МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ СТАВРОПОЛЬСКОГО КРАЯ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ

В.С. Тоискин

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УЧЕБНОЕ ПОСОБИЕ

Часть 1

Ставрополь, 2009

УДК 681.518(075.8)
ББК 32.813я73
Т 50

Печатается по решению
редакционно-издательского совета
ГОУ ВПО Ставропольский государственный
педагогический институт

Рецензенты:
доктор технических наук, профессор, заслуженный деятель науки РФ
В.А. Цымбал (Серпуховской военный институт)
кандидат физико-математических наук, доцент ***Е.Н. Косова***

Тоискин В.С.
Т 50 **Интеллектуальные информационные системы**: Учебное пособие. Часть 1. – Ставрополь: Изд-во СГПИ, 2009. – 181 с.

В пособии рассматриваются исторические этапы, методологические основы теории интеллектуальных информационных систем, их классификация, основные свойства и области применения. Основное внимание уделяется общим вопросам построения экспертных систем, их архитектуре, режимам работы, этапам разработки. Обсуждаются стратегии извлечения знаний, модели выявления знаний от экспертов и вопросы обработки экспертных оценок. Излагаются основы теории байесовских сетей вывода. Теоретические положения иллюстрируются примерами. Рассматривается технология работы с экспертной оболочкой Hugin.

Материал учебного пособия соответствует требованиям Государственного образовательного стандарта высшего профессионального образования к одноименной дисциплине специальности «Прикладная информатика (в экономике)».

Пособие предназначено для студентов, обучающихся по специальности «Прикладная информатика (в экономике)», и может быть использовано аспирантами и в системе повышения квалификации.

УДК 681.518(075.8)
ББК 32.813я73

© Тоискин В.С., 2009
© Ставропольский государственный
педагогический институт, 2009

Введение

Эволюция информационных технологий и систем все в большей степени определяется их интеллектуализацией. Интеллектуальные информационные системы и технологии – одна из наиболее перспективных и быстро развивающихся научных и прикладных областей информатики. Она оказывает существенное влияние на все научные и технологические направления, связанные с использованием компьютеров, и уже сегодня дает обществу то, что оно ждет от науки – практически значимые результаты, многие из которых способствуют кардинальным изменениям в сферах их применения.

Целями интеллектуальных информационных технологий являются, во-первых, расширение круга задач, решаемых с помощью компьютеров, особенно в слабоструктурированных предметных областях, и, во-вторых, повышение уровня интеллектуальной информационной поддержки современного специалиста.

Ключевым компонентом научного фундамента интеллектуальных информационных систем является искусственный интеллект (ИИ). Для создания и развития ИИ как научного направления за рубежом много сделали Н. Винер, У. Маккаллох, У. Питтс, Д. Маккарти (который впервые ввел термин «artificial intelligence»), Ф. Розенблат, А. Сазерленд, М. Минский, С. Пейперт, А. Ньюэлл, Г. Саймон, Дж. Шоу, Э. Фейгенбаум, А. Кольмероэ, Н. Хомский, Т. Виноград, М. Куиллиан, Р. Шенк, Н. Нильсон, П. Уинстон, Л. Заде, Р. Редди, Д. Ленат, Дж. Хинтон, Дж. Андерсон, Ж.-Л. Лорьер и многие другие. В СССР, а затем в России со становлением и развитием ИИ связывают имена А.А. Ляпунова, А.И. Берга, Г.С. Поспелова, М.Л. Цетлина, М.М. Бонгарда, М.А. Гаврилова, А.П. Ершова, В.Н. Пушкина, Л.Т. Кузина, А.С. Нариньяни, А.И. Половинкина, В.В. Чавчанидзе, В.К. Финна, Э.В. Попова, Э.Х. Тыугу, Н.Н. Непейводы, И.П. Кузнецова, О.И. Ларичева, А.И. Галушкина, А.Н. Горбаня, А.В. Чечкина и многих других. Следует отметить исключительную роль Д.А. Поспелова и его научной школы: В.Н. Вагина, Т.А. Гавриловой, А.П. Еремеева, Г.С. Осипова, В.Ф. Хорошевского и др.

Интеллектуальные информационные системы или системы, основанные на знаниях – это один из видов автоматизированных информационных систем. Интеллектуальные информационные системы

представляет собой комплекс программных, лингвистических и логико-математических средств для реализации основной задачи: осуществление поддержки деятельности человека и поиска информации в режиме продвинутого диалога на естественном языке.

Перечень задач, решаемых интеллектуальными информационными системами весьма впечатляющ: интерпретация данных, диагностика, мониторинг, проектирование, прогнозирование, планирование, обучение, управление, поддержка принятия решений и др.

Интеллект как атрибут информационной системы определяется формированием «образа» (изменением структуры внутренних связей в базе знания), влияющего на реакцию на внешние воздействия. Он проявляется только в актах общения со столь же сложными объектами и активизируется в системе в процессе реорганизации внутренних информационных связей.

Основной цикл управления интеллектуальной системы основан на работе со знанием. Классический основной цикл управления не может быть распространен на системы, требующие интеллектуального управления, потому что управление через «образ», во-первых, требует существенного учета конкретного накопленного знания, формально распределенного между руководителем и системой, а, во-вторых, более критично к изменению информации в процессе принятия решения. Соответственно, интеллектуальные информационные системы, опирающиеся на принятие решений с использованием знания, имеет существенные, принципиальные отличия от традиционных информационных систем.

Дисциплина «Интеллектуальные информационные системы» дает представление студентам о состоянии разработки и тенденциях развития информационных систем, их различных приложениях. Студенты смогут сориентироваться, какие именно модели и методы современных интеллектуальных информационных систем могут быть использованы при решении тех или иных задач в области профессиональной деятельности. В частности рассматриваемые системы могут быть использованы при решении тех или иных задач экономического анализа и принятия решений.

Цель данного пособия – дать студентам комплекс ориентирующих знаний по основным понятиям интеллектуальных информационных систем и возможностям их использования в различных областях применения в соответствии с требованиями Государственного образовательного стандарта по специальности «Прикладная информатика (в экономике)».

1. Введение в интеллектуальные информационные системы

1.1. Методологические основы теории искусственного интеллекта

Термин «*искусственный интеллект*» относится к группе терминов и понятий, получивших весьма широкое распространение, в том числе и у не специалистов.

Для того чтобы методически верно подойти к определению основных понятий теории искусственного интеллекта, сначала кратко рассмотрим основные этапы его истории.

С конца 40-х годов ученые все большего числа университетских и промышленных исследовательских лабораторий устремились к дерзкой цели: построение компьютеров, действующих таким образом, что по результатам работы их невозможно было бы отличить от человеческого разума.

Исследователи, работающие в области искусственного интеллекта (ИИ), обнаружили, что вступили в схватку с весьма запутанными проблемами, далеко выходящими за пределы традиционной информатики. Оказалось, что, прежде всего, необходимо понять механизмы процесса обучения, природу языка и чувственного восприятия. Выяснилось, что для создания машин, имитирующих работу человеческого мозга, требуется разобраться в том, как действуют миллиарды его взаимосвязанных нейронов. И тогда многие исследователи пришли к выводу, что, пожалуй, самая трудная проблема, стоящая перед современной наукой – познание процессов функционирования человеческого разума, а не просто имитация его работы. Это непосредственно затрагивало фундаментальные теоретические проблемы психологической науки. Даже в настоящее время нет единого взгляда на сам предмет исследований – интеллект.

Не пытаясь сразу четко и полно определить это понятие, заметим, что большинство людей трактуют искусственный интеллект как умение решать сложные задачи; другие рассматривают его как способность к обучению, обобщению и аналогиям; третьи – как возможность взаимодействия с внешним миром путем общения, восприятия и осознания воспринятого. Тем не менее, многие исследователи ИИ склонны принять тест машинного интеллекта, предложенный в начале 50-х годов выдающимся английским математиком и специалистом по вычислительной технике Аланом Тьюрингом.

Компьютер можно считать разумным, – утверждал Тьюринг, – если он способен заставить нас поверить, что мы имеем дело не с машиной, а с человеком.

Идея создания мыслящих машин «человеческого типа», которые, казалось бы, думают, двигаются, слышат, говорят, и вообще ведут себя как живые люди уходит корнями в глубокое прошлое. Еще древние египтяне и римляне испытывали благоговейный ужас перед культовыми статуями, которые жестикулировали и изрекали пророчества (разумеется, не без помощи жрецов). Средневековые летописи полны рассказов об автоматах, способных ходить и двигаться почти также как их хозяева – люди. В средние века и даже позднее ходили слухи о том, что у кого-то из мудрецов есть гомункулы (маленькие искусственные человечки) – настоящие живые, способные чувствовать существа.

В XVIII в., благодаря развитию техники, особенно разработке часовых механизмов, интерес к подобным изобретениям возрос. В 1736 г. французский изобретатель Жак де Вокансон изготовил механического флейтиста в человеческий рост, который исполнял двенадцать мелодий, перебирая пальцами отверстия и дуя в мундштук, как настоящий музыкант. В середине 1750-х годов австриец Фридрих фон Кнаус сконструировал серию машин, которые умели держать перо и могли писать довольно длинные тексты. Другой мастер, Пьер Жак-Дроз из Швейцарии, построил пару изумительных по сложности механических кукол размером с ребенка: мальчика, пишущего письма, и девушку, играющую на клавесине.

Успехи механики XIX в. стимулировали еще более честолюбивые замыслы. Так, в 1830-х годах английский математик Чарльз Бэббидж задумал, правда, так и не завершив, сложный цифровой калькулятор, который он назвал Аналитической машиной. Позднее, в 1914 г., испанец Леонардо Торрес-и-Кеведо изготовил электромеханическое устройство, способное разыгрывать простейшие шахматные эндшпили почти также хорошо, как и человек.

Эти достижения могут быть охарактеризованы как механический подход к созданию искусственного интеллекта.

Только после второй мировой войны появились устройства, подходящие для достижения заветной цели – моделирования разумного поведения. Это были электронные цифровые вычислительные машины. В то время многие ученые пришли к выводу, что наступит тот день, когда автоматические вычислители смогут имитировать не вычислительные

процессы, свойственные человеческому мышлению, в том числе восприятие и обучение, распознавание образов, понимание повседневной речи и письма, принятие решений в неопределенных ситуациях.

К концу 50-х годов сформировалась новая, более или менее самостоятельную ветвь информатики, получившая название «искусственный интеллект». Исследования в области ИИ, первоначально сосредоточенные в нескольких университетских центрах США – Массачусетском технологическом институте, Технологическом институте Карнеги в Питтсбурге, Станфордском университете, ныне ведутся во многих странах.

На протяжении всей своей короткой истории исследователи в области ИИ всегда находились на переднем крае информатики. Многие ныне обычные разработки, в том числе усовершенствованные системы программирования, текстовые редакторы и программы распознавания образов, в значительной мере рассматриваются на работах по ИИ. Теории, новые идеи и разработки ИИ неизменно привлекают внимание тех, кто стремится расширить области применения и возможности компьютеров, сделать их более «дружелюбными», то есть более похожими на разумных помощников и активных советчиков. Несмотря на многообещающие перспективы, ни одну из разработанных до сих пор программ ИИ нельзя назвать «разумной» в обычном понимании этого слова. Все сказанное относится к так называемому электронному периоду создания ИИ.

Попытки построить машины, способные к разумному поведению, относятся к кибернетическому этапу и в значительной мере вдохновлены идеями Норберта Винера, одной из выдающихся личностей в интеллектуальной истории Америки. Помимо математики он обладал широкими познаниями в других областях, включая нейропсихологию, медицину, физику и электронику.

Винер был убежден, что наиболее перспективны научные исследования в так называемых пограничных областях, которые нельзя конкретно отнести к той или иной конкретной дисциплине. Они лежат где-то на стыке наук. Винеру и его сотруднику Джулиану Бигелоу принадлежит разработка принципа «обратной связи», который был успешно применен при разработке нового оружия с радиолокационным наведением. Принцип обратной связи заключается в использовании информации, поступающей из окружающего мира, для изменения поведения машины. В основу разработанных Винером и Бигелоу систем наведения были положены тонкие математические

методы. В дальнейшем Винер на принципе обратной связи разработал теории как машинного, так и человеческого разума. Он доказывал, что именно благодаря обратной связи, все живое приспосабливается к окружающей среде и добивается своих целей. Он писал, что все машины, претендующие на «разумность», должны обладать способность преследовать определенные цели и приспосабливаться, т.е. обучаться. Созданной им науке Винер дает название кибернетика, что в переводе с греческого означает «рулевой».

Следует отметить, что принцип «обратной связи», введенный Винером был в какой-то степени предугадан Сеченовым в явлении «центрального торможения» в «Рефлексах головного мозга» (1863 г.) и рассматривался как механизм регуляции деятельности нервной системы, и который лег в основу многих моделей произвольного поведения в отечественной психологии.

К этому времени и другие ученые стали понимать, что создателям вычислительных машин есть чему поучиться у биологии. Среди них был нейрофизиолог и поэт-любитель Уоррен Маккалох, обладавший, как и Винер, философским складом ума и широким кругом интересов. В 1942 г. Маккалох, участвуя в научной конференции в Нью-йорке, услышал доклад одного из сотрудников Винера о механизмах обратной связи в биологии. Высказанные в докладе идеи перекликались с собственными идеями Маккалоха относительно работы головного мозга. В течение следующего года Маккалох в соавторстве с блестящим 18-летним математиком Уолтером Питтсом, разработал теорию деятельности головного мозга. Эта теория и являлась той основой, на которой сформировалось широко распространенное мнение, что функции компьютера и мозга в значительной мере сходны. Так начался период нейронного подхода к созданию ИИ.

Маккалох и Питтс выдвинули гипотезу, что нейроны можно упрощенно рассматривать как устройства, оперирующие двоичными числами. Двоичные числа – рабочий инструмент одной из систем математической логики. Английский математик XIX в. Джордж Буль, предложивший эту остроумную систему, показал, что логические утверждения можно закодировать в виде единиц и нулей, где единица соответствует истинному высказыванию, а нуль – ложному, после чего этим можно оперировать как обычными числами. В 30-е годы XX в. пионеры информатики, в особенности американский ученый Клод Шеннон, поняли, что двоичные единица и нуль вполне соответствуют двум состояниям электрической цепи (включено-

выключено), поэтому двоичная система идеально подходит для электронно-вычислительных устройств. Маккалох и Питтс предложили конструкцию сети из электронных «нейронов» и показали, что подобная сеть может выполнять практически любые вообразимые числовые или логические операции. Далее они предположили, что такая сеть в состоянии также обучаться, распознавать образы, обобщать, т.е. она обладает всеми чертами интеллекта. Теории Маккаллоха-Питтса в сочетании с работами Винера вызвали огромный интерес к разумным машинам.

Из кибернетического или нейромодельного подхода к машинному разуму скоро сформировался так называемый «восходящий метод»: движение от простых аналогов нервной системы примитивных существ, обладающих малым числом нейронов, к сложнейшей нервной системе человек. Конечная цель виделась в создании «адаптивной сети», «самоорганизующейся системы» или «обучающейся машины» – все эти названия разные исследователи использовали для обозначения устройств, способных следить за окружающей обстановкой и с помощью обратной связи изменять свое поведение в полном соответствии с господствовавшей в те времена бихевиористской школой психологии, т.е. вести себя так же, как живые организмы.

Однако отнюдь не во всех случаях возможна аналогия с живыми организмами. Основной трудностью, с которой столкнулся «восходящий метод на заре своего существования, была высокая стоимость электронных элементов. Слишком дорогой оказывалась даже модель нервной системы муравья, состоящая из 20 тыс. нейронов, не говоря уже о нервной системе человека, включающей около 100 млрд. нейронов. Даже самые совершенные кибернетические модели содержали лишь несколько сотен нейронов.

В середине 1958 г. Фрэнком Розенблатом была предложена модель электронного устройства, названного им перцептроном, которое должно было бы имитировать процессы человеческого мышления. Перцептрон должен был передавать сигналы от «глаза», составленного из фотоэлементов, в блоки электромеханических ячеек памяти, которые оценивали относительную величину электрических сигналов. Эти ячейки соединялись между собой случайным образом в соответствии с господствующей тогда теорией, согласно которой мозг воспринимает новую информацию и реагирует на нее через систему случайных связей между нейронами. Два года спустя была продемонстрирована первая действующая машина «Марк-1», кото-

рая могла научится распознавать некоторые из букв, написанных на карточках, которые подносили к его «глазам», напоминающие кинокамеры. Перцептрон Розенблата оказался наивысшим достижением «восходящего» или нейромодельного метода создания искусственного интеллекта.

Ведущие представители так называемого «нисходящего метода» специализировались, в отличие от представителей «восходящего метода», в составлении для цифровых компьютеров общего назначения программ решения задач, требующих от людей значительного интеллекта, например, для игры в шахматы или поиска математических доказательств. К числу защитников «нисходящего метода» относились Марвин Минский и Сеймур Пейперт, профессора Массачусетского технологического института. В основном ИИ стал синонимом «нисходящего» подхода, который выражался в составлении все более сложных программ для компьютеров, моделирующих сложную деятельность человеческого мозга.

С точки зрения методологии ИИ твердо перешел на научные методы. Теперь, для того чтобы быть принятыми, гипотезы должны подвергаться проверке в строгих практических экспериментах, а значимость результатов должна подтверждаться данными статистического анализа. Кроме того, в настоящее время имеется возможность воспроизводить эксперименты с помощью Internet, а также совместно используемых репозитариев тестовых данных и кода.

Методология – это система правил, которые позволяют в каждый момент теоретической работы делать выбор направления, развивать его в надеже получить ожидаемый результат приемлемого качества. От методологии зависит всё. Любое решение, полученное теоретиком в области его интересов, это воплощение его методологии.

Большинство методологий имеют частный характер, т.е. конкретная методология позволяет решать узкий класс задач.

Перед создателем ИИ стоит уникальная задача в плане выбора и использования своей методологии. ИИ – это универсальная методология, которая должна быть способна решать все задачи, решаемые человеком. Это трудная задача по той причине, что сам ИИ и есть воплощение методологии его автора.

Создатель ИИ пользуется некоторой методологией в своём разуме. Цель создателя – воссоздать работу своего разума. Стало быть, воссоздать свою методологию. Очевидно, что методологией ИИ (т.е. принципом работы ИИ), должна быть та методология, которая рабо-

тала в голове создателя и привела его к созданию ИИ. Это рекурсия. Разум воссоздаёт себя – это основное методологическое правило, которое должно быть отражено в методологии ИИ.

Проблематика искусственного интеллекта – область сотрудничества многих научных дисциплин (информатика, математика, логика, психология, нейрофизиология и др.). Разумеется, ключевая роль в области искусственного интеллекта принадлежит математике. Вместе с тем к этой проблематике имеют прямое отношение, с одной стороны, многие технические дисциплины, а с другой – социальные. Все эти дисциплины обладают своим особым предметом, своими методами, своим специфическим языком. Обращаясь к проблематике искусственного интеллекта, они вынуждены иметь дело с новыми понятиями, методами, объяснительными структурами.

Философские и метанаучные подходы к проблематике ИИ способны укреплять творческую перспективу, понимание того, что эта область знания и технической деятельности представляет собой чрезвычайно многомерную, динамичную и слабо организованную структуру, что в ней могут возникать и возникают новые точки роста и новые интегративные образования, которые лежат в стороне от нынешнего магистрального пути развития информационных технологий, но способны определить их будущее.

1.2. Основные понятия и определения теории интеллектуальных информационных систем

Следует отметить, что строгого (формального, научного) определения понятия «естественный интеллект», вообще говоря, не существует. Поэтому еще труднее определить понятие «искусственный интеллект». Для того чтобы решить эту задачу, необходимо уяснить значение таких терминов, как интеллект; психика; сознание; разум.

Интеллект. Различают формулировки данного понятия по нескольким направлениям: * философскую; биологическую; психологическую.

В философии под интеллектом понимают познание, понимание, рассудочную способность к абстрактно-аналитическому расчленению (Г. Гегель), способность к образованию понятий (Э. Кант).

В психологии под интеллектом понимают характеристику умственного развития индивидуума, определяющую его способность це-

ленаправленно действовать, рационально мыслить и эффективно взаимодействовать с окружающим миром.

В биологии под интеллектом понимают способность адекватно реагировать (принимать решения) в ответ на изменение окружающей обстановки.

Важно отметить, что интеллект – это свойство отдельного субъекта. В частности, интеллектом может обладать не только человек, но и любой объект, имеющий указанные выше качества: способность к образованию понятий, абстрактно-аналитическому мышлению, целенаправленному действию.

Разум. В отличие от интеллекта разум – категория сугубо человеческая, опирающаяся на сознание как высшую форму психологической деятельности. Принципиальным моментом в определении разума так же, как и сознания, является их общественный, социальный характер, поскольку и то и другое понятия сформировались в результате совместной человеческой деятельности. Разум выступает как творческая познавательная деятельность, раскрывающая сущность действительности. Посредством разума мышление синтезирует результаты познания, создает новые идеи, выходящие за пределы сложившихся систем знания.

Сознание. Это понятие также трактуется различными науками неоднозначно.

С точки зрения философии сознание – свойство высокоорганизованной материи – мозга, выступающее как осознанное бытие, субъективный образ объективного мира, субъективная реальность.

При социологическом подходе сознание рассматривается, прежде всего, как отображение в духовной жизни людей интересов и представлений различных социальных групп, классов, наций, общества в целом.

В психологии сознание трактуется как особый, высший уровень организации психической жизни субъекта, выделяющего себя из окружающей действительности, отражающего эту действительность в форме психических образов, которые служат регуляторами целенаправленной деятельности. Важнейшей функцией сознания является мысленное построение действий и предвидение их последствий, контроль и управление поведением личности, ее способность отдавать себе отчет в том, что происходит как в окружающем, так и в собственном духовном мире.

Психика. Психика – это свойство высокоорганизованной материи – мозга, являющееся особой формой отражения действительности и

включающее такие понятия, как ощущение, восприятие, память, чувства, воля, мышление и др. Отметим, что мышление и память, которыми обычно характеризуют интеллект, входят в понятие психики составными частями.

В психике выделяют две компоненты: чувственную (ощущения, восприятие, эмоции) и рациональную, мыслительную (интеллект, мышление). Другие составляющие психики – память и волю – можно разделить на память чувств и память мыслей; волю чувств и волю мыслей (инстинкты и долг перед собой и обществом соответственно).

Например, можно помнить, как берется сложный интеграл (память мыслей), а можно помнить ощущение напряжения и усталости при изучении способа его взятия (память чувств), когда воля чувств (инстинкт самосохранения, желание отдохнуть) боролись с волей мыслей (сознанием необходимости изучения этого способа).

Отметим, что принципиально невозможно моделировать сознание и разум во всей полноте, так как для этого пришлось бы моделировать не только человека, но и всю систему его социально-общественных отношений. В то же время моделировать интеллект как одну из компонент психики отдельных индивидуумов вполне возможно, хотя и очень сложно.

К этому выводу примыкает еще один: искусственный интеллект – это модель рациональной, мыслительной составляющей психики. Не моделируются эмоции, ощущения, воля, память чувств и т. п. Машинное сочинение стихов и музыки – это моделирование лишь логической компоненты психической деятельности, сопровождающей эти виды творчества (соблюдение рифмы, размера, законов композиции, гармонии и т. п.).

Учитывая сказанное, можно заключить, что понятие «искусственный интеллект» объединяет:

искусственный бессловесный интеллект – модель компоненты психики живых существ, отражающая их способность принимать решения, изменять поведение и т. д. на уровне инстинктов, не имеющих словесного выражения (самосохранение, размножение, приспособление и т. п.;

искусственный словесный интеллект – модель рациональной компоненты психической деятельности человека без учета ее социального содержания;

искусственный разум – искусственный словесный интеллект, дополненный социальной компонентой.

Приведенные определения основаны на теоретических рассуждениях и в силу этого носят общий характер.

Существуют, по крайней мере, три подхода к определению этого понятия, носящие большую практическую направленность.

Достаточно полным определением понятия «искусственный интеллект» первого типа является следующее: ИИ – это область исследований, в рамках которых разрабатываются модели и методы решения задач, традиционно считавшихся интеллектуальными и не поддающимися формализации и автоматизации.

Применительно к данному определению является справедливым суждение, что интеллектуальной может считаться такая искусственно созданная система, для которой выполняется тест Тьюринга, состоящий в следующем. Испытатель через посредника общается с невидимым для него собеседником – человеком или системой. Интеллектуальной может считаться та система, которую испытатель в процессе такого общения не может отличить от человека.

В качестве другого определения, достаточно точно отражающего характер второго подхода, может рассматриваться следующее: искусственный интеллект – это область исследований, в которой изучаются системы, строящие результирующий вывод для задач с неизвестным алгоритмом решения на основе неформализованной исходной информации, использующие технологии символьного программирования и средства вычислительной техники со специальной (не фон Неймановской) архитектурой.

Наконец, наиболее цитируемым определением третьего типа является следующее: искусственный интеллект – это область знаний, которая находит применение при решении задач, связанных с обработкой информации на естественном языке, автоматизацией программирования, управлением роботами, машинным зрением, автоматическим доказательством теорем, разумными машинами извлечения информации и т. д.

Можно рассмотреть и такое, в какой-то степени обобщающее, определение: искусственный интеллект – научная дисциплина, задачей которой является разработка математических описаний функций человеческого (словесного) интеллекта с целью аппаратурной, программной и технической реализации этих описаний средствами вычислительной техники.

В последние годы многие специалисты согласились, что дискуссия по вопросу об определении самого термина ИИ приобрела схо-

ластический характер, не дает конструктивных результатов теории и практике и может быть бесконечной. Поэтому вместо термина ИИ предлагается использовать другой – *новая информационная технология решения инженерных задач*, что подчеркивает приоритетную роль поиска, анализа и синтеза информации в системах искусственного интеллекта.

Интеллектуальной информационной системой (ИИС) называется система, способная целеустремленно, в зависимости от состояния информационных входов, изменять не только параметры функционирования, но и сам способ своего поведения, который зависит не только от текущего состояния информационных входов, но также и от предыдущих состояний системы. Или по-другому ИИС – это система или устройство с программным обеспечением, имеющее возможность с помощью встроенного процессора настраивать свои параметры в зависимости от состояния внешней среды.

Взаимосвязь основных терминов предметной области показана на рисунке 1.1.

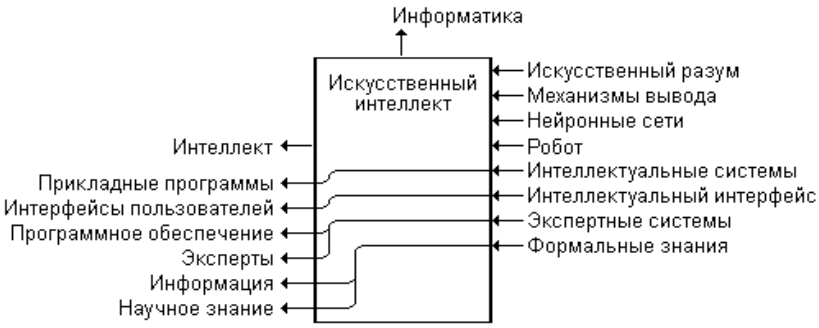


Рисунок 1.1. Основные термины искусственного интеллекта

Знания – в информатике – вид информации, отражающей опыт специалиста (эксперта) в определенной предметной области, его понимание множества текущих ситуаций и способы перехода от одного описания объекта к другому.

Интеллектуальный интерфейс – интерфейс непосредственного взаимодействия ресурсов информационного комплекса и пользователя посредством программ обработки текстовых запросов пользователя.

Механизм вывода (*Логический вывод*; *Машина вывода*) – в системах искусственного интеллекта – процедура нахождения решений

задач. Принципы построения механизма вывода определяются способом представления знаний и видом моделируемых рассуждений.

Экспертная система – система искусственного интеллекта, включающая знания об определенной слабо структурированной и трудно формализуемой узкой предметной области и способная предлагать и объяснять пользователю разумные решения.

Эксперт – специалист, который за годы обучения и практической деятельности, научился эффективно решать задачи, относящиеся к конкретной предметной области.

ИИС объединяют в себе возможности систем управления базами данных (СУБД), лежащих в основе информационных систем (ИС), и технологию искусственного интеллекта, благодаря чему хранение в них информации сочетается с её обработкой и подготовкой для использования при принятии решений. В начале ИИС, называемые также *системами, основанными на знаниях*, рассматривались как средство, позволяющее не экспертам принимать решения с таким же качеством, как и эксперты в конкретной области. Однако стало ясно, что эта технология в действительности способна достигать большего объёма знаний и быстрее реагировать, чем группа специалистов.

Первоначально ИИС использовали знания нескольких экспертов в каждой из областей знаний. В настоящее время базы знаний частично формируются посредством машинного обучения, используя методы индукции, дедукции, генетические алгоритмы и некоторые другие методы извлечения знаний.

Для ИИС 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 системы к развитию в соответствии с объективными изменениями области знаний.

