интеллектуальности в той или иной степени реализуются в системах управления знаниями.

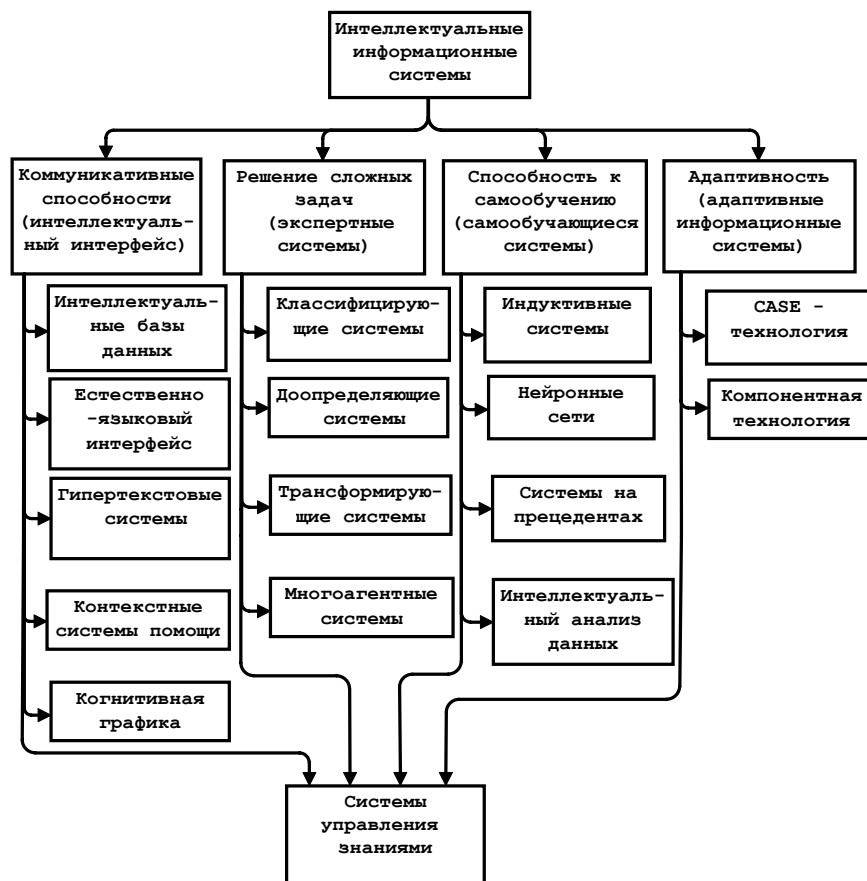


Рис. 1.2. Классификация интеллектуальных информационных систем

1.2. Системы с интеллектуальным интерфейсом

Интеллектуальные базы данных отличаются от обычных баз данных возможностью выборки по запросу необходимой информации, которая может явно не храниться, а выводиться из имеющихся в базе данных. Вывод неявной информации осуществляется путем интерпретации следующих зависимостей:

- вычислительных зависимостей атрибутов, например, «вывести список товаров, цена которых выше среднеотраслевой»,
- структурных отношений объектов, например, «вывести список товаров-заменителей некоторой продукции»,
- логических зависимостей факторов принятия решений, например, «вывести список потенциальных покупателей некоторого товара».

Для выполнения первого типа запроса необходимо сначала проведение статистического расчета среднеотраслевой цены по всей базе данных, а уже после этого собственно отбор данных. Для выполнения второго типа запроса необходимо вывести значения характерных признаков объекта, а затем поиск по ним аналогичных объектов. Для третьего типа запроса требуется сначала определить список посредников-продавцов, выполняющих продажу данного товара, а затем провести поиск связанных с ними покупателей.

Во всех перечисленных типах запросов требуется осуществить поиск по условию, которое должно быть доопределено в ходе решения задачи. Интеллектуальная система без помощи пользователя по структуре базы данных сама строит путь доступа к файлам данных. Формулирование запроса осуществляется в диалоге с пользователем, последовательность шагов которого выполняется в максимально удобной для пользователя форме. Запрос к базе данных может формулироваться и с помощью естественно-языкового интерфейса.

Естественно-языковой интерфейс предполагает трансляцию естественно-языковых конструкций на внутримашинный уровень представления знаний. Для этого необходимо решать задачи морфологического, синтаксического и семантического анализа и синтеза высказываний на естественном языке. Так, морфологический анализ предполагает распознавание и проверку правильности написания слов по словарям, синтаксический контроль - разложение входных сообщений на отдельные компоненты (определение структуры) с проверкой соответствия грамматическим правилам внутреннего представления знаний и выявления недостающих частей и, наконец, семантический анализ - установление смысловой правильности синтаксических конструкций. Синтез высказываний решает обратную задачу преобразования внутреннего представления информации в естественно-языковое.

Естественно-языковой интерфейс используется для:

- доступа к интеллектуальным базам данных;
- контекстного поиска документальной текстовой информации;
- голосового ввода команд в системах управления;
- машинного перевода с иностранных языков.

Гипертекстовые системы предназначены для реализации поиска по ключевым словам в базах текстовой информации. Интеллектуальные гипертекстовые системы отличаются возможностью более сложной семантической организации ключевых слов, которая отражает различные смысловые отношения терминов. Таким образом, механизм поиска работает прежде всего с базой знаний ключевых слов, а уже затем непосредственно с текстом. В более широком плане сказанное распространяется и на поиск мультимедийной информации, включающей помимо текстовой и цифровой информации графические, аудио и видео-образы.

Системы контекстной помощи можно рассматривать, как частный случай интеллектуальных гипертекстовых и естественно-языковых систем. В отличие от обычных систем помощи, навязывающих пользователю схему поиска требуемой информации, в системах контекстной помощи пользователь описывает проблему (ситуацию), а система с помощью дополнительного диалога ее конкретизирует и сама выполняет поиск относящихся к ситуации рекомендаций. Такие системы относятся к классу систем распространения знаний (Knowledge Publishing) и создаются как приложение к системам документации (например, технической документации по эксплуатации товаров).

Системы когнитивной графики позволяют осуществлять интерфейс пользователя с ИИС с помощью графических образов, которые генерируются в соответствии с происходящими событиями. Такие системы используются в мониторинге и управлении оперативными процессами. Графические образы в наглядном и интегрированном виде описывают множество параметров изучаемой ситуации. Например, состояние сложного управляемого объекта отображается в виде человеческого лица, на котором каждая черта отвечает за какой-либо параметр, а общее выражение лица дает интегрированную характеристику ситуации.

Системы когнитивной графики широко используются также в обучающих и тренажерных системах на основе использования принципов виртуальной реальности, когда графические образы моделируют ситуации, в которых обучающемуся необходимо принимать решения и выполнять определенные действия.

1.3. Экспертные системы

Назначение экспертных систем заключается в решении достаточно трудных для экспертов задач на основе накапливаемой базы знаний, отражающей опыт работы экспертов в рассматриваемой проблемной области. Достоинство применения экспертных систем заключается в возможности принятия решений в уникальных ситуациях, для которых алгоритм заранее не известен и формируется по исходным данным в виде цепочки рассуждений (правил принятия решений) из базы знаний. Причем решение задач предполагается осуществлять в условиях неполноты, недостоверности, многозначности исходной информации и качественных оценок процессов.

Экспертная система является инструментом, усиливающим интеллектуальные способности эксперта, и может выполнять следующие роли:

- консультанта для неопытных или непрофессиональных пользователей;
- ассистента в связи с необходимостью анализа экспертом различных вариантов принятия решений;
- партнера эксперта по вопросам, относящимся к источникам знаний из смежных областей деятельности.

Экспертные системы используются во многих областях, среди которых лидирует сегмент приложений в бизнесе (рис. 1.3) [39].

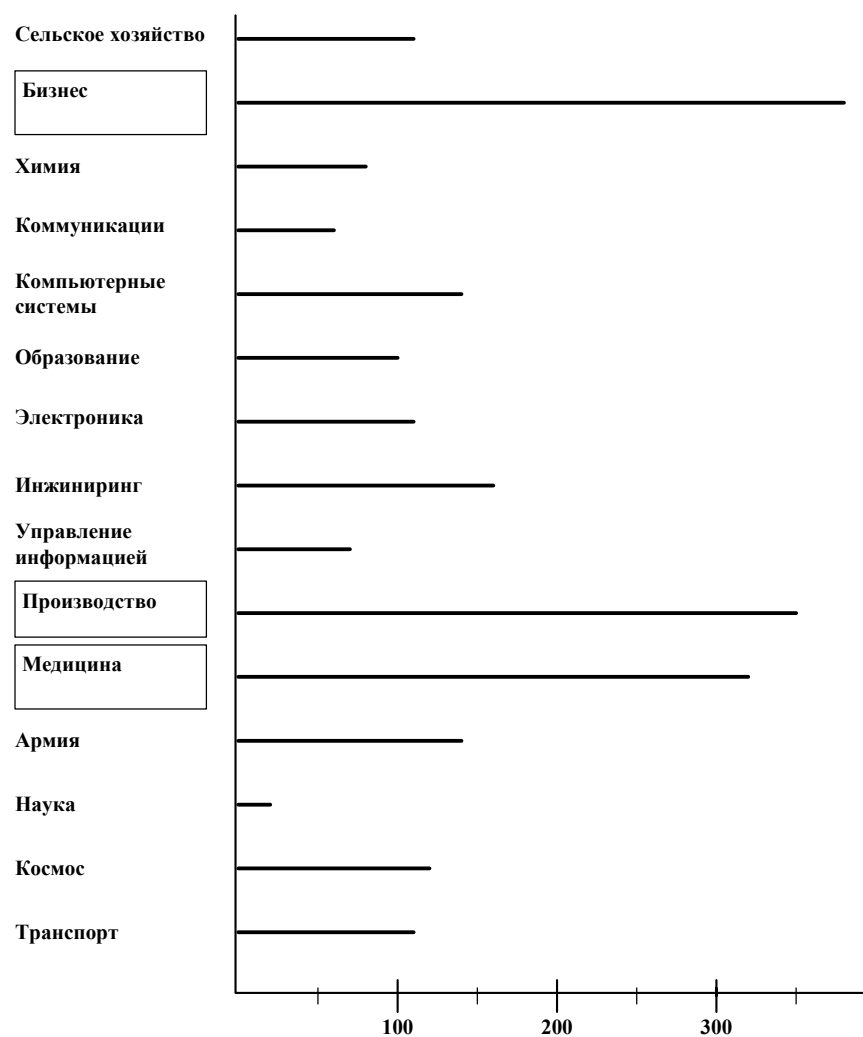


Рис.1.3. Области применения экспертных систем

Архитектура экспертной системы [1,5,7,14,21,31,37] включает в себя два основных компонента: базу знаний (хранилище единиц знаний) и программный инструмент доступа и обработки знаний, состоящий из механизмов вывода заключений (решения),

приобретения знаний, объяснения получаемых результатов и интеллектуального интерфейса (рис.1.4). Причем центральным компонентом экспертной системы является база знаний, которая выступает по отношению к другим компонентам как содержательная подсистема, составляющая основную ценность. “Know-how” базы знаний хорошей экспертной системы оценивается в *сотни тысяч долларов*, в то время как программный инструментарий - в *тысячи или десятки тысяч долларов*.



Рис. 1.4. Архитектура экспертной системы

База знаний - это совокупность единиц знаний, которые представляют собой формализованное с помощью некоторого метода представления знаний отражение объектов проблемной области и их взаимосвязей, действий над объектами и, возможно, неопределенностей, с которыми эти действия осуществляются.

В качестве методов представления знаний чаще всего используются либо правила, либо объекты (фреймы), либо их комбинация. Так, правила представляют собой конструкции:

Если <условие>

То <заключение> CF (Фактор определенности) <значение>

В качестве факторов определенности (CF), как правило, выступают либо условные вероятности байесовского подхода (от 0 до 1), либо коэффициенты уверенности нечеткой логики (от 0 до 100). Примеры правил имеют следующий вид:

Правило 1: Если Коэффициент рентабельности > 0.2

То Рентабельность = "удовл." CF 100

Правило 2: Если Задолженность = "нет" и

Рентабельность = "удовл."

То Финансовое_сост. = "удовл." CF 80

Правило 3: Если Финансовое_сост. = "удовл." и Репутация="удовл."

То Надежность предприятия = "удовл." CF 90

Фреймы представляют собой совокупность атрибутов, описывающих свойства и отношения с другими фреймами. В отличие от записей баз данных каждый фрейм имеет уникальное имя. Часть атрибутов отражают типизированные отношения, такие как "род - вид" (super-class - sub-class), "целое - часть" и др. Вместо конкретных значений атрибутов объектов могут задаваться значения по умолчанию (указатель наследования атрибутов устанавливается в S), присущие целым классам объектов, или присоединенные процедуры (process). Пример фреймов представлен на рис. 1.5.

Интеллектуальный интерфейс. Обмен данными между конечным пользователем и ЭС выполняет программа интеллектуального интерфейса, которая воспринимает сообщения пользователя и преобразует их в форму представления базы знаний и, наоборот, переводит внутреннее представление результата обработки в формат пользователя и выдает сообщение на требуемый носитель. Важнейшим требованием к организации диалога пользователя с ЭС является естественность, которая не означает буквально формулирование потребностей пользователя

предложениями естественного языка, хотя это и не исключается в ряде случаев. Важно, чтобы последовательность решения задачи была гибкой, соответствовала представлениям пользователя и велась в профессиональных терминах.

ПРЕДПРИЯТИЕ ОТРАСЛИ#1

Имя слота	Указатель наследования	Тип	Значение
Super-class	U	FRAME	ROOT
Sub-class	U	FRAME	Предприятие
Код предприятия	U	String	101
Код отрасли	U	String	123
Отраслевой коэффициент рентабельности	U	Real	20

ПРЕПРИЯТИЕ#1

Имя слота	Указатель наследования	Тип	Значение
Super-class	S	FRAME	Предприятие отрасли
Sub-class	-	-	-
Код предприятия	S	String	101
Код отрасли	S	String	123
Отраслевой коэффициент рентабельности	S	Real	20
Коэффициент рентабельности		Real	25
Задолженность		String	Нет
Репутация		String	Удовл
Финансовое состояние		Process	Fin_sost
Надежность		Process	Nad

Рис. 1.5. Фреймовое представление знаний

Механизм вывода. Этот программный инструмент получает от интеллектуального интерфейса преобразованный во внутреннее представление запрос, формирует из базы знаний конкретный алгоритм решения задачи, выполняет алгоритм, а полученный результат предоставляется интеллектуальному интерфейсу для выдачи ответа на запрос пользователя.

В основе использования любого механизма вывода лежит процесс нахождения в соответствии с поставленной целью и описанием конкретной ситуации (исходных данных) относящихся к решению единиц знаний (правил, объектов, прецедентов и т.д.) и связыванию их при необходимости в цепочку рассуждений, приводящую к определенному результату. Для представления знаний в форме правил это может быть прямая (рис. 1.6) или обратная (рис. 1.7) цепочка рассуждений.

Для фреймового (объектно-ориентированного) представления знаний характерно применение механизма наследования атрибутов, когда значения атрибутов передаются по иерархии от вышестоящих классов к нижестоящим (например, на рис.1.5. код отрасли, отраслевой коэффициент рентабельности). Также при заполнении атрибутов фрейма необходимыми данными запускаются на выполнение присоединенные процедуры.



Рис. 1.6. Прямая цепочка рассуждений

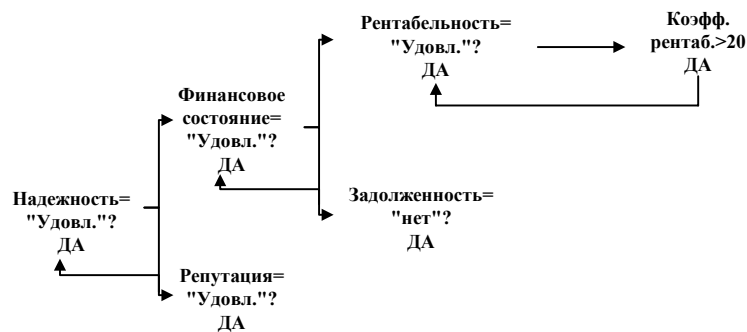


Рис. 1.7. Обратная цепочка рассуждений

Механизм объяснения. В процессе или по результатам решения задачи пользователь может запросить объяснение или обоснование хода решения. С этой целью ЭС должна предоставить соответствующий механизм объяснения. Объяснительные способности ЭС определяются возможностью механизма вывода запоминать путь решения задачи. Тогда на вопросы пользователя "Как?" и "Почему?" получено решение или запрошены те или иные данные система всегда может выдать цепочку рассуждений до требуемой контрольной точки, сопровождая выдачу объяснения заранее подготовленными комментариями. В случае отсутствия решения задач объяснение должно выдаваться пользователю автоматически. Полезно иметь возможность и гипотетического объяснения решения задачи, когда система отвечает на вопросы, что будет в том или ином случае.

Однако не всегда пользователя может интересовать полный вывод решения, содержащий множество ненужных деталей. В этом случае система должна уметь выбирать из цепочки только ключевые моменты с учетом их важности и уровня знаний пользователя. Для этого в базе знаний необходимо поддерживать модель знаний и намерений пользователя. Если же пользователь продолжает не понимать полученный ответ, то система должна быть способна в диалоге на основе поддерживаемой модели проблемных знаний обучать пользователя тем или иным фрагментам знаний, т.е. раскрывать более подробно отдельные понятия и зависимости, если даже эти детали непосредственно в выводе не использовались.

Механизм приобретения знаний. База знаний отражает знания *экспертов* (специалистов) в данной проблемной области о действиях в различных ситуациях или процессах решения характерных задач. Выявлением подобных знаний и последующим их представлением в базе знаний занимаются специалисты, называемые *инженерами знаний*. Для ввода знаний в базу и их последующего обновления ЭС должна обладать механизмом приобретения знаний. В простейшем случае это интеллектуальный редактор, который позволяет вводить единицы знаний в базу и проводить их синтаксический и семантический контроль, например, на непротиворечивость. В более сложных случаях механизм приобретения знаний позволяет извлекать знания в результате использования специальных сценариев интервьюирования экспертов, или из вводимых примеров реальных ситуаций, как в случае индуктивного вывода, или из текстов, или из опыта работы самой интеллектуальной системы.

Классы экспертных систем. По степени сложности решаемых задач экспертные системы можно классифицировать следующим образом:

- *По способу формирования решения* экспертные системы разделяются на два класса: аналитические и синтетические. Аналитические системы предполагают выбор решений из множества известных альтернатив (определение характеристик объектов), а синтетические системы - генерацию неизвестных решений (формирование объектов).
- *По способу учета временного признака* экспертные системы могут быть статическими или динамическими. Статические системы решают задачи при неизменяемых в процессе решения данных и знаниях, динамические системы допускают такие изменения. Статические системы осуществляют монотонное непрерываемое решение задачи от ввода исходных данных до конечного результата, динамические системы предусматривают возможность пересмотра в процессе решения полученных ранее результатов и данных.
- *По видам используемых данных и знаний* экспертные системы классифицируются на системы с детерминированными (четко определенными) знаниями и неопределенными знаниями. Под

неопределенностью знаний (данных) понимается их неполнота (отсутствие), недостоверность (неточность измерения), двусмысленность (многозначность понятий), нечеткость (качественная оценка вместо количественной).

- По числу используемых источников знаний экспертные системы могут быть построены с использованием одного или множества источников знаний. Источники знаний могут быть альтернативными (множество миров) или дополняющими друг друга (кооперирующими).

В соответствии с перечисленными признаками классификации, как правило, выделяются следующие четыре основные класса экспертных систем (рис. 1.8)

	Анализ	Синтез	
Детерминированность знаний	Классифицирующие	Трансформирующие	Один источник знаний
Неопределенность знаний	Доопределяющие	Многоагентные	Множество источн. знаний
	Статика	Динамика	

Рис. 1.8. Классы экспертных систем

Классифицирующие экспертные системы. К аналитическим задачам прежде всего относятся задачи распознавания различных ситуаций, когда по набору заданных признаков (факторов) выявляется сущность некоторой ситуации, в зависимости от которой выбирается определенная последовательность действий. Таким образом, в соответствии с исходными условиями среди альтернативных решений находится одно, наилучшим образом удовлетворяющее поставленной цели и ограничениям.

Экспертные системы, решающие задачи распознавания ситуаций, называются классифицирующими, поскольку определяют принадлежность анализируемой ситуации к некоторому классу. В качестве основного метода формирования решений используется метод логического дедуктивного вывода от общего к частному, когда путем подстановки исходных данных в некоторую

совокупность взаимосвязанных общих утверждений получается частное заключение.

Доопределяющие экспертные системы. Более сложный тип аналитических задач представляют задачи, которые решаются на основе неопределенных исходных данных и применяемых знаний. В этом случае экспертная система должна как бы доопределять недостающие знания, а в пространстве решений может получаться несколько возможных решений с различной вероятностью или уверенностью в необходимости их выполнения. В качестве методов работы с неопределенностями могут использоваться байесовский вероятностный подход, коэффициенты уверенности, нечеткая логика. Доопределяющие экспертные системы могут использовать для формирования решения несколько источников знаний. В этом случае могут использоваться эвристические приемы выбора единиц знаний из их конфликтного набора, например, на основе использования приоритетов важности, или получаемой степени определенности результата, или значений функций предпочтений и т.д.

Для аналитических задач классифицирующего и доопределяющего типов характерны следующие *проблемные области*:

- Интерпретация данных - выбор решения из фиксированного множества альтернатив на базе введенной информации о текущей ситуации. Основное назначение - определение сущности рассматриваемой ситуации, выбор гипотез, исходя из фактов. Типичным примером является экспертная система анализа финансового состояния предприятия.
- Диагностика - выявление причин, приведших к возникновению ситуации. Требуется предварительная интерпретация ситуации с последующей проверкой дополнительных фактов, например, выявление факторов снижения эффективности производства.
- Коррекция - диагностика, дополненная возможностью оценки и рекомендации действий по исправлению отклонений от нормального состояния рассматриваемых ситуаций.

Трансформирующие экспертные системы. В отличие от аналитических статических экспертных систем синтезирующие динамические экспертные системы предполагают повторяющееся преобразование знаний в процессе решения задач, что связано с

характером результата, который нельзя заранее предопределить, а также с динамичностью самой проблемной области.

В качестве методов решения задач в трансформирующих экспертных системах используются разновидности гипотетического вывода:

- генерации и тестирования, когда по исходным данным осуществляется генерация гипотез, а затем проверка сформулированных гипотез на подтверждение поступающими фактами;
- предположений и умолчаний, когда по неполным данным подбираются знания об аналогичных классах объектов, которые в дальнейшем динамически адаптируются к конкретной ситуации в зависимости от ее развития;
- использование общих закономерностей (метауправления) в случае неизвестных ситуаций, позволяющих генерировать недостающее знание.

Многоагентные системы. Для таких динамических систем характерно распределенное решение задач несколькими программными агентами, каждый из которых обладает собственной базой знаний и механизмом вывода [30,32]. Программные агенты, как правило, выполняют поручения людей, субъектов решения задачи, и в этом смысле их заменяют. При этом они реагируют на события во внешней среде (реактивные агенты), обрабатывают ситуации, принимают решения, передают результаты решения задач пользователям и во внешнюю среду. Наиболее интеллектуальные (когнитивные) агенты способны обучаться и изменять правила своего поведения.

При совместном решении задач несколькими программными агентами образуются многоагентные системы (МАС), с централизованным или децентрализованным управлением. В первом случае МАС должна иметь, по крайней мере, один агент, который выполняет роль координатора (диспетчера), планирующего и контролирующего реализацию процессов. Во втором случае все агенты самостоятельны в своем поведении. Интеграция работы программных агентов и соответствующих источников знаний осуществляется на динамической основе путем обмена между ними получаемыми результатами, например, через "доску объявлений" (рис. 1.9).

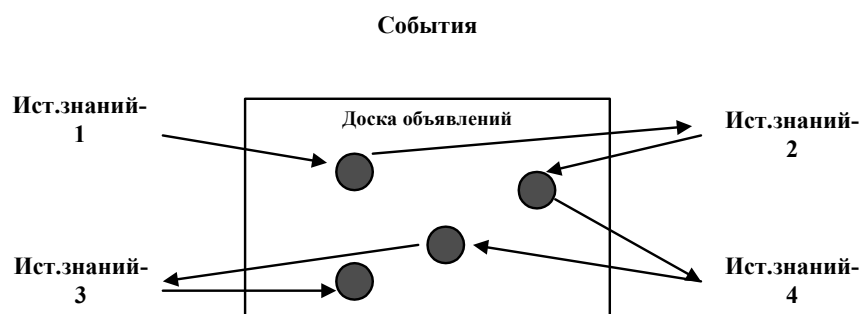


Рис. 1.9. “Доска объявлений”

Для многоагентных систем характерны следующие особенности:

- Проведение альтернативных рассуждений на основе использования различных источников знаний с механизмом устранения противоречий;
- Распределенное решение проблем, которые разбиваются на параллельно решаемые подпроблемы, соответствующие самостоятельным источникам знаний;
- Применение множества стратегий работы механизма вывода в зависимости от типа решаемой проблемы;
- Обработка больших массивов данных, содержащихся в базе данных, и способность к самообучению, изменяющая правила поведения агентов;
- Использование различных математических моделей и внешних процедур, хранимых в базе моделей;
- Способность прерывания решения задач в связи с необходимостью получения дополнительных данных и знаний от пользователей, моделей, параллельно решаемых подпроблем.

Для синтезирующих динамических экспертных систем наиболее применимы следующие *проблемные области*:

- Проектирование - определение конфигурации объектов с точки зрения достижения заданных критериев эффективности и ограничений, например, проектирование бюджета предприятия,

портфеля инвестиций, конфигурация изделий в электронной коммерции.

- Прогнозирование - предсказание последствий развития текущих ситуаций на основе математического и эвристического моделирования, например, прогнозирование трендов на биржевых торгах.
- Планирование - выбор последовательности действий пользователей по достижению поставленной цели, например, планирование цепочек поставки продукции (supply chain management).
- Диспетчирование - распределение работ во времени, составление расписаний, например, планирование графика потоков работ (workflow).
- Мониторинг - слежение за текущей ситуацией с возможной последующей коррекцией. Для этого выполняется диагностика, прогнозирование, а в случае необходимости планирование и коррекция действий пользователей, например, мониторинг сбыта готовой продукции.
- Управление - мониторинг, дополненный реализацией действий в автоматических системах, например, принятие решений на биржевых торгах.

По данным публикации [39], в которой проводится анализ 12500 действующих экспертных систем, распределение экспертных систем по проблемным областям имеет следующий вид (рис. 1.10):

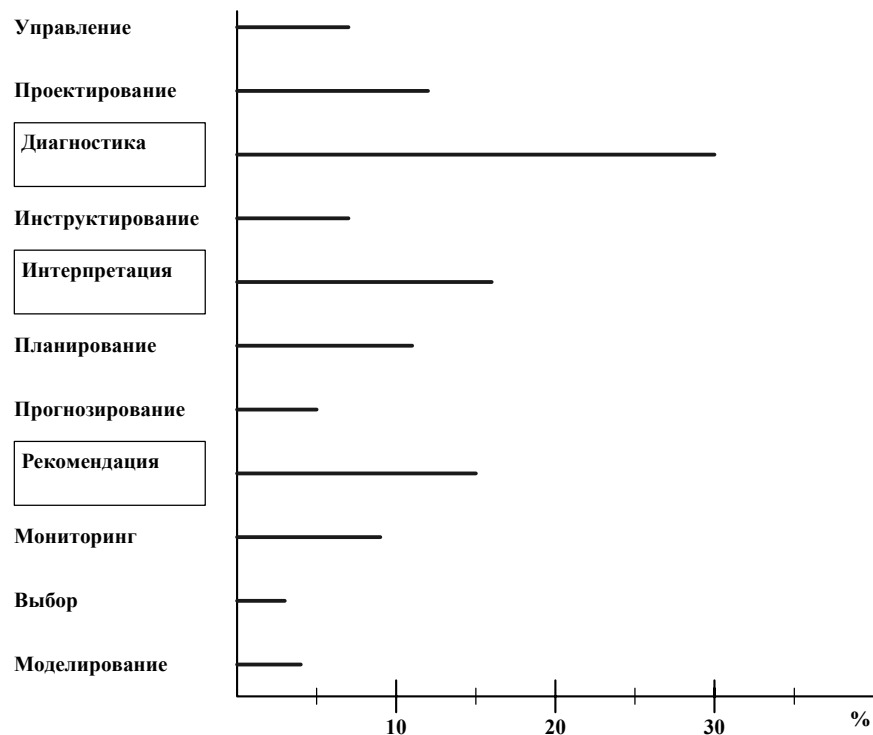


Рис. 1.10. Проблемные области экспертных систем

1.4. Самообучающиеся системы

В основе самообучающихся систем лежат методы автоматической классификации примеров ситуаций реальной практики (обучения на примерах). Примеры реальных ситуаций накапливаются за некоторый исторический период и составляют *обучающую выборку*. Эти примеры описываются множеством признаков классификации. Причем обучающая выборка может быть:

- “с учителем”, когда для каждого примера задается в явном виде значение признака его принадлежности некоторому классу ситуаций (классообразующего признака);

- “без учителя”, когда по степени близости значений признаков классификация система сама выделяет классы ситуаций.

В результате обучения системы автоматически строятся обобщенные правила или функции, определяющие принадлежность ситуаций классам, которыми обученная система пользуется при интерпретации новых возникающих ситуаций. Таким образом, автоматически формируется база знаний, используемая при решении задач классификации и прогнозирования. Эта база знаний периодически автоматически корректируется по мере накопления опыта реальных ситуаций, что позволяет сократить затраты на ее создание и обновление.

Общие недостатки, свойственные всем самообучающимся системам, заключаются в следующем:

- возможна неполнота и/или зашумленность (избыточность) обучающей выборки и, как следствие, относительная адекватность базы знаний возникающим проблемам;
- возникают проблемы, связанные с плохой смысловой ясностью зависимостей признаков и, как следствие, неспособность объяснения пользователям получаемых результатов;
- ограничения в размерности признакового пространства вызывают неглубокое описание проблемной области и узкую направленность применения.

Индуктивные системы. Обобщение примеров по принципу от частного к общему сводится к выявлению подмножеств примеров, относящихся к одним и тем же подклассам, и определению для них значимых признаков.

Процесс классификации примеров осуществляется следующим образом:

1. Выбирается признак классификации из множества заданных (либо последовательно, либо по какому-либо правилу, например, в соответствии с максимальным числом получаемых подмножеств примеров);
2. По значению выбранного признака множество примеров разбивается на подмножества;
3. Выполняется проверка, принадлежит ли каждое образовавшееся подмножество примеров одному подклассу;
4. Если какое-то подмножество примеров принадлежит одному подклассу, т.е. у всех примеров подмножества совпадает

значение классообразующего признака, то процесс классификации заканчивается (при этом остальные признаки классификации не рассматриваются);

5. Для подмножеств примеров с несовпадающим значением классообразующего признака процесс классификации продолжается, начиная с пункта 1. (Каждое подмножество примеров становится классифицируемым множеством).

Процесс классификации может быть представлен в виде дерева решений, в котором в промежуточных узлах находятся значения признаков последовательной классификации, а в конечных узлах - значения признака принадлежности определенному классу. Пример построения дерева решений на основе фрагмента таблицы примеров (таблица 1.1) показан на рис. 1.11.

Таблица 1.1

Классообр. признак	Признаки классификации			
Цена	Спрос	Конкуренция	Издержки	Качество
низкая	низкий	маленькая	маленькие	низкое
высокая	низкий	маленькая	большие	высокое
высокая	высокий	маленькая	большие	низкое
высокая	высокий	маленькая	маленькие	высокое
высокая	высокий	маленькая	маленькие	низкое
высокая	высокий	маленькая	большие	высокое

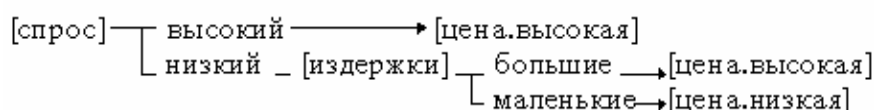


Рис. 1.11. Фрагмент дерева решений

Анализ новой ситуации сводится к выбору ветви дерева, которая полностью определяет эту ситуацию. Поиск решения осуществляется в результате последовательной проверки признаков классификации. Каждая ветвь дерева соответствует одному правилу решения:

Если Спрос=“низкий” и Издержки=“маленькие”
 То Цена=“низкая”

Примерами инструментальных средств, поддерживающих индуктивный вывод знаний, являются 1st Class (Programs in Motion), Rulemaster (Radian Corp.), ИЛИС (ArgusSoft), KAD (ИПС Переяславль-Залесский).

Нейронные сети. В результате обучения на примерах строятся математические решающие функции (передаточные функции или функции активации), которые определяют зависимости между входными (X_i) и выходными (Y_j) признаками (сигналами) [36] (рис. 1.12).

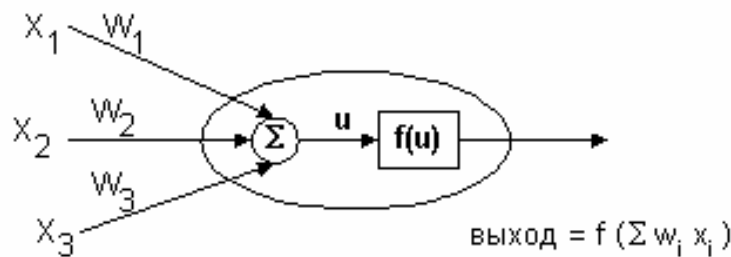


Рис.1.12. Решающая функция - “нейрон”

Каждая такая функция, называемая по аналогии с элементарной единицей человеческого мозга - нейроном, отображает зависимость значения выходного признака (Y) от взвешенной суммы (U) значений входных признаков (X_i), в которой вес входного признака (W_i) показывает степень влияния входного признака на выходной:

$$Y = f\left(\sum_i W_i * X_i\right)$$

Решающие функции используются в задачах классификации на основе сопоставления их значений при различных комбинациях значений входных признаков с некоторым пороговым значением. В случае превышения заданного порога считается, что нейрон сработал и таким образом распознал некоторый класс ситуаций. Нейроны используются и в задачах прогнозирования, когда по значениям входных признаков после их подстановки в выражение

решающей функции получается прогнозное значение выходного признака.

Функциональная зависимость может быть линейной, но, как правило, используется сигмоидальная форма, которая позволяет вычленять более сложные пространства значений выходных признаков. Такая функция называется логистической (рис.1.13).

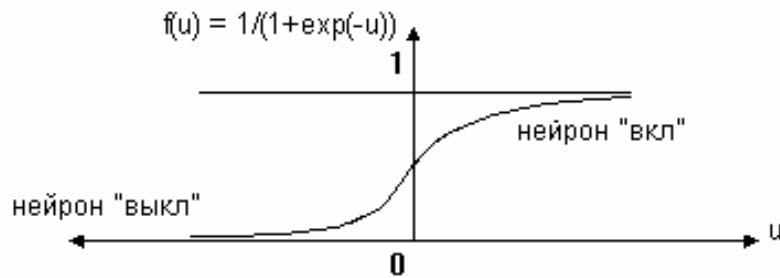


Рис.1.13. Логистическая (сигмоидальная) функция

Нейроны могут быть связаны между собой, когда выход одного нейрона является входом другого. Таким образом, строится нейронная сеть (рис. 1.14), в которой нейроны, находящиеся на одном уровне, образуют слои.

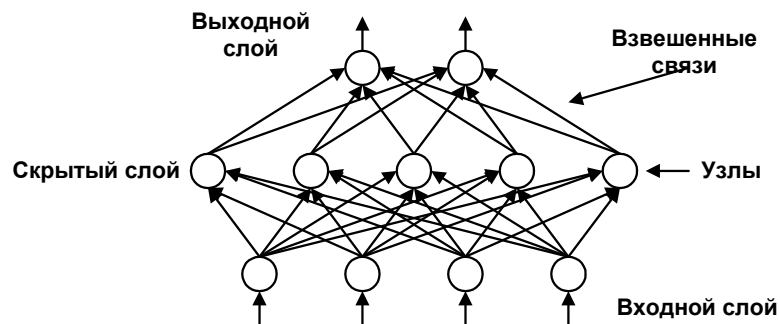


Рис.1.14. Нейронная сеть

Обучение нейронной сети сводится к определению связей (синапсов) между нейронами и установлению силы этих связей (весовых коэффициентов). Алгоритмы обучения нейронной сети упрощенно сводятся к определению зависимости весового коэффициента связи двух нейронов от числа примеров, подтверждающих эту зависимость.

Наиболее распространенным алгоритмом обучения нейронной сети является алгоритм обратного распространения ошибки. Целевая функция по этому алгоритму должна обеспечить минимизацию квадрата ошибки в обучении по всем примерам:

$$\min \sum_i (T_i - Y_i)^2, \text{ где}$$

T_i - заданное значение выходного признака по i - му примеру;

Y_i - вычисленное значение выходного признака по i - му примеру.

Сущность алгоритма обратного распространения ошибки сводится к следующему:

1. Задать произвольно небольшие начальные значения весов связей нейронов.
2. Для всех обучающих пар “значения входных признаков - значение выходного признака” (примеров из обучающей выборки) вычислить выход сети (Y).
3. Выполнить рекурсивный алгоритм, начиная с выходных узлов по направлению к первому скрытому слою, пока не будет достигнут минимальный уровень ошибки.

Вычислить веса на $(t+1)$ шаге по формуле:

$$W_{ij}(t+1) = W_{ij}(t) + \eta \delta_i X_i, \text{ где}$$

$W_{ij}(t)$ - вес связи от скрытого i -го нейрона или от входа к j -му нейрону на шаге t ;

X_i - выходное значение i -го нейрона;

η - коэффициент скорости обучения;

δ_i - ошибка для j -го нейрона.

Если j -й нейрон - выходной, то

$$\delta_i = Y_i(1 - Y_i)(T_i - Y_i)$$

Если j -й нейрон находится в скрытом внутреннем слое, то

$$\delta_j = X_j(1 - X_j) \sum_k \delta_k W_{jk}, \text{ где}$$

k - индекс всех нейронов в слое, расположенном вслед за слоем с j -м нейроном.

Выполнить шаг 2.

Достоинство нейронных сетей перед индуктивным выводом заключается в решении не только классифицирующих, но и прогнозных задач. Возможность нелинейного характера функциональной зависимости выходных и входных признаков позволяет строить более точные классификации.

Сам процесс решения задач в силу проведения матричных преобразований проводится очень быстро. Фактически имитируется параллельный процесс прохода по нейронной сети в отличие от последовательного в индуктивных системах. Нейронные сети могут быть реализованы и аппаратно в виде нейрокомпьютеров с ассоциативной памятью.

Последнее время нейронные сети получили стремительное развитие и очень активно используются в финансовой области. В качестве примеров внедрения нейронных сетей можно назвать:

- "Система прогнозирования динамики биржевых курсов для Chemical Bank" (фирма Logica);
- "Система прогнозирования для Лондонской фондовой биржи" (фирма SearchSpace);
- "Управление инвестициями для Mellon Bank" (фирма NeuralWare) и др.

В качестве инструментальных средств разработки нейронных сетей следует выделить инструментальные средства NeuroSolution, NeuralWorks Professional II/Plus, Process Advisor, NeuroShell 2, BrainMaiker Pro, NeurOn-line и др.

Системы, основанные на прецедентах (Case-based reasoning).

В этих системах база знаний содержит описания не обобщенных ситуаций, а собственно сами ситуации или прецеденты. Тогда поиск решения проблемы сводится к поиску по аналогии (выводу от частного к частному):

1. Получение подробной информации о текущей проблеме;
2. Сопоставление полученной информации со значениями признаков прецедентов из базы знаний;

3. Выбор прецедента из базы знаний, наиболее близкого к рассматриваемой проблеме;
4. В случае необходимости выполняется адаптация выбранного прецедента к текущей проблеме;
5. Проверка корректности каждого полученного решения;
6. Занесение детальной информации о полученном решении в базу знаний.

Так же как и для индуктивных систем прецеденты описываются множеством признаков, по которым строятся индексы быстрого поиска. Но в отличие от индуктивных систем допускается поиск с получением множества допустимых альтернатив, каждая из которых оценивается некоторой мерой схожести с анализируемой ситуацией. Обычно в качестве меры схожести двух прецедентов принимается функция от взвешенной суммы нормализованных на некоторой общей относительной шкале отклонений значений совпадающих признаков прецедентов. Формально полная схожесть SIM между прецедентами A и B , описанными p признаками, может быть выражена:

$$SIM(A, B) = F(Sim_1(a_1, b_1), Sim_2(a_2, b_2), \dots, Sim_p(a_p, b_p)), \text{ где}$$

$Sim_i(a_i, b_i)$ – локальная схожесть (отклонение) значений i -го признака двух прецедентов A и B , нормализованная на шкале $[0, 1]$

В качестве функций полной схожести F могут использоваться следующие функции:

$SIM(A, B) = \sqrt{\frac{1}{p} \sum_{i=1}^p [Sim_i(a_i, b_i)]^2}$	Евклидова расстояния
$SIM(A, B) = \sqrt[r]{\frac{1}{p} \sum_{i=1}^p [Sim_i(a_i, b_i)]^r}$	Минковского
$SIM(A, B) = \sqrt[r]{\sum_{i=1}^p \omega_i [Sim_i(a_i, b_i)]^r}$	Минковского с весами
$SIM(A, B) = \max_i \omega_i Sim_i(a_i, b_i)$	Максимум

где $p > 0, \omega_i > 0$ - p количество признаков, ω_i - относительный вес такой, что $\sum \omega_i = 1$.

Далее наиболее подходящие решения адаптируются по специальным алгоритмам к реальным ситуациям. В качестве методов адаптации в основном применяют методы повторной конкретизации переменных, уточнения параметров, замены одних компонентов решения другими [5]. Обучение системы сводится к запоминанию каждой новой обработанной ситуации с принятыми решениями в базе прецедентов.

Наиболее известными инструментами разработки приложений, использующих поиск прецедентов, являются: CBR-Express (Inference), REMIND (Cognitive Systems), ReCall (Isoft S.A.), KATE tools (Acknosoft), Pattern Recognition Workbench (Unica) и др. С помощью этих систем можно создать различные приложения для решения задач диагностики, анализа рисков, предсказания, контроля и обучения. Системы, основанные на прецедентах, применяются как системы распространения знаний с расширенными возможностями или как системы контекстной помощи. Например, в работе "горячей линии" центра технической поддержки компании Dell в Дублине используется система CBR-Express [28], которая помогает сотрудникам центра отвечать на большее число запросов по телефону. Благодаря этой системе производительность труда 200 сотрудников центра выросла до 3 тысячи звонков в день от пользователей. Пример подобной консультации может иметь следующий вид (рис. 1.15).

Описание ситуации (проблемы)

Не печатает принтер

Вопросы

Включено ли питание?	да
Прошло ли тестирование?	да
Замята ли бумага?	да
Подключен ли драйвер?	не знаю

Действия

Освободите бумагу	уверенность 80
Загрузите драйвер	уверенность 50

Рис. 1.15. Пример диалога с CBR-системой

Извлечение знаний на основе информационных хранилищ (Data mining based on Data Warehouse). В отличие от интеллектуальной базы данных информационное хранилище представляет собой хранилище извлеченной значимой информации из оперативных баз данных, которое предназначено, в первую очередь, для оперативного анализа данных (реализации OLAP - технологии) [29]. Формулирование запроса осуществляется в результате применения интеллектуального интерфейса, позволяющего в диалоге гибко определять значимые признаки анализа для произвольной группировки данных. Наиболее известными инструментальными средствами поддержки информационных хранилищ и OLAP - технологий являются такие инструментальные средства, как статистический пакет прикладных программ SAS, специализированные программы Business Objects, Oracle Express, отечественные программные продукты PolyAnalyst, Контур Стандарт и др.

Типичными задачами оперативного ситуационного анализа, решаемых на основе информационных хранилищ, являются:

- Определение профиля потребителей конкретного товара;
- Предсказание изменений ситуации на рынке;
- Анализ зависимостей признаков ситуаций (корреляционный анализ) и др.

Для решения перечисленных задач требуется применение методов извлечения знаний из баз данных (Data Mining или Knowledge Discovery), основанные на применении методов многомерного статистического анализа, индуктивных методов построения деревьев решений, нейронных сетей, генетических алгоритмов.

Рассмотрим сущность применения ранее не рассмотренного метода, основанного на применении генетических алгоритмов. Пусть требуется определить набор экономических показателей, которые в наибольшей степени влияют на положительную динамику поведения рынка. Тогда набор показателей можно рассматривать как набор хромосом, определяющих качества индивида, то есть решения поставленной задачи. Значения показателей, определяющих решение, при этом соответствуют генам.

Поиск оптимального решения поставленной задачи похож на эволюцию популяции индивидов, представленных их наборами хромосом [12]. В этой эволюции действуют три механизма: отбор сильнейших - наборов хромосом, которым соответствуют наиболее оптимальные решения; скрещивание - производство новых индивидов при помощи смешивания хромосомных наборов отобранных индивидов; и мутации - случайные изменения генов у некоторых индивидов популяции. В результате смены поколений, в конце концов, вырабатывается такое решение поставленной задачи, которое уже не может быть далее улучшено.

Генетические алгоритмы имеют ряд недостатков. Критерий отбора хромосом и сама процедура являются эвристическими и далеко не всегда гарантируют нахождения лучшего решения. Как и в реальной жизни, эволюция может остановиться на какой-либо непродуктивной ветви. С другой стороны можно подобрать примеры, когда из эволюции генетическим алгоритмом исключаются перспективные продолжения. Это особенно становится заметно при решении задач большой размерности со сложными внутренними связями. В качестве примера разработки систем на основе генетических алгоритмов можно привести систему GeneHunter фирмы Ward Systems Group.

Применение методов интеллектуального анализа на основе информационных хранилищ на практике все в большей степени демонстрирует необходимость интеграции интеллектуальных и

традиционных информационных технологий, комбинированное использование различных методов представления и вывода знаний, усложнение архитектуры информационных систем (см. параграф 1.6. и главу 7).

1.5. Адаптивные информационные системы

В условиях динамического развития экономических объектов возрастают требования к **адаптивности** информационных систем к изменениям. Эти требования сводятся к следующему:

- ИС в каждый момент времени должна адекватно поддерживать организацию бизнес-процессов.
- Реконструкция ИС должна проводиться всякий раз, как возникает потребность в реорганизации бизнес-процессов.
- Реконструкция ИС должна проводиться быстро и с минимальными затратами.

Учитывая высокую динамичность современных бизнес-процессов, можно сделать вывод о том, что адаптивность ИС немыслима без **интеллектуализации** ее архитектуры. Ядром адаптивной ИС является постоянно развиваемая **модель проблемной области (предприятия)**, поддерживаемая в специальной базе знаний - **репозитории**, на основе которого осуществляется генерация или конфигурация программного обеспечения. Таким образом, проектирование и адаптация ИС сводится, прежде всего, к построению модели проблемной области и ее своевременной корректировке. Отсюда адаптивную систему можно отнести к классу интеллектуальной информационной системы, основанной на модели проблемной области.

При проектировании информационной системы обычно используются два подхода: **оригинальное** или **типовое проектирование**. Первый подход предполагает разработку информационной системы “с чистого листа” в соответствии с требованиями экономического объекта, второй подход - адаптацию типовых разработок к особенностям экономического объекта. Первый подход, как правило, реализуется на основе применения **систем автоматизированного проектирования ИС** или **CASE-технологий**, например, таких как, Designer 2000 (Oracle), SilverRun (SilverRun Technology), Natural LightStorm (Software AG) и др.,

второй подход - на основе применения **систем компонентного проектирования ИС**, например, таких как R/3 (SAP), BAAN IV (Baan Corp), Галактика (Новый Атлант) и др.

С точки зрения адаптивности информационной системы к бизнес-процессам экономического объекта оба подхода ориентируются на предварительное тщательное изучение экономического объекта и его моделирование [10,29]. Отличие подходов заключается в следующем: при использовании CASE-технологии на основе репозитория при возникновении изменения выполняется каждый раз **генерация** (пересоздание) программного обеспечения, а при использовании компонентной технологии - **конфигурация** программ и только в редких случаях их переработка с помощью CASE - средств, например, использования языков четвертого поколения (4GL).

Для моделирования проблемной области и последующих конфигураций информационной системы из отдельных компонентов (программных модулей) используется специальный программный инструментарий, например, R/3 Business Engineer и BAAN DEM (Dynamic Enterprise Modeler). Несомненным достоинством применения модельно-ориентированных компонентных систем, таких как R/3 или BAAN IV, перед CASE - технологиями является накопление опыта проектирования информационных систем для различных отраслей и типов производства в виде **типовых** моделей или так называемых **референтных/ссылочных (reference) моделей**, которые поставляются вместе с программным продуктом в форме наполненного репозитория. Таким образом, вместе с программным продуктом пользователи приобретают базу знаний «know-how» об эффективных методах организации и управления бизнес-процессами, которые можно адаптировать в соответствии со спецификой конкретного экономического объекта.

В обобщенном виде конфигурация адаптивных информационных систем на основе компонентной технологии [29] представлена на рис. 1.16.

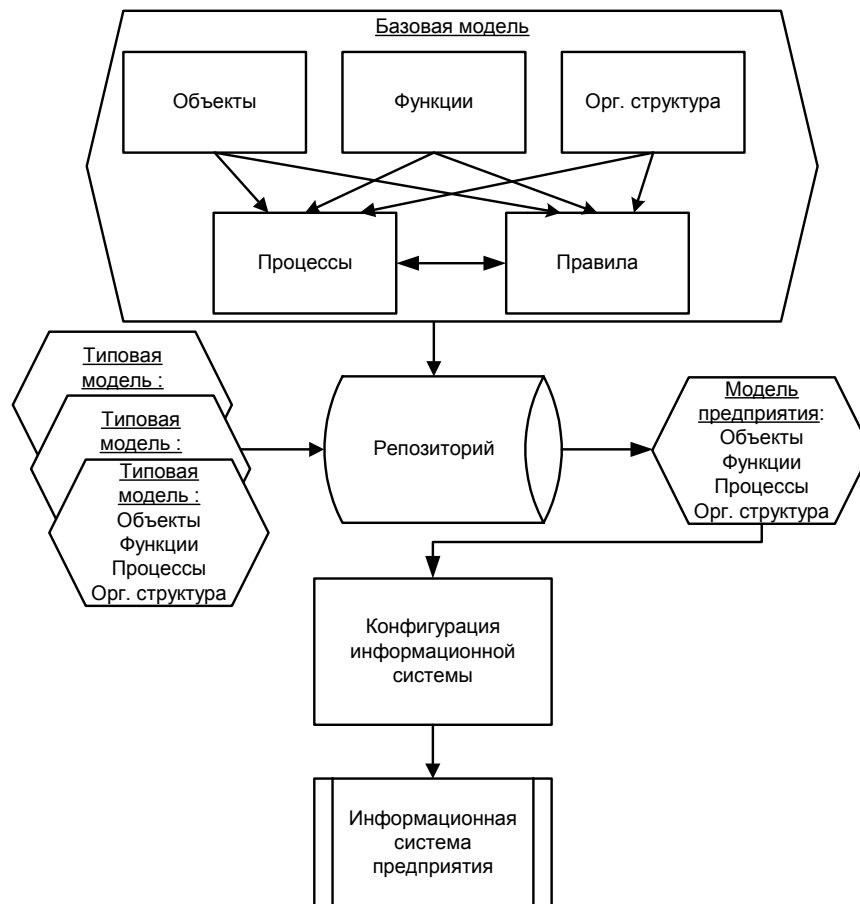


Рис. 1.16. Конфигурация адаптивной информационной системы на основе компонентной технологии

Базовая модель репозитория содержит описание *объектов, функций (операций), процессов (совокупности операций)*, которые реализуются в программных модулях компонентной системы. При этом большое значение в базовой модели имеет задание **правил (бизнес-правил)** поддержания целостности информационной системы, которые устанавливают условия проверки корректности совместного применения операций бизнес-процессов и поддерживающих их программных модулей. Таким образом,

многообразие и гибкость определения бизнес-процессов и соответствующих конфигураций информационной системы задается с помощью набора бизнес-правил.

Типовые модели соответствуют типовым конфигурациям информационной системы, выполненным для определенных отраслей (автомобильная, электронная, нефтегазовая и т.д.) или типов производства (индивидуальное, серийное, массовое, непрерывное и т.д.).

Модель предприятия (проблемной области) строится либо путем привязки или копирования фрагментов основной или типовой моделей в соответствии со специфическими особенностями предприятия, например, как в инструментальном средстве BAAN Orgware, либо в результате просмотра этих моделей и экспертного опроса, как в инструментальном средстве R/3 Business Engineer. Причем в последнем случае пользователю предлагается определить значения не всех параметров, а только тех, которые связаны между собой в контексте диалога и описаны бизнес-правилами.

Сформированная модель предприятия в виде метаописания хранится в репозитории и при необходимости может быть откорректирована. Далее по сформированной модели предприятия автоматически осуществляется конфигурация информационной системы, в ходе которой выполняется семантический контроль по соответствующим бизнес-правилам.

Недостатками описанной схемы конфигурации информационной системы является отсутствие средств оценки модели предприятия. Для того чтобы можно было выбирать оптимальные варианты конфигурации информационной системы, как правило, используются средства экспорта модели во внешние системы моделирования. Так, для системы R/3 предусмотрен экспорт (импорт) моделей в (из) среду(ы) инструментального средства ARIS Toolset, который позволяет осуществлять функционально-стоимостной анализ эффективности моделируемых бизнес-процессов и их динамическое имитационное моделирование.

1.6. Системы управления знаниями

Непрерывные изменения, происходящие в экономике, диктуют необходимость постоянного обновления знаний предприятий и организаций, как интеллектуального капитала, обеспечивающего устойчивые стратегические позиции предприятий на рынке. По выражению Мильнера Б.З. «формируется новая функция управления, в задачу которой входит аккумуляция интеллектуального капитала, выявление и распространение имеющейся информации и опыта, создание предпосылок для распространения и передачи знаний. Именно знания становятся источником высокой производительности, инноваций и конкурентных преимуществ» [16]. Новая функция управления знаниями реализуется как совокупность процессов систематического приобретения, синтеза, обмена и использования знаний внутри организации. Для управления знаниями характерно коллективное формирование и использование как внутренних, так и внешних источников знаний (информационных ресурсов).