Возможность ИИС работать со слабо структурированными данными подразумевает наличие следующих качеств:

решать задачи, описанные только в терминах мягких моделей, когда зависимости между основными показателями являются не вполне определенными или даже неизвестными в пределах некоторого класса;

способность к работе с неопределёнными или динамичными данными, изменяющимися в процессе обработки, позволяет использовать ИИС в условиях, когда методы обработки данных могут изменяться и уточняться по мере поступления новых данных;

способность к развитию системы и извлечению знаний из накопленного опыта конкретных ситуаций увеличивает мобильность и гибкость системы, позволяя ей быстро осваивать новые области применения;

возможность использования информации, которая явно не хранится, а выводится из имеющихся в базе данных, позволяет уменьшить объёмы хранимой фактуальной информации при сохранении богатства доступной пользователю информации.

В различных ИИС перечисленные признаки интеллектуальности развиты в неодинаковой степени и редко, когда все четыре признака реализуются одновременно.

1.3. Классификация интеллектуальных информационных систем

Условно каждому признаку интеллектуальности соответствует свой класс ИИС (рисунок 1.2).

Системы с интеллектуальным интерфейсом

Системы с интеллектуальным интерфейсом включают в свой состав следующие классы систем.

Интеллектуальные базы данных, отличающиеся от обычных баз данных тем, что по запросу можно выбрать информацию, которая может явно не храниться в базе данных, а выводиться из информации, имеющейся в базе данных. Формулирование запроса осуществляется в диалоге с пользователем.

Естественно-языковый интерфейс предполагает трансляцию естественно-языковых конструкций на внутри машинный уровень представления знаний.

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

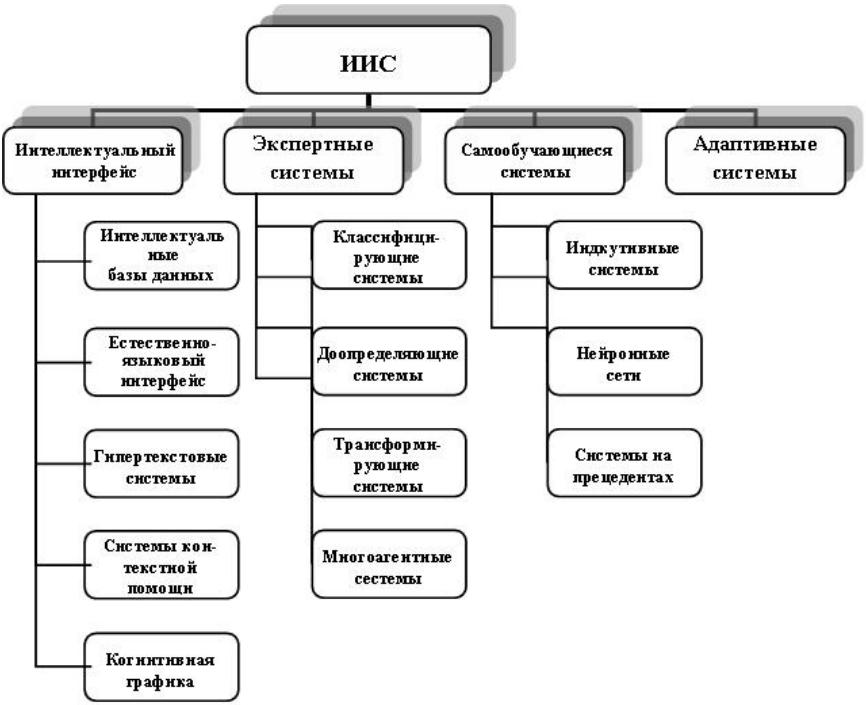


Рисунок 1.2. Классификация интеллектуальных информационных систем

Гипертекстовые системы предназначены для реализации поиска по ключевым словам в базах текстовой информации. Интеллектуальные гипертекстовые системы отличаются возможностью сложной семантической организации ключевых слов, которая отражает различные смысловые отношения терминов. Возможен 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 системах контекстной помощи пользователь описывает проблему, 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.

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. Они также нашли широкое применение в обучающих и тренажерных системах на основе использования принципов виртуальной реальности, когда графические образы моделируют ситуации, в которых необходимо принимать решения и выполнять определенные действия.

Экспертные системы

Экспертные системы предназначены для решения так называемых неформализованных задач. К неформализованным относятся такие задачи, которые обладают одной или несколькими из следующих характеристик:

задачи нельзя решить в числовой форме;

цели не удается выразить в терминах точно определенной целевой функции;

не существует алгоритмического решения задач;

алгоритмическое решение существует, но его нельзя использовать из-за ограниченности ресурсов.

Следует подчеркнуть, что неформализованные задачи представляют большой и очень важный класс задач. Многие специалисты считают, что эти задачи являются наиболее массовым классом задач, решаемых на ЭВМ.

Экспертные системы отличаются от систем обработки данных тем, что в них в основном используются символьный (а не числовой) способ представления, символьный вывод и эвристический поиск решения (а не исполнение известного алгоритма).

Самообучающиеся системы

В основе самообучающихся систем лежат методы автоматической классификации примеров ситуаций реальной практики (обучения на примерах). В результате обучения системы автоматически форми-

руются знания, используемые при решении задач классификации и прогнозирования.

Существуют несколько классов самообучающихся систем: индуктивные системы, нейронные сети, системы, основанные на прецедентах.

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

Нейронные сети являются технологией обработки информации, вдохновленной изучением мозга и нервной системы человека. Нейронные сети позволяют решать задачи прогнозирования, классификации, поиска оптимальных вариантов, и совершенно незаменимы в тех случаях, когда в обычных условиях решение задачи основано на интуиции или опыте, а не на строгом (в математическом смысле) ее описании.

Базы знаний систем, основанных на прецедентах (Case-based reasoning) содержат описания разнообразных ситуаций. Поиск решения проблемы сводится к поиску по аналогии (абдуктивному выводу от частного к частному) подходящей ситуации:

В отличие от индуктивных систем допускается нечеткий поиск с получением множества допустимых альтернатив, каждая из которых оценивается некоторым коэффициентом уверенности.

Адаптивные системы

Ядром адаптивных систем является постоянно развиваемая модель проблемной области, поддерживаемая в специальной базе знаний, – репозитории, на основе которого осуществляется генерация или конфигурация программного обеспечения. Таким образом, проектирование и адаптация системы сводится, прежде всего, к построению модели проблемной области и ее своевременной корректировке.

1.4. Данные и знания

При изучении интеллектуальных систем традиционно возникает вопрос: «Что же такое знания и чем они отличаются от обычных данных, десятилетиями обрабатываемых ЭВМ, от информации?» В обиходе эти термины часто отождествляют.

В популярной или научной литературе нет (и быть не может) универсальных определений этих понятий. Можно говорить лишь о некоторых подходах, разделяемых или не разделяемых различными авторами.

Существует множество способов определять понятия. Один из широко применяемых способов, основан на идее интенсионала. Интенсионал понятия – это определение через понятие более высокого уровня абстракции с указанием специфических свойств. Другой способ определяет понятие через перечисление понятий более низкого уровня иерархии или фактов, относящихся к определяемому. Это есть экстенсионал понятия.

Пример. Понятие «персональный компьютер». Его интенсионал: «Персональный компьютер – это дружественная ЭВМ, которую можно поставить на стол и купить менее чем за 1000…3000 долларов». Экстенсионал этого понятия: «Персональный компьютер – это Mac, IBM PC, Apple, Sinkler...» .

Данные – это отдельные факты, характеризующие объекты, процессы и явления в предметной области, а также их свойства.

Данные – сведения, представленные в определенной знаковой системе и на определенном материальном носителе для обеспечения возможностей хранения, передачи, приема и обработки.

Данные включают в себя (состоят из) описания объектов, их окружения, явлений, фактов. Это структуросоставляющее понятие, бесконечное множество реализаций записей на любых языках, доступных для нашего восприятия и интерпретации как целостных образований.

Информация – это данные, сопровождающиеся смысловой нагрузкой, помещенные в некоторый контекст; данные, как-либо оцениваемые приёмником информации (в частном случае это может быть человек, малая группа). Как правило, получение информации связывают с уменьшением неопределенности существующего выбора; ответ на какой-либо заданный, либо подразумеваемый вопрос. (При этом то, что для одних личностей (или с одной точки зрения) может быть данными, для других вполне может быть информацией).

Знание – зафиксированная и проверенная практикой информация, которая может многократно использоваться людьми для решения тех или иных задач; это выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой области; это хорошо структурированные данные, или данные о данных, или метаданные; это информация, которая отражает объективные свойства и связи некоторых объектов, явлений, процессов, сущностей и отношений между ними как в субъективном, так и в объективном (научном) выражении.

Знание, в общем случае, является переменной во времени и контексте совокупностью именованных отношений между данными.

Это структурообразующее понятие, бесконечный процесс изменения связей данных.

С позиций искусственного интеллекта под знаниями подразумевают информацию, которая необходима программе, чтобы она вела себя «интеллектуально». Обычно такая информация представляется в виде фактов и правил, которые ЭВМ может использовать при решении задач по таким алгоритмам, как логические выводы.

Общение на уровне знания должно предопределять возможность равнопонятного (воспринимаемого в одном контексте) двум и более системам использования взаимосвязанной информации на языке, способном нести и передавать данные и знание о предметной области.

Различия между данными и знаниями становятся очевидными, если применительно к искусственному интеллекту рассмотреть некоторые операции их обработки.

При обработке на ЭВМ данные трансформируются, условно проходя следующие этапы:

данные как результат измерений и наблюдений;

данные на материальных носителях информации (таблицы, протоколы, справочники;

модели (структуры) данных в виде диаграмм, графиков, функций;

данные в компьютере на языке описания данных;

базы данных на машинных носителях.

Знания связаны с данными, основываются на них, но представляют результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе выполнения какой-либо практической деятельности. Они получаются эмпирическим путем.

При обработке на ЭВМ знания трансформируются аналогично данным:

знания в памяти человека как результат мышления;

материальные носители знаний (учебники, методические пособия);

поле знаний – условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих;

знания, описанные на языках представления знаний;

базы знаний.

Одним из проявлений интеллектуальности компьютерной системы является возможность организации управления в ней знаниями человека (эксперта в некоторой предметной области). Теоретическими и практическими вопросами представления и обработки знаний в компьютерных системах занимается отдельное направле-

ние искусственного интеллекта – инженерия знаний. Программы, использующие достижения в этой области, образуют отдельный класс компьютерных систем, основанный на знаниях.

Знания могут быть классифицированы по следующим категориям:

поверхностные – знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области и *глубинные* – абстракции, аналогии, схемы, отображающие структуру и процессы в предметной области;

декларативные – знания, представляющие собой описания фактов и явлений, а также связанных с ними закономерностей и *процедурные* – знания, описывающие действия, которые возможны при манипулировании фактами и явлениями для достижения намеченных целей;

явные – знания, которые сформулированы, зафиксированы, представлены в виде накопленного опыта, который может быть выделен и представлен в форме методик, инструкций, руководств, рекомендаций к действию;

неявные (*неформальные*) – знания, которые трудно, либо затруднительно формализовать; интуитивные знания, ощущения, впечатления, мнения. Если неявные знания не «извлечь» из экспертов (сотрудников) с помощью, например, инженеров по знаниям, не формализовать, то они могут быть утеряны по тем или иным причинам;

процедурные – знания, «растворенные» в алгоритмах,

декларативные – знаний сосредоточенные в структурах данных (таблицы, списки, абстрактные типы данных);

скрытые – знания, неразрывно связанные с его носителем: человеком, малой группой, организацией, с их индивидуальным и корпоративным опытом.

Экстенсиональные – простейшие знания, представленные в виде фактов, описывающих конкретное состояние объекта в конкретный момент или промежуток времени (факты принято задавать совокупностью пар вида «атрибут – значение», называемых атрибутными парами);

интенсиональные – знания, представленные в виде фактов, задаваемых парами «атрибут – домен», где домен – совокупность всех допустимых значений, которые может принимать конкретный атрибут.

Предметной или проблемной областью мы будем называть совокупность данных и знания, представляющуюся достаточной для информационного (заданного на контекстно-зависимом языке) обеспечения решения некоторой задачи или совокупности задач интеллектуальной системы.

Сложность понятия знание, всегда выражаемого на некотором языке отношений, заключена в множественности возможностей его реализации и неразрывностью с понятием данные, в непрерывном процессе его изменения, учет которого обеспечивает реализацию контекстной связи данных.

Можно выделить несколько уровней представления знания в его связи с данными:

уровень интеллектуальной системы – знание и данные существуют в форме языковой модели предметной области (модели, во многих, если не во всех случаях неотличимой от самой предметной области, существующей как языковое описание) и как описание составляющих этой системы на уровне контекстно-зависимого языка;

уровень информационной системы – знание и данные существуют в форме языковой модели (именно модели, выделяющей из реальной системы нечто, признаваемое "существенным") на основе использования контекстно-независимых языков;

уровень математической модели – данные, формализованные на уровне языка формул и передаточных функций, содержат в себе знание как формализованные правила и аппарат производства выводов;

уровень фактографической модели – текстовые записи с фиксированной на уровне языка их представления системой отношений между ними (например, табличная запись).

Приведенный список характеризует различные уровни работы с информацией, выделяет качественно различные группы информационных технологий, особо подчеркивая возможности работы со знанием на уровне контекстно-зависимого описания предметной области, т.е. на уровне семантики и контекстной зависимости трактовки каждой информационной посылки.

Между первым и последующими уровнями представления знания проходит стена, отделяющая интеллектуальные системы от неинтеллектуальных, а само понятие знания не является исключительной принадлежностью интеллектуальных систем и кардинально изменяется на каждом уровне его представления.

На первом уровне оно связано с выражением контекстно-зависимых отношений между данными. Здесь знание определяется текущей структурой связей между данными в некоторой информационной базе, которая изменяется после каждого акта общения, после обработки каждого входящего сообщения на контекстно-зависимом языке.

Это высшая инженерная форма представления знания, допускающая для сохранения семантики интерпретации данных как актуализацию этих данных и возможных типов отношений между ними, так и актуализацию самой структуры их связей.

На втором уровне знание понимается лишь как процедура единственно возможной содержательной интерпретации связей данных. Отношения между данными практически неизменны и определены исходной структурой базы данных. Знание, как это и следует предполагать, для неинтеллектуального уровня, определяемого использованием алгоритмического языка, обеднено стабильностью связей между данными.

Можно сказать, что это уровень фиксированного знания, знания, которое мы понимаем как справочные сведения с возможностью актуализации данных, но не связей между ними (если, конечно, эти связи не выведены на уровень данных).

На третьем уровне речь идет об использовании различного рода математических выражений, экономно описывающих данные и их связи. В пределах этого языка находятся и интерпретируются связи между данными, то есть «пространный» контекстно-независимый язык здесь по возможности заменяется функциональным языком математики. Все остальное здесь аналогично второму уровню.

Четвертый уровень фактически соответствует предельно обедненному второму уровню и рассматривается только в связи с необходимостью обеспечения сравнительного анализа с другими подходами к определению понятия знания. Это связано с тем, что фактографические данные, как записи с обедненной структурой их связей, зачастую выделяются в отдельную группу.

Обычно в литературе указывается, что основные исследования и разработки, которые ведутся в области технологии интеллектуальных систем, охватывают все уровни работы с данными, при этом рассматриваются особенности знания, которые отличают его от данных.

Правильный подход здесь один: во-первых, все уровни работы с данными, кроме первого, не являются интеллектуальной задачей, а во-вторых, данные не существуют без отношений между ними (того или иного варианта упорядочения, изменяющегося в ту или иную сторону по мере получения новых информационных посылок) и соответственно, не отличаются, а неразделимы со знанием и текущим контекстом интерпретации.

Это можно высказать кратко: когда знание перестает быть процессом, а контекст предопределен – интеллект исчезает.

Особенности знаний:
внутренняя интерпретируемость;
рекурсивная структурированность и связность;
шкалирование;
семантическое пространство с метрикой;
наличие активности;
функциональная целостность;
независимость;
ситуативность.

Внутренняя интерпретируемость. Каждая информационная единица должна иметь уникальное имя, по которому ИС находит ее, а также отвечает на запросы, в которых это имя упомянуто. Поясним понятие на следующем примере.

Фамилия	**Год рождения**	**Специальность**	**Курс**
Никифоров	1990	Прикладная информатика	3
Аникеева	1988	Математика	5
Петрушин	1991	История	2
Сидорова	1992	Информатика	1

Если, например, в память ЭВМ нужно было записать сведения о сотрудниках учреждения, представленные в таблице, то без внутренней интерпретации в память ЭВМ была бы занесена совокупность из четырех машинных слов, соответствующих строкам этой таблицы. При этом информация о том, какими группами двоичных разрядов в этих машинных словах закодированы сведения о специалистах, у системы отсутствуют. Они известны лишь программисту, который использует данные таблицы для решения возникающих у него задач. Система не в состоянии ответить на вопросы типа: «Что тебе известно о Нифорове?» или «Есть ли среди студентов обучающийся по специальности «Математика» ?

При переходе к знаниям в память ЭВМ вводится информация о некоторой протоструктуре информационных единиц. В рассматриваемом примере она представляет собой специальное машинное слово, в котором указано, в каких разрядах хранятся сведения о фамилиях, годах рождения, специальностях и курсе обучения. При

этом должны быть заданы специальные словари, в которых перечислены имеющиеся в памяти системы фамилии, года рождения, специальности и курсы. Все эти атрибуты могут играть роль имен для тех машинных слов, которые соответствуют строкам таблицы. По ним можно осуществлять поиск нужной информации. Каждая строка таблицы будет экземпляром протоструктуры.

Информационная единица для первой строки с именем «Никифоров» будет выглядеть так: (Никифоров <Год рождения, 1990> <Специальность, Прикладная информатика> <Курс, 3>). Информационная единица для столбца с именем Год рождения будет записана в следующем виде: (Год рождения <Никифоров, 1990> <Аникеева, 1998> <Петрушин, 1991> <Сидорова, 1992>). Круглые скобки выделяют содержание информационной единицы, а угловые скобки – определенные самостоятельные части, называемые *слотами*. В общем случае полная запись информационной единицы будет выглядеть так: (Имя информационной единицы <Имя первого слота, Значение первого слота> <Имя второго слота, Значение второго слота>…).

В памяти ЭВМ вся эта таблица хранится как отдельная информационная единица.

Внутренняя интерпретируемость знаний формально отождествляется с наличием некоторой функции (способа) связывания переменных с теми или иными значениями или иначе, – способом порождения пар <*Vi*, *Vj*> (связанной переменной), где *Vi* – имя переменной, а *Vj* – исходный объект. Такая функция или способ называется также интерпретирующей функцией (или просто интерпретацией).

Поскольку данные – это пары вида <C, C>, где C – константа, то, очевидно, для них не требуется интерпретирующая функция (т.к. данные представляют сами себя).

Рекурсивная структурированность и связность. Информационные единицы должны обладать гибкой структурой. Для них должен выполняться «принцип матрешки», т.е. рекурсивная вложимость одних информационных единиц в другие. Каждая информационная единица может быть включена в состав любой другой, и из каждой информационной единицы можно выделить некоторые составляющие ее информационные единицы. Другими словами, должна существовать возможность произвольного установления между отдельными информационными единицами отношений типа «часть – целое», «род – вид» или «элемент – класс». Такая структура получила название *фрейма* (рисунок 1.3).

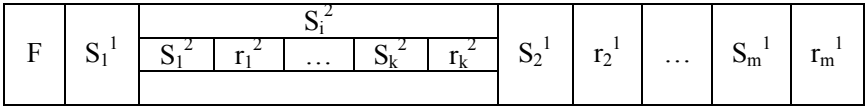


Рисунок 1.3. Структура фрейма:
F –имя фрейма; S_1^1 – имя первого слота; S_i^2 – имя второго слота, содержащего k слотов второго уровня вложенности

Пользуясь предыдущим примером, нетрудно представить себе фрейм, скажем с четырьмя уровнями вложений. Первый уровень – название класса «Студенты». Второй уровень могли бы составить 2 слота – «Год рождения до 1990 включительно» и «Год рождения после 1990». Третий уровень вложения – слоты с именами «Никифоров», «Аникеева», «Петрушин», «Сидорова». В качестве четвертого уровня – слоты, отражающие признаки каждого студента (год рождения, специальность, курс) согласно таблице.

Между слотами различных уровней могут быть самые разные отношения. Вложенными могут быть и фреймы, имеющие уже свои многоуровневые вложения. Вложенной может быть команда или даже целая программа. Все это придает фреймовым структурам большую гибкость и многосвязность.

Таким образом, в информационной базе между информационными единицами должна быть предусмотрена возможность установления связей различного типа. Прежде всего, эти связи могут характеризовать отношения между информационными единицами. Семантика отношений может носить декларативный или процедурный характер. Например, две или более информационные единицы могут быть связаны отношением «одновременно», две информационные единицы – отношением «причина – следствие» или отношением «быть рядом». Приведенные отношения характеризуют декларативные знания. Если между двумя информационными единицами установлено отношение «аргумент – функция», то оно характеризует процедурное знание, связанное с вычислением определенных функций. Далее будем различать отношения структуризации, функциональные отношения, каузальные отношения и семантические отношения. С помощью первых задаются иерархии информационных единиц, вторые несут процедурную информацию, позволяющую находить (вычислять) одни информационные единицы через другие, третьи задают причинно-следственные связи, четвертые соответствуют всем остальным отношениям.

Между информационными единицами могут устанавливаться и иные связи, например, определяющие порядок выбора информационных единиц из памяти или указывающие на то, что две информационные единицы несовместимы друг с другом в одном описании.

Рекурсивная структурированность определяется через понятие концепта. *Концепт* есть семантическое целое понятий. Понятие P_1 и P_2 образуют семантическое целое, если имеет место какое-то из перечисленных условий:

(A) P_1 и P_2 связаны друг с другом как отношение и один из его аргументов;

(B) P_1 и P_2 связаны друг с другом как действие и его носитель или субъект;

(C) P_1 и P_2 связаны друг с другом как функция и ее аргумент (объект и его свойство).

По индукции следует, что множество $P = \{P_1, P_2, ..., P_n\}$ понятий образует семантическое целое, если каждый P_i в P образует семантическое целое как минимум с одним из P_j, где $P_j \in P/\{P_i\}$.

Концепт C есть упорядоченное множество C_i, где C_i - семантическое целое, содержащееся в семантическом целом, представляющем C.

Рассмотрим концепт C: «Студент Петр сдает экзамен по информатике». Здесь можно выделить следующие семантические целые:

C_1 = <Студент Петр> (объект Петр; свойство - студент)

C_2 = <Петр сдает экзамен> (объект Петр; действие - сдает экзамен)

C_3 = <Экзамен по химии> (объект экзамен; свойство - по информатике)

Таким образом, исходный концепт C образован простым объединением

$C = C_1 \cup C_2 \cup C_3$.

Отметим, что результат не зависит от порядка слагаемых.

Рекурсивность операции объединения $\cup$ показывается элементарно через тождество $\varnothing \cup X = X$, $\{a\} \cup X = \{a, X\}$, $\{a, Y\} \cup X = \{a\} \cup \{X \cup Y\}$.

Перечисленные три особенности знаний позволяют ввести общую модель представления знаний, которую можно назвать семантической сетью, представляющей собой иерархическую сеть, в вершинах которой находятся информационные единицы. Эти единицы снабжены индивидуальными именами. Дуги семантической сети соответствуют различным связям между информационными единицами.

При этом иерархические связи определяются отношениями структуризации, а неиерархические связи – отношениями иных типов.

Шкалирование. Систематизация информационных единиц предполагает использование различного вида шкал. Это могут быть строгие *метрические* шкалы, к примеру, шкала упорядочения людей по возрасту, шкала весовых категорий и т.п.; *неметрические* или *размытые* шкалы – Никогда, Очень редко, Редко, Часто, Очень часто, Почти всегда, Всегда и т.п.; *оппозиционные* шкалы типа Хороший – Плохой, Острый – Тупой и т.п.

Семантическое пространство с метрикой. На множестве информационных единиц в некоторых случаях полезно задавать отношение, характеризующее ситуационную близость информационных единиц, т.е. силу ассоциативной связи между информационными единицами. Его можно было бы назвать отношением релевантности для информационных единиц. Такое отношение дает возможность выделять в информационной базе некоторые типовые ситуации (например, «покупка», «регулирование движения на перекрестке»). Отношение релевантности при работе с информационными единицами позволяет находить знания, близкие к уже найденным. Знания, близкие по своему смыслу, объединяют в *кластеры*.

Наличие активности. С момента появления ЭВМ и разделения используемых в ней информационных единиц на данные и команды создалась ситуация, при которой данные пассивны, а команды активны. Все процессы, протекающие в ЭВМ, инициируются командами, а данные используются этими командами лишь в случае необходимости. Для интеллектуальных систем эта ситуация не приемлема. Как и у человека, в интеллектуальных системах актуализации тех или иных действий способствуют знания, имеющиеся в системе. Таким образом, выполнение программ в интеллектуальной системе должно инициироваться текущим состоянием информационной базы. Появление в базе фактов или описаний событий, установление связей может стать источником активности системы.

Функциональная целостность. Под функциональной целостностью знаний понимается их непротиворечивость, независимость исходных посылок и разрешимость. Поскольку понятия непротиворечивости и разрешимости являются фундаментальными понятиями логики, то их рассмотрение вынесено в самостоятельные подразделы. Ограничимся здесь указанием на то, что непротиворечивость знаний означает невозможность появления в базе знаний двух взаи-

моисключающих фактов φ и $\overline{\varphi}$ (типа «пациент жив» и «пациент мертв»). Хотя принципиально возможно рассуждение на основе противоречивых посылок, т.к. в настоящее время не существует развитой теории для этого случая.

Требование разрешимости заключается в том, что любое истинное знание, формализуемое в базе знаний системы, может быть выведено в ней с помощью машины вывода. Требование разрешимости, к сожалению, не всегда выполнимо.

6. *Независимость* означает невозможность вывода единого знания из другого формальным способом.

7. *Ситуативность*. Наличие ситуативных связей определяет совместимость тех или иных знаний, хранимых в памяти. В качестве таких связей могут выступать отношения времени, места, действия, причины и пр. Для нас важны модели представления знаний в ЭВМ. При этом основными требованиями к представлению знаний являются однородность представления, простота понимания, непротиворечивость и полнота.

Перечисленные особенности информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в базы знаний (БЗ). Совокупность средств, обеспечивающих работу со знаниями, образует систему управления базой знаний (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы внутренняя интерпретируемость, структуризация, связность, введена семантическая мера и обеспечена активность знаний.

Существуют десятки моделей (или языков) представления знаний (МПЗ) для различных предметных областей. Большинство из них может быть сведено к следующим типам:

формальные МПЗ;

неформальные (семантические, реляционные) МПЗ;

продукционные;

интегрированные МПЗ.

К формальным МПЗ относятся модели, построенные на *основе исчисления высказываний* и *исчисления предикатов*.

К неформальным МПЗ относятся:

продукционные МПЗ;

семантические сети;

фреймовые МПЗ

Интегрированные МПЗ совмещают в себе модели различных типов.

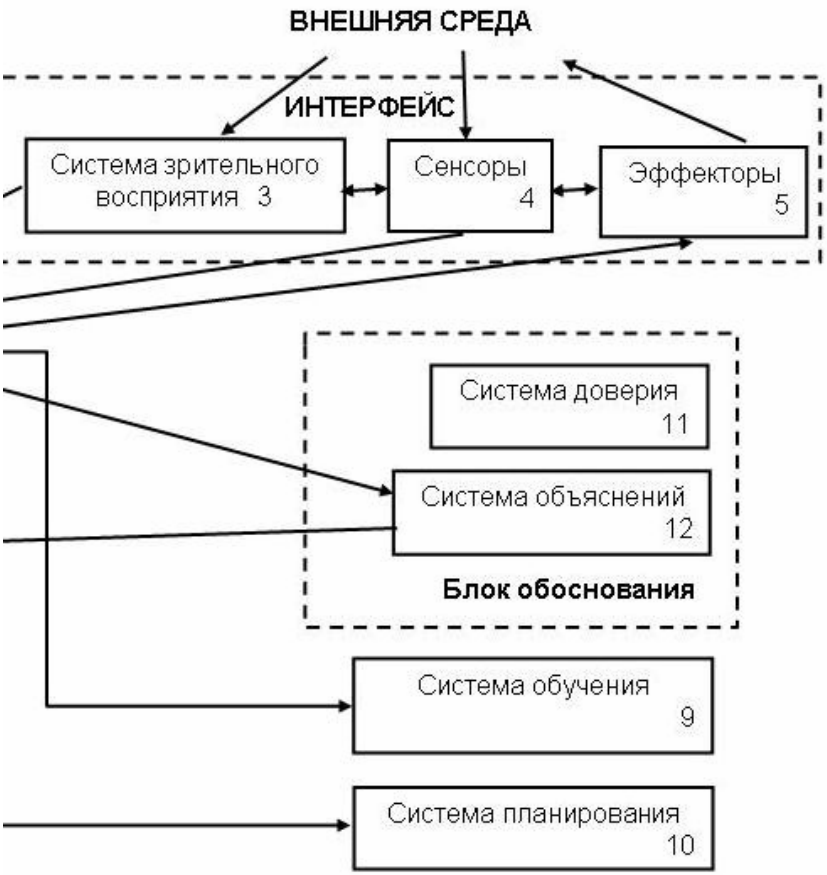
1.5. Архитектура систем искусственного интеллекта

Интеллектуальные информационные системы относятся к интеллектуальными программным системам, общая структура которых представлена на рисунке 1.4.