Согласно результатам опроса руководителей компаний, входящих в список Fortune 1000, 97% менеджеров заявили, что существуют процессы, играющие для компании определяющую роль, и которые могли бы быть значительно усовершенствованы, если бы только о них знали больше сотрудников. В том же опросе 87% его участников утверждают, что дорогостоящие ошибки возникают именно потому, что служащие вовремя не получили необходимую информацию [2].

Качество используемых знаний непосредственно влияет на эффективность следующих деловых процессов:

- Принятие управленческих решений в стратегическом, тактическом и оперативном управлении в результате получения своевременного доступа к релевантным знаниям;
- Инновационная деятельность за счет возможности коллективного формирования идей и сокращения затрат на дублирование работ, обеспечивающая ускорение инновационного цикла;
- Непрерывное повышение квалификации работников предприятий в режиме реального времени.

- Предоставление партнерам (поставщикам, подрядчика, клиентам) в дополнение к своим основным услугам доступа к накопленным знаниям, включая консалтинг и обучение.

Под *системой управлением знаниями (СУЗ)* будем понимать совокупность организационных процедур, организационных подразделений (служб управления знаниями) и компьютерных технологий, которые обеспечивают интеграцию разнородных источников знаний и их коллективное использование в деловых процессах.

Согласно данным Попова Э.В., мировые расходы на создание и эксплуатацию СУЗ, составившие в 1999 году 2 млрд. долл., увеличатся до 12 млрд. долл. к 2003 году, в которых 7,9 млрд. долл. будут связаны с оказанием услуг, 1,8 млрд. долл. – с программным обеспечением, 0,9 млрд. долл. с поддержанием инфраструктуры, 1,4 млрд. долл. – с обеспечением внутренних ресурсов. Вместе с тем, как отмечают эксперты Delphi Consulting Group, в настоящее время только 12 процентов корпоративных знаний формализовано в компьютерных базах данных и знаний [22-24].

Отличительной особенностью системы управления знаниями является интеграция множества разнородных, часто территориально распределенных источников знаний для решения общих задач. СУЗ интегрирует знания, как из внутренних, так и из внешних источников. Источники знаний могут иметь недокументированную форму (неявные знания экспертов), документированную текстовую, табличную, графическую форму и структурируемую форму в виде баз знаний экспертных систем.

К внутренним источникам знаний относятся:

- техническая документация, описание производственных и деловых процессов,
- внутрифирменные базы данных (data bases) и информационные хранилища (data warehouse),
- базы знаний опыта работы специалистов («лучшей практики»),
- описание профилей знаний специалистов (экспертов),
- специализированные экспертные системы

Внешние источники знаний включают:

- материалы публикаций и новости, содержащиеся в ИНТЕРНЕТ,
- электронные обучающие системы,
- внешние базы данных партнеров и статистические базы данных в региональном, продуктовом и отраслевом разрезах,
- справочники экспертов и консалтинговых компаний, специализирующихся в конкретных проблемных областях, ссылки на форумы в ИНТЕРНЕТ
- референтные модели организации бизнес-процессов (отраслевые и типовые решения).

Система управления знаниями обычно используется в двух аспектах:

- Обеспечение качественными знаниями процессов решения различных задач.
- Создание интерактивной среды взаимодействия специалистов в процессе решения задач.

С точки зрения первого аспекта СУЗ в отличие от традиционных информационных систем документационного обеспечения (информационно-поисковых систем) превращают знания в законченный продукт с высокой потребительной стоимостью, поскольку знание в отличие от набора относящейся к запросу всевозможной информации точно соответствует характеру решаемой задачи и может использоваться непосредственно при выработке решения.

С точки зрения второго аспекта СУЗ создает интерактивную среду общения людей, в которой повышается креативная способность генерации новых знаний, сразу попадающих в корпоративную память для последующего использования. С помощью СУЗ любое предприятие или организация превращаются в обучающуюся организацию, создающую «спираль знаний», в которой «неизвестные (неявные) знания должны быть выявлены и распространены, чтобы стать частью индивидуализированной базы знаний каждого работника. Спираль возобновляется всякий раз для подъема на новый уровень, расширяя базу знаний, применяемую к разным областям организации» [16].

Таким образом, для того чтобы процесс обновления знания был постоянным, необходимо создание постоянно функционирующих систем управления знаниями, которые могли бы не только объединять индивидуальные источники знаний отдельных пользователей, но и извлекать знания из внешних источников знания, баз статистических данных, информационного пространства ИНТЕРНЕТ. Для этого требуется возможность подключения корпоративной системы управления знаниями к другим системам управления знаний на базе единых подходов к концептуализации знаний.

Для систем управления знаниями характерны следующие особенности:

Интеллектуальное ассистирование. Система управления знаниями в отличие от экспертных систем не заменяет эксперта в процессе решения задач, а помогает ему, обеспечивая релевантной информацией и правилами принятия решений в конкретной ситуации. При этом в процессе решения задачи пользователь рассматривает различные варианты решений, представляемые системой управления знаниями, модифицирует постановку задачи или моделирует ситуацию, выбирая, таким образом, наиболее приемлемые решения. Может быть и другой режим решения задачи, когда пользователь самостоятельно решает задачу, а результат решения оценивает с помощью системы управления знаниями на предмет правильности и эффективности, например, обращаясь с помощью системы управления знаниями к коллегам-экспертам или экспертным системам.

Сбор и систематическая организация знаний из различных источников. Интеграция множества разнородных источников знаний осуществляется на основе единой системы концептуализации знаний. Основное требование к источникам знаний – предотвратить потерю и увеличить доступность всех видов корпоративного знания путем обеспечения централизованного, хорошо структурированного информационного хранилища. Структурированность информационного хранилища предполагает создание и описание единой системы знаний на основе таксономии концептуальных понятий, базы метазнаний или онтологии, через которую можно получать доступ к различным источникам знаний.

Минимизация проектирования базы знаний, как наполнения концептуальной схемы. Не все источники знаний могут быть известны, или они могут быть противоречивы, или часто изменяться. Система управления знаниями, с одной стороны, должна обеспечивать методы обработки таких ситуаций, а с другой стороны, обеспечивать легкое подключение новых источников знаний по мере их появления. При этом должна модифицироваться и сама концептуальная схема системы управления знаниями.

Быстрая адаптация системы управления знаниями к изменяющимся информационным потребностям. Адаптация системы управления знаниями осуществляется на основе обратной связи пользователей к системе в результате формулирования ими новых требований и разрезов анализа, а также информирования о пробелах, неверности и запаздывании знаний. Кроме того, происходит наполнение базы прецедентов с указанием успешного и неудачного решения проблем пользователями. При этом выполняется преобразование индивидуального знания в групповое знание, доступное для других пользователей системы управления знаниями. Форма собираемого и распределяемого знаний становится генеративной и повторно-используемой.

С этой точки зрения полезно накопление базы знаний прецедентов решения задач, обобщение характерных ситуаций и ошибок и распространение формируемых знаний.

Интеграция с существующей программной средой. К системе управления знаниями подключаются разнообразные аналитические инструменты, которые позволяют проводить извлечение неявной информации, содержащейся в источниках знаний, с помощью таких методов как статистический анализ и нейронные сети, экспертные системы, математическое и имитационное моделирование. Эти инструменты позволяют обнаруживать закономерности в отражаемой действительности и выявлять наиболее рациональное поведение в существующих условиях. Пользователь становится исследователем-экспериментатором, который не просто запрашивает интересующего его информацию, но и выдвигает и проверяет различные гипотезы. В этом смысле полезно подключение специализированных инструментальных программных средств.

Активная презентация релевантной информации. Система управления знаниями становится компетентным партнером в кооперативном решении проблем, учитывающим круг информационных потребностей пользователя и формирующим для него знания, исходя из его предполагаемых интересов. Особенно эффективно это свойство реализуется в системах управления отношениями с клиентами (CRM – customer relationship management).

Интеграция разнородных источников знаний, междисциплинарный характер их использования, необходимость привлечения внешних источников знаний, обмен знаниями между пользователями предполагает проведение разработки *архитектуры системы управления знаниями на основе общего информационного пространства в виде интегрированной памяти*, которую можно представить на трех взаимодействующих уровнях [1,43]:

Синтаксический (объектный) уровень – хранение аннотированных с помощью специально разработанной системы категорий источников знаний и их индексирование.

Семантический (понятийный) уровень – определение концептуальной модели структуры знаний (системы категорий), общей для всех источников знаний, то есть разработка онтологии [15].

Прагматический уровень (уровень приложений) – определение цели и ограничений на решение интеллектуальной задачи пользователем, то есть задание его глобального и локального контекста. При этом должен быть определен профиль знаний пользователя.

Рассмотрим перечисленные уровни организации корпоративной памяти более детально. При описании организации знаний на этих уровнях приводятся примеры конкретных действующих СУЗ, описанные в [1,2,11,19, 21-24,38,43].

Объектный уровень корпоративной памяти

Возможные источники знаний, которые подключаются к СУЗ, представлены в таблице 1.2.

Таблица 1.2

Источники знаний

Источники знаний	Методы приобретения знаний	Примеры знаний
Люди и группы	Документирование, Структурирование, Подключение	Примеры ситуаций, Правила, объекты Профиль знания
Данные	Извлечение	Статистические базы данных, Информационные хранилища
Тексты	Извлечение	Публикации, Новости, Методики
Графические схемы	Преобразование	Референтные модели процессов, организационных структур

Опыт специалистов предприятия или внешних экспертов представляет основной источник знаний, который может быть представлен в нескольких формах:

- в документальной форме, например, в базе прецедентов, отражающей описания конкретных ситуаций из реальной практики работы людей;
- в структурированной базе знаний экспертных систем;
- в неявной форме путем непосредственного подключения специалистов и экспертов к вычислительной сети посредством описания их профиля знаний.

Опыт ведения прецедентов лучшей практики известен, например, в компании Hughes Electronics, входящая в состав General Motors, которая ведет базу данных лучших проектов реконструкции предприятий. С каждым проектом связывается краткое описание и информация для контакта с ответственными лицами.

Базы статистических данных и информационные хранилища представляют собой источник для извлечения неявных

знаний с помощью методов интеллектуального анализа данных: индуктивного построения деревьев решений, кластерного и регрессионного анализа, построения нейронных сетей. Информационные хранилища могут содержать огромные объемы данных. Например, банк Chase Manhattan Bank имеет хранилище объемом более 560 Гбайт, компания MasterCard OnLine – 1,2 Тбайт. Для сбора статистических данных могут использоваться специализированные инструментальные средства. Например, для извлечения знаний из финансовой информации в системе EDGAR (Electronic Data Gathering and Retrieval System) была разработана система EdgarScan, функционирующая в среде Internet.

Текстовые источники знаний подключаются к СУЗ с помощью технологии фильтрации анализируемых в источнике знаний текстовых сообщений на основе списка тем, определяющих иерархию таксономии терминов конкретной предметной области. Как правило, фильтрация проводится по категориям и приоритетам важности. Например, система Odie (On demand Information Extractor) каждую ночь сканирует около 1000 статей с последними новостями для извлечения знаний о тенденциях в менеджменте. Odie, разработанная для американских и европейских обозревателей новостей, использует распознавание стилизованных фраз в статьях деловых новостей и знания о синтаксических правилах для распознавания соответствующих событий в сфере бизнеса.

Графические схем референтных моделей хранятся в специальных репозиториях, описывающих метаинформацию об организации бизнес-процессов. Например, в репозитории системы R/3 SAP содержится около 100 испытанных организационно-экономических сценариев и моделей для различных отраслей. Референтные модели с помощью экспертных правил могут быть преобразованы в конкретные модели организации бизнес-процессов предприятий.

Концептуальный уровень структуры знаний корпоративной памяти

В основе концептуального уровня структуры знаний лежит таксономия используемых понятий (онтология), предназначенная для идентификации различных компонентов знания. Онтологию можно рассматривать, как систему рубрикации предметной области, с помощью которой интегрируются разнородные источники знаний. С другой стороны, онтология рассматривается, как словарь-тезаурус, совместно используемый в СУЗ для упрощения коммуникации пользователей, формулирования и интерпретации их запросов.

Концептуальный уровень корпоративной памяти обеспечивает семантическую интерпретацию запросов к СУЗ, которая реализует унифицированный интеллектуальный доступ к множеству источников знаний. В результате достигаются следующие ключевые преимущества:

- Точный и эффективный доступ СУЗ к источникам знаниям, релевантным контексту задачи (конкретной ситуации).
- Лучшее понимание и интерпретацию пользователем полученных знаний в данном контексте с возможностью дополнительных справочных обращений к корпоративной памяти.
- Информационное моделирование, то есть итерационное уточнение информационных потребностей в процессе решения задачи.

В качестве примера применения словарей-тезаурусов можно привести промышленную информационно-поисковую систему Retrieval Ware (Canbera), позволяющую осуществлять с использованием обычного web-браузера полнотекстовый и атрибутивный поиск в разнообразных источниках знаний: электронных архивах текстовых документов, web-ресурсах, базах форматированных данных, электронных таблицах, графических, звуковых и визуальных образах (в 250 форматах). Семантическая сеть, отражаемая в словаре-тезаурусе для английского языка, включает 400000 слов и более 1600000 связей между ними. В словаре локализованной версии программного продукта «Русский семантический сервер» в настоящее время отражено около 42 тысяч слов и идиоматических выражений. Технология семантического поиска позволяет также использовать одновременно несколько словарей. Например, одновременно с базовым словарем, система может использовать отраслевой словарь, внутренний словарь

организации, а также личный словарь пользователя, которые могут разрабатываться по мере необходимости.

Ограниченность тезаурусной организации знаний связана с узкой направленностью на решение только поисковых задач. Для расширения круга задач СУЗ, связанных с выбором альтернатив при обосновании проектов, принятии решений, обучении, требуется более сложная организация знаний, предполагающая определение в декларативной форме логики решения задач или создание онтологий задач.

В качестве хорошего примера такого рода системы можно назвать систему Ontolingua. Система Ontolingua предназначена для поддержки формального специфицирования задач пользователя на основе библиотеки формальных описаний фрагментов задач, моделей и понятий, а также для ведения самой библиотеки фрагментов (рис.1.17).

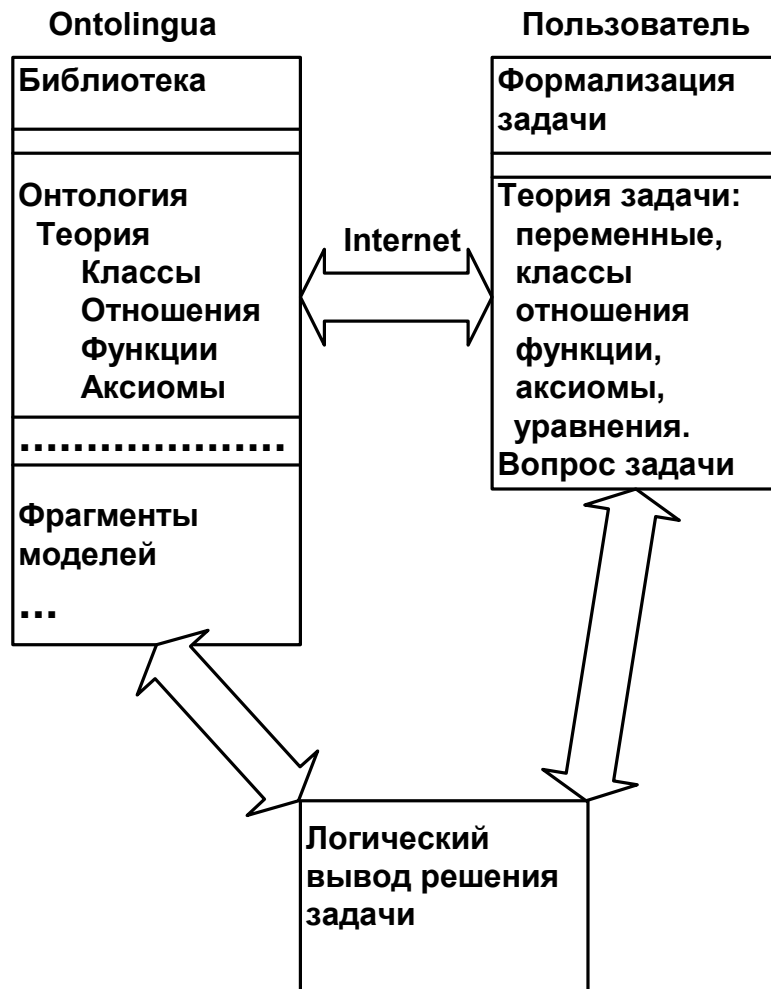


Рис.1.17. Использование онтологической системы Ontolingua

Отологии могут использоваться не только в системах управления знаниями, но и в транзакционных системах таких, как системы электронной коммерции, логистические системы, виртуальные предприятия, в которых требуется многоагентная технология обмена знаниями.

Уровень приложения

Интеллектуальные задачи, которые решаются на основе СУЗ, отличаются слабой формализованностью, предполагающей нечеткость постановки целей решения задачи и описания условий решения задачи. Кроме того, уровень знаний и система критериев оценки решения у различных пользователей могут отличаться. Обычно типичное решение интеллектуальной задачи сводится к следующим шагам:

- Формулировка постановки задачи.
- Отбор релевантных задаче источников знаний.
- Понимание отобранного материала (обучение, консультация).
- Решение задачи (выработка, конфигурация решения).
- Проверка допустимости решения задачи (оценка решения задачи, консультация с экспертами, экспертными системами).
- Принятие решений и мониторинг его реализации.
- Запоминание результатов решения задачи в корпоративной памяти.

На каждом шаге решения интеллектуальной задачи может использоваться СУЗ, работа которой сводится к итерационной серии поисков в корпоративной памяти, обеспечивающей точность достижения цели каждого этапа.

Например, на предприятии решается задача выбора стратегии повышения эффективности производства, которая относится к классу слабо формализованных задач. Для начала необходимо запросить информацию о возможных целях, критериях и методах повышения эффективности производства. Далее описать свое предприятие. На основе сформулированной задачи система произведет отбор источников знаний. Изучение материала может вызвать уточняющий или объясняющий диалог. Принятое решение может быть отправлено выбранным с помощью СУЗ специалистам на экспертизу. В процессе реализации решения информация об отдельных шагах заносится в базу знаний для корректировки стратегии на основе выявленных отклонений и для использования при решении аналогичных задач в будущем.

На каждом этапе решения интеллектуальной задачи определяются требования по входу и выходу, используемым методам и средствам, а также используемым ресурсам (конкретным исполнителям, временным, материальным и стоимостным ограничениям) и чисто внешне выглядит как заполнение некоторой экранной формы, в которой фиксируются:

- Общая цель активности.
- Контекстная информация, известная из состояния процесса или задаваемая вручную на данном шаге.

Экранная форма, как правило, заполняется с помощью списков ключевых слов, связанных с рубриками онтологии. Таким образом, экранная форма запроса отражает текущий локальный контекст поиска, который определяет особенности выполнения следующего шага решения задачи.

Однократный доступ к знаниям в современных СУЗ сводится в основном либо к поиску релевантных знаний со стороны конечных пользователей, либо к распространению знаний среди пользователей со стороны системы управления знаниями. Таким образом, в качестве инициаторов процессов управления знаниями могут быть как люди, так и СУЗ. С другой стороны носителями знаний, к которым обращаются инициаторы процессов, также могут быть, как люди, так и СУЗ. Возможные варианты взаимодействия людей и СУЗ показаны в таблице 1.3.

Таблица 1.3

Доступ к источникам знаниям

Инициаторы процесса	Носители знаний	
	Люди	Базы знаний
Люди	Сети персонала Интеллектуальные агенты	Технология «pool – хранилищ» - машины поиска и интеллект. агенты
Система управления знаниями	«Выталкивающая – push» технология – посылка знаний непосредственно людям	Путеводители (маршрутные карты), многоагентные системы

Связывание людей в СУЗ осуществляется с помощью многоагентной технологии на основе репозитория умений людей. У каждого участника процесса управления знаниями в вычислительной сети может существовать компьютерный двойник – интеллектуальный агент, который вступает в переговоры с аналогичными агентами в сети для решения общей задачи. Примером такой многоагентной системы может служить система ContactFinder, которая осуществляет поиск экспертов в сети, используя технологию доски объявлений, которая доступна для обмена сообщениями между агентами. При этом велика роль онтологии, как словаря общения агентов и описания профиля их знаний.

Связывание людей и СУЗ для отбора релевантных знаний осуществляется с помощью поисковых машин в хранилищах знаний («pool –хранилищ»). Недостаточная интеллектуальность существующих поисковых машин приводит к нерелевантности отбираемых знаний. В настоящее время разрабатываются поисковые машины на базе применения онтологий, например, система SHOE, OntoBroker и др. В качестве графических инструментов визуализации карты знаний при навигации путей доступа могут использоваться специальные инструменты, например, Perspecta и InXight.

Связывание СУЗ и людей осуществляется по «толкающей» технологии (push-технологии), по которой СУЗ изучает информационные потребности пользователей и в соответствии с ними самостоятельно распространяет или доставляет знания до потребителей. Например, InfoFinder изучает интересы пользователей по наборам формируемых сообщений или запрашиваемых документов. Кроме того, InfoFinder использует эвристические методы для сбора дополнительных, более точных сведений о запросах пользователей, что позволяет InfoFinder находить документы, предугадывая запросы пользователя.

Связывание СУЗ и СУЗ в процессе реализации более сложных запросов, когда одна СУЗ не может справиться с поставленной задачей и обращается за помощью к другой СУЗ, как правило, выполняется по многоагентной технологии. Кроме того, могут использоваться специальные путеводители знаний, с помощью которых упрощается навигация путей отбора знаний в

интегрированном пространстве знаний, например, путеводитель компании Coopers&Librand – “Michelin Guides”, облегчающий понимание содержания и местоположения источников знаний. Компания Andersen Consulting имеет центральное хранилище карт знаний, с помощью которой связываются различные источники знаний.

Что следует запомнить

Интеллектуальная информационная система (ИИС) - это ИС, которая основана на концепции использования базы знаний для генерации алгоритмов решения экономических задач различных классов в зависимости от конкретных информационных потребностей пользователей.

Важнейшие признаки классификации ИИС: развитые коммуникативные способности, сложность (плохая формализуемость алгоритма), способность к самообучению, адаптивность.

Основные подклассы ИИС: интеллектуальные базы данных, в т.ч. с интерфейсами, использующими естественный язык, гипертекст и мультимедиа, когнитивную графику; статические и динамические экспертные системы; самообучающиеся системы на принципах индуктивного вывода, нейронных систем, поиска прецедентов, организации информационных хранилищ; адаптивные информационные системы на основе использования CASE-технологий и/или компонентных технологий, системы управления знаниями.

Система с интеллектуальным интерфейсом - это ИИС, предназначенная для поиска неявной информации в базе данных или тексте для произвольных запросов, составляемых, как правило, на ограниченном естественном языке.

Экспертная система (ЭС) - это ИИС, предназначенная для решения слабоформализуемых задач на основе накапливаемого в базе знаний опыта работы экспертов в проблемной области.

Участники процесса разработки и эксплуатации ЭС: эксперты, инженеры по знаниям, пользователи.

Эксперт - специалист, знания которого помещаются в базу знаний.

Инженер по знаниям - специалист, который занимается извлечением знаний и их формализацией в базе знаний.

Пользователь - специалист, интеллектуальные способности которого расширяются благодаря использованию в практической деятельности ЭС.

Основные составные части архитектуры ЭС: база знаний, механизмы вывода, объяснения, приобретения знаний, интеллектуальный интерфейс.

База знаний - это центральный компонент ЭС, который определяет ценность ЭС и с которым связаны основные затраты на разработку.

База знаний - это хранилище единиц знаний, описывающих атрибуты и действия, связанные с объектами проблемной области, а также возможные при этом неопределенности.

Единица знаний - это элементарная структурная единица, (описание одного объекта, одного действия), которая имеет законченный смысл. В качестве единиц знаний обычно используются **правила** и/или **объекты**.

Неопределенность знаний - это или неполнота, или недостоверность, или многозначность, или качественная (вместо количественной) оценка единицы знаний.

Механизм вывода - это обобщенная процедура поиска решения задачи, которая на основе базы знаний и в соответствии с информационной потребностью пользователя строит цепочку рассуждений (логически связанных единиц знаний), приводящую к конкретному результату.

Дедуктивный вывод (от общего к частному) - вывод частных утверждений путем подстановки в общие утверждения других известных частных утверждений. Различают **прямую** (от данных к цели) и **обратную** (от цели к данным) **цепочки рассуждений (аргументации)**.

Индуктивный вывод (от частного к общему) - вывод (обобщение) на основе множества частных утверждений общих утверждений (из примеров реальной практики правил).

Абдуктивный вывод (от частного к частному) - вывод частных утверждений на основе поиска других аналогичных утверждений (прецедентов).

Механизм приобретения знаний - это процедура накопления знаний в базе знаний, включающая ввод, контроль полноты и непротиворечивости единиц знаний и, возможно, автоматический вывод новых единиц знаний из вводимой информации.

Механизм объяснения - это процедура, выполняющая обоснование полученного механизмом вывода результата.

Интеллектуальный интерфейс - это процедура, выполняющая интерпретацию запроса пользователя к базе знаний и формирующая ответ в удобной для него форме.

Назначение экспертной системы: консультирование и обучение неопытных пользователей, ассистирование экспертам в решении задач, советы экспертам по вопросам из смежных областей знаний (интеграция источников знаний).

Статическая экспертная система - это ЭС, решающая задачи в условиях не изменяющихся во времени исходных данных и знаний.

Динамическая экспертная система - это ЭС, решающая задачи в условиях изменяющихся во времени исходных данных и знаний.

Аналитическая экспертная система - это ЭС, осуществляющая оценку вариантов решений (проверку гипотез).

Синтетическая экспертная система - это ЭС, осуществляющая генерацию вариантов решений (формирование гипотез).

Классы решаемых задач в экспертной системе: интерпретация, диагностика, прогнозирование, проектирование, планирование, мониторинг, коррекция, управление.

Самообучающаяся система - это ИИС, которая на основе примеров реальной практики автоматически формирует единицы знаний.

Система с индуктивным выводом - это самообучающаяся ИИС, которая на основе обучения на примерах реальной практики строит деревья решений.

Нейронная сеть - это самообучающаяся ИИС, которая на основе обучения на примерах реальной практики строит ассоциативную сеть понятий (нейронов) для параллельного поиска на ней решений.

Система, основанная на прецедентах, - это самообучающаяся ИИС, которая в качестве единиц знаний хранит собственно прецеденты решений (примеры) и позволяет по запросу подбирать и адаптировать наиболее похожие прецеденты.

Интеллектуальный анализ данных (Data Mining) – это набор методов автоматического извлечения знаний из специально организованных информационных хранилищ (**Data Warehouse**), к которым относят статистические методы, индуктивный вывод, нейронные сети, генетические алгоритмы.

Адаптивная информационная система - это ИИС, которая изменяет свою структуру в соответствии с изменением модели проблемной области.

Модель проблемной области - отражение структуры объектов, функций, процессов, правил, связанных с функционированием проблемной области.

Репозиторий - хранилище метазнаний о структуре фактуального и операционного знания или модели проблемной области.

Case - технология - технология, позволяющая генерировать информационную систему на основе модели проблемной области, хранимой в репозитории.

Компонентная технология - технология, позволяющая конфигурировать информационную систему из готовых типовых компонентов на основе модели проблемной области, хранимой в репозитории.

Система управления знаний (СУЗ) – взаимосвязанная совокупность организационных процедур, людей и информационных технологий, которая обеспечивает сбор, накопление, организацию, распространение и использование знаний для решения задач качественного информационного обслуживания (обеспечения) выполнения деловых процессов и интерактивного взаимодействия специалистов.

Корпоративная память – хранилище источников знаний и их метаописания для коллективного использования в организации.

Источники знаний СУЗ – опыт специалистов, представляемый в виде прецедентов, структурированных баз знаний экспертных систем, описаний профилей знаний; документальных источников знаний внутри и вне организации; баз и хранилищ

форматированных данных; референтных моделей организации деловых процессов.

Онтология – концептуальное описание структуры знаний для формализованных и не формализованных источников.

Приложения СУЗ – поиск и использование знаний для решения интеллектуальных задач обоснований решений, проектирования, инноваций; обучение; обмен знаниями в процессе взаимодействия специалистов; распространение знаний для ознакомления в потенциальном плане.

Литература

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: Учебник.-М.: Питер, 2000. – 382 с. :ил.
2. Герстинг А., Ивс Б. Как больше узнать об управлении знаниями // Computerworld, 1999, № 35.
3. Зайцева Ж. Н., Рубин Ю. Б., Титарев Л. Г., Тихомиров В. П., Хорошилов А. В., Усков В. Л., Филиппов В. М. Открытое образование – стратегия XXI века для России / Под общей редакцией Филиппова В. М. и Тихомирова В. П. - М.: Издательство МЭСИ, 2000. – 324с.
4. Дубров А.М., Лагоша Б.А., Хрусталеv Е.Ю. Моделирование рисковvх ситуаций в экономике и бизнесе: Учеб. пособие / Под ред. Б.А. Лагоши. – Финансы и статистика, 1999. – 176с. :ил.
5. Джексон П. Введение в экспертные системы.: Пер. с англ.: Учебное пособие. – М.: Изд. дом «Вильямс», 2001. – 624с. :ил.
6. Информационные системы в экономике: Учебник / Под ред. проф. В.В. Дика. - М.: Финансы и статистика, 1996. - 272с. :ил.
7. Искусственный интеллект. Книга 1. Системы общения и экспертные системы / Под ред. проф. Э.В.Попова.- М.: Радио и связь, 1990. - 461с. :ил.
8. Искусственный интеллект. Книга 2. Модели и методы / Под ред. проф. Д.А.Поспелова. - М.: Радио и связь, 1990. - 304 с. :ил.
9. Искусственный интеллект. Книга 3. Программные и аппаратные средства / Под. ред. В.Н.Захарова, В.Ф.Хорошевского. - М.: Радио и связь, 1990. - 320с. :ил.
- 10.Калянов Г.Н. Консалтинг при автоматизации предприятий: Научно-практическое издание. Серия «Информатизация России на пороге XXI века». - М.: СИНТЕГ, 1997. - 316 с. :ил.

11. Каменова М.С. Корпоративные информационные системы: технологии и решения // Системы управления базами данных, №4, 1995.
12. Курейчик В.М., Курейчик В.В. Эволюционные, синергетические и гомеостатические стратегии в искусственном интеллекте: состояние и перспективы //Новости искусственного интеллекта. М.: РАИИ, 2000. №3, с. 39 – 65.
13. Ларичев О.И., Браун Р. Количественный и вербальный анализ решений: сравнительное исследование возможностей и ограничений //Экономика и математические методы. – 1998. – т.34. – вып.4. – с.97-107.
14. Левин Р., Дранг В., Эделсон Б. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на Бэйсике / Пер. с англ.- М.: Финансы и статистика, 1991.- 239с. :ил.
15. Мизогучи Р. Шаг в направлении инженерии онтологий // Новости искусственного интеллекта. М.: РАИИ, 2000. №1-2, с. 11 – 36.
16. Мильнер Б.З. Теория организации. М.: ИНФРА-М, 1999. – 478 с.
17. Мишенин А.И. Теория экономических информационных систем: Учебник. - М.: Финансы и статистика, 1998 - 166 с. :ил.
18. Обработка знаний / Пер. с япон.: Под ред. С.Осуга. - М.: Мир, 1989. - 292 с. :ил.
19. О'Лири Д. Е. Управление корпоративными знаниями // Открытые системы, 1998, №4-5
20. Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса: Реинжиниринг организаций и современные информационные технологии. - М.: Финансы и статистика, 1997. -336с.: ил.
21. Попов Э.В., Шапот М.Д., Кисель Е.Б., Фоминых И.Б. Статические и динамические экспертные системы: Учебное пособие. - М: Финансы и статистика, 1996. -320с. : ил.
22. Попов Э.В. Корпоративные системы управления знаниями. // Новости искусственного интеллекта. 2001, №1. –с. 14-25.
23. Попов Э.В., Василевский А.С. Предприятия нового типа и управление знаниями в реинжиниринге. // Сборник научных трудов 4-й Российской научно-практической конференции «Реинжиниринг бизнес-процессов на основе современных информационных технологий». М.: МЭСИ, 2000. – с. 9 – 15..

24. Попов Э.В. Управление корпоративными знаниями. // Сборник научных трудов 6-й Российской научно-практической конференции «Реинжиниринг бизнес-процессов на основе современных информационных технологий. Системы управления знаниями». М.: МЭСИ, 2002. - с.11-19.
25. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных фактов. - М.: Радио и связь, 1989.- 184 с. :ил.
26. Приобретение знаний / Пер. с япон.: Под ред. С.Осуга., Ю. Сазки - М.: Мир, 1990. - 292 с. :ил.
27. Представление и использование знаний / Пер. с япон.: Под ред. Х.Уэно, М.Исидзука. - М.: Мир, 1990. - 220 с. :ил.
28. С миру по байту «Искусственный интеллект "горячей линии" Dell» // Computerworld, №9, 1998.
29. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем: Учебник / Под ред. Ю.Ф. Тельнова. – М.: Финансы и статистика, 2001. – 512 с. : ил.
30. Тарасов В.Б. Искусственная жизнь и нечеткие эволюционные многоагентные системы // Теория и системы управления, 1998, N5. – с. 12-23.
31. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. - М.: Финансы и статистика, 1990.- 319с. :ил.
32. Трахтенгерц Э.А. Компьютерная поддержка принятия решений. М.: Синтег, 1998. – 360с. :ил.
33. Тельнов Ю.Ф., Сорова А.А., Андреева Н.В. Проектирование баз знаний: Учебное пособие.- М.: МЭСИ, 1992.-100с. :ил.
34. Тельнов Ю.Ф., Диго С.М., Полякова Т.М. Интеллектуальные системы обработки данных: Учебное пособие.- М.: МЭСИ, 1989.- 102с. :ил.
35. Тельнов Ю.Ф. Интеллектуальные информационные системы в экономике: Учебное пособие /2-изд. – М.: СИНТЕГ, 1999.– 214 с.
36. Уоссермен Ф. Нейрокомпьютерная техника. Теория и практика. / Пер с англ. Ю.А. Зуева М.:Мир, 1992 -237 с. :ил.
37. Уотерман Д. Руководство по экспертным системам. / Пер. с англ.; Под. ред. Стефанюка В.Л. - М.: Мир, 1989.- 388 с. :ил.
38. Davenport T.H. Some Principles of Knowledge Management. Apr, 1996. <http://www.netobjects.com>

39. Durkin J. Expert Systems: a view of the field // IEEE Expert, 1996, No 2, p. 56- 63
40. Harmon P. The intelligente software development tools market // Part I. Intelligent Software Strategies. - 1995, Vol. 11, No 2. p. 1 - 12.
41. Harmon P. The intelligente software development tools market // Part II. Intelligent Software Strategies. - 1996, Vol. 12, No 3. p. 1 - 16.
42. Martinson, F.R. Schindler. Organizational visions for technology assimilation: the strategic road to knowledge-based systems success // IEEE Transactions on engineering management, 1995, Vol 42, No 1, p 10 - 18.
43. O'Leary D.E. Using AI in Knowledge Management: Knowledge Bases and Ontologies. // IEEE Intelligent Systems. 1998, No 3. p.34-39.
44. Ross R.G. The Business Rule Book. Classifying, Defining and Modelling Rules. Data Base Research Group, Inc. -1997, 394 p.
45. K.D. Althoff, E. Auriol, R. Barletta, M. Manago, "A Review of Industrial Case-Based Reasoning Tools" // AI Intelligence, 1995.
46. R. Lopez de Mantaras, E. Plaza, "Case-Based Reasoning: an overview" // AI Communications, № 10, 1997, pp. 21-29.
47. B. Bartsch-Sporl, M. Lenz, A. Hubner, "Case-Based Reasoning – Survey and Future Directions ", Knowledge-Based Systems, Survey and Future Directions, Springer Verlag, Berlin, Heidelberg, № 4, 1999.

Глава 2. Технология создания экспертных систем

2.1. Этапы создания экспертной системы

Слабая формализуемость процесса принятия решений, его альтернативность и нечеткость, качественная и символическая природа используемых знаний, динамичность изменения проблемной области - все эти характерные особенности применения экспертных систем обуславливают сложность и большую трудоемкость их разработки по сравнению с другими подклассами ИИС. Поэтому в дальнейшем вопросы проектирования и реализации интеллектуальных информационных систем будут рассматриваться для класса экспертных систем.

Извлечение знаний при создании экспертной системы предполагает изучение множества источников знаний, к которым относятся специальная литература, базы фактуальных знаний, отчеты о решении аналогичных проблем, а самое главное опыт работы специалистов в исследуемой проблемной области - экспертов. Успех проектирования экспертной системы во многом определяется тем, насколько компетентны привлекаемые к разработке эксперты и насколько они способны передать свой опыт инженерам по знаниям. Вместе с тем, эксперты не имеют представления о возможностях и ограничениях ЭС. Следовательно, процесс разработки ЭС должен быть организован инженерами по знаниям таким образом, чтобы в процессе их итеративного взаимодействия с экспертами они получили весь необходимый объем знаний для решения четко очерченных проблем. Этапы проектирования экспертной системы представлены на рис. 2.1.

На начальных этапах идентификации и концептуализации, связанных с определением контуров будущей системы, инженер по знаниям выступает в роли ученика, а эксперт - в роли учителя, мастера. На заключительных этапах реализации и тестирования инженер по знаниям демонстрирует результаты разработки, адекватность которых проблемной области оценивает эксперт. На этапе тестирования это могут быть совершенно другие эксперты.

На этапе тестирования созданные экспертные системы оцениваются с позиции двух основных групп критериев: точности и полезности.

С точностью работы связаны такие характеристики, как правильность делаемых заключений, адекватность базы знаний проблемной области, соответствие применяемых методов решения проблемы экспертным. Поэтому конечные оценки системе ставят специалисты в проблемной области - эксперты. Полезность же экспертной системы характеризуется степенью удовлетворения требований пользователя в части получения необходимых рекомендаций, легкости и естественности взаимодействия с системой, надежности, производительности и стоимости эксплуатации, способности обоснования решений и обучения, настройки на изменение потребностей. Оценивание экспертной системы осуществляется по набору тестовых примеров, как из предшествующей практики экспертов, так и специально подобранных ситуаций. Результаты тестирования подлежат статистической обработке, после чего делаются выводы о степени точности работы экспертной системы.

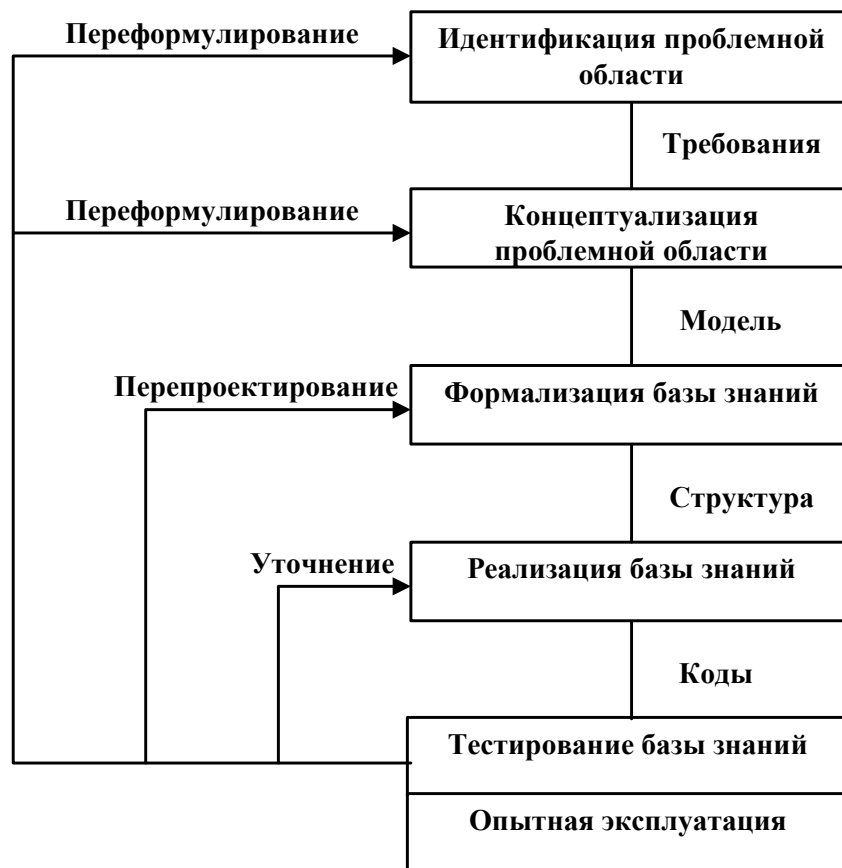


Рис.2.1. Этапы создания экспертной системы

Следующий этап жизненного цикла экспертной системы - внедрение и опытная эксплуатация в массовом порядке без непосредственного контроля со стороны разработчиков и переход от тестовых примеров к решению реальных задач. Важнейшим критерием оценки становятся соотношение стоимости системы и ее эффективности. На этом этапе осуществляется сбор критических замечаний и внесение необходимых изменений. В результате опытной эксплуатации может потребоваться разработка новых специализированных версий, учитывающих особенности проблемных областей.

На всех этапах разработки инженер по знаниям играет активную роль, а эксперт - пассивную. По мере развития самообучающихся свойств экспертных систем роль инженера по знаниям уменьшается, а активное поведение заинтересованного в эффективной работе экспертной системы пользователя-эксперта возрастает. Описание приемов извлечения знаний инженерами знаний представлено в таблице 2.1.

Таблица 2.1

Приемы извлечения знаний	
Приемы	Описание
1. Наблюдение	Инженер наблюдает, не вмешиваясь, за тем, как эксперт решает реальную задачу
2. Обсуждение задачи	Инженер на представительном множестве задач неформально обсуждает с экспертом данные, знания и процедуры решения
3. Описание задачи	Эксперт описывает решение задач для типичных запросов
4. Анализ решения	Эксперт комментирует получаемые результаты решения задачи, детализируя ход рассуждений
5. Проверка системы	Эксперт предлагает инженеру перечень задач для решения (от простых до сложных), которые решаются разработанной системой
6. Исследование системы	Эксперт исследует и критикует структуру базы знаний и работу механизма вывода
7. Оценка системы	Инженер предлагает новым экспертам оценить решения разработанной системы

Первые два этапа разработки экспертной системы составляют логическую стадию, не связанную с применением четко определенного инструментального средства. Последующие этапы реализуются в рамках физического создания проекта на базе выбранного инструментального средства. Вместе с тем, процесс создания экспертной системы, как сложного программного продукта, имеет смысл выполнять методом прототипного проектирования, сущность которого сводится к постоянному наращиванию базы знаний, начиная с логической стадии. Технология разработки прототипов представлена в таблице 2.2.

Прототипная технология создания экспертной системы означает, что простейший прототип будущей системы реализуется с помощью любого подручного инструментального средства еще на этапах идентификации и концептуализации, в дальнейшем этот прототип детализируется, концептуальная модель уточняется, реализация выполняется в среде окончательно выбранного инструментального средства. После каждого этапа возможны итеративные возвраты на уже выполненные этапы проектирования, что способствует постепенному проникновению инженера по знаниям в глубину решаемых проблем, эффективности использования выделенных ресурсов, сокращению времени разработки, постоянному улучшению компетентности и производительности системы.

Пример разработки экспертной системы гарантирования (страхования) коммерческих займов CLUES (loan-underwriting expert systems) [21] представлен в таблице 2.3. Эта система создавалась в интегрированной среде ART группой разработчиков в составе одного менеджера проекта, двух инженеров по знаниям, двух программистов, ответственных за сопряжение ЭС с существующей информационной системой и аналитическим инструментом, одного контролера качества. Сложность созданной системы: 1000 правил, 180 функций, 120 объектов. Эффективность: при оценке 8500 кредитов в месяц годовая экономия на обработке информации составляет 0,91 млн. долл., при 30000 кредитов - 2,7 млн. долл. При этом в 50% случаев система принимает самостоятельные решения, в остальных случаях дает экспертам диагностику возникающих проблем. Время оценки кредита сократилось с 50 минут до 10-15 минут. Перечисленные показатели эффективности позволили компании Contrywide расширить сферу своей деятельности во всех штатах США и увеличить оборот с 1 млрд. долл. в месяц в 1991 году до 5 млрд. долл. в 1993 году.

Таблица 2.2.

Этапы разработки прототипов ЭС

Этап разработки	Характер прототипа	Количество правил	Срок разработки	Стоимость
Идентификация	Демонстрацион-ный	50 - 100	1 - 2 мес.	
Концептуализация	Исследователь-ский	200 - 500	3 - 6 мес.	25 - 50т.\$
Формализация				
Реализация	Действующий	500 - 1000	6 - 12 мес.	
Тестирование	Промышленный	1000 - 1500	1 - 1,5 года	300т.\$
Опытная эксплуатация	Коммерческий	1500 - 3000	1,5 - 3 года	2 - 5 млн.\$

2.2. Идентификация проблемной области

Этап идентификации проблемной области включает определение назначения и сферы применения экспертной системы, подбор экспертов и группы инженеров по знаниям, выделение ресурсов, постановку и параметризацию решаемых задач.

Начало работ по созданию экспертной системы инициируют руководители компаний (предприятий, учреждений). Обычно необходимость разработки экспертной системы в той или иной сфере деятельности связана с затруднениями лиц, принимающих решение, что сказывается на эффективности функционирования проблемной области. Эти затруднения могут быть обусловлены недостаточным опытом работы в данной области, сложностью постоянного привлечения экспертов, нехваткой трудовых ресурсов для решения простых интеллектуальных задач, необходимостью интеграции разнообразных источников знаний. Как правило, назначение экспертной системы связано с одной из следующих областей:

- обучение и консультация неопытных пользователей;
- распространение и использование уникального опыта экспертов;
- автоматизация работы экспертов по принятию решений;
- оптимизация решения проблем, выдвижение и проверка гипотез.

Таблица 2.3.

Разработка экспертной системы CLUES

Период времени	Этап
Ноябрь 1991г.	Постановка проблемы
Январь 1992г.	Создание отдела ЭС
Февраль - апрель 1992г.	Интервьюирование экспертов
Апрель - май 1992г.	Моделирование и создание первого прототипа
Май - июнь 1992г.	Кодирование (реализация)
Июнь - сентябрь 1992г.	Внутреннее тестирование. Системная интеграция
Сентябрь - декабрь 1992г.	Альфа-тестирование на известных примерах
Декабрь - январь 1993г.	Бета-тестирование на реальных примерах

Февраль 1993г.	Внедрение в отрасли розничной торговли (20% кредитов)
Май 1993г.	Внедрение в потребительский сектор (10% кредитов)
Август 1993г.	Внедрение в отрасли оптовой торговли (35% кредитов)
Февраль 1994г.	Внедрение в корреспондентскую сеть (35% кредитов)

Сфера применения экспертной системы характеризует тот круг задач, который подлежит формализации, например, "оценка финансового состояния предприятия", "выбор поставщика продукции", "формирование маркетинговой стратегии" и т.д. Обычно сложность решаемых в экспертной системе проблем должна соответствовать трудоемкости работы эксперта в течение нескольких часов. Более сложные задачи имеет смысл разбивать на совокупности взаимосвязанных задач, которые подлежат разработке в рамках нескольких экспертных систем.

Ограничивающими факторами на разработку экспертной системы выступают отводимые сроки, финансовые ресурсы и программно-техническая среда. От этих ограничений зависит количественный и качественный состав групп инженеров по знаниям и экспертов, глубина прорабатываемых вопросов, адекватность и эффективность решения проблем. Обычно различают три стратегии разработки экспертных систем (таблица 2.4) [20]:

- широкий набор задач, каждая из которых ориентирована на узкую проблемную область;
- концентрированный набор задач, определяющий основные направления повышения эффективности функционирования экономического объекта;
- комплексный набор задач, определяющий организацию всей деятельности экономического объекта.

Таблица 2.4.

Стратегии разработки экспертных систем

	Широкий набор задач	Концентрированный набор задач	Комплексный набор задач
Назначение	Автоматизация	Стандартизация, повышение качества	Реорганизация бизнес-процессов
Требования к разработчикам	Эксперты-пользователи	Профессиональные команды	Междисциплинарные команды
Стоимость	Низкая на проект	Высокая на проект	Высокая на проект
Риск	Диверсифицированный	Концентрированный	Концентрированный
Примеры	DuPont du Nemours Оболочка Insight Plus Сотни экспертных систем. Сотни правил в каждой ЭС	DEC, ЭС конфигурирования компьютеров XCON, продажи XSEL 17000 правил, эффект 27 млн. долл.	Xerox Среда разработки информационной системы ART-Enterprise (Inference) Интеллект. моделирование ReThink (Gensym)

После предварительного определения контуров разрабатываемой экспертной системы инженеры по знаниям совместно с экспертами осуществляют более детальную постановку проблем и параметризацию системы. К основным параметрам проблемной области относятся следующие:

- класс решаемых задач (интерпретация, диагностика, коррекция, прогнозирование, планирование, проектирование, мониторинг, управление);
- критерии эффективности результатов решения задач (минимизация использования ресурсов, повышение качества продукции и обслуживания, ускорение оборачиваемости капитала и т.д.);
- критерии эффективности процесса решения задач (повышение точности принимаемых решений, учет большего числа факторов, просчет большего числа альтернативных вариантов, адаптивность к изменениям проблемной области и информационных потребностей пользователей, сокращение сроков принятия решений);
- цели решаемых задач (выбор из альтернатив, например, выбор поставщика или синтез значения, например, распределение бюджета по статьям);
- подцели (разбиение задачи на подзадачи, для каждой из которых определяется своя цель);
- исходные данные (совокупность используемых факторов);
- особенности используемых знаний (детерминированность/неопределенность, статичность/динамичность, одноцелевая/многоцелевая направленность, единственность/множественность источников знаний).

2.3. Построение концептуальной модели

На этапе построения концептуальной модели создается целостное и системное описание используемых знаний, отражающее сущность функционирования проблемной области. От качества построения концептуальной модели проблемной области во многом зависит насколько часто в дальнейшем по мере развития проекта будет выполняться перепроектирование базы знаний. Хорошая концептуальная модель может только уточняться (детализироваться или упрощаться), но не перестраиваться.

Результат концептуализации проблемной области обычно фиксируется в виде наглядных графических схем на объектном, функциональном и поведенческом уровнях моделирования:

- объектная модель описывает структуру предметной области как совокупности взаимосвязанных объектов;
- функциональная модель отражает действия и преобразования над объектами;
- поведенческая модель рассматривает взаимодействия объектов во временном аспекте.

Первые две модели описывают статические аспекты функционирования проблемной области, а третья модель - динамику изменения ее состояний. Естественно, что для различных классов задач могут требоваться разные виды моделей, а следовательно, и ориентированные на них методы представления знаний. Рассмотрим каждую из представленных видов моделей.

Объектная модель отражает фактуальное знание о составе объектов, их свойств и связей. Элементарной единицей структурного знания является факт, описывающий одно свойство или одну связь объекта, который представляется в виде триплета:

Предикат (Объект, Значение).

Если предикат определяет название свойства объекта, то в качестве значения выступает конкретное значение этого свойства, например:

профессия ("Иванов", "Инженер").

Если предикат определяет название связи объекта, то значению соответствует объект, с которым связан первый объект, например:

работает ("Иванов", "Механический цех").

В качестве важнейших типизированных видов отношений рассматриваются следующие:

“род” - “вид” (обобщение);
 “целое” - “часть” (агрегация);
 “причина” - “следствие”;
 “цель” - “средство”;
 “функция” - “аргумент”;
 “ассоциация”;
 “хронология”;

“пространственное положение” и др.

Так, отношения обобщения ("род" - "вид") фиксируется на уровне названий классов объектов, например:

есть-подкласс (Инженеры, Личности).

Под классом объектов понимается совокупность объектов с одинаковым набором предикатов (свойств и связей). Класс объектов часто описывается в виде n-арного реляционного отношения, например:

личности (ФИО, Профессия, Подразделение, ...).

Если объекты обладают частично пересекающимся набором предикатов, то осуществляется более сложная классификация объектов: класс объектов по значениям какого-либо свойства (признака) разбивается на подклассы таким образом, что класс объектов содержит общие для подклассов свойства и связи, а каждый из подклассов отражает специфические свойства и связи, например:

личности (ФИО, Год рождения, Профессия, Подразделение,...)

инженеры (ФИО, ВУЗ, Оклад, ...)

рабочие (ФИО, Разряд, Тарифная ставка, ...)

При этом подклассы объектов автоматически наследуют общие свойства и связи вышестоящих классов, а совокупность взаимосвязанных по отношению обобщения классов объектов образует иерархию наследования свойств.

Отношение агрегации классов объектов ("целое" - "часть") отражает составные части объектов, которое можно представить в бинарном виде на именах двух классов объектов:

есть-часть (Оборудование, Цех);

есть-часть (Рабочие, Цех).

Аналогично представляются другие семантические отношения:

- причина-следствие (Задолженность, Банкротство);
- аргумент-функция (Спрос, Цена);
- средство-цель (Покупка акций, Прибыль);

- ассоциация (Производство, Обслуживание);
- хронология (Отгрузка, Поставка);
- пространственное положение (Сборка, Технический контроль).

Обычно объектное знание представляется графически средствами ER-моделей (модель "Сущность - Связь") - см. [15].

Функциональная модель описывает преобразования фактов, зависимости между ними, показывающие, как одни факты образуются из других. В качестве единицы функционального знания определим функциональную зависимость фактов в виде импликации:

$$A1_ \wedge A2_ \wedge A3_ \wedge \dots \wedge A_n_ \rightarrow B,$$

означающей, что факт В имеет место только в том случае, если имеет место конъюнкция фактов или их отрицаний A1, A2, ... , An , например:

сбыт (Товар, "Слабый") и
 прибыль (Товар, "Ничтожная") и
 потребители (Товар, "Любители нового") и
 число_конкурентов (Товар, "Небольшое") -- >
 жизненный_цикл (Товар, "Выведение на рынок").

Функциональную зависимость фактов можно трактовать как отражение следующих отношений фактов:

- "Причина" - "Следствие";
- "Средство" - "Цель";
- "Аргумент" - "Функция";
- "Ситуация" - "Действие".

В качестве термов конъюнкции фактов могут выступать более сложные логические условия.

Функциональная модель строится путем последовательной декомпозиции целей, а именно: для цели определяются подцели, для которых в свою очередь устанавливаются подцели и так дальше, пока в качестве подцелей не окажутся исходные факты (процесс декомпозиции "сверху" - "вниз"). Каждой цели (подцели) соответствует некоторая задача (подзадача), которая не может быть решена, пока не будут достигнуты ее нижестоящие подцели (решены подзадачи). Таким образом, функциональная модель

отражает в обобщенной форме процесс решения характерных для нее задач.

Обычно функциональные зависимости фактов представляются графически в виде деревьев целей или графов "И"- "ИЛИ" (рис. 2.2), в которых каждый зависимый факт представляет собой целевую переменную - корневую вершину, а определяющие его факты-аргументы - связанные с корнем подчиненные вершины, условие конъюнкции (совместности анализа факторов) обозначается пересекающей дугой, а условие дизъюнкции (независимости влияния на цель факторов) никак не обозначается, причем если какой-либо факт-аргумент, в свою очередь, определяется другими фактами-аргументами, то он становится подцелью.

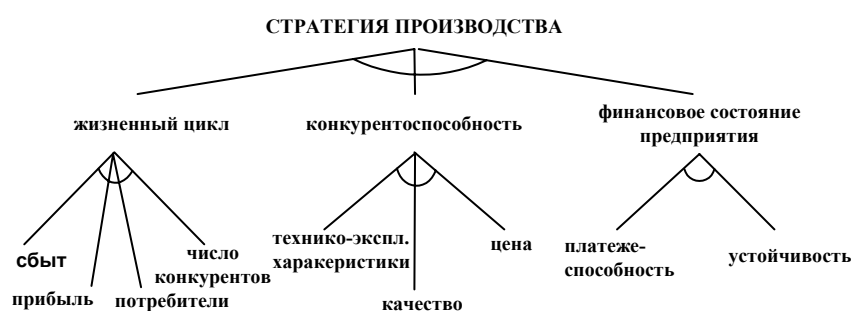


Рис.2.2. Дерево целей

Поведенческая модель отражает изменение состояний объектов в результате возникновения некоторых событий, влекущих за собой выполнение определенных действий (процедур). Состояние объекта - это изменяющиеся во времени значения некоторого свойства. Набор действий, связанный с некоторым событием, составляет поведение объекта, которое выражается в виде правил или процедур. Задача определения поведенческой модели заключается в определении связей событий с поведением объектов и изменением их состояний. Как правило, событие отражается в форме сообщения, посылаемого объекту. Пример поведенческой модели в виде диаграммы потоков событий представлен в таблице 2.5.

Таблица 2.5.

ЗАКАЗ

СОБЫТИЕ (Сообщение)	ПОВЕДЕНИЕ (Действие)	СОСТОЯНИЕ (Жизненный цикл)
Оформляется	Создание заказа Проверка выполнимости	Оформлен
Откладывается	Заказ у производителя	Отложен
Выполняется	Отправляется Выписываются платежные документы	Выполнен
Оплачивается	Уничтожение заказа	Оплачен

2.4. Формализация базы знаний

На этапе формализации базы знаний осуществляется выбор метода представления знаний. В рамках выбранного формализма осуществляется проектирование логической структуры базы знаний.

Рассмотрим классификацию методов представления знаний с точки зрения особенностей отображения различных видов концептуальных моделей, а именно: соотношения структурированности и операционности, детерминированности и неопределенности, статичности и динамичности знаний (рис. 2.3).



Рис.2.3. Классификация методов представления знаний

Так, объектные методы представления знаний в большей степени ориентированы на представление структуры фактуального знания, а правила - операционного.

- Логическая модель реализует и объекты, и правила с помощью предикатов первого порядка, является строго формализованной моделью с универсальным дедуктивным и монотонным методом логического вывода “от цели к данным”;
- Продукционная модель позволяет осуществлять эвристические методы вывода на правилах и может обрабатывать неопределенности в виде условных вероятностей или коэффициентов уверенности, а также выполнять монотонный или немонотонный вывод;
- Семантическая сеть отображает разнообразные отношения объектов;

- Фреймовая модель, как частный случай семантической сети, использует для реализации операционного знания присоединенные процедуры;
- Объектно-ориентированная модель, как развитие фреймовой модели, реализуя обмен сообщениями между объектами, в большей степени ориентирована на решение динамических задач и отражение поведенческой модели.

Логическая модель предполагает унифицированное описание объектов и действий в виде предикатов первого порядка. Под предикатом понимается логическая функция на N - аргументах (признаках), которая принимает истинное или ложное значение в зависимости от значений аргументов. Отличие заключается в том, что для объектов соответствующие реляционные отношения задаются явно в виде фактов, а действия описываются как правила, определяющие логическую формулу вывода фактов из других фактов. Пример фрагмента базы знаний подбора претендентов на вакансии в языке логического программирования ПРОЛОГ представлен на рис. 2.4. (Обозначения: ":-" - "если", "," - "и", "." - "конец утверждения").

```
vibor (Fio,Dolgnost) :-
    pretendent (Fio, Obrazov, Stag),
    vacancy (Dolgnost, Obrazov, Opyt),
    Stag>=Opyt.
pretendent ("Иванов","среднее" 10).
pretendent ("Петров","высшее", 12).
vacancy ("менеджер", "высшее", 10).
vacancy ("директор", "высшее", 15).
```

Рис. 2.4. Пример фрагмента базы знаний на языке ПРОЛОГ

Механизм вывода осуществляет дедуктивный перебор фактов, относящихся к правилу по принципу "сверху - вниз", "слева - направо" или обратный вывод методом поиска в глубину. Так, в ответ на запрос vibor(X,Y) получим: X="Петров", Y="менеджер".

Правила могут связываться в цепочки в результате использования одинакового предиката в посылке одного и в заключении другого правила.

Для логической модели характерна строгость формального аппарата получения решения. Однако полный последовательный перебор всех возможных решений может приводить к комбинаторным взрывам, в результате чего поставленные задачи могут решаться недопустимо большое время. Кроме того, работа с неопределенностями знаний должна быть запрограммирована в виде самостоятельных метаправил, что на практике затрудняет разработку баз знаний с помощью логического формализма.

Продукционные модели используются для решения более сложных задач, которые основаны на применении эвристических методов представления знаний, позволяющих настраивать механизм вывода на особенности проблемной области и учитывать неопределенность знаний.

В продукционной модели основной единицей знаний служит правило в виде: "если <посылка>, то <заключение>", с помощью которого могут быть выражены пространственно-временные, причинно-следственные, функционально-поведенческие (ситуация - действие) отношения объектов. Правилами могут быть описаны и сами объекты: "объект - свойство" или "набор свойств - объект", хотя чаще описания объектов фигурируют только в качестве переменных ("атрибут - значение") внутри правил. В основном продукционная модель предназначена для описания последовательности различных ситуаций или действий и в меньшей степени для структурированного описания объектов.

Продукционная модель предполагает более гибкую организацию работы механизма вывода по сравнению с логической моделью. Так, в зависимости от направления вывода возможна как прямая аргументация, управляемая данными (от данных к цели), так и обратная, управляемая целями (от целей к данным). Прямой вывод используется в продукционных моделях при решении, например, задач интерпретации, когда по исходным данным нужно определить сущность некоторой ситуации или в задачах прогнозирования, когда из описания некоторой ситуации требуется вывести все следствия. Обратный вывод применяется, когда нужно проверить определенную гипотезу или небольшое множество гипотез на соответствие фактам, например, в задачах диагностики.

Отличительной особенностью продукционной модели является также способность осуществлять выбор правил из множества возможных на данный момент времени (из конфликтного набора) в зависимости от определенных критериев, например, важности, трудоемкости, достоверности получаемого результата и других характеристик проблемной области. Такая стратегия поиска решений называется поиском в ширину. Для ее реализации в описание продукций вводятся предусловия и постусловия в виде:

$\langle A, B, C \rightarrow D, E \rangle$, где

- импликация $C \rightarrow D$ представляет собственно правило;
- A - предусловие выбора класса правил;
- B - предусловие выбора правила в классе;
- E - постусловие правила, определяющее переход на следующее правило.

В предусловиях и постусловиях могут быть заданы дополнительные процедуры, например, по вводу и контролю данных, математической обработке и т.д. Введение предусловий и постусловий позволяет выбирать наиболее рациональную стратегию работы механизма вывода, существенно сокращая перебор относящихся к решению правил.

Сами правила могут иметь как простой, так и обобщенный характер. Простые правила описывают продукции над единичными объектами, обобщенные правила определяются на классах объектов (аналогично правилам языка ПРОЛОГ).

Для обработки неопределенностей знаний продукционная модель использует, как правило, либо методы обработки условных вероятностей Байеса, либо методы нечеткой логики Заде.

Байесовский подход предполагает начальное априорное задание предполагаемых гипотез (значений достигаемых целей), которые последовательно уточняются с учетом вероятностей свидетельств в пользу или против гипотез, в результате чего формируются апостериорные вероятности:

$$P(H / E) = P(E / H) * \frac{P(H)}{P(E)} \text{ и}$$

$$P(\overline{H} / E) = P(E / \overline{H}) * \frac{P(\overline{H})}{P(E)},$$

где $P(H)$ - априорная вероятность гипотезы Н;

$P(\overline{H}) = 1 - P(H)$ - априорная вероятность отрицания гипотезы Н;

$P(E)$ - априорная вероятность свидетельства Е;

$P(H / E)$ - апостериорная (условная) вероятность гипотезы Н при условии, что имеет место свидетельство Е;

$P(\overline{H} / E)$ - апостериорная (условная) вероятность отрицания гипотезы Н при условии, что имеет место свидетельство Е;

$P(E / H)$ - вероятность свидетельства гипотезы Е при подтверждении гипотезы Н;

$P(E / \overline{H})$ - вероятность свидетельства гипотезы Е при отрицании гипотезы Н.

Найдем отношения левых и правых частей представленных уравнений:

$$\frac{P(H / E)}{P(\overline{H} / E)} = \frac{P(E / H)}{P(E / \overline{H})} * \frac{P(H)}{P(\overline{H})}$$

или

$$O(H / E) = Ls * O(H), \text{ где}$$

$O(H)$ - априорные шансы гипотезы Н, отражающие отношение числа позитивных проявлений гипотезы к числу негативных;

$O(H / E)$ - апостериорные шансы гипотезы Н при условии наличия свидетельства Е;

Ls - фактор достаточности, отражающий степень воздействия на шансы гипотезы при наличии свидетельства Е.

Аналогично выводится зависимость:

$$O(H / \overline{E}) = Ln * O(H), \text{ где}$$

$O(H/\bar{E})$ - апостериорные шансы гипотезы Н при условии отсутствия свидетельства Е;

L_n - фактор необходимости, отражающий степень воздействия на шансы гипотезы при отсутствии свидетельства Е.

Шансы и вероятности связаны уравнениями:

$$O = \frac{P}{1-P} \text{ и } P = \frac{O}{O+1}$$

Отсюда апостериорная вероятность гипотезы рассчитывается через апостериорные шансы, которые в свою очередь получают путем перемножения априорных шансов на факторы достаточности или необходимости всех относящихся к гипотезе свидетельств в зависимости от их подтверждения или отрицания со стороны пользователя. Свидетельства рассматриваются как независимые аргументы на дереве целей.

Рассмотрим использование байесовского подхода на примере оценки надежности поставщика. Фрагмент подмножества правил представляется следующим образом:

Если Задолженность = "есть",

То Финансовое_состояние = "удовл." $L_s = 0.01$, $L_n = 10$

Если Рентабельность = "есть",

То Финансовое_состояние = "удовл." $L_s = 100$, $L_n = 0.001$

Пусть оцениваемое предприятие является рентабельным и без задолженностей. Априорная вероятность удовлетворительного финансового состояния любого поставщика составляет 0.5. Тогда расчет апостериорных шансов и вероятности удовлетворительного финансового состояния осуществляется по формулам:

$$O(H/E1, E2) = 1 * 10 * 100 = 1000;$$

$$P(H/E1, E2) = \frac{1000}{1001} = 0,99.$$

Для байесовского подхода к построению продукционной базы знаний характерна большая трудоемкость статистического

оценивания априорных шансов и факторов достаточности и необходимости.

Подход на основе нечеткой логики. Более простым, но менее точным методом оценки достоверности используемых знаний является применение нечеткой логики, в которой вероятности заменяются на экспертные оценки определенности фактов и применения правил (факторы или коэффициенты уверенности). Факторы уверенности могут рассматриваться и как весовые коэффициенты, отражающие степень важности аргументов в процессе вывода заключений. Итоговые факторы уверенности получаемых решений главным образом отражают порядок достоверности результата, а не его точность, что вполне приемлемо во многих задачах.

Факторы уверенности измеряются по некоторой относительной шкале, например, от 0 до 1 (100). В отличие от теории вероятностей сумма факторов уверенностей некоторых альтернативных значений необязательно составляет 1. Предполагается, что оценка факторов уверенностей исходных данных задается пользователем при описании конкретной ситуации, а факторы уверенности применения правил определяются инженерами знаний совместно с экспертами при наполнении базы знаний. Факторы уверенности правил в системе GURU задаются после служебного слова cf. Например,

Если Предприятие = "малое"

То Надежность = "есть" cf 30

Фактор уверенности можно определить и как степень принадлежности некоторого элемента подмножеству объектов, обладающих некоторым свойством. Например, определяется степень принадлежности (фактор уверенности принадлежности) предприятия подмножеству предприятий с высоким уровнем рентабельности.

Нечетким множеством называется совокупность упорядоченных пар, составленных из элементов и степеней их принадлежности некоторому подмножеству универсального множества, выделяемому по некоторому свойству. Например, нечеткими множествами могут быть подмножества предприятий с

неудовлетворительной, удовлетворительной, хорошей, отличной рентабельностью.

Нечеткое множество определяется математически как множество пар вида:

$$A = \{(x, \mu_A(x))\}, \text{ где}$$

x есть элемент универсального множества X ,

$\mu_A(x)$ есть степень принадлежности элемента x подмножеству универсального множества X , определяемому свойством A .

Нечеткое множество можно также задать с помощью функции принадлежности $\mu_A \rightarrow X[0,1]$, которая вычисляет степень принадлежности элемента x подмножеству A на интервале от 0 до 1.

Один и тот же элемент универсального множества в силу нечеткости может одновременно принадлежать нескольким подмножествам или нечетким множествам. Например, одно и то же предприятие на некотором интервале значений рентабельности (рис.2.5) может относиться и к классу предприятий с неудовлетворительной рентабельностью, и к классу предприятий с удовлетворительной рентабельностью. Множество допустимых значений принадлежностей элемента нескольким нечетким множествам определяет лингвистическую переменную (нечеткую переменную в терминологии системы GURU). Например, для рентабельности предприятия 6,7% получается оценка в виде лингвистической переменной:

$$\{\text{“неудовлетворительно” cf } 66, \text{ “удовлетворительно” cf } 33\}.$$

Присвоение значений лингвистической переменной на интервалах значений элементов может быть задано с помощью продукционных правил. Например:

Если Коэф. рент. > 5 и Коэф. рент. ≤ 10

То Рентабельность = $\{\text{«неудовл.» cf } \mu_n(\text{Коэф. рент.}),$
 $\text{«удовл.» cf } \mu_y(\text{Коэф. рент.})\}$

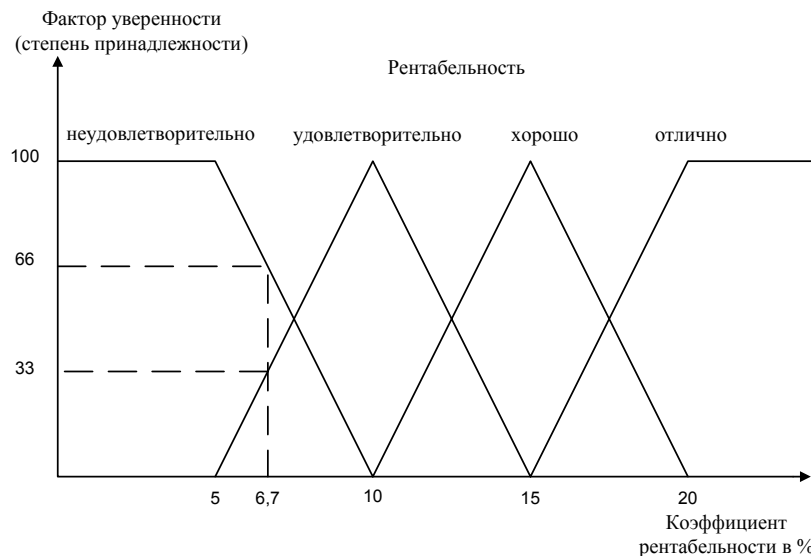


Рис. 2.5. Лингвистическая переменная «Рентабельность»

Рассмотрим алгебру объединения факторов уверенностей (степеней принадлежности) при выполнении правил с несколькими лингвистическими переменными (аргументами) и при назначении лингвистическим переменным разными правилами нескольких значений.

При объединении факторов уверенности конъюнктивно или дизъюнктивно связанных аргументов используются следующие формулы:

Конъюнкция (А и В): $\min(cfA, cfB)$ или $cfA * cfB / 100$

Дизъюнкция (А или В): $\max(cfA, cfB)$ или $cfA + cfB - cfA * cfB / 100$

Объединение факторов уверенности в посылках правил осуществляется чаще всего, как, например, в программном средстве GURU, по формулам "min" для конъюнкции, арифметических операторов и операторов отношения и "max" для дизъюнкции. При объединении факторов уверенности левых и правых частей правил и одинаковых результатов нескольких правил соответственно используются формулы "произведения" (левая и правая части) и "суммы" (одинаковые результаты правых частей).

Для объединения одинаковых результатов нескольких правил используется оператор "+=", который означает не присваивание значения, а добавление значения в лингвистическую (нечеткую) переменную. Для удаления значения из лингвистической (нечеткой) переменной используется оператор "-=". Факторы уверенности в последнем случае объединяются по формуле:

$$cfA*(100 - cfB)/100$$

Рассмотрим применение аппарата нечеткой логики на примере оценки надежности поставщика, в котором кроме фактора финансового состояния учитывается и фактор формы собственности. Пусть государственное предприятие не имеет задолженность с уверенностью 60 и предполагается, что его рентабельность удовлетворительна с уверенностью 80. Фрагмент множества правил имеет следующий вид:

Правило 1: Если Задолженность = "нет" и Рентабельность = "удовл."
То Финансовое_состояние = "удовл." cf 100

Правило 2: Если Финансовое_состояние = "удовл."
То Надежность += "есть" cf 90

Правило 3: Если Предприятие = "государств."
То Надежность += "есть" cf 50

Результат выполнения первого правила:

$$cf(\text{посылки}) = \min(60, 80) = 60,$$
$$cf(\text{Фин_сост.} = \text{"удовл."}) = 60 * 100 / 100 = 60.$$

Результат выполнения второго правила:

$$cf(\text{Надежность} = \text{"есть"}) = 60 * 90 / 100 = 54$$

Результат выполнения третьего правила:

$$cf(\text{Надежность} = \text{"есть"}) = 100 * 50 / 100 = 50$$

Объединение факторов уверенности одинаковых результатов (Надежность="есть"), полученных во втором и третьем правилах осуществляется по формуле: $54 + 50 - 54 * 50 / 100 = 87$.

Заметим, что в системе GURU при выполнении нескольких правил, обрабатывающих одно и то же значение переменной операторами "+=" и "-=", объединение коэффициентов уверенности осуществляется последовательно сначала для всех "+=", а затем "-=".

Динамические модели. Моделирование рассуждений человека, как правило, не сводится только к прямой или обратной аргументации. Сложные проблемы решаются путем выдвижения во времени нескольких гипотез с анализом подтверждающих фактов и непротиворечивости следствий. Причем для многоцелевых проблемных областей происходит увязка гипотез по общим ограничениям. При этом возможны задержки в принятии решений, связанные со сбором подтверждающих фактов, доказательством подцелей, входящих в ограничения.

Следовательно, для подобных динамических проблем важна рациональная организация памяти системы для запоминания и обновления получаемых промежуточных результатов, обмен данными между различными источниками знаний для достижения нескольких целей, изменение стратегий вывода с выдвижения гипотез (прямая аргументация) к их проверке (обратная аргументация). Целям построения таких гибких механизмов вывода служит применение технологии "доски объявлений", через которую в результате осуществления событий источники знаний обмениваются сообщениями.

В целях динамического реагирования на события некоторые продукционные модели используют специальные правила-демоны, которые формулируются следующим образом: "Всякий раз, как происходит некоторое событие, выполнить некоторое действие". Например:

Всякий раз, как становится известным значение переменной "Поставщик",

Выполнить набор правил "Финансовый анализ предприятия"

В программном средстве GURU подобное правило будет записано следующим образом:

```
IF:   KNOWN("Поставщик") = true  
THEN: CONSULT FIN_AN
```

Для динамических экспертных систем характерна также обработка времени как самостоятельного атрибута аргументации логического вывода:

Если в течение дня уровень запаса понизился больше, чем на 50 %

То выполнить набор правил “Выбор поставщика для поставки”

Общим недостатком всех формализмов представления знаний, основанных на правилах, является недостаточно глубокое отражение семантики проблемной области, что может сказываться на гибкости формулирования запросов пользователей к экспертным системам. Этот недостаток снимается в объектно-ориентированных методах представления знаний.

Семантические сети. Объектно-ориентированные методы представления знаний берут начало от семантических сетей, в которых типизируются отношения между объектами. Элементарной единицей знаний в семантической сети служит триплет (см. объектную концептуальную модель), в котором имя предиката представляет помеченную дугу между двумя узлами графа, соответствующими двум связанным объектам (рис. 2.6):

Важнейшими типизированными отношениями объектов являются: "Род" - "Вид", "Целое" - "Часть", "Причина" - "Следствие", "Средство" - "Цель", "Аргумент" - "Функция", "Ситуация" - "Действие". Типизация отношений позволяет однозначно интерпретировать смысл отображаемых в базе знаний ситуаций и настраивать механизм вывода на особенности этих отношений. Так, отображение отношений "Род" - "Вид" дает возможность осуществлять наследование атрибутов классов объектов и, таким образом, автоматизировать процесс вывода заключений от общего к частному.

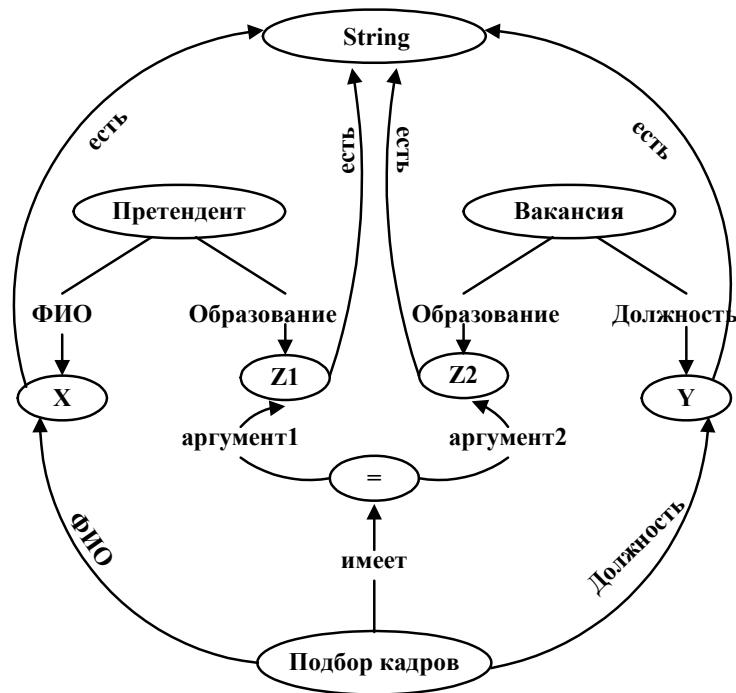


Рис. 2.6. Пример семантической сети

Фреймы. Развитием семантических сетей являются фреймовые методы представления знаний, в которых все атрибуты (поименованные отношения) объектов собираются в одну структуру данных, называемую фреймом. Причем в качестве значений слотов (атрибутов) могут выступать как обычные значения данных, так и действия, направленные на получение этих значений. Таким образом, действия реализуются в виде присоединенных процедур или процедур-демонов, вызываемых по определенным условиям. В этом плане фреймовый метод представления знаний в большей степени операционно-ориентирован по сравнению с семантической сетью.

Неопределенность описания знаний реализуется в результате неполного заполнения всех слотов. Фреймовая модель способна делать предположения о значениях данных на основе

механизма наследования свойств в иерархии обобщения. В качестве способов наследования атрибутов применяются следующие возможности: S - идентичность значений одноименных слотов; U - различные значения одноименных слотов; R - значение слота фрейма должно находиться в пределах, заданных в одноименном слоте фрейма верхнего уровня; O - в случае неизвестности значения слота фрейма нижнего уровня принимается значение слота фрейма верхнего уровня.

Способность изменения значений слотов с течением времени позволяет решать динамические задачи. Во фреймовых моделях могут выполняться как прямая, так и обратная аргументация, когда в прямом направлении в зависимости от состояния слотов фреймов запускаются процедуры-демоны (неизвестно значение - "if-needed", известно значение - "if-added", удаляется значение - "if-removed"), а обратная аргументация срабатывает путем запуска присоединенных процедур при обращении к неизвестным значениям атрибутов. Фреймовые модели позволяют более гибко комбинировать прямой и обратный вывод. Пример описания фреймовой модели представления знаний для задачи заключения контрактов с поставщиками показан на рис. 2.7:

Объектно-ориентированная модель, аналогичная во многих отношениях фреймовой модели, также предусматривает инкапсуляцию процедур в структуры данных и механизм наследования. Отличия заключаются в четком различии понятий класс объектов и экземпляр объекта, а также в способе активации процедур к объектам. Для объектно-ориентированной модели характерны такие черты, как скрытие данных и их доступность только через методы (присоединенные процедуры) класса, наследование как атрибутов, так и методов (в последнем случае обеспечивается необходимый уровень абстракции данных и полиморфизм использования процедур). Обращение к объектам, то есть вызов методов класса, осуществляется либо из внешних программ, либо из других объектов путем посылки сообщений.

КОНТРАКТ

Имя слота	Указатель	Тип	Значение	IF-NEEDED	IF-ADDED	IF-REMOVED
Super-class	U	FRAME	ROOT			
Sub-class	U	FRAME	Проект			
Код изделия	U	String	Отвергнутый Заключенный	Ввести	Проектировать Заключить	Отвергнуть
Статус	U	Boolean	101			
Код поставщика	U	String	Y			
Срок поставки	U	Date	123			
Объем поставки	U	Real	01.06.96			
Стоимость	U	Real	2000			
			10000			

ПРОЕКТИРУЕМЫЙ КОНТРАКТ

Имя слота	Указатель наследования	Тип	Значение	IF-NEEDED	IF-ADDED	IF-REMOVED
Super-class	S	FRAME	Контракт			
Sub-class	-	-				
Код изделия	S	String	101		Выбор поставщика	
Статус	S	Boolean	Y		Надежность	
Код поставщика	S	String	123			
Срок поставки	S	Date	01.06.96			
Объем поставки	S	Real	2000			
Стоимость	S	Process	Расчет			

Рис.2.7. Пример фреймовой модели

Рассмотрим пример объектно-ориентированной модели обработки заказов, которые являются динамическими объектами, меняющими свое состояние в течение своего жизненного цикла.

Описание поведенческой модели для класса объектов "Заказ" приведено в таблице 2.5. Описание объектно-ориентированной модели представлено на рис.2.8. Псевдокод основных методов класса объектов "Заказ" дается на рис.2.9.

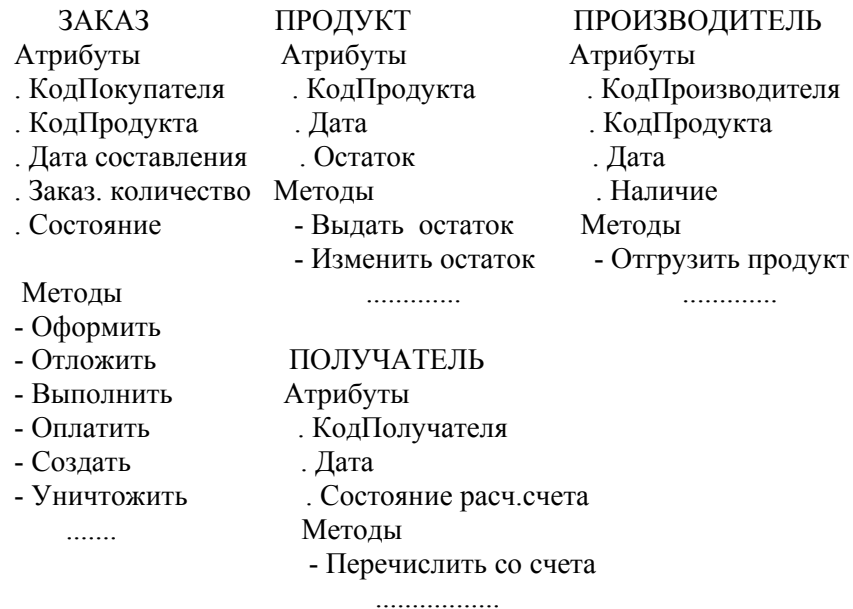


Рис.2.8. Объектно-ориентированная модель

```
Заказ.Оформить(КодПокупателя, КодПродукта, Дата, Количество):
{
Заказ.Создать(КодПокупателя,КодПродукта,Дата,Количество,Сост)
```

```

A = Продукт.Выдать_остаток(КодПродукта, Остаток);
Если Заказ.Количество >= A
То      Заказ.Отложить(КодПокупателя,КодПродукта,Дата,
                        Количество);
Иначе Заказ.Выполнить(КодПокупателя,КодПродукта,Дата,
                        Количество);
Заказ.Состояние = "Оформлен". }

```

Рис.2.9. Реализация метода "Оформить заказ"

2.5. Выбор инструментальных средств реализации экспертной системы

На этапе реализации экспертной системы происходит физическое наполнение базы знаний и настройка всех программных механизмов в рамках выбранного инструментального средства, а при необходимости и допрограммирование специализированных модулей программного инструмента.

Особенности реализации экспертной системы во многом определяются характером инструментального средства, в качестве которого могут выступать программные оболочки (shells), генераторы (интегрированные среды), языки представления знаний (языки программирования). Так, оболочки имеют реализованные механизмы вывода, накопления, объяснения знаний, диалоговый компонент, что, с одной стороны, упрощает разработку программной части экспертной системы, поскольку не требуется программирование, а с другой стороны, усложняет разработку базы знаний вследствие возможного несоответствия формализма системы требованиям структуры. Использование языков представления знаний таких как: язык логического программирования PROLOG, язык функционального программирования LISP, язык объектно-ориентированного программирования SmallTalk, язык продукционных правил OPS5 и др. повышает гибкость разрабатываемой системы и одновременно увеличивает трудоемкость разработки.

Наиболее приемлемыми инструментальными средствами для создания экспертных систем являются генераторы или

интегрированные среды разработки, например, G2 (фирма Gensym, дистрибьютор фирма ArgusSoft), ART-Enterprise (фирма Inference, дистрибьютор фирма Весть-Метатехнология), Nexpert Objects (дистрибьютор ТОРА Центр), GURU (фирма MDBS), которые позволяют настраивать программные средства на особенности проблемных областей, при необходимости предоставляют возможность программировать на встроенных языках четвертого поколения и осуществлять эффективный экспорт/импорт данных с другими инструментальными средствами.

Среди отечественных разработок следует отметить экспертную оболочку ЭКО (ArgusSoft) и программный комплекс SIMER - MIR (Исследовательский центр искусственного интеллекта ИПС РАН), который предоставляет инструментальные средства как автоматизации разработки, так и поддержки экспертных систем.

Инструментальные средства создания и поддержки экспертных систем являются дорогостоящими продуктами и стоят от тысяч до десятков тысяч долларов. Однако для готовых баз знаний инструментальные средства могут поставляться в исполнительской версии (RUN-TIME) на порядок дешевле.

В последнее время активно стал развиваться рынок проблемно и предметно ориентированных систем. Причем цена предметно-ориентированных систем может быть на порядок выше универсальных, преимущество же их заключается в более простой адаптации к конкретной предметной области, а следовательно, и в сокращении затрат на разработку. Например, интеллектуальная система для разработки финансовых приложений Cogensys Judgment Software (Cogensys Corp) стоит 200 тыс. долл.

Среди специализированных инструментальных средств интеллектуальных систем основной удельный вес занимают экспертные системы реального времени, позволяющие динамически управлять непрерывными процессами (70% рынка). Бесспорным лидером в разработке экспертных систем реального времени является фирма Gensym с инструментальным средством G2 (дистрибьютор в России - фирма ArgusSoft), имеющая внедрения в таких компаниях как IBM, NASA, General Electric, Nissan и др. [5]. На базе G2, в свою очередь, созданы такие проблемно-ориентированные комплексы, как GDA для решения задач диагностики, ReThink и eSCOR для моделирования бизнес-

процессов на различных уровнях абстракции, NeurOnline для поддержки нейронной сети, Operations Expert для управления телекоммуникациями и др.

В процессе жизненного цикла разработки экспертной системы инструментальные средства могут сменять друг друга по мере расширения базы знаний. Так, на этапе проектирования прототипа требуется его быстрая разработка в ущерб производительности, в то время как на этапе разработки промышленной версии на первый план выходит обеспечение эффективности функционирования.

На выбор инструментальных средств экспертной системы, в основе которых лежит определенный метод представления знаний, основное влияние оказывает класс решаемых задач (проблемных областей) и соответственно характер полученной концептуальной модели, определяющий множество требований в части отображения объектов, действий над объектами, методов обработки неопределенностей, механизмов вывода (таблица 2.6.) [19]:

Инструментальные средства, в свою очередь, характеризуются определенными возможностями по реализации этих требований. В таблице 2.7. представлено выполнение требований для наиболее популярных инструментальных средств [5,8,9] (цифрой показывается номер места в упорядоченной последовательности).

Тогда сущность алгоритма выбора инструментальных средств сводится к наложению требований проблемной области к формализмам знаний на возможности инструментальных средств и определению наилучших по заданным ограничениям (таблица 2.8).

Таблица 2.6.

Матрица «Метод представления знаний – Классы задач»

Описание метода представления знаний	Классы решаемых задач				
	Интер- претация	Диагно- стика	Прогно- зирование	Проекти- рование	Плани- рование
Описание объектов:					

Предикаты	4	2	4	4	4
Правила	1	1	1	3	2
Семант. сеть	2	4	3	2	3
Фреймы(Объекты)	3	3	2	1	1
Действия:					
Предикаты	3	2	4	4	4
Правила	1	1	1	2	2
Сообщения	2	3	3	1	1
Процедуры	4	4	2	3	3
Неопределенность:					
Неполнота	2	4	3	1	1
Нечеткость	3	2	4	3	3
Многозначность	4	3	2	2	2
Недостоверность	1	1	1	4	4
Вывод:					
Прямая цепочка	1	3	1	3	1
Обратн. цепочка	3	1	6	6	6
Объект.ориент.	2	2	2	1	2
Гипотет. вывод	4	4	4	2	5
Обраб. времени	6	6	3	5	3
Доска объявлений	5	5	5	4	4

Таблица 2.7.

Матрица «Метод представления знаний – программное средство»

Описание метода представления знаний	Программные инструментальные средства					
	ЭКО ArgusSoft	GURU MDBS	Nexpert Object Neuron Data	LEVEL-5 Level Co	ART Enterprise Inference	G Gen
Описание объектов:						
Семант. сеть						*

Объекты (Фреймы)			*	*	*	*
Предикаты						
Правила	*	*	*	*	*	*
Действия:						
Правила	*	*	*	*	*	*
Сообщения			*	*	*	
Предикаты						*
Процедуры	*	*	*	*	*	*
Неопределен-ность:						
Наслед. свойств			*	*	*	*
Коэф. уверен.		*		*	*	*
Неч. множества		*		*	*	
Усл. вероятн.	*					
Вывод:						
Прямая цепочка		*	*	*	*	*
Обратн. цепочка	*	*	*	*	*	*
Объект. ориент.			*	*	*	*
Гипотет. вывод					*	*
Обраб. времени				*		*
Доска объявл.						*

Таблица 2.8.

Матрица «Класс решаемых задач – Программное средство»

Классы решаемых задач	Программные инструментальные средства					
	ЭКО	GURU	Nexpert Object	LEVEL- 5	ART Enterpris e	G2
Интерпретация	3	1	1	1	2	3
Диагностика	1	2	2	2	3	2
Прогнозирование	2	3	4	3	4	4
Проектирование	-	-	3	5	1	5
Планирование	-	-	5	4	5	1

Что следует запомнить

Этапы создания экспертных систем: идентификация, концептуализация, формализация, реализация, тестирование, внедрение.

Прототип экспертной системы - это расширяемая (изменяемая) на каждом последующем этапе версия базы знаний с возможной модификацией программных механизмов. Различают прототипы: демонстрационный, исследовательский, действующий, промышленный, коммерческий.

Этап идентификации проблемной области - определение требований к разрабатываемой ЭС, контуров рассматриваемой проблемной области (объектов, целей, подцелей, факторов), выделение ресурсов на разработку ЭС.

Этап концептуализации проблемной области - построение концептуальной модели, отражающей в целостном виде сущность функционирования проблемной области на объектном (структурном), функциональном (операционном), поведенческом (динамическом) уровнях.

Объектная модель - отражение на семантическом уровне фактуального знания о классах объектов, их свойств и отношений. Элементарная единица объектного знания - это **триплет**: “объект - свойство (отношение) - значение” или двухместный предикат.

Функциональная модель - отражение зависимостей фактов, определяющих условия образования одних фактов из других. Элементарная единица функционального знания - **импликация** фактов.

Дерево целей (граф “И”-”ИЛИ”) фиксирует зависимость целевого предиката (переменной) от множества факторов - определяющих предикатов (переменных).

Дерево решений фиксирует зависимость значений целевого предиката от комбинации значений факторов.

Поведенческая модель - отражение выполняемых действий над объектами (фактами) в зависимости от происходящих во времени событий.

Этап формализации базы знаний - выбор метода представления знаний, в рамках которого проектируется логическая структура базы знаний. **Методы представления знаний** различаются характером представления объектного, функционального, поведенческого видов знаний и реализацией неопределенностей, т.е. ориентацией на определение структуры объектов или действий над ними, детерминированность или неопределенность, статику или динамику проблемной области.

Метод представления (модель) знаний - это совокупность средств структурирования и обработки единиц знаний. Методы представления знаний различаются характером представления объектного, функционального, поведенческого видов знаний и реализацией неопределенностей, т.е. ориентацией на определение структуры объектов или действий над ними, детерминированность или неопределенность, статику или динамику проблемной области.

Логическая модель - это модель, в которой область определения предиката задается либо перечислением фактов, либо в виде импликаций (правил).

Продукционная модель - факты - значения переменных, операции над фактами - правила. Правила выбираются из конфликтных наборов с помощью задаваемых эвристических критериев: приоритетов, достоверности, стоимости и т.д.

Простые правила - обрабатывают отдельные значения переменных.

Обобщенные правила - обрабатывают классы объектов.

Правила, управляемые данными:

Если <условие> То <заключение>

Правила, управляемые событиями:

Всякий раз, как <событие> То <действие>

Обработка неопределенностей знаний основана на использовании условных вероятностей или нечеткой логики.

Семантическая сеть отражает как объектное, так и операционное знание в виде двухместных предикатов (бинарных отношений). Различают типизированные отношения “род” - ”вид”, “целое” - “часть”, “причина” - “следствие” и др.

Фреймовая модель - это семантическая сеть с N-арными отношениями и присоединенными процедурами. Используются **механизмы наследования свойств** по иерархии классов объектов и вызова процедур в зависимости от происходящих событий.

Объектно-ориентированная модель предусматривает **инкапсуляцию** процедур (методов) в структуры данных классов объектов, к которым разрешен доступ только через эти методы. Механизм наследования свойств распространяется и на методы, обеспечивая свойство **полиморфизма** процедур.

Этап реализации ЭС представляет отображение структуры базы знаний в среде выбранного инструментального средства, а также настройка и/или доработка программных механизмов. Различают программные оболочки, инструментальные среды и языки представления знаний; универсальные инструментальные, проблемно-ориентированные и предметно-ориентированные инструментальные программные средства.

Алгоритм выбора инструментального средства. Требования класса решаемых задач в части реализации объектов, операций и неопределенностей налагаются на возможности инструментальных средств по представлению выявленных особенностей знаний, в результате чего формируется ранжированный список инструментальных средств.

Этап тестирования оценивает экспертную систему с позиции двух основных групп критериев: точности и полезности.

Точность работы: правильность заключений, адекватность базы знаний проблемной области, соответствие методов решения проблемы экспертным.

Полезность: ответы на запросы пользователя; удобство интерфейса; объяснение получаемых результатов; надежность, адаптируемость, производительность и стоимость эксплуатации.

Этап внедрения и опытной эксплуатации - это переход от тестовых примеров к решению реальных задач.

Литература

1. Буч Г. Объектно-ориентированное проектирование с примерами применения: Пер. с англ. - М.: Конкорд, 1992. - 519с.
2. Ин Ц., Соломон Д. Использование Турбо-Пролога. Пер. с англ. - М.: Мир, 1993. - 608с.
3. Марселлус Д. Программирование экспертных систем на ТУРБО ПРОЛОГЕ. Пер. с англ. - М.: Финансы и статистика, 1994. -256с.
4. Нечеткие множества в моделях управления и искусственного интеллекта/ Под ред. Поспелова Д.А. - М.: Наука, 1986 - 312 стр.
5. Попов Э.В., Шапот М.Д., Кисель Е.Б., Фоминых И.Б. Статические и динамические экспертные системы. - М: Финансы и статистика, 1996. -320с.
6. Построение экспертных систем / Под ред. Ф. Хейес-Рот, Д.Уотерман, Д.Ленат / Пер. с англ. - М.: Мир, 1987. - 441 с.
7. Представление и использование знаний / Пер. с япон.; Под ред. Х.Уэно, М.Исидзука. - М.: Мир, 1989. - 220 с.
8. Рыбина Г.В. Особенности современных подходов к построению экспертных систем // Труды международного конгресса «Искусственный интеллект в XXI веке». – М.: Физматлит, 2001, том 1. –383 – 390с.
9. Системы управления базами данных и знаний: Справочное издание/ Наумов А.Н., Вендров А.М., Иванов В.К. и др./ Под ред. Наумова А.Н. - М.: Финансы и статистика, 1991 - 180 стр.
10. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. Пер. с англ. - М.: Финансы и статистика, 1990.- 319с.
11. Тельнов Ю.Ф., Сорова А.А., Андреева Н.В. Проектирование баз знаний. Учебное пособие.- М.: МЭСИ, 1992.-100с.
12. Тельнов Ю.Ф., Диго С.М., Полякова Т.М. Интеллектуальные системы обработки данных. Учебное пособие.- М.: МЭСИ, 1989.- 102с.

13. Уотерман Д. Руководство по экспертным системам. / Пер. с англ.; Под. ред. Стефанюка В.Л. - М.: Мир, 1989.- 388 с.
14. Форсайт Р. Экспертные системы: принципы и примеры. / Пер. с англ. - М.: Радио и связь, 1987.
15. Цикритзис Д., Лоховски Ф. Модели данных. /Пер. с англ. - М.: Финансы и статистика, 1985. -344с.
16. Шлеер С., Меллор С. Объектно-ориентированный анализ: Моделирование мира в состояниях. Пер. с англ. - Киев: Диалектика, 1993. -240с.
17. Элти Дж., Кумбс М. Экспертные системы: концепции и примеры /Пер. с англ. - М.: Финансы и статистика, 1987.- 191 с.
18. Decision support systems. Putting theory into practice. Edited by R.H. Sprague, H. Watson. Prentice-Hall, 1993. -437p.
19. Gevarter W.B.. The Nature and Evaluation of Commercial Expert Systems. Building Tools.- Computer, May, 1987. p 24-41.
20. Martinson, F.R. Schindler. Organizational visions for technology assimilation: the strategic road to knowledge-based systems success. IEEE Transactions on engineering management, 1995, Vol 42, No 1, p 10 - 18.
21. Talebzaheh, Mandutianu S., Winner C.F. Countrywide Loan-Underwriting Expert System. AI Magazine, 1995, april, p. 51 - 64.

Глава 3. Реализация экспертных систем экономического анализа деятельности предприятия

3.1. Особенности экспертных систем экономического анализа

Архитектура экспертной системы экономического анализа (особенности формирования базы знаний, выбора методов логического вывода, пользовательского интерфейса) во многом зависит от целей и глубины анализа: внешнего (для сторонних организаций) или внутреннего (для самого предприятия).

Целью **внешнего анализа** предприятия является определение общего состояния предприятия, т.е. интерпретация его экономического положения с точки зрения выявления возможностей эффективного взаимодействия с ним внешних организаций. Таким анализом занимаются банки при выдаче кредитов, инвесторы при размещении своего капитала, фирмы-партнеры при осуществлении закупочно-сбытовой или подрядной деятельности.

Наиболее зарекомендовавшим себя методом внешнего анализа, интегрирующим множество различных экономических показателей предприятия, служит **рейтинговый метод**, который формирует “снизу-вверх” интегральную оценку финансового состояния предприятия.

Примером экспертной системы внешнего анализа является система оценки кредитоспособности предприятия EvEnt (рис.3.1) [11], в которой общая оценка кредитоспособности суммируется из оценок отдельных факторов с учетом их весовой значимости на общую оценку по формуле:

$$O_i = \sum_j (W_{ij} * O_{ij}), \text{ где}$$

O_{ij} - оценка влияния j - го фактора на i - й вышестоящий фактор по некоторой числовой шкале, а W_{ij} - вес (коэффициент) влияния j - го фактора на i - й фактор.

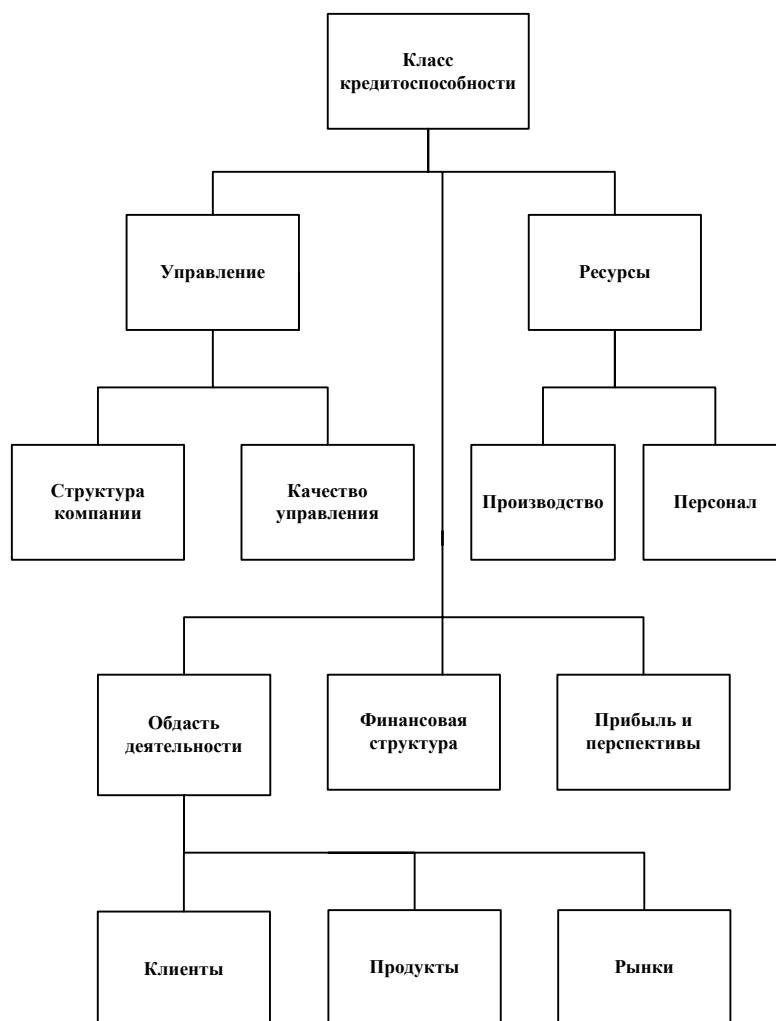


Рис 3.1. Оценка кредитоспособности предприятия

Правила базы знаний оценивают отдельные факторы, реализуя так называемый **дизъюнктивный (независимый) подход** к построению правил. Примеры правил имеют следующий вид:

IF: Управление = “удовлетворительно”
 THEN: Фин.состояние += “удовлетворительно” cf 40

IF: Финансовая структура = “удовлетворительно”
 THEN: Фин.состояние += “удовлетворительно” cf 60

IF: Ресурсы = “удовлетворительно”
 THEN: Фин.состояние += “удовлетворительно” cf 50

IF: Качество управления = “удовлетворительно”
 THEN: Управление += “удовлетворительно” cf 80

IF: Структура управления = “удовлетворительно”
 THEN: Управление += “удовлетворительно” cf 90

и т.д.

В качестве весов в данном примере используются факторы уверенности, поэтому вместо формулы при разработке аналогичной системы может применяться формула объединения факторов уверенности для дизъюнкции (см. § 2.4).

В результате внедрения системы EvEnt для 80 % ситуаций решения формируются без экспертов. Если раньше на оценку предприятия экспертом банка требовалось в среднем 2-3 недели, то после внедрения экспертной системы основные затраты стали связываться со сбором и вводом исходных данных в течение 2-3 дней, а собственно оценка предприятия занимает порядка 20 минут. При этом стоимость экспертизы в среднем сократилась с 10000 долларов до 1000 долларов.

В качестве метода внешнего анализа может применяться также **метод классификации ситуаций**, когда по множеству признаков классификации, в качестве которых в данном случае выступает множество показателей деятельности предприятия, последовательно строится дерево решений, отражающее эту классификацию. В случае индуктивного вывода дерево решений строится по обучающей выборке автоматически.

Пример классифицирующего дерева решений для оценки кредитоспособности предприятий, построенного в системе индуктивного вывода ИЛИС [5] по обучающей выборке из 100

реально оцененных в одном из банков предприятий, представлен на рис.3.2. В обучающей выборке в качестве классифицирующих признаков использовались коэффициенты автономии, мобильности, отношения собственных и заемных средств, покрытия, абсолютной ликвидности, ликвидности, а также качественные признаки репутации и величины. Классообразующим признаком является признак “Класс кредитоспособности” (1 - высший класс, 5 - низший класс). В результате обобщения примеров обучающей выборки часть признаков была формально отброшена: коэффициенты мобильности, ликвидности и величина предприятия, причем по различным ветвям дерева решений наблюдалась различная последовательность классификации. Для каждой отдельной ветки дерева решения строится правило, в котором все признаки классификации последовательно связываются в конъюнкцию (&) факторов левой части правила (так называемый **конъюнктивный подход**), например:

IF: Кпокр. ≥ 1.55 & Кпокр. < 2 & Репутац=3 & Ксоб.заем. ≥ 0.625 & Ксоб.заем. < 0.75 & Кавтоном. ≥ 0.375 & Кавтоном. < 0.6
 THEN: Кред.сп = 1

Ограничения метода классификации ситуаций (конъюнктивного подхода) по сравнению с рейтинговым методом (дизъюнктивным подходом) при использовании правил принятия решений связаны с необходимостью жесткого задания всех признаков классификации по соответствующему пути дерева решения. Отсутствие хотя бы одного из признаков может привести к неудаче логического вывода.

Для **внутреннего экономического анализа** свойственен поиск направлений повышения эффективности деятельности предприятия, т.е. диагностика узких мест и определение рекомендаций по их устранению. В основе диагностики лежит **метод последовательной декомпозиции** “сверху-вниз” или дезагрегации “целое - часть”, когда проблема последовательно разбивается на подпроблемы, пока на каком-либо уровне не станет ясным, какая подпроблема в действительности имеет место. Примером применения декомпозиционного метода к построению экспертных систем служит система внутреннего финансового анализа FINEX (рис.3.3) [11].