Рисунок 1.4. Структура интеллектуальной информационной системы

Система содержит 13 функциональных блоков, часть которых может быть объединена в функциональные группы. Одной такой группой является интеллектуальный интерфейс, обеспечивающий эффективную связь всей системы с пользователем и внешней средой.

В состав интеллектуального интерфейса могут входить блоки 1-4 и 13. Лингвистический процессор обеспечивает связь пользователя с



системой на естественном (почти всегда ограниченном) языке: ввод и понимание системой текстов на нем и вывод текстов, вырабатываемых системой. Для голосового общения пользователя с системой

используется система анализа и синтеза речи. Информация от внешней среды воспринимается системой с помощью сенсоров, представляющих собой аппаратно реализуемые чувствительные элементы. При этом зрительная информация перед поступлением в систему обрабатывается в системе зрительного восприятия. Если система имеет возможность воздействовать на внешнюю среду, то в состав интеллектуального интерфейса должен быть включен блок эффекторов. Система когнитивной графики позволяет пользователю воспринимать результаты работы системы в графической форме и общаться с ней на языке графики.

Центральным блоком интеллектуальной системы является решатель – вычислительная система, состоящая из одной или нескольких ЭВМ (процессоров), связанная с базами данных и знаний, а также с остальными блоками системы. Целенаправленная работа системы обеспечивается системой планирования, хранящей априорно введенные цели, а также запоминающей новые цели, полученные с помощью системы обучения. Последняя участвует также в формировании новых знаний, возникающих в ходе анализа взаимодействия интеллектуальной системы с внешней средой. Группа блоков обоснования, включающих систему объяснения и систему доверия, служит для обоснования полученных системой решений (если пользователь интересуется этим) с привлечением информации, содержащейся в базе данных.

Заметим, что все перечисленные блоки, за исключением блоков 4 и 5, могут быть реализованы как на специальных аппаратных средствах, так и в решателе с использованием его логико-вычислительных возможностей. Кроме того, в зависимости от степени развития и функциональных возможностей конкретных интеллектуальных систем в их структуре часть перечисленных блоков может не входить.

Для установления соответствия между конкретными функциональными структурами основных типов интеллектуальных систем и типовой схемой, представленной на рисунке 1.4, рассмотрим таблицу.

Вид интеллектуальной системы	**Номер блока**											
	1	2	3	4	5	6	7	8	9	10	11 и 12	13
Экспертная	+	0	-	-	-	+	+	+	0	0	+	0
Информационная	+	0	-	-	-	+	+	+	-	-	-	-

Расчетно-логическая (гибридная)	+	-	-	-	-	+	+	+	0	+	-	+
Проектирование научных исследований	+	-	0	-	-	+	+	+	-	+	0	+
Обучающая	+	0	-	-	-	+	+	+	+	+	+	-
Интеллектуальные роботы	+	+	+	+	+	+	+	+	0	+	0	0

В ней для каждого вида интеллектуальной системы показано, какие блоки в этот вид обязательно входят (+) и какие блоки не входят (–). Нулями отмечены блоки, которые могут входить или не входить в соответствующую систему в зависимости от характера решаемых задач и степени технического совершенства системы.

ИИС являются результатом логического развития СОД. В ИИС реализуются три группы основных функций: представления и обработки данных; рассуждения; общения с пользователем на естественном языке.

Эти обобщенные функции реализуются, как правило, следующей совокупностью процедур:

1ф
- накопление знаний о предметной области;
- классификация знаний по критерию прагматической полезности и непротиворечивости;
- структурирование знаний в направлении их использования в конкретной области;
- автоматическое поддержание базы знаний при ее пополнении;
- получение и обработка знаний от нескольких экспертов.

2ф
- инициализация процессов получения новых знаний;
- соотнесение новых знаний со старыми;
- пополнение знаний с получением логического вывода, отражающего закономерности в предметной области и накопленных знаниях;
- обобщение знаний на основе более частных знаний;
- логическое планирование своей деятельности;
- осуществление вывода на основе рассуждений по аналогии и т.п.

Зф
- общение на естественном языке (или подмножестве профессионального языка);
- обучение;
- адаптация в процессе взаимодействия к специалистам разной квалификации;
- введение знаний о целях и возможностях пользователя, а также о собственных возможностях;
- формирование по запросу пользователя объяснений своей деятельности;
- документирование информации в форме, необходимой пользователю.

Естественно, что вышеперечисленные процедуры должны быть реализованы той или иной формой организации системы, т.е. ее архитектурой.

В связи с тем, что в данном курсе нас будут интересовать в основном первые две функции, то рассмотрим обобщенную структурно-функциональную схему ИИС упрощенного вида (рис.1.5).

Архитектура конкретной ИИС определяется функциями конкретного состава задач и их связями между собой.

В приведенной схеме можно выделить два основных блока: машины баз знаний; решатель и вспомогательные блоки: систему общения на естественном языке (ЕЯ), рецепторы и эффекторы.

Это деление на блоки абсолютно четко соответствует обобщенным функциям ИИС.

МБЗ реализуют первую функцию ИИС – функцию представления и обработки знаний и состоит из блоков:

База фактов содержит факты, носящие конкретный характер:

факты, характеризующие текущую ситуацию, текущее состояние;

факты, характеризующие уже имевшие место ситуации (опыт).

База правил содержит элементарные выражения, называемые в теории ИИ продукциями. Здесь содержатся закономерности, представляющие, как правило, причинно-следственные связи предметной области. Это предложения типа ЕСЛИ–ТО–ИНАЧЕ.

База процедур хранит то, что в обычных СОД называется прикладными программами. С помощью этих процедур выполняются все необходимые вычисления, преобразования и другие нужные системе последовательности внутренних действий.

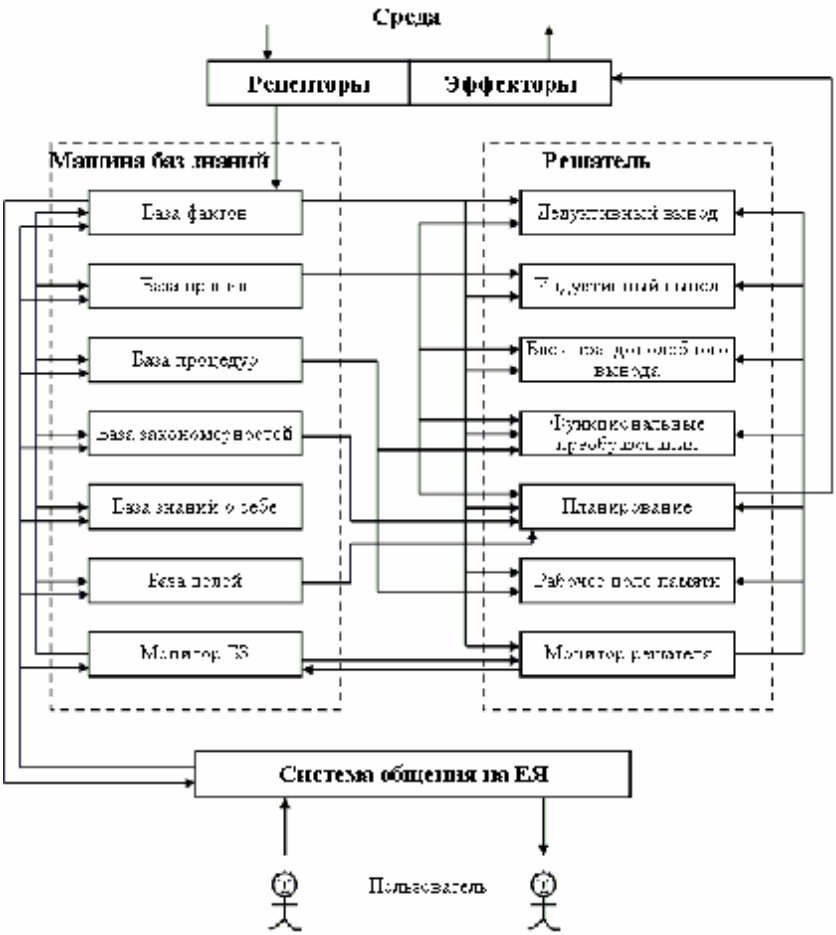


Рисунок 1.5. Структурно-функциональная схема ИИС

База закономерностей содержит различные сведения, относящиеся к особенностям той предметной области, в которой будет функционировать система. Это, например, законы предметной области, эмпирические зависимости.

База знаний о себе содержит списки того, что хранится в текущий момент в остальных базах:

сведения о том, как представляются единицы информации различного типа;

сведения о том, как взаимодействуют отдельные части системы;

сведения о том, как получено решение любой конкретной задачи.

Другими словами, хранятся на том или ином уровне подробности описания самой системы, ее возможностей и способов функционирования.

База целей содержит такие целевые структуры, которые позволяют организовать процессы движения от исходных фактов, закономерностей, правил и процедур к достижению той цели, которая поступила в систему от пользователя или сформирована в самой системе в процессе ее функционирования. База целей – это тоже способ представления знаний, как и база правил, но ориентирована на связи объектов между собой через сообщения, что позволяет реализовывать стратегию вывода, например, по сценариям.

Этот момент является одним из существенных отличий ИИС от СОД. В традиционной СОД схема передачи управления и использования данных предопределена в программе. Обработка информации осуществляется последовательными шагами, а ветвление имеет место в заранее выбранных точках. В ИИС ходом рассуждения управляют данные и ветвление в ходе рассуждения – норма. При этом правила дают возможность на любом шаге оценить ситуацию и предпринять соответствующие действия.

Монитор баз знаний – это программа управления всеми базами, входящими в базу знаний. Эта программа организует их взаимодействие между собой.

Т.о., база фактов – это база данных, а база правил, база закономерностей, база целей составляют основу базы знаний предметной области.

МДВ реализует вторую функцию ИИС – функцию рассуждений, которая состоит из 7 элементов:

БДВ реализует дедуктивные рассуждения, с помощью которых на основании общих закономерностей из базы закономерностей, конкретных фактов из базы фактов и правил вывода из базы правил выводятся новые факты. С помощью этого же блока происходит поиск пути на сценариях при заданной конечной цели.

Однако строгий формальный вывод может быть обеспечен не всегда, может отсутствовать необходимая информация, могут существовать ограничения на принятые решения и т.п. Поэтому *БИВ* целесообразен в ИИС для организации вывода новых знаний на основе обобщений отдельных понятий и фактов, замена операций с отдельными понятиями и фактами операциями с их множествами. Конечно реализация вывода по индукции труднее, чем по дедукции, но зато ИИС получает значительные интеллектуальные преимущества.

В процессе индуктивного и дедуктивного выводов возможны ошибки. Чтобы их устранить, необходимо использовать определенные указатели правдоподобия сформированных правил, реализуемых в *БПВ*.

Поскольку ИИС должна работать на естественном языке или его подмножестве из области профессионального языка, то этот блок и должен отражать те степени правдоподобия правил, фактов, которые имеются в естественном языке (ЕЯ) в виде выражений: «вероятно», «часто», «много раз», … В этом блоке целесообразно моделировать и правдоподобные рассуждения, т.е. элементы переноса свойств, выявленных для одних фактов и ситуаций, на другие, кажущиеся по набору признаков похожими на уже изученные (рассуждения по аналогии и ассоциации).

Блок планирования – этот блок связан со всеми БМЗ, планирует процесс вывода в зависимости от конкретной ситуации.

Монитор решателя – программа, управляющая всеми блоками решателя.

Блок рабочего поля памяти отражает реальную ситуацию использования памяти компьютера при решении интеллектуальных задач. В нее блоки индуктивного и дедуктивного вывода вызывают из БЗ необходимые знания, чтобы не искажать БЗ различными преобразованиями, нужными для решения задач.

2. Экспертные системы

2.1. Определение и классификация экспертных систем

Экспертные системы предназначены для решения так называемых неформализованных задач. Ранее мы уже дали понятие неформализованных задач, а также определили их свойства.

Экспертная система отличается от прочих прикладных программ наличием следующих признаков.

Моделирует *механизм мышления человека*, умеющего решать задачи определенной проблемной области. Основное внимание уделяется воспроизведению компьютерными средствами методики решения проблемы, которая применяется экспертом.

Система, помимо выполнения вычислительных операций, формирует определенные *соображения и выводы, основываясь на тех знаниях,* которыми она располагает.

При решении задач основными являются *эвристические и приближенные методы*, которые, в отличие от алгоритмических, не всегда гарантируют успех. Эвристика – это знание, приобретенное человеком по мере накопления практического опыта решения проблем. Такие методы являются приблизительными в том смысле, что, во-первых, они не требуют исчерпывающей исходной информации, и, во-вторых, существует определенная степень уверенности (или неуверенности) в том, что предлагаемое решение является верным.

На рисунке 2.1 представлена связь основных понятий предметной области «Экспертные системы».

Экспертная система **(ЭС)** – система искусственного интеллекта, включающая знания об определенной слабо структурированной и трудно формализуемой узкой предметной области и способная предлагать и объяснять пользователю разумные решения. Экспертная система состоит из базы знаний, механизма логического вывода и подсистемы объяснений.

База знаний (БЗ) – семантическая модель, описывающая предметную область и позволяющая отвечать на такие вопросы из этой предметной области, ответы на которые в явном виде не присутствуют в базе. База знаний является основным компонентом интеллектуальных и экспертных систем.

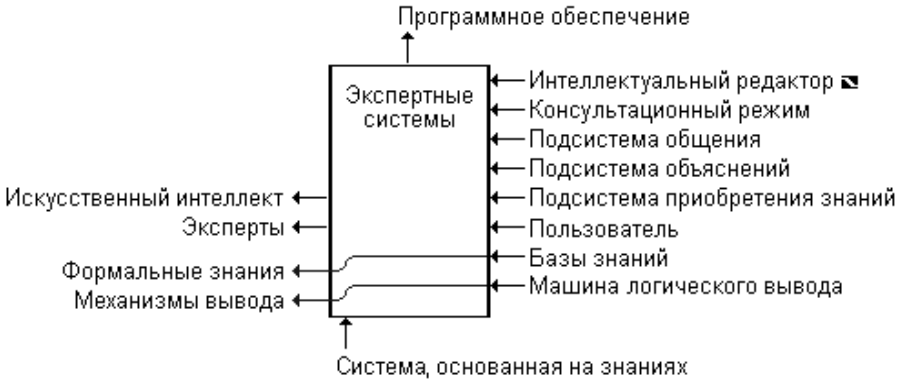


Рисунок 2.1. Основные понятия предметной области

Интеллектуальный редактор базы знаний – программа, предоставляющая инженеру по знаниям возможность создавать базу знаний в интерактивном режиме. Интеллектуальный редактор включает в себя систему шаблонов языка представления знаний, подсказки и другие сервисные средства, облегчающих работу с базой.

Консультационный режим – в экспертных системах – интерактивный режим эксплуатации базы знаний, при котором пользователь продвигается к решению задачи.

Машина логического вывода **(**Дедуктивная машина; Блок логического вывода; Решатель) – программа, моделирующая механизм рассуждений и оперирующая знаниями и данными с целью получения новых данных из знаний и других данных, имеющихся в рабочей памяти. Обычно машина логического вывода использует программно реализованный механизм дедуктивного логического вывода или механизм поиска решения в сети фреймов или семантической сети.

Подсистема общения – программа, входящая в состав экспертной системы, служащая для ведения диалога с пользователем, в ходе которого ЭС запрашивает у пользователя необходимые факты для процесса рассуждения, и предоставляющая возможность пользователю в определенной степени контролировать и корректировать ход рассуждений экспертной системы.

Подсистема объяснений – программа, позволяющая пользователю получить ответы на вопросы: «Как было получено то или иное решение?» – обычно ответ на этот вопрос представляет собой трасси-

ровку всего процесса вывода решения с указанием использованных фрагментов базы знаний; и «Почему было принято такое решение?» – обычно ответ на этот вопрос есть ссылка на умозаключение, непосредственно предшествовавшее полученному решению.

Подсистема приобретения знаний – программа, предназначенная для корректировки и пополнения базы знаний. Подсистема приобретения знаний – в простейшем случае – интеллектуальный редактор базы знаний. Подсистема приобретения знаний – в более сложных случаях – средства для извлечения знаний из баз данных, из неструктурированного текста, из графической информации и т.д.

Пользователь в экспертных системах – проблемный специалист, для которого предназначена экспертная система. Считается, что квалификация пользователя недостаточно высока, и поэтому он нуждается в помощи и поддержке своей деятельности со стороны экспертной системы.

Программное обеспечение (ПО) – комплекс программ, обеспечивающих обработку или передачу данных, предназначенных для многократного использования и применения разными пользователями. По видам выполняемых функций программное обеспечение подразделяется на системное, прикладное и инструментальное. Программное обеспечение согласно ГОСТ 19781-90 – совокупность программ системы обработки информации и программных документов, необходимых для их эксплуатации.

Система, основанная на знаниях – система искусственного интеллекта, в которой предметные знания представлены в явном виде и отделены от прочих знаний системы.

В основу экспертных систем положены три принципа:

1. Мощность ЭС обусловлена в первую очередь мощностью базы знаний и возможностью ее пополнения.

2. Знания, позволяющие ЭС получить качественные и эффективные решения, являются в основном эвристическими, экспериментальными.

Причина этого заключается в том, что решаемые задачи являются неформализованными или слабо формализованными. Необходимо также подчеркнуть, что знания экспертов имеют индивидуальный характер, т.е. свойственны конкретному человеку.

3. Учитывая неформализованность решаемых задач и эвристический, личностный характер используемых знаний, пользователь (эксперт) должен иметь возможность непосредственно взаимодействовать с экспертной системы в виде диалога.

Класс «экспертные системы» сегодня объединяет несколько тысяч различных программных комплексов, которые можно классифицировать по различным классификационным признакам.

1. Тип приложений

Возможность взаимодействия приложения с другими программными средствами:

изолированное приложение – ЭС, не способная взаимодействовать с другими программными системами (например, с БД, электронными таблицами, пакетами прикладных программ, контроллерами, датчиками и т.п.);

интегрированное приложение – ЭС и другие программные системы, с которыми она взаимодействует в ходе работы.

Возможность исполнять приложение на разнородной аппаратуре и переносить его на различные платформы:

закрытые приложения исполняются только в программной среде данной фирмы и могут быть перенесены на другие платформы только путем перепрограммирования приложения;

открытые приложения ориентированы на исполнение в разнородном программно-аппаратном окружении и могут быть перенесены на другие платформы без перепрограммирования.

Архитектура приложения:

централизованное приложение реализуется на базе центральной ЭВМ, с которой связаны терминалы;

распределенное приложение обычно используется архитекра клиент-сервер.

2. Стадия существования

Характеризует степень завершенности разработки ЭС. Принято выделять следующие стадии:

исследовательский прототип – решает представительный класс задач проблемной области, но может быть неустойчив в работе и не полностью проверен. При наличии развитых инструментальных средств для разработки исследовательского прототипа требуется примерно 2-4 месяца. База знаний исследовательского прототипа обычно содержит небольшое число исполняемых утверждений;

действующий прототип надежно решает любые задачи проблемной области, но при решении сложных задач может потребовать чрезмерно много времени и (или) памяти. Доведение системы от начала разработки до стадии действующего прототипа требует примерно 6 - 9 месяцев, при этом количество исполняемых утвер-

ждений в БЗ увеличивается по сравнению с исследовательским прототипом;

промышленная система обеспечивает высокое качество решения всех задач при минимуме времени и памяти. Обычно процесс преобразования действующего прототипа в промышленную систему состоит в расширении базы знаний и ее тщательной отладке. Доведение ЭС от начала разработки до стадии промышленной системы с применением развитых инструментальных средств требует не менее 12 - 18 месяцев;

коммерческая система пригодна не только для использования разработчиком, но и для продажи различным потребителям, доведение системы до коммерческой стадии требует примерно от 2 до 5 лет. Приведенные здесь сроки справедливы для ЭС средней сложности.

3. Масштаб ЭС

Характеризует сложность решаемых задач и связан с типом используемой ЭВМ. По этому признаку различают:

малые ЭС предназначены для первичного обучения и исследования возможности применения технологии ЭС для рассматриваемого класса задач. Системы такого типа могут быть реализованы на персональных компьютерах;

средние ЭС охватывают весь спектр необходимых приложений и обычно интегрированы с БД, электронными таблицами т.д. Системы такого масштаба чаще всего реализуются на рабочих станциях;

большие ЭС имеют доступ к высокомощным БД и реализуются на рабочих станциях или на специализированных компьютерах;

символьные ЭС создаются с исследовательскими целями и реализуются на специализированных компьютерах, ориентированных на обработку символьных данных.

4. Тип проблемной среды

Это понятие включает описание предметной области (множество сущностей, описывающих множество объектов, их характеристик и отношений между объектами) и решаемых в ней задач. Иначе говоря, проблемная среда включает структуры данных и решаемые с ними задачи, представленные в виде исполняемых утверждений (правил, процедур, формул и т. п.). В связи с этим проблемная среда определяется характеристиками соответствующей предметной области и характеристиками типов решаемых в ней задач.

5. Предметная область

Тип предметной области:

статический – входные данные не изменяются за время сеанса работы приложения, значения других (не входных) данных изменяются только самой экспертной системой;

динамический – входные данные, поступающие из внешних источников, изменяются во времени, значения других данных изменяются ЭС или подсистемой моделирования внешнего окружения.

Способ описания сущностей предметной области:

совокупность атрибутов и их значений (фиксированный состав сущностей);

совокупность классов (объектов) и их экземпляров (изменяемый состав сущностей).

Способ организации сущностей в БЗ:

неструктурированная БЗ;

структурирование сущностей в БЗ по различным иерархиям («частное - общее», «часть - целое», «род - вид»), что обеспечивает наследование свойств сущностей.

Структурирование БЗ способствует:

ограничению круга сущностей, которые должны рассматриваться механизмом вывода, и сокращению количества перебираемых вариантов в процессе выбора решения;

обеспечению наследования свойств сущностей, т.е. передачи свойств вышерасположенных в иерархии сущностей нижерасположенным, что значительно упрощает процесс приобретения и использования знаний.

6. Классы решаемых задач

Тип решаемых задач:

задачи анализа или синтеза;

статические или динамические задачи.

Общность исполняемых утверждений:

частные исполняемые утверждения, содержащие ссылки на конкретные сущности (объекты);

общие исполняемые утверждения, относящиеся к любым сущностям заданного типа (вне зависимости от их числа и имени). Использование общих утверждений позволяет значительно лаконичнее представлять знания.

7. Тип решаемой задачи

Интерпретация данных – процесс определения смысла данных, результаты которого должны быть согласованными и корректными. Экспертные системы, как правило, проводят многовариантный анализ данных;

диагностика - процесс соотнесения объекта с некоторым классом объектов и/или обнаружение неисправностей в системе (отклонений параметров системы от нормативных значений);

мониторинг – непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы;

проектирование – создание ранее не существовавшего объекта и подготовка спецификаций на создание объектов с заранее определенными свойствами. Степень новизны может быть разной и определяется видом знаний, заложенных в ЭС, и методами их обработки. Для организации эффективного проектирования и реинжиниринга требуется формировать не только сами проектные решения, но и мотивы их принятия. ЭС, решающие задачи проектирования, реализуют процедуры вывода решения и объяснения полученных результатов;

прогнозирование – предсказание последствий некоторых событий или явлений на основе анализа имеющихся данных. Прогнозирующие ЭС логически выводят вероятные следствия из заданных ситуаций. В прогнозирующих ЭС в большинстве случаев используются динамические модели, в которых значения параметров «подгоняются» под заданную ситуацию. Выводимые из этих моделей следствия составляют основу для прогнозов с вероятностными оценками;

планирование – построение планов действий объектов, способных выполнять некоторые функции. Работа ЭС по планированию основана на моделях поведения реальных объектов, которые позволяют проводить логический вывод последствий планируемой деятельности;

обучение – использование компьютера для обучения какой-либо дисциплине или предмету. Экспертные системы обучения выполняют такие функции, как диагностика ошибок, подсказывание правильных решений; аккумулирование знаний о гипотетическом «ученике» и его характерных ошибках; диагностирование слабости в познаниях обучаемых и нахождение соответствующих средств для их ликвидации. Системы обучения способны планировать акт общения с учеником в зависимости от успехов ученика для передачи необходимых знаний;

управление – функция организованной системы, поддерживающая определенный режим ее деятельности. Экспертные системы данного типа предназначены для управления поведением сложных систем в соответствии с заданными спецификациями;

поддержка принятия решений – совокупность процедур, обеспечивающая лицо, принимающее решения, необходимой информацией

и рекомендациями, облегчающими процесс принятия решения. Такого рода ЭС оказывают помощь специалистам в выборе и/или генерации наиболее рациональной альтернативы из множества возможных при принятии ответственных решений.

Задачи интерпретации данных, диагностики, поддержки принятия решений относятся к задачам анализа, задачи проектирования, планирования и управления – к задачам синтеза. К комбинированному типу задач относятся обучение, мониторинг и прогнозирование.

8. Способ представления знаний

Логический вывод;
семантические сети;
фреймы;
правила продукции;
нейронные сети;
генетические алгоритмы;
критериальные методы и многомерное планирование;
стохастические модели.

9. Механизмы вывода и моделирования

Структура процесса получения решения:

построение дерева вывода на основе обучающей выборки (индуктивные методы приобретения знаний) и выбор маршрута на дереве вывода в режиме решения задачи;

компиляция сети вывода из специфических правил в режиме приобретения знаний и поиск решения на сети вывода в режиме решения задачи;

генерация сети вывода и поиск решения в режиме решения задачи, при этом генерация сети вывода осуществляется в ходе выполнения операции сопоставления, определяющей пары «правило – совокупность данных», на которых условия этого правила удовлетворяются;

в режиме решения задач ЭС осуществляет выработку правдоподобных предположений (при отсутствии достаточной информации для решения); выполнение рассуждений по обоснованию (опровержению) предположений; генерацию альтернативных сетей вывода; поиск решения в сетях вывода.

Поиск (выбор) решения:

направление поиска – от данных к цели, от целей к данным двунаправленный поиск;

порядок перебора вершин в сети вывода – «поиск в ширину» .

Процесс генерации предположений и сети вывода:

режим – генерация в режиме приобретения знаний, генерация в режиме решения задачи;

полнота генерируемой сети вывода – операция сопоставления применяется ко всем правилам и ко всем типам указанных правилах сущностей в каждом цикле работы механизма вывода (обеспечивается полнота генерируемой сети); используются различные средства для сокращения количества правил и (или) сущностей, участвующих в операции сопоставления.

10. Средства приобретения знаний

Уровень языка приобретения знаний:

формальный язык;

ограниченный естественный язык;

язык пиктограмм и изображений;

ЕЯ и язык изображений.

Тип приобретаемых знаний:

данные в виде таблиц, содержащих значения входных и выходных атрибутов, по которым индуктивными методами строится дерево вывода;

специализированные правила;

общие и специализированные правила.

Тип приобретаемых данных:

атрибуты и значения;

объекты;

классы структурированных объектов и их экземпляры, получающие значения атрибутов путем наследования.

В ряде работ приводится обобщенная классификация ЭС, соответствующая рисунку 1.2. В основу классификации положена сложность решаемых задач, в соответствие с чем выделяют следующие классы ЭС.

По способу формирования решения ЭС разделяются на два класса: *аналитические* и *синтетические*.

По способу учета временного признака ЭС могут быть *статическими* или *динамическими*.

По видам используемых данных и знаний ЭС 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 ЭС 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 четыре основных класса ЭС (рисунок 2.2).

Рисунок 2.2. Обобщенная классификация ЭС

Классифицирующие ЭС. 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 недос-

тающие 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, диагностика, коррекция.

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 характерна интеграция в базе знаний нескольких разнородных источников знаний, обменивающихся между собой получаемыми результатами на динамической основе.

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 заключений в зависимости от типа решаемой проблемы;

обработка больших массивов данных, содержащихся в базе данных;

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, прогнозирование, диспетчеризация – распределение работ во времени, планирование, мониторинг, управление.

2.2. Архитектура статических и динамических экспертных систем

Типичная статическая ЭС состоит из следующих основных компонентов (рисунок 2.3):

подсистемы логического вывода (интерпретатора, машины вывода);

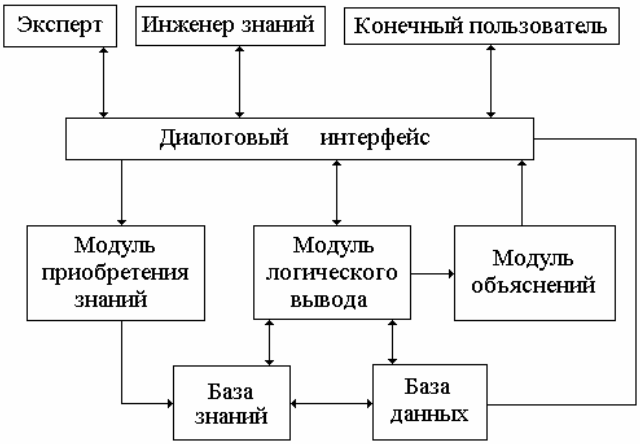


Рисунок 2.3. Статическая ЭС

базы знаний (БЗ);
базы данных;
подсистемы приобретения знаний;
подсистемы объяснения знаний;
диалогового компонента (интерфейса пользователя).