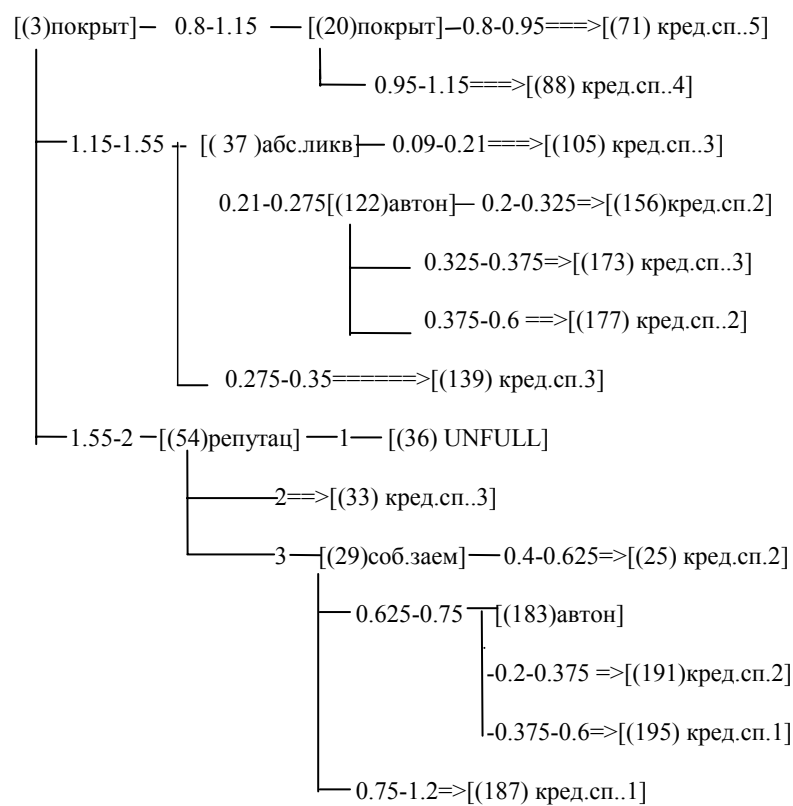


Рис.3.2. Дерево решений «Определение класса кредитоспособности»

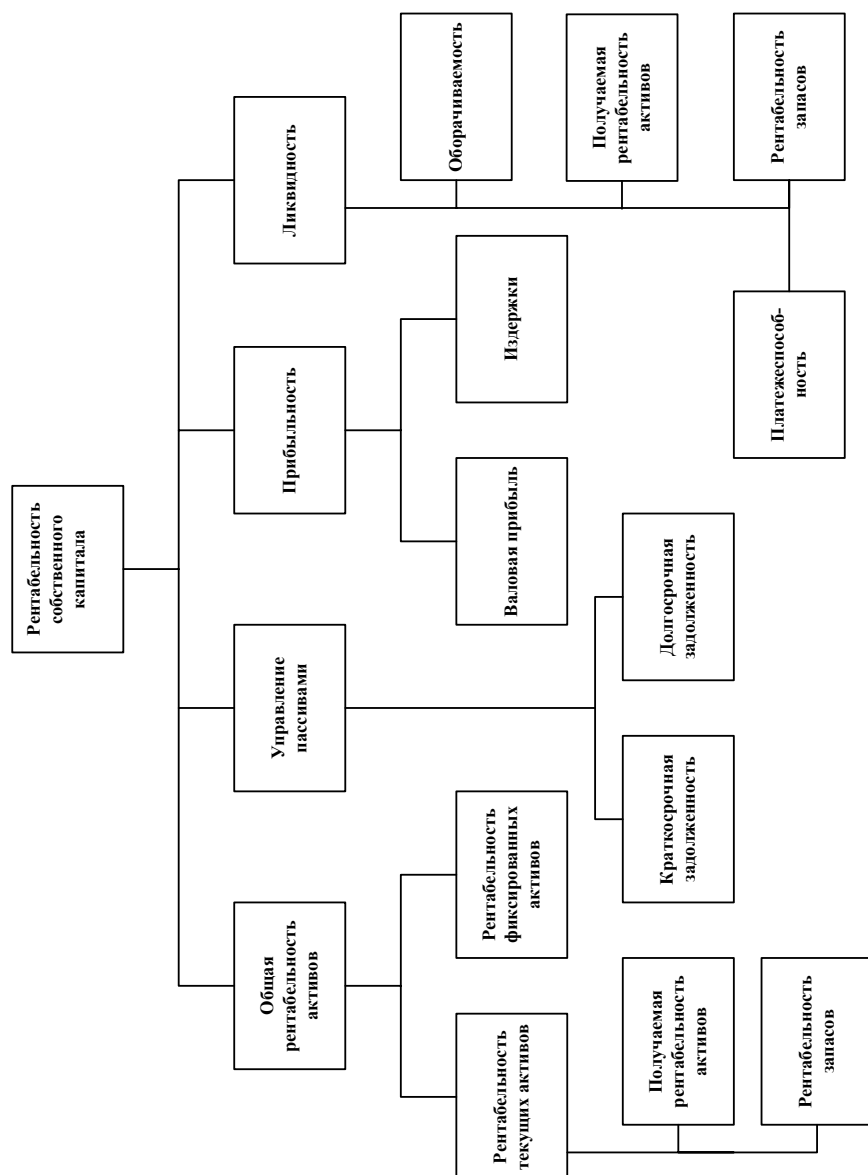


Рис.3.3. Диагностика рентабельности предприятия

В случае применения экспертной системы внутреннего финансового анализа FINEX экспертиза осуществляется автоматически на основе введенных данных финансовой отчетности. При этом анализ финансовых показателей выполняется последовательно по принципу “сверху-вниз” и “слева-направо” в соответствии с деревом взаимосвязи показателей. В случае обнаружения некоторого “узкого места” (неудовлетворительного значения показателя) может быть включен диалоговый режим работы экспертной системы, в котором система последовательно опрашивает пользователя на предмет качественной оценки тех или иных процессов, причем вопросы задаются в порядке, зависящем от предыдущих ответов.

Для проведения комплексного экономического анализа предприятия возможно комбинированное применение различных методов структурной организации баз знаний, интегрируемых в рамках одной информационно-аналитической системы. Примером такой комплексной системы может служить Интегрированная консультирующая система (Integrated Consulting System - ICS), разработанная фирмой SRI International с целью анализа конкурентных возможностей и планирования стратегии производства в отрасли, характеризуемой высокой степенью дифференциации продуктов [11].

Система ICS отличается от других аналогичных информационно-аналитических систем возможностью интеграции методов количественного и качественного анализа, в которой отражаются знания о технологиях, производителях, рынках. Система ICS внедрена в компании полипропиленовой промышленности, действующей в 97 регионах с 300 видами продуктов и 24 технологическими процессами.

Программная реализация системы ICS (архитектура системы представлена на рис. 3.4) выполнена в виде комбинации экспертных систем и аналитико-алгоритмических процедур, использующих совместно глобальную базу данных, которая включает знания о предметной области в шести фреймах («рынок», «технология», «компания», «рынок-технология», «компания-рынок», «технология-компания»). Обмен текущими результатами выполняемых работ программных компонентов осуществляется через доску объявлений.

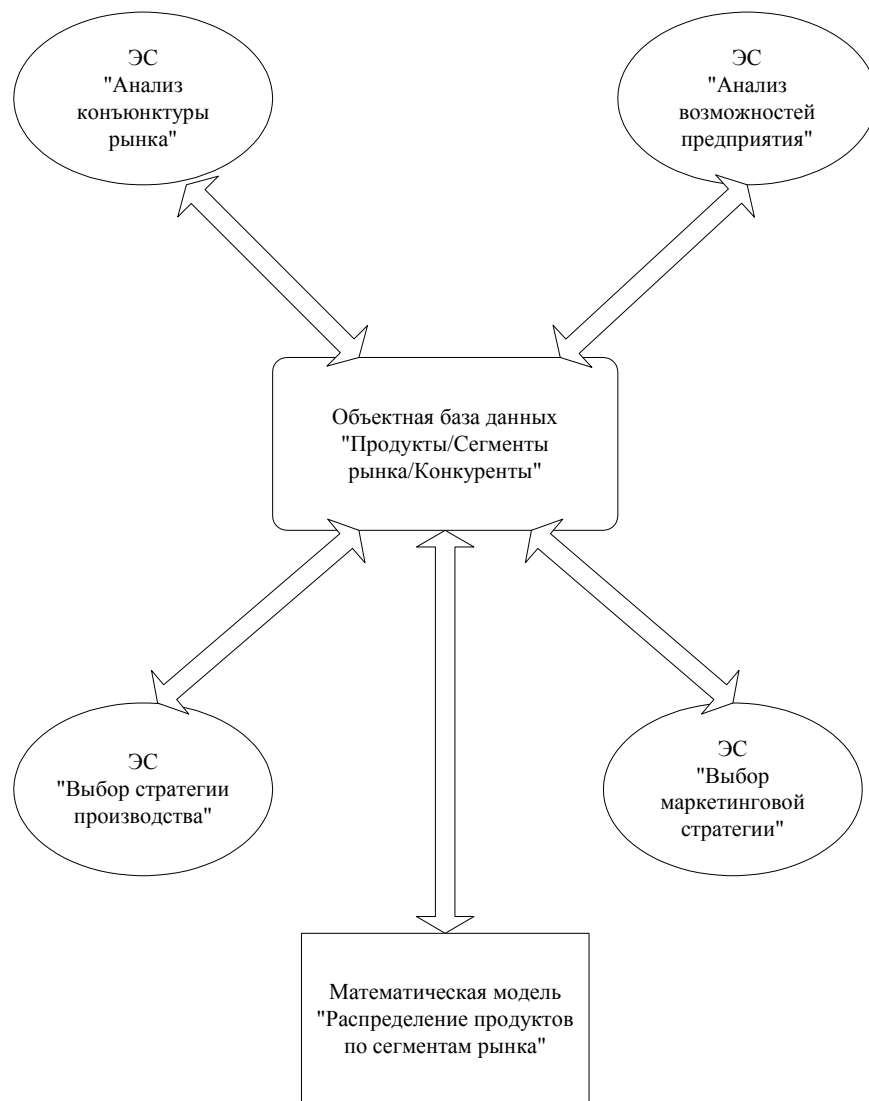


Рис. 3.4. Архитектура интегрированной консультирующей системы ICS

Интегрированная консультирующая система ICS выполняет следующие функции:

- Определение потенциальных возможностей (долей рынка) конкурентов на рынке (экспертная система «Анализ конъюнктуры рынка»).
- Определение потенциальных возможностей (долей на рынках) компании (экспертная система «Анализ возможностей предприятия»).
- Получение возможного распределения долей рынков предприятия (оптимизационная модель).
- Уточнение распределения долей рынков предприятия с учетом эвристических правил (экспертная система «выбор стратегии производства»).
- Выбор конкурентной стратегии поведения на рынке в части захвата или отдачи доли рынка (экспертная система «выбор маркетинговой стратегии»).

Первые две функции реализуются аналитическими экспертными системами, которые основаны на применении рейтингового метода, в частности для получения взвешенных оценок используется метод анализа иерархий Саати. Две последние функции, требующие проведения анализа складывающихся рыночных ситуаций, основаны на применении экспертных систем с классифицирующими правилами.

3.2. Экспертная система анализа финансового состояния предприятия

Рассматриваемый в учебном пособии исследовательский прототип экспертной системы "Финансовый анализ предприятий" реализован в среде интегрированного ППП Интерэксперт (GURU) с использованием рейтингового, классификационного и декомпозиционного подходов [3].

Функциями экспертной системы финансового анализа предприятия являются:

- Ввод и проверка правильности составления бухгалтерской отчетности;
- Анализ финансового состояния предприятия;

- Анализ результатов финансово-хозяйственной деятельности предприятия и диагностика эффективности использования ресурсов.

В ходе ввода и проверки бухгалтерской отчетности осуществляется логический контроль зависимостей различных статей баланса предприятия, отчета о финансовых результатах и их использовании, справки к этому отчету и приложений к балансу. При этом правила логического контроля выполняются последовательно по декомпозиционному методу.

Анализ финансового состояния предприятия предполагает комплексную рейтинговую и классификационную оценку платежеспособности и финансовой устойчивости предприятия.

Анализ результатов финансово-хозяйственной деятельности предусматривает оценку важнейших показателей рентабельности и оборачиваемости капитала. Диагностика эффективности использования ресурсов сводится к поиску отклонений в использовании основных и оборотных средств от нормативных значений с последующей декомпозицией анализа.

Общая схема оценки различных показателей в процессе анализа финансового состояния предприятия, реализованная в структуре базы знаний экспертной системы, представлена в виде дерева целей, связывающего цели финансового анализа со значениями финансовых показателей (рис. 3.5). Реализация набора правил в среде ППП ИНТЕРЭКСПЕРТ (GURU) для данного дерева целей приводится в приложении 2.

Рассмотрим более подробно реализацию данной экспертной системы. Оценка финансового состояния предприятия может быть одной из следующих:

- отличной (оптимальный, но редко встречающийся результат, абсолютно устойчивое финансовое состояние, все показатели удовлетворяют нормальным ограничениям, ухудшения финансового состояния по сравнению с предыдущими периодами деятельности не отмечено);
- хорошей (нормальное финансовое состояние, чаще встречающийся результат, большинство показателей удовлетворяют нормальным значениям, платежеспособность предприятия в таком случае гарантирована);

- удовлетворительной (часто встречающийся результат), большинство показателей удовлетворяют нормальным значениям или имеют место незначительные отклонения, может быть отмечена тенденция к ухудшению финансового состояния, финансовая устойчивость удовлетворительная);
- неудовлетворительной:
 - а) в допустимых пределах (большинство показателей не удовлетворяет нормальным значениям, однако не замечено тенденции к общему ухудшению финансового состояния, могут быть лишь отдельные незначительные ухудшения по отдельным показателям, возможно постепенное улучшение положения);
 - б) кризисной (предприятие на грани банкротства: баланс абсолютно неликвиден, финансовые показатели не удовлетворяют нормальным значениям, необходимы немедленные меры по санации предприятия).

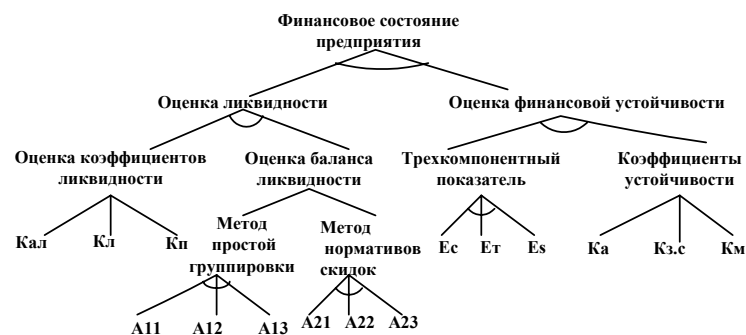


Рис.3.5. Дерево целей оценки финансового состояния предприятия

При проведении экспертизы необходимо учитывать ряд факторов, оказывающих влияние на финансовое состояние предприятия:

- уровень и состав (по срокам погашения) обязательств;
- уровень и состав (в разрезе ликвидности) собственных средств;
- общая тенденция развития предприятия (изменения в структуре отчетности по сравнению с предшествующими периодами деятельности);
- особенности вида деятельности (отрасли) предприятия;
- общеэкономические факторы (например, уровень инфляции).

Так, для получения итоговой оценки финансового состояния требуется выделить промежуточные этапы анализа (подцели):

- оценка ликвидности - это анализ платежеспособности предприятия с точки зрения краткосрочной перспективы, оценка средств предприятия, достаточных для уплаты долгов по всем краткосрочным обязательствам и одновременного бесперебойного осуществления процесса производства и реализации продукции;
- оценка финансовой устойчивости представляет собой анализ с точки зрения долгосрочной перспективы, оценку финансовой обеспеченности бесперебойного процесса деятельности в перспективе и степени зависимости предприятия от внешних кредиторов и инвесторов. Финансовую устойчивость предприятия определяет соотношение основных и оборотных средств (запасов и затрат) и величин собственных и заемных источников их формирования.

При нахождении значений сформулированных подцелей и оценке финансового состояния в целом следует принимать во внимание, что невозможно точно установить, насколько удовлетворительными (неудовлетворительными) являются те или иные значения показателей. Кроме того, сами нормативные значения некоторых финансовых показателей, на основании которых строятся выводы, зависят от множества обстоятельств: структуры баланса, особенностей деятельности предприятия, экономической ситуации в стране. Поэтому, несмотря на то, что все возможные исходы решения проблемы могут быть описаны, они оцениваются с некоторой степенью уверенности (достоверности). При этом задача оценки финансового состояния предприятия приобретает нечеткий характер.

Оценка ликвидности (платежеспособности) получается как по обобщенным показателям (финансовым коэффициентам) в результате их проверки на соответствие нормативным ограничениям, так и на основе соотношений статей актива и пассива баланса (ликвидности баланса). Для оценки такого баланса кроме простого (приближенного) метода группировки существует более точный метод нормативов-скидок. При расчете показателей платежеспособности используются следующие формулы:

1) Финансовые коэффициенты ликвидности:

а) коэффициент абсолютной ликвидности ($K_{ал}$):

$$K_{ал} = \frac{D}{Kt + Rp + Ko},$$

где D - денежные средства и краткосрочные финансовые вложения;

Kt - краткосрочные кредиты и заемные средства;

Rp - расчеты и прочие пассивы;

Ko - ссуды, не погашенные в срок.

б) коэффициент ликвидности ($Kл$):

$$Kл = \frac{Ra}{Kt + Rp + Ko},$$

где Ra - денежные средства, расчеты и прочие активы.

в) коэффициент покрытия ($Kп$):

$$Kп = \frac{(Z - Sf) + Ra}{Kt + Rp + Ko},$$

где Z - запасы и затраты;

Sf - расходы будущих периодов;

Ra - денежные средства, расчеты и прочие активы.

Оценка коэффициентов ликвидности для промышленных предприятий осуществляется на основе проверки ограничений.

Коэффициент абсолютной ликвидности:

- $K_{ал} \leq 0,2$ - неудовлетворительное значение с уверенностью 50%;
- $K_{ал} > 0,2$ и $\leq 0,7$ - “удовлетворительно” с уверенностью 30%;
- $K_{ал} > 0,7$ и $\leq 1,0$ - “удовлетворительно” с уверенностью 50%;
- $K_{ал} > 1,0$ - “удовлетворительно” с уверенностью 30%.

Коэффициент ликвидности:

- $Kл \leq 0,6$ - неудовлетворительное значение с уверенностью 60%;
- $Kл > 0,6$ и $\leq 0,8$ - “удовлетворительно” с уверенностью 40%;
- $Kл > 0,8$ и $\leq 1,0$ - “удовлетворительно” с уверенностью 60%;
- $Kл > 1,0$ - “удовлетворительно” с уверенностью 40%.

Коэффициент покрытия:

- $Kп \leq 1,0$ - неудовлетворительное значение с уверенностью 70%;
- $Kп > 1,0$ и $\leq 2,0$ - “удовлетворительно” с уверенностью 50%;
- $Kп > 2,0$ и ≤ 3 - “удовлетворительно” с уверенностью 70%;
- $Kп > 3,0$ - “удовлетворительно” с уверенностью 50%.

2) Ликвидность баланса:

а) Расчет ликвидности баланса простым методом:

- наиболее ликвидные активы

$$A1 = D;$$

- быстрореализуемые активы

$$A2 = Ra - D;$$

- медленнореализуемые активы

$$A3 = Z + Sf + FT + RTa ,$$

где Sf - расходы будущих периодов,

FT - долгосрочные финансовые вложения,

RTa - расчеты с участниками (учредителями);

- наиболее срочные пассивы

$$П1 = Rp + Ko;$$

- краткосрочные пассивы

$$П2 = Kt - Ko;$$

- долгосрочные и среднесрочные пассивы

$$П3 = KT + RTp ,$$

где KT - долгосрочные и среднесрочные кредиты и заемные средства,

RTp - расчеты за имущество;

б) Расчет ликвидности баланса методом нормативов скидок:

$$A1 = D;$$

$$A2 = 0,82a + 0,7\Gamma + 0,5(Z - Sf - \Gamma),$$

где Γ - готовая продукция;

$$A3 = 0,22a + 0,3\Gamma + 0,5(Z - Sf - \Gamma) + FT + RTa;$$

$$П1 = 0,8Rp + Ko;$$

$$П2 = 0,2Rp + Kt - Ko;$$

$$П3 = KT + RTp.$$

Для обоих методов оценки ликвидности баланса рассчитываются платежные излишки/недостатки по следующим формулам:

- платежный излишек/недостаток наиболее ликвидных активов:

$$A11(A21) = A1 - П1;$$

- платежный излишек/недостаток быстрореализуемых активов

$$A12(A22) = A2 - П2;$$

- платежный излишек/недостаток медленнореализуемых активов

$$A13(A23) = A3 - П3.$$

В общей оценке платежеспособности наибольшее значение придается оценке ликвидности баланса по сравнению с оценкой коэффициентов ликвидности, например, факторы уверенности назначаются в соотношении 2 к 1. В оценке ликвидности баланса метод нормативов-скидок играет уточняющую роль, он лишь немного увеличивает хорошие значения и уменьшает плохие значения ликвидности.

Оценку баланса ликвидности можно представить в виде следующей матрицы ("+" - избыток, "-" - недостаток) :

A11	A12	A13	Оценка	Уверенность
+	+	+	Удовл.	100
+	+	-	Удовл.	80
+	-	+	Удовл.	75
-	+	+	Удовл.	70
-	-	-	Неудовл.	100
-	-	+	Неудовл.	75
-	+	-	Неудовл.	70
+	-	-	Неудовл.	60

Оценка финансовой устойчивости формируется из оценок трехкомпонентного показателя типа финансовой ситуации, определяющего покрытие основных и оборотных средств собственными и заемными финансовыми источниками, а также оценок коэффициентов устойчивости по сравнению с нормативными значениями. Основное влияние на финансовую устойчивость при этом оказывает оценка трехкомпонентного показателя примерно в соотношении 2 к 1. При расчете показателей финансовой устойчивости используются следующие формулы:

1) Трехкомпонентный показатель типа финансовой ситуации:

- излишек/недостаток собственных оборотных средств:

$$+Ec = Ic - F - Z;$$

- излишек/недостаток собственных, долгосрочных и среднесрочных заемных средств:

$$+Em = Ic - F - Z + Kt;$$

- излишек/недостаток общей величины основных источников формирования запасов и затрат:

$$+E = Ic - F - Z + Kt + KT;$$

Полученное значение трехкомпонентного показателя может характеризовать состояние финансовой устойчивости как:

- абсолютно устойчивое: запасы минимальны, если
 $+E_c > 0$ и $+E_t > 0$ и $+E > 0$;

- нормальное: нормальные величины запасов, если
 $+E_c < 0$ и $+E_t > 0$ и $+E > 0$;

- неустойчивое: избыток запасов, если
 $+E_c < 0$ и $+E_t < 0$ и $+E > 0$;

Такое состояние можно восстановить путем привлечения долгосрочных и среднесрочных кредитов и заемных средств или обоснованным снижением уровня запасов;

- кризисное: чрезмерная величина неподвижных и малоподвижных запасов, если
 $+E_c < 0$ и $+E_t < 0$ и $+E < 0$;

В этом случае денежные средства, краткосрочные ценные бумаги и дебиторская задолженность не покрывают даже кредиторской задолженности и просроченных ссуд.

2) Коэффициенты финансовой устойчивости предприятия:

- коэффициент автономии:

$$K_a = \frac{Ic}{B},$$

где B - итог актива и пассива баланса;

- коэффициент соотношения заемных и собственных средств:

$$K_{з/с} = \frac{K + Rp}{Ic};$$

- коэффициент маневренности:

$$K_m = \frac{Ic - F}{Ic};$$

- коэффициент обеспеченности запасов и затрат собственными источниками финансирования:

$$Kоб = \frac{Ис - F}{Z}.$$

Оценка коэффициентов финансовой устойчивости для промышленных предприятий осуществляется на основе проверки ограничений:

Коэффициент автономии:

- $Kа \leq 0,2$ - неудовлетворительное значение с уверенностью 100%;
- $Kа > 0,2$ и $\leq 0,5$ - “удовлетворительно” с уверенностью 75%;
- $Kа > 0,5$ - “удовлетворительно” с уверенностью 100%.

Коэффициент соотношения заемных и собственных средств:

- $Kз/с < 0,7$ - удовлетворительное значение с уверенностью 100%
- $Kз/с \geq 0,7$ - неудовлетворительное значение с уверенностью 100%

Коэффициент маневренности:

- $Kм = 0,5$ - удовлетворительное значение с уверенностью 100%;
- $Kм < 0,5$ - неудовлетворительное значение с уверенностью 100%.

Коэффициент обеспеченности запасов и затрат собственными источниками финансирования:

- $Kоб \leq 0,6$ - неудовлетворительное значение с уверенностью 100%;
- $Kоб > 0,6$ и $\leq 0,8$ - удовлетворительное с уверенностью 100%;
- $Kоб > 0,8$ - удовлетворительное с уверенностью 80%.

Оценки показателей платежеспособности и финансовой устойчивости корректируются в зависимости от оценки тенденции развития предприятия (динамики значений показателей). При этом производится сравнение показателей отчетного периода со средней величиной этих показателей за предшествующий период

деятельности предприятия с учетом инфляционных процессов. В случае улучшения значений показателей коэффициент уверенности удовлетворительной оценки увеличивается, предположим, на 10%, а в случае ухудшения коэффициент уверенности соответственно уменьшается.

Проверка ограничений на значения отдельных показателей и их последующая оценка задается в виде правил базы знаний в следующей форме:

Если: < посылка > То: < заключение > Уверенность < значение > ,

например:

Если: $A11 > 0$ и $A12 > 0$ и $A13 > 0$

То: Ликвидность баланса = "Удовлетворительная"
Уверенность 100

В случае независимого воздействия на оценку некоторой целевой переменной нескольких показателей (соответственно нескольких правил, оценивающих эту переменную) коэффициент уверенности (КУ) итоговой оценки формируется рейтинговым методом по формуле нормализованного сложения:

$$КУ_{рез\ i} = КУ_{рез\ i-1} + КУ_{фактора\ i} - КУ_{рез\ i-1} * КУ_{фактора\ i} / 100$$

$$(КУ_{рез\ 1} = КУ_{фактора\ 1})$$

Например, коэффициент уверенности оценки ликвидности на основе значений финансовых коэффициентов ликвидности рассчитывается следующим образом:

Кал = 1,1 --> Ликвидность коэф-в = "Удовлетвор."
Уверенность 30

Кл = 1,5 --> Ликвидность коэф-в = "Удовлетвор."
Уверенность 40

Кп = 3,1 --> Ликвидность коэф-в = "Удовлетвор."
Уверенность 50

Ликвидность коэф-в = "Удовлетвор." Уверенность 79

Влияние нескольких факторов на оценку некоторого показателя финансового состояния можно также представить в виде

правил базы знаний, в посылке которых находится конъюнкция данных факторов, например:

IF: Ликвидность коэф-в = "Удовл." AND Ликвидность баланса = "Удовл."

THEN: Платежеспособность = "Удовл." cf 100

IF: Ликвидность коэф-в="Неудовл." AND Ликвидность баланса="Удовл."

THEN: Платежеспособность = "Удовл." cf 80

IF: Ликвидность коэф-в="Удовл." AND Ликвидность баланса="Неудовл."

THEN: Платежеспособность = "Удовл." cf 40

IF: Ликвидность коэф-в="Недовл." AND Ликвидность баланса="Недовл."

THEN: Платежеспособность = "Неудовл." cf 100

IF: Платежеспособность="Удовл." AND Финанс.устойчивость = "Удовл."

THEN: Финансовое состояние = "Удовл." cf 100

IF: Платежеспособность="Неудовл." AND Финанс. устойчивость="Удовл"

THEN: Финансовое состояние = "Удовл." cf 50

IF: Платежеспособность="Удовл." AND Финанс. устойчивость="Неуд."

THEN: Финансовое состояние = "Удовл." cf 20

IF: Платежеспособность="Неудовл." AND Финанс.устойчивость="Неуд."

THEN: Финансовое состояние = "Неудовл." cf 100

Объединение коэффициентов уверенности значений показателей для таких правил осуществляется следующим образом:

- из коэффициентов уверенности посылки правила выбирается минимальный;

- коэффициенты уверенности посылки и заключения правила объединяются по формуле:

$$\text{КУ правила} = \text{КУ посылки} * \text{КУ заключения} / 100 .$$

Например, коэффициент уверенности удовлетворительного финансового состояния предприятия на основе значений платежеспособности и финансовой устойчивости рассчитывается следующим образом:

Платежеспособность = "Удовл." Уверенность 60

Финанс. устойчивость = "Неудовл." Уверенность 70

Посылка правила	Уверенность 60
Заключение правила	Уверенность 20
<hr/>	
Финансовое состояние = "Удовл."	Уверенность 12

Решение задачи оценки финансового состояния предприятия с помощью экспертной системы сводится к выборке из базы знаний на основе известных исходных данных взаимосвязанной цепочки правил логического вывода заключения (рис.3.5).

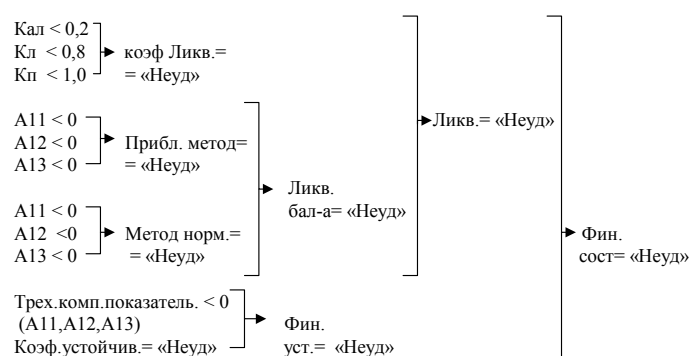


Рис.3.5. Пример прямой цепочки вывода оценки финансового состояния предприятия

В процессе эксплуатации экспертной системы допускается настройка базы знаний на особенности предприятия путем интерактивного введения нормативных ограничений и

оценивающих коэффициентов уверенности. Программная реализация предусматривает интегрированное использование экспертной системы совместно с расчетными процедурами, использующими необходимую информацию непосредственно из базы данных.

3.3. Экспертная система анализа эффективности результатов финансово-хозяйственной деятельности предприятия

Для детального внутреннего анализа результатов финансово-хозяйственной деятельности предприятия предназначена экспертная система, последовательно анализирующая факторы, влияющие на рентабельность предприятия.

Рентабельность предприятия, являющаяся интегральной оценивающей характеристикой эффективности его финансово-хозяйственной деятельности, рассчитывается как отношение полученного дохода (прибыли) к средней величине использования ресурсов. Существуют множество показателей рентабельности, среди которых следует перечислить такие, как показатели рентабельности активов, текущих активов, реализованной продукции, собственного капитала, инвестиций. Рентабельность текущих активов тесно связана с такими экономическими показателями, как оборачиваемость средств и себестоимость продукции. Поэтому этот показатель выбран в качестве корневого в экспертной системе анализа результатов финансово-хозяйственной деятельности предприятия.

Общая схема оценки различных показателей в процессе анализа рентабельности, оборачиваемости средств и себестоимости продукции предприятия, реализованная в структуре базы знаний экспертной системы, представлена в виде дерева взаимосвязи показателей (агрегации “целое - часть”) на рис. 3.6., которое отражает последовательность диагностики результатов финансово-хозяйственной деятельности предприятия.

Рассмотрим взаимосвязи представленных в дереве показателей. Показатель рентабельности текущих активов ($R_{та}$), отражающий получение чистой прибыли (ЧП) на один рубль оборотных средств (ОБС):

$$Pma = \frac{ЧП}{ОбС}$$

может быть выведен через показатели оборачиваемости текущих активов (Ота) и рентабельности реализованной продукции (Ррп):

$$Pma = Ota * Ppn .$$



Рис. 3.7. Дерево целей “Оценка эффективности ФХД предприятия”

Оборачиваемость текущих активов определяется отношением выручки от реализации продукции к величине оборотных средств:

$$Oma = \frac{Bpn}{ОбС}.$$

В дальнейшем анализ оборачиваемости текущих активов детализируется по составным элементам оборотных средств. Для этого рассчитываются показатели оборачиваемости товарно-материальных запасов (готовой продукции и производственных запасов) и дебиторской задолженности.

Рентабельность реализованной продукции рассчитывается как отношение чистой прибыли к выручке от реализации продукции (Врп):

$$Ppn = \frac{ЧП}{Врп}.$$

Выручка от реализации продукции в стоимостном выражении складывается из затрат на производство реализованной продукции (Зпр), или себестоимости, и результата от реализации, или прибыли (Пр):

$$Врп = Зпр + Пр.$$

Затраты на производство продукции включают прямые затраты (сырье и материалы, оплату труда рабочих, отчисления на социальные нужды, брак в производстве и прочие расходы) и постоянные затраты (общехозяйственные, общепроизводственные и коммерческие расходы).

Диагностика результатов финансово-хозяйственной деятельности предприятия осуществляется путем просмотра описанного дерева целей по принципу сверху-вниз и слева-направо. Каждому уровню дерева соответствует правило, сопоставляющее значение показателя с требуемым уровнем. Для сопоставления рассчитанных показателей со среднеотраслевыми значениями такие абсолютные показатели, как прямые и постоянные затраты, переводятся в относительные посредством отнесения их к объему реализации продукции в рублях. Значения рассчитанных показателей оцениваются также путем сравнения с аналогичными

значениями за предшествующий период, обеспечивая тем самым динамический анализ. Например:

IF: $R_{та} \geq R_{та\ отр.}$ и $R_{та} \geq R_{та\ пред.}$

THEN: Рентабельность тек.активов += "Удовл."

IF: $R_{та} \geq R_{та\ отр.}$ и $R_{та} < R_{та\ пред.}$

THEN: Рентабельность тек.активов += {"Удовл." cf 70, "Неудовл." cf 30}

IF: $R_{та} < R_{та\ отр.}$ и $R_{та} \geq R_{та\ пред.}$

THEN: Рентабельность тек.активов += {"Удовл." cf 30, "Неудовл." cf 70}

IF: $R_{та} < R_{та\ отр.}$ и $R_{та} < R_{та\ пред.}$

THEN: Рентабельность тек.активов += "Неудовл."

В приведенных правилах: $R_{та}$ - показатель рентабельности текущих активов предприятия, $R_{та\ пред.}$ - показатель рентабельности текущих активов предприятия за прошлый период, $R_{та\ отр.}$ - показатель рентабельности текущих активов по отрасли.

Сформированное качественное значение показателя фиксируется в специальной переменной "Финансово-хозяйственный результат". В случае неудовлетворительного значения того или иного показателя экспертная система осуществляет переход на анализ следующего уровня дерева показателей по функции KNOWN, например:

IF: Рентабельность тек.активов = "Неудовлетв."

AND KNOWN("Оборачиваемость тек.активов")

AND KNOWN("Рентабельность реал. продукции")

THEN: Финансово-хозяйственный результат+=

"Рентабельность тек.активов неудовлетворительна"

IF: Рентабельность тек.активов = "Удовлетв."

THEN: Финансово-хозяйственный результат+=

"Рентабельность тек.активов удовлетворительна"

IF: Оборачиваемость тек.активов = "Неудовлетв."

AND KNOWN("Оборачиваемость гот.продукции")

```

AND KNOWN("Оборачиваемость матер. запасов")
THEN: Финансово-хозяйственный результат+=
      " Оборачиваемость тек.активов неудовлетворительна"

IF: Оборачиваемость тек.активов = "Удовлетв."
THEN: Финансово-хозяйственный результат+=
      " Оборачиваемость тек.активов удовлетворительна"

```

.....

Переход на следующий уровень анализа обеспечивается в результате использования функции "KNOWN ", которая заставляет искать значение заданной в скобках переменной. По мере просмотра дерева целей в переменной "Финансово-хозяйственный результат" последовательно накапливаются диагностические сообщения о неудовлетворительности значений тех или иных показателей, которые в конце работы экспертной системы распечатываются для более глубокого анализа.

Что следует запомнить

Внешний экономический анализ проводится внешними для предприятия субъектами: инвесторами, кредиторами, партнерами, поставщиками, аудиторами, налоговыми и таможенными службами, страховыми организациями и т.д. Для внешнего анализа используются **интерпретирующие экспертные системы**.

Внутренний экономический анализ проводится внутри предприятия для выявления путей повышения эффективности деятельности. Для внутреннего анализа используются **диагностические экспертные системы**.

Для **интерпретации данных** используются **рейтинговый** или **классификационный методы**. Чем больше признаков (факторов) оценки ситуации, тем предпочтительнее рейтинговый метод по сравнению с классификационным.

Рейтинговый метод - получение суммарной оценки ситуации по ряду независимых признаков, при этом используется **дизъюнктивный подход** к построению правил. Этот метод неточный, гибкий.

Классификационный метод - ситуации классифицируются как различные комбинации значений признаков, при этом

используется **конъюнктивный подход** к построению правил. Этот метод точный, жесткий.

В основе диагностики экономической ситуации лежит **метод декомпозиции** “сверху-вниз” или **деагрегации** “целое - часть”. Переход от цели к анализу подцелей производится в случае отклонений цели от нормативного значения.

Литература

1. Банковское дело. / Под ред. О.И.Лаврушина. - М.: Банковский и биржевой научно-консультативный центр, 1992. - 428с.
2. Крейнина М.Н. Анализ финансового состояния и инвестиционной привлекательности акционерных обществ в промышленности, строительстве и торговле.- М.: АО “ДИС”, “МВ -Центр”, 1993. - 256 с.
3. Морозова Е.В., Семушкина Н.В., Тельнов Ю.Ф. Реализация экспертной системы финансового анализа и планирования деятельности предприятия в условиях неопределенности используемых знаний // Пятая национальная конференция с международным участием “Искусственный интеллект - 96”. Сб. научных трудов. - Казань, 1996, с. 210 -212.
4. Нечеткие множества в моделях управления и искусственного интеллекта. / Под ред. Поспелова Д.А. - М.: Наука, 1986. - 312 стр.
5. Попов Э.В., Шапот М.Д., Кисель Е.Б., Фоминых И.Б.. Статические и динамические экспертные системы, М: Финансы и статистика, 1996. -320с.
6. Системы управления базами данных и знаний: Справочное издание/ Наумов А.Н., Вендров А.М., Иванов В.К. и др., под ред. Наумова А.Н. - М.: Финансы и статистика, 1991. - 180 стр
7. Стоянов Е.А., Стоянова Е.С. Экспертная диагностика и аудит финансово-хозяйственного положения предприятия. - М.: Перспектива, 1992. - 90 с.
8. Тельнов Ю.Ф., Скорова А.А., Андреева Н.В. Проектирование баз знаний. Учебное пособие.- М.: МЭСИ, 1992.-100с.
9. Шеремет А.Д, Сайфулин Р.С. Методика финансового анализа предприятия. - М.: ИНФРА, 1996. - 176 с.
- 10.Humpert, P. Holey. Expert systems in finance planning. Expert Systems, May 1988, Vol 5, No 2, p. 78 - 100.

11. Management expert systems. / Ed. by C.J.Ernst. Addison-Wesley, 1988.
-210 p.
12. Talebzadeh, S. Mandutianu, C.F. Winner. Countrywide Loan-
Underwriting Expert System. AI Magazine, 1995, april, p. 51 - 63.

Глава 4. Реализация экспертных систем инвестиционного проектирования

4.1. Особенности экспертных систем инвестиционного проектирования

Инвестиционное проектирование, как правило, сводится к решению следующих задач:

- определение целей инвестирования капитала,
- оценка рынка и выбор типов инвестиций,
- проектирование портфеля инвестиций,
- мониторинг портфеля инвестиций.

Первая задача является аналитической и предполагает оценку финансового состояния инвестора, в частности, выявление свободных финансовых ресурсов, оценку допустимой степени риска при заданной структуре и ликвидности капитала, необходимой степени доходности и срочности для реализации поставленных инвестором целей, которыми могут быть: поддержание стабильного финансового положения, завоевание лидирующего положения на рынке, максимизация текущей прибыли, сочетания различных целей.

Для выполнения всестороннего анализа целей инвестора осуществляются финансовые расчеты, после которых выполняется набор правил, интерпретирующий полученные финансовые показатели. В зависимости от определенного рейтинга инвестора и намерений ему выдаются рекомендации относительно степени риска, доходности и срочности инвестиций.

Во второй задаче исследуется состояние рынка капитала на текущий момент времени, прогнозируется его развитие и возможность участия на нем инвестора. Свободные денежные средства могут быть вложены в различные активы: государственные облигации, депозиты коммерческих банков, акции и облигации коммерческих структур, недвижимость, валюту, драгоценные металлы и т.д. Каждое инвестиционное средство (финансовый инструмент - ФИ) имеет свою тенденцию развития, характеризуется определенными условиями вклада и получения дохода. На этом этапе ставится задача определить набор типов

инвестиций, наилучшим образом соответствующих сформированным целям инвестирования и ограничениям инвестора.

Третья задача проектирования портфеля инвестиций подразумевает выбор для каждой из рекомендуемых форм инвестиций конкретных видов и определение их наиболее эффективных сочетаний. На выбор видов инвестиций в случае предполагаемой покупки акций и облигаций коммерческих структур решающее воздействие оказывает отраслевая принадлежность и характеристики конкретных предприятий-эмитентов.

Выбор отраслевой направленности инвестиций предусматривает анализ коммерческой информации о рентабельности, объемах спроса и предложения, уровне конкуренции, характеристик этапа жизненного цикла выпускаемой продукции. Также анализируется информация в разрезе предприятий-эмитентов: рентабельность, доля рынка, конкурентоспособность, финансовая устойчивость и др. Естественно, что в этих условиях экспертная система должна иметь доступ к базам данных коммерческой информации и вычислять совокупный рейтинг предполагаемых инвестиций. Для построения диверсифицированных портфелей инвестиций важно так подбирать состав портфеля, чтобы в среднем он удовлетворял определенным требованиям доходности, риска и срочности с учетом налогообложения и инфляционных процессов.

Четвертая задача предполагает динамическое регулирование состава портфеля инвестиций (мониторинг), исходя из потребностей инвестора и изменения текущей ситуации на рынке капиталов. В рамках определенного процентного соотношения безрисковых и рискованных финансовых инструментов портфеля конкретные виды инвестиций динамически изменяются.

Формирование и управление портфелем инвестиций относится к задачам синтеза решений, зависящих от множества факторов, которые не могут быть заданы заранее каким-либо конечным множеством. Вследствие этого для экспертных систем проектирования инвестиций, относящихся к классу синтезирующих ЭС, характерны следующие особенности:

- взаимосвязанность процессов решения задач, в которых отдельные этапы итеративно связаны между собой;

- множественность источников знаний, рассматривающих процесс принятия решения с различных точек зрения;
- использование интегрированных баз данных, определяющих массовый характер многовариантных выводов решений;
- автоматическая качественная интерпретация объемных количественных данных, содержащихся в базе данных;
- сочетание формализованных и эвристических методов решения задач, когда эвристические методы упрощают перебор вариантов, уточняют полученные формализованными методами решения или восполняют пробелы в знаниях;
- изменение характера
- построения диалога "человек-ЭВМ" в сторону большей активности пользователя, варьирующего своими целями и ограничениями.

4.2. Экспертная система определения целей и типов инвестиций

В [14] описывается экспертная система инвестиционного проектирования, созданная с помощью генератора экспертных систем Intelligence Service (Франция), которая осуществляет функции определения целей инвестирования и выбора наиболее подходящих типов инвестиционных средств. Рассмотрим реализацию этой экспертной системы с точки зрения применения методов выбора типов инвестиционных средств, удовлетворяющих требованиям пользователя и, прежде всего, в части доходности и риска, которые являются практически общими для всех подобных систем.

Каждый тип инвестиционных средств (продукт), как правило, характеризуется определенным профилем в части доходности, риска, срочности, типа дохода и т.д. Суть правил экспертной системы заключается в объективном выявлении профиля инвестора (на что он может претендовать) и сопоставлении полученного профиля с профилем инвестиционного средства. При полном или частичном совпадении клиенту выдается список подходящих типов инвестиционных средств.

Экспертная система может работать в двух режимах:

- автономном, когда клиент доверяет экспертной системе выбор видов инвестиционных средств;

- ручном, когда клиент с помощью экспертной системы проверяет конкретный тип инвестиционного средства.

Решение задачи инвестиционного проектирования разбивается на следующие этапы:

- Определение целей инвестиции;
- Определение возможностей размещения инвестиций;
- Выбор типа инвестиций.

Каждый из этих этапов реализуется в виде самостоятельного набора правил (базы знаний). Переход от одного этапа решения к другому управляется метаправилами, содержащимися в специальном наборе правил.

Так, для запуска набора правил, определяющего цели инвестиции, должна быть выполнена проверка следующего метаправила:

```
IF: KNOWN("Цели_инвестиции_определены") = false
/*не известны */
THEN: Consult Purpose_of_investment
/* выполнить набор правил "Определение целей инвестиции" */
```

Набор правил "Purpose_of_investment" содержит следующие правила:

```
IF: KNOWN("Налоговые льготы") = true AND
    KNOWN("Тип_рынка ") = true AND
    KNOWN("Финансовое_основание") = true AND
    KNOWN("Тип_дохода") = true
THEN: Определение_целей_инвестиции = true

IF: KNOWN("Налоговые льготы") = false
THEN: PUTFORM Налоговые_льготы;
    GETFORM Налоговые_льготы

IF: KNOWN("Тип_рынка") = false
THEN: PUTFORM Тип_рынка;
    GETFORM Тип_рынка и т.д.
```

Суть этих правил заключается в следующем: последовательно проверяется известность значений переменных о требуемых налоговых льготах, типе рынка, типе получаемого дохода, финансовом основании клиента. В случае неизвестности

значений, клиенту выдается список возможностей (Putform), из которых он выбирает конкретные варианты (Getform).

В результате выполнения набора правил “Purpose_of_investment” цели инвестиции становятся известными, и управление возвращается в набор метаправил. Тогда выполняются метаправила запуска набора правил определения возможностей размещения инвестиций:

```
IF: Цели_инвестиции_определены = true /* определены */
THEN: Consult Possibility
/*выполнить набор правил “Определение возможностей размещения
инвестиций”*/
```

```
IF: Цели_инвестиции_определены = false /* не определены*/
THEN: Output “Цели инвестора не определены,
        конец консультации”
```

Аналогично при выходе из набора правил “Purpose_of_investment” в наборе метаправил проверяются значения переменной “Возможности_размещения_инвестиций_определены”:

```
IF: Возможности_размещения_инвестиций_определены = false
/* нет*/
```

```
THEN: Output “Возможности размещения инвестиций
        не определены, конец консультации”
```

```
IF: Возможности_размещения_инвестиций_определены = true /*да*/
AND Требуется_установить_профиль_клиента = true /* да*/
THEN: Consult Prove_profile_klient
/* выполнить набор правил “Определение профиля клиента”*/.
```

Переменная “Требуется_установить_профиль_клиента” принимает истинное значение в том случае, если у клиента отсутствует интерес к какому-либо конкретному типу инвестиционных средств:

```
IF: Интерес_клиента=false /* не определен */
THEN: Требуется_установить_профиль_клиента = true /* да*/
```

В противном случае у клиента запрашивается, какой конкретно его интересует тип инвестиционного средства, и

переменная “Требуется_установить_профиль_клиента”
устанавливается в ложное значение:

IF: Интерес_клиента=true /*определен */ AND
KNOWN(“Инвест_средство_клиента”) = true /* известно*/
THEN: Требуется_установить_профиль_клиента = false /* нет*/

При ложном значении переменной “Требуется установить профиль клиента” и известности конкретного требуемого клиентом инвестиционного средства при условии, что предварительно установлены возможности размещения инвестиций, набор правил “Prove_profile_klient” выполняется с целью доказательства возможности инвестиций в конкретный тип инвестиционного средства:

IF: Инвест_средство_клиента = “краткоср.сбережения”AND
Определение_возможностей_размещения_инвестиций=true
AND Требуется_установить_профиль_клиента =false
THEN: Consult Prove_profile_klient to test “Краткоср_профиль”
IF: Инвест_средство_клиента = “среднеср.сбережения” AND
Определение_возможностей_размещения_инвестиций=true
AND Требуется_установить_профиль_клиента =false
THEN: Consult Prove_profile_klient to test “Среднеср_профиль”
.....

Набор правил “Prove_profile_klient” имеет корневое правило, которое выполняется в случае неизвестности профиля потребности клиента в конкретном инвестиционном средстве:

IF: KNOWN(“Краткоср_профиль”) = true AND
KNOWN(“Среднеср_профиль”) = true AND
KNOWN(“Долгоср_профиль”) = true AND
KNOWN(“Льготный_профиль”) = true AND
.....
KNOWN(“Профиль_недвиж”) = true
THEN: Профиль_клиента_определен = true

Это правило заставляет экспертную систему выполнить проверку достижимости всех возможных профилей потребностей клиента.

Для каждого профиля выполняется некоторое подмножество правил, каждое из которых отражает зависимость профиля инвестиционного средства от сходных факторов. Обобщенное дерево целей определения каждого из перечисленных профилей инвестиционных средств (потребностей клиента) представлено на рис.4.1



Рис.4.1. Дерево целей “Оценка профиля клиента”.

Финансовое_основание: отличное, достаточное, недостаточное.

Тип_дохода: совокупный доход, текущая прибыль.

Допустимый риск: низкий, средний, относительно высокий, высокий.

Отношение_к_рынку: финансовый_рынок, рынок_недвижимости,...

Отношение_к_налоговым_льготам: заинтересованное, безразличное.

Перечисленные факторы участвуют в правилах в различных комбинациях, причем не обязательно все. Правила могут быть также скорректированы с учетом таких факторов, как возраст, социальный статус, семейное положение инвестора.

IF: Отношение_к_рынку = “Недвижимость” AND

Тип_дохода = “Совокупный доход” AND

Допустимый_риск = “Средний риск в течение длительного периода времени” AND
Финансовое_основание = “Отличное” AND
Сумма_инвестиции >= Средний_размер_инвестиции
THEN: Профиль_недвижимость = true

IF: Отношение_к_рынку = “Финансовый рынок” AND
Тип_дохода = “Текущая прибыль” AND
(Допустимый_риск <> “Средний риск в период более 3 мес.”
OR Допустимый_риск <> “Высокий риск в период более 3 мес.”)
AND Финансовое_основание = “Достаточное”
AND Сумма_инвестиции >= Средний_размер_инвестиции
THEN: Краткоср_профиль = true и т.д.

Правила, определяющие финансовое основание клиента, имеют следующий вид:

IF: Сбережения >= Нормативный_уровень AND
Портфель инвестиций = “есть” AND
Недвижимость = “есть”
THEN: Финансовое_основание = “отличное”

IF: Сбережения >= Нормативный_уровень AND
(Портфель инвестиций = “есть” OR
Недвижимость = “есть”)
THEN: Финансовое_основание = “достаточное”

IF: Сбережения < Нормативный_уровень
THEN: Финансовое_основание = “недостаточное”

4.3. Экспертная система формирования портфеля инвестиций

Одной из первых экспертных систем в области формирования портфелей инвестиций является система Plan-Power (1986) [13], архитектура которой включает три основные подсистемы (рис.4.2):

1. Диагностика существующей ситуации и целей;
2. Разработка плана инвестиций (портфеля);
3. Оформление плановой документации.

Пользователь вводит следующую информацию:

- Персональные данные;
- Финансовое положение;
- Цели инвестиции;
- Допустимую степень риска;
- Ожидаемый доход;
- Предполагаемую срочность инвестиций.

Исходная информация о ситуации клиента вводится во фреймы базы знаний. Для этого пользователю может быть задано до 80 страниц вопросов посредством специальных экранных форм (реально значительно меньше, так как целые разделы вопросов в зависимости от намерений пользователя могут быть опущены, а другие ответы могут быть получены по умолчанию). В базе знаний динамически поддерживается множество фреймов, описывающих инвестиционные продукты (средства). Во фреймах базы знаний отражается также макроэкономическая ситуация: степень инфляции, налоговые ставки, процентные ставки по кредитам, ГКО и т.д. Всего фреймовая база знаний насчитывает около 500 классов объектов с 2500 слотами и 1500 присоединенными процедурами (“когда-необходимо”). Фреймы организованы в иерархическую систему с наследованием свойств.

Диагностическая подсистема анализирует финансовую ситуацию клиента, прогнозирует ее развитие в будущем и формирует список достоинств и недостатков. Далее по этим данным формируются возможные цели инвестирования, которые сопоставляются с введенными пользователем. Рекомендации системы могут быть как простыми (например, задолженность клиента слишком высока), так и сложными (например, оценка платежей, размещения активов по инвестиционным категориям, достижимость клиентом целей).

Подсистема планирования (планировщик) включает модули размещения активов, страхования, налоговых платежей, продажи (погашения) активов, которые в процессе планирования координируются между собой (рис.4.3).

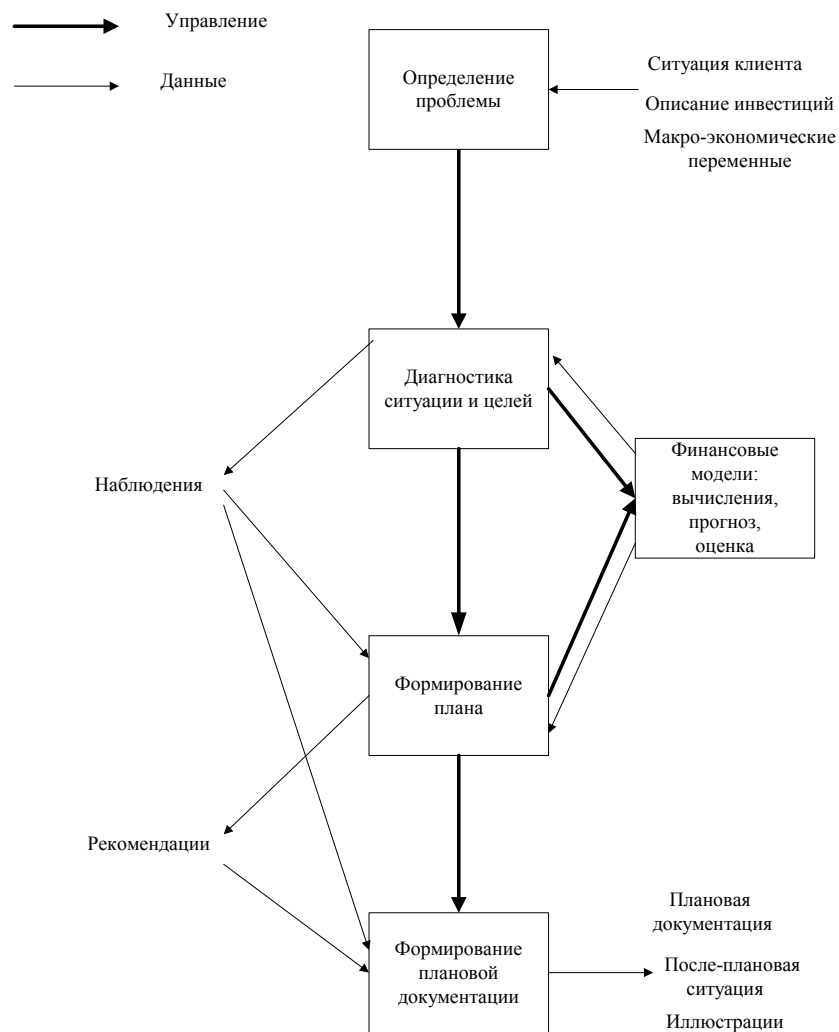


Рис.4.2. Архитектура ЭС PlanPower

Так, вначале формируется список возможностей, который представляет собой последовательность предполагаемых типов

инвестиций. Когда некоторый пункт добавляется в этот список, он проверяется на соответствие состоянию экономики, финансовым правилам, налоговым требованиям, предпочтениям клиентов, их состоянию, выявленному на первой стадии планирования.

Экспертная система расставляет приоритеты включенным пунктам в соответствии с текущей ситуацией и целями. Эти директивы вводятся планировщиком в начале процесса.

Модель размещения активов определяет пропорции инвестиций по категориям (например, материальным активам, недвижимости, вкладам с фиксированным доходом и т.д.) с учетом статуса инвестора (например, возраста, семейного положения, социального положения).

Сформированный список абстрактных возможностей в дальнейшем детализируется на предмет конкретных инвестиций, их объемов и временных характеристик. Пространство возможных решений и ограничения очень сложны. Конкретные действия должны оцениваться в свете информации, формируемой финансовыми моделями, общими принципами диверсификации активов по категориям и ограничениями, задаваемыми клиентом или планировщиком в терминах предпочтений или целей.

PlanPower решает проблему сокращения пространства возможных решений путем использования эвристик, оформляемых в виде правил. Прежде всего, возможности в списке упорядочиваются по некоторым эвристическим правилам, например, конкретные инвестиции в рамках некоторой категории упорядочиваются по степени доходности при соблюдении ограничений на заданную степень риска или, наоборот, по степени риска в рамках определенной доходности.

Некоторые правила имеют высший приоритет при использовании. Например, перед тем, как размещать активы, сначала всегда решаются задачи страхования. Другие правила позволяют на основе опыта определять количественные характеристики инвестиций без необходимости моделирования множества возможностей. Если необходимо, полученные результаты могут быть проверены путем моделирования. Полезно оставлять пользователю небольшую альтернативность выбора, когда система формирует несколько близких решений, которые трудно в дальнейшем дифференцировать. Сложность эвристической базы

знаний оценивается в 1200 обобщенных единиц знаний - эвристических процедур (чанков), объединяющих группы правил, что соответствует примерно 6000 простых правил с 3-5 аргументами в посылке правила и 1-2 заключениями.

Экспертные системы диагностики и планирования для обоснования решений обращаются к финансовым моделям, которые осуществляют расчеты денежных потоков, чистой стоимости, доходов, стоимости инвестиций и их логическую оценку. Финансовые модели содержат 1500 правил и процедур, которые завязаны в цепочки зависимостей так, что при изменении данных цепочки автоматически срабатывают.

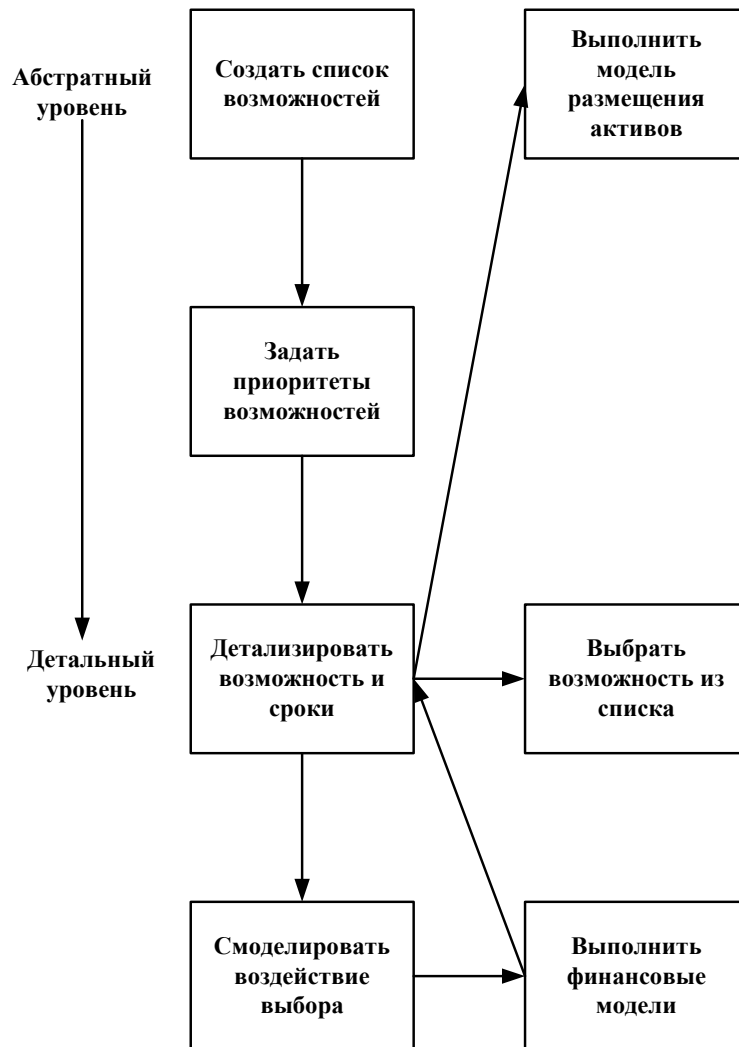


Рис.4.3. Схема работы планировщика

Подсистема оформления плановой документации включает:

- Результаты диагностики клиентской ситуации;
- Рекомендации по действиям;

- Объяснения рациональности этих действий;
- Релевантную финансовую информацию.

В результате клиенту выдается от 20 до 120 страниц текста инвестиционного плана и до 40 табличных и графических иллюстраций. Кроме того, выводится прогноз состояния клиента в результате моделирования выполнения плана.

Описанная экспертная система инвестиционного проектирования может работать как в автоматическом, так и интерактивном режимах. В интерактивном режиме (“что-если”) пользователь активно вмешивается в процесс решения задачи, может не соглашаться с рекомендациями системы и заставлять ее моделировать собственные предположения. Экспертная система функционирует как посредством прямого вывода планируемых рекомендаций, так и обратного вывода для проверки конкретных финансовых целей.

В целом следует отметить, что экспертная система PlanPower является сложным продуктом. Эта система разрабатывалась в течение 2 лет 10 инженерами по знаниям при участии 6 экспертов, а затем в течение одного года проходила тестовые испытания. Несомненное достоинство экспертной системы PlanPower – комплексность охвата функций инвестиционного проектирования.

4.4. Экспертная система мониторинга портфеля инвестиций

Для решения задач мониторинга портфеля инвестиций чаще всего используются методы технического анализа и прогнозирования рыночных цен, по которым можно предсказывать изменение доходности и надежности конкретных финансовых инструментов (ФИ) - ценных бумаг, валюты, драгметаллов и т.д. В отличие от фундаментального анализа, в основе которого лежит анализ экономической ситуации и/или финансового положения эмитентов, в техническом анализе рассматриваются тенденции в движении самих цен. В частности, технический анализ предполагает, что колебания цен периодически повторяются в соответствии с концепцией цикличности поведения рынка. Анализ статистических данных о движении цен позволяет выявить эвристические закономерности в изменении тенденций, которые хорошо представляются в виде правил принятия решений. Множество таких эвристических правил достаточно подробно

рассмотрено в литературе по техническому анализу рынка [1, 4]. Все эти правила могут быть собраны в виде общей базы знаний.

Рыночные тенденции по статистическим данным наглядно выявляются на графиках. Сущность правил технического анализа сводится к тому, чтобы на основе ежедневных данных о биржевых ценах на ФИ и объемах сделок, отображаемым на графиках, устанавливать (прогнозировать) изменения цен на рынке и выявлять ключевые дни их смены, в соответствии с которыми участники рынка занимают длинные (покупка ФИ по минимальной цене с целью продажи в последующем по максимальной цене) или короткие (продажа ценных бумаг (ЦБ) по максимальной цене с последующей скупкой по минимальной цене) позиции. Часто длинную позицию называют игрой на повышение, или “бычьей” позицией, а короткую позицию - игрой на понижение, или “медвежьей” позицией.

Технический анализ осуществляется на основе правил анализа следующих графиков:

1. гистограмм цен и оборота по ЦБ;
2. диаграмм скользящих средних;
3. графиков моделей движения цен (тренды, "треугольники", "флаги", "голова-плечи", "впадины/вершины", "блюдца").

Гистограммы цен и оборота позволяют определять ключевые дни поворота тенденций. Обычно на гистограммах (рис.4.4) для каждого дня торгов колебания цен показываются вертикальным отрезком, на котором отмечаются минимальная (нижняя точка отрезка) и максимальная (верхняя точка отрезка) цены, а также цена открытия торгов (черточка слева) и/или цена закрытия торгов (черточка справа).

Примеры различных характеристик дня торгов на рис.4.4 описывают следующие ситуации:

- Замкнутый день. Максимальная и минимальная цены находятся в пределах предыдущего дня. Заключительная цена не имеет большого значения. Если заключительная цена выше заключительной цены предыдущего дня, то следует покупать ФИ. Если заключительная цена ниже заключительной цены предыдущего дня, то следует продавать ФИ.
- Незамкнутый день. Высокая и низкая цены для данного дня превосходят таковые для предшествующего дня. Основное

значение имеет заключительная цена. Если заключительная цена растёт, то следует покупать ФИ, в противном случае - продавать.

- Сохранение тенденции. Максимальные, минимальные и заключительные цены одновременно возрастают или убывают. Если позиции участников рынка определены, то не предпринимать никаких действий. В противном случае при возрастающем тренде занимать длинную позицию, а при убывающем - короткую.
- Возможность смены тенденции. Цена закрытия совпадает с минимальной ценой при возрастающем тренде или цена закрытия совпадает с максимальной ценой при убывающем тренде. В первом случае необходимо готовиться к закрытию длинной позиции или занятию короткой, во втором случае следует поступать наоборот - закрывать короткую позицию или занимать длинную.
- Перемена хода закрытия цены. Цены сначала следуют в том же направлении, что и в предыдущий день, а затем изменяются на противоположные. Заключительные цены ниже соответствующих в предшествующий день. Предупреждающий сигнал о вероятном окончании тенденции цен.
- Ключевая перемена тенденции. Незамкнутый день и перемена заключительной цены. Заключительная цена дня выходит за пределы диапазона предшествующего дня. Сильный сигнал о конце тенденции цен, требуется смена позиции на рынке на противоположную. Ключевая перемена тенденции особенно сильно проявляется в критические дни, в когда отмечается рост объемов сделок, которые отражаются на отдельной гистограмме.

На гистограммах цен нередко возникают разрывы между отрезками цен смежных дней, которые также дают характеристики ценовых тенденций (рис. 4.4).

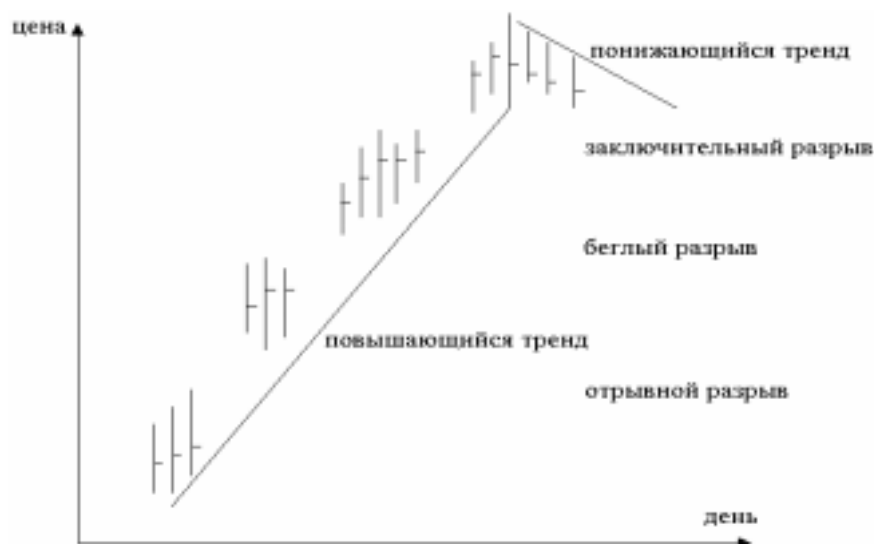


Рис.4.4. Ценовые тенденции.

- Обыкновенный (общий разрыв). Последующие дни ликвидируют разрыв. Тенденция к повышению или понижению не просматривается. Объем торговли низкий. Неустойчивый слабый рынок. Пассивность инвесторов, нет интереса.
- Отрывной разрыв. Резкий подъем или спад, ажиотажный спрос или предложение ФИ. Увеличивается объем сделок. Окончание фазы консолидации рынка, начало нового тренда. В последующие дни тенденция будет продолжаться.
- Беглый разрыв. Подъем или спад рынка, проявляется тенденция цен, объем торговых сделок средний. Образуется в середине тренда. Таким образом, можно предсказать окончание тренда. Обычно не покрывается в последующие дни.
- Завершающий разрыв. Завершающий разрыв (разрыв истощения) наступает после отрывного и беглого разрывов. Тенденция цен к повышению или понижению исчерпывает себя и меняется на противоположную. При подъеме уровень цен при закрытии следующего дня меньше нижнего уровня цен разрыва.

Полезным инструментом технического анализа может быть сравнение текущих рыночных цен со скользящими средними за разные периоды времени (рис. 4.5). В том случае, когда средние

цены лежат на графике ниже текущих заключительных цен, рынок характеризуется повышающимся трендом, а когда средние цены лежат выше текущих заключительных цен, рынок имеет понижающийся тренд. Это правило можно применить и для сравнения средних цен с минимальными (максимальными) текущими ценами или средними ценами, подсчитанными с другой периодичностью. Причем чем меньше период вычисления скользящей средней, тем более точно определение поворотных точек тенденции, а чем больше период, тем меньшее влияние на прогноз цены оказывает краткосрочная ситуация на рынке (уменьшается значение случайных факторов).

Рассмотрим также анализ трендов и построение соответственно линий поддержки и сопротивления (рис. 4.4). При возрастающем тренде график линии поддержки, например, линейной регрессии, как правило, строится по минимальным текущим ценам, а при понижающемся тренде - по максимальным текущим ценам.

Зная выражение функции регрессии, можно прогнозировать минимальные цены на ФИ при покупке в случае повышающегося тренда и максимальные цены на ФИ при продаже в случае понижающегося тренда. В том случае, когда заключительная цена на ЦБ становится меньше минимальной по уравнению регрессии при повышающемся тренде или больше максимальной при понижающемся тренде, происходит смена тенденции, и день, в который произошло изменение, должен быть запомнен как начало нового тренда. Возможен также стационарный (застойный) тренд, когда цены на ЦБ колеблются между относительно постоянными максимальными и минимальными ценами и соответственно горизонтальными линиями поддержки и сопротивления.

Обозначим используемые в эвристических правилах переменные:

XD_i - характеристика i -го дня;

$MIND_i$ - минимальная цена i -го дня;

$MAXD_i$ - максимальная цена i -го дня;

FD_i - заключительная цена i -го дня ;

VD_i - объем сделок i -го дня;

AD_i - скользящая средняя цена i -го дня;

$REGR_i = A * i + B$ - цена на графике линейной регрессии i -го дня ;

RAZR - количество разрывов.

Тогда, набор правил имеет следующий вид:

- R1: Если $MAXDi \leq MAXDi-1$ и $MINDi \geq MINDi-1$ и $FDi > FDi-1$
то $XDi +=$ "повышение" cf 20 /* замкнутый день */.
- R2: Если $MAXDi \leq MAXDi-1$ и $MINDi \geq MINDi-1$ и $FDi < FDi-1$
то $XDi +=$ "понижение" cf 20 /* замкнутый день */.
- R3: Если $MAXDi \leq MAXDi-1$ и $MINDi \geq MINDi-1$ и $FDi = FDi-1$
то $XDi +=$ "неизвестно" cf 100 /* замкнутый день */.
- R4: Если $MAXDi > MAXDi-1$ и $MINDi < MINDi-1$ и
 $XDi-1 =$ "повышение" и $FDi > FDi-1$,
то $XDi +=$ "повышение" cf 40 /* незамкнутый день */.
- R5: Если $MAXDi > MAXDi-1$ и $MINDi < MINDi-1$ и
 $XDi-1 =$ "повышение" и $FDi \leq FDi-1$ и $VDi > VDi-1$,
то $XD +=$ "понижение" cf 40 /* ключевой день на понижение */.
- R6: Если $MAXDi > MAXDi-1$ и $MINDi < MINDi-1$ и
 $XDi-1 =$ "понижение" и $FDi < FDi-1$,
то $XD +=$ "понижение" cf 40 /* незамкнутый день */.
- R7: Если $MAXDi > MAXDi-1$ и $MINDi < MINDi-1$ и
 $XDi-1 =$ "понижение" и $FDi \geq FDi-1$ и $VDi > VDi-1$,
то $XD +=$ "повышение" cf 40 /* ключевой день на повышение */.
- R8: Если $MAXDi > MAXDi-1$ и $MINDi > MINDi-1$ и $FDi > Fdi-1$
и $XDi-1 =$ "повышение",
то $XDi +=$ "повышение" cf 40 /* сохранение тенденции */.
- R9: Если $MAXDi < MAXDi-1$ и $MINDi < MINDi-1$ и $FDi < Fdi-1$
и $XDi-1 =$ "понижение",
то $XDi +=$ "понижение" cf 40 /* сохранение тенденции */.

- R10: Если $MAXDi > MAXDi-1$ и $MINDi > MINDi-1$ и $XDi-1 = \text{"понижение"}$,
то $XDi += \text{"повышение"}$ cf 50 /*смена тенденции - лок. мин.*/.
- R11: Если $MAXDi < MAXDi-1$ и $MINDi < MINDi-1$ и $XDi-1 = \text{"повышение"}$,
то $XDi += \text{"понижение"}$ cf 50 /*смена тенденции - лок. макс.*/.
- R12: Если $MAXDi > MAXDi-1$ и $FDi = MINDi$ и $XDi-1 = \text{"повыш."}$,
то $XDi += \text{"понижение"}$ cf 40 /* сигнал на понижение */.
- R13: Если $MINDi < MINDi-1$ и $FDi = MAXDi$ и $XDi-1 = \text{"понижение"}$,
то $XDi += \text{"повышение"}$ cf 40 /* сигнал на повышение */.
- R14: Если $MAXDi > MAXDi-1$ и $MINDi \geq MINDi-1$ и $XDi-1 = \text{"повышение"}$ и $FDi < FDi-1$,
то $XDi += \text{"понижение"}$ cf 30 /* сигнал на понижение */.
- R15: Если $MAXDi \leq MAXDi-1$ и $MINDi < MINDi-1$ и $XDi-1 = \text{"понижение"}$ и $FDi > FDi-1$,
то $XDi += \text{"повышение"}$ cf 30 /* сигнал на повышение */.
- R16: Если $MINDi > MAXDi-1$ и $RAZR = 0$ и $VDi = VDi-1$,
то $XDi += \text{"повышение"}$ cf 40
/* обычный разрыв на повышение */; $RAZR = RAZR+1$.
- R17: Если $MINDi > MAXDi-1$ и $RAZR = 1$ и $VDi \gg VDi-1$,
то $XDi += \text{"повышение"}$ cf 80
/* отрывной разрыв на повышение */; $RAZR = RAZR+1$.
- R18: Если $MINDi > MAXDi-1$ и $RAZR = 2$ и $VDi > VDi-1$,
то $XDi += \text{"повышение"}$ cf 60
/* беглый разрыв на повышение */; $RAZR = RAZR+1$.
- R19: Если $MINDi > MAXDi-1$ и $RAZR = 3$ и $VDi = VDi-1$,
то $XDi += \text{"повышение"}$ cf 40
/* заключ. разрыв на повышение */; $RAZR = 0$.

Для понижения цен можно построить правила R20 - R23, аналогичные правилам R16 - R19. Правила, сравнивающие скользящую среднюю цену с заключительной ценой, имеют следующий вид:

R24: Если $AD_i < FD_i$ и $XD_{i-1} = \text{"повышение"}$,
то $XD_i += \text{"повышение"}$ cf 60.

R25: Если $AD_i > FD_i$ и $XD_{i-1} = \text{"повышение"}$,
то $XD_i += \text{"понижение"}$ cf 60.

R26: Если $AD_i \geq FD_i$ и $XD_{i-1} = \text{"понижение"}$,
то $XD_i += \text{"понижение"}$ cf 60.

R27: Если $AD_i < FD_i$ и $XD_{i-1} = \text{"понижение"}$,
то $XD_i += \text{"повышение"}$ cf 60.

/* Сравнение цены линии поддержки с ценой дня*/

R28: Если $REGR_i \leq MIND_i$ и $XD_{i-1} = \text{"повышение"}$,
то $XD_i += \text{"повышение"}$ cf 80.

R29: Если $REGR_i > MIND_i$ и $XD_{i-1} = \text{"повышение"}$,
то $XD_i += \text{"понижение"}$ cf 80.

R30: Если $REGR_i > MAXD_i$ и $XD_{i-1} = \text{"понижение"}$,
то $XD_i += \text{"понижение"}$ cf 80.

R31: Если $REGR_i > MAXD_i$ и $XD_{i-1} = \text{"повышение"}$,
то $XD_i += \text{"повышение"}$ cf 80.

В приведенном наборе правил фактор уверенности играет роль степени важности используемого метода оценки тенденции. Так, в данном примере методы анализа разрывов и линий поддержки уравновешены и имеют большие веса по сравнению с методом анализа скользящих средних, который, в свою очередь, имеет больший вес по сравнению с анализом тенденций по двум смежным дням (убывание значения фактора уверенности от анализа тренда к анализу локальной ситуации).

В результате анализа различных графиков пользователю выдаются обобщенные рекомендации. Например:

Анализируемый день относится к дню, сохраняющему тенденцию на повышение, поскольку минимальная, максимальная и заключительная цены выше предыдущих, оборот рынка находится в пределах предшествующего дня (фактор уверенности - 40).

В предшествующие дни были отрывной и беглый разрывы, что свидетельствует о том, что рассматривается заключительная часть повышающегося тренда (фактор уверенности - 40).

Заключительная цена больше средней скользящей за три дня, что также свидетельствует о тенденции на повышение (фактор уверенности - 60).

Цены выше линии регрессии (фактор уверенности - 80).

Общий вывод: тенденция на следующий торговый день должна сохраняться с фактором уверенности 97%. (Объединение факторов уверенности выполнено по формуле нечеткой логики - см. параграф 2.4). Вместе с тем, попадание графика в завершающий разрыв свидетельствует о скором завершении тенденции.

Для более оперативного анализа рынка в течение торгового дня могут использоваться экспертные системы реального времени, например, на базе инструментального средства G2, которые позволяют учитывать при принятии решения фактор времени. Примерами таких правил являются:

Анализ изменений за время

Если скорость изменения курса валюты за последний час большая и
позиция длинная и
требуемый риск низкий

То хеджировать валюту в опцион

Временной лимит на рассуждение

Если любой опцион акций в дискаунте

То купить опцион и продать акции

(Затратить 20 секунд на принятие решений)

Регулярный мониторинг

Если изменение цены акции > 5% в течение последних 10 минут

То оценить опционы акции

(Выполнять каждые 5 минут)

Среди систем технического анализа финансовых инструментов, позволяющих объединять результаты прогнозирования на основе множества методов, можно назвать отечественную систему "Forex 94" (Уралвнешторгбанк), которая позволяет:

1. Оперативно анализировать финансовые рынки на основе сбора, хранения и обработки статистической информации по курсам валют, прогнозировать оптимальные цены и сроки торговых

сделок на поставку/закупку товаров в зависимости от курса валют;

2. Графически представлять информацию о текущем состоянии валютных рынков; исследовать в реальном масштабе времени конъюнктуру рынка с использованием 22 основных методов технического анализа и обобщенного индикатора "Радуга рынка";
3. Прогнозировать тенденции валютного рынка с использованием аппарата (MNP-1, MNN-1,2,3) нейронных сетей как на основе данных технического анализа, так и оригинальных индикаторов рынка;
4. Вести журнал сделок на продажу/покупку валюты, следить за позицией участников рынка и соблюдением установленных лимитов, использовать механизм "STOP LOSS" и "STOP PROFIT".

Система "Forex 94" ориентирована на работу с данными, поставляемыми любой финансовой информационной системой (REUTER Money 2000, TENFORE), поддерживающих режим динамического обмена данными в среде Windows. При отсутствии такой информационной системы пользователь может осуществлять ручной ввод данных.

Основные методы технического анализа в системе "Forex 94":

- Осциллятор (Oscillator - OSC);
- Пересечение средних (Moving Average Convergence - Divergence MACD);
- Индекс относительной величины цен (Relative StrengthIndex - RSI);
- Усредненное RSI (Moving Average RSI - MARSI);
- Метод Каири (Kairi - KRI);
- Накопление - раздача (ACD);
- Окончательный осциллятор (Ultimate Oscillator - UOS);
- Канал цен (Price Channel - PCU);
- Индекс канала твердых цен (Commodity Channel Index - CCI);
- Полоса Боулинджера (Boulinger Bonds - BBU);
- Процент R (Percent R - PCR);
- Индекс направления движения (Direction Movement Index - PDM);

- Ключ поворота (Key Reversal - KPV);
- Параболическая система цен (Price Time Parabolic system -PTP);
- Альфа-бета тренд (- ABT);
- Колебание цен (Fast Stochastics - PKF);
- Пересечение скользящих средних (Clossing moving averages - MAV);
- Линии поддержки и сопротивления;
- CPR;
- “Радуга рынка” – Rainbow (RNBW);
- EVS.

Набор методов технического анализа устанавливается автоматически или выбирается пользователем. Результаты работы нескольких методов могут отображаться на одном экране. Например, на одном графике (рис. 4.5) отображены гистограммы дней, по которым могут определяться ключевые повороты тенденций, скользящие средние, данные прогнозирования на основе применения нейронных сетей.

В качестве одного из основных инструментов технического анализа следует выделить интегральный показатель “Радуга рынка” (RNBW), который объединяет прогнозные данные всех используемых методов по оригинальной методике взвешивания получаемых результатов (верхняя полоса на рис. 4.5). На этой полосе желтым цветом отмечается тенденция рынка “вверх”, а чистый желтый цвет фиксирует завершение этой тенденции и скорый переход рынка “быков” на рынок “медведей”. Красный и розовый цвета характеризуют понижающийся ценовой тренд, а чистый малиновый цвет свидетельствует о завершении этой тенденции и скором переходе рынка “медведей” в рынок “быков”.

Для оценки текущего решения о покупке или продаже валюты пользователю выводится сигнал в виде окрашенной руки: желтой (Sell) для продажи, красной (Buy) для покупки, черный (Out) для воздержания от операций.

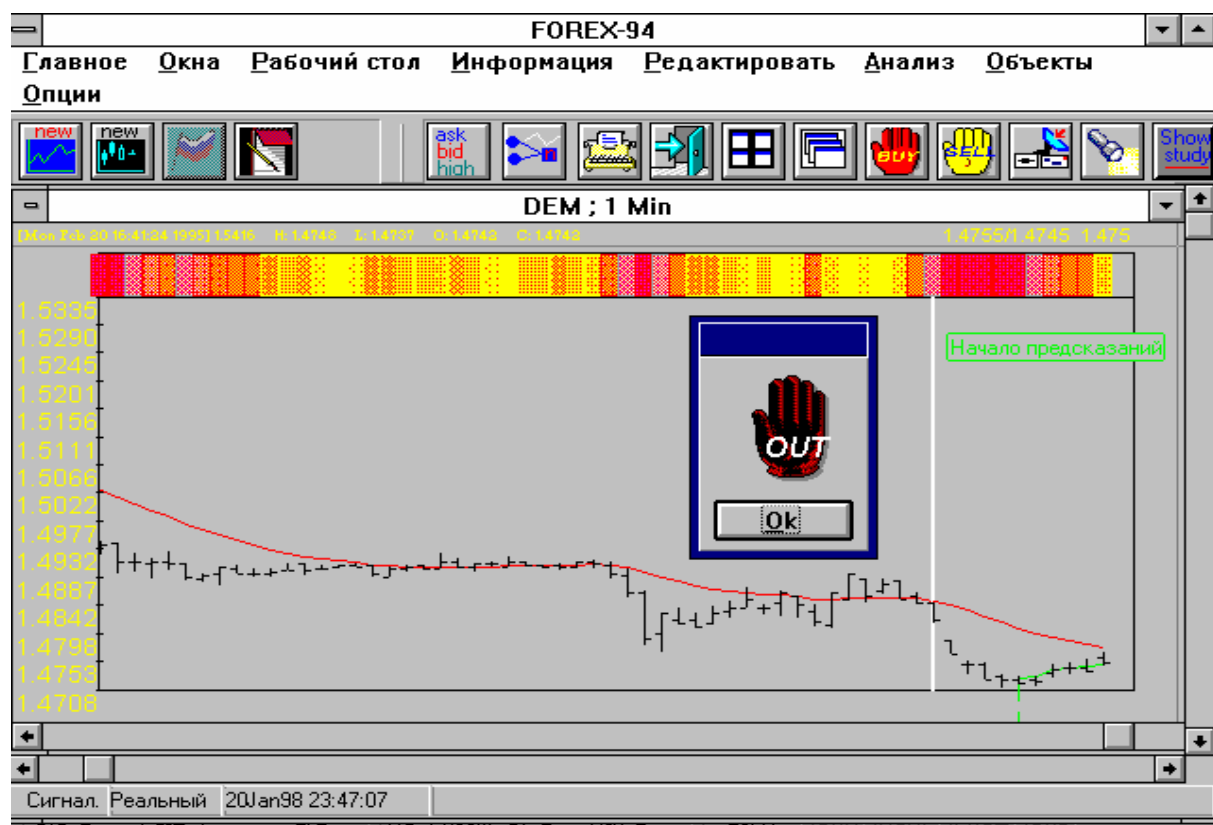


Рис.4.5. Пример работы системы FOREX94

Используемые в системе "Forex-94" нейронные сети прогнозируют решения о покупке/продаже валюты на основе их обучения по изменениям курсов валют за определенные периоды времени. Тогда прогноз сводится к сопоставлению произошедших изменений курсов валют с изменениями за эквивалентный период времени, "запомненными" в нейронной сети, по которым происходит активация решающей прогнозной функции (нейрона). В процессе обучения нейронная сеть использует алгоритм обратного распространения ошибок, который позволяет свести их к заданным интервалам погрешности. В качестве исходной информации для

прогноза нейронная сеть может использовать не только конкретные значения валютных курсов, но и сигналы методов технического анализа. Вместе с тем, нейронные сети “Forex-94” позволяют прогнозировать не сам валютный курс, а лишь его скользящее среднее за заданный интервал времени.

Система позволяет также обнаруживать нестабильность валютных курсов на основе методов теории динамического хаоса. Для применения этого метода необходимо задать порог нестабильности (обычно от 0 до 0.6), определяющий чувствительность системы к изменениям курса.

Параметры методов технического анализа могут выбираться пользователем по своему усмотрению или устанавливаться системой автоматически. В последнем случае оптимизация параметров и обучение нейронных сетей производится в соответствии с рядом встроенных критериев, например, получение максимальной прибыли при минимальных рисках.

По каждому из используемых методов технического анализа можно получать показатель доходности, который рассчитывается в результате применения соответствующих рекомендаций на конкретный день, а также определять достоверность этих рекомендаций (от 0 до 1).

Реализация нейронной сети прогнозирования рынка ГКО в ППП NeurOn-Line (Gensym) представлена в приложении 3. В приложении показана схема системы прогнозирования, исходные и результатные данные, конфигурация основных блоков нейронной сети.

Что следует запомнить

Инвестиционное проектирование - сложная проблемная область, для которой требуется работа нескольких динамически взаимодействующих экспертных систем.

Метаправила входят в базу знаний планировщика и управляют вызовом экспертных систем (наборов правил) в зависимости от событий, связанных с действиями пользователя или механизма вывода.

ЭС “Определение профиля клиента” формирует цели инвестиций на основе многофакторного анализа возможностей клиента и состояния рынка.

Оценка риска инвестиций осуществляется на основе статистических методов получения среднеквадратических отклонений доходности инвестиционных средств или путем оценки неопределенности ситуации с помощью методов нечеткой логики.

ЭС “Формирование портфеля инвестиций” осуществляет комбинаторный перебор инвестиционных средств в соответствии с сформированной целью инвестиций (доходностью, допустимой степенью риска и срочностью) на основе **базы данных** и **эвристических правил**. В процессе формирования портфеля подключаются **математические модели** расчета доходности и риска инвестиций.

Различные уровни абстракции используются в процессе работы механизма вывода: сначала портфель формируется на уровне типов инвестиционных средств (определяются пропорции типов инвестиционных средств), а затем на уровне конкретных инвестиционных средств.

Множество источников знаний ЭС “Формирования портфеля инвестиций”: “Налогообложение”, “Страхование”, “Кредитование” и др. используются для оценки инвестиций с позиции достижения общих целей портфеля.

Эвристические правила определяют приоритеты выбора типов и конкретных инвестиционных средств из имеющегося множества на основе отраженного в правилах базы знаний опыта экспертов.

Активный диалог пользователя предполагает возможность непосредственного участия пользователя в оценке альтернативных вариантов инвестиционных решений.

ЭС “Мониторинг портфеля инвестиций” осуществляет оперативный анализ эффективности портфеля инвестиций и формирует предложения по его обновлению.

Технический анализ выполняет анализ и прогнозирование доходности инвестиционных средств по характеристикам динамического тренда. При этом используются правила, учитывающие выявленные **статистические закономерности** (тренды) или **эвристики** (характерные повторения в движении цен).

Нейронные сети широко используются в мониторинге инвестиций для определения **рейтинга инвестиционных средств и прогнозирования цен.**

Литература

1. Инвестиционно-финансовый портфель /Отв. ред. Рубин Ю.Б., Солдаткин В.И. - М.: "Соминтек", 1993. - 752 с.
2. Котлер Ф. Основы маркетинга. Пер с англ. - М.: Прогресс, 1993. - 736с.
3. Левин Р., Дранг В., Эделсон Б. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на Бэйсике./ Пер. с англ.- М.: Финансы и статистика, 1991.- 239с.
4. Первозванский А.А., Первозванская Т.Н. Финансовый рынок: расчет и риск. - М.: Инфра-М, 1994. - 192 с.
5. Саати Т. Принятие решений: Метод анализа иерархий. - М.: Радио и Связь, 1993. - 254 с.
6. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. - М.: Финансы и статистика, 1990.- 319с.
7. Тюрин Ю.Н., Макаров А.А. Анализ данных на компьютере/ Под ред. В.Э.Фигурнова.- М.: Инфра-М.- М.: Финансы и статистика, 1994. - 384 с.
8. Четыркин Е. М. Методы финансовых и коммерческих расчетов. - М.: "Дело", "BusinessРечь", 1992. - 320с.
9. Шапиро В.Д. Управление проектами. - СПб.: "ДваТри", 1996 - 610 с.
- 10.Эддоус М., Стэнфилд Р. Методы принятия решений / Пер. с англ.- М.: Аудит, ЮНИТИ, 1997. - 590 с.
- 11.Энгел Л., Бойл Б. Как покупать акции. - М.: "Дело", 1993. - 272 с.
- 12.Blanning R.W. Foundations of expert system for management. - Koln,1990.-380 p.
- 13.Humpert, P. Holey. Expert systems in finance planning. Expert Systems, May 1988, Vol 5, No 2, p. 78 - 100.
14. Management expert systems. / Ed. by C.J.Ernst. Addison-Wesley, 1988 -210 p.

Глава 5. Реализация динамических экспертных систем управления бизнес-процессами

5.1. Особенности реализации динамических экспертных систем управления бизнес-процессами

Современные технологии бизнеса характеризуются высокой динамичностью, связанной с постоянно изменяющимися потребностями рынка, ориентацией производства товаров и услуг на индивидуальные потребности заказчиков и клиентов, непрерывным совершенствованием технических возможностей и сильной конкуренцией. В этих условиях требуется разработка информационных систем, которые, с одной стороны, должны оперативно поддерживать функционирование существующих бизнес-процессов, а, с другой стороны, давать прогнозы на долговременной основе об эффективности организации бизнес-процессов и рекомендации по их реорганизации. Первой цели соответствуют *динамические экспертные системы оперативного управления бизнес-процессами*, второй цели - *интеллектуальные системы моделирования бизнес-процессов*. В том и другом случае под бизнес-процессом будем понимать взаимосвязанную совокупность материальных, информационных, финансовых потоков или рабочих потоков (workflow), проходящих через взаимодействующие подразделения предприятия и направленных на удовлетворение потребностей клиента (изготовление товара или оказание услуги).

К динамическим бизнес-процессам на предприятии относят:

- управление заказами;
- управление запасами;
- оперативно-календарное планирование и управление производством.

Организация перечисленных бизнес-процессов определяется целями и задачами предприятия и зависит от конкретных видов выпускаемой продукции и оказываемых услуг. Вместе с тем, перечисленные бизнес-процессы в современных системах управления настолько сильно взаимосвязаны, как, например, в системах реализации заказов “Точно в срок” или ”Канбан” [4-6,10],

что процессы обработки заказов клиентов, производства, закупок и сбыта рассматриваются фактически как единый бизнес-процесс.

В отличие от описания предприятия на основе иерархической функциональной структуры, которую трудно объективно оценить, описание процессов позволяет точно представить цели, исследуемые характеристики (в том числе динамические) и конечные результаты каждого вида деятельности. Бизнес-процессы определяют прохождение потоков работ независимо от иерархии и границ подразделений, которые их выполняют, и представляют последовательность взаимосвязанных операций. Модель бизнес-процесса должна отражать как направление рабочих потоков, так и бизнес-правила обработки событий, в зависимости от которых выполняются операции [7-8, 11-12, 20]. Пример модели реализации заказов клиента показан на рис. 5.1.

На представленной модели бизнес-процесса в прямоугольниках показываются операции и подразделения, которые их осуществляют; в овальных прямоугольниках - события; слева и справа от операций - соответственно входные и выходные материальные, информационные и финансовые или рабочие объекты; в кружках снизу от операций - используемые постоянные ресурсы; простыми стрелками - рабочие потоки; утолщенными стрелками - потоки управления. Таким образом, представленная модель бизнес-процесса отражает динамические потоки событий (управления) и рабочих объектов (данных).

Типичными задачами, которые решаются динамическими экспертными системами оперативного управления бизнес-процессами, являются:

- Мониторинг бизнес-процессов и оперативное информирование лиц, принимающих решение, об отклонениях;
- Упреждающая диагностика, прогнозирование отклонений в параметрах операций бизнес-процессов;
- Динамическое распределение ресурсов в соответствии с изменяющейся обстановкой;
- Планирование действий, диспетчирование и составление сетевых графиков работ;
- Моделирование последствий принимаемых решений по изменению процессов.

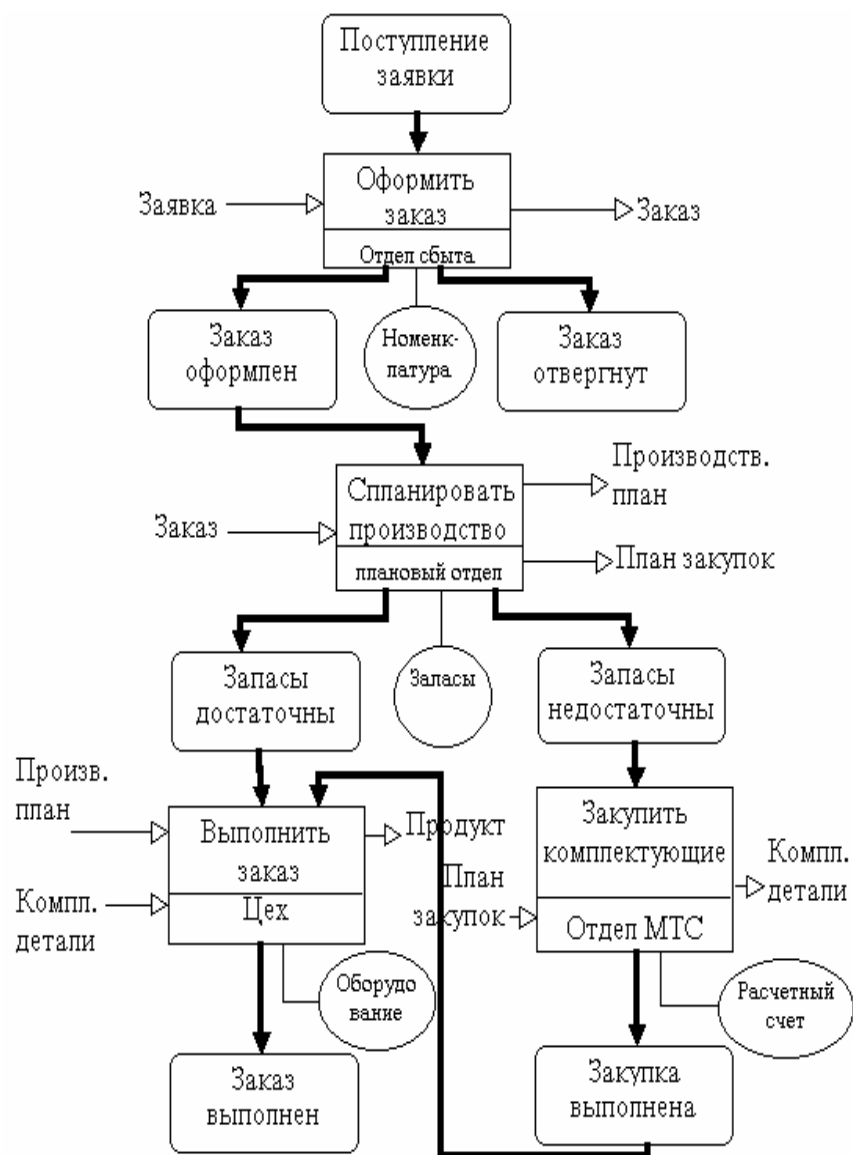


Рис.5.1. Модель бизнес-процесса реализации заказа клиента

В основные задачи интеллектуальных систем динамического моделирования для реинжиниринга (реорганизации) бизнес-процессов входят:

- Определение оптимальной последовательности выполняемых операций, которая приводит к сокращению длительности цикла изготовления и продажи товаров и услуг, обслуживания клиентов. Следствие оптимизации - повышение оборачиваемости капитала и рост всех экономических показателей фирмы.
- Оптимизация использования ресурсов в различных бизнес-процессах, которая приводит к минимизации издержек производства и обращения.
- Построение адаптивных бизнес-процессов, нацеленных на быструю адаптацию к изменениям потребностей конечных потребителей продукции, производственных технологий, поведения конкурентов на рынке и, как следствие, повышение качества обслуживания клиентов в условиях динамичности внешней среды.
- Отработка рациональных схем взаимодействия с партнерами, сочетания бизнес-процессов, которые оптимизируют финансовые потоки, обеспечение равномерности поступления и использования денежных средств.

Для обоих классов интеллектуальных систем характерны общие особенности реализации:

- Объектно-ориентированный характер модели проблемной области;
- Динамическое создание и уничтожение временных (рабочих) объектов;
- Динамическое поведение как постоянных объектов (ресурсов), так и временных (рабочих) объектов;
- Многоальтернативность выполнения бизнес-процесса в зависимости от возникающих событий;
- Анализ и обработка временных характеристик бизнес-процессов.

Перечисленные особенности динамических систем управления бизнес-процессами определяют выбор инструментальных программных средств компании Gensym [7-8]. Полный перечень программных продуктов представлен в таблице 5.1. [22].

Таблица 5.1.

Программные продукты компании Gensym

Продукт	Назначение
G2	Среда визуальной разработки интеллектуальных систем реального времени (ядро, открытое для всех остальных программных продуктов)
ReThink	Универсальный инструмент дискретного имитационного моделирования деловых и производственных процессов
eSCOR	Специализированный инструмент имитационного моделирования и поддержки процессов управления цепочками поставок
G2 Diagnostic Assistant (GDA)	Инструмент создания и поддержки систем управления технологическими процессами в реальном масштабе времени с диагностикой и обеспечением принятия решений
NeurOn-Line NeurOn-Line Studio	Инструмент поддержки процессов анализа и прогнозирования на основе нейронных сетей в реальном масштабе времени и off-line по собранным историческим данным
Operations Expert (Integrity)	Инструмент для разработки и поддержки приложений, связанных с управлением сетями
G2 Agent Development Environment (G2 ADE)	Среда разработки многоагентных приложений в распределенной сети
Intelligent Transaction Monitoring (ITM)	Агентная технология для отбора транзакций заданного профиля в сети, напр., финансовых транзакций
G2 Gateway, Telewindows G2 Weblink, G2 Javalink G2 ActiveXLink, G2 CORBALink G2 SAP Bridge	Набор интерфейсов с внешней средой для G2 – приложений (баз данных, web-приложений, ERP-систем и т.д.)

5.2. Экспертная система динамического управления запасами

Управление запасами представляет собой сложную экономическую задачу с противоречивыми критериями эффективности. С одной стороны, запасы призваны обеспечить экономическую безопасность бизнеса, связанную с неравномерностью спроса, в частности, в условиях увеличения сбыта запасы обеспечивают быструю реализацию поступающих заказов за счет накопленных запасов готовой продукции или сырья, необходимого для дополнительного производства. Дефицит запасов может привести к издержкам в связи с дополнительными затратами на реализацию поступившего заказа, а в некоторых случаях и к потере сбыта и заказчиков. С другой стороны, сверхнормативные запасы увеличивают себестоимость продукции за счет непроизводственных затрат на поставку, складирование и хранение запасов. По оценкам экономистов [5], каждый процент сокращения уровня запасов может быть приравнен к 10 процентному росту оборота, и если бы удалось поставить под контроль хотя бы 75 % колебаний уровня инвестиций в товарно-материальные запасы, экономика развитых стран никогда бы не испытывала кризисные явления.

Информационные системы управления запасами на базе экономико-математического моделирования и современных информационных технологий позволяют сократить затраты на поддержание дорогостоящих запасов в части:

- снижения затрат, связанных с созданием и хранением запасов;
- сокращения времени поставок;
- более четкого соблюдения сроков поставок;
- увеличения гибкости производства, его приспособляемости к условиям рынка;
- повышения качества изделий;
- увеличения производительности.

Особенность системы управления запасами заключается в том, что все звенья товародвижения взаимосвязаны. В условиях динамичности рынка нельзя рассматривать систему сбыта, систему производства и систему снабжения независимо друг от друга. Система управления запасами как раз связывает все перечисленные звенья в единое целое, рассматривая цепочки товародвижения как

единые бизнес-процессы, в которых отлаживаются взаимодействия между клиентами, подразделениями предприятия и его партнерами-смежниками и поставщиками.

Решение задачи определения уровня запасов зависит от следующих факторов:

- колебаний в сроках поставки сырья и материалов;
- колебаний заказов на готовую продукцию;
- выбранной стратегии обслуживания клиентов (работа на заказ клиента или на магазинную продажу).

В долгосрочном плане задача оптимального определения уровня запасов в силу вероятностного характера колебаний потребности в запасах может быть решена только на основе применения методов имитационного моделирования, что необходимо для определения характеристик производительности системы в условиях случайности процессов (см. параграф 5.3). В краткосрочном плане задача управления запасами решается в результате применения методов динамической диагностики и мониторинга, которые позволяют в каждый момент времени на базе оперативной информации о состоянии заказов, запасов и возможностях предприятия и поставщиков принимать оперативные решения по поддержанию запасов на необходимом уровне.

Одной из систем управления запасами является система, основанная на анализе допустимого уровня запасов (SIC – Statistical Inventory Control). По этой системе при поступлении на предприятие заказов осуществляется проверка уровня запасов. Если уровень запасов достигает некоторой пороговой отметки, то инициируется процесс пополнения запасов. В случае производственного характера предприятия цеха, изготавливающие готовую продукцию, пополняют эти запасы и осуществляют производственный процесс. В свою очередь, по цепочке назад формируются заказы к комплектующим подразделениям и так вплоть до отдела материально-технического снабжения. Таким образом, в каждом звене цепочки бизнес-процесса по одной и той же схеме анализируется уровень запасов, который динамически пополняется, обеспечивая бесперебойность снабжения каждого последующего звена бизнес-процесса. В случае посреднического характера предприятия число промежуточных звеньев сводится до одного, когда предприятие-посредник связывает клиентов-заказчиков с

поставщиками продукции (реализация этого случая в дальнейшем рассматривается в качестве примера в данном параграфе).

Основная проблема в системе управления запасами сводится к определению допустимого уровня запасов, который, с одной стороны, обеспечивает бесперебойность бизнес-процесса, а с другой стороны, минимизирует издержки, связанные с их поддержанием.

Различают следующие виды запасов [5,6,10]:

- Максимальный желательный запас - экономически целесообразный уровень в данной системе управления запасами, используется как ориентир при расчете объема заказа на поставку.
- Пороговый уровень запаса используется для определения момента времени выдачи очередного заказа на поставку.
- Текущий запас соответствует уровню запаса в любой момент времени.
- Гарантийный запас предназначен для непрерывного снабжения потребителя в случае задержек поставок и соответствует запасу, который может быть пополнен за среднее (максимальное) отклонение во времени поставки.

Существует множество вариантов систем управления запасами, обеспечивающих различные уровни запасов:

- Система с фиксированным размером заказа;
- Система с фиксированным интервалом времени между заказами;
- Система с фиксированным интервалом времени между заказами и проверкой порогового уровня запаса;
- Система “Минимум - максимум”.

Система с фиксированным размером заказа

Система с фиксированным размером заказа предполагает в моменты времени нарушения порогового уровня запасов при условии равномерности последующего расходования запасов выдачу фиксированного заказа на поставку продукции. Размер фиксированного заказа соответствует времени поставки, в течение которого уровень запасов снизится до гарантийного (рис. 5.2).

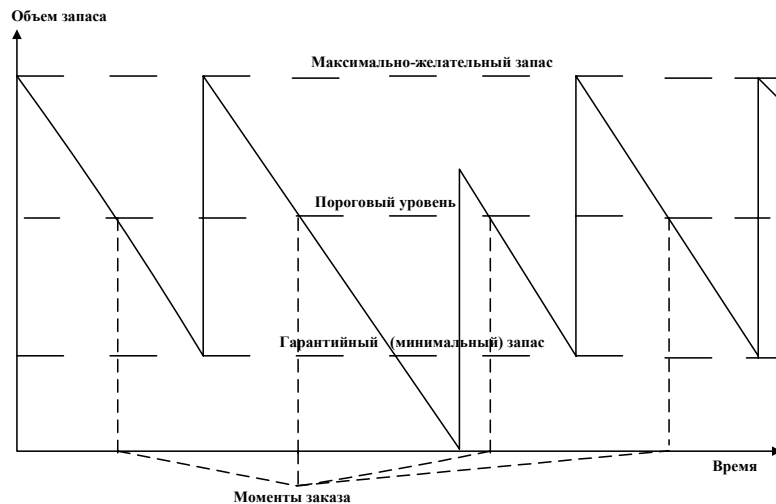


Рис.5.2. Система с фиксированным размером заказа

Расчетные формулы основных показателей уровней запасов имеют следующий вид:

$$\text{Оптимальный размер заказа на пополнение запаса} = \sqrt{\frac{2AS}{i}},$$

где А - затраты на поставку одного заказа,

S - потребность в заказываемом продукте (ежегодный спрос на запас продукции),

i - стоимость хранения единицы продукции в запасе.

Гарантийный запас = Максимальное время задержки в поставке*

Среднедневное потребление продукта;

Пороговый уровень запаса = Гарантийный запас + Среднее время поставки *Среднедневное потребление продукта;

Максимальный уровень запаса = Гарантийный запас + Оптимальный размер заказа на поставку

Правило принятия решения о выдаче заказа на поставку продукции имеет следующий вид:

Если (Текущий уровень запаса - Количество заказа клиента)
< Пороговый уровень запаса
То Выдать заказ на поставку

Система с фиксированным размером заказа позволяет оперативно отслеживать уровень запаса, более динамично реагируя на изменение спроса. При этом снижаются требования к поддержанию максимального уровня запаса. Нефиксированные интервалы поставки усложняют взаимодействие с постоянными поставщиками, следовательно, каждый раз может возникать задача выбора поставщика.

Система с фиксированным интервалом времени между заказами

Система с фиксированным интервалом времени между заказами используется, когда через равные интервалы времени при условии равномерности последующего расходования запасов инициируется заказ на пополнение запаса. Интервал времени рассчитывается таким образом, чтобы в момент достижения гарантийного запаса осуществлялось пополнение запасов. В том случае, если произойдет задержка в поступлении продукции, пополнение запаса должно произойти в момент исчерпания гарантийного запаса (рис. 5.3).

Расчетные формулы основных показателей уровней запасов имеют следующий вид:

$$\text{Фиксированный интервал между поставками} = \sqrt{\frac{2A}{iS}}$$

$$\text{Гарантийный запас} = \text{Максимальное время задержки в поставке} * \text{Среднедневное потребление товара};$$

$$\text{Максимальный уровень запаса} = \text{Гарантийный запас} + \text{Интервал между поставками} * \text{Среднедневное потребление товара};$$

$$\text{Размер заказа на пополнение запаса} = \text{Максимальный уровень запаса} - \text{Текущий запас} + \text{Среднее время поставки} * \text{Среднедневное потребление товара};$$

Правило принятия решений о выдаче заказа на поставку продукции имеет следующий вид:

Через фиксированный интервал между поставками
Выдать заказ на поставку

Система с фиксированным интервалом времени между заказами упрощает мониторинг состояния запасов и взаимодействие с постоянными поставщиками, а, следовательно, удешевляется система оформления заказов. При этом возрастают требования к поддержанию максимального уровня запаса и соответственно издержки хранения.