Ядро экспертной системы – база знаний и машина вывода. Последнюю считают аналогом СУБД и иногда называют Системой управления базой знаний (СУБЗ)

База знаний – важная компонента экспертной системы, она предназначена для хранения долгосрочных данных, описывающих рассматриваемую предметную область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области. В качестве предметной области выбирается узкая (специальная) прикладная область. Далее для создания ЭС в выбранной области собираются факты и правила, которые помещаются в базу знаний вместе с механизмами вывода и упрощения. В отличие от всех остальных компонент ЭС, база знаний – «переменная» часть системы, которая может пополняться и модифицироваться инженерами знаний и опыта использование ЭС, между консультациями (а в некоторых системах и в процессе консультации). Существует несколько способов представления знаний в ЭС, однако общим для всех них является то, что знания представлены в символьной форме (элементарными компонентами представления знаний являются тексты, списки и другие символьные структуры). Тем самым, в ЭС реализуется принцип символьной природы рассуждений, который заключается в том, что процесс рассуждения представляется как последовательность символьных преобразований. Существуют динамические и статические базы знаний. Динамическая база знаний изменяется со временем. Ее содержимое зависит и от состояния окружающей. Новые факты, добавляемые в базу знаний, являются результатом вывода, который состоит в применении правил к имеющимся фактам. В системах с монотонным выводом факты, хранимые в базе знаний, статичны, то есть не изменяются в процессе решения задачи. В системах с немонотонным выводом допускается изменение или удаление фактов из базы знаний. В качестве примера системы с немонотонным выводом можно привести ЭС, предназначенную для составления перспективного плана капиталовложения компании. В такой системе по вашему желанию могут быть изменены даже те данные, которые после вывода уже вызвали срабатывание каких-либо пра-

вил. Иными словами, имеется возможность модифицировать значения атрибутов в составе фактов, находящихся в рабочей памяти. Изменение фактов в свою очередь приводит к необходимости удаления из базы знаний заключений, полученных с помощью упомянутых правил. Тем самым вывод выполняется повторно для того, чтобы пересмотреть те решения, которые были получены на основе подвергшихся изменению фактов.

База данных (рабочая память) предназначена для хранения исходных данных решаемой задачи.

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 сообщения на носитель.

Экспертные системы имеют две категории пользователей и два отдельных «входа», соответствующих различным целям взаимодействия пользователей с ЭС:

обычный пользователь (эксперт), которому требуется консультация ЭС – диалоговый сеанс работы с ней, в процессе которой она решает некоторую экспертную задачу. Диалог с ЭС осуществляется через диалоговый процессор – специальную компоненту ЭС. Существуют две основные формы диалога с ЭС: диалог на ограниченном подмножестве естественного языка (с использованием словаря-меню (при котором на каждом шаге диалога система предлагает выбор профессионального лексикона экспертов) и диалог на основе из нескольких возможных действий);

экспертная группа инженерии знаний, состоящая из экспертов в предметной области и инженеров знаний. В функции этой группы входит заполнение базы знаний, осуществляемое с помощью специализированной диалоговой компоненты ЭС – подсистемы приобретения знаний, которая позволяет частично автоматизировать этот процесс.

Машина вывода. Основу ЭС составляет подсистема логического вывода, которая использует информацию из базы знаний (БЗ), генерирует рекомендации по решению искомой задачи. Чаще всего для представления знаний в ЭС используются системы продукций и семантические сети. Допустим, БЗ состоит из фактов и правил (если <посылка> то <заключение>). Если ЭС определяет, что посылка верна, то правило признается подходящим для данной консультации, и оно запускается в действие. Запуск правила означает принятие заключения данного правила в качестве составной части процесса кон-

сультации. Цель ЭС – вывести некоторый заданный факт, который называется целевым утверждением (то есть в результате применения правил добиться того, чтобы этот факт был включен в рабочее множество), либо опровергнуть этот факт (то есть убедиться, что его вывести невозможно, следовательно, при данном уровне знаний системы он является ложным). Целевое утверждение может быть либо «заложено» заранее в базу знаний системы, либо извлекается системой из диалога с пользователем. Работа системы представляет собой последовательность шагов, на каждом из которых из базы выбирается некоторое правило, которое применяется к текущему содержимому рабочего множества. Цикл заканчивается, когда выведено либо опровергнуто целевое утверждение. Цикл работы экспертной системы иначе называется логическим выводом. Логический вывод может происходить многими способами, из которых наиболее распространенные – прямой порядок вывода и обратный порядок вывода. Прямой порядок вывода – от фактов, которые находятся в рабочем множестве, к заключению. Если такое заключение удается найти, то оно заносится в рабочее множество. Прямой вывод часто называют выводом, управляемым данными.

Механизм объяснения предоставляет пользователю обоснование хода решения задачи. Он поясняет, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

Механизм приобретения знаний в простейшем случае позволяет вводить единицы знаний в БЗ и проводить их синтаксический и семантический контроль. Более развитый механизм приобретения знаний позволяет извлекать знания путем специального сценария интервьюирования экспертов.

ЭС статического типа используют в тех приложениях, где можно не учитывать изменения окружающего мира, происходящие за время решения задачи. Они нашли применение в широком классе приложений.

В динамических экспертных системах представляются знания, меняющиеся во времени. Кроме того, эти системы должны вырабатывать решения в реальном времени и моделировать различные состояния окружающего мира. Поэтому они снабжаются блоками моделирования внешнего мира и сопряжения с внешним миром (рис.2.4).

Инженер по знаниям (или инженер-когнитолог) – специалист, извлекающий знания для проектирования и заполнения базы знаний. Он же может быть разработчиком экспертной системы.

Подсистема приобретения знаний предназначена для добавления в базу знаний новых правил и модификации имеющихся. В ее задачу входит приведение правила к виду, позволяющему подсистеме вывода применять это правило в процессе работы. В более сложных системах предусмотрены еще и средства для проверки вводимых или модифицируемых правил на непротиворечивость с имеющимися правилами.

Экспертная система работает в двух режимах: режиме приобретения знаний и в режиме решения задачи (называемом также режимом консультации или режимом использования ЭС).

В *режиме приобретения знаний* общение с ЭС осуществляет (через посредничество инженера по знаниям) эксперт. В этом режиме эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из проблемной области. Эксперт описывает проблемную область в виде совокупности данных и правил. Данные определяют объекты, их характеристики и значения, существующие в области экспертизы. Правила определяют способы манипулирования с данными, характерные для рассматриваемой области.

Отметим, что режиму приобретения знаний в традиционном подходе к разработке программ соответствуют этапы алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом, в отличие от традиционного подхода в случае ЭС разработку программ осуществляет не программист, а эксперт (с помощью ЭС), не владеющий программированием.

В *режиме консультации* общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС, пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к ЭС за результатом, не умея получить его сам), или быть специалистом (в этом случае пользователь может сам получить результат, но он обращается к ЭС с целью либо ускорить процесс получения результата, либо возложить на ЭС рутинную работу). В режиме консультации данные о задаче пользователя после обработки их диалоговым компонентом поступают в рабочую память. Решатель на основе входных данных из рабочей памяти, общих данных о проблемной области и правил из БЗ формирует решение задачи. ЭС при решении задачи не только исполняет предписанную последовательность операции, но и предварительно формирует ее. Если реакция системы не понятна пользователю, то он может потребовать объяснения.

В архитектуру *динамической ЭС,* по сравнению со статической ЭС, вводятся два компонента: подсистема моделирования внешнего мира и подсистема связи с внешним окружением (рисунок 2.4).

Рисунок 2.4. Динамическая ЭС

Последняя осуществляет связи с внешним миром через систему датчиков и контроллеров. Кроме того, в динамической ЭС традиционные компоненты статической ЭС (база знаний и машина вывода) претерпевают существенные изменения, чтобы отразить временную логику происходящих в реальном мире событий. Динамические ЭС решают задачи, в которых:

данные изменяются во времени, поступают от внешних источников и их необходимо хранить и анализировать;

выполняются одновременно временные рассуждения о нескольких асинхронных задачах;

обеспечивается механизм рассуждений при ограниченных ресурсах;

обеспечивается «предсказуемость» поведения системы, т.е. запущенная задача выполняется в строгом соответствии с временными ограничениями;

моделируется окружающий мир, в котором отражаются различные состояния;

протоколируются действия;

обеспечивается достаточно эффективно и удобно наполнение БЗ;

обеспечивается настройка системы на решаемые задачи;

реализуется пользовательский интерфейс для различных категорий;

осуществляется защита информации по категориям пользователей.

Подчеркнем, что структура ЭС, представленная на рис. 2.3 и 2.4, отражает только компоненты (функции), и многое остается «за кадром».

Опыт применения динамических ЭС в показывает следующие преимущества их использования в АСУ предприятия.

Динамические ЭС являются мощным средством поддержки решений на крупных предприятиях со сложной структурой управления.

Подход к оптимизации осуществляется не только за счет математических расчетов, но и при помощи знаний экспертов, заложенных в базу знаний.

Динамические ЭС легко интегрируются в существующие АСУ предприятия.

Открытость современных инструментальных средств дает возможность масштабирования ЭС внутри предприятия.

Благодаря встроенным возможностям современных динамических экспертных систем становится возможным их применение как систем проектирования, систем обнаружения неисправностей, а также для диагностики, планирования и оптимизации производства.

2.3. Этапы разработки экспертных систем

В настоящее время сложилась определенная технология разработки ЭС, которая включает следующие шесть этапов: идентификация, концептуализация, формализация, выполнение, тестирование и опытная эксплуатация (рисунок 2.5.)

Этап идентификации

Этап идентификации связан, прежде всего, с осмыслением тех задач, которые предстоит решить будущей ЭС, и формированием требований к ней. Результатом данного этапа является ответ на вопрос,

что надо сделать и какие ресурсы необходимо задействовать (идентификация задачи, определение участников процесса проектирования и их роли, выявление ресурсов и целей).

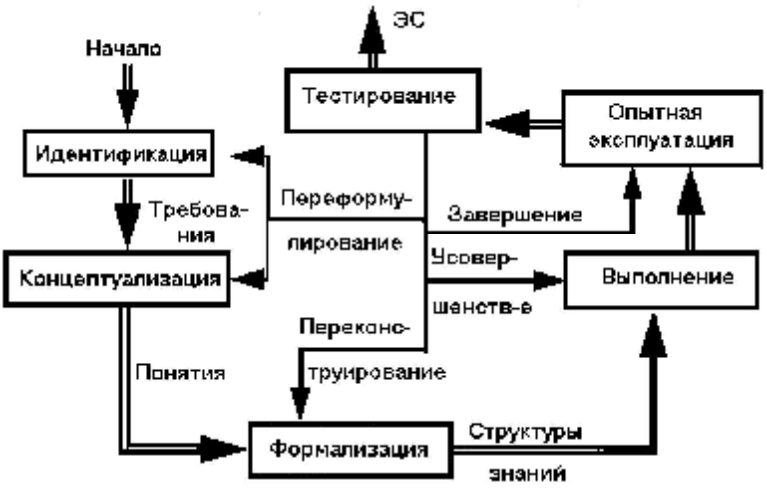


Рисунок 2.5. Этапы разработки ЭС

Обычно в разработке ЭС участвуют не менее трех-четырех человек — один эксперт, один или два инженера по знаниям и один программист, привлекаемый для модификации и согласования инструментальных средств. Кроме того, для сложных систем считается целесообразным привлекать к основному циклу разработки несколько экспертов. К процессу разработки ЭС могут по мере необходимости привлекаться и другие участники.

Идентификация задачи заключается в составлении неформального (вербального) описания, в котором указываются: общие характеристики задачи; подзадачи, выделяемые внутри данной задачи; ключевые понятия (объекты), их входные (выходные) данные; предположительный вид решения, а также знания, относящиеся к решаемой задаче.

В процессе идентификации задачи инженер по знаниям и эксперт работают в тесном контакте. Начальное неформальное описание задачи экспертом используется инженером по знаниям для уточнения терминов и ключевых понятий. Эксперт корректирует описание задачи, объясняет, как решать ее и какие рассуждения лежат в основе

того или иного решения. После нескольких циклов, уточняющих описание, эксперт и инженер по знаниям получают окончательное неформальное описание задачи.

При проектировании ЭС типичными ресурсами являются источники знаний, время разработки, вычислительные средства и объем финансирования. Для эксперта источниками знаний служат его предшествующий опыт по решению задачи, книги, известные примеры решения задач, а для инженера по знаниям – опыт в решении аналогичных задач, методы представления знаний и манипулирования ими, программные инструментальные средства.

При идентификации целей важно отличать цели, ради которых создается ЭС, от задач, которые она должна решать. Примерами возможных целей являются: формализация неформальных знаний экспертов; улучшение качества решений, принимаемых экспертом; автоматизация рутинных аспектов работы эксперта (пользователя); тиражирование знаний эксперта.

Этап концептуализации

На данном этапе проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области (ПО), включающей основные концепты и отношения. На этапе концептуализации определяются следующие особенности задачи: типы доступных данных; исходные и выводимые данные, подзадачи общей задачи; используемые стратегии и гипотезы; виды взаимосвязей между объектами ПО, типы используемых отношений (иерархия, причина – следствие, часть – целое и т.п.); процессы, используемые в ходе решения; состав знаний, используемых при решении задачи; типы ограничений, накладываемых на процессы, используемые в ходе решения; состав знаний, используемых для обоснования решений.

Существует два подхода к процессу построения модели предметной области, которая является целью разработчиков ЭС на этапе концептуализации.

Признаковый или *атрибутивный подход* предполагает наличие полученной от экспертов информации в виде троек объект – атрибут – значение атрибута, а также наличие обучающей информации. Этот подход развивается в рамках направления, получившего название формирование знаний или «машинное обучение».

Второй подход, называемый *структурным* (или *когнитивным*), осуществляется путем выделения элементов предметной области, их взаимосвязей и семантических отношений.

Для атрибутивного подхода характерно наличие наиболее полной информации о предметной области: об объектах, их атрибутах и о значениях атрибутов. Кроме того, существенным моментом является использование дополнительной обучающей информации, которая задается группированием объектов в классы по тому или иному содержательному критерию. Тройки объект – атрибут – значение атрибута могут быть получены с помощью так называемого метода реклассификации, который основан на предположении что задача является объектно-ориентированной и объекты задачи хорошо известны эксперту. Идея метода состоит в том, что конструируются правила (комбинации значений атрибутов), позволяющие отличить один объект от другого. Обучающая информация может быть задана на основании прецедентов правильных экспертных заключений, например, с помощью метода извлечения знаний, получившего название «анализ протоколов мыслей вслух».

При наличии обучающей информации для формирования модели предметной области на этапе концептуализации можно использовать весь арсенал методов, развиваемых в рамках задачи распознавания образов.

Структурный подход к построению модели предметной области предполагает выделение следующих когнитивных элементов знаний: понятия, взаимосвязи, метапонятия, семантические отношения.

Выделяемые понятия предметной области должны образовывать систему, под которой понимается совокупность понятий, обладающая следующими свойствами: уникальностью (отсутствием избыточности); полнотой (достаточно полным описанием различных процессов, фактов, явлений и т.д. предметной области); достоверностью (валидностью – соответствием выделенных единиц смысловой информации их реальным наименованиям) и непротиворечивостью (отсутствием омонимии).

При построении системы понятий с помощью «*метода локального представления*» эксперта просят разбить задачу на подзадачи для перечисления целевых состояний и описания общих категорий цели. Далее для каждого разбиения (локального представления) эксперт формулирует информационные факты и дает им четкое наименование (название). Считается, что для успешного решения задачи построения модели

предметной области число таких информационных фактов в каждом локальном представлении, которыми человек способен одновременно манипулировать, должно быть примерно равно семи.

«*Метод вычисления коэффициента использования*» основан на следующей гипотезе. Элемент данных (или информационный факт) может являться понятием, если:

он используется в большом числе подзадач;

используется с большим числом других элементов данных;

редко используется совместно с другими элементами данных по сравнению с общим числом его использования во всех подзадачах (это и есть коэффициент использования).

Полученные значения могут служить критерием для классификации всех элементов данных и, таким образом, для формирования системы понятий.

«*Метод формирования перечня понятий*» заключается в том, что экспертам (желательно, чтобы их было больше двух) дается задание составить список понятий, относящихся к исследуемой предметной области. Понятия, выделенные всеми экспертами, включаются в систему понятий, остальные подлежат обсуждению.

«*Ролевой метод*» состоит в том, что эксперту дается задание обучить инженера по знаниям решению некоторых задач предметной области. Таким образом, эксперт играет роль учителя, а инженер по знаниям – роль ученика. Процесс обучения записывается на магнитофон. Затем третий участник прослушивает магнитофонную ленту и выписывает на бумаге все понятия, употребленные учителем или учеником.

При использовании метода «*составления списка элементарных действий*» эксперту дается задание составить такой список при решении задачи в произвольном порядке.

В методе «*составление оглавления учебника*» эксперту предлагается представить ситуацию, в которой его попросили написать учебник. Необходимо составить на бумаге перечень предполагаемых глав, разделов, параграфов, пунктов и подпунктов книги.

«*Текстологический метод*» формирования системы понятий заключается в том, что эксперту дается задание выписать из руководств (книг по специальности) некоторые элементы, представляющие собой единицы смысловой информации.

Группа методов установления взаимосвязей предполагает установление семантической близости между отдельными понятиями. В основе установления взаимосвязей лежит психологический эффект

«*свободных ассоциаций*», а также фундаментальная категория близости объектов или концептов.

Эффект свободных ассоциаций заключается в следующем. Испытуемого просят отвечать на заданное слово первым пришедшим на ум словом. Как правило, реакция большинства испытуемых (если слова не были слишком необычными) оказываются одинаковой. Количество переходов в цепочке может служить мерой «смыслового расстояния» между двумя понятиями. Многочисленные опыты подтверждают гипотезу, что для двух любых слов (понятий) существует ассоциативная цепочка, состоящая не более чем из семи слов.

«*Метод свободных ассоциаций*» основан на психологическом эффекте, описанном выше. Эксперту предъявляется понятие с просьбой назвать как можно быстрее первое пришедшее на ум понятие из сформированной ранее системы понятий. Далее производится анализ полученной информации.

В методе «*сортировка карточек*» исходным материалом служат выписанные на карточки понятия. Применяются два варианта метода. В первом эксперту задаются некоторые глобальные критерии предметной области, которыми он должен руководствоваться при раскладывании карточек на группы. Во втором случае, когда сформулировать глобальные критерии невозможно эксперту дается задание разложить карточки на группы в соответствии с интуитивным пониманием семантической близости предъявляемых понятий.

«*Метод обнаружения регулярностей*» основан на гипотезе о том, что элементы цепочки понятия, которые человек вспоминает с определенной регулярностью, имеют тесную ассоциативную взаимосвязь. Для эксперимента произвольным образом отбирается 20 понятий. Эксперту предъявляется одно из числа отобранных. Процедура повторяется до 20 раз, причем каждый раз начальные концепты должны быть разными. Затем инженер по знаниям анализирует полученные цепочки с целью нахождения постоянно повторяющихся понятий (регулярностей). Внутри выделенных таким образом группировок устанавливаются ассоциативные взаимосвязи.

Кроме рассмотренных выше неформальных методов для установления взаимосвязей между отдельными понятиями применяются также формальные методы. Сюда в первую очередь относятся методы *семантического дифференциала* и *репертуарных решеток*.

Выделенные понятия предметной области и установленные между ними взаимосвязи служат основанием для дальнейшего построения

системы метапонятий — осмысленных в контексте изучаемой предметной области системы группировок понятий. Для определения этих группировок применяют как неформальные, так и формальные методы.

Интерпретация, как правило, легче дается эксперту, если группировки получены неформальными методами. В этом случае выделенные классы более понятны эксперту. Причем в некоторых предметных областях совсем не обязательно устанавливать взаимосвязи между понятиями, так как метапонятия, образно говоря, «лежат на поверхности».

Последним этапом построения модели предметной области при концептуальном анализе является установление семантических отношений между выделенными понятиями и метапонятиями. Установить семантические отношения – это значит определить специфику взаимосвязи, полученной в результате применения тех или иных методов. Для этого необходимо каждую зафиксированную взаимосвязь осмыслить и отнести ее к тому или иному типу отношений.

Существует около 200 базовых отношений, например, «часть – целое», «род – вид», «причина – следствие», пространственные, временные и другие отношения. Для каждой предметной области помимо общих базовых отношений могут существовать и уникальные отношения.

«*Прямой метод*» установления семантических отношений основан на непосредственном осмыслении каждой взаимосвязи. В том случае, когда эксперт затрудняется дать интерпретацию выделенной взаимосвязи, ему предлагается следующая процедура. Формируются тройки: понятие 1 – связь – понятие 2. Рядом с каждой тройкой записывается короткое предложение или фраза, построенное так, чтобы понятие 1 и понятие 2 входили бы в это предложение. В качестве связок используются только содержательные отношения и не применяются неопределенные связки типа «похож на» или «связан с…».

Для «*косвенного метода*» необязательно иметь взаимосвязи, а достаточно лишь наличие системы понятий. Формулируется некоторый критерий, для которого из системы понятий выбирается определенная совокупность концептов. Эта совокупность предъявляется эксперту с просьбой дать вербальное описание сформулированного критерия. Концепты предъявляются эксперту все сразу (желательно на карточках). В случае затруднений эксперта прибегают к разбиению отобранных концептов на группы с помощью более мелких критериев. Исходное количество концептов может быть произвольным, но после разбиения на группы в каждой из таких групп должно быть не более десяти концептов. После того как составлены описания по всем группам, эксперту предлагают объединить эти описания в одно.

Следующий шаг в косвенном методе установления семантических отношений – это анализ текста, составленного экспертом. Концепты заменяют цифрами (это может быть исходная нумерация), а связки оставляют. Тем самым строится некоторый граф, вершинами которого служат концепты, а дугами – связки (например, «ввиду», «приводит к», «выражаясь с одной стороны», «обусловливая», «сочетаясь», «определяет», «вплоть до» и т.д.) Этот метод позволяет устанавливать не только базовые отношения, но и отношения, специфические для конкретной предметной области.

Рассмотренные выше методы формирования системы понятий и метапонятий, установления взаимосвязей и семантических отношений в разных сочетаниях применяются на этапе концептуализации при построении модели предметной области.

Этап формализации

Теперь все ключевые понятия и отношения выражаются на некотором формальном языке, который либо выбирается из числа уже существующих, либо создается заново. Другими словами, на данном этапе определяются состав средств и способы представления декларативных и процедурных знаний, осуществляется это представление и в итоге формируется описание решения задачи ЭС на предложенном (инженером по знаниям) формальном языке.

Выходом этапа формализации является описание того, как рассматриваемая задача может быть представлена в выбранном или разработанном формализме. Сюда относится указание способов представления знаний (фреймы, сценарии, семантические сети и т.д.) и определение способов манипулирования этими знаниями (логический вывод, аналитическая модель, статистическая модель и др.) и интерпретации знаний.

Этап выполнения

Цель этого этапа – создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний.

Главное в создании прототипа заключается в том, чтобы этот прототип обеспечил проверку адекватности идей, методов и способов представления знаний решаемым задачам. Создание первого прототипа должно подтвердить, что выбранные методы решений и спосо-

бы представления пригодны для успешного решения, по крайней мере, ряда задач из актуальной предметной области, а также продемонстрировать тенденцию к получению высококачественных и эффективных решений для всех задач предметной области по мере увеличения объема знаний.

После разработки первого прототипа ЭС-1 круг предлагаемых для решения задач расширяется, и собираются пожелания и замечания, которые должны быть учтены в очередной версии системы ЭС-2. Осуществляется развитие ЭС-1 путем добавления дружественного интерфейса, средств для исследования базы знаний и цепочек выводов, генерируемых системой, а также средств для сбора замечаний пользователей и средств хранения библиотеки задач, решенных системой.

Выполнение экспериментов с расширенной версией ЭС-1, анализ пожеланий и замечаний служат отправной точкой для создания второго прототипа ЭС-2. Процесс разработки ЭС-2 итеративный. Он может продолжаться от нескольких месяцев до нескольких лет в зависимости от сложности предметной области, гибкости выбранного представления знаний и степени соответствия управляющего механизма решаемым задачам (возможно, потребуется разработка ЭС-3 и т.д.). При разработке ЭС-2, кроме перечисленных задач, решаются следующие:

анализ функционирования системы при значительном расширении базы знаний;

исследование возможностей системы в решении более широкого круга задач и принятие мер для обеспечения таких возможностей;

анализ мнений пользователей о функционировании ЭС;

разработка системы ввода-вывода, осуществляющей анализ или синтез предложений ограниченного естественного языка, позволяющей взаимодействовать с ЭС-2 в форме, близкой к форме стандартных учебников для данной области.

Если ЭС-2 успешно прошла этап тестирования, то она может классифицироваться как промышленная экспертная система.

Этап тестирования

В ходе данного этапа производится оценка выбранного способа представления знаний в ЭС в целом. Для этого инженер по знаниям подбирает примеры, обеспечивающие проверку всех возможностей разработанной ЭС.

Различают следующие источники неудач в работе системы: тестовые примеры, ввод-вывод, правила вывода, управляющие стратегии.

Показательные тестовые примеры являются наиболее очевидной причиной неудачной работы ЭС. В худшем случае тестовые примеры могут оказаться вообще вне предметной области, на которую рассчитана ЭС, однако чаще множество тестовых примеров оказывается слишком однородным и не охватывает всю предметную область. Поэтому при подготовке тестовых примеров следует классифицировать их по подпроблемам предметной области, выделяя стандартные случаи, определяя границы трудных ситуаций и т.п.

Ввод-вывод характеризуется данными, приобретенными в ходе диалога с экспертом, и заключениями, предъявленными ЭС в ходе объяснений. Методы приобретения данных могут не давать требуемых результатов, так как, например, задавались неправильные вопросы или собрана не вся необходимая информация. Кроме того, вопросы системы могут быть трудными для понимания, многозначными и не соответствующими знаниям пользователя. Ошибки при вводе могут возникать также из-за неудобного для пользователя входного языка. В ряде приложения для пользователя удобен ввод не только в печатной, но и в графической или звуковой форме.

Выходные сообщения (заключения) системы могут оказаться непонятны пользователю (эксперту) по разным причинам. Например, их может быть слишком много или, наоборот, слишком мало. Также причиной ошибок может являться неудачная организация, упорядоченность заключений или неподходящий пользователю уровень абстракций с непонятной ему лексикой.

Наиболее распространенный источник ошибок в рассуждениях касается правил вывода. Важная причина здесь часто кроется в отсутствии учета взаимозависимости сформированных правил. Другая причина заключается в ошибочности, противоречивости и неполноте используемых правил. Если неверна посылка правила, то это может привести к употреблению правила в неподходящем контексте. Если ошибочно действие правила, то трудно предсказать конечный результат. Правило может быть ошибочно, если при корректности его условия и действия нарушено соответствие между ними.

Нередко к ошибкам в работе ЭС приводят применяемые управляющие стратегии. Изменение стратегии бывает необходимо, например, если ЭС анализирует сущности в порядке, отличном от «естественного» для эксперта. Последовательность, в которой данные рассматриваются ЭС, не только влияет на эффективность работы системы, но и может приводить к изменению конечного результата.

Так, рассмотрение правила А до правила В способно привести к тому, что правило В всегда будет игнорироваться системой. Изменение стратегии бывает также необходимо и в случае неэффективной работы ЭС. Кроме того, недостатки в управляющих стратегиях могут привести к чрезмерно сложным заключениям и объяснениям ЭС.

Критерии оценки ЭС зависят от точки зрения. Например, при тестировании ЭС-1 главным в оценке работы системы является полнота и безошибочность правил вывода. При тестировании промышленной системы превалирует точка зрения инженера по знаниям, которого в первую очередь интересует вопрос оптимизации представления и манипулирования знаниями. И, наконец, при тестировании ЭС после опытной эксплуатации оценка производится с точки зрения пользователя, заинтересованного в удобстве работы и получения практической пользы.

Этап опытной эксплуатации

На этом этапе проверяется пригодность ЭС для конечного пользователя. Пригодность ЭС для пользователя определяется в основном удобством работы с ней и ее полезностью. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). В свою очередь, удобство работы с ЭС подразумевает естественность взаимодействия с ней (общение в привычном, не утомляющем пользователя виде), гибкость ЭС (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях неопытного пользователях).

В ходе разработки ЭС почти всегда осуществляется ее модификация. Выделяют следующие виды модификации системы: переформулирование понятий и требований, переконструирование представления знаний в системе и усовершенствование прототипа.

Источниками знаний для экспертных систем могут быть учебники, справочники, материалы конкретных исследований в проблемной области, экономическая информация и т.п. Классическим источником знаний является эксперт в данной области, знания которого получает разработчик систем, выступающий в роли инженера знаний. Поэтому мы рассмотрим стратегии извлечения знаний из эксперта.

2.4. Стратегии извлечения знаний

При разработке экспертных систем можно выделить три стратегии получения знаний:

извлечение знаний;

формирование знаний;

приобретение знаний.

Извлечение знаний – это процедура взаимодействия инженера по знаниям с источником знаний. В результате процедуры становятся явными процесс рассуждений специалистов при принятии решений и структура их представлений о предметной области. Процесс извлечения знаний – это длительная и трудоемкая процедура, в которой инженеру по знаниям необходимо воссоздать модель предметной области, которой пользуются эксперты при принятии решений.

Существующая классификация методов извлечения знаний показана на рисунке 2.6.

Основной принцип классификации методов извлечения знаний связан с источником знаний. Коммуникативные методы охватывают все виды контактов с экспертом, а текстологические касаются методов извлечения знаний из документов и литературы. Разделение этих групп методов на верхнем уровне не означает их антагонистичности. обычно инженер по знаниям комбинирует эти методы.

Рисунок 2.6. Классификация методов извлечения знаний

В свою очередь коммуникативные методы делятся на активные и пассивные. Пассивные методы подразумевают, что ведущая роль в процедуре извлечения знаний передается эксперту, а инженер по знаниям только протоколирует рассуждения эксперта во время его реальной работы. В активных методах инициатива находится полностью в руках инженера по знаниям, который активно контактирует с экспертом.

Пассивные методы на первый взгляд достаточно просты, но на самом деле требуют от инженера по знаниям умения четко анализировать «поток сознания» эксперта и выявлять в нем значимые фрагменты знаний. Отсутствие обратной связи (пассивность инженера по знаниям) значительно ослабляет эффективность этих методов, чем и объясняется их обычно вспомогательная роль при активных методах.

Активные методы можно разделить на две группы в зависимости от числа экспертов, отдающих свои знания. Если их число больше одного, то целесообразно помимо серии индивидуальных контактов с каждым применять и методы групповых обсуждений предметной области. Такие групповые методы обычно активизируют мышление участников дискуссий и позволяют выявлять весьма нетривиальные аспекты их знаний. В свою очередь, индивидуальные методы на сегодняшний день остаются ведущими.

Пассивные методы

Согласно классификации (см. рисунок 2.5) к этой группе относятся:

наблюдения;

анализ протоколов «мыслей вслух»;

лекции.

Наблюдения

В процессе наблюдений инженер по знаниям находится непосредственно рядом с экспертом во время его профессиональной деятельности или имитации этой деятельности; При подготовке к сеансу извлечения эксперту необходимо объяснить цель наблюдений и попросить максимально комментировать свои действия.

Во время сеанса аналитик записывает все действия эксперта, его реплики и объяснения. Может быть сделана и видеозапись в реальном масштабе времени. Непременное условие этого метода – невмешательство аналитика в работу эксперта хотя бы на первых порах.

Существуют две разновидности проведения наблюдений:

наблюдение за реальным процессом;

наблюдение за имитацией процесса.

Обычно используют обе разновидности.

Сеансы наблюдений могут потребовать от инженера по знаниям:

овладения техникой стенографии для фиксации действий эксперта в реальном масштабе времени;

ознакомления с методиками хронометрирования для четкого структурирования производственного процесса по времени,

развития навыков «чтения по глазам», наблюдательности к жестам, мимике и другим невербальным компонентам общения;

серьезного предварительного знакомства с предметной областью, так как из-за отсутствия обратной связи иногда многое непонятно в действиях экспертов. Протоколы наблюдений после сеансов в ходе домашней работы тщательно расшифровываются, а затем обсуждаются с экспертом. Таким образом, наблюдения – один из наиболее распространенных методов извлечения знаний на начальных этапах разработки. Обычно он применяется не самостоятельно, а в совокупности с другими методами.

Анализ протоколов «мыслей вслух»

Протоколирование «мыслей вслух» отличается от наблюдений тем, что эксперта просят не просто прокомментировать свои действия и решения, но и объяснить, как это решение было найдено, т.е. продемонстрировать всю цепочку своих рассуждений. Во время рассуждения эксперта все его слова протоколируются инженером по знаниям: при этом полезно отмечать даже паузы и междометия. Вопрос об использовании для этой цели магнитофонов и диктофонов является дискуссионным, поскольку магнитофон иногда действует на эксперта парализующе, разрушая атмосферу доверительности, которая может и должна возникать при непосредственном общении.

Основной трудностью при протоколировании «мыслей вслух» является принципиальная сложность для любого человека объяснить, как он думает. При этом существуют экспериментальные психологические доказательства, что люди не всегда в состоянии достоверно описать мыслительные процессы. Кроме того, часть знаний, хранящихся в невербальной форме, вообще слабо коррелируют с их словесным описанием.

Расшифровка полученных протоколов производится инженером по знаниям самостоятельно с коррекциями на следующих сеансах извлечения знаний. Удачно проведенное протоколирование «мыслей вслух» является одним из наиболее эффективных методов извлечения, поскольку в нем эксперт может проявить себя максимально ярко, он ничем не скован, он как бы свободно парит в потоке своих

умозаключений и рассуждений. Для большого числа экспертов это самый приятный и лестный способ извлечения знаний.

Лекции

Лекция – самый старый способ передачи знаний. Лекторское искусство издревле высоко ценилось во всех областях науки и культуры. Но нас интересует не столько способность к подготовке и чтению лекций, сколько способность эту лекцию слушать, конспектировать и усваивать. Уже говорилось, что экспертов чаще всего не выбирают, и поэтому инженер по знаниям учить эксперта чтению лекций не сможет. Но если у эксперта опыт преподавателя, то можно воспользоваться таким концентрированным фрагментом знаний, как лекция.

В лекции эксперту предоставлено много степеней свободы для самовыражения; при этом необходимо сформулировать эксперту тему и задачу лекции. При такой постановке опытный лектор может заранее структурировать свои знания, ход рассуждении. От инженера по знаниям в этой ситуации требуется лишь грамотно законспектировать лекцию и в конце задать необходимые вопросы.

Метод извлечения знаний в форме лекций, как и все пассивные методы, используют в начале разработки как эффективный способ быстрого погружения инженера по знаниям в предметную область.

Активные методы

Активные индивидуальные методы

Активные индивидуальные методы извлечения знаний на сегодняшний день наиболее распространенные. В той или иной степени к ним прибегают при разработке практически любой экспертной системы.

К основным активным методам можно отнести:

анкетирование;

интервью;

свободный диалог;

игры с экспертом.

В этих методах активную функцию выполняет инженер по знаниям, который пишет сценарий и режиссирует сеансы извлечения знаний. Игры с экспертом существенно отличаются от других методов. Три оставшихся метода очень схожи между собой и отличаются лишь по степени свободы, которую может себе позволить инженер по знаниям при проведении сеансов извлечения знаний.

Анкетирование – наиболее жесткий метод, т.е. наиболее стандартизованный. Инженер по знаниям заранее составляет вопросник или анкету, размножает ее и использует для опроса нескольких экспертов. Это основное преимущество анкетирования.

Сама процедура может проводиться двумя способами:

аналитик вслух задает вопросы, и сам заполняет анкету по ответам эксперта;

эксперт самостоятельно заполняет анкету после предварительного инструктирования.

Выбор способа зависит от конкретных условий. Второй способ нам кажется предпочтительнее, так как у эксперта появляется неограниченное время на обдумывание ответов.

Вопросник (анкета) заслуживает отдельного разговора. Существует несколько общих рекомендаций при составлении анкет. Эти рекомендации универсальны, т.е. не зависят от предметной области.

Во-первых, анкета не должна быть монотонной и однообразной, т.е. вызывать скуку или усталость. Это достигается вариациями вопросов, сменой тематики, вставкой вопросов-шуток и игровых вопросов. Во-вторых, анкета должна быть приспособлена к языку экспертов. В-третьих, следует учитывать, что вопросы влияют друг на друга, и поэтому последовательность вопросов должна быть строго продумана. В-четвертых, желательно стремиться к оптимальной избыточности. Известно, что в анкете всегда много лишних вопросов, часть из них необходима – это так называемые контрольные вопросы, а другую часть нужно минимизировать. И, наконец, в-пятых, у анкеты должны быть «хорошие манеры», т.е. ее язык ясен, понятен, предельно вежлив. Методическим мастерством составления анкеты овладевают только на практике.

Под *интервью* понимаем специфическую форму общения инженера по знаниям и эксперта, в которой инженер по знаниям задает эксперту серию заранее подготовленных вопросов с целью извлечения знаний о предметной области.

Интервью очень близко к анкетированию, когда аналитик сам заполняет анкету, занося туда ответы эксперта. Основное отличие в том, что интервью позволяет аналитику опускать ряд вопросов в зависимости от ситуации, вставлять новые вопросы в анкету, изменять темп, разнообразить ситуацию общения.

Теперь подробнее о центральном звене активных индивидуальных методов – о вопросах. Инженеры по знаниям редко задумываются: умеют ли они задавать вопросы? В философии и математике эта проблема обсуждается с давних пор. Существует даже специальная ветвь математической логики – логика вопросов.

Укажем три основные характеристики вопросов, которые влияют на качество интервью:

стиль вопроса (понятность, лаконичность, терминология);

порядок вопросов (логическая последовательность и немонотонность);

уместность вопросов (этика, вежливость).

Вопрос в интервью – это не просто средство общения, но и способ передачи мыслей и позиции аналитика. Отсюда необходимость фиксировать в протоколах не только ответы, но и вопросы, предварительно отработав их форму и содержание.

Очевидно, что любой вопрос имеет смысл только в контексте. Поэтому вопросы может готовить инженер по знаниям, уже овладевший ключевым набором знаний.

Все вопросительные предложения можно разбить на два типа.

Вопросы с неопределенностью, относящиеся ко всему предложению (например: «Действительно Вы столкнулись с трудностями при внедрении «1С» ?).

Вопросы с неполной информацией (например: «При каких условиях «1С» Вас не устраивает?»), часто начинающиеся со слов «кто», что», «где», «когда» и т.д.

Открытый вопрос называет тему или предмет, оставляя экспертам полную свободу в том, что касается формы и содержания ответа (« Как Вы боретесь со срывами графика поставок?»).

В закрытом вопросе специалист предприятия выбирает ответ из набора предложенных. Например: «Укажите, пожалуйста, что Вы делаете при срыве графика поставок: а) наказываю виноватых, б) подключаю дополнительные ресурсы, в) обращаюсь к партнерам, г) докладываю начальству». Закрытые вопросы легче обрабатывать при последующем анализе, но они более опасны, так как «закрывают» ход рассуждений специалиста и «программируют» его ответ в определенном направлении. При составлении сценария интервью полезно чередовать открытые и закрытые вопросы, особенно тщательно продумывая закрытые, поскольку для их составления требуется определенная эрудиция в предметной области.

Личный вопрос касается непосредственно индивидуального опыта специалиста («Скажите, пожалуйста, Иван Данилович, в Вашей практике как Вы определяете себестоимость продукта?»). Личные вопросы обычно активизируют мышление специалиста, «играют» на его самолюбии; они всегда украшают интервью.

Безличный вопрос направлен на выявление наиболее распространенных и общепринятых закономерностей предметной области («Что влияет на себестоимость?»).

При составлении вопросов следует учитывать, что языковые способности специалиста, как правило, ограничены и вследствие скованности, замкнутости или робости он может не сразу высказать свое мнение и предоставить знания, которые от него требуются (даже если предположить, что он их четко для себя формулирует). Поэтому часто при «зажатости» сотрудников компании аналитик использует не прямые вопросы, которые непосредственно указывают на предмет или тему («Применяете ли Вы методы сетевого планирования?»), а косвенные, которые лишь неявно указывают на интересующий предмет («А что Вы думаете о целесообразности применения…»). Иногда приходится задавать несколько десятков косвенных вопросов вместо одного прямого.

Вербальные вопросы – это традиционные устные вопросы. Вопросы с использованием наглядного материала разнообразят интервью и снижают утомляемость интервьюируемого. В таких вопросах используют фотографии, рисунки и карточки. Например, специалисту предлагаются цветные картонные карточки, на которых выписаны документы, с которыми он работает. Затем аналитик просит разложить эти карточки в порядке убывания значимости.

Деление вопросов по функции на основные, зондирующие, контрольные связано с тем, что часто основные вопросы интервью, направленные на выявление знаний, не срабатывают, специалист по каким-то признакам уходит в сторону от вопроса, отвечает нечетко. Тогда аналитик использует зондирующие вопросы, которые направляют рассуждения специалиста в нужную сторону. Например, если не сработал основной вопрос («Кто имеет право подписи договора?»), аналитик начинает задавать зондирующие вопросы («Всегда ли процесс оформления договора визирует директор? А если его нет?» и т.д.).

Контрольные вопросы применяют для проверки достоверности и объективности информации, полученной в интервью ранее («Скажите, пожалуйста, а Ваши филиалы также считают себестоимость?»). Контрольные вопросы должны быть составлены хитро, чтобы не обидеть работников компании недоверием (для этого используют повторение вопросов в другой форме, уточнения, ссылки на другие источники).

И, наконец, о нейтральных и наводящих вопросах. В принципе интервьюеру рекомендуют быть беспристрастным, следовательно, и во-

просы его должны носить нейтральный характер, т.е. не должны указывать на отношение интервьюера к данной теме. Наводящие вопросы, напротив, заставляют респондента (в данном случае специалиста предприятия) прислушаться или даже принять во внимание позицию интервьюера. К примеру, нейтральный вопрос может звучать так: «Какие факторы Вы учитываете при начислении премии сотрудникам?». Наводящий вопрос в той же ситуации: «Не правда ли, очень трудно оценить личные качества сотрудников при премировании?».

Кроме перечисленных выше, полезно различать и включать в интервью следующие вопросы:

контактные (ломающие лед между аналитиком и специалистом предприятия); буферные (для разграничения отдельных тем интервью);

оживляющие память специалистов (для реконструкции отдельных случаев из практики);

провоцирующие (для получения спонтанных, неподготовленных ответов).

Свободный диалог – это метод извлечения знаний в форме беседы инженера по знаниям и эксперта, в которой нет жестко регламентированного плана и вопросника.

Внешне свободная и легкая форма этого метода требует высочайшей профессиональной и психологической подготовки. Рисунок 2.7 графически иллюстрирует схему такой подготовки.

Квалифицированная подготовка к диалогу помогает аналитику стать истинным драматургом или сценаристом будущих сеансов, т.е. запланировать гладкое течение процедуры извлечения знаний.

В свободном диалоге важно выбрать правильный темп или ритм беседы: без больших пауз, так как специалист может отвлечься, но и без «гонки», иначе оба участника быстро утомляются, и нарастает напряженность; кроме того, некоторые люди говорят и думают медленно. Умение чередовать разные темпы, напряжение и разрядку в беседе существенно влияет на результат.

Подготовка к диалогу, так же, как и к другим активным методам извлечения знаний, включает составление плана проведения сеанса, в котором необходимо предусмотреть следующие стадии.

начало беседы (знакомство, создание у специалиста предприятия «образа» аналитика, объяснение целей и задач работы);

диалог по извлечению знаний;

заключительная стадия (благодарность за потраченное время, подведение итогов, договоренность о последующих встречах).

Рисунок 2.7. Схема подготовки к беседе

Пример интервью инженера знаний с экспертом

Представим гипотетический сценарий, иллюстрирующий некоторые шаги, которые инженер знаний мог бы предпринять, чтобы разработать экспертную систему. Предметной областью для сценария выбрано оценивание размера компенсации за причиненный ущерб, вызванный выпуском дефектной продукции.

Проблема. В некоторой страховой компании за последнее время резко увеличилось количество заявлений о выплате по страховке, в то время как число опытных адвокатов ограничено. Расходы компании повысились, и по многим заявлениям были выплачены большие деньги. Появилось ощущение, что суммы выплат по заявлениям неоправданно завышены. Надо найти способ помочь адвокатам оценивать сумму компенсации более точно. Решено использовать экспертную систему.

Первый этап построения ЭС. Идентификация, задачей которой является определить участников, ресурсы, цели и характеристики проблемы.

Источники знаний: чтение литературы, изучение предшествующих исков, разговоры с экспертами и адвокатами, выявление ресурсов для построения.

Цель – построить демонстрационный прототип.

Второй этап построения ЭС. Концептуализация, цель которой – выявить в явном виде основные понятия и связи, необходимые для нахождения решений в этой области. Источники знаний – примеры, которые разбираются вместе с экспертом.

Один из примеров может быть такой. Некто N. (назовем его истец) открывал бутылку с шампанским, выпущенную застрахованной компанией «ВиН» и попал себе пробкой в глаз, что привело к потере зрения. После операции выяснилось, что имеется вероятность 5%-10% того, что N. будет страдать глаукомой. Специальные убытки истца в сумме составляют 4500 руб. Выясним понятие общих убытков на основе диалога инженера с экспертом.

ИНЖЕНЕР: Как вы думаете, чему равны общие убытки в рассматриваемом случае?

ЭКСПЕРТ: Ну**,** вероятно, они составят около 50 000 руб. – о нет,

пожалуй, 60 000 руб.

ИНЖЕНЕР: Как вы пришли к такому выводу?

ЭКСПЕРТ: У меня богатый жизненный опыт. Я просто смотрю на факты и оцениваю стоимость.

ИНЖЕНЕР: Какой фактор представляется вам наиболее важным?

ЭКСПЕРТ: Ну, возможно, здесь наиболее важный фактор травма глаза. Он заслуживает некоторой компенсации за страдания.

ИНЖЕНЕР: 60000 руб.?

ЭКСПЕРТ: Нет, травма сама по себе оценивается примерно в 18000 руб.

ЭКСПЕРТ: Понимаю, истцу должна быть выплачена компенсация за боль и страдания.

ЭКСПЕРТ: Не совсем точно. Я думаю, что сама травма оценивается только в 10000 руб. 8000 руб. – это компенсация за то, что заявитель провел 4 дня ослепшим на один глаз. Эта ситуация сопряжена с большим страхом.

ИНЖЕНЕР: Итак, 10000 руб. нужно выплатить истцу за саму травму, 8000 руб. – за неудобства, которые испытал истец, в то время как 10000 руб. отражают боль и шоковое состояние после травмы.

ЭКСПЕРТ: Это верно за исключением того, что я не назвал бы его страх за слепоту неудобством. Тот факт, что он должен будет носить очки, является примером неудобства. Компенсация за это составляет 2500 руб.

ИНЖЕНЕР: Следовательно, истцу надо выплатить 2500 руб. за необходимость носить очки.

ЭКСПЕРТ: Не совсем верно. 2500 руб. он должен получить за неудобство, связанное с ношением очков. Еще $2500 за частичную потерю зрения.

Итак, эксперт оценивает различные аспекты ранения такие, как травма, страх, неудобства, частичная потеря трудоспособности, а затем складывает эти оценки. Из дальнейшего разговора выяснилось, что истец не получил работу диктора из-за необходимости носить очки, что было оценено в 5000 руб. Возможные будущие осложнения были оценены в 30000 руб. Общая сумма составила 58000 руб. С этой оценкой согласился эксперт.

Результатом интервью является набор понятий, которые, как полагает инженер по знаниям, имеют непосредственное отношение к определению стоимости телесного повреждения.

Задача: определение стоимости телесного повреждения.

Характеристики повреждения: место повреждения глаз.

Результат повреждения: несколько снизилась острота зрения, потеря зрения на одном глазе, продолжительность слепоты 4 дня, неопределенность относительно постоянства слепоты, вероятность осложнения 10%, необходимость носить очки.

Факторы, влияющие на оценку повреждения: фактор травмы; фактор страха; фактор неудобства; фактор частичной утраты работоспособности; фактор обезображивания; фактор возможных осложнений в будущем.

Следующим шагом в построении экспертной системы является построение операционной и фактуальной моделей знаний.

Активные групповые методы

К групповым методам извлечения знаний относятся:

ролевые игры;

дискуссии за круглым столом с участием нескольких экспертов;

мозговой штурм.

Основное достоинство групповых методов – это возможность одновременного поглощения знаний от нескольких экспертов, взаимодействие которых вносит элемент принципиальной новизны от наложения разных взглядов и позиций.

Метод круглого стола предусматривает обсуждение какой-либо проблемы из выбранной предметной области, в котором принимают участие с равными правами несколько экспертов. Обычно вначале

участники высказываются в определенном порядке, а затем переходят к живой свободной дискуссии. Число участников дискуссии колеблется от трех до пяти-семи. Существует и специфика, связанная с поведением человека в группе.

Во-первых, от инженера по знаниям подготовка круглого стола потребует дополнительных усилий, как организационных (место, время, обстановка, минеральная вода, чай, кворум и т.д.), так и психологических (умение вставлять уместные реплики, чувство юмора, память на имена и отчества, способность гасить конфликтные ситуации и т.д.).

Во-вторых, большинство участников будут говорить под воздействием «эффекта фасада» совсем не то, что они сказали бы в другой обстановке, т.е. желание произвести впечатление на других экспертов будет существенно «подсвечивать» их высказывания.

Задача дискуссии – коллективно, с разных точек зрения, под разными углами исследовать спорные гипотезы предметной области. Обычно эмпирические области богаты таким дискуссионным материалом. Для остроты на круглый стол приглашают представителей разных научных направлений и разных поколений, это также уменьшает опасность получения односторонних знаний.

Перед началом дискуссии ведущему следует:

убедиться, что все правильно понимают задачу (т.е. происходит сеанс извлечения знаний);

установить регламент и четко сформулировать тему.

По ходу дискуссии проследить, чтобы слишком эмоциональные и разговорчивые эксперты не подменили тему, и критика позиций друг друга была обоснованной.

Мозговой штурм или мозговая атака – один из наиболее распространенных методов раскрепощения и активизации творческого мышления. Впервые этот метод был использован в США как способ получения новых идей в условиях запрещения критики. Замечено, что боязнь критики мешает творческому мышлению, поэтому основная идея штурма – это отделение процедуры генерирования идей в замкнутой группе специалистов от процесса анализа и оценки высказанных идей.

Как правило, штурм длится около 40 мин. Участникам (до 10 человек) предлагается высказывать любые идеи (шутливые, фантастические, ошибочные) на заданную тему (критика запрещена). Обычно высказывается более 50 идей. Регламент до двух минут на выступление. Самый интересный момент штурма – это наступление пика

(ажиотажа), когда происходит непроизвольная (бессознательная) генерация гипотез участниками. При последующем анализе всего лишь 10 - 15% идей оказываются разумными, но среди них бывают весьма оригинальные. Оценивает результаты обычно группа экспертов, не участвовавшая в генерации.

Ведущий мозгового штурма – инженер по знаниям – должен свободно владеть аудиторией, подобрать активную группу экспертов – «генераторов», не зажимать плохие идеи – они могут служить катализатором хороших. Искусство ведущего – это искусство задавать вопросы аудитории, «подогревая» генерацию. Основной девиз штурма – «чем больше идей, тем лучше». Фиксация сеанса – традиционная (протокол или магнитофон).

Экспертные игры

Игрой называют такой вид деятельности, который отражает (воссоздает) другие ее виды. Понятие экспертной игры, или игры с экспертами, в целях извлечения знаний восходит к трем источникам – это понятие деловой игры, широко используемое при подготовке специалистов и моделировании, и понятие диагностической игры, а также компьютерные игры, все чаще применяемые в обучении.

Под деловой игрой чаще всего понимают эксперимент, где участникам предлагается производственная ситуация, а они на основе своего жизненного опыта, общих и специальных знаний и представлений принимают решения. Решения анализируются, и вскрываются закономерности мышления участников эксперимента. Если участниками такой игры становятся эксперты, то игра из деловой превращается в экспертную. Из трех основных типов деловых игр (учебных, планово-производственных и исследовательских) к экспертным ближе всего исследовательские, которые используются для анализа систем, проверки правил принятия решений.

Диагностическая игра – это та же деловая игра, но применяемая конкретно для диагностики методов принятия решения в предметной области, например в медицине.

Классификация экспертных игр приведена на рисунке 2.8.

Плодотворность моделирования реальных ситуаций в играх подтверждается сегодня практически во всех областях науки и техники. Они развивают логическое мышление, способности быстро принимать решения, вызывают интерес у экспертов.

Текстологические методы

Группа текстологических методов объединяет методы извлечения знаний, основанные на изучении специальных текстов из учебников, монографий, статей, методик и других носителей профессиональных знаний.

Рисунок 2.8. Классификация экспертных игр

Задачу извлечения знаний из текстов можно сформулировать как задачу понимания и выделения смысла текста. Сам текст на естественном языке является лишь проводником смысла, а замысел и знания автора лежат во вторичной структуре (смысловой структуре или макроструктуре текста), надстраиваемой над естественным текстом.

При этом можно выделить две такие смысловые структуры: M_1 смысл, который пытался заложить автор, это его модель мира, и M_2 смысл, который постигает читатель, в данном случае инженер по знаниям в процессе интерпретации I. При этом T - это результат вербализации V (рисунок 2.9).

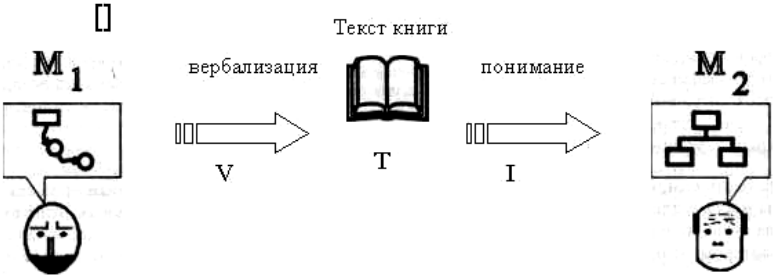


Рисунок 2.9. Извлечение знаний из текстов

Сложность процесса заключается в принципиальной невозможности совпадения знаний, образующих M_1 и M_2, из-за того, что M_1 образуется за счет всей совокупности представлений, потребностей, интересов и опыта автора, лишь малая часть которых находит отражение в тексте *T*. Соответственно и M_2 образуется в процессе интерпретации текста *T* за счет привлечения всей совокупности научного и человеческого багажа читателя. Таким образом, два инженера по знаниям извлекут из одного *T* две различные модели M_2^i *и* M_2^j.

Встает задача: выяснить, за счет чего можно достичь максимальной адекватности M_1 и M_2, помня при этом, что понимание всегда относительно, поскольку это синтез двух смыслов «свое – чужое».

Рассмотрим подробнее, какие источники питают модель M_1 и создают текст *T*. Существуют два компонента любого научного текста. Это первичный материал наблюдений α и система научных понятий β в момент создания текста. В дополнение к этому, помимо объективных данных экспериментов и наблюдений, в тексте обязательно присутствуют субъективные взгляды автора γ, результат его личного опыта, а также некоторые «общие места» или «вода» δ, кроме этого, любой научный текст содержит заимствования из других источников (статей, монографий и т. д.) θ. При этом все компоненты погружены в языковую среду *L*. Можно записать

$$T = (\alpha, \beta, \gamma, \delta, \theta)_L.$$

Таким образом, компоненты научного текста *T:*

наблюдения - α,

научные понятия - β,

субъективные взгляды - γ,

общие места - δ,

заимствования - θ.

При этом компоненты α, γ, часть а входят и в модель M_1.

При извлечении знаний аналитику, интерпретирующему текст, приходится решать задачу декомпозиции этого текста (на перечисленные выше компоненты для выделения истинно значимых для реализации базы знаний фрагментов). Сложность интерпретации научных и специальных текстов заключается еще и в том, что любой текст приобретает смысл только в контексте, где под *контекстом* понимается окружение, в которое «погружен» текст.

Различают микро- и макроконтекст. *Микроконтекст* – это ближайшее окружение текста. Так, предложение получает смысл в контексте

абзаца, абзац в (контексте главы и т. д. *Макроконтекст* – это вся система знаний, связанная с предметной областью (т. е. знания об особенностях и свойствах, явно не указанных в тексте). Другими словами, любое знание обретает смысл в контексте некоторого метазнания.

Говоря на языке современного языкознания, *понимание* - это формирование «второго текста», т.е. семантической структуры (понятийной структуры). В нашей терминологии – это попытка воссоздания семантической структуры M_1 в процессе формирования модели M_2, т. е. это первый шаг структурирования знаний.

Как происходит процесс понимания? Изложим одну из возможных схем. Эта схема согласуется со стратегией изучения всего нового. *Основными моментами понимания текста* являются:

1. Выдвижение предварительной гипотезы о смысле всего текста (предугадывание).
2. Определение значений непонятных слов (т. е. специальной терминологии).
3. Возникновение общей гипотезы о содержании текста (о знаниях).
4. Уточнение значения терминов и интерпретация отдельных фрагментов текста под влиянием общей гипотезы (от целого к частям).
5. Формирование некоторой смысловой структуры текста за счет установления внутренних связей между отдельными важными (ключевыми) словами и фрагментами, а также за счет образования абстрактных понятий, обобщающих конкретные фрагменты знаний.
6. Корректировка общей гипотезы относительно содержащихся в тексте фрагментов знаний (от частей к целому).
7. Принятие основной гипотезы, т. е. формирование M_2. Следует отметить наличие как дедуктивной (от целого к частям), так и индуктивной (от частей к целому) составляющей процесса понимания. Такой двуединый подход позволяет охватывать текст как смысловое единство особого рада, с его основными признаками, такими, как связность, цельность, законченность и др.