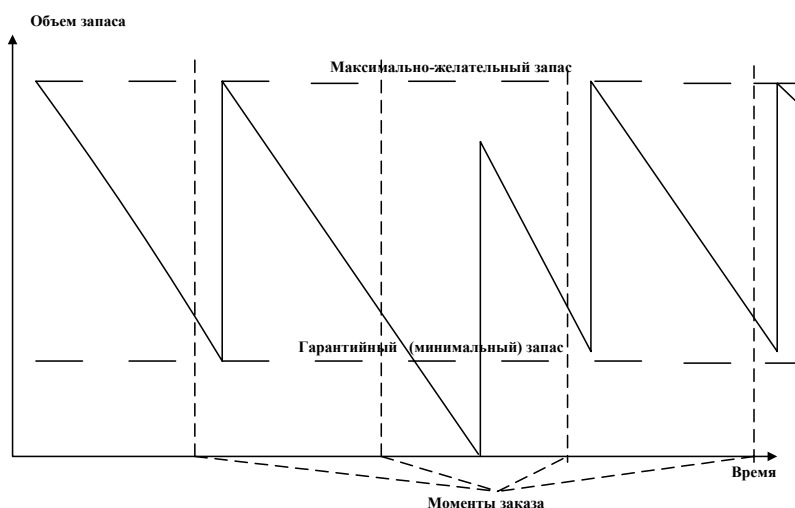


Рис.5.3. Система с фиксированным интервалом времени между заказами

Система с фиксированным интервалом времени между заказами и проверкой порогового уровня запаса

Система с фиксированным интервалом времени между заказами и проверкой порогового уровня запаса предусматривает выдачу заказа на пополнение запаса в фиксированные интервалы времени и при нарушении порогового уровня запаса. В этой системе заказы делятся на два класса: плановые и внеплановые. Плановые

заказы реализуются по системе управления запасами с фиксированным интервалом времени между заказами, внеплановые заказы выполняются при достижении порогового уровня запасов.

Расчетные формулы основных показателей уровней запасов имеют следующий вид:

Гарантийный запас = Максимальное время задержки в поставке *
Среднедневное потребление товара;

Пороговый уровень запаса = Гарантийный запас + Среднее время поставки *Среднедневное потребление товара;

Максимальный уровень запаса = Гарантийный запас +
Интервал между поставками *Среднедневное потребление товара;

Размер заказа на пополнение запаса (в фиксированные интервалы времени) = Максимальный уровень запаса - Текущий запас +
Среднее время поставки *Среднедневное потребление товара;

или

Размер заказа на пополнение запаса (при нарушении порогового уровня запаса) = Максимальный уровень запаса - Пороговый уровень запаса + Среднее время поставки*Среднедневное потребление товара.

Описанная комбинированная система в большей степени ориентирована на неравномерность спроса. Так, если фиксированные интервалы времени приурочены к моментам подачи постоянных заказов на продажу готовой продукции, то соответственно в эти же моменты времени будут инициироваться и заказы на пополнение готовой продукции от поставщиков. Если же в течение фиксированных интервалов времени приходят дополнительные заказы на готовую продукцию от клиентов, то пополнение запасов в этом случае происходит по системе проверки порогового уровня запасов.

Система “Минимум - максимум”

Разновидностью комбинированной системы управления запасами является система “Минимум - максимум”. Суть этой системы заключается в том, что в фиксированные интервалы времени производится проверка порогового уровня запасов

(минимума). Если текущий запас больше порогового уровня, то заказ на пополнение запаса не выдается. В противном случае выдается заказ на пополнение уровня запаса по формуле до максимального уровня.

Система “Минимум - максимум” применяется в случае большой стоимости материально-технического снабжения (стоимости оформления заказа на пополнение запасов и последующей поставки), сопоставимой со стоимостью хранения запасов и даже с издержками дефицита. Система позволяет сократить число поставок. Эта система может применяться в случае насыщенности рынка продукцией, предназначенной для пополнения запасов.

Сравнительный анализ описанных систем управления запасами показывает, что для нестабильного рынка или рынка, ориентированного на индивидуальные заказы клиентов, в наибольшей степени подходят системы, основанные на анализе порогового уровня запасов (с фиксированным размером заказа или комбинированная система), как наиболее динамические.

Реализация экспертной системы управления запасами на основе анализа порогового уровня с помощью инструментального средства G2

Функциональные возможности G2 описаны в таблице 5.2. Графическая модель проблемной области управления запасами для одного вида продукции в среде G2 представлена на рис. 5.4. Описание всех задействованных в модели проблемной области сущностей (items) дается на рис. 5.5. Примеры описания классов объектов (складов, заказов, списка заказов, дорог) заданы на рис. 5.6 - 5.7. Задание пиктограммы (икон) для объектов класса осуществляется с помощью графического редактора, а его аналитическое представление в разделе описание класса объектов “icon description” (пример определения пиктограммы класса объектов “склады” представлен на рис. 5.8). (см. glav5доп-файл с рис)

Оформление заказа клиента выполняет отдел сбыта, который вводит данные о заказе в экспертную систему. Описание действий по нажатию кнопки “заказать”, заданию размера заказа с помощью слайдера и нажатию кнопки “ОК” представлено на рис. 5.9. В частности, по нажатию кнопки “заказать” создается на ее рабочем

пространстве, названном “отдел сбыта” новый динамический объект “заказ”. С помощью слайдера (ползунка) осуществляется ввод размера заказа, который переносится с помощью кнопки “ОК” в последний заказ списка.

Введенный размер заказа анализируется с использованием правил на предмет возможности выполнения заказа (рис. 5.10). В случае возможности выполнения заказа осуществляется уменьшение запаса на складе. В случае невозможности отгрузки товара заказ окрашивается в красный цвет, а менеджер получает соответствующее сообщение.

Одновременно с помощью других правил производится анализ необходимости пополнения запасов на складе в случае нарушения порогового уровня (рис. 5.11). При этом состояние запаса на складе окрашивается на модели в желтый цвет (нарушен пороговый уровень) или в красный цвет (достигнут минимальный уровень запаса). В случае потребности в пополнении запаса запускается процедура (MOVE) поставки фиксированной партии товара на склад, которая имитирует процесс доставки груза от поставщика на склад (рис. 5.12).

Заметим, что при составлении базы правил использовались правила инициализации начального состояния системы при ее запуске (initially), правила, реагирующие на происходящие события (whenever – всякий раз, как происходит событие), и обобщенные правила (for any – для любого объекта класса). В принципе вместо правил обработки событий могут использоваться методы классов объектов. Пример использования методов представлен на рис. 5.13.

Экспертная система для управления запасами по системе с фиксированными интервалами выдачи заказа на поставку отличается от рассмотренной системы тем, что правила обновления текущего уровня запасов являются безусловными (unconditionally), которые запускаются на выполнение через фиксированный интервал сканирования правил (scanning), задаваемый в таблице атрибутов правила.

Экспертные системы для остальных систем управления запасами будут включать комбинацию условных и безусловных правил.

5.3. Интеллектуальная система динамического моделирования бизнес-процессов

Целью реинжиниринга бизнес-процессов (БПР) является целостное и системное проведение реорганизации материальных, финансовых и информационных потоков, направленное на упрощение организационной структуры, перераспределение и минимизацию использования различных ресурсов, сокращение сроков реализации потребностей клиентов, повышение качества их обслуживания [7,9,11,18,20]. Мероприятия по реинжинирингу бизнес-процессов позволяют:

- повысить степень обоснованности проектов по реорганизации деятельности предприятия с учетом анализа и прогнозирования внешних и внутренних факторов развития экономической ситуации;
- анализировать и прогнозировать деятельность предприятия с учетом множества вариантов организации бизнеса и различных схем поведения предприятия на рынке;
- оптимизировать использование материальных, финансовых, человеческих и информационных ресурсов на различных стадиях жизненного цикла проекта реорганизации предприятия;
- разрабатывать обоснованные рекомендации по изменению организационной структуры предприятия и внедрению информационных технологий.

В качестве примера успешного проведения реинжиниринга бизнес-процессов можно привести реорганизацию процессов непроизводственных закупок компании Xerox [7]. В этой компании объем непроизводственных закупок составляет 3 млрд. долл. и включает такие пункты как фурнитура, компьютеры, соглашения об аренде и консалтинговые контракты. В более чем половине случаев непроизводственные закупки были неэффективны, то есть некоторые закупки делались от множества производителей, что приводило к избыточному исследованию цен и составлению списков поставщиков. Реорганизация в 1994 году отдела непроизводственных закупок на основе ручных методов сэкономила 143 миллиона долларов. Однако Xerox определила необходимость проведения дальнейшей реорганизации.

Компания Хегох поставила цель перепроектировать и объединить существующие 17 произвольных и разобщенных процессов в более централизованно-распределенную систему непроизводственных закупок, которая должна обеспечивать 87000 служащих. Основные улучшения новой информационной системы нацелены на сокращение цикла закупок с 2-3 недель до 2 дней и 85 процентное сокращение стоимости на одну транзакцию и по крайней мере 5 процентное общее сокращение стоимости закупок. Основные этапы реорганизации включают: отображение/анализ существующих процессов, перепроектирование и разработку новых более эффективных процессов, их тестирование и реализацию новой системы. Для реорганизации использовались инструментальные средства моделирования и реинжиниринга бизнес-процессов.

Так, компания Хегох уже отображала существующие процессы, применяя простые средства построения диаграмм ABC FlowCharter (Micrografx Inc.). Оценка 17 процессов и 314 процедур показала, что 70 процентов процедур не имеют ценности. Работая совместно с консалтинговой фирмой Meritus Consulting Services на основе ее бумажной технологии компания разработала модель из 42 процедур, объединенных в два типа процессов: на основе контрактов и без контрактов. Для исследования динамической модели использовалось инструментальное средство имитационного моделирования ReThink фирмы Gensym.

В результате моделирования произошло уточнение 42 процедур путем их дальнейшей модификации. Внедрение новой системы дает первоначальное сокращение общей стоимости закупок на 10-15 процентов, а в последующем 5 процентов в год. Новая информационная система разработана фирмой Rational Software Corp. в объектно-ориентированной среде C++.

Анализ проведения работ по реорганизации деятельности компании показывает, что важнейшим этапом реинжиниринга является моделирование как существующих, так и проектируемых бизнес-процессов. Причем, как правило, все материальные, информационные, финансовые потоки имеют изменяющиеся во времени характеристики, например, такие как интенсивность поступления заказов в различные периоды времени по различным видам продукции и услуг. Следствием динамичности бизнес-

процесса является неравномерность использования основных ресурсов: структурных подразделений предприятия, оборудования, информационной системы, источников финансовых средств. В этих условиях задачу анализа эффективности предлагаемых мероприятий по реинжинирингу бизнес-процессов можно решить только в результате проведения имитационных экспериментов. К наиболее известным инструментальным программным средствам имитационного моделирования относятся такие программные продукты, как: ReThink и eSCOR (Gensym), PILGRIM (Мегатрон), РДО (МГТУ), ARIS Simulation (IDS prof. Scheer), ProMolel (Service Model) и др.

К особенностям универсального инструментального средства имитационного моделирования бизнес-процессов ReThink, эффективно реализованным и выгодно отличающим его от других инструментальных средств данного класса, относятся следующие возможности:

- простое графическое конструирование и реконструкция моделей бизнес-процессов пользователями-менеджерами;
- построение моделей на различных уровнях абстракции в зависимости от целей моделирования и точек зрения менеджеров;
- многосценарность имитационных экспериментов для одной модели и параллельность выполнения одного сценария для множества альтернативных моделей;
- моделирование длительности операций на основе различных законов распределения и/или реальных графиков событий или статистических файлов;
- динамический анализ стоимостных затрат выполнения бизнес-процессов, моделирование финансовых потоков;
- варьирование в процессе моделирования типами и размерами ресурсов с различными стоимостными характеристиками за пооперационное и временное их использование;
- применение бизнес-правил при выборе направлений выполнения бизнес-процессов;
- наглядная анимация выполнения бизнес-процессов с возможностью динамического изменения пользователем организации процесса в течение проведения имитационного эксперимента;

- сбор и обработка статистики о выполнении бизнес-процессов, возможность дополнительной обработки результатов имитационных экспериментов с помощью встроенной электронной таблицы G2;
- графическое представление статистических результатов моделирования в виде графиков, стандартных и пользовательских отчетов.
- отображение модели в объектно-ориентированное представление языка UML для последующей разработки программного обеспечения информационной системы.
- преобразование IDEF-модели, являющейся результатом структурного анализа бизнес-процессов, в имитационную модель ReThink.

В дальнейшем будем рассматривать реализацию имитационных моделей бизнес-процессов в среде инструментального программного комплекса Rethink. Так, имитационные модели бизнес-процессов строятся на основе следующих базовых компонентов [7,8,12] (см. приложение 4).

- **Рабочие объекты** - те объекты, над которыми осуществляются процессы. Они проходят через блоки моделирования и обрабатываются ими, аккумулируя статистику производительности и стоимостных затрат в каждой точке моделируемого процесса.
- **Ресурсы** - те объекты, с помощью которых осуществляются процессы. Они предназначены для ограничения количества одновременно исполняемых операций на основе их объема и состава и определяют состав стоимостных затрат результата процесса. Менеджер ресурсов управляет выделением ресурсов по определенным правилам.
- **Блоки моделирования** выполняют такие операции над рабочими объектами, как генерация рабочих объектов (Source) и их уничтожение (Sink), исполнение задач (Task), разветвление процессов (Branch) и объединение путей (Merge), установление (Associate) и разрыв ассоциаций (Reconcile) между объектами, сохранение рабочих объектов в хранилищах (Store) и их извлечение (Retrieve), включение рабочих объектов в списки

(Insert) и их удаление из списков (Remove), перенос пользовательских атрибутов рабочих объектов (Copy Attribute) и копирование объектов (Copy).

- **Сценарии** управляют механизмом моделирования дискретных событий и позволяют проводить одновременное исполнение нескольких моделей.
- **Инструменты** позволяют получать и отображать в числовой и графической форме данные о производительности моделируемых процессов. Кроме того, инструменты позволяют в заданных точках модели вводить необходимые значения параметров.

Имитационное моделирование бизнес-процесса выполнения заказов по системе “Канбан” в среде инструментального средства ReThink

Суть системы “Канбан” заключается в следующем: заказ на требуемый продукт выполняется сразу со склада готовой продукции и одновременно формируется производственный заказ на восполнение истраченного запаса. В том случае, если данного продукта нет на складе, выполнение заказа клиента задерживается, и для него также оформляется производственный заказ, реализация которого приводит, как бы, к повторной попытке выполнить заказ клиента.

Если процесс производства достаточно сложен, то по описанному принципу для производственного заказа формируется заказ (задание) на комплектующие изделия и так дальше назад по технологической цепочке до закупок сырья и материалов. В имитационной модели каждый раз блок “производство” разворачивается в подпроцесс (открывается “матрешка”), пока на последнем уровне не откроется подпроцесс закупок материалов.

Графическая модель бизнес-процесса выполнения заказов в среде ReThink для простого производства, когда готовая продукция по заказу изготавливается из сырья, представлена на рис. 5.14 - 5.15. На рис. 5.14 отражена модель процесса выполнения заказа на верхнем уровне детализации (блоки: поступление, оформление заказа, проверка его выполнимости со склада, производство для восполнения запаса, предварительное производство для отсутствующего на складе продукта, складирование готовой продукции, отгрузка и доставка заказа). На рис. 5.15 показана

модель производства (блоки: оформление задания на заготовку материала, проверка выполнимости задания со склада, закупка материалов для восполнения запаса, предварительная закупка материала, отсутствующего на складе, складирование материалов, заготовка материалов, изготовление продукции).

Рабочие объекты: Заказы, Задания, Продукты, Материалы. Определения классов объектов заданы на специальном подпространстве “Классы объектов”. Сами рабочие объекты на рабочих пространствах модели показываются кружками.

Заказы на готовую продукцию генерируются в блоке-источнике “Поступление заказа” динамически. Для каждого заказа в блоке “Оформление задания” создается задание на заготовку материала. Продукты и материалы хранятся в складах (пулах) “Продукты” и “Материалы” (рис. 5.14). Начальное создание запасов продуктов и материалов производится генераторами на рабочем подпространстве “Инициализация” (рис. 5.14). Извлечение рабочих объектов из пулов задается в блоках “Отгрузка заказа”, “Заготовка материалов”, а помещение рабочих объектов в пулы - соответственно в блоках “Складирование готовых продуктов”, “Складирование материалов”.

Ресурсы: Оборудование. Пул оборудования «Цех» и его подпространство с конкретными станками показаны на рис. 5.14. Можно было бы задать также такие ресурсы, как “Рабочие”, “Работники отдела заказов”, “Работники отдела закупок”, “Поставщики” и др.

Использование ресурсов в процессе задается с помощью менеджера ресурсов, который прикрепляется к соответствующему блоку (в данном случае к блоку “Изготовление продуктов”). Для менеджера ресурсов можно задать следующие параметры (рис. 5.16): количество используемого ресурса, способ выбора ресурса (произвольный, по наибольшему и наименьшему приоритету, по наименьшей стоимости, по наименьшей степени использования), стоимостные характеристики использования (на операцию, на единицу отработанного времени).

Блоки. Имитация бизнес-процесса реализуется в блоках моделирования. В блоке генерации рабочих объектов (Source) с заданной частотой (задается интервал времени между событиями) по

определенному закону распределения, как правило, экспоненциальному, создаются динамические рабочие объекты, которые в соответствии с графом модели бизнес-процесса проходят через другие блоки. В каждом из блоков рабочий объект задерживается на время, заданное в установке временных параметров блока (рис. 5.16): для блока определяется закон распределения, как правило, нормальный, среднее время и стандартное отклонение. Для каждого блока могут быть заданы также maximum activities - число одновременно выполняемых операций (иначе это ограничение определяется через менеджер используемых ресурсов), а также стоимостные характеристики в целом на одну операцию или единицу времени. Рабочие объекты, проходя через блоки, накапливают в своих атрибутах время обработки объекта (рабочее и общее с задержками в очередях), а также стоимость обработки по всем операциям, включая стоимость используемых ресурсов, если последняя задана. В конце процесса рабочие объекты, как правило, удаляются в блоке Sink. В процессе имитации в блоках (рис. 5.17) накапливается статистика о числе поступивших на обработку рабочих объектов (TOTAL STARTS), обработанных рабочих объектов (TOTAL STOPS) и находящихся в обработке на данный момент времени (CURRENT ACTIVITIES), формируется также временная (Duration Subtable) и стоимостная статистика (Cost Subtable).

Среди блоков моделирования блоком общего назначения является задача (Task), который в отличие от других блоков может быть декомпозирован на самостоятельном подпространстве на подблоки, т.е. представлен в виде подпроцесса, как, например, блок “Производство”. Остальные блоки выполняют специфические функции, такие как копирование, разветвление, сохранение, извлечение объектов и др.

Блок “Задача” в отличие от других блоков может иметь сколько угодно входов и сколько угодно выходов. Причем на разных входных и выходных путях могут быть объекты, принадлежащие разным классам (тип объекта задается в таблице пути между двумя блоками). Например, у блока “Оформление задания” на входе указывается заказ клиента, а на выходе - задание на заготовку материала.

Если у блока задачи несколько входов, то поступление объектов синхронизируется, то есть операция в блоке выполняется в том случае, когда на все входные пути поступят объекты. Например, у блока “Изготовление продукта” на входе указаны заказ и материал, на выходе заказ и продукция. В последнем случае изготовление продукта по заказу клиента синхронизировано с поступлением материала. Заметим, что в модели произошло раздвоение процесса: в результате копирования объекта “заказ” одна копия направлена на блок “Изготовление продукта” для синхронизации с моментом поступления материала, а другая копия - на порождение в блоке “Оформление задания” задания на заготовку материала. По системе Канбан на один заказ создаются две карточки: одна карточка заказа клиента уничтожается, когда выполняется другая карточка производственного заказа. В результате выполнения блока “Изготовление продукта” готовая продукция отправляется на склад, а заказ на уничтожение.

Блоки “Разветвление” в модели выполнения заказов использовались для проверки возможности выполнения заказа или задания в момент их поступления. В модели были заданы вероятности прохождения тех или иных путей, например, как 10 к 1, то есть в 10 случаях заказ (задание) выполняется сразу и только в одном случае задерживается для соответственно предварительного производства изделия или закупки материала. Числа пропорции ставятся на путях, исходящих от блока разветвления. Возможно применение и других условий разветвления, таких как проверка типа объекта или значения пользовательского атрибута.

При сложной структуре изделия, состоящего из множества комплектующих деталей, целесообразно использовать контейнерные объекты - изделия, которые в качестве своих атрибутов содержат списки, в данном случае деталей, и соответственно потребуются применение блоков “Включение в список”, “Исключение из списка”.

Сценарий задается пиктограммой “Человеческая голова”, на экран рядом с иконкой выводится текущее время моделирования и состояние модели (рис. 5.14). В таблице сценария определяется максимальное моделируемое время (например, один месяц, полгода, один год и т.д.). В рабочем пространстве стандартного сценария задается режим моделирования: дискретных событий,

пошаговый или синхронизации с реальным временем (например, за одну секунду час реальной работы), а также активации/деактивации и запуска модели. Один сценарий может быть параллельно запущен для нескольких моделей, а для одной модели может быть задано несколько сценариев. Обычно в сценарии имитационного эксперимента предполагается задание необходимых исходных данных для работы имитационной модели.

Инструменты и отчеты. С помощью инструментов осуществляется ввод/вывод данных в процессе моделирования. Для ввода исходных данных (например, интенсивности заказов) используются специальные установщики, с которыми устанавливается связь слайдеров или блоков ввода. Для вывода результатов моделирования используются зонды, к которым присоединяются передатчики текущих значений и графики (рис. 5.18).

В описанном примере на графиках выводятся:

- Длительность цикла выполнения заказа в часах с помощью зонда “Delta-time” (рис. 5.15);
- Загрузка оборудования с помощью зонда “Sample-value” (рис. 5.14)

Длительность выполнения заказа рассчитывается как разность времени завершения обработки заказа и времени его создания. На графике наряду с длительностью выполнения каждого заказа может быть выведена также скользящая средняя за некоторый период времени.

В графике загрузки оборудования количество занятого ресурса выводится для каждого события, вызывающего обработку заказа.

Анализ выходной информации показывает, что длительность выполнения заказа колеблется минимально от 4 часов до максимально 24 часов. Следовательно, заданная пропорция числа заказов, выполнимых сразу со склада, к числу заказов, требующих предварительного изготовления, как 10 к 1, неудовлетворительна. Вместе с тем, загрузка в среднем 1,1 станка из 3 свидетельствует о незагруженности производства и необходимости повышения интенсивности поступления заказов клиентов.

Характеристики производительности предприятия выводятся с помощью специального отчета в таблице свободной формы (рис.

5.18). Причем эти характеристики могут выводиться как по всему процессу, так и по отдельным операциям. В данном примере выводятся:

- Общее (суммарное) количество заказов, выполненных предприятием за заданный период времени;
- Общее (суммарное) количество выполненных заданий на закупку материалов за заданный период времени;
- Текущее количество продуктов, находящихся на складе готовой продукции;
- Текущее количество материалов, находящихся на складе.

В системе могут быть получены и стандартные отчеты по всем блокам, путям, рабочим объектам, ресурсам.

Имитационное моделирование процессов управления цепочками поставок в среде инструментального средства eSCOR

На базе универсального инструмента ReThink создан специализированный пакет имитационного моделирования цепочек поставок eSCOR, который поддерживает стандарт SCOR (Supply-Chain Operations Reference Model), разработанный независимой неприбыльной корпорацией Supply-Chain Council.

Стандарт SCOR представляет межотраслевую методологию, позволяющую описывать цепочки поставок компонентов, в которой участвуют поставщики, производители, дистрибьюторы, потребители [21]. В частности с помощью этой методологии описываются четыре основных бизнес-процесса: планирование, снабжение, производство, поставка продукции и услуг. Стандарт SCOR предоставляет набор метрик, оценивающих эффективность цепочек поставок для совершенствования организации, оптимизации планирования, сокращения сроков и затрат, повышения качества.

Референтные модели, описанные в SCOR, обеспечивают:

- Определение взаимодействий с потребителями от ввода заказа до оплаты счетов.
- Отражение материальных потоков от поставщиков поставщиков до потребителей потребителей, включая оборудование, материалы, комплектующие детали, услуги, программное обеспечение.

- Моделирование маркетинговых транзакции от формирования интегрированных потребностей рынка до заполнения заказов.

Референтные модели поддерживают различные типы организации цепочек поставок в зависимости от точки привязки заказа: заказ на поставку готовой продукции и услуг, заказ на сборку из стандартных комплектующих деталей, заказ с учетом оригинальных материалов, заказ с учетом оригинального проекта (см. главу 6).

Инструментальное средство eSCOR [22] используется как в стратегических целях построения оптимальной конфигурации цепочки поставок, так и в повседневном оперативном управлении поставками в целях своевременного определения узких мест в логистическом процессе.

eSCOR обеспечивает библиотеку специализированных функциональных блоков, связанных с построением цепочек поставок, к которым относятся, например, блоки покупателей, продавцов, продуктов, ресурсов. В схемах логистических процессов отдельно отражаются прямые материальные потоки (от получения заказа до его полного выполнения) и обратные финансовые потоки (оплаты счетов).

Имитационная модель цепочки поставок, как правило, строится на четырех уровнях:

- Межорганизационный уровень – взаимодействие предприятий в цепочке поставок (см. рис.5.19).
- Уровень предприятия – организация логистического процесса внутри предприятия: отражается взаимодействие процессов сбыта, производства, снабжения, управления запасами.
- Уровень процесса – организация отдельного процесса, например, процесс сбыта организуется как набор подпроцессов: прием заказа, оформление договора, отгрузка, прием оплаты и т.д.
- Уровень подпроцесса – набор операций конкретной процедуры, например, оформления заказа. Обычно на четвертом уровне детализации используются универсальные блоки инструментального средства ReThink.

Для анализа эффективности организации цепочек поставок инструментальное средство eSCOR предоставляет набор стандартных метрик, которые группируются следующим образом:

- Производительность поставок.
- Производительность выполнения заказов.
- Полный жизненный цикл заказа.
- Время реакции на требование в цепочке поставки.
- Гибкость производства.
- Затраты на управление цепочками поставок.
- Затраты на гарантийное обслуживание и возврат брака.
- Полный платежный цикл.
- Состояние и оборачиваемость запасов в днях.
- Рентабельность.

Примеры составления отчета о финансовых потоках в табличной и графической форме представлены соответственно на рис. 5.20- 5.21.

Таким образом, с помощью имитационных моделей, подключенным к реально функционирующим информационным системам, можно оперативно конфигурировать цепочки поставок с учетом формируемого прогноза показателей использования всех основных ресурсов. Особенно интересно в этом плане подключение инструментальных средств eSCOR и ReThink к работе интеллектуальных агентов, создаваемых с помощью инструментального средства G2 ADE.

Что следует запомнить

Бизнес-процесс - это взаимосвязанная совокупность материальных, информационных, финансовых потоков (рабочих потоков - workflow), проходящих через взаимодействующие подразделения предприятия и направленных на выполнение заказа клиента (изготовление товара или осуществление услуги).

Модель бизнес-процесса должна отражать как последовательность операций (направление рабочих потоков), так и события, в зависимости от которых выполняются конкретные цепочки операций (бизнес-правила).

Рабочий объект - это сущность, над которой осуществляется некоторая операция (действие, функция, преобразование). Рабочие объекты могут быть материальными, финансовыми, информационными.

Ресурсы - это сущности (субъекты), с помощью которых осуществляются бизнес-процессы.

Операция (действие, функция) преобразует входные рабочие объекты в выходные или модифицирует их.

Событие фиксирует факт завершения выполнения некоторой операции и образования нового состояния объекта или нового объекта.

Динамические ЭС оперативного управления бизнес-процессами предназначены для планирования, диагностики, мониторинга и коррекции бизнес-процессов в реальном времени.

Система управления цепочками поставок связывает процессы снабжения, производства, дистрибуции, потребления на уровне взаимодействующих предприятий: поставщиков, производителей, посредников, потребителей.

Система управления запасами управляет состоянием запасов готовой продукции, комплектующих деталей, материалов, в зависимости от уровня потребления которых инициируются процессы производства и снабжения.

Классы систем управления запасами: с фиксированным размером заказа на закупку; с фиксированным интервалом времени между заказами; комбинированные системы. Для рынка, ориентированного на реализацию индивидуальных заказов клиентов, наиболее предпочтительны системы с фиксированным размером заказа или комбинированные системы.

Динамические ЭС оперативного управления запасами позволяют оперативно реагировать на колебания в сроках поставки сырья и материалов и заказов на готовую продукцию, учитывая выбранную стратегию обслуживания клиентов.

Основные классы объектов динамических ЭС оперативного управления запасами: “запасы”, “заказы клиентов”, “заказы на закупку”.

Основные правила реагируют на события, связанные с появлением и изменением статуса заказов клиентов, изменением уровня запасов на складе, а также состоянием поставщиков.

Рейнжиниринг бизнес-процессов - это фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов (БП) для достижения коренных улучшений в основных

показателях деятельности предприятия: сроках, качестве, затратах, сервисе.

Интеллектуальная система моделирования бизнес-процессов предназначена для анализа на динамической основе эффективности организации бизнес-процессов, прогнозирования последствий реализации рекомендаций по реинжинирингу бизнес-процессов.

Имитационная модель бизнес-процесса по заданным законам распределения генерирует рабочие объекты, которые обрабатываются в компьютере в ускоренном масштабе времени в блоках операций, формируя временную и стоимостную статистику осуществления процессов за определенные периоды времени.

Сценарий имитационного эксперимента определяет исходные условия выполнения имитационной модели (интенсивность событий, временные и стоимостные характеристики обработки рабочих объектов). Один и тот же сценарий может быть применен к нескольким имитационным моделям. Для одной имитационной модели может быть задано несколько сценариев (вариантов) моделирования.

Результаты имитационных экспериментов выводятся в виде графиков и отчетов и отражают сгенерированную временную и стоимостную статистику за моделируемый период времени.

Имитационная модель системы управления заказами “Канбан” разработана как совокупность вложенных друг в друга моделей, каждая из которых отражает одинаковую последовательность операций по реализации заказа со склада и формированию заказа на предшествующий по технологической цепи этап бизнес-процесса по восполнению запаса на текущем этапе бизнес-процесса.

Стандарт SCOR (Supply-Chain Operations Reference Model) - межотраслевая методология, позволяющая описывать референтные (типовые) модели цепочек поставок компонентов, в которых участвуют поставщики, производители, дистрибьюторы, потребители.

Литература

1. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2000.
2. Данилов А.В., Григорьев С.В., Тельнов Ю.Ф. Имитационное моделирование процессов управления запасами. / Шестая национальная конференция с международным участием “Искусственный интеллект - 98”. Сб. научных трудов. - Пущино, РАИИ, 1998, т.3.
3. Емельянов А.А., Власова Е.А., Дума Р.В.. Имитационное моделирование экономических процессов. / Под ред. А.А. Емельянова. – М.: Финансы и статистика, 2002.
4. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО. – М.: АНВИК, 1998.
5. Логистика: Учебное пособие/Под ред. Б.А. Аникина.- М.: ИНФРА-М, 1997.
6. Неруш Ю.М. Коммерческая логистика: Учебник для вузов. - М.: Банки и биржи, ЮНИТИ, 1997.
7. Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса: Реинжиниринг организаций и современные информационные технологии. - М.: Финансы и статистика, 1997.
8. Попов Э.В., Шапот М.Д., Кисель Е.Б., Фоминых И.Б. Статические и динамические экспертные системы. - М.: Финансы и статистика, 1995.
9. Робсон М., Уллах Ф. Практическое руководство по реинжинирингу бизнес-процессов /Пер с англ. - М.: Аудит. ЮНИТИ, 1997.
10. Сергеев В.И. Менеджмент в бизнес-логистике. – М.: Информационно-издательский дом «ФИЛИНЪ», 1997.
11. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем / Учебник под ред. Тельнова Ю.Ф. – М.: Финансы и статистика, 2001.
12. Тельнов Ю.Ф. Реинжиниринг бизнес-процессов. - М.: МЭСИ, 1999.
13. Тельнов Ю.Ф. Интеллектуальная система управления логистическими процессами. М.: Теория и системы управления, 1999, №5.

14. Тельнов Ю.Ф. Реинжиниринг логистических бизнес-процессов предприятия на основе имитационной модели в контуре корпоративной информационной системы. М.: ЛОГИНФО. Логистический менеджмент и технологии, 2001, №4
15. Тюрин Ю.Н., Макаров А.А. Анализ данных на компьютере/ Под ред. В.Э.Фигурнова. - М.: Инфра-М- М.: Финансы и статистика,, 1995.
16. Шлеер С., Меллор С. Объектно-ориентированный анализ: Моделирование мира в состояниях. Пер. с англ. - Киев: Диалектика, 1993.
17. Эддоус М., Стэнфилд Р. Методы принятия решений / Пер. с англ. М.: Аудит, ЮНИТИ, 1997.
18. Hammer M., Champy J. Reengineering the Corporation. A Manifesto for Business Revolution. HarperBusiness, 1993.
19. Ross R.G. The Business Rule Book. Classifying, Defining and Modelling Rules. Data Base Research Group, Inc. -1997, 394 p.
20. Scheer A-W. Business Process Engineering: Reference Models for Industrial Enterprises. - 1995.
21. www.supply-chain.org.
22. www.gensym.com/b2b_modelling/papers/escor.htm

Глава 6. Технология создания адаптивной информационной системы на базе методологии «BAAN - Организационная среда»

6.1. Жизненный цикл адаптивной информационной системы на основе компонентной технологии

Компонентная технология создания информационных систем предполагает настройку типовых компонентов (программных модулей) в соответствии с особенностями конкретного предприятия. В учебном пособии компонентная технология рассматривается на примере технологии внедрения системы комплексной автоматизации предприятия BAAN IV [4,10], обладающей множеством альтернативных функциональных возможностей для различных типов бизнес-процессов, например, изготовление продукции для магазинной продажи, на заказ, в соответствии с индивидуальными проектами и т.д.

Адаптивность компонентов системы BAAN IV при создании реальных корпоративных информационных систем главным образом обеспечивается благодаря использованию методологии «BAAN Организационная среда» (BAAN Orgware). В основе этой методологии лежит **концепция динамического моделирования предприятия (DEM - Dynamic Enterprise Modeling)**, которая основана на принципе динамического отображения в ИС на всех этапах жизненного цикла модели предприятия (модели проблемной области), причем жизненный цикл предполагает непрерывное развитие ИС в соответствии с изменением модели. По концепции DEM информационная система является результатом конфигурации компонентов (программных модулей) BAAN IV на базе модели предприятия, автоматизировано поддерживаемой в репозитории системы.

Жизненный цикл адаптивной информационной системы в соответствии с методологией BAAN Orgware состоит из стадий, определяемых **методом целенаправленного управления проектом (Target)**, который позволяет быстро и эффективно привести в соответствие бизнес-процессы предприятия и систему BAAN IV. Независимо от того, создается ли новая ИС или осуществляется модернизация существующей ИС, данный метод использует три

стадии внедрения, каждая из которых подразделяется на три определенных этапа (рис.6.1):

Стадия 1. Обоснование выбора системы комплексной автоматизации предприятия BAAN IV.

- Обследование предприятия на предмет необходимости внедрения системы BAAN IV.
- Принятие решения о внедрении системы BAAN IV.
- Заключение контракта на проведение работ по внедрению системы BAAN IV.

Стадия 2. Реализация проекта по внедрению системы BAAN IV.

- Создание и утверждение модели предприятия (моделирование).
- Настройка и конфигурация ИС (пилотное-проектирование).
- Сдача ИС в эксплуатацию (внедрение).

Стадия 3. Оптимизация ИС в процессе эксплуатации.

- Расширение функциональных возможностей в процессе эксплуатации.
- Контроль за выполнением бизнес-процессов со стороны руководства предприятия.
- Анализ реализации стратегических целей предприятия.

Стадия 1: Выбор системы BAAN IV

Внедрение системы BAAN IV начинается с анализа деятельности и потребностей предприятия. Перед тем, как будет принято решение о приобретении системы BAAN IV, консультанты по программному продукту должны определенное время затратить на предварительное изучение характера деятельности и бизнес-целей предприятия. Результатом анализа является разработка плана (коммерческого предложения), содержащего рекомендации по программным и аппаратным средствам, а также необходимым работам. Во время первой стадии выясняется, каким образом приложения BAAN IV будут функционировать в условиях предприятия. Консультанты подготавливают рекомендации по программному обеспечению, демонстрируют руководству предприятия систему, а также модель бизнес-функций, разработанную на основе полученной информации о предприятии.

План включает предложение по изменению организационной структуры, настройке, внешним интерфейсам и архитектуре. Затем, когда предварительная структура информационной системы определена, разрабатывается проект соглашения, служащего основой для сотрудничества. В соглашение входят план и бюджет проекта, удовлетворяющие обе стороны.

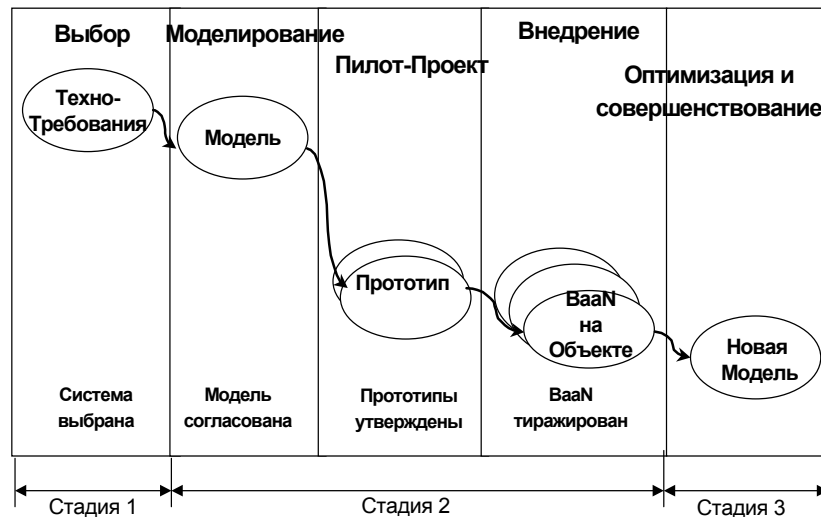


Рис.6.1. Метод целенаправленного внедрения BAAN IV

Стадия 2: Внедрение системы BAAN IV

Во время второй стадии консультанты по продукту BAAN IV в тесном сотрудничестве с представителями предприятия - ключевыми пользователями, работают над определением модели предприятия на уровне бизнес-процессов и адаптацией приложений BAAN IV согласно сформулированным на предыдущей стадии требованиям. В своей работе консультанты используют референтные типовые модели бизнес-процессов для определенных типов производства и отраслей, которые модифицируются в соответствии со сформулированными требованиями в проектную модель предприятия. На пилотной стадии происходит установка системы, проводится обучение пользователей и определяются

детальные процедуры по продукту BAAN IV. В результате окончания второй стадии информационная система должна быть полностью готова к эксплуатации.

Стадия 3: Обеспечение оптимальной поддержки текущей деятельности

На этой стадии осуществляется непрерывное совершенствование проектной модели предприятия и адаптация ИС к необходимым изменениям. Для анализа точности, надежности и эффективности работы ИС предусмотрено проведение регулярных проверок эффективности выполнения системой операций. Функция проверки осуществляет контроль выполнения бизнес-процессов и их взаимодействия, оценивает реализацию стратегических целей предприятия на основе анализа показателей деятельности предприятия.

В процессе внедрения системы BAAN IV на предприятии множество специалистов используют методы и средства методологии BAAN-Orgware:

- консультанты по предпродажной деятельности, которым оказывается поддержка в презентациях продукта;
- специалисты по внедрению (инженеры по внедрению, менеджеры проекта, эксперты по использованию), которые используют методы и средства BAAN-Orgware в ходе анализа, обучения и внедрения;
- советники и бизнес-консультанты, которые выполняют проекты совершенствования бизнеса;
- разработчики программного обеспечения, которые пользуются средствами документирования бизнес-процессов;
- конечные пользователи, которым оказывается поддержка в ходе эксплуатации.

Достоинства процесса внедрения BAAN IV на основе метода целенаправленного управления проектом заключаются в следующем (рис. 6.2):

- короткая стадия выбора, в течение которой проводится презентация продукта в конкретной отрасли с целью обеспечения потенциального заказчика информацией о применимости BAAN IV на предприятии;

- короткая стадия внедрения, в ходе которой оперативно осуществляется конфигурация BAAN IV в соответствии с моделью предприятия;
- непрерывная модернизация ИС на основе постоянного обновления модели предприятия и гибкой технологии адаптации программных модулей.

Процесс внедрения

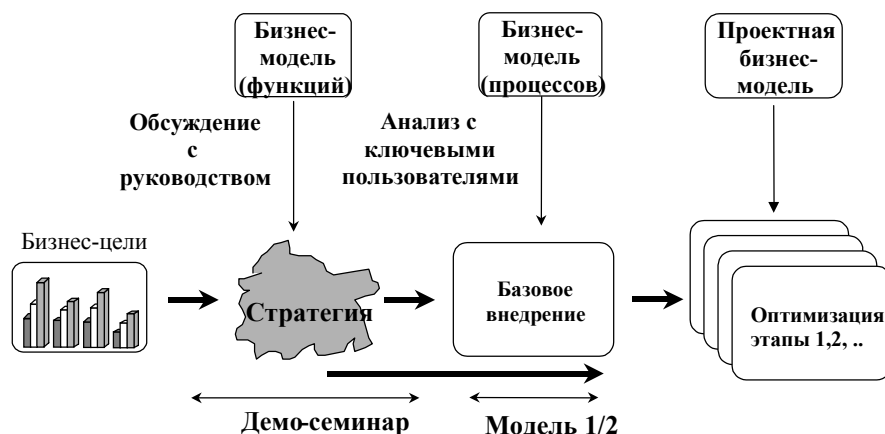


Рис. 6.2. Процесс внедрения BAAN IV

Таким образом, использование методологии внедрения «BAAN - Организационная среда» позволяет обеспечивать высокую адаптивность ИС к изменениям. Благодаря гибкости системы и ее мощности можно легко переходить от старой системы к новой системе BAAN. Более того, это дает возможность вначале быстро провести автоматизацию существующих бизнес-процессов «как есть», а затем выполнить бизнес-реинжиниринг с целью построения системы «как надо», используя средства BAAN по анализу эффективности функционирования ИС. Вследствие такого подхода обеспечивается непрерывное улучшение организации бизнес-процессов, быстрая отдача от внедрения информационных технологий и снижение затрат на создание и модернизацию ИС (рис.6.3) [4,10].

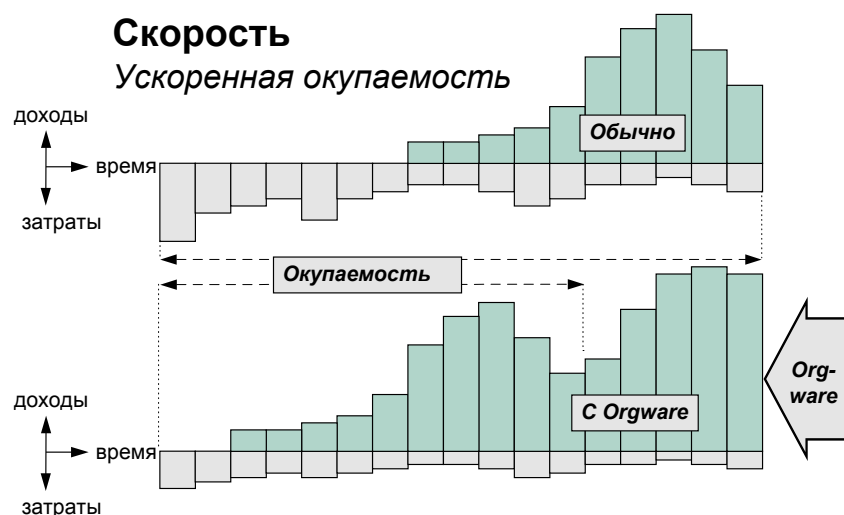


Рис.6.3. Эффективность от внедрения системы BAAN

Методология внедрения системы комплексной автоматизации предприятия BAAN-Orgware полностью поддерживается мощным набором инструментальных средств, интегрированных в семействе продуктов компании BAAN IV, к которым относятся:

- Инструмент моделирования предприятия (Enterprise Modeler), позволяющий отображать модели предприятий на различных уровнях детализации и на различных этапах жизненного цикла информационной системы, а также выполнять конфигурацию информационной системы.
- Администратор деятельности предприятия (Enterprise Performance Manager), с помощью которого отображается система показателей деятельности предприятия, позволяющая оценивать эффективность организации бизнес-процессов после внедрения информационной системы.
- Средства внедрения (Enterprise Implementer), нацеленные на составление бизнес-плана по внедрению BAAN IV, в частности планирование финансовых ресурсов, выбор программно-технических средств, составление календарного плана графика выполнения работ.

6.2. Моделирование предприятия и конфигурация информационной системы

Модель предприятия состоит из модели бизнес-функций; модели реализующих их бизнес-процессов; бизнес-правил, проверяющих целостность модели предприятия и выполняющих конфигурацию информационной системы; модели бизнес-организации (структуры), определяющей роли участников бизнес-процессов по взаимодействию с информационной системой. Использование перечисленных компонентов модели предприятия в процессе моделирования и конфигурации информационной системы показано на рис. 6.4.

Моделирование предприятия осуществляется с помощью инструментального средства Enterprise Modeler на трех уровнях: уровне основных данных, который не зависит от конкретной конфигурации информационной системы; референтном уровне, соответствующем некоторой типовой конфигурации информационной системы, и проектном уровне, определяющем конфигурацию информационной системы конкретного предприятия (рис. 6.5). На уровне основных данных производится наполнение репозитория определениями бизнес-функций, бизнес-процессов и бизнес-правил, для чего используется графический редактор и редактор экранных форм. Референтная модель представляет собой скопированное и отконтролированное подмножество базовой модели, в которую добавляются определения ролей участников бизнес-процессов. Проектная модель формируется как подмножество референтной и/или базовой моделей с определением фаз внедрения (оптимизации) информационной системы, то есть последовательности реализации различных вариантов бизнес-функций, и указанием конкретных исполнителей работ бизнес-процессов. На основе проектной модели осуществляется генерация автоматизированных рабочих мест конкретных исполнителей бизнес-процессов.

Процесс динамического моделирования предприятий

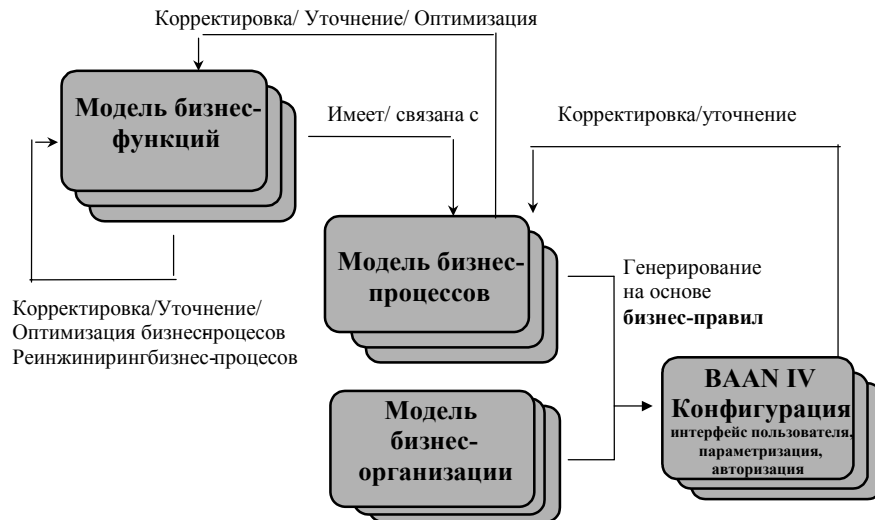


Рис. 6.4. Процесс моделирования и конфигурации ИС

Построение референтной модели предприятия в принципе не обязательно. Наличие референтной модели при покупке программного продукта предполагает заполненность репозитория базовыми данными. После внедрения информационной системы на конкретном предприятия проектная модель может быть переведена в разряд референтной модели, которая будет служить основой для новых внедрений или модернизаций информационной системы. В случае необходимости внесения изменений в референтную или проектную модели требуется в начале выполнить их в модели основных данных, а затем скопировать внесенные изменения в референтную или проектную модели.

Процедура создания модели предприятия

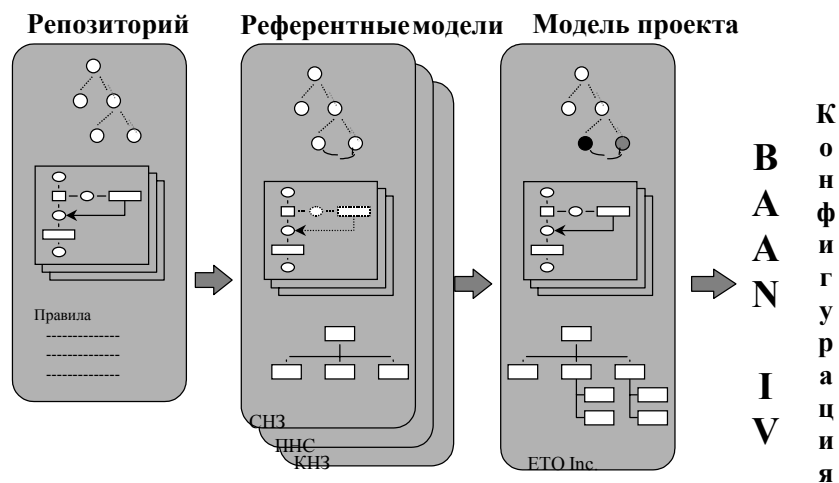


Рис. 6.5. Технология конфигурации информационной системы с использованием Enterprise Modeler

I. Моделирование основных данных

Моделирование бизнес-функций

Бизнес-функции используются для реализации бизнес-целей (повышения рентабельности, увеличения оборачиваемости капитала, сокращения запасов и т.д.) и описываются общепринятыми в бизнесе терминами (не терминами программных модулей BAAN IV).

Модель бизнес-функций в концепции BAAN IV представляет собой иерархическую декомпозицию функциональной деятельности предприятия (рис. 6.6). В корне дерева указывается тип производства предприятия (заказное, мелкосерийное, серийное, массовое, с непрерывным циклом производства). Далее указываются классы бизнес-процессов или мега-функции (развития предприятия, выполнения процессов оперативной деятельности, процессов вспомогательных служб). Главные функции отображают функциональные подсистемы предприятия, к которым относятся

следующие функции: маркетинг, бизнес-планирование, закупки, производство, сбыт продукции, послепродажное обслуживание, ремонт оборудования и т.д. Главные функции разбиваются на основные функции, которым соответствуют конкретные бизнес-процессы, например, оформление заказа, поставка, платеж и т.д. Основные функции могут иметь варианты исполнения, подразумевающие поэтапное внедрение или в терминологии BAAN оптимизацию. Основным функциям в дальнейшем назначаются бизнес-процессы.

Модель бизнес-функций предназначена для решения следующих задач:

- Используется как первый логический шаг в процессе реализации бизнес-целей, на базе которой в дальнейшем создается модель бизнес-процессов.
- Применяется в качестве вспомогательного средства советников и консультантов при проведении презентаций для руководства предприятия, на которых демонстрируется предлагаемая модель предприятия.
- Отражает отраслевое «ноу-хау» в области совершенствования хозяйственной деятельности, наилучшей практики ведения бизнеса и показателей деятельности, направления оптимизации бизнес-функций.
- Используется в качестве инструмента перевода бизнес-модели, определенной в общепринятых в бизнесе терминах, в специфическое для BAAN IV решение.
- Используется в мониторинге выполнения бизнес-функций на основе анализа показателей деятельности предприятия.

Модель бизнес-функций

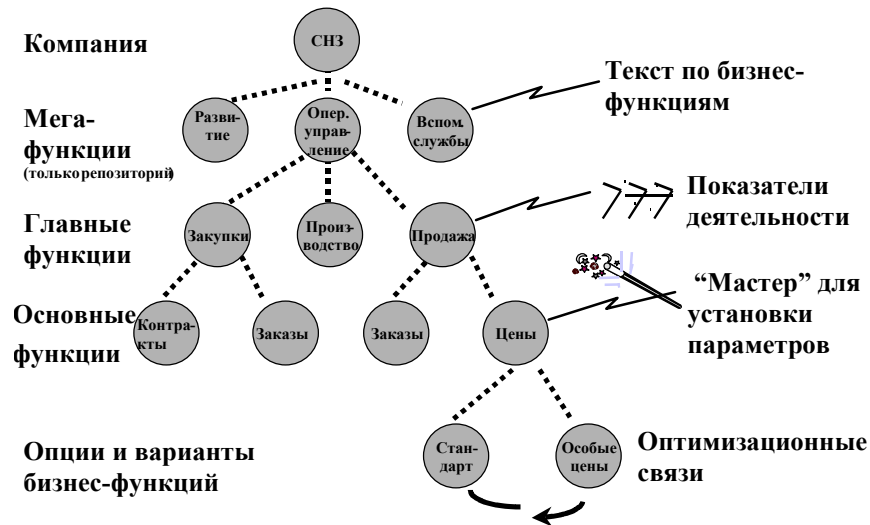


Рис. 6.6. Компоненты модели бизнес-функций

Модель бизнес-функций имеет следующие особенности:

- Бизнес-функции показаны как узлы в древовидной диаграмме, последовательно раскрываемой для пользователя.
- Бизнес-функции могут поясняться текстами, в которых описываются характеристики отдельной бизнес-функции, представляющие инструкцию по выполнению или комментарий, описывающий сущность функции.
- Бизнес-функции могут быть связаны с показателями деятельности предприятия модуля BAAN IV Enterprise Performance Manager (Рис. 6.7), оценивающие эффективность их выполнения после внедрения информационной системы. Набор показателей графически оформляется в виде дерева показателей («скелета рыбы»), каждый из которых может быть детализирован в виде самостоятельного поддерева и иметь представления в виде таблиц и графиков. Причем показатели на дереве окрашиваются в три цвета: зеленый - нормально, красный

- ненормально, желтый - предостережение о возможных негативных тенденциях.
- Между вариантами бизнес-функции могут существовать отношения оптимизации, отражаемые на диаграмме бизнес-функций стрелками, которые показывают последовательность внедрения вариантов бизнес-функций (см. рис. П5.1 в приложении 5).
- К модели бизнес-функции может быть подключена мастер-программа для задания параметров конфигурации программных модулей.