Центральными моментами процесса *I* являются шаги 5 и 7, т. е. формирование смысловой структуры или выделение «опорных», ключевых, слов или «смысловых вех», а также заключительное схватывание «смысловых вех» в единую *семантическую структуру*.

При анализе текста важно выявление внутренних связей между отдельными элементами текста и понятиями. Традиционно выделяют два вида связей в тексте – *эксплицитные* (или явные связи), которые выражаются во внешнем дроблении текста, и *имплицитные* (скрытые связи

– внутренние). Эксплицитные связи делят текст на параграфы с помощью перечисления компонентов вводных слов (или коннекторов) типа «во-первых ..., во-вторых ..., однако и т. д.». Имплицитные, или внутренние связи между отдельными «смысловыми вехами» вызывают основное затруднение при понимании.

Итак, *семантическая структура* текста образуется в сознании познающего субъекта с помощью знаний о языке, знаний о мире, а также общих знаний (фоновых), в той предметной области, которой посвящен текст. «Тексты пишут для посвященных». Другими словами, если текст не является научно-популярными, то для его адекватного прочтения требуется некоторая подготовка.

Таким образом, путь к знаниям удлиняется еще на одно звено. Если мы раньше говорили, что сами текстологические методы редко употребляются как самостоятельный метод извлечения, а обычно используются как некоторая подготовка к коммуникативному взаимодействию, то теперь утверждаем, что и для прочтения текстов нужна подготовка.

Подготовкой к прочтению специальных текстов является выбор совместно с экспертами некоторого «базового» списка литературы, который постепенно введет аналитика в предметную область. В этом списке могут быть учебники для начинающих, главы и фрагменты из монографий, популярные издания. Только после ознакомления с «базовым» списком целесообразно приступать к чтению специальных текстов.

Компоненты формирования смысла текста M_2:
личный опыт α аналитика - φ,
общенаучная эрудиция аналитика - ε,
предварительные знания аналитика о предметной области – ω,
экстракт текста T -$(\alpha, \beta, \gamma, \theta)$

Таким образом, на процесс понимания I и модель M_2 влияют следующие компоненты:
экстракт компонентов $(\alpha, \beta, \gamma, \theta)$ почерпнутый из текста T;
предварительные знания аналитика о предметной области ω;
общенаучная эрудиция аналитика ε;
его личный опыт φ.

$$M_2=[(\alpha, \beta, \gamma, \theta)' \omega, \varepsilon, \varphi].$$

Процесс I - это очень сложный, не поддающийся формализации процесс, на который существенным образом влияют такие чисто индиви-

дуальные компоненты, как когнитивный стиль познания, интеллектуальные характеристики и др.

Но процедура разбивки текста на части (« смысловые группы»), а затем сгущение, сжатие содержимого каждого смыслового куска в «смысловую веху» является, видимо, основой для любого индивидуального процесса понимания. Такая компрессия текста в виде набора ключевых слов, передающих основное содержание текста, может служить удобной методологической основой для проведения текстологических процедур извлечения знаний

В качестве ключевого слова может служить любая часть речи (существительное, прилагательное, глагол и т. д.) или их сочетание. *Набор ключевых слов* (НКС) – это набор опорных точек, по которым развертывается текст при кодировании в память и осознается при декодировании, это семантическое ядро цельности.

В качестве примера приведем результаты эксперимента по формированию НКС. Знания извлекались из следующего текста.

«Теория фреймов» относится к психологическим понятиям, касающимся понимания того, что мы видим и слышим. Эти способы восприятия трактуются с последовательной точки зрения, на их основании осуществляется концептуальное моделирование, целесообразность полученных моделей исследуется вместе с различными проблемами, возникающими в этих двух областях.

Для осознания того факта, что заданная информация в этих областях имеет единственный смысл, человеческая память, прежде всего, должна быть способна увязывать эту информацию со специальными концептуальными объекта МП. В противном случае не удается систематизировать информацию, которая выглядит разрозненной. В основе теории фреймов лежит восприятие фактов посредством сопоставления полученной извне информации с конкретными элементами и значениями, а также с рамками, определенными для каждого концептуального объекта в нашей памяти. Структура, представляющая эти рамки, называется фреймом. Поскольку между различными концептуальными объектами имеются некоторые аналогии, то образуется иерархическая структура с классификационными и обобщающими свойствами. Собственно она представляет собой иерархическую структуру отношений типа «абстрактное – конкретное». Сложные объекты представлены комбинацией нескольких фреймов, другими словами, они соответствуют фреймовой сети. Кроме того, каждый фрейм дополняется связанными с ним фактами и процедурой, обеспечивающей выполнение запросов к другим фреймам.

Причиной, по которой представление знаний фреймами выглядит достаточно точным, является возможность более полного описания процесса мышления человека посредством определения крупной и структурированной основной единицы представления знаний и более тесной связи знаний, основанных на фактах, и процедурных знаний. Тем не менее, как было отмечено ее автором, теорию фреймов следует скорее отнести к теории постановки задач, чем к результативной теории. Можно считать, что она существенно повышает уровень и детализирует механизм памяти человека, выводов, понимания и обучения».

Интересно, что одна из гипотез лингвостатистики о том, что наиболее употребляемые слова являются наиболее важными с точки зрения содержания текста, т. е. отражают его тематическую структуру, частично подтвердилась.

Выделим три вида текстологических методов:

анализ специальной литературы,

анализ учебников,

анализ методик.

Обозначенные три метода существенно отличаются, во-вторых, по степени концентрированности специальных знаний, и, во-вторых, по соотношению специальных и фоновых знаний. Наиболее простым методом является анализ учебников, в которых логика изложения обычно соответствует логике предмета и поэтому макроструктура такого текста будет, наверное, более значима, чем структура текста какой-нибудь специальной статьи. Анализ методик затруднен как раз сжатостью изложения и практическим отсутствием комментариев, т. е. фоновых знаний, облегчающих понимание для неспециалистов. Поэтому можно рекомендовать для практической работы комбинацию перечисленных методов.

В заключение приведём одну из возможных практических методик анализа текстов с целью извлечения и структурирования знаний.

1. Составление «базового» списка литературы для ознакомления с предметной областью и чтение по списку.
2. Выбор текста для извлечения знаний.
3. Первое знакомство с текстом (беглое прочтение). Для определения значения незнакомых слов – консультации со специалистами или привлечение справочной литературы.
4. Формирование первой гипотезы о макроструктуре текста.
5. Внимательное прочтение текста с выписыванием ключевых слов и выражений, т.е. выделение «смысловых вех» (компрессия текста).

6. Определение связей между ключевыми словами, разработка макроструктуры текста в форме графа или «сжатого» текста (реферата).

7. Формирование поля знаний на основании макроструктуры текста.

2.5. Выявление знаний от экспертов

Эффективность начальных этапов разработки ЭС (этапов идентификации и концептуализации) во многом определяется успешным формированием авторитетной группы экспертов и получением от них качественных знаний, составляющих основу любой ЭС.

Суть процесса выявления знаний заключается в организации проведения экспертами интуитивно-логического анализа проблемной области с количественной оценкой формулируемых ими суждений. На этом этапе эксперты:

формируют объекты и понятия предметной области (цели, решения, альтернативные ситуации и т.д.);

производят измерение характеристик (вероятности свершения событий, коэффициенты значимости целей, предпочтение решений и т.д.).

Экспертное оценивание представляет собой процесс *измерения*, который можно определить как процедуру сравнения объектов по выбранным показателям (признакам). В этом определении фигурируют три понятия: объект, показатель (признак) и процедура сравнения.

Объектами могут быть предметы, явления, решения. В качестве *показателей сравнения* могут использоваться пространственно-временные, физические, психические и другие свойства и характеристики объектов. *Процедура сравнения* включает в себя:

определение причинно-следственной связи между объектами;

а также установление степени влияния одних объектов на другие.

Последнее обстоятельство требует проведения сравнения объектов, определяющих какой-либо результат, по степени их влияния на него. Введение конкретных показателей сравнения позволяет экспертам устанавливать отношения между объектами, например, «больше», «лучше», «более чем», «хуже», «одинаковы», «предпочтительнее» и т.д.

Для формального описания множества объектов и отношений между ними вводится понятие эмпирической системы с отношениями:

$$M = < O ; R >,$$

где $O=(O_1,O_2,...,O_n)$ – множество объектов предметной области, $R=(R_1,R_2,...,R_n)$ – множество отношений между ними.

Запись $O_iR_kO_j$ означает, что объект O_i находится в отношении R_k к объекту O_j. Такое отношение называется бинарным (двуместным) поскольку связывает два объекта. Отношения могут быть трехместными и т.д. В общем случае *k*-местными, по числу объектов, которые они связывают. Определим основные свойства отношений:

отношение R *рефлексивно*, если O_iRO_j – истинно;

отношение R *симметрично*, если из O_iRO_j следует O_jRO_i;

отношение R *транзитивно*, если из O_iRO_j и O_jRO_k следует O_iRO_k.

Отношение, которое обладает свойствами рефлексивности, симметричности и транзитивности называется отношением *эквивалентности.*

$$O_i \sim O_j$$

При экспертном оценивании, кроме отношения эквивалентности используется отношение *порядка*. Это отношение может означать, например: «раньше чем», «более чем», «сильнее чем», «предпочтительнее чем» и т.д. Отношение порядка антирефлексивно, транзитивно и обозначается:

$$O_i \ \phi \ O_j$$

для случая, если O_i предпочтительнее O_j.

Экспертное оценивание в процессе выявления знаний от экспертов, использует наряду с эмпирической, также *числовую* систему с отношениями:

$$H = < N ; S >,$$

где N – множество действительных чисел, а S – множество отношений между числами. Именно эта система позволяет не только устанавливать причинно-следственные связи между объектами предметной области, но и определять их взаимосвязи и влияние.

При экспертном оценивании предметной области важным является возможность для *эмпирической системы* с отношениями построения *числовой системы* с отношениями, описывающими влияние объектов и отношения между ними с помощью чисел.

Для того чтобы числовая система сохраняла свойства и отношения объектов, необходимо, чтобы она была *изоморфной* эмпирической системе. Для пояснения этого понятия определим понятие подобности двух систем. Две системы с отношениями:

$$M = (O ; R_1, R_2, ..., R_k),$$
$$H = (N ; S_1, S_2, ..., S_m)$$

называются *подобными*, если:

число отношения (заданных на множестве объектов и действительных чисел) одинаково, то есть $k = m$;

местность отношений одинакова (например ***Ri*** и ***Si*** двуместные отношения).

Определив понятие подобности, мы можем теперь дать определение изоморфности двух систем (числовой и эмпирической). Числовая система с отношениями:

$$H = (N ; S_1, S_2, ..., S_m)$$

изоморфна эмпирической системе:

$$M = (O ; R_1, R_2, ..., R_k),$$

если:

эти системы подобны;

существует взаимно однозначное отображение (функция) ***f*** объектов на числовое множество такое, что отношение R_k между объектами имеет место тогда и только тогда, когда имеет место отношение S_m между числами, отображающими объекты на числовой оси.

Например, для случая двуместных отношений $O_i R_k O_j$ это будет иметь место тогда и только тогда, когда имеет место $r_i S_k r_j$, где r_i и r_j получены отображением объектов $r_i=f(O_i)$ и $r_j=f(O_j)$.

Проблема *единственности* определяет: сколькими способами можно описать данную эмпирическую систему различными изоморфными числовыми системами, и как эти числовые системы связаны между собой. Эта проблема формулируется, как проблема определения типа шкалы. *Шкалой* называется совокупность:

эмпирической системы;

числовой системы;

и отображения, то есть $< M, H, f >$.

Пусть $< M, H, f >$ и $< M, H, g >$ две шкалы с разными отображениями, тогда возникает вопрос о взаимосвязи числовых значений, полученных этими отображениями. Например $r_i = f(O_i)$, $r_i' = g(O_i)$. Связь между числами r_i и r_i' запишем с помощью функции φ : $r_i = \varphi (r_i')$ или $f(O_i) = \varphi [g (O_i)]$.

Функция φ называется допустимым преобразованием шкалы. Единственность описания эмпирической системы числовыми системами выражается в свойствах допустимого преобразования шкалы, то есть в свойствах функции φ.

К наиболее часто используемым при экспертном оценивании методам относятся: ранжирование, парное сравнение, непосредственная оценка. При описании каждого из перечисленных методов будем полагать, что имеется конечное число измеряемых объектов и сфор-

мулирован один или несколько признаков сравнения, по которым изучается степень влияния объектов на результат.

Следовательно, методы измерения будут различаться лишь процедурой сравнения объектов. Эта процедура включает:

построение отношений между объектами эмпирической системы;

выбор функции f, отображающей объекты эмпирической системы на числовую систему;

определение шкалы измерений.

Рассмотрим подробно все эти вопросы, возникающие при использовании каждого из методов измерений.

Метод ранжирования

Ранжирование – это процедура упорядочения объектов по степени их влияния на результат, выполняется экспертом в процессе выявления его знаний. На основе своих знаний и опыта эксперт располагает объекты в порядке предпочтения, руководствуясь одним или несколькими показателями сравнения. В зависимости от вида отношений между объектами возможны различные варианты упорядочения объектов.

Пусть среди объектов нет *эквивалентных* по степени влияния на результат. В этом случае между объектами существует отношение строгого порядка, обладающее свойствами:

несимметричности (если $O_i \phi O_j$, то $O_i \phi O_j$);

транзитивности (если $O_i \phi O_j$, $O_j \phi O_k$, то $O_i \phi O_k$);

и связности (для любых двух объектов, либо $O_i \phi O_j$, либо $O_j \phi O_i$).

В результате сравнения всех объектов по отношению строгого порядка эксперт составляет упорядоченную последовательность:

$$O_1 \phi O_2 \phi \ldots \phi O_n, \quad (1)$$

где объект с номером один является наиболее предпочтительным из всех объектов, объект с номером два менее предпочтителен чем первый, но предпочтительнее всех остальных и т.д.

Полученная система с отношением порядка $< O, \phi >$ образует *серию*. Для серии доказано существование числовой системы:

элементами которой являются числа;

а отношение порядка ϕ есть отношение «больше чем», «предпочтительнее чем».

Это означает, что существует числовое представление $f(O_i)$ такое, что последовательности (1) соответствует последовательность чисел:

$$f(O_1) > f(O_2) > \ldots > f(O_n). \quad (1)$$

В практике экспертного ранжирования чаще всего используется последовательность натуральных чисел:

$$r_1=f(O_1)=1;\ r_2=f(O_2)=2;\ .\ .\ ;\ r_n=f(O_n)=n.$$

Числа $r_1, r_2, ..., r_n$ называются рангами. Наиболее предпочтительному присваивается ранг 1, второму – ранг 2 и т.д. На практике, среди объектов могут быть и эквивалентные по степени их влияния на результат. Например, упорядочение может иметь вид:

$$O_1 \phi O_2 \phi O_3 \sim O_4 \sim O_5 \phi \ldots \phi O_{n-1} \sim O_n \qquad (2)$$

В этой последовательности объекты O_3, O_4 и O_5 эквивалентны между собой, а O_{n-1} и O_n – между собой. Для эквивалентных объектов принято назначать одинаковые ранги, равные среднему арифметическому значению рангов, приписываемых одинаковым объектам. Такие ранги получили название *связанных рангов*. Для примера упорядочение (2) в случае n=10 ранги объектов O_3, O_4 и O_5 будут одинаковыми и равными:

$$r_3 = r_4 = r_5 = (3+4+5)/3 = 4$$
$$r_9 = r_{10} = (9+10)/2 = 9{,}5$$

Как видно из примера связанные ранги могут быть дробными. Удобство использования связанных рангов заключается в том, что сумма рангов n-объектов равна сумме натуральных чисел от 1 до n. При этом любые комбинации связанных рангов не изменяют эту сумму. Это обстоятельство существенно упрощает обработку результатов ранжирования при групповой экспертной оценке.

Метод парных сравнений

Парное сравнение представляет собой процедуру установления предпочтения объектов при сравнении всех возможных пар. В отличие от ранжирования, при котором осуществляется упорядочение всех объектов сразу, парное сравнение представляет для экспертов более простую задачу. При сравнении каждой пары объектов возможны *отношения* либо *порядка*, либо эквивалентности. Парное сравнение есть измерение в *шкале порядка*.

В результате сравнения каждой пары объектов O_i и O_j эксперт должен упорядочить эту пару, высказывая, что

либо $O_i \phi O_j$, либо $O_j \phi O_i$, либо $O_i \sim O_j$.

Переход от эмпирической системы к числовой системе с отношениями осуществляется выбором такой функции f, что

если $O_i \phi O_j$, то $f(O_i) > f(O_j)$,
если $O_j \phi O_i$, то $f(O_i) < f(O_j)$.

Наконец, если объекты эквивалентны, то естественно предположение, что $f(O_i)=f(O_j)$. Наиболее часто в практике экспертного оценивания используют следующие числовые представления:

Эмпирическая система	Представление 1		Представление 2		Представление 3	
	$f(O_i)$	$f(O_j)$	$f(O_i)$	$f(O_j)$	$f(O_i)$	$f(O_j)$
$O_i \phi O_j$	2	0	1	-1	1	0
$O_i \sim O_j$	1	1	0	0	0,5	0,5

Результаты сравнения экспертом всех пар объектов удобно представить в виде таблицы, столбцы и строки которой составляют объекты, а в ячейках таблицы проставляются числовые значения.

Пример: В качестве примера рассмотрим табличное отображение результатов проведенного парного сравнения пяти объектов при использовании числового представления 1.

O_i	O_i: O_1	O_2	O_3	O_4	O_5
O_1	1	2	2	1	2
O_2	0	1	2	1	0
O_3	0	0	1	0	1
O_4	1	1	2	1	0
O_5	0	2	1	2	1

Из этой таблицы следует, что объект O_1 предпочтительнее объектов O_2, O_3, O_5 и эквивалентен O_4. Объект O_2 предпочтительнее O_3, эквивалентен O_4 и менее предпочтителен, чем O_1 и O_5. Сравнение объектов во всех возможных парах не дает полного упорядочения всех объектов. Поэтому возникает задача о ***ранжировке объектов*** на основе парного сравнения**.**

Метод непосредственной оценки

Непосредственная оценка представляет собой процедуру приписывания объектам числовых значений в шкале интервалов. Эти значение соответствует степени влияния того или иного объекта на наблюдаемый результат.

В процессе выявления знаний эксперт должен поставить в соответствие каждому объекту точку на непрерывной числовой оси, например, на отрезке [0,1]. Естественно потребовать, чтобы эквивалентным по воздействиям объектам приписывалось бы одно и тоже число.

Измерение предпочтения в шкале интервалов можно выполнить с высокой степенью доверия только при хорошей информированности экспертов о свойствах объектов и предметной области.

В ряде случаев, с целью ослабления этих условий, но, естественно, за счет уменьшения точности измерения вместо непрерывной числовой оси рассматривают большую оценку, которая использует 5, 10, 100 – бальные шкалы.

Однако непосредственная оценка не всегда должна использовать числовые шкалы. Например, цвет объекта невозможно представить в виде какого-либо числового значения, а переход к значениям частот спектра во многих случаях затруднителен для эксперта.

Один из подходов к формированию и оценке компетентности группы экспертов

При формировании группы экспертов на стадии выявления знаний необходимо учитывать такие характеристики экспертов, как

компетентность – степень квалификации эксперта в данной области знаний;

креативность – способность решать творческие задачи;

отношение к экспертизе – негативное или пассивное отношение, или занятость существенно влияет на качество работы эксперта в группе;

конформизм – подверженность влиянию авторитетов, при котором мнение авторитета может подавлять лиц, обладающих более высокой компетентностью;

коллективизм и самокритичность.

Рассмотрим один из возможных путей количественного описания характеристик эксперта, основанный на вычислении относительных коэффициентов компетентности по результатам высказывания специалистов о составе экспертной группы.

Суть методики сводится к тому, что ряду специалистов предлагается высказать мнение о списочном составе экспертной группы. Если в этом списке появляются лица, не вошедшие в исходный список, им тоже предлагается назвать специалистов для участия в экспертизе. После нескольких этапов будет получен достаточно полный список кандидатов в группу.

По результатам опроса составляется матрица, по строкам и столбцам которой записываются фамилии экспертов, а элементами таблицы являются переменные:

$$x_{ij} = \begin{cases} 1, \text{ если j - ый эксперт назвал i - ого} \\ 0, \text{ если j - ый эксперт не назвал i - ого} \end{cases}$$

При этом эксперт может включать себя или не включать в экспертную группу (то есть x_{ij}=0 или x_{ij}=1). По данной таблице можно вычислить относительные коэффициенты компетентности, используя алгоритм решения задач о лидере [1]. Введем относительные коэффициенты компетентности h-порядка для каждого эксперта:

$$K_i^h = \frac{\sum_{j=1}^{m} x_{ij} k_j^{h-1}}{\sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} k_j^{h-1}}, \quad (i = \overline{1,m};\ h = 1,2\ldots) \quad (3)$$

где m – число экспертов в списке (размерность матрицы $\|x_{ij}\|$), h – номер порядка коэффициента компетентности. Коэффициенты компетентности нормированы так, что их сумма равна единице:

$$\sum_{i=1}^{m} k_i^h = 1, \quad h=1,2\ldots. \quad (4)$$

По формуле (3) можно вычислить значение компетентности для различных порядков, начиная с первого. При h=1 выражение (3) будет иметь вид:

$$k_i^1 = \frac{\sum_{j=1}^{m} x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij}}, \quad i = 1,2,\ldots,m. \quad (5)$$

Смысл этой формулы в том, что подсчитывается число голосов, поданных за i-го эксперта и делится на общее число голосов, поданных за всех экспертов. Таким образом, коэффициент компетентности первого порядка – это относительное число экспертов, высказавшихся за включение i-го эксперта в группу.

Относительный коэффициент компетентности второго порядка получают из (1) для h=2 при условии, что $k_j^1 (j=1,2\ldots m)$ определены по (5):

$$k_i^2 = \frac{\sum_{j=1}^{m} x_{ij} k_j^1}{\sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} k_j^1}, \quad i = 1,2,\ldots,m.$$

Коэффициенты второго порядка представляют собой относительное количество голосов, взвешенных коэффициентом компетентности первого порядка.

Последовательно вычисляя относительные коэффициенты компетентности более высокого порядка, можно убедиться, что процесс быстро сходится после 3-4 вычислений. В общем случае коэффициенты относительной компетентности определяются как:

$$k_i = \lim_{h\to\infty} k_i^h \ , \quad \sum_{i=1}^{n} k_i = 1 .$$

Пример: В результате опроса трех экспертов о составе экспертной группы получены данные (x_{ij}) о мнении каждого из них по включению экспертов в рабочую группу. Эти данные сведены в таблицу:

	Мнения экспертов		
	Эксперт 1 (A)	Эксперт 2 (B)	Эксперт 3 (C)
Эксперт 1 (A)	1	1	1
Эксперт 2 (B)	0	1	0
Эксперт 3 (C)	1	0	1

Результаты пошаговой обработки полученных данных по описанному выше алгоритму будут иметь вид.

На первом шаге, полагая равную компетентность всех экспертов, принимаем $k^0 = [\ 1\ 1\ 1\ ]^{\mathrm{T}}$ и вычисляем коэффициенты относительной компетентности первого порядка:

$$y = \sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} = 1+1+1+0+1+0+1+0+1 = 6 ,$$

$$k_A^1 = k_1^1 = \frac{1}{y} \times \sum_{j=1}^{3} x_{1j} k_j^0 = \frac{1}{6} \times (1\times 1 + 1\times 1 + 1\times 1) = \frac{3}{6} = 0.5 ,$$

$$k_B^1 = k_2^1 = \frac{1}{y} \times \sum_{j=1}^{3} x_{2j} k_j^0 = \frac{1}{6} \times (0\times 1 + 1\times 1 + 0\times 1) = \frac{1}{6} \approx 0.167 ,$$

$$k_C^1 = k_3^1 = \frac{1}{y} \times \sum_{j=1}^{3} x_{3j} k_j^0 = \frac{1}{6} \times (1\times 1 + 0\times 1 + 1\times 1) = \frac{2}{6} \approx 0.333 .$$

На втором шаге, используя полученные значения, вычисляем коэффициенты относительной компетентности второго порядка:

$$y = \sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} k_j^1 = 1\times\frac{3}{6}+1\times\frac{1}{6}+1\times\frac{2}{6}+0\times\frac{3}{6}+1\times\frac{1}{6}+0\times\frac{2}{6}+1\times\frac{3}{6}+0\times\frac{1}{6}+1\times\frac{2}{6} = \frac{12}{6} = 2$$

$$k_A^2 = k_1^2 = \frac{1}{y}\times\sum_{j=1}^{3} x_{1j} k_j^1 = \frac{1}{2}\times(1\times\frac{3}{6}+1\times\frac{1}{6}+1\times\frac{2}{6}) = \frac{1}{2}\times\frac{6}{6} = 0.5,$$

$$k_B^2 = k_2^2 = \frac{1}{y}\times\sum_{j=1}^{3} x_{2j} k_j^1 = \frac{1}{2}\times(0\times\frac{3}{6}+1\times\frac{1}{6}+0\times\frac{2}{6}) = \frac{1}{2}\times\frac{1}{6} \approx 0.083,$$

$$k_C^2 = k_3^2 = \frac{1}{y}\times\sum_{j=1}^{3} x_{3j} k_j^1 = \frac{1}{2}\times(1\times\frac{3}{6}+0\times\frac{1}{6}+1\times\frac{2}{6}) = \frac{1}{2}\times\frac{5}{6} \approx 0.417.$$

Продолжая аналогичные вычисления до тех пор, пока k_i^h не будут отличаться от k_i^{h-1} с точностью 0.01, получим

$$k^3 = \begin{bmatrix}0.5 & 0.042 & 0.458\end{bmatrix}^T$$

$$k^4 = \begin{bmatrix}0.5 & 0.02 & 0.48\end{bmatrix}^T$$

$$k^5 = \begin{bmatrix}0.5 & 0.01 & 0.49\end{bmatrix}^T$$

При $h\to\infty$ $k^h \to \begin{bmatrix}0.5 & 0.0 & 0.5\end{bmatrix}^T$.

Можно показать, что предельные значения коэффициентов компетентности представляют собой компоненты собственного вектора для максимального собственного числа матрицы $\boldsymbol{X}=\|\boldsymbol{x_{ij}}\|$. Собственные числа матрицы $\boldsymbol{X}$ определяются как корни алгебраического уравнения:

$$|X - \lambda\times E| = 0.$$

где λ – вектор собственных чисел матрицы голосования $|X|$, E – единичная матрица. Собственный вектор матрицы, соответствующий максимальному собственному числу, вычисляется из системы m+1 порядка линейных алгебраических уравнений:

$$XK = \lambda_0 K, \qquad \sum_{i=1}^{m} k_i = 1,$$

где $K = \begin{bmatrix}k_1, k_2, \ldots, k_m\end{bmatrix}$ – вектор компетентности, являющийся собственным вектором матрицы X для максимального собственного числа.

Существуют подходы к оценке компетентности, основанные на учете апостериорных данных, то есть результатов экспертного оценивания.

Характеристика и режимы работы группы экспертов

Основными характеристиками группы экспертов является достоверность экспертизы и затраты на нее. Обе эти характеристики определяют количество экспертов в группе и их качество.

Наряду с компетентностью, каждый из экспертов может характеризоваться достоверностью его суждений. Эта характеристика определяется на основе информации о прошлом опыте его участия в решении проблем. Количественно достоверность эксперта оценивают по формуле:

$$D_i = \frac{Nn_i}{N_i} \ ,(i=1,2...m)$$

где Nn_i – число опросов, когда эксперт дал приемлемое практикой решение, N_i – общее число случаев участия i-ого эксперта в экспертизе.

Можно также учесть вклад каждого эксперта в достоверность всей группы. Относительная достоверность выражается в виде:

$$D_i^{om} = \frac{D_i}{\frac{1}{m}\sum_{i=1}^{m} D_i},$$

где m – число экспертов в группе, а в знаменателе – средняя достоверность группы экспертов.

Для решения проблем с высоким уровнем информационного потенциала знаний увеличение количества экспертов в группе приводит, как это следует из теории обработки наблюдений, к монотонному возрастанию достоверности экспертизы. Экспериментальные исследования подтверждают эту зависимость.

Основной этап совместной работы группы экспертов – опрос.

Результатом опроса является информация, выражающая предпочтение экспертов и содержательное обоснование этих предпочтений. Наличие как числовых данных, так и содержательных высказываний экспертов, приводит к необходимости применения качественных и количественных методов обработки результатов группового экспертного оценивания.