Моделирование бизнес-процессов

Модель бизнес-процесса - это некоторая последовательность работ, которая включает сеансы BAAN (шаги программных модулей - описание процесса в терминах BAAN IV), а также неавтоматизированные процедуры и приложения, разработанные в другой программной среде. Различают два типа бизнес-процессов: основной, соответствующей реализации некоторой основной бизнес-функции (из начала в конец), и детальный процесс, соответствующий одному виду работ основной функции и описывающий, как правило, последовательность операций на одном рабочем месте в цепочке процесса (рис. 6.8). Детальный процесс является вложенным в основной (число уровней вложенности неограниченно). Например, в приложении 6 описывается бизнес-процесс закупки материалов, в который встраивается детальный бизнес-процесс БУХ (Учет счетов-фактур), выполняемый бухгалтерией (приложение 5 рис. П5.8).

В качестве автоматизируемых работ в бизнес-процессах могут использоваться *утилиты*, представляющие собой наборы вспомогательных сеансов (типа “показ” и “печать”), которые могут не обязательно выполняться пользователями, например, обращение к правовой информационно-справочной системе в процессе выполнения некоторой ручной работы.



Рис. 6.7. Дерево показателей деятельности предприятия

Модель бизнес-процесса позволяет наглядно представить и задокументировать:

- Последовательность выполнения работ с возможными разветвлениями для демонстрации функциональных возможностей информационной системы ключевым и конечным пользователям.
- Привязку сеансов BAAN IV к конкретным работам для последующего конфигурирования информационной системы.
- Процедуры начального запуска и установочные процедуры, необходимые для эксплуатации BAAN IV.
- Определить администрирование выполнения работ посредством назначения исполнителей работ, соответствующих ролей, прав доступа, рабочих инструкций (на этапах построения референтной и проектной моделей).

Модель бизнес-процессов

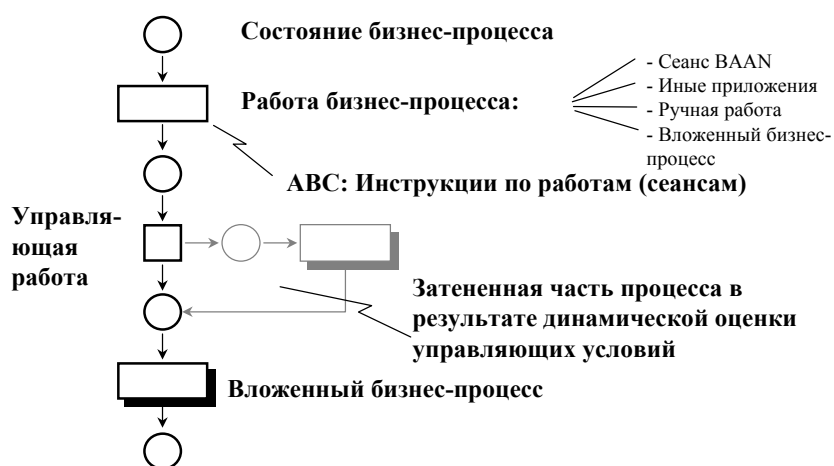


Рис.6.8. Компоненты модели бизнес-процесса

Бизнес-процессы обладают следующими характеристиками:

- Бизнес-процессы состоят из ряда компонентов, а именно состояний (входных и выходных каналов), работ и управления (рис. 6.9). Состояние предшествует каждой работе и следует за ней. Работа может представлять собой:

- сеанс работы с BAAN IV,
 - сеанс работы с другим прикладным программным обеспечением,
 - неавтоматизированную ручную деятельность,
 - вложенный бизнес-процесс,
 - управляющую работу (статическое или динамическое условие).
- Бизнес-процессы моделируются на основе сети Петри [9]. Работа описывается как переход объектов из входного состояния в выходное. Входное состояние определено, когда во входной канал попадает знак (объект), в этот момент начинается работа. Обработанный знак попадает в выходной канал, определяя выходное состояние работы. Содержание вложенного бизнес-процесса определяется на более низком уровне. Управление копируют знаки в каналы выхода по следующим правилам (рис. 6.9):
 - во все каналы (AND);
 - по меньшей мере в один канал (OR);
 - только в один установленный канал (XOR).

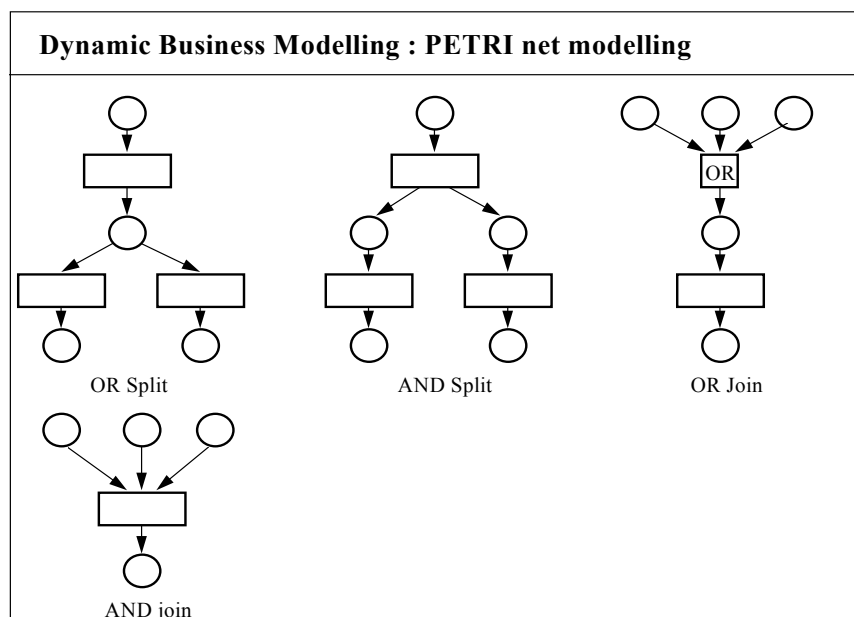


Рис. 6.9. Управление переходами между работами

- С работами могут быть соотнесены должностные инструкции, документы и коды утилит. Должностные инструкции могут содержать особую вспомогательную информацию для сеанса в пределах конкретной работы (бизнес-процесса). В рамках определенной работы возможно сопоставление с документом, созданным в рамках данного вида деятельности. Коды утилит - это блоки сеансов, которые могут быть использованы в качестве вспомогательных сеансов, например, сеанс вывода на экран справочника для ознакомления, распечатка выходных документов.
- Ссылка на другой бизнес-процесс может быть приложена к такому компоненту как состояние, в результате чего может быть дано определение бизнес-процесса на нескольких страницах.
- Любой узел управления может иметь несколько исходящих стрелок. Каждой стрелке соответствует определенное значение условия. Значение условия соответствует выбранному варианту бизнес-функции (статическое условие) или установленному для

нее параметру (динамическое условие). Таким образом, в зависимости от варианта бизнес-функции или установленных параметров схема бизнес-процесса компонуется динамически.

- На стадиях построения референтной и проектной модели работам могут быть назначены сотрудники (на уровне типов и конкретных работников) и выполняемые ими функции.
- В результате конфигурации информационной системы в качестве интерфейса пользователя используется получается структура меню и/или графическая диаграмма бизнес-процесса.

Представление бизнес-правил

Бизнес-правила - это декларативные условные выражения, связанные с организацией бизнеса и использования программного продукта BAAN IV, которые хранятся в репозитории и применяются для контроля корректности построенной бизнес-модели и управления процессом конфигурации информационной системы. Бизнес-правила вводятся в репозиторий на стадии моделирования основных данных, а используются на стадиях построения референтных и проектных моделей при запуске соответствующих редакторов.

В инструментальном программном средстве DEM различают четыре вида бизнес-правил (рис. 6.10):

- правила целостности проверяют согласованность совместного использования компонентов бизнес-моделей,
- правила выбора осуществляют преобразования моделей бизнес-функций в модели бизнес-процессов,
- правила установки параметров задают значения параметров настройки сеансов BAAN (программных модулей),
- правила установки статических условий конфигурируют организацию бизнес-процессов.

Рассмотрим определение бизнес-правил более детально. Примеры редактирования бизнес-правил представлены в приложении 5 на рис. П5.3 – П5.5.

Бизнес-правила

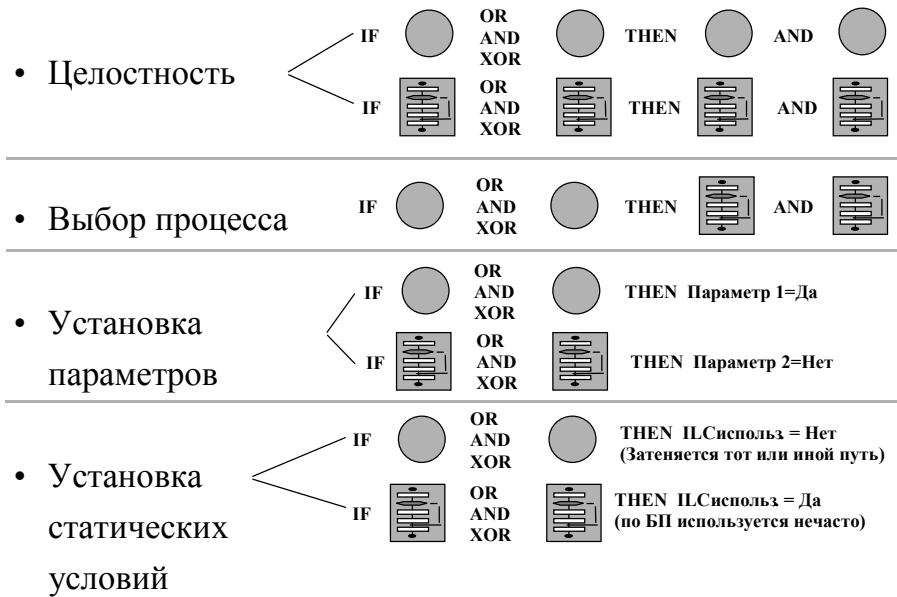


Рис.6.10. Виды бизнес-правил

Правила целостности используются для проверки согласованности референтной или проектной модели с точки зрения полноты и непротиворечивости используемых бизнес-функций и бизнес-процессов. В частности при копировании компонентов из модели основных данных в референтную или проектную модель осуществляется проверка полноты переноса необходимых бизнес-функций и бизнес-процессов, например:

Если присутствует вариант бизнес-функции «Прямая поставка», то бизнес-функции «Обработка заказа на закупку» и «Обработка заказа на продажу» должны быть представлены в модели.

Правила преобразования автоматически переносят в референтную или проектную модель из модели основных данных определения бизнес-процессов, которые соответствуют основным бизнес-функциям. Например:

Если определена основная бизнес-функция «Обработка контракта», то необходимо выбрать бизнес-процесс «Оформление контракта».

Правила установки параметров представляют собой конфигурационные правила, которые определяют глобальные значения параметров для всех модулей и сеансов БААН. Параметры включают как значения параметров работы программных модулей, так и подключение программных модулей (сеансов БААН). Например:

*Если был определен вариант бизнес-функции «Обработка заказа на покупку в режиме ЭОД (Электронного обмена данными)»,
То включается значение параметра «Электронный обмен данными»*

Правила статических условий – конфигурационные правила, с помощью которых устанавливается связь вариантов бизнес-функций со статическими условиями (управляющими работами на графическом представлении бизнес-процесса).

*Если выбран вариант бизнес-функции «Ручной учет»
То включить статическое условие «Учет»*

Статические условия (см. приложение 5 рис. П5.6) в свою очередь определяют условие включения или исключения ветви работ в бизнес-процессе проектной модели. Например, при выборе на первой фазе внедрения информационной системы варианта бизнес функции «ручной учет» (приложение 5 рис. П5.7) срабатывает правило выбора статического условия (приложение 5 рис. П5.5), которое включает статическое условие «учет» в положение «не требуется учет (автоматизированный) счетов-фактур», а следовательно ветвь бизнес-процесса «учет счетов-фактур» деактивируется (осветляется на графическом представлении модели – см. приложение 5 рис. П5.8). Таким образом, статические условия являются параметрами (принимающими значения “да” или “нет”), которые определяют ветку графической диаграммы, по которой осуществляется конфигурация бизнес-процесса.

Наряду со статическими условиями инструмент моделирования предприятий DEM позволяет отображать в качестве управляющих работ и динамические условия, которые проверяются в ходе выполнения бизнес-процесса на стадии эксплуатации сконфигурированной информационной системы.

II Референтное моделирование

Референтная модель получается в результате копирования компонентов из модели основных данных. При этом автоматически запускаются на выполнение все относящиеся к ней бизнес-правила.

Моделирование бизнес-организации

На стадии референтного и проектного моделирования осуществляется построение модели бизнес-организации (рис. 6.11), которая представляет собой описание организационной структуры и структуры персонала типового предприятия отрасли (референтная модель) или какого-либо конкретного предприятия (проектная модель).

Структура персонала может быть описана на абстрактном уровне с помощью описания ролей (референтная модель) или на уровне конкретных сотрудников (проектная модель). Через указатели роли сотрудников устанавливается связь организационной структуры с бизнес-процессами. Указатель роли определяет тип работника, который может выполнять ту или иную работу (экономист, бухгалтер, менеджер и т.д.). Для каждой роли определяются полномочия в выполнении функций, права доступа к информации, должностные инструкции (рис. 6.12). При назначении работы конкретному работнику всегда осуществляется проверка роли, которую он может выполнять. Если при этом тип конкретного работника не соответствует роли, то последний не получает доступ на стадии эксплуатации информационной системы к выполнению работы.

Модель организационной структуры

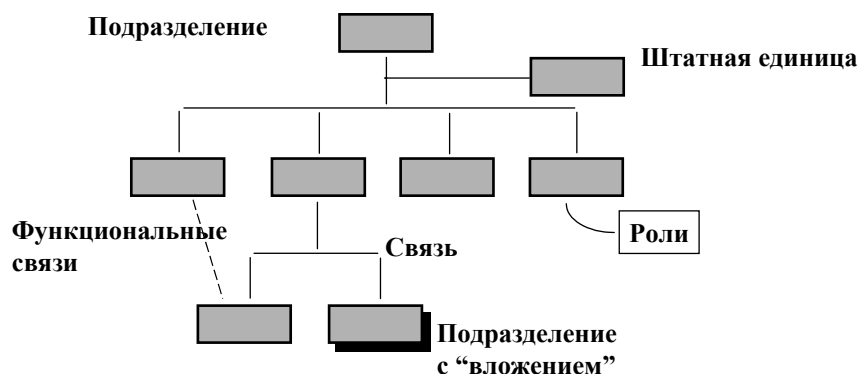


Рис. 6.11. Компоненты модели бизнес-организации

Построение модели бизнес-организации преследует следующие цели:

- получить представление о настоящей и, по возможности, будущей (целевой) организационных структурах и задокументировать их.
- сделать возможным создание пользовательского интерфейса BAAN IV (генерацию автоматизированного рабочего места) по выполняемым ролям сотрудникам (на основе референтной модели) или конкретным сотрудникам (по проектной модели),

Модель бизнес-организации обладает следующими характеристиками:

- Организационная модель состоит из ряда организационных компонентов (подразделений, отделов). Данным организационным компонентам можно присвоить атрибуты: название, тип и произвольный текст.
- Между организационными компонентами, иерархически неподчиненными друг другу, могут быть установлены функциональные отношения. С данными функциональными отношениями можно соотнести текстовое описание.

- Организационная модель может состоять из одной или более организационных схем. Данные схемы могут соподчиняться друг другу с помощью определения компонентов субдиаграмм (например, у отдела может быть собственная организационная структура).
- Для бизнес-функций и работ бизнес-процессов на референтном и проектном уровнях могут быть назначены указатели ролей и конкретные работники. При этом на соответствующих графических диаграммах (см. приложение 5 рис. П5.7-П5.8) отображаются специальные пиктограммы с изображением людей.
- Организационные модели на проектном уровне могут определяться по фазам оптимизации для наглядного отражения последовательности этапов внедрения информационной системы.

Назначение ролей

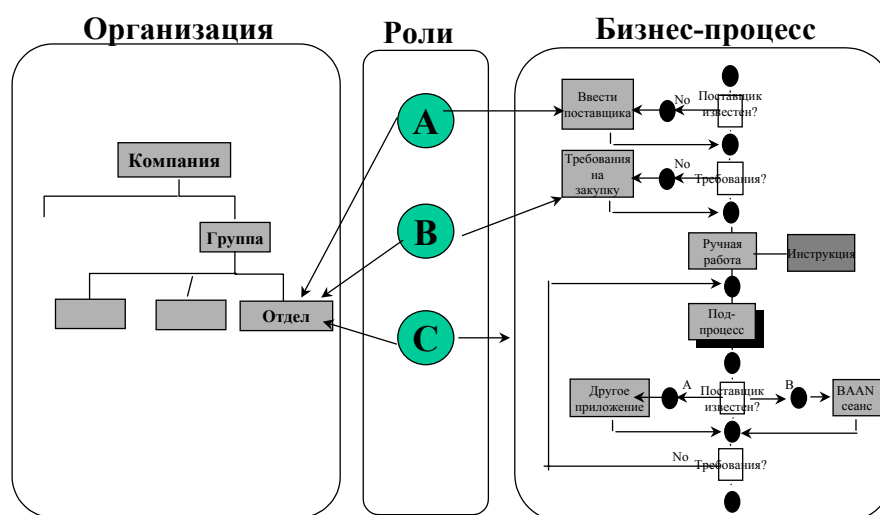


Рис. 6.12. Установление связи модели бизнес-организации с референтной моделью бизнес-процесса

III. Проектное моделирование

Моделирование проекта - процесс, посредством которого референтная модель анализируется и возможно модифицируется для конкретного предприятия. Проектная модель формируется путем копирования основных компонентов из референтной модели и/или модели основных данных.

Определение фаз внедрения

На этапе разработки модели-проекта для каждой основной бизнес-функции устанавливаются фазы внедрения или оптимизации бизнеса. Фазы внедрения (оптимизации) по проектным моделям являются этапами в цикле преобразований и автоматизации бизнес-процесса, например, первая фаза внедрения – «ручной учет счетов-фактур», вторая фаза внедрения – «автоматизированный учет счетов фактур (приложение 5 рис. П5.7). Фазы внедрения задаются на самом нижнем уровне отображения бизнес-функций, то есть для вариантов бизнес-функций.

Каждой фазе внедрения в дальнейшем могут соответствовать версии модели. Программный модуль Enterprise Modeler предоставляет широкие возможности по управлению версиями. В каждый момент времени пользователь может быть активен в одной текущей версии. Вместе с тем, каждая версия бизнес-модели может состоять из компонентов, полученных из разных версий. Для формирования версий бизнес-моделей в разных операционных средах широко используются средства экспорта/импорта моделей.

Разработка конкретной организационной схемы

Разработка конкретной организационной схемы сводится к назначению исполняемым ролям участников бизнес-процессов (типам работников) конкретных исполнителей с указанием основных его атрибутов (ФИО и прав доступа).

IV. Конфигурация информационной системы

Настройка программных модулей (сеансов BAAN)

При выборе пользователем фазы оптимизации проектной модели система автоматически выбирает по правилам статических условий и активизирует все необходимые работы бизнес-процессов (сеансы BAAN), оставляя неиспользуемые сеансы на графической диаграмме в осветленном виде. Пример выбора работ бизнес-

процесса, соответствующих фазе внедрения представлен в приложении 5 на рис. П5.8.

Посредством правил установки параметров программные модули (сеансы BAAN) получают автоматически значения необходимых режимов работы. Программное обеспечение BAAN IV позволяет распечатывать и вручную корректировать предложенные значения параметров. При этом могут быть отражены различия между установкой предложенных и текущих параметров.

Генерация пользовательского диалога

Генератор интерфейса пользователя BAAN IV используется для создания экранов диалога в виде выпадающего меню или блок-схемы бизнес-процесса, по которой можно осуществлять навигацию и запуск работ (сеансов BAAN) на выполнение.

Процесс создания интерфейса пользователя управляется из референтной модели или проектной модели бизнес-процесса. В частности генерация структуры меню пользовательского диалога осуществляется по фазам оптимизации проектной модели (фазам внедрения). Если в поле "Фаза оптимизации" данные не введены, то генерация меню выполняется на основании информации о всех фазах (см. приложение 5 рис. П5.9).

Пользовательский интерфейс может создаваться как для каждого отдельного пользователя, так и в виде общего диалога для всех участников бизнес-процесса. В первом случае автоматически определяются права доступа пользователей. В результате генерации всем пользователям назначаются отдельные структуры меню с открытыми для использования сеансами BAAN, на обращение к которым у пользователей имеются полномочия. Во втором случае роли пользователей могут не приниматься во внимание. Тогда в меню включаются все работы независимо от того, кто конкретно будет работать на автоматизированном рабочем месте (см. приложение 5 рис. П5.9). Достоинство получаемого интерфейса заключается в его персонификации в виде автоматизированных рабочих мест и в возможности отслеживания с помощью графического диалога пользователя выполняемой работы в рамках отраженной в бизнес-процессе технологии (см. приложение 5 рис. П5.10).

Что следует запомнить:

Корпоративная информационная система (ИС) - система сбора, регистрации, передачи, накопления, обработки и распространения информации, предназначенная для реализации функций управления: целеполагания, прогнозирования, планирования, учета, контроля, анализа хозяйственной деятельности, оперативного регулирования.

Адаптивность ИС - способность ИС быстро и адекватно изменять структуру программного, информационного, технического, организационного обеспечения в соответствии с изменяющимися бизнес-процессами и информационными потребностями предприятия.

Интеллектуализация ИС - обеспечение адаптивности ИС на основе постоянно обновляемой в репозитории (специальной базе знаний) модели предприятия.

Конфигурация ИС - настройка параметров и связывания типовых программных модулей в соответствии с моделью предприятия.

Методология проектирования ИС предполагает наличие некоторой концепции, принципов проектирования, реализуемых набором методов, которые в свою очередь должны поддерживаться некоторыми инструментальными программными средствами.

Система комплексной автоматизации предприятия - набор типовых проектных решений, охватывающих всю необходимую совокупность функциональных компонентов ИС (программных модулей).

Репозиторий - хранилище описания модели предприятия (метаинформации).

Модель предприятия (проблемной области) или Бизнес-модель - отражение объектов, функций, правил, процессов, которые определяют структуру информационной системы.

Модель основных данных - функционально-полная модель гипотетического предприятия, заложенная в некоторой системе комплексной автоматизации предприятия, поддерживается в репозитории.

Типовая (референтная) бизнес-модель - модель, которая является подмножеством модели основных данных для какой-либо отрасли или типа производства.

Проектная модель - модель конкретного предприятия, для которого создается ИС, которая представляет подмножество референтной модели и/или модели основных данных.

Модель бизнес-организации - модель организационной структуры предприятия.

Бизнес-цель (цель) - критерий эффективности функционирования предприятия.

Бизнес-функция (функция) - это некоторый функциональный вид деятельности предприятия. Бизнес-функции могут быть связаны отношениями агрегации «целое-часть» и организованы в иерархическую систему.

Бизнес-процесс (процесс) - реализация одной бизнес-функции, как совокупность работ, выполняемых в определенной последовательности. Бизнес-процесс может включать вложенные подпроцессы.

Бизнес-правило (правило) - отражает совместимость бизнес-функций и/или бизнес-процессов, реализацию бизнес-функций бизнес-процессами, параметры и условия выполнения бизнес-процессов. Используются для проверки корректности моделирования предприятия и конфигурации ИС.

Литература

1. Автоматизация управления предприятием / Баронов В.В., Калянов Г.Н., Попов Ю.Н., Рыбников А.И., Титовский И. Н. – М.: ИНФРА-М, 2000.
2. Информационные системы в экономике: Учебник / Под ред. проф. В.В. Дика. - М.: Финансы и статистика, 1996.
3. Калянов Г.Н. Консалтинг при автоматизации предприятий: Научно-практическое издание. Серия «Информатизация России на пороге XXI века». - М.: СИНТЕГ, 1997.
4. Одинцов А.В., Норенков Ю.И., Горин А.Д. Динамическое моделирование предприятия. - Информационные технологии, 1997, № 2.
5. Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса. М.: Финансы и статистика, 1997.

6. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем. М.: Финансы и статистика, 2001.
7. Тельнов Ю.Ф. Компонентная технология реинжиниринга бизнес-процессов и конфигурации информационной системы предприятия на основе управления знаниями. / Труды 7-ой Национальной конференции по искусственному интеллекту с международным участием (КИИ'2000). - М.: Издательство Физико-математической литературы, 2000.
8. Тельнов Ю.Ф. Компонентная технология реинжиниринга бизнес-процессов. В кн.: Реинжиниринг бизнес-процессов на основе современных информационных технологий / Материалы 4-й Российской научно-практической конференции. - М.: МЭСИ, 2000.
9. Юдицкий С.А., Кутанов А.Т. Технология проектирования архитектуры информационно-управляющих систем. - М.: Институт проблем управления, 1993.
10. BAAN - Подсистема «Методология внедрения». Компания BAAN-Россия, М.:Москва, 1997.
11. Lucas H.C. Information Technology for Management. Sixth edition. International Editions, 1997.
12. Perreault Yves, Vlasic Tom. Implementing BAAN IV, QUE Corp., 1998.
13. Henk Rietveld. Dynamic Business Modeling in Baan Business Organizer 4.0, 1995.
14. Ross R.G. The Business Rule Book. Classifying, Defining and Modelling Rules. Data Base Research Group, Inc. -1997.

Глава 7. Технология создания систем управления знаниями

7.1. Этапы создания системы управления знаниями

Проектирование СУЗ (рис.7.1) декомпозируется на этапы, которые свойственны разработке любой другой интеллектуальной информационной системы (см. параграф 2.1). Вместе с тем, у СУЗ есть целый ряд особенностей, которые определяют характер выполнения отдельных этапов. К таким особенностям относятся:

- Коллективное использование знаний предполагает объединение и распределение источников знаний по различным субъектам, а, следовательно, решение организационных вопросов администрирования и оптимизации деловых процессов, связывающих пользователей СУЗ.
- Состав источников знаний определяется в принципе, конкретные источники знаний, особенно внешние источники знаний, могут добавляться по мере развития проекта. Таким образом, задача проектирования СУЗ приобретает непрерывный характер.
- Поскольку СУЗ имеет многоцелевое назначение, возникает потребность в интеграции разнообразных источников знаний на основе единого семантического описания пространства знаний.

Первые две особенности предопределяют изменения в организации процесса проектирования СУЗ. Прежде всего, в составе проектной группы, занимающейся идентификацией проекта, в определенной степени стирается грань между ролями участников в качестве экспертов и пользователей, поскольку один и тот же человек при эксплуатации СУЗ может быть и источником, и получателем знаний. Следовательно, организация проектных групп должна в большей степени соответствовать требованиям создания команд реинжиниринга бизнес-процессов [6,12,16], в которых участники равноправны и основные задачи сводятся к определению функциональной направленности СУЗ: какие функции должны выполняться, какие знания должны интегрироваться, каков должен быть регламент предоставления и использования знаний. Таким образом, проектные группы в большей степени ориентированы на решение задачи, что должна делать СУЗ, а не как эти функции реализуются. В дальнейшем разработанный регламент

функционирования СУЗ поддерживается специально созданной службой (подразделением) управления знаний.

Концептуальная проработка реализации СУЗ с учетом третьей особенности проектирования СУЗ в основном сводится к созданию онтологии, которое выполняется классически в результате взаимодействия инженеров по знаниям и экспертов. Разработка и поддержка онтологии в масштабе целого предприятия требует постоянных усилий для ее развития. Так как предприятия часто вовлечены в различные виды деятельности, то для одной СУЗ может потребоваться несколько онтологий. Для сокращения затрат на разработку онтологий целесообразно использовать онтологии, разработанные специализированными проектными организациями, которые могут использоваться на принципах тиражирования (разделения доступа) и повторного использования в форме подписки на централизованное обслуживание.

Программная реализация СУЗ имеет меньшее значение, поскольку в основном сводится к адекватному выбору инструментальных средств и их настройке. Проблемы программной реализации могут возникнуть при интеграции разнородных инструментов поддержки СУЗ, например, документальной поисковой системы, баз знаний прецедентов и электронной системы коммуникаций. В этом случае может потребоваться специализированная программная реализация с учетом требований разрабатываемой СУЗ.

В дальнейшем в учебном пособии с учетом содержания излагаемого предмета основной упор делается на проектирование и реализацию онтологий.

7.2. Идентификация проблемной области

Определение состава задач (назначение). На стадии идентификации проблемной области, в первую очередь, определяется состав решаемых задач. В принципе возможно создание узкоспециализированных систем по отдельным функциям управления: маркетинга, инжиниринга, менеджмента отдельных типов процессов: продаж, закупок, финансов и т.д. Однако интегрированный характер применения СУЗ делает целесообразным их создание для цепочек взаимосвязанных процессов, например, инновации объединяют маркетинг, инжиниринг, рекламу, бизнес-

планирование; стратегическое планирование – инновации, финансовый менеджмент, контроллинг бизнес-процессов различных видов и т.д. Другое дело, что разработка СУЗ может начинаться с отдельных областей, например, с маркетинга, не требуя одновременной разработки всех требуемых онтологий и источников знаний. Интерфейсная (инструментальная) и технологическая часть не зависит от числа решаемых задач и источников знаний, может апробироваться на отдельных функциональных приложениях.

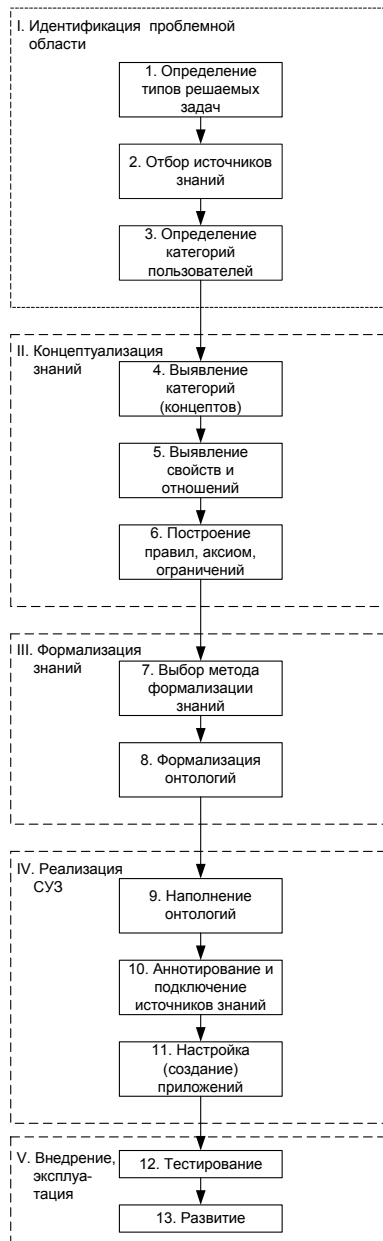


Рис. 7.1. Этапы проектирования СУЭ

Отбор источников знаний. В качестве источников знаний легче всего подсоединяются базы данных оперативных информационных систем через механизм создания информационных хранилищ. Аналогично подсоединяется система электронных и неэлектронных архивов документов, которая может иметь централизованную и децентрализованную схему управления.

Для создания баз знаний прецедентов требуется определить набор типовых бизнес-процессов, для которых будут отбираться прецеденты, например, разработка проектов, заключение договоров, проведение PR-акций и т.д.

Для формализации знаний специалистов (экспертов) необходимо построить карты знаний, в которых была бы отражена степень владения работниками различными категориями знаний, отражаемых как в документальной, так и не документальной форме.

Состав внешних источников знаний определяется в принципе, это могут быть: базы данных статистических органов, базы знаний государственных органов и специализированных коммерческих и консалтинговых компаний, базы данных и знаний партнеров, клиентов, конкурентов. Предпочтение отдается базам знаний, которые проаннотированы в каком-либо общем для СУЗ стандарте описания знаний. При этом рассматривается ценовой фактор доступа к информационным ресурсам.

Определение категорий пользователей

Карты знаний облегчают и выявление основных категорий пользователей, к которым могут относиться:

- Лица, принимающие решения.
- Аналитики, подготавливающие информацию для принятия решений.
- Технические работники, обновляющие знания.
- Новички, требующие обучения.
- Смешанные категории пользователей.

Для категорий пользователей в дальнейшем будут определены права доступа к тем или иным источникам знаний.

Требования к организации СУЗ оформляются в виде различных диаграмм деловых процессов, например, с помощью инструментального средства ARIS Toolset [6,16].

Центральное место в проектировании СУЗ занимает онтология, которая определяет и интегрирует все источники знаний

и приложения. Требования к разработке онтологии оформляются в специальном документе, например, в следующем виде [19] (табл. 7.1):

Таблица 7.1.

Спецификация требований к онтологии

Предметная область:	Подбор и повышение квалификации персонала в компании ABC
Назначение:	<p>Онтология служит для обмена знаниями между департаментом управления персоналом и менеджерами проектов при отборе персонала требуемой квалификации</p> <p>Онтология используется для семантического поиска необходимых квалификационных характеристик для выполнения определенных видов работ</p>
Область значений:	Онтология содержит концепты (категории) управления персоналом. Концепты используемых квалификаций в технологиях и на рынке рассматриваются детально, концепты управления бизнесом на более общем уровне. Включает 400-600 понятий.
Поддерживаемые приложения:	Система управления квалификаций персонала в ИНТРАНЕТ среде (Skill Management System)
Источники знаний:	<ul style="list-style-type: none"> – Web страницы департамента управления персоналом – Руководство о развитии персонала (внутренний документ) – Спецификация продукции и технологий (внутренний документ) – Интервью с работниками департамента управления персоналом и менеджерами проектов

7.3. Концептуализация знаний с помощью онтологий

Онтология – это точное (явное) описание концептуализации знаний [21,22]. Ряд авторов под онтологиями понимают согласованные словари терминов и их значений. В более широком смысле онтология больше, чем просто согласованный словарь терминов. Назначение онтологий сводится к обеспечению следующих возможностей [7]:

1. Повышение интеллектуальности систем управления знаниями на основе представления того, что часто остается неявным. Во всех видах человеческой деятельности находятся *предположения*, которые обычно остаются неявными. Базы знаний традиционных экспертных систем основывается на реализации концептуальной модели, которая остается скрытой для пользователя и потому трудно понимаемой. В отличие от экспертных систем онтология формализует неявные *предположения* и концептуализацию. Декларативное представление в онтологиях концептуализации на основе семантических отношений и аксиоматики делает системы управления знаниями интеллектуальными.
2. Стандартизация на основе описания целевого мира в виде словаря, согласованного среди людей. Таким образом, возможна спецификация компонентов функциональности, понимание и сравнение различных систем в виде общего словаря. Также с помощью онтологий решается задача разделения знаний между различными пользователями и/или компьютерными системами (совместного использования), а также их повторное использование для новых ситуаций.
3. Систематизация знаний, которая позволяет интегрировать разнородные источники знаний на основе единой многоаспектной таксономии, представляемой в общем словаре.
4. Реализация метамодельной функциональности для конструирования. Онтология снабжает необходимыми понятиями, отношениями и ограничениями, которые используются как строительные блоки для построения конкретной модели решения задачи. Далее модели, построенные на основе онтологии, выполняются с использованием модулей, привязанных к выбранным понятиям.

5. Создание теории содержания. Онтология дает теорию содержания, позволяя постепенно обобщать понятия конкретной проблемной области.

К проектированию онтологии знаний предъявляются следующие требования [21]:

1. *Ясность* — четкая передача смысла введенных терминов (концептов).
2. *Согласованность* — логическая непротиворечивость определений.
3. *Расширяемость* — возможность монотонного расширения и/или специализации без необходимости пересмотра уже существующих понятий.
4. *Минимум влияния кодирования* — инвариантность к методам представления знаний.
5. *Минимум онтологических обязательств* — отражение только наиболее существенных предположений о моделируемом мире.

Структура знаний, представляемая в онтологии, должна отображать постоянные парадигматические отношения понятий, независимые от контекста решения задачи, и правила формирования переменных синтагматических отношений понятий, возникающих в некотором контексте решения задачи.

Парадигматические отношения понятий. К таким отношениям относятся отношения синонимии, омонимии, полисемии, обобщения («Род-Вид»), агрегации («Целое-Часть»), «Причина – Следствие», «Цель – Средство», «Аргумент-Функция» и ролевые ассоциации «Использует», «Обрабатывает» и т.д., которые превращают словарь в тезаурус. Пример фрагмента онтологии информационных систем представлен на рис.7.2.

Для понятий онтологии могут быть также заданы свойства (атрибуты), формулы вычислений и примеры экземпляров. Например, свойства использования операций проектирования различными категориями участников процесса разработки информационной системы представлены на рис. 7.3.

Онтология отражает *синтагматические отношения понятий* в виде семантических ограничений (аксиом, логических правил). С помощью правил, с одной стороны, можно правильно интерпретировать смысловые отношения в конкретном контексте

решения задачи, то есть контролировать целостность определений пользователя, а с другой стороны, можно расширять базовый список понятий за счет использования зависимостей свойств, которые связывают их с другими понятиями, то есть осуществлять логический вывод.

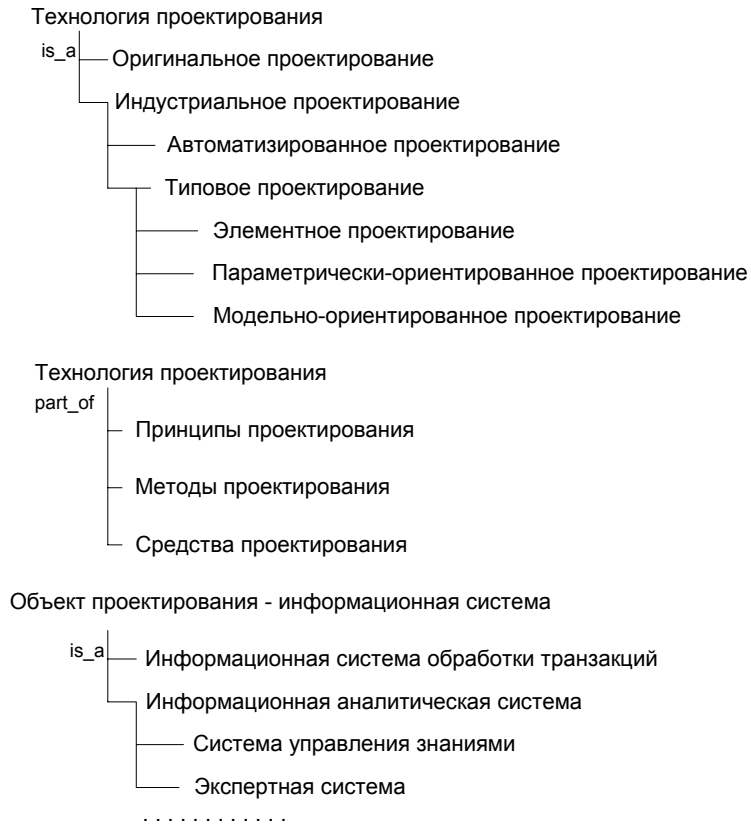


Рис. 7.2. Фрагмент онтологии информационных систем

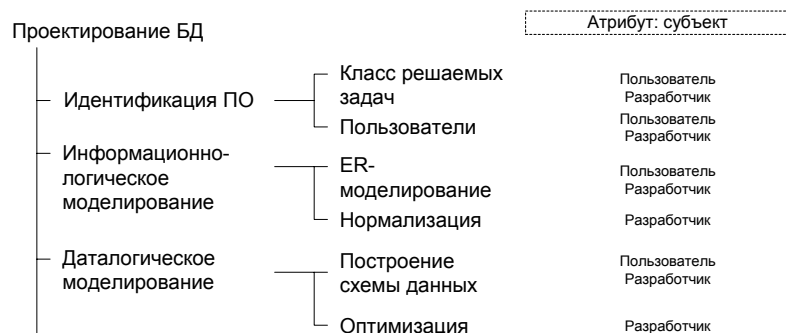


Рис. 7.3. Фрагмент онтологии проектирования БД

Различают как стандартные правила, например, симметричности, инверсности, транзитивности, так и специализированные правила, например:

Если Реинжиниринг бизнес-процессов = «проводится»

То Проектирование информационной системы = «проводится»

Онтологическое знание организуется на трех уровнях:

- онтология верхнего уровня (метаонтология);
- онтология предметной области;
- онтология задач.

Метаонтология или *базовая онтология* отражает такие общие понятия, как «сущность», «класс», «свойство», «значение», «типы данных», «типы отношений», «процесс», «событие» и т.д. Определения общих категорий позволяют системе, прежде всего, контролировать синтаксические конструкции понятий предметных и проблемных областей, которые объявляются, как наследники общих категорий.

Онтология предметной области определяет набор понятий, используемых при решении различных интеллектуальных задач и независимых от самого метода решения задач.

При построении онтологии предметной области проводится классификация (таксономия) понятий предметной области, выявляются свойства и отношения понятий, строятся процедурные определения (логические правила), расширяющие семантику модели предметной области.

Онтология задач имеет дело с понятиями, описывающими методы преобразования объектов предметной области в процессе решения задач. Например, для задач обучения в качестве методов обучения могут использоваться дедуктивный (от общего к частному), индуктивный (от частного к общему), абдуктивный (от частного к частному) методы; для задач стратегического анализа - SWOT-анализ, анализ системы сбалансированных показателей, проектный анализ и т.д. С помощью понятий, свойств и отношений описывается сущность используемых методов, устанавливается последовательность их выполнения. Правила определяют условия применения методов при конкретных обстоятельствах. Пример одного из таких правил в задачах обучения представлен на рис. 7.4.

Если ТипОбучаемого = ТипСубъекта И
 ЦельОбучаемого = ЦельУчебногоЭлемента
 То Выдать Список учебных-элементов, соответствующих концептам-стадиям ПО,
 инициируемых ТипомСубъекта и ЦелиУчебногоЭлемента,
 в последовательности процесса

Рис. 7.4. Пример правила онтологии задач

Введение онтологии задач позволяет расширить класс интеллектуальных задач, решаемых с помощью СУЗ, в частности перейти от простых поисковых задач к задачам конфигурации, когда система автоматически разбивает задачу на подзадачи, для каждой подзадачи выбирает метод решения задачи, а для каждого метода отбирает необходимые единицы предметных знаний. Такая СУЗ является не просто интеллектуальной информационно-поисковой системой, но и системой, которая планирует и генерирует решение задачи. В этом аспекте СУЗ должна обладать развитым механизмом вывода и по своей реализации сближается с классом экспертных систем, но на более развитой семантической основе.

7.4. Формализация онтологического знания

В основе формализации онтологий, с одной стороны, лежат общепризнанные методы представления знаний, к которым в первую очередь относят методы исчисления предикатов и объектно-ориентированного подхода: семантические сети и фреймы. В частности, языки KIF (Knowledge Interchange Format) [23],

основанный на исчислении предикатов, и ОКВС (Open Knowledge Base Connectivity) [20], основанный на фреймовом представлении знаний, позволяют хорошо отображать логическое представление онтологий, транслируемое в конкретные модели представления знаний СУЗ.

С другой стороны, с учетом широкого применения в СУЗ гипертекстовых средств навигации стандартных браузеров, подключения внешних источников знаний и распространения знаний в среде ИНТЕРНЕТ и ИНТРАНЕТ, возникает потребность в расширении известных языков разметки HTML и XML методами описания онтологических знаний с помощью специальных семантических конструкций. Онтологические расширения языков разметки HTML и XML позволяют разрабатывать и использовать различные семантические анализаторы. В этом смысле концептуальное представление действительно не зависит от программной среды. Рассмотрим использование языков представления онтологического знания более подробно.

1. Языки, основанные на исчислении предикатов

Языки данной группы основаны на декларативной семантике и являются логически всесторонними, т.е. обеспечивают выражение произвольных логических предложений. Кроме того, с помощью этих языков хорошо представляется метазнание. Это позволяет пользователю представлять знания в явном виде и разрешает пользователю описывать новые конструкции представления знаний без изменения самого языка. Наиболее широко используемым языком данной группы является язык KIF. Хотя KIF не предназначен для совместного использования с программами в качестве языка представления знаний, он может быть использован и для этих целей. Специальные инструментальные средства, например Ontolingua, транслируют предложения KIF в конкретные представления знаний: Loom на базе KL-ONE (MacGregor, 1991), Epikit (Genesereth, 1990), Algernon на базе фреймовой системы Access Limited Logic (Crawford & Kuipers, 1989), относительно просто написать трансляторы в CycL (Lenat & Guha, 1990), KEE (Fikes & Kehler, 1985), EXPRESS (Spiby, 1991) [22].

Язык KIF представляет собой язык, который разработан специально для обмена знаниями между различными программными агентами, созданными в отличающейся программно-технической

среде и обслуживающими потребности различных участников системы управления знаниями. Обычно, при взаимодействии программного агента с источником знаний, представленным в другом формате, происходит преобразование формата онтологии во внутреннее представление знаний агента. С этой точки зрения KIF является действительно концептуальным языком обмена знаниями, а не языком представления знаний.

Язык KIF имеет статус международного стандарта:

- ISO/IEC 10646-1:1993, *Information Technology (IT) - Universal Multiple-Octet Coded Character Set (UCS)*.
- ISO/IEC 14481:1998, *Information Technology (IT) - Conceptual Schema Modeling Facilities (CSMF)*.

Язык KIF представляет префиксную нотацию для исчисления предикатов и является LISP-подобным языком обработки списков. В языке KIF используются три типа выражений: термы – для обозначения объектов, предложения для отражения фактов и определения для представления постоянных зависимостей фактов: отношений, классов объектов, функций, которые реализуют аксиоматику (правила) интерпретации фактов.

В качестве термов используются индивидуальные переменные, функциональные термы, термы отношений, термы списков и т.д. Переменные начинаются с префикса ? и могут использоваться в кванторах всеобщности (for all) и существования (exists).

Отношения представляют основные конструктивные элементы для отражения структуры знания в виде предложений и определяют множества кортежей. Отношения обозначаются константами, которые выполняют роль предикатных символов. Функции представляют специальный вид отношений с N+1 аргументами, в которых (N+1)-й аргумент представляет результат выполнения функции на N аргументах. Классы объектов представляются унарными отношениями. Например, (C ?q) означает, что объект, соответствующий переменной ?q, является экземпляром класса C.

Предложения формируются из списков, в которых первым элементом является константа отношения, а остальные элементы являются термами. В предложениях используются операторы

конъюнкции (and), дизъюнкции (or), импликации (\Rightarrow), эквивалентности (\Leftrightarrow) и отрицания (not). Например, следующее логическое предложение, соответствующее правилу формирования вопроса в обучающей системе «Задавать вопросы новичку, которые не должны соответствовать знаниям эксперта» в нотации KIF, будет иметь следующий вид:

```
(forall ?L
  ( $\Rightarrow$  (newcomer ?L)
    (not exists ?Q)
      (and (learner ?L)
        (question ?Q)
          (knowledge ?L ?Q)
            (not (expert ?L))))))
```

Символы newcomer (новичок), learner (ученик), question (вопрос), knowledge (знание), expert (эксперт) являются константами отношений (предикатами). Выражения ?L, ?Q - обозначают переменные, соответственно личность и элемент знаний. Предложение (newcomer ?L) отражает факт, что объекты, соответствующие переменной ?L, являются новичками. Предложение not (expert ?L) означает отрицание факта, что личности ?L являются экспертами (отлично освоившими курс обучения). Предложение question ?Q определяют вопросы к элементам знаний ?Q. Предложение knowledge отражает бинарное отношение, которое определяет знание элементов ?Q личностями ?L.

Определения в языке KIF задаются операторами defrelation, deffunction и defobject, которые соответствуют определениям отношений, функций и классов объектов. Рассмотрим примеры из [22].

Определение отношения имеет следующий вид:

(defrelation CONNECTS (?comp1 ?comp2)

"Бинарное отношение связи между компонентами.

Связанные компоненты могут быть частями друг друга."

```
:def (and (component ?comp1)
          (component ?comp2)
          (not (subpart-of ?comp1 ?comp2))
          (not (subpart-of ?comp2 ?comp1))))
```

Пример соответствует бинарному отношению двух компонентов. Аргументы ?comp1 and ?comp2 определены кванторами всеобщности. Предложение после ключевого слова :def означает логическое ограничение на аргументах, которое отражает условие необходимости. Вместо :def можно использовать знак импликации \Rightarrow . На языке исчисления предикатов данное условие будет иметь вид:

$$\forall x,y \text{ connects}(x,y) \Rightarrow \text{component}(x) \wedge \text{component}(y) \wedge \neg \text{subpart-of}(x,y) \wedge \neg \text{subpart-of}(y,x).$$

Если вместо :def использовать ключевое слово :iff-def или знак эквивалентности \Leftrightarrow , то логическое ограничение будет определять условие необходимости и достаточности, т.е.:

$$\forall x,y \text{ connects}(x,y) \Leftrightarrow \text{component}(x) \wedge \text{component}(y) \wedge \neg \text{subpart-of}(x,y) \wedge \neg \text{subpart-of}(y,x).$$

Класс объектов определяется как унарное отношение. Например,

(defclass CONNECTION (?connection)

"Соединение есть специальный тип модуля для передачи энергии/вещества/информации, который не имеет внутренних переменных."

```
:def (and (module ?connection)
          (not (exists ?variable
                      (internal-state-variables ?connection ?variable)))))
```

Предложение `:def` задает условие необходимости. Соединение определяется как подкласс класса объектов модули, который не имеет внутренних переменных. Логическое выражение имеет следующий вид:

$$\forall x [\text{connection}(x) \Rightarrow \text{module}(x) \wedge \neg \exists y \text{ internal-state-variables}(x,y)]$$

Функция определяется подобно отношению, только результатный аргумент выводится из списка аргументов. Например, функция возведения в квадрат имеет следующее определение:

```
(deffunction SQUARED (?n) :-> ?value
  "Возведение в квадрат есть умножение числа на себя."
  :def (and (number ?n)
            (nonnegative-number ?value))
  :lambda-body (* ?n ?n))
```

Как в определении отношений, аргументы функции ограничиваются условием необходимости, следующим за ключевым словом `:def`. В частности для аргументов задается ограничение на доменах, например, условие `(number ?n)` означает, что функция определена только для чисел. Ограничение диапазона значений переменной функции специфицируется предложением, ограничивающим значение функции после ключевого слова `:->` (например, `?value` есть неотрицательное число). В теле `deffunction` разрешается использование предложения `:lambda-body`, являющееся выражением терма KIF, которое назначает значению функции выражение на аргументах. Например, `*` есть функция и `lambda-body` есть терм, который назначает произведение числа на себя. Логическое выражение определенной в примере функции имеет вид:

$$\forall x,y [y = \text{squared}(x)] \Rightarrow \text{number}(x) \wedge \text{nonnegative-number}(y)$$

$$\forall x \text{ squared}(x) = x^2$$

Определения классов в языке KIF отображаются в различные фреймовые представления знаний (KEE, KL-ONE и др.). Например, определение класса авторов в языке KIF:

```
(define-class AUTHOR (?author)
  "Автор – это личность, которая создает документы. Автор
  должен создать по крайней мере один документ. Автор известен по
  своему имени."
  :def (and (person ?author)
    (= (value-cardinality ?author AUTHOR.NAME) 1)
    (value-type ?author AUTHOR.NAME biblio-name)
    (>= (value-cardinality ?author AUTHOR.DOCUMENTS) 1)
    (<=> (author.name ?author ?name)
      (person.name ?author ?name))))
```

может быть преобразовано в следующее каноническое фреймовое представление:

```
(define-class AUTHOR (?author)
  :axiom-def (and (subclass-of author person)
    (slot-value-cardinality author author.name 1)
    (slot-value-type author author.name biblio-name)
    (minimum-slot-cardinality author author.documents 1)
    (same-slot-values author author.name person.name)))
```

Метка `:axiom-def` указывает, что эта аксиома определена без свободных переменных. Предложения в `:axiom-def` определяют константу `author`, а не переменную экземпляра `?author`. Такое каноническое определение проходит через трансляторы Ontolingua.

2. HTML – подобные языки

В качестве основы для языков описания онтологий и онтологического аннотирования текстов может выступать язык разметки данных HTML, дополненный специальными тегами (указателями, управляющими дескрипторами). С помощью тегов происходит выделение семантических фрагментов текста, которые унифицировано интерпретируются семантическими анализаторами различных программных средств.

Языки данной группы в целом позволяют описать объекты онтологии (далее будем называть их концептами), отношения между ними и определить некоторые правила вывода. Основное назначение таких языков состоит в возможности описания онтологии, аннотирования необходимых WEB-страниц концептами онтологии и

дальнейшее осуществление поиска данных WEB-страниц с помощью специальной поисковой машины. Использование данных языков позволяет при осуществлении поиска в сети выводить более релевантную информацию на запросы пользователей, чем при использовании стандартных поисковых механизмов. Расширение HTML онтологическими конструкциями осуществляется в инструментальном средстве Ontobroker с помощью анкерных тегов `<a onto >` [1]. В инструментальном средстве SHOE (Simple Ontology Extensions - проект исследовательской группы PLUS Университета штата Мэриленд [29]) язык HTML расширяется набором специализированных тегов.

Онтология, описываемая с помощью SHOE, представляет собой иерархию концептов (категорий, классов), связанных отношением «род-вид» (IS-A), отношения между этими концептами (либо свойства этих концептов) и правила.

При описании онтологии в заголовке HTML-документа указывается тот факт что, онтология создана на основе определенной спецификации:

```
<META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">
```

Затем, в теле документа, открывается тег `<ONTOLOGY>` — это начало онтологии (закрывающий тег `</ONTOLOGY>` ставится в конце онтологии), после чего, в этом же теге, определяются характеристики онтологии:

- ID="уникальный идентификатор онтологии",
- VERSION="версия онтологии — строковая переменная, описывающая версию онтологии (чувствительна к регистру, допустимые символы: буквы, цифры, дефис)",
- BACKWARD-COMPATIBLE-WITH = "список предыдущих версий",
- DESCRIPTION = "описание данной онтологии".

Например, имеем следующее описание:

```
<ONTOLOGY  
  id=cs-dept-ontology
```

```
DESCRIPTION="Онтология высшего учебного заведения"  
VERSION="1.0">
```

```
.....  
</ONTOLOGY>
```

Если создаваемая онтология основывается на уже имеющихся в других онтологиях классах или отношениях, следует указать ссылки на них с помощью тега `<USE-ONTOLOGY>` с обязательными характеристиками `ID=` "идентификатор онтологии", `VERSION=`"версия онтологии", `PREFIX=` "префикс используемых категорий, отношений и правил из ссылаемой онтологии" и необязательной характеристикой `URL=`"URL документа, содержащего онтологию". Например, имеем следующее описание:

```
<USE-ONTOLOGY  
  id=base-ontology VERSION="1.0"  
  
  URL="http://www.cs.umd.edu/projects/plus/SHOE/onts/base1.0.html"  
  PREFIX="base">
```

Категории онтологии (концепты, классы объектов) образуют иерархию обобщения «род-вид».