2.6. Обработка экспертных оценок

Задачи обработки

В зависимости от целей экспертного оценивания и метода учета экспертных оценок возникают следующие основные задачи:

построение обобщенной оценки понятий и объектов на основе индивидуальных оценок экспертов;

построение обобщенной оценки на основе парного сравнения объектов каждым из экспертов;

определение относительных весов взаимосвязи объектов;

определение зависимостей между ранжировками;

определение согласованности мнений экспертов;

оценка надежности обработки результатов.

При решении многих задач недостаточно упорядочения объектов по одному или группе показателей. Необходимо иметь числовые значения для каждого объекта, определяющие его предпочтение перед другими объектами. Наличие таких оценок позволит определить обобщенную оценку для всей группы экспертов.

Определение согласованности мнений экспертов производится путем вычисления *числовой меры*, характеризующей степень близости индивидуальных мнений. Анализ значения *меры согласования* способствует выработке правильного суждения об общем уровне знаний по решаемой проблеме и выявлению группировок мнений экспертов.

Обработка экспертных оценок позволяет вскрыть связанные показатели сравнения и осуществить группировку по степени связи. Так, например, если показатели сравнения – различные цели, а объекты сравнения – средства достижения этих целей, то установление взаимосвязи между ранжировками, упорядочивающими средства с точки зрения достижения целей, позволяет обоснованно ответить на вопрос: «*в какой степени достижение одной цели при данных средствах способствует достижению других целей*» (то есть установить причинно-следственную связь).

Оценки, получаемые на основе обработки, представляют собой случайные объекты, поэтому одной из важнейших задач процедуры обработки является определение их надежности.

Групповая экспертная оценка объектов при непосредственном оценивании

Существует множество подходов к решению данной задачи. С целью иллюстрации рассмотрим один из простейших. Пусть m экспер-

тов провели оценку n объектов по l показателям. Результаты оценивания представлены величинами x_{ij}^{h}, где i – номер объекта, j – номер эксперта, h – номер показателя. Величины x_{ij}^{h}, полученные методам непосредственного оценивания, представляют собой числа из некоторого отрезка числовой оси, или баллы.

В качестве групповой оценки для каждого из объектов можно принять среднее взвешенное значение его оценки:

$$x_i = \sum_{h=1}^{l}\sum_{j=1}^{m} q_h x_{ij}^{h} k_j \ , \ (i=1,2...n) \quad (6)$$

где q_h – коэффициенты весов показателей сравнения объектов, k_j – коэффициенты компетентности экспертов. Величины q_h и k_j являются нормированными, то есть

$$\sum_{h=1}^{l} q_h = 1 \ , \ \sum_{j=1}^{m} k_j = 1 \, .$$

Коэффициенты q_h могут быть определены экспертным путем, как средний коэффициент веса h-ого показателя по всем экспертам, то есть

$$q_h = \sum_{j=1}^{m} q_{hj} k_j \ .$$

Коэффициенты компетентности экспертов можно вычислить по апостериорным данным, то есть по результатам оценки объектов. Основной идеей этого вычисления является предположение о том, что компетентность эксперта должна оцениваться по степени согласованности его оценок с групповой оценкой объектов.

Для упрощения дальнейшего изложения, ограничимся рассмотрением случая h=1. То есть когда групповое оценивание объектов проводится на основе только одного показателя. Алгоритм вычисления групповых оценок и коэффициентов компетентности экспертов для этого случая имеет вид:

а) начальные условия при t=0

$$k_j^0 = \frac{1}{m} \ (j = \overline{1,m}) \, ,$$

т.е. начальное значение коэффициентов компетентности для всех экспертов принимается одинаковым и равным.

б) рекуррентные соотношения для t=1,2,3 ...

$x_i^t = \sum_{j=1}^{m} x_{ij} k_j^{t-1}, \ (i = \overline{1,n})$ - групповая оценка для i-ого объекта на t-ом шаге на основе индивидуальных оценок x_{ij}.

$\lambda^t = \sum_{i=1}^{n} \sum_{j=1}^{m} x_i^t x_{ij}$ - нормировочный коэффициент

$k_j^t = \frac{1}{\lambda^t} \sum_{i=1}^{n} x_{ij} x_i^t, \ (j = \overline{1,i}$ - коэффициенты компетентности j-ого эксперта на t-ом шаге

$k_m^t = 1 - \sum_{j=1}^{m-1} k_j^t$ - коэффициенты компетентности m-ого эксперта из условия нормировки.

в) признак окончания итерационного процесса

$$\max(\left|x_i^t - x_i^{t-1}\right|) < E \ .$$

Сходимость данной итерационной процедуры доказана в литературе для случая, когда индивидуальные оценки неотрицательны, а эксперты и объекты не распадаются на отдельные группы (то есть когда каждая группа экспертов не оценивает объекты своей группы). В большинстве практических задач эти условия выполняются, что доказывает сходимость алгоритма.

Пример. Три эксперта (m=3) оценили значение двух мероприятий (n=3) по степени их влияния на решение одной из проблем (l=1). Результатами экспертизы явились нормированные оценки мероприятий $x_{1j}+x_{2j}=1$, j=1,2,3.

x_{ij}	Эксперт 1	Эксперт 2	Эксперт 3
Мероприятие 1	0,3	0,5	0,2
Мероприятие 2	0,7	0,5	0,8

Вычислим групповые оценки мероприятий, приводящих к решению проблемы и коэффициенты компетентности каждого из экспертов. Для этого воспользуемся приведенным выше алгоритмом, задавшись точностью вычисления E=0,001.

Средние оценки объектов первого приближения (при t=1) будут равны:

$$x_1^1 = \frac{1}{3}(0{,}3 + 0{,}5 + 0{,}2) = 0{,}333 \qquad x_2^1 = \frac{1}{3}(0{,}7 + 0{,}5 + 0{,}8) = 0{,}667,$$

$$x^1 = (0{,}333; 0{,}667).$$

Вычислим нормировочный коэффициент λ^1:

$$\lambda^1 = \sum_{i=1}^{2}\sum_{j=1}^{3} x_{ij} x_i^1 = x_1^1(0{,}3 + 0{,}5 + 0{,}2) + x_2^1(0{,}7 + 0{,}5 + 0{,}8) = 0{,}333 * 1 + 0{,}667 * 2 = 1{,}665$$

.

Значение коэффициентов компетентности первого приближения примут значения:

$$k_1^1 = \frac{1}{1{,}665}(0{,}3 * 0{,}333 + 0{,}7 * 0{,}667) = 0{,}34,$$

$$k_2^1 = \frac{1}{1{,}665}(0{,}5 * 0{,}333 + 0{,}5 * 0{,}667) = 0{,}30,$$

$$k_3^1 = 1 - (0{,}34 + 0{,}30) = 0{,}36$$

и тогда $k^1 = (0{,}34; 0{,}30; 0{,}36)$.

Вычисляя групповые оценки второго и т.д. приближения, получим:

$$x^2 = (0{,}324; 0{,}676)$$

$$\lambda^2 = 1{,}676$$

$$k^2 = (0{,}341; 0{,}298; 0{,}361)$$

$$x^3 = (0{,}3235; 0{,}6765)$$

$$\lambda^3 = 1{,}6765$$

$$k^3 = (0{,}341; 0{,}298; 0{,}361)$$

Результат третьего шага удовлетворяет условию окончания итерационного процесса и за значение групповой оценки принимается $x \approx x^3 = (0{,}3235;\ 0{,}6765)$.

Обработка парных сравнений

При установлении причинно-следственных зависимостей между объектами предметной области, экспертам в ряде случаев сложно выразить их численно. То есть трудно установить количественно степень влияния той или иной причины (объекта) на конкретное следствие. Особенно психологически это сложно, если таких объектов много.

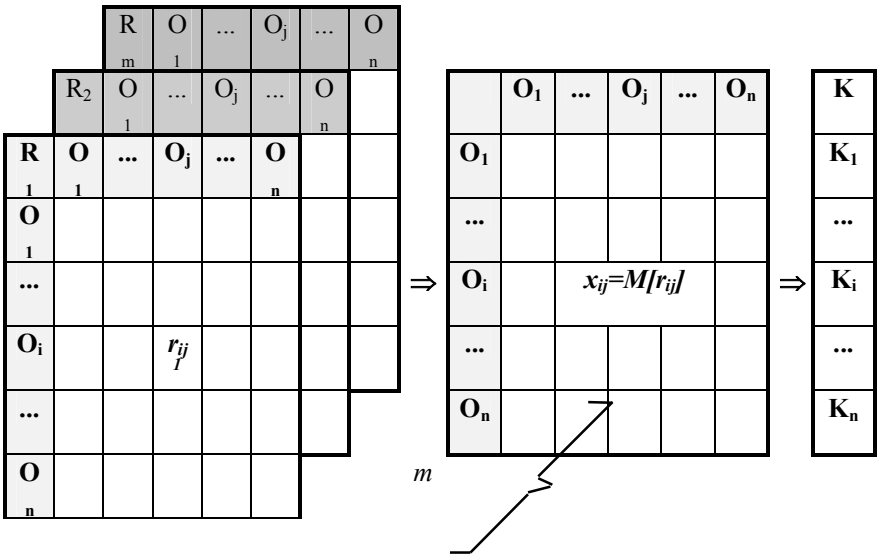
Вместе с тем, эксперты сравнительно легко решают задачу парного сравнения. Эта задача состоит в том, что эксперт устанавливает

предпочтения объектов при сравнении всех возможных пар. То есть эксперт, рассматривая все возможные пары объектов, в каждой из них устанавливает ту причину, которая, по его мнению, оказывает большое влияние на следствие. Возникает вопрос, как получить ***оценку всей совокупности объектов*** на основе результатов парного сравнения, выполненного группой экспертов.

Пусть каждый из ***m*** экспертов производит оценку влияния на результат всех пар объектов, давая числовую оценку:

$$r_{ij}^{h} = \begin{cases} 1 & \text{, если объект } O_i \text{ более значим, чем } O_j \\ 0{,}5 & \text{, объекты } O_i \text{ и } O_j \text{ равноправны} \\ 0 & \text{, если объект } O_i \text{ менее значим, чем } O_j \end{cases}$$

где h=1,2,...m – номер эксперта, i,j=1,2,...n – номера объектов, исследуемых при экспертизе. Т. е. по результатам экспертизы имеем m-таблиц (матриц) следующего вида:



Последовательность обработки парных сравнений заключается в том, что на основании таблиц парных сравнений m-экспертов строится матрица математических ожиданий оценок всех пар объектов.

Затем по этой матрице вычисляется вектор коэффициентов относительной важности объектов.

Если при оценке пары $\boldsymbol{O_{ij}}$ из общего количества экспертов m_i высказались в пользу предпочтение $\boldsymbol{O_i}$, $\boldsymbol{m_j}$ экспертов в пользу $\boldsymbol{O_j}$, а m_p считает эти объекты равноправными, то оценка математического ожидания дискретной случайной величины r_{ij} будет равна:

$$x_{ij} = M[r_{ij}^h] = 1*\frac{m_i}{m} + 0{,}5*\frac{m_p}{m} + 0*\frac{m_j}{m}\ ,\ h=\overline{1,m}.$$

Т.к. общее количество экспертов $m = m_i + m_p + m_j$, то определяя отсюда $\boldsymbol{m_p}$ и подставляя его в вышеприведенное выражение, получим:

$$x_{ij} = \frac{m_i}{m} + 0{,}5\left(\frac{m - m_i - m_j}{m}\right) = \frac{1}{2} + \frac{m_i - m_j}{2m}.$$

Очевидно, что $x_{ij} + x_{ji} = 1$. Совокупность величин $\boldsymbol{x_{ij}}$ образуют матрицу $\boldsymbol{X}=\|x_{ij}\|$ размерности n x n, на основе которой можно построить ранжировку всех объектов и определить коэффициенты относительной важности объектов, то есть вектор

$$k = [k_1, k_2, \ldots k_n]^T.$$

Одним из способов определения значений элементов вектора K является итерационный алгоритм вида:

а) начальное условие $t=0$

$$k^0 = [\underbrace{1\ 1\ 1\ \ldots\ 1}_{n}]^T$$

б) рекуррентные соотношения

$$k^t = \frac{1}{\lambda^t} * X * k^{t-1}$$

$$\lambda^t = [1\ 1\ 1\ \ldots\ 1] * X * k^{t-1},\ t = (1,2,\ \ldots\ n)$$

где $\boldsymbol{X}$ – матрица математических ожиданий оценок пар объектов, $\boldsymbol{k^t}$ – вектор

коэффициентов относительной важности объектов порядка $\boldsymbol{t}$.

$$\sum_{i=1}^{n} k_i^t = 1 \text{ – условие нормировки.}$$

в) признак окончания $\|k^t - k^{t-1}\| < E$.

Если матрица X неотрицательна и неразложима (то есть путем перестановки строк и столбцов ее нельзя привести к треугольному виду), то при увеличении порядка $t \to \infty$ величина λ^t сходится к максимальному собственному числу матрицы X, то есть

$$k = \lim_{t \to \infty} k^t \ , \ \sum_{i=1}^{n} k_i = 1 .$$

Это утверждение следует из теоремы Перрона-Фробениуса и доказывает сходимость приведенного выше алгоритма.

Пример. Предположим, что в результате опроса трех (m=3) экспертов о степени влияния на результат трех (n=3) различных факторов (объектов) получены следующие таблицы парных сравнений:

Эксперт 1(R_1)

	O_1	**O_2**	**O_3**
O_1	0,5	1	1
O_2	0	0,5	0
O_3	0	1	0,5

Эксперт 2(R_2)

	O_1	**O_2**	**O_3**
O_1	0,5	0,5	0,5
O_2	0,5	0,5	0,5
O_3	0,5	0,5	0,5

Эксперт 3(R_3)

	O_1	**O_2**	**O_3**
O_1	0,5	1	0,5
O_2	0	0,5	0
O_3	0,5	1	0,5

Для получения групповой оценки степени влияния каждого из объектов на результат, построим матрицу математических ожиданий оценок каждой из пар объектов, которая для рассматриваемого примера будет иметь вид:

	O_1	**O_2**	**O_3**
O_1	3/6	5/6	4/6
O_2	1/6	3/6	1/6
O_3	2/6	5/6	3/6

Значения элементов этой матрицы получены из следующих выражений:

$$x_{11} = \frac{1}{2} + \frac{0-0}{2*3} = \frac{1}{2} \quad x_{11} = x_{22} = x_{33} = \frac{3}{6} \ , \ x_{12} = \frac{1}{2} + \frac{2-0}{2*3} = \frac{5}{6} \quad x_{21} = 1 - x_{12} = \frac{1}{6}$$

$$x_{13} = \frac{1}{2} + \frac{1-0}{2*3} = \frac{4}{6} \quad x_{31} = 1 - x_{13} = \frac{2}{6} \quad , \ x_{23} = \frac{1}{2} + \frac{0-2}{2*3} = \frac{1}{6} \quad x_{32} = 1 - x_{23} = \frac{5}{6}$$

Воспользуемся вышеописанным алгоритмом для получения вектора относительной важности объектов. Для наглядности, каждый из шагов представим в виде:

шаг 0:

$$K^0 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$$

шаг 1:

$$Y^1 = X \times K^0 = \frac{1}{6} \times \begin{bmatrix} 3 & 5 & 4 \\ 1 & 3 & 1 \\ 2 & 5 & 3 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{6} \times \begin{bmatrix} 3+5+4 \\ 1+3+1 \\ 2+5+3 \end{bmatrix} = \frac{1}{6} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix}$$

$$\lambda^1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times Y^1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times \frac{1}{6} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \frac{1}{6} \times 27 = \frac{27}{6}$$

$$K^1 = \frac{1}{\lambda^1} \times Y^1 = \frac{6}{27} \times \frac{1}{6} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \frac{1}{27} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \begin{bmatrix} 0{,}444 \\ 0{,}185 \\ 0{,}370 \end{bmatrix}$$

шаг 2:

$$Y^2 = X \times K^1 = \frac{1}{6} \times \begin{bmatrix} 3 & 5 & 4 \\ 1 & 3 & 1 \\ 2 & 5 & 3 \end{bmatrix} \times \frac{1}{27} \times \begin{bmatrix} 12 \\ 5 \\ 10 \end{bmatrix} = \frac{1}{6 \times 27} \times \begin{bmatrix} 36+25+40 \\ 12+15+10 \\ 24+25+30 \end{bmatrix} = \frac{1}{6 \times 27} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix}$$

$$\lambda^2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times Y^2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times \frac{1}{6 \times 27} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix} = \frac{217}{6 \times 27}$$

$$K^2 = \frac{1}{\lambda^2} \times Y^2 = \frac{6 \times 27}{217} \times \frac{1}{6 \times 27} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix} = \frac{1}{217} \times \begin{bmatrix} 101 \\ 37 \\ 79 \end{bmatrix} = \begin{bmatrix} 0{,}465 \\ 0{,}171 \\ 0{,}364 \end{bmatrix}$$

$$\max\left(\left|0{,}465 - 0{,}444\right| , \left|0{,}171 - 0{,}185\right| , \left|0{,}364 - 0{,}3701\right| \right) = 0{,}021 > 0{,}001$$

Продолжая итерационный процесс до тех пор, пока норма оценки не будет меньше заданной ($\max\limits_i (|K_i^t - K_i^{t-1}|) < 0{,}001$) получим:

$$K^3 = \begin{bmatrix} 0{,}468 & 0{,}169 & 0{,}363 \end{bmatrix}^T$$

$$K^4 = \begin{bmatrix} 0{,}468 & 0{,}169 & 0{,}363 \end{bmatrix}^T$$

На четвертом шаге выполняется условие выхода, что позволяет за групповую оценку степени влияния на результат принять вектор коэффициентов относительной важности объектов вида:

$$K = \left[0{,}468 \quad 0{,}169 \quad 0{,}363\right]^{\mathrm{T}}$$

Определение обобщенных ранжировок

При групповой экспертной оценке каждому *i*-ому объекту каждый из *j*-ых экспертов присваивает r_{ij}. В результате проведения экспертного оценивания получается матрица рангов $\| \boldsymbol{r_{ij}} \|$ размерности n x m, где n – число объектов ($i=\overline{1,\mathrm{n}}$), а *m* – число экспертов ($j=\overline{1,\mathrm{m}}$).

Самый простейший способ получения *обобщенной ранжировки* заключается в ранжировании объектов по величине сумм рангов, полученных каждым объектов от всех экспертов. В этом случае для матрицы ранжировок $\|r_{ij}\|$ вычисляются суммы:

$$r_i = \sum_{j=1}^{m} r_{ij} \quad (i = 1,2,\ldots \mathrm{n}).$$

Далее объекты упорядочиваются по цепочке неравенств $r_k < r_l <..$ $.<r_q$, где $r_k = \min\limits_{i}(r_i)$, $r_l = \min\limits_{i,\, i\neq k}(r_i)$, ..., $r_q = \max\limits_{i}(r_i)$. Отсюда следует обобщенная ранжировка объектов:

$$\mathbf{O_k} \ \phi \ \mathbf{O_l} \ \phi \ \ldots \ \phi \ \mathbf{O_q}.$$

Для учета компетентности экспертов достаточно умножить *i*-ю ранжировку на коэффициенты компетентности *j*-го эксперта $0 \le k_j \le 1$. В этом случае вычисление суммы рангов для *i*-ого объекта производится по формуле:

$$r_i = \sum_{j=1}^{m} k_j r_{ij} \ ,$$

что позволяет упорядочить объекты по цепочке неравенств. Следует отметить, что построение таких обобщенных ранжировок является корректной процедурой только в том случае, если ранги назначаются как места объектов в виде натуральных чисел 1,2,...,*n*.

Однако ранги объектов определяют только порядок расположение объектов по показателям сравнения. Ранги как числа не дают возможность сделать вывод о том, на сколько или во сколько раз предпочтительнее один объект по сравнению с другим. Если ранг 3, то отсюда не следует делать вывод о том, что объект, с рангом 1, в три

раза предпочтительнее, чем объект, имеющий ранг, равный трем.

Вместе с тем для использования в ЭС знаний, полученных от экспертов, необходимо не только упорядочение или ранжирование объектов по степени их влияния или воздействия на какой-либо результат, но и определение количественной оценки степени влияния каждого из объектов на результат.

Простейшим методом для реализации этой задачи является подход, основанный на построении обобщенной ранжировки путем перехода от матрицы ранжировок к матрице парных сравнений. Для этого на основе матрицы $\| r_{ij} \|$ строится m матриц парных сравнений R_j (j=1,2,...,m), где m – число экспертов. Элементы этих матриц определяются следующим образом:

$$R_j = \left\| r_{ik}^j \right\| = \begin{cases} 1 & \text{, если } \mathbf{O}_i^j \ \phi \ \mathbf{O}_k^j, \text{ то есть } r_{ij} < r_{kj} \\ 0{,}5 & \text{, если } \mathbf{O}_i^j \sim \mathbf{O}_k^j, \text{ то есть } r_{ij} = r_{kj} \\ 0 & \text{, если } \mathbf{O}_i^j \ \pi \ \mathbf{O}_k^j, \text{ то есть } r_{ij} > r_{kj} \end{cases}$$

где j – номер эксперта, i и k – номера сравниваемых объектов.

Затем к полученным матрицам парных сравнений всех экспертов применяется рассмотренный ранее метод обработки парных сравнений. Его итерационная процедура позволяет получить *коэффициенты относительной важности* объектов по степени их влияния на результат. Проиллюстрируем применение этого подхода на примере.

Пример. Пусть три эксперта (m=3) провели ранжировку трех объектов (n=3) по степени их влияния на какой-либо результат и таблица ранжировок имеет вид:

Объект O_i	Эксперт 1	Эксперт 2	Эксперт 3
O_1	1	1	2
O_2	2	3	1
O_3	3	2	3

На основе этой таблицы матрица парных сравнений для первого эксперта будет иметь вид:

$$R_1 = \left\| R_{kj}^1 \right\| = \begin{array}{c} \begin{matrix} O_1 & O_2 & O_3 \end{matrix} \\ \left\| \begin{matrix} 0{,}5 & 1 & 1 \\ 0 & 0{,}5 & 1 \\ 0 & 0 & 0{,}5 \end{matrix} \right\| \end{array} \begin{matrix} O_1 \\ O_2 \\ O_3 \end{matrix}$$

Аналогичные матрицы парных сравнений для второго и третьего эксперта будут иметь вид:

$$R_2 = \begin{Vmatrix} 0{,}5 & 1 & 1 \\ 0 & 0{,}5 & 0 \\ 0 & 1 & 0{,}5 \end{Vmatrix}; \; R_3 = \begin{Vmatrix} 0{,}5 & 0 & 1 \\ 1 & 0{,}5 & 1 \\ 0 & 0 & 0{,}5 \end{Vmatrix}$$

Используя метод обработки парных сравнений, получим последовательность векторов коэффициентов относительной важности объектов:

Шаг	K_1	K_2	K_3
0	1,0	1,0	1,0
1	0,481	0,330	0,185
2	0,489	0,346	0,156
3	0,5	0,348	0,152
4	0,5	0,349	0,151

Итерационная процедура с заданной точностью (E=0,001) является сходящейся на четвертом шаге к значениям:

$$K = \begin{bmatrix} 0{,}500 & 0{,}349 & 0{,}151 \end{bmatrix}^{\mathrm{T}},$$

что позволяет оценить количественно степень влияния каждого объекта на результат, полученный на основе исходного ранжирования экспертов.

Все рассмотренные методы получения групповых оценок позволяют получить достоверные результаты в случае хорошо подобранной группы экспертов и согласованности их мнений. Если это не так, то встает задача определения количественной оценки степени согласованности экспертов. Получение количественной меры позволяет более обоснованно интерпретировать причины в расхождении мнений.

Для оценки меры согласованности мнений группы экспертов используют, в частности, *дисперсионный и энтропийный коэффициенты конкордации*. Кроме этого, при обработке результатов ранжирования могут возникать задачи:

определения зависимости между ранжировками двух экспертов;

связи между достижением двух различных целей при решении одной и той же совокупности проблем;

взаимосвязи между признаками (объектами).

В этих случаях мерой взаимосвязи может служить *коэффициент ранговой корреляции*. Характеристикой взаимосвязи множества ранжировок будет являться матрица коэффициентов ранговой корреляции. Известны коэффициенты ранговой корреляции Спирмена и Кендалла.

2.7. Экспертные системы с неопределенными знаниями

Неопределенности в ЭС и проблемы порождаемые ими

В жизни часто приходится оценивать гипотезы, для которых имеется неполная или недостаточная информация. Иногда трудно сделать точные оценки, но, несмотря на неопределенность, мы принимаем разумные решения. Чтобы ЭС были полезными, они тоже должны уметь это делать. Классическим примером этой задачи является медицинская диагностика. Всегда существуют некоторые сомнения в четкости проявления симптомов того или иного заболевания. Сомнения в наличии у пациента конкретного заболевания сохраняются даже в том случае, когда все его симптомы отчетливо выражены.

Как же проявляется и учитывается неопределенность в экспертных системах? Рассмотрим простейшую ситуацию. Пусть используется правило:

если (A), то (B)

и предположим никакие другие правила и посылки не имеют отношения к рассматриваемой ситуаций. Где же возникает неопределенность? В ЭС она может быть двух типов:

неопределенность в *истинности самой посылки* (например, если степень уверенности в том, что *A* истинно составляет 90%, то какие значения примет *B*);

неопределенность *самого правила* (например, мы можем сказать, что в большинстве случаев, но всегда, если есть *A*, то есть также и *B*).

Еще более сложная ситуация возникает в случае, если правило имеет вид:

если (A и B), то C,

где мы можем с некоторой степенью быть уверены как в истинности каждой из посылок (*A*, *B*), а тем более их совместного проявления, так и в истинности самого вывода. Существуют четыре важные проблемы, которые возникают при проектировании и создании ЭС с неопределенными знаниями.

1.Как количественно выразить степень определенности при установлении истинности (или ложности) некоторой части данных?

2. Как выразить степень поддержки заключения конкретной посылкой?

3. Как использовать совместно две (или более) посылки, независимо влияющие на заключение?

4. Как быть в ситуации, когда нужно обсудить цепочку вывода для подтверждения заключения в условиях неопределенности?

Прежде всего, рассмотрим возможности использования теории вероятности при вводе в условиях неопределенности.

Теория субъективных вероятностей

Основное понятие вероятности настолько естественно, что оно играет значительную роль в повседневной жизни. Разговоры, касающиеся вероятности дождя или хорошего урожая в огороде, часто встречаются в нашей жизни. Понятие вероятности было разработано несколько столетий назад. Но уже тысячи лет человек использует такие слова, как «может быть», «шанс», «удача» или иные их эквиваленты в разговорном языке.

Вероятность события классически определяется как отношение случаев, в которых данное событие происходит к общему числу наблюдений.

Однако возможны и другие определения. В настоящее время существует несколько интерпретаций теории вероятностей. Рассмотрим три наиболее доминирующих взгляда.

Объективистский взгляд. Заключается в том, что рассматривает вероятность отношения исходов ко всем наблюдениям в течение длительного времени. Другими словами, этот подход основан на законе больших чисел, гарантирующим то, что при наличии достаточно большого количества наблюдений частота исходов, интересующего события будет стремиться к объективной вероятности.

Персонофицированный, субъективисткий или основанный на суждениях взгляд заключается в том, что вероятностная мера рассматривается как степень доверия того, как отдельная личность судит об истинности некоторого высказывания. Этот взгляд постулирует, что данная личность имеет в некотором смысле отношение к этому событию. Но это не отрицает возможности того, что две приемлемые личности могут иметь различные степени доверия для одного и того же суждения. Термин «байесовский» часто используется как синоним субъективной вероятности.

Необходимый или логический характеризуется тем, что вероятностная мера расширяется на множество утверждений, имеющих логическую связь такую, что истинность одного из них может выводиться из другого. Другими словами, вероятность измеряет степень доказуемости логически выверенного заключения. Такой взгляд можно рассматривать как расширение обычной логики.

Эти вероятностные интерпретации используют и различные схемы вывода. Однако существует всего две школы вероятностных расчетов: школа Паскаля (или общепринятая), школа Бэкона (или индуктивная). Расчеты по Паскалю используют байесовские правила для проверки и обработки мер доверия. Вычисления по Бэкону используют правила логики для доказательства или опровержения гипотез. Таким образом, общепринятые вероятности (по Паскалю) не могут быть получены из индуктивных вероятностей (по Бэкону) и, наоборот. Объективистский и субъективный взгляды используют расчеты по Паскалю. Те, кто поддерживают логические выводы, используют расчеты по Бэкону.

Существуют ЭС, построенные на обоих из этих направлений. Однако в ЭС базы знаний накапливают человеческие знания, поэтому для представления знаний экспертов с учетом вероятностей наиболее подходящими являются интерпретация на основе субъективных доверий. В результате чего и большинство современных ЭС, использующих теорию вероятностей, являются «байесовскими».