Для определения категорий используется тег `<DEF-CATEGORY>` с вложенной обязательной характеристикой `NAME=`"имя класса" и следующими необязательными характеристиками: `ISA=`"имя категории, подклассом которой является описываемый класс", `DESCRIPTION=`"описание данной категории" (обычно полное, понятное человеку описание), `SHORT=`"краткое обозначение категории".

Например, фрагмент определений категорий онтологии имеет следующий вид:

```
<DEF-CATEGORY  
  SHORT="person" ISA="base.SHOEEntity" NAME="Person">  
<DEF-CATEGORY  
  SHORT="worker" ISA="Person" NAME="Worker">  
<DEF-CATEGORY  
  SHORT="faculty member" ISA="Worker" NAME="Faculty  
member">
```

<DEF-CATEGORY

SHORT="professor" ISA="Faculty member" NAME="Professor">

Отношения в SHOE могут иметь как бинарный, так и N-арный характер. Бинарные отношения могут использоваться для определения свойств категорий, задаваемых на стандартных доменах: число, строка, дата.

Для определения отношения между экземплярами категорий (или между экземплярами и данными) используется тег <DEF-RELATION> имеющий следующие характеристики: NAME="имя отношения" и необязательные DESCRIPTION="описание данного отношения" (обычно полное, понятное человеку описание), SHORT="краткое обозначение отношения".

После этого, указав все необходимые описания отношения, тег закрывается, и перечисляются те аргументы, которые участвуют в этом отношении. Важной особенностью является то, что каждый аргумент занимает строго определенную пронумерованную позицию в отношении (аргументы с номерами 1, 2, 3 и т. д.). В бинарных отношениях первый аргумент определяет категорию, из которой выходит отношение (FROM - направление "от"), а второй аргумент – категорию, к которой направлено отношение (TO - направление "к").

Список аргументов участвующих в N-арном отношении представляет собой N-ное количество тегов <DEF-ARG > со следующими характеристиками:

- POS=(целое положительное число, указывающее на место аргумента в отношении | [FROM | TO] — в случае если описывается бинарное отношение, то 1-й аргумент — FROM, а 2-й аргумент — TO)
- TYPE="любая категория либо используемый как аргумент тип данных (указывается пользовательский либо базовый тип данных)"
- SHORT="краткое понятное обозначение аргумента отношения (если необходимо)"

Каждое описанное отношение должно закрываться тегом </DEF-RELATION>.

Пример фрагмента определений отношений имеет следующий вид:

```

<DEF-RELATION
  SHORT="was written by" NAME="publicationAuthor">
  <DEF-ARG TYPE="Publication" POS="1">
  <DEF-ARG SHORT="author" TYPE="Person" POS="2">
</DEF-RELATION>
<DEF-RELATION
  SHORT="was written on" NAME="publicationDate">
  <DEF-ARG TYPE="Publication" POS="1">
  <DEF-ARG TYPE=".DATE" POS="2">
</DEF-RELATION>

```

В онтологиях могут задаваться правила вывода, на основе которых из источников знаний извлекаются неявно относящиеся к запросу аннотированные с помощью SHOE документы.

Правила в SHOE состоят из двух частей — тела («body») и заголовка («head»). В теле правила (часть IF) описываются утверждения, которые соответствуют условию истинности утверждений, описанных в заголовке правила (часть THEN). Следует отметить, что в каждой части может содержаться неограниченное количество утверждений, но не менее одного. В качестве данных, определенных в этих утверждениях, могут выступать как константы, так и переменные. Утверждениями тела могут быть классы, отношения и стандартные отношения («equal», «notEqual», «greaterThan», «greaterThanOrEqual», «lessThanOrEqual», «lessThan»). Утверждениями заголовка являются классы или отношения.

Для того чтобы исключить возможное возникновение двусмысленностей в правилах, должны соблюдаться следующие условия:

- заголовок правила не должен содержать переменных, которые не содержатся где-либо в теле правила;
- если в теле правила содержится более чем одна переменная, то каждая из этих переменных должна быть связана с другими переменными отношениями (помимо прямой связи, учитываются сложные, через другие переменные).

В случае если эти условия не будут соблюдены, правило считается неверным и может быть игнорировано.

Описание правила открывается тегом <DEF-INFERENCE> с необязательной характеристикой DESCRIPTION="описание данного правила".

Описание тела правила заключается между тегами <INF-IF> и </INF-IF>. Внутри этих тегов помещаются утверждения, для описания которых предназначены специальные теги.

Для описания категорий используется тег <CATEGORY> с характеристикой NAME="полный префикс.имя категории", и FOR = "экземпляр" с необязательной пометкой USAGE=("VAR" или "CONST"), которая указывает на то, имеется ли ввиду переменная, то есть по сути любой экземпляр, описанный в аннотированном документе (пометка «VAR»), либо конкретный экземпляр (ключ экземпляра - например, строка URL страницы SHOE) (пометка «CONST» — она установлена по умолчанию). Слово «USAGE=» необязательно, даже если стоит метка «VAR» или «CONST».

Для описания отношений в теле правила используются открывающий тег <RELATION> с характеристикой NAME="полный префикс.имя_отношения" и закрывающий его тег </RELATION>. Между этими тегами помещается список аргументов отношения. Следует, однако, учитывать, что если число аргументов в утверждении–отношении правила больше либо меньше числа аргументов в отношении, на котором оно основывается, то это правило будет неверным и может быть игнорировано. Каждый из аргументов описывается тегом <ARG> с характеристиками: POS=(целое положительное число, указывающее на место аргумента в отношении либо, если имеет место бинарное отношение, одно из зарезервированных слов, также указывающих на место аргумента в отношении FROM или TO); VALUE="ключ экземпляра или имя переменной" с необязательной пометкой USAGE=("VAR" или "CONST").

Для описания сравнений используются открывающий тег <COMPARISON> с характеристикой OP="ключевое_слово" («equal» (равно), «notEqual» (не равно), «greaterThan» (больше), «greaterThanOrEqual» (больше либо равно), «lessThanOrEqual» (меньше либо равно), «lessThan» (меньше)) и закрывающий его тег </COMPARISON>. Между этими тегами помещаются аргументы сравнения, которые описываются аналогично аргументам отношений.

Следует учитывать следующее:

- сравнение показывает то, что аргумент 1 «больше» (меньше, равен, не равен и т.п.) аргумента 2;
- для строк, имен отношений и категорий, ключей экземпляров сравнения «больше/меньше» не имеют значения;
- дата 1 «меньше» даты 2, если дата 1 была раньше даты 2;
- NO «меньше» YES.

Для описания заголовка правила предназначены тег <INF-THEN> и закрывающий его тег </INF-THEN>. Как уже было сказано выше, в заголовке правила могут быть описаны категории и отношения. Синтаксис их описания соответствует синтаксису описания категорий и отношений в теле правила.

Описание правила закрывается тегом </DEF-INFERENCE>.

Пример, фрагмента правил онтологии имеет следующий вид:

```
<DEF-INFERENCE
  DESCRIPTION="Transitivity of Suborganizations. If
  subOrganization(x,y) and subOrganization(y,z) then
  subOrganization(x,z)">
  <INF-IF>
    <RELATION NAME="subOrganization">
      <ARG POS="FROM" VAR VALUE="x">
      <ARG POS="TO" VAR VALUE="y">
    </RELATION>
    <RELATION NAME="subOrganization">
      <ARG POS="FROM" VAR VALUE="y">
      <ARG POS="TO" VAR VALUE="z">
    </RELATION>
  </INF-IF>
  <INF-THEN>
    <RELATION NAME="subOrganization">
      <ARG POS="FROM" VAR VALUE="x">
      <ARG POS="TO" VAR VALUE="z">
    </RELATION>
  </INF-THEN>
</DEF-INFERENCE>
```

Реализации онтологии информационных систем
представлена в приложении.

Аннотирование документов с использованием SHOE

Документы могут аннотироваться с помощью SHOE, используя категории и отношения разных онтологий. Использование каждой онтологии должно объявляться в качестве используемого экземпляра. Объявление экземпляров происходит в теле HTML-документов, причем описание не должно находиться внутри каких-либо других тегов, отличных от <body>. Кроме того, описание экземпляров не должно прерываться какими-либо тегами HTML.

Объявление экземпляра начинается тегом <INSTANCE> с обязательной характеристикой KEY="ключ экземпляра" и необязательной характеристикой DELEGATE-TO="список экземпляров, которым разрешено делать объявления от имени данного экземпляра (ключи в списке разделяются пробелами)". Делегирование списка экземпляров необходимо для того, чтобы объединить разрозненные данные о некотором экземпляре в один документ. Закрывается описание экземпляра тегом </INSTANCE>.

Объявление всех категорий и отношений заключается между тегами <INSTANCE> и </INSTANCE>. Внутри описания экземпляра можно объявить другой экземпляр («подэкземпляр»), что происходит аналогично объявлению самого экземпляра.

Прежде чем описывать экземпляры классов и отношения между ними, следует определить, из какой онтологии эти элементы будут взяты, и выбрать для каждой из них определенный уникальный префикс. Для объявления той или иной онтологии используется тег <USE-ONTOLOGY> с обязательными характеристиками ID="идентификатор онтологии", VERSION="версия онтологии", PREFIX="используемый префикс" и необязательной характеристикой URL="URL документа, содержащего онтологию". Таким образом, объявление используемых онтологий одинаково как для описания экземпляра, так и для описания самой онтологии.

Экземпляры могут быть классифицированы. Они могут быть определены, как относящиеся к одной или нескольким категориям. Для этого используется тег <CATEGORY> с обязательной характеристикой NAME="полный префикс.имя категории" и необязательной характеристикой FOR="ключ экземпляра". Если последняя характеристика не указана, то к объявленной категории

относится весь объявляемый экземпляр. В противном случае следует указать ключ.

Экземпляры могут объявлять отношения с элементами, для чего используются тег `<RELATION>` с обязательной характеристикой `NAME="полный префикс.имя отношения">` и закрывающий его тег `</RELATION>`. Между этими тегами располагается список аргументов отношения, объявляемых тегом `<ARG>` с характеристиками `POS=(целое положительное или FROM или TO (для бинарного отношения))` и `VALUE="ключ экземпляра"`.

Если «ключом экземпляра» в бинарном отношении является ключ описываемого экземпляра, то описание этого аргумента в отношении можно опустить. Это верно только в том случае, если в характеристике `POS` используются слова `FROM` и `TO` для определения места аргумента в отношении. В случае если для определения места аргумента используются числа, это будет ошибкой. Также ошибочным будет присваивание разным аргументам одинаковой позиции. Но если в объявлении отношения аргументов не будет вовсе, то это отношение будет принято как не содержащее аргументов.

3. XML-подобные языки

В качестве основы для таких языков выступает расширяемый язык разметки XML, что уже само по себе дает данным языкам намного более широкие возможности в области представления данных, чем языкам, основанным на HTML. В настоящее время существует порядка 20 различных языков, основанных на языке XML с добавлением расширенных возможностей, позволяющих описать онтологию. В качестве признанных международными стандартами языков являются: XML, RDF (Resource Description Framework) [17], OXML, DAML+OIL [26].

Сам язык XML в принципе не обладает практически никакими возможностями в области представления онтологий. В нем отсутствуют специальные конструкции, позволяющие описать взаимоотношения между концептами онтологии, правила вывода, ограничения и т.д. Данный язык предназначен исключительно для представления данных. Основным достоинством языка является то, что для работы с документами, подготовленными с помощью него, достаточно обычного интернет-браузера, т.е. не требуется никаких

дополнительных средств. Но, вместе с тем, это и накладывает некоторые ограничения, так как отсутствует «интеллектуальность» при поиске информации.

XML – документ представляет собой размеченное дерево (рис. 7.5), в котором каждый узел имеет следующую конструкцию: метка + атрибут/значение + содержание.

Вместе с тем, XML не пригоден для Web-ресурсов широкого назначения, разделенного использования источников знаний в СУЗ, требуется заранее подготовленное соглашение о метках, примитивах и доменах словаря, в основном используется в ИНТРАНЕТ приложениях или небольших сообществах ИНТЕРНЕТ пользователей.

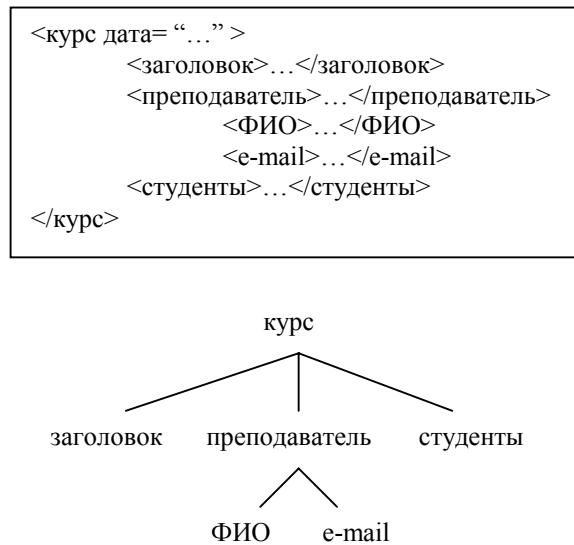


Рис. 7.5. Структура XML представления описания учебного курса

Языки RDF и DAML+OIL имеют значительно более богатые возможности в описании онтологий по сравнению с XML, хотя сами представляют расширение XML. Данные языки позволяют (в разной степени) описать концепты, отношения между ними, поддерживают

иерархию концептов и их наследование, а также иерархию и наследование свойств концептов, позволяют задавать некоторые правила вывода.

RDF (Resource Description Framework) – язык описания Web-ресурсов

Язык онтологий для Web-приложений создан международной организацией W3C (World Wide Web Consortium) при участии представителей различных заинтересованных организаций, поддерживается Netscape.

Базовым строительным блоком в RDF является триплет «объект — атрибут — значение», часто записываемый в виде A(O,V), который читается, как: объект O имеет атрибут A со значением V. Например,

HasName ('http://www.w3.org/employee/id1321', «Jim Lerner»)

AuthorOf('http://www.w3.org/employee/id1321', 'http://www.books.org/ISBN0012515866')

HasPrice('http://www.books.org/ISBN0012515866', «\$62»).

Эту связь можно представить в семантической сети, как ребро с меткой A, соединяющее два узла, O и V:

[O]-A—>[V].

В качестве значений могут выступать объекты, таким образом, организуется связывание триплетов. На рис. 7.6. показан пример графического представления семантической сети, связывающей два типа триплетов: автор книги (author-of) и опубликовано (publ-by).

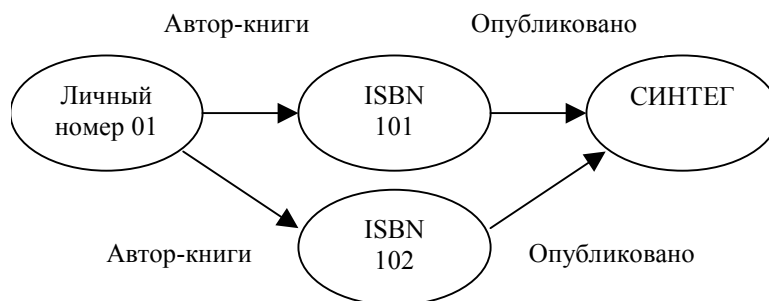


Рис. 7.6. Пример графического представления семантической сети в RDF

Кроме того, RDF допускает форму представления, в которой любое выражение RDF в триплете может быть объектом или значением, в качестве значений могут выступать триплеты. Таким образом, получаются вложенные (гнездовые) структуры (рис. 7.7).

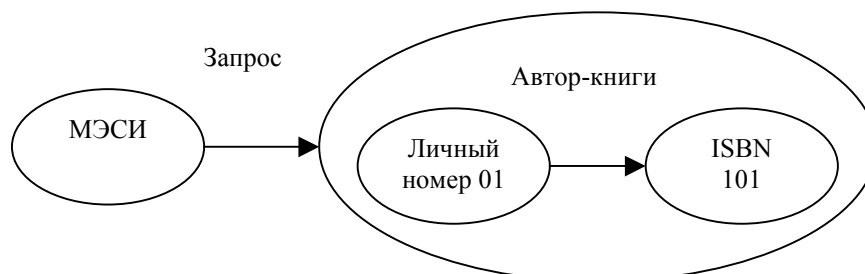


Рис.7.7. Гнездовая структура

Синтаксис представления триплетов одинаков и для определения классов, и для определения отношений (свойств): первый элемент – имя предиката, второй элемент – идентификация информационного ресурса (объекта), третий элемент – характеристика ресурса (значения свойства или отношения). Объекты RDF можно определить как экземпляры одного или нескольких классов (Class) с помощью свойства type. Характеристика подкласса (subClassOf) позволяет задавать иерархию обобщения для классов, а характеристика подсвойства (subPropertyOf) – иерархию обобщения для свойств. Ограничения на свойства задаются с помощью конструкций домена (Domain) и диапазона (Range). Пример описания онтологии на RDF имеет следующий вид [19]:

```

<rdf:Description ID="MotorVehicle">
  <rdf:type resource="http://www.w3.org/...#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/...#Resource"/>
</rdf:Description>
<rdf:Description ID="Truck">
  <rdf:type resource="http://www.w3.org/...#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>

```

```

</rdf:Description>
<rdf:Description ID="registeredTo">
  <rdf:type resource="http://www.w3.org/...#Property"/>
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Description>
<rdf:Description ID="ownedBy">
  <rdf:type resource="http://www.w3.org/...#Property"/>
  <rdfs:subPropertyOf rdf:resource="#registeredTo"/>
</rdf:Description>

```

Достоинства RDF заключаются в стандартизации представления парадигматических отношений по сравнению с XML. Вместе с тем, язык не приспособлен для отражения синтагматических отношений, что обедняет применение онтологии из-за неспособности делать логические выводы.

Язык OIL+DAML – фреймовый язык для Web-приложений

Языки OIL (Ontology Interchange Language) и DAML (DARPA Agent Markup Language) разрабатывались независимо друг от друга, соответственно в Европе и США, как расширение языка RDF. Поскольку в основу языков были положены одни и те же принципы, в последствии стало возможным их объединение. В качестве принципов разработки языка используются следующие:

- Синтаксис XML/RDF.
- Семантика и логический вывод дескриптивной логики.
- Примитивы моделирования знаний – ограниченные фреймы ОКВС.

Фреймовое представление онтологического знания в DAML+OIL обеспечивает следующие преимущества:

- Интуитивно понятное представление знаний для большинства пользователей.
- Широкий набор примитивов моделирования.
- Использование стандартов фреймовых систем ОКВС, XOL.
- Поддержка логического вывода: выявление противоречивости, определение скрытых суперклассов.
- Определение непересекаемости классов.

- Классы могут быть примитивными с условием необходимости (импликации), например, `elephant => animal that has-colour grey` и определяемыми с условием необходимости и достаточности (эквивалентности), например, `vegetarian < = > person who eats meat nor fish`.
- Слабые и сильные ограничения слотов, соответственно определения «может иметь» (`has value`) и «имеет» (`type value`).
- Произвольные выражения классов вместо имен (использование дескриптивной логики), например, `slot-constraint eats value-type NOT (OR meat fish)`
- Ограничения размерности, например, `slot-constraint eats max-cardinality 1 plant`.
- Задание подслотов, например, `daughter-of sub-slot of child-of`.
- Задание свойств для слотов фреймов или отношений, например, транзитивность (для отношения «целое-часть»), симметричность (для отношения соединения компонентов) и др.

Ограниченность фреймового представления связана со следующими особенностями:

- Не поддерживается механизм передачи значений слотов по умолчанию.
- Ограниченно представляются аксиомы, в основном через механизм ограничений слотов и определений подклассов.

К основным конструктивным элементам построения онтологий в языке OIL+DAML относятся следующие элементы:

- `class-def` – определение класса,
- `subclass-of` – определение подкласса,
- `slot-def` – определение слота,
- `subslot-of` – определение подслота,
- `domain` – определение домена,
- `range` – определение диапазона,
- `class-expressions` – выражения класса,
- `AND`, `OR`, `NOT` – логические операторы в выражения класса,
- `slot-constraints` – ограничения слота,
- `has-value`, `value-type` – имеет значение, тип значения в ограничении,
- `cardinality` – кардинальность (число значений),
- `slot-properties` – свойства слота,

- trans, symm – транзитивность, симметричность.

Причем первые 6 конструктивных элементов наследуются от языка RDF, а остальные конструктивные элементы являются оригинальными.

Пример концептуального представления онтологии в языке DAML+OIL [19] представлен на рис.7.8 .

```

class-def animal           % животные есть класс
class-def plant           % растения есть класс
  subclass-of NOT animal   % которые не есть животные
class-def tree
  subclass-of plant        % деревья есть тип растений
class-def branch
  slot-constraint is-part-of % ветви есть часть некоторых деревьев
    has-value tree
    max-cardinality 1
class-def defined carnivore % плотоядное животное есть животное
  subclass-of animal
    slot-constraint eats    % которое ест некоторых других животных
    value-type animal
class-def defined herbivore % травоядное животное есть животное
  subclass-of animal,
  NOT carnivore             % которое не есть плотоядное животное
  slot-constraint eats    % которое ест растения или их части
    value-type plant OR
    (slot-constraint is-part-of
      has-value plant)

```

Рис. 7.8. Концептуальное представление онтологии в языке DAML+OIL

Соответствующее описание фрагмента онтологии в синтаксисе языка OIL представлено на рис.7.9.

```

<rdfs:Class rdf:ID="herbivore">
  <rdf:type
rdf:resource="http://www.ontoknowledge.org/#DefinedClass"/>
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    <oil:NOT>
      <oil:hasOperand rdf:resource="#carnivore"/>
    </oil:NOT>
  </rdfs:subClassOf>
</rdfs:Class>

```

Рис. 7.9. Фрагмент онтологии в синтаксисе языка OIL

7.5. Реализации систем управления знаниями в инструментальной среде

Выбор инструментальных средств реализации СУЗ во многом определяется требуемой функциональностью использования СУЗ: информационный поиск во внешних и/или внутренних источниках знаний, коллективное решение задач, обучение, обоснование решений и т.д.

Для узкоспециализированных целей, ориентированных на поиск в ИНТЕРНЕТ ресурсах, используются специализированные системы. Например, система SHOE [29] обеспечивает аннотацию документов (Knowledge Annotator), сбор знаний в централизованную базу знаний (Expose), выполнение поисковых запросов (SHOE Search). Аналогичный проект (KA)² [1,27] включает большой набор инструментов: редактор онтологий (OntoEdit), извлечение знаний из текстов (Text-To-Onto), средство гиперболической визуализации (Hiperbolic), средство поиска (OntoBroker) и др. Полный набор инструментальных возможностей OntoEdit представлен в таблице 7.2. Специализированная корпоративная информационно-поисковая система Canberra RetrievalWare [8,10,11] помимо поддержания словарей (тезауруса) и реализации запросов к внутренним информационным ресурсам позволяет отслеживать текстовую информацию на заданных узлах сети Internet, выполнять автоматическую рубрикацию поступающих на запросы документы и

т.д. В условиях такого разнообразия становится очевидным, что выбор конкретного набора инструментов будет определяться наилучшим соотношением потребительная ценность/затраты в соответствующем классе программных средств и открытостью для возможных доработок.

Таблица 7.2.

Инструментальные возможности OntoEdit

Характеристика	Значение
Номер версии	2.5, 2002-05 3.0, 2002-08 (Collaborative Engineering, Ontology Server)
Архитектура	Локальная версия (2.0, 2.5, 3.0) Клиент-сервер (2.5, 3.0)
Расширяемость	Plug-ins
Хранение онтологий	Файлы (2.0, 2.5, 3.0) Ontology Server, Relational Databases (3.0)
Интероперабельность с другими инструментами	OntoAnnotate OntoBroker OntoMat Semantic Miner (Поиск на основе онтологий)
Трансляция в языки	DAML+OIL Flogic RDF(S) OXML SQL-3
Экземпляры классов	Да
Экземпляры отношений	Да
Метаклассы	Нет
Атрибуты классов	Да
Атрибуты экземпляров	Да
Локальные атрибуты	Да
Глобальные атрибуты	Да
Кардинальность слота	Да

Тип значения слота	Нет
Значение слота по умолчанию	Нет
Создание новых фасетов	Нет
Максимальная арность отношений	2
Таксономия	Подкласс
Исключающая декомпозиция	Да
Покрывающая декомпозиция	Нет
Множественное наследование	Да
Аксиомы	Да
Продукционные правила	Да
Присоединение механизма вывода	Да
Автоматическая классификация	Да
Обработка исключений	Нет
Выполнение процедур	Нет
Наследование	Множественное
Ограничения/Целостность	Уровень проверки
Проверка и верификация	Да
Управление конфигураций	Да
Документация	Да
Интеграция онтологий	Да
Слияние	Да
Совместная работа	Да
Контроль изменений пользователя	Нет
Уровень защиты	Защита объектов
Управление групповой работой	Нет
Поддержка библиотеки онтологий	Да

Независимо от конкретной функциональной направленности программных инструментов реализации СУЗ включаемые инструментальные средства должны обеспечивать две основные группы функций:

- Создание и поддержание источников знаний.
- Доступ к источникам знаний.

Создание и поддержание источников знаний

Инструментальные средства создания и поддержания источников знаний, как правило, имеют следующие функциональные возможности:

- Создание и поддержание онтологий.
- Аннотирование источников знаний.
- Подключение источников знаний.
- Автоматическая рубрикация и индексирование источников знаний.

Создание онтологий обеспечивается средствами онтологических редакторов, которые должны реализовывать ввод категорий, отношений на категориях, ограничений (аксиом) в графической визуальной форме, освобождающей разработчика от знания внутреннего синтаксиса представления онтологических конструкций. Кроме того, редакторы обеспечивают контроль возникающих ошибок. Пример использования редактора OntoEdit представлен на рис. 7.10 – 7.11., а OilEd на рис. 7.12.

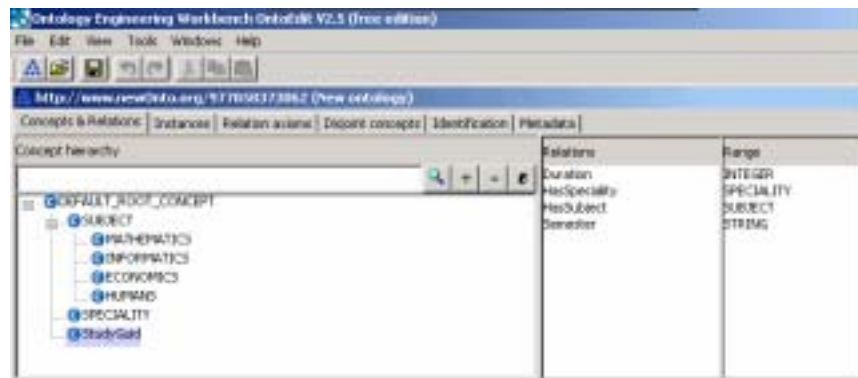


Рис.7.10. Определение категорий (концептов) и отношений в редакторе OntoEdit

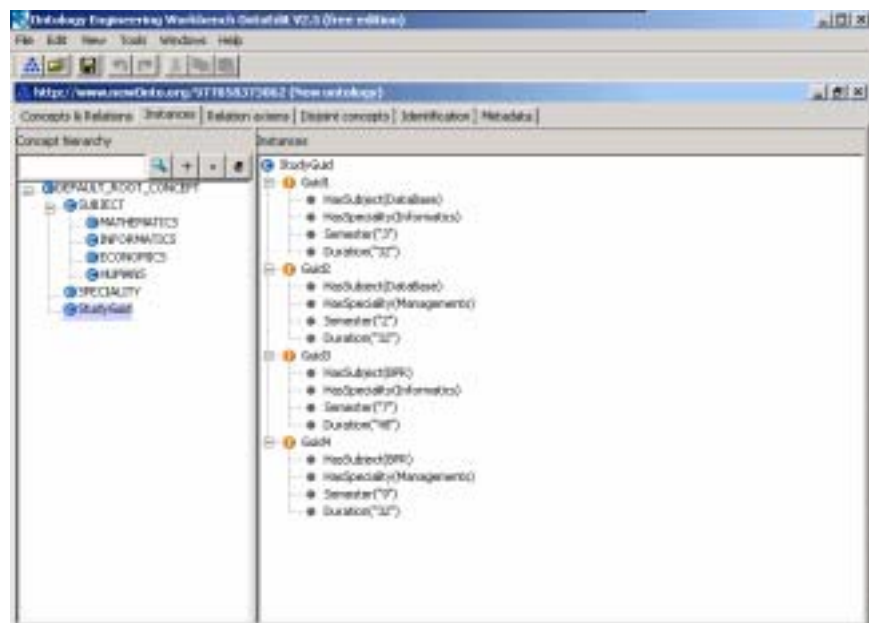


Рис. 7.11. Определение примеров (экземпляров) в редакторе OntoEdit

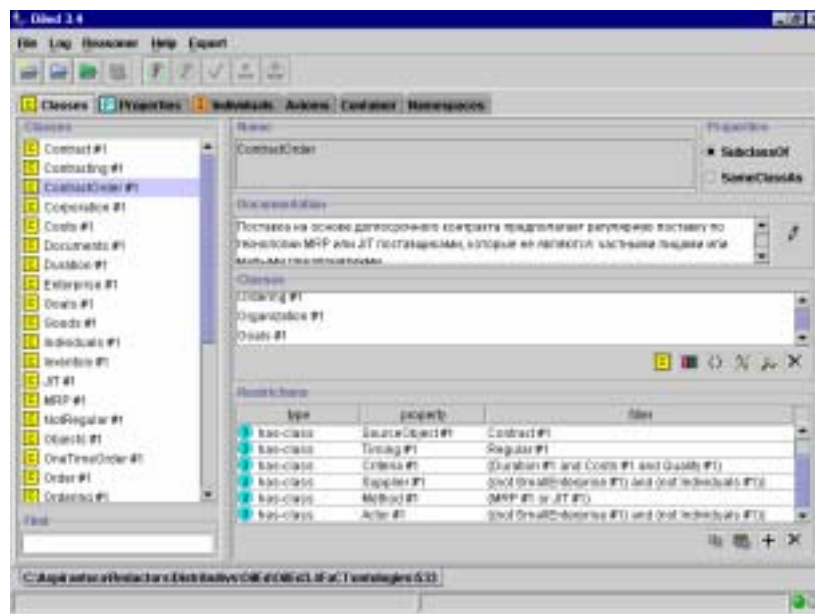


Рис. 7.12. Определение классов в редакторе OntoEdit

При интеграции нескольких источников знаний полезно иметь средства трансляции онтологий из одного представления в другое, осуществлять объединение онтологий в одной системе управления знаниями. В качестве примера можно привести систему Ontolingua, которая обладает средствами трансляции в различные представления. В проекте (KA)² имеется средство объединения/отображения онтологий (Merge/Map), а также их трансляции в DTD (document type definition) XML.

Ручная аннотация источников знаний

Сущность ручной аннотации сводится в задании для источника знаний категорий онтологии, к которым он относится и по которым в дальнейшем произойдет индексирование документа. Для упрощения этой работы могут использоваться специальные программы, которые позволяют пользователю просматривать и выбирать из онтологий необходимые категории для аннотации, например, программа Knowledge Annotator системы SHOE или

программа OntoAnnotate, сопряженная с инструментальным средством OntoEdit.

Подключение источников знаний

Подключение источников знаний к общей системе осуществляется путем их индексирования. Например, для включения аннотированного текста в СУЗ, поддерживаемую SHOE, необходимо внести заявку на размещение текста на сайте исследовательской группы PLUS, для чего в заявке указывается URL текста. После этого программа Expose собирает знания со всех www-страниц и помещает их в БЗ Parka, в которой также хранится информация всех существующих онтологий. Таким образом, знания распространяемого источника становятся доступными для специально разработанной в Java SHOE Search – поисковой машины.

Автоматическая рубрикация и индексирование источников знаний

Описанные выше ручная аннотация и индексирование источников знаний допустимы, когда некоторая организация выставляет свой источник знаний на всеобщее обозрение во внешнюю среду. В процессе эксплуатации СУЗ в систему поступают большие объемы документов различной природы, например, при полнотекстовом отборе в Интернете или по электронной почте, которые вручную практически не возможно проаннотировать и проиндексировать. В этом случае используются программы автоматической рубрикации и индексирования.

Например, сервер рубрикации системы RetrievalWare в соответствии с ранее введенными запросами по мере поступления документов распределяет их по тематическим рубрикам (категориям) (по запросам определяется частота используемых терминов, которая определяет степень их важности для рубрикации). При этом один документ может входить в несколько рубрик. Система позволяет подключать и внешние средства рубрикации и кластеризации поступающих документов.

Для автоматического индексирования документов используется программный модуль Internet Spider системы RetrievalWare, который отслеживает текстовую информацию на заданных узлах сети Internet, сканируя неограниченное количество

страниц Интернет. Полученная таким образом информация автоматически индексируется RetrievalWare. При конфигурировании Internet Spider определяются такие параметры, как тип извлекаемых документов, имя хоста, каталог, домен, глубина, широта и частота сканирования. Internet Spider способен работать с любыми объемами данных – от корпоративной интрасети до внешних узлов Интернет. Критичные по времени процессы выполняются параллельно на нескольких процессорах (на многопроцессорных машинах). Возможна работа в многосерверной конфигурации. Имеются средства извлечения знаний из текстов, например, инструментальное средство Text-to-Onto проекта КА² .

Доступ к источникам знаний

Инструментальные средства доступа к источникам знаний СУЗ должны обеспечивать следующие функциональные возможности:

- Реализация запросов
- Навигация и просмотр
- Коммуникация пользователей
- Распространение знаний

Реализация запросов

Существующие поисковые системы обеспечивают только полнотекстовый поиск, например, в системах Altavista (WWW), Verity (Intranet). В результате возникают проблемы нерелевантности результатов поиска.

СУЗ, основанные на применении онтологии знаний, улучшают релевантность поиска по запросу за счет:

- Семантического контроля.
- Анализа контекста.
- Механизма расширения содержания.

Инструментальные средства реализации запросов к источникам знаний обычно включает интерфейс формулирования запросов, механизм вывода, машину доступа к информационным ресурсам, обычно поддерживающую многоагентную технологию. К таким инструментальным средствам относятся, например, Ontobroker (проект КА²), SHOE Search.

В системе SHOE составление запроса сводится к заполнению экранной формы путем последовательного определения из списков используемых онтологий, категорий онтологий, отношений (атрибутов) и ограничений на значения отношений (атрибутов), которые могут быть связаны достаточно сложными логическими выражениями. Пример заполнения экранной формы в системе SHOE представлен на рис. 7.13.

В процессе реализации запроса по ограничениям атрибутов и аксиомам, зафиксированным в онтологиях, осуществляется проверка корректности запроса и его расширение семантически значимыми отношениями. Например, к используемым категориям добавляются синонимичные обозначения, транзитивно удаленные термины.

Полученный результат поиска в некоторых системах может упорядочиваться по степени релевантности запросу. Так, в системе Retrieval Ware оценка релевантности отбираемых источников знаний запросам осуществляется на основе вычисления степени близости расширяющих терминов исходному запросу. Например, на запрос «прокат автомобилей» будет выдаваться информация, связанная с деятельностью предприятий бытового обслуживания, и не будет выдаваться информация о предприятиях металлургии или автомобилестроения, хотя слово прокат также относится к металлургии, а при производстве автомобиля может использоваться технология прокатки стального листа. Таким образом, система вычислит очень высокую степень близости между понятиями «прокат автомобилей» и «бытовое обслуживание» и недопустимо большое семантическое расстояние между понятиями «прокат автомобилей» и «металлургия» или «автомобилестроение». Вычисление семантического расстояния между понятиями выполняется на основе весовых коэффициентов, приписываемых по шкале (0,1) отношениям между категориями в семантической сети тезауруса.

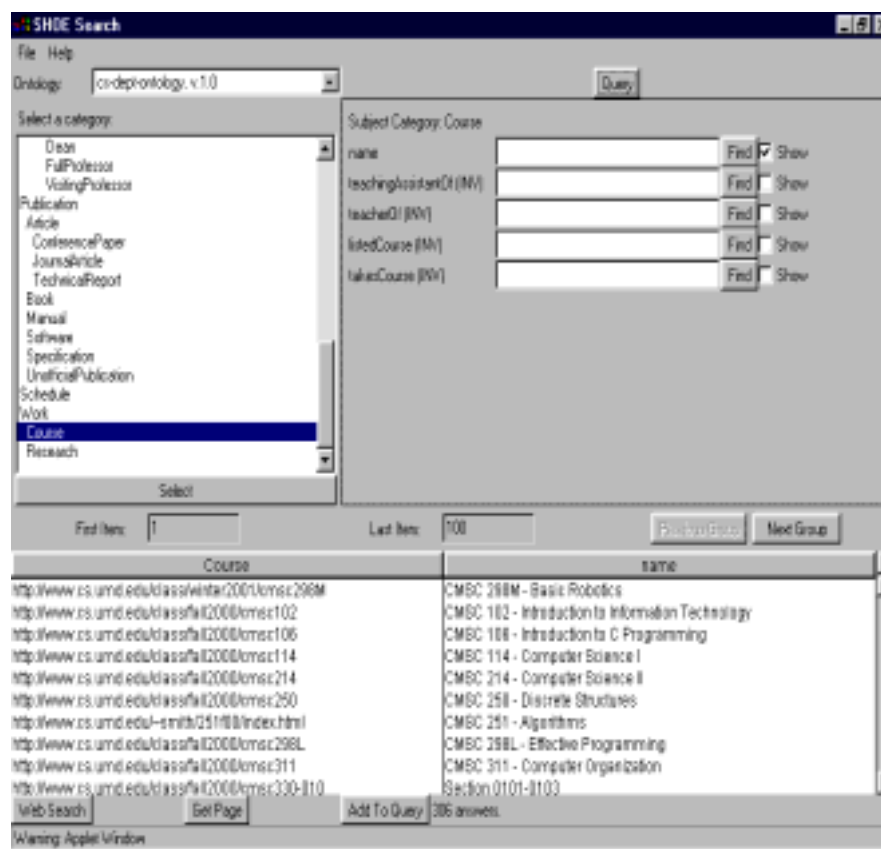


Рис. 7.13. Пример заполнения экранной формы в системе SHOE

В системе Retrieval Ware существует возможность поиска документов по образцу (query by example) – при этом система сама выбирает из документа наиболее статистически и семантически значимые слова и формирует из них сложный логический запрос, учитывающий и структуру, и смысловое содержание документа.

Навигация и просмотр

Существующие системы, поддерживающие информационные порталы (например, Yahoo) обладают следующими недостатками средств навигации пользователя:

- Жесткая система категорий (рубрик), не позволяющая пользователю самостоятельно формировать маршрут просмотра источников знаний на основе собственной рубрикации.
- Отсутствие или слабое справочное руководство по категориям, которое раскрывало бы понятийную сущность рубрик.
- Отсутствие визуализации иерархии рубрик, которое давало бы целостное представление о структуризации источников знаний.

С помощью онтологии обеспечиваются следующие возможности навигации:

- Визуализация иерархий рубрик навигации.
- Многоаспектная навигация в источниках знаний.
- Гипертекстовый доступ к глоссарию, раскрывающему смысловое содержание категорий.
- Динамическое формирование маршрута навигации в зависимости от уровня знаний и потребностей пользователя.

Для реализации перечисленных возможностей используется либо общепринятый интерфейс с выбором из меню, либо специально разработанный интерфейс с применением средств визуализации и навигации на гиперболическом представлении онтологии (Hyperbolic Geometry). На рис. 7.14 показано гиперболическое представление онтологии, в котором узлы онтологии представляются шарами, а отношения – соединяющими линиями.

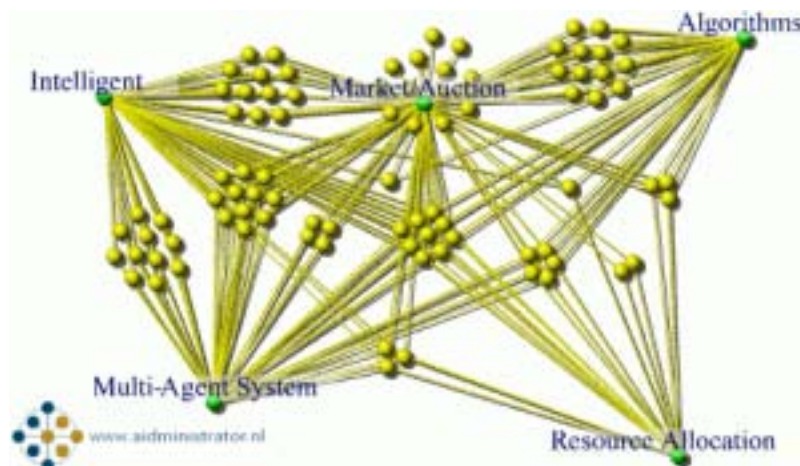


Рис.7.14. Гиперболическое представление онтологии

Взаимодействие пользователей СУЗ (коммуникация и распространение знаний)

Существующие системы поддержки дискуссионных форумов (Knowlity), средств группового взаимодействия (Lotus Notes) отличаются отсутствием или слабой персонализацией участников коллективного обмена знаниями. Простейшие доски объявлений, которыми, по сути, являются дискуссионные форумы, приводят к случайности формирования участников обсуждений на основе только названия темы и не гарантирует достижения цели.

В противовес применяемым средствам группового взаимодействия СУЗ, основанные на онтологиях, обеспечивают следующие возможности:

- Целенаправленный выбор партнера по решению проблемы на основе более точной ее формулировки в категориях онтологии.
- Онтологическое профилирование потенциальных участников группового взаимодействия, когда профили знаний участников описываются в категориях онтологии и подключаются к сети, как и документальные ресурсы.

Реализация потребностей пользователя в консультации эксперта может быть решена так же, как и поисковая задача с использованием многоагентной технологии, осуществляемой,

например, в таких поисковых системах, как MARPI, OntoSeek, (ONTO)² и др. [1].

Отличительной особенностью применения многоагентной технологии для коллективного решения задач в СУЗ является необходимость планирования использования ресурсов во времени с учетом требований экономии стоимостных затрат.

Многоагентная система организует взаимодействие агентов для совместного решения определенной задачи. В качестве агентов выступают активные программные объекты в нотации объектно-ориентированного подхода, которые помимо коммуникации между собой на основе собственного знания способны генерировать различные варианты решений и выбирать из них наиболее оптимальные по заданному критерию, а также при необходимости модифицировать собственные знания о правилах принятия решений.

Так, для решения задачи поиска эксперта для консультации (рис. 7.15) агент процесса обрабатывает требования пользователя, в соответствии с требованием находит и посылает запросы агентам-ресурсам. Агенты ресурсов в свою очередь проверяют возможность выполнить запрос в отведенные сроки и посылают свои условия агенту процесса. Агент процесса принимает решение и составляет расписание (график) выполнения работ агентами ресурсов, которые в соответствии с расписанием предоставляют свои ресурсы пользователю.

При распространении знаний СУЗ агент процесса периодически анализирует поведение (требования) пользователей и синтезирует за пользователя запросы на удовлетворение его потенциальных потребностей. В этом случае агент процесса должен обладать способностью самообучения на основе интеллектуального анализа запросов.

На основе многоагентной технологии решаются не только задачи консультирования, но и задачи коллективной разработки проектов, планирования и управления нерегламентированными работами (workflow, электронные торги и др.). В этом случае роли агентов ресурсов должны быть уточнены в соответствии с характером выполняемой работы, например, по специализации (эксперт, инженер, менеджер, администратор и др.) и квалификации (руководитель проекта, руководитель группы, исполнитель, ученик и др.).

При разработке проекта члены проектной группы имеют своих программных агентов, которые сначала договариваются между собой о создании эффективной группы на основе общей цели и достижения компромисса частных целей, а затем после переговоров о распределении ролей взаимодействуют. Последовательность взаимодействия агентов генерируется на основе онтологии задач, а понимание агентами сущности решаемой задачи происходит на основе предметной онтологии. В качестве онтологий задач используются онтологии проведения переговоров и взаимодействия участников проекта.

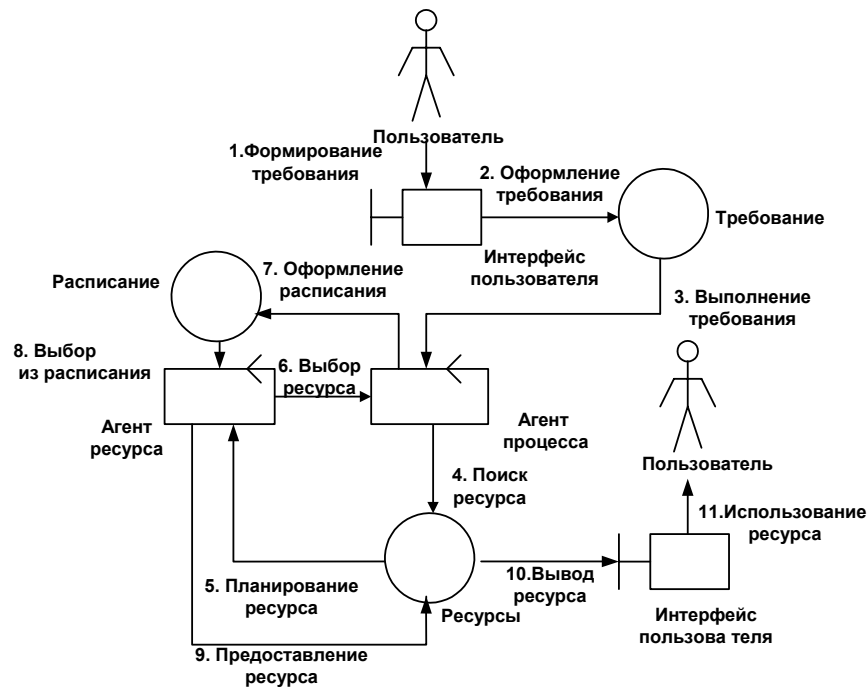


Рис. 7.15. Схема взаимодействия агентов

Что следует запомнить

Проектирование СУЗ направлено на создание корпоративной системы, требующей решения организационных вопросов взаимодействия сотрудников предприятия или реинжиниринга.

Стадия идентификации определяет требования к источникам знаний разрабатываемой СУЗ, набор решаемых задач, категории пользователей. В отличие от экспертных систем СУЗ отличаются способностью к расширению источников знаний и использованию разнообразных инструментальных средств.

Система управления знаниями интегрирует множество разнородных источников знаний, вследствие чего требуется проектирование единого семантического описания пространства знаний, в качестве которого выступает онтология.

Онтология – это точное (явное) описание концептуализации знаний в виде множества используемых понятий (концептов) и набора парадигматических и синтагматических отношений.

Парадигматические отношения – отношения синонимии, омонимии, полисемии, обобщения «род-вид», агрегации «целое-часть», причинно-следственные и другие отношения, которые превращают словарь понятий (концептов, категорий) в тезаурус.

Синтагматические отношения отражают семантические ограничения в виде аксиом или логических правил, которые позволяют правильно строить переменные отношения понятий в конкретном контексте решения задачи.

Метаонтология оперирует общими концептами и отношениями, которые не зависят от конкретной предметной области. Концептами метаязыка являются такие общие понятия, как: «объект», «свойство», «значение», «событие» и т.д.

Предметная онтология отражает понятия, описывающие конкретную предметную область, отношения, семантически значимые для данной предметной области, и множество интерпретаций этих понятий и отношений.

Онтология задач в качестве понятий содержит типы решаемых задач, а отношения этой онтологии, как правило, специфицируют декомпозицию задач на подзадачи.

Формализация семантической структуры знаний требует применения языков исчисления предикатов для отображения аксиоматики и объектно-ориентированного представления знаний

для отображения таксономии используемых понятий. Наиболее пригодными языками с этой точки зрения являются KIF и семантические сети и фреймовые языки, удовлетворяющие требованиям ОКВС.

Коллективное использование СУЗ в INTRANET и INTERNET среде требует расширения языков разметки HTML и XML в части отображения онтологического знания.

Основное назначение языков представления знаний на базе RDF, DAML+OIL, OXML, SHOE, расширяющих языки разметки, заключается в возможности описания онтологии, аннотировании необходимых источников знаний концептами онтологии, по которым строятся индексы поиска знаний, подключающие источники знаний к СУЗ.

Инструментальные программные средства СУЗ должны включать средства онтологического инжиниринга (редакторы онтологий), ручного аннотирования источников знаний, автоматизированной рубрикации источников знаний, реализации поисковых запросов, навигации в пространстве знаний, установления интерактивного взаимодействия участников обмена знаниями, автоматизированного извлечения знаний из различных источников с помощью специальных инструментов.

Универсальный характер инструментальных программных средств СУЗ предполагает, что основной акцент в проектировании систем управления знаниями должен делаться на семантическое представление знаний и обмен знаниями между различными машинными агентами СУЗ. Пренебрежение этим принципом превращает СУЗ в набор плохо связанных между собой информационных технологий.

Литература

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. М.: Питер, 2000. – 382 с.
2. Зайцева Ж. Н., Рубин Ю. Б., Титарев Л. Г., Тихомиров В. П., Хорошилов А. В., Усков В. Л. Филиппов В. М. Открытое образование – стратегия XXI века для России / Под общей редакцией Филиппова В. М. и Тихомирова В. П. - М.: Издательство МЭСИ, 2000. – 324с.
3. Зырянов М. Первый этап «Осеннего марафона» компании «Весть-Метатехнология». От поиска информации — к управлению знаниями. Еженедельник "Computerworld Россия", #46/1999. Издательство "Открытые системы".
<http://www.jobworld.ru/cw/1999/>
4. Декер С., Мельник С., ван Хермелен Ф., Фенсел Д., Клейн М., Брукстра Д., Эрдманн М, Хоррокс Я. Semantic Web: роли XML и RDF. Открытые системы, #09/2001
5. Ин Ц., Соломон Д. Использование Турбо-Пролога. М.: Мир, 1993. – 606 с.
6. Каменнова М., Громов А., Ферапонтов М., Шматалюк А. Моделирование бизнеса. Методология ARIS. Практическое руководство/ Под ред. М.С. Каменновой.: М.: Изд-во «Серебряные нити», 2001. – 327с/
7. Мизогучи Р. Шаг в направлении инженерии онтологий. Новости искусственного интеллекта. М.: РАИИ, 2000. №1-2, с. 11 – 36.
8. Попов Э.В. Корпоративные системы управления знаниями. // Новости искусственного интеллекта. 2001, №1. –с. 14-25.
9. Представление и использование знаний / Пер. с япон.; Под ред. Х.Уэно, М.Исидзука. - М.: Мир, 1989. - 220 с.
10. Промышленная информационно-поисковая система Excalibur RetrievalWare (Convera Technologies Corp.)
www.vest.msk.ru.
11. Русский семантический сервер. www.vest.msk.ru.
12. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем:

- Учебник / Под ред. Ю.Ф. Тельнова. – М.: Финансы и статистика, 2001. – 512 с. : ил.
13. Тельнов Ю.Ф., Скорова А.А., Андреева Н.В. Проектирование баз знаний. Учебное пособие.- М.: МЭСИ, 1992.-100с.
 14. Тельнов Ю.Ф., Смирнова Г.Н., Диго С.М., Шишков М.Ю. Модели описания предметных областей баз данных. М.: МЭСИ, 1985. – 106 с.
 15. Цикритзис Д., Лоховски Ф. Модели данных. /Пер. с англ. - М.: Финансы и статистика, 1985. -344с.
 16. Шеер А-В. Моделирование бизнес-процессов. Издание 2-е. Пер с англ. / Под науч.ред. М.С. Каменновой. М.: Изд-во «Серебрянные нити», 2000. – 205с.
 17. Питц-Маултис Н., Кирк Ч. XML: Пер с англ. – СПб.: БХВ-Петербург, 2001. – 736с.:ил.
 18. Abecker A., Bernardi A, Hinkelman K, Kuhn O. Toward a Technology for Organizational Memories. // IEEE Intelligent Systems 1998. No. 3. p 40 – 48.
 19. Akkermans H. (VUA), Schnurr H.-P., Studer R., Sure Y. (AIFB), On-To-Knowledge review. Juan-Les-Pins/France, October 06, 2000
 20. Chaudhri V. K., Farquhar A., Fikes R, Karp P. D., Rice J. P. Open Knowledge Base Connectivity 2.0.2. February 3, 1998.
 21. Gruber T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing / /International Journal of Human and Computer Studies. – 1993, No. 43(5/6). p 907 – 928.
 22. Gruber T. R. A Translation Approach to Portable Ontologies // Knowledge Acquisition – 1993, No. 5(2). p 199 – 220.
 23. Knowledge Interchange Format draft proposed American National Standard (dpANS) NCITS.T2/98-004.
 24. O’Leary D.E. Using AI in Knowledge Management: Knowledge Bases and Ontologies. // IEEE Intelligent Systems. 1998, No 3. p.34-39.
 25. O’Leary D. E. «Enterprise Knowledge Management», – IEEE Computer, 3, March 1998, pp. 54-61.
 26. A. Lopatenko's Resource Guide to Metadata for Science, Research, Education and Technology. (OIL)
http://e-science.narod.ru/Metadata_Science.htm

27. Maedche A., Motik B., Volz R. KAON — A Framework for Semantics-based E-Services. Forschungszentrum Informatik FZI, 76131 (<http://www.fzi.de/wim>). Karlsruhe, Institute AIFB, University of Karlsruhe, 76128 Karlsruhe (<http://www.aifb.uni-karlsruhe.de/WBS>).
28. Motik B., Maedche A. and Volz R. A Conceptual Modeling Approach for Semantics-Driven Enterprise Applications. FZI Research Center for Information Technologies at the University of Karlsruhe, D-76131 Karlsruhe, Germany.
29. Searching the Web with SHOE, by Jeff Heflin and James Hendler. In AAAI-2000 Workshop on AI for Web Search. 2000.