Байесовское оценивание

Перед тем, как ввести теорему Байеса, рассмотрим некоторые фундаментальные понятия теории вероятностей. Пусть A некоторое событие реального мира. Совокупность всех элементарных событий называется выборочным пространством или пространство событий (Ω). Вероятность события A обозначается $p(A,)$ и каждая вероятностная функция p должна удовлетворять трем аксиомам:

1. Вероятность любого события А является неотрицательной, т.е.

$$p(A) \geq 0 \quad для \quad \forall A \in \Omega$$

2. Вероятность всех событий выборочного пространства равна 1, т.е.

$$p(\Omega) = 1.$$

3. Если k событий $A_1, A_2, \ldots, A_k$ являются взаимно независимыми (т.е. не могут подойти одновременно), то вероятность, по крайней мере, одного из этих событий равна сумме отдельных вероятностей, или

$$p(A_1 \cup A_2 \cup \quad \ldots \quad \cup A_k) \quad = \quad \sum_{i=1}^{k} p(A_i)$$

Аксиомы 1 и 2 можно объединить, что дает

$$1 \geq p(A) \geq 0 \quad для \quad \forall A \in \Omega .$$

Это утверждение показывает, что вероятность любого события находится между 0 и 1. По определению, когда $p(A) = 0$, то событие А никогда не произойдет. В том случае и когда $p(A) = 1$, то событие А должно произойти обязательно.

Дополнение к A, обозначаемое $(\neg A)$, содержит совокупность всех событий в Ω за исключением A. Т.к. A и $\neg A$ являются взаимонезависимыми (т.е. $A \cup \neg A = \Omega$), то из аксиомы 3 следует:

$$p(A) + p(\neg A) = p(A \cup \neg A) = p(\Omega) = 1 .$$

Переписывая это равенство в виде $p(\neg A) = 1 - p(A)$, мы получает путь для получения $p(\neg A)$ из $p(A)$.

Предположим теперь, что $B \in \Omega$ некоторое другое событие. Тогда вероятность того, что произойдет A при условии, что произошло B записывается в виде $p(A \mid B)$ и называется *условной вероятностью* события A при заданном событии B.

Вероятность того, что оба события A и B произойдут $p(A \cap B)$ называется *совместной вероятностью событий* A и B. Условная вероятность $p(A|B)$ равна отношению совместной вероятности $p(A \cap B)$ к вероятности события В, при условии, что она не равна 0, т. е.

$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

Аналогично условная вероятность события B при условии A, обозначаемая $p(B \mid A)$ равна:

$$p(B|A) = \frac{p(B \cap A)}{p(A)}$$

и таким образом

$$p(B \cap A) = p(B|A) \times p(A) .$$

Так, как совместная вероятность коммутативна (т.е. от перестановки мест сумма не меняется), то

$$p(A \cap B) = p(B \cap A) = p(B|A) \times p(A) .$$

Подставляя это равенство в ранее полученное выражение для условной вероятности $p(A| B)$ получим правило Байеса:

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B)} .$$

В ряде случае наше знание того, что произошло событие В, не влияет на вероятность события А (или наоборот А на В). Другими

словами, вероятность события А не зависит от того, что произошло или нет событие В, так что

$$p(A \mid B) = p(A) \text{ и } p(B \mid A) = p(B) .$$

В этом случае говорят, что события *A* и *B* являются независимыми.

Теорема Байеса как основа управления неопределенностью

Приведенные выше соотношения предполагают определенную связь между теорией вероятностей и теорией множеств. Если *A* и *B* являются непересекающимися множествами, то объединение множеств соответствует сумме вероятностей, а пересечение – произведению вероятностей, т. е.

$$p(A \cup B) = p(A) + p(B) \text{ и } p(A \cap B) = p(A) * p(B)$$

Без предположения независимости эта связь является неточной и формулы должны содержать дополнительные члены включения и исключения (так например, $p(A \cup B)=p(A)+p(B)-p(A \cap B)$). Продолжая теоретико – множественное обозначение *B* можно записать как:

$$B = (B \cap A) \cup (B \cap \neg A)$$

Так как это объединение явно непересекающееся, то:

$$p(B)=p((B \cap A) \cup (B \cap \neg A))=p(B \cap A)+ p(B \cap \neg A) = p(B|A)\, p(A) + p(B|\neg A)p(B)$$

Возвращаясь к обозначению событий, а не множеств, последнее равенство может быть подставлено в правило Байеса:

$$p(A|B) = \frac{p(B|A) \times p(A)}{p(B|A) \times p(A) + P(B|\neg A) \times p(\neg A)} .$$

Это равенство является основой для использования теории вероятности в управлении неопределенностью. Оно обеспечивает путь для получения условной вероятности события *B* при условии *A*. Это соотношение позволяет ЭС управлять неопределенностью и «делать вывод вперед и назад».

2.8. Логический вывод на основе субъективной вероятности

Простейший логический вывод

Рассмотрим случай, когда все правила в экспертной системе отражаются в форме:

если < *H* является истинной >, то <*E* будет наблюдаться с вероятностью *p*>.

Очевидно, если H произошло, то это правило говорит о том, что событие E происходит с вероятностью p. Но что будет, если состояние H неизвестно, а E произошло? Использование теоремы Байеса позволяет вычислить вероятность того, что H истинно. Замена «A» и «B» на «H» и «E» не существенна для формулы Байеса, но с её помощью мы можем покинуть общую теорию вероятности и перейти к анализу вероятностных вычислений в ЭС. В этом контексте:

H – событие, заключающееся в том, что данная гипотеза верна;

E – событие, заключающееся в том, что наступило определённое доказательство (свидетельство), которое может подтвердить правильность указанной гипотезы.

Переписывая формулу Байеса в терминах гипотез и свидетельств, получим:

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E|H) \times p(H) + p(E|\neg H) \times p(\neg H)} .$$

Это равенство устанавливает связь гипотезы со свидетельством и, в то же время наблюдаемого свидетельства с пока ещё не подтверждённой гипотезой. Эта интерпретация предполагает также определение *априорной вероятности* гипотезы $p(H)$, назначаемой H до наблюдения или получения некоторого факта.

В экспертных системах вероятности, требуемые для решения некоторой проблемы, обеспечивается экспертами и запоминается в базе знаний. Эти вероятности включают:

априорные вероятности всех возможных гипотез $p(H)$;

условные вероятности возникновения свидетельств при условии существования каждой из гипотез $p(E \mid H)$.

Так, например, в медицинской диагностике эксперт должен задать априорные вероятности всех возможных болезней в некоторой медицинской области. Кроме того, должны быть определены условные вероятности проявления тех или иных симптомов при каждой из болезней. Условные вероятности должны быть получены для всех симптомов и болезней, предполагая, что все симптомы независимы в рамках одной болезни.

Два события E_1 и E_2 являются условно независимыми, если их совместная вероятность при условии некоторой гипотезы H равна произведению условных вероятностей эти событий при условии H, то есть

$$p(E_1 E_2 \mid H) = p(E_1 \mid H) \cdot p(E_2 \mid H).$$

Пользователи дают ЭС информацию о наблюдениях (наличии определённых симптомов) и ЭС вычисляет $p(H_i|E_j \dots E_k)$ для всех гипотез ($H_1, \dots, H_m$) в свете предъявленных симптомов ($E_j, \dots, E_k$) и вероятностях, хранимых в БЗ.

Вероятность $p(H_i \mid E_j \dots E_k)$ называется *апостериорной вероятностью* гипотез H_i по наблюдениям ($E_j, \dots, E_k$). Эти вероятности дают сравнительное ранжирование всех возможных гипотез, то есть гипотез с ненулевыми апостериорными вероятностями. Результатом вывода ЭС является выбор гипотезы с наибольшей вероятностью.

Однако приведённая выше формула Байеса ограничена в том, что каждое свидетельство влияет только на одну гипотезу. Можно обобщить это выражение на случай множественных гипотез ($H_1, \dots, H_m$) и множественных свидетельств ($E_1, \dots, E_n$). Вероятности каждой из гипотез при условии возникновения некоторого конкретного свидетельства E можно определить из выражения:

$$p(H_i|E) = \frac{p(E|H_i) \times p(H_i)}{\sum_{k=1}^{m} p(E|H_k) \times p(H_k)}, \qquad i = \overline{1,m} \ .$$

а в случае множественных свидетельств:

$$p(H_i|E_1E_2\text{K}\ E_n) = \frac{p(E_1E_2\text{K}\ E_n|H_i) \times p(H_i)}{\sum_{k=1}^{m} p(E_1E_2\text{K}\ E_n|H_k) \times p(H_k)}, \qquad i = \overline{1,m} \ .$$

К сожалению, данное выражение имеет ряд недостатков. Так, знаменатель требует от нас знания условных вероятностей всех возможных комбинаций свидетельств и гипотез, что делает правило Байеса малопригодным для ряда приложений. Однако в тех случаях, когда возможно предположить условную независимость свидетельств, правило Байеса можно привести к более простому виду:

$$p(H_i|E_1E_2\text{K}\ E_n) = \frac{p(E_1|H_i) \times p(E_2|H_i) \times \text{K} \times p(E_n|H_i) \times p(H_i)}{\sum_{k=1}^{m} p(E_1|H_k) \times p(E_2|H_k) \times \text{K} \times p(E_n|H_k) \times p(H_k)}, \qquad i = \overline{1,m}$$

.

Вместе с тем предположения о независимости событий в ряде случаев подавляют точности суждений и свидетельств в ЭС.

Распространение вероятностей в ЭС

Вероятности событий распространяются по БЗ экспертной системы на основе правила Байеса для вычисления всех апостериорных вероятностей гипотез при условии наблюдаемых свидетельств. Эти апостериорные вероятности дают ранжированную информацию о потенциально истинной гипотезе. Рассмотрим пример, иллюстрирующий этот процесс.

Пример. Предположим, что в некоторой БЗ имеется всего три взаимно независимых гипотезы: H_1, H_2, H_3, которые имеют априорные вероятности: $p(H_1)$, $p(H_2)$, $p(H_3)$, соответственно. Правила БЗ содержат два условно независимых свидетельства, которые поддерживают исходные гипотезы в различной степени. Априорные и условные вероятности всех гипотез и свидетельств этого примера имеют следующие значения:

p() i	1	2	3
$p(H_i)$	0,5	0,3	0,2
$p(E_1\|H_i)$	0,4	0,8	0,3
$p(E_2\|H_i)$	0,7	0,9	0,0

При этом исходные гипотезы характеризуют событие, связанное с определением надежности некоторой фирмы:

H_1 – средняя надежность фирмы,
H_2 – высокая надежность фирмы,
H_3 – низкая надежность фирмы.

Событиями, являющимися условно независимыми свидетельствами, поддерживающими исходные гипотезы являются: E_1 – наличие прибыли у фирмы и E_2 – своевременный расчет с бюджетом.

В процессе сбора фактов вероятности гипотез будут повышаться, если факты поддерживают их или уменьшаться, если опровергают их. Предположим, что мы имеем только одно свидетельство E_1 (то есть с вероятностью единица наступил факт E_1). Наблюдая E_1, мы вычисляем апостериорные вероятности для гипотез согласно формуле Байеса для одного свидетельства:

$$p(H_i|E_1) = \frac{p(E_1|H_i) \times p(H_i)}{\sum_{k=1}^{3} p(E_1|H_k) \times p(H_k)}, \qquad i = 1,2,3 \ .$$

Таким образом:

$$p(H_1|E_1) = \frac{0,4 \times 0,5}{0,4 \times 0,5 + 0,8 \times 0,3 + 0,3 \times 0,2} = 0,40 \quad ,$$

$$p(H_2|E_1) = \frac{0{,}8 \times 0{,}3}{0{,}4 \times 0{,}5 + 0{,}8 \times 0{,}3 + 0{,}3 \times 0{,}2} = 0{,}48 \quad ,$$

$$p(H_3|E_1) = \frac{0{,}3 \times 0{,}2}{0{,}4 \times 0{,}5 + 0{,}8 \times 0{,}3 + 0{,}3 \times 0{,}2} = 0{,}12 \quad .$$

После того как E_1 произошло доверие к гипотезам H_1 и H_3 понизилось, в то время как доверие к H_2 возросло. В тех случаях, когда имеются факты, подтверждающие как событие E_1, так и событие E_2, то апостериорные вероятности исходных гипотез также могут быть вычислены по правилу Байеса:

$$p(H_i|E_1E_2) = \frac{p(E_1E_2|H_i) \times p(H_i)}{\sum_{k=1}^{3} p(E_1E_2|H_k) \times p(H_k)}, \qquad i = 1{,}2{,}3 \cdot$$

Так как события E_1 и E_2 условно независимые при данных гипотезах H_i, то формулу Байеса можно переписать в виде:

$$p(H_i|E_1E_2) = \frac{p(E_1|H_i) \times p(E_2|H_i) \times p(H_i)}{\sum_{k=1}^{3} p(E_1|H_k) \times p(E_2|H_i) \times p(H_k)}, \qquad i = 1{,}2{,}3 \cdot$$

Откуда:

$$p(H_1|E_1E_2) = \frac{0{,}4 \times 0{,}7 \times 0{,}5}{0{,}4 \times 0{,}7 \times 0{,}5 + 0{,}8 \times 0{,}9 \times 0{,}3 + 0{,}3 \times 0{,}0 \times 0{,}2} = 0{,}393 \quad ,$$

$$p(H_2|E_1E_2) = \frac{0{,}8 \times 0{,}9 \times 0{,}3}{0{,}4 \times 0{,}7 \times 0{,}5 + 0{,}8 \times 0{,}9 \times 0{,}3 + 0{,}3 \times 0{,}0 \times 0{,}2} = 0{,}607 \quad ,$$

$$p(H_3|E_1E_2) = \frac{0{,}3 \times 0{,}9 \times 0{,}2}{0{,}4 \times 0{,}7 \times 0{,}5 + 0{,}8 \times 0{,}9 \times 0{,}3 + 0{,}3 \times 0{,}0 \times 0{,}2} = 0{,}0 \quad .$$

Хотя исходным ранжированием было H_1, H_2, и H_3, только H_1 и H_2 остались после получения свидетельств E_1 и E_2. При этом H_1, более вероятно, чем H_2. На этом примере мы рассмотрели процесс распространения вероятностей по элементам ЭС при поступлении в неё тех или иных свидетельств.

Последовательное распространение вероятностей

Однако реально распространение вероятностей происходит поэтапно с суммированием отдельных свидетельств и их влияния на условную вероятность по мере поступления отдельных E_i. Это можно сделать, используя априорные и апостериорные вероятности, следующим образом:

1. Задаём $p(H_i)$ – априорную вероятность событий H_i.
2. Для полученных свидетельств E_j записываем $p(E_j \mid H_i)$.
3. С учётом теоремы Байеса подсчитываем $p(H_i \mid E_j)$ в зависимости от исхода E_j, то есть вычисляем апостериорную вероятность события H_i.
4. Теперь можно не обращать внимания на все наступившие E_j и переобозначить текущую апостериорную вероятность события H_i, как новую априорную вероятность H_i. Итак, пусть $p(H_i)$ равна $p(H_i|E_j)$ в зависимости от значения E_j.
5. Затем выберем новое свидетельство для рассмотрения и перейдём к п.2.

Проиллюстрируем эту последовательность на приведенном выше примере в предположении, что сначала поступило свидетельство E_2. Тогда

$$p(H_1|E_2) = \frac{0{,}7 \times 0{,}5}{0{,}7 \times 0{,}5 + 0{,}9 \times 0{,}3 + 0{,}0 \times 0{,}2} = 0{,}565 \quad ,$$

$$p(H_2|E_2) = \frac{0{,}9 \times 0{,}3}{0{,}7 \times 0{,}5 + 0{,}9 \times 0{,}3 + 0{,}0 \times 0{,}2} = 0{,}435 \quad ,$$

$$p(H_3|E_2) = \frac{0{,}0 \times 0{,}2}{0{,}7 \times 0{,}5 + 0{,}9 \times 0{,}3 + 0{,}0 \times 0{,}2} = 0{,}0 \quad .$$

Полученные вероятности можно принять за новые апостериорные вероятности гипотез H_1, H_2, и H_3, то есть

$$p(\tilde{H}_1) = 0{,}565 \ , \quad p(\tilde{H}_2) = 0{,}435 \ , \quad p(\tilde{H}_3) = 0{,}0$$

И если теперь дополнительно поступит свидетельство E_2, то новые апостериорные вероятности гипотез могут быть вычислены только на основе вновь поступившего свидетельства:

$$p(H_1|E_1E_2) = p(\tilde{H}_1|E_1) = \frac{0{,}4 \times 0{,}565}{0{,}4 \times 0{,}565 + 0{,}8 \times 0{,}435 + 0{,}3 \times 0{,}0} = 0{,}393 \quad ,$$

$$p(H_2|E_1E_2) = p(\tilde{H}_2|E_2) = \frac{0{,}8 \times 0{,}435}{0{,}4 \times 0{,}565 + 0{,}8 \times 0{,}435 + 0{,}3 \times 0{,}0} = 0{,}607 \quad ,$$

$$p(H_3|E_1E_2) = p(\tilde{H}_3|E_2) = \frac{0{,}3 \times 0{,}0}{0{,}4 \times 0{,}565 + 0{,}8 \times 0{,}435 + 0{,}3 \times 0{,}0} = 0{,}0 \quad .$$

Из приведенного примера видно, что итерационная процедура последовательного распределения вероятностей по мере поступления свидетельств позволяет получить результаты аналогичные непосредственному применению правила Байеса для случая одновременного двух поступивших свидетельств.

2.9. Байесовские сети доверия как средства разработки экспертных систем

Основные понятия и определения

Сети доверия — это новый инструмент для решения перечисленных проблем, в которых регулируются информационные потоки. При представлении информации в базе знаний при помощи байесовских сетей можно создать согласованную и непротиворечивую *вероятностную базу знаний* без необходимости в предположении условной независимости.

Байесовские сети доверия – Bayesian Belief Network – используются в тех областях, которые характеризуются наследованной неопределённостью. Эта неопределённость может возникать вследствие:

неполного понимания предметной области;

неполных знаний;

когда задача характеризуется случайностью.

Таким образом, байесовские сети доверия (БСД) применяют для моделирования ситуаций, содержащих неопределённость в некотором смысле. Для байесовских сетей доверия иногда используется ещё одно название *причинно-следственная сеть*, в которых случайные события соединены причинно-следственными связями.

Соединения методом причин и следствий позволяют более просто оценивать вероятности событий. В реальном мире оценивание наиболее часто делается в направлении от “наблюдателя” к “наблюдению”, или от “эффекта” к “следствию”, которое в общем случае более сложно оценить, чем направление “следствие –> эффект”, то есть в направлении от следствии.

Рассмотрим пример сети (рисунок 2.9), в которой вероятность пребывания вершины «**e**» в различных состояниях (e_k) зависит от состояний (c_i, d_j) вершин «**c**» и «**d**» и определяется выражением:

$$p(e_k) = \sum_i \sum_j p(e_k \mid c_i, d_j) \times p(c_i, d_j)$$

где $p(e_k \mid c_i, d_j)$ – вероятность пребывания в состоянии e_k в зависимости от состояний c_i, d_j. Так как события, представленные вершинами «**c**» и «**d**» независимы, то:

$$p(e_k \mid c_i, d_j) = p(c_i) \cdot p(d_j).$$

Рассмотрим пример более сложной сети (рисунок 2.10). Данный рисунок иллюстрирует условную независимость событий. Для оцен-

ки вершин «**c**» и «**d**» используются те же выражения, что и для вычисления $p(e_k)$, тогда

$$p(c_i) = \sum_m \sum_n p(c_i | A_{1m}, B_{1n}) \times p(A_{1m}) \times p(B_{1n}),$$

$$p(d_j) = \sum_m \sum_n p(d_j | A_{2m}, B_{2n}) \times p(A_{2m}) \times p(B_{2n}).$$

Из этих выражений видно, что вершина «**e**» условно не зависит от вершин A_1, A_2, B_1, B_2, так как нет стрелок непосредственно соединяющих эти вершины.

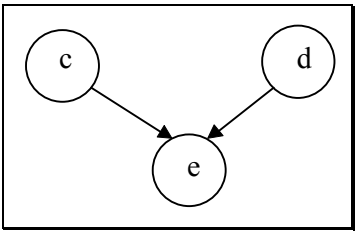


Рисунок 2.9. Пример простейшей байесовской сети доверия.

Рассмотрев эти примеры, попробуем теперь более точно определить основные понятия, используемые в БСД. *Байесовские сети доверия* — это направленный ациклический граф, обладающий следующими свойствами:

каждая вершина представляет собой событие, описываемое случайной величиной, которая может иметь несколько состояний;

все вершины, связанные с "родительскими" определяются таблицей условных вероятностей (ТУВ) или функцией условных вероятностей (ФУВ);

для вершин без "родителей" вероятности её состояний являются безусловными (маргинальными).

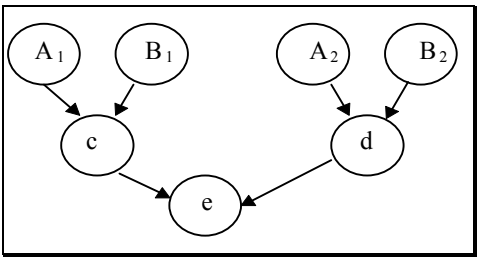


Рисунок 2.10. Двухуровневая БСД.

Другими словами, в байесовских сетях доверия вершины представляют собой случайные переменные, а дуги – вероятностные зависимости, которые определяются через таблицы условных вероятностей. Таблица условных вероятностей каждой вершины содержит вероятности состояний этой вершины при условии состояний её "родителей".

Пример построения простейшей байесовской сети доверия

В этом примере рассматриваем небольшую яблочную плантацию «яблочного Джека». Однажды Джек обнаружил, что его прекрасное яблочное дерево лишилось листвы. Теперь он хочет выяснить, почему это случилось. Он знает, что листва часто опадает, если:

- дерево засыхает в результате недостатка влаги;
- или дерево болеет.

Данная ситуация может быть смоделирована байесовской сетью доверия, содержащей 3 вершины: «Болеет», «Засохло» и «Облетело».

Рисунок 2.11. Пример байесовской сети доверия с тремя событиями.

В данном простейшем случае рассмотрим ситуацию, при которой каждая вершина может принимать всего лишь два возможных состояний и, как следствие находится в одном из них, а именно:

Вершина (событие) БСД	Состояние 1	Состояние 2
“Болеет”	« болеет»	« нет»
“Засохло”	« засохло»	« нет»
“Облетело”	« да»	« нет»

Вершина “Болеет» говорит о том, что дерево заболело, будучи в состоянии «болеет», в противном случае она находится в состоянии «нет». Аналогично для других двух вершин. Рассматриваемая байесовская сеть доверия, моделирует тот факт, что имеется причинно-следственная зависимость от события “Болеет» к событию “Облетело» и от события “Засохло” к событию “Облетело”. Это отображено стрелками на байесовской сети доверия.

Когда есть причинно-следственная зависимость от вершины А к другой вершине В, то мы ожидаем, что когда А находится в некотором определённом состоянии, это оказывает влияние на состояние В. Следует быть внимательным, когда моделируется зависимость в байесовских сетях доверия. Иногда совсем не очевидно, какое направление должна иметь стрелка.

Например, в рассматриваемом примере, мы говорим, что имеется зависимость от “Болеет» к “Облетело”, так как когда дерево болеет, это может вызывать опадание его листвы. Опадание листвы является следствием болезни, а не болезнь – следствием опадания листвы.

На приведенном выше рисунке дано графическое представление байесовской сети доверия. Однако это только качественное представление байесовской сети доверия. Перед тем, как назвать это полностью байесовской сетью доверия необходимо определить *количественное* представление, то есть множество таблиц условных вероятностей:

Априорная вероятность *p*(“Болеет”)

Болеет = « болеет»	Болеет = «нет»
0,1	0,9

Априорная вероятность *p*(“Засохло”)

Засохло = « засохло»	Засохло = «нет»
0,1	0,9

Таблица условных вероятностей *p*(“Облетело”|”Болеет”, ”Засохло”)

	Засохло = «засохло»		Засохло = «нет»	
	Болеет = «болеет»	Болеет = «нет»	Болеет = «болеет»	Болеет = «нет»
Облетело = «да»	0,95	0,85	0,90	0,02
Облетело = «нет»	0,05	0,15	0,10	0,98

Приведенные таблицы иллюстрируют ТУВ для трёх вершин байесовской сети доверия. Заметим, что все три таблицы показывают вероятность пребывания некоторой вершины в определённом состоянии, обусловленным состоянием её родительских вершин. Но так как вершины Болеет и Засохло не имеют родительских вершин, то их вероятности являются маргинальными, т.е. не зависят (не обусловлены) ни от чего.

На данном примере мы рассмотрели, что и как описывается очень простой байесовской сетью доверия. Современные программные средства (такие, как MSBN, Hugin и др.) обеспечивают инструментарий для построения таких сетей, а также возможность использования байесовских сетей доверия для введения новых свидетельств и получения решения (вывода) за счёт пересчёта новых вероятностей во всех вершинах, соответствующих вновь введенным свидетельствам.

В нашем примере пусть известно, что дерево сбросило листву. Это свидетельство вводится выбором состояния «да» в вершине

“Облетело”. После этого можно узнать вероятности того, что дерево засохло. Для приведенных выше исходных данных, результаты вывода путем распространения вероятностей по БСД будут: *p*(“Болеет» = «болеет» | “Облетело» = «да») = 0,47; *p*(“Засохло” = «засохло» | “Облетело» = «да») = 0,49.

Процесс рассуждения (вывода) в байесовских сетях доверия

Следует отметить, что следствием байесовской теоремы является то, что она поддерживает оценку графа в обоих направлениях. Процесс рассуждения в ЭС сопровождается распространением по сети вновь поступивших свидетельств.

Введение в байесовские сети доверия новых данных приводит к возникновению переходного процесса распространения по байесовской сети доверия вновь поступившего свидетельства. После завершения переходного процесса каждому высказыванию, ассоциированному с вершинами графа, приписывается апостериорная вероятность, которая определяет степень доверия к этому высказыванию (believe – доверять (англ.)):

$$Bel(V_j^i) \stackrel{\Delta}{=} p\left(V_j^i | D\right),$$

где D – объединения всех поступивших в систему данных;

V_j^i – композиционные высказывания, составленные из элементарных, то есть множество значений X_i составляют V_j^i;

X_i – пропозиционные переменные (то есть переменные, значениями которых являются высказывания), определяющие состояние вершин БСД.

При этом процесс распространения вероятностей в БСД основывается на механизме пересчёта, в основе функционирования которого лежит следующая последовательность действий:

1. С каждой вершиной сети ассоциирован вычислительный процесс (процессор), который получает сообщения от соседних (связанных с ним дугами) процессоров.

2. Этот процессор осуществляет пересчёт апостериорных вероятностей $Bel(V_j^i)$ для всех возможных значений V_j^i данной переменной X_i, и посылает соседим вершинам ответные сообщения.

3. Деятельность процессора инициируется нарушением условий согласованности с состояниями соседних процессоров и продолжается до восстановления этих условий.

В некоторых системах, реализующих байесовские сети доверия, используется метод noisy or gate, позволяющий существенно упростить

вычислительный процесс. Суть его заключается в том, что в ряде примеров вершина «**y**» может быть условно независима от целого ряда вершин «$\mathbf{x_r}$», где r = 1,2,..., n. Для того чтобы сократить оценку 2^n вероятностей, которые необходимы при использовании таблиц условных вероятностей, и используется данный метод. Согласно ему вероятность «**y**» в зависимости от n вершин «$\mathbf{x_r}$» оценивается как:

$$p(y \mid x_1 x_2 \text{K}\ x_n) = 1 - \prod_{i=1}^{n} (1 - p(y \mid x_i)),$$

что позволяет оценить только $p(y \mid x_1)$, $p(y \mid x_2)$ ... $p(y \mid x_n)$, и на их основании определить оценку $p(y \mid x_1 x_2 \ldots x_n)$.

Байесовские сети доверия как одно из направлений современных экспертных систем

Выбор байесовских сетей доверия в качестве ЭС по сравнению с другими направлениями их построения обусловлен рядом причин.

1. Логический вывод в байесовских сетях доверия является *трактуемым с вычислительной точки зрения*, так как теория, лежащая в его основе, имеет аксиоматическое обоснование, отработанное в течение последних десятилетий. В то время как системы, основанные на теории нечётких множеств, на теории функций доверия, теории Демпстера - Шефера не имеет строгого математического обоснования и в большинстве случаев используют эвристические процедуры (ЭС типа MYCIN, EMYCIN и т.д.).

2. Показано, что психологически проще выполнять субъективное вероятностное оценивание причинно-следственных связей.

3. Метод noisy or gate обеспечивает эффективное вычисление условных вероятностей.

4. Несмотря на то, что теорию вероятности зачастую критикуют с точки зрения её использования в «знаниях», она не нарушает общих представлений о «замкнутом мире» объектов.

Одними из наиболее распространённых для современных ПЭВМ программных систем, реализующих теорию байесовских сетей доверия, являются: «MSBN» фирмы Microsoft, Hugin фирмы Hugin AIS, Дания.

Представление знаний с использованием байесовской сети доверия и условная независимость событий

Рассмотрим фрагмент представления медицинской БЗ, в которой можно выделить заболевания, симптомы их проявления, а также факторы риска, влияющие на возникновение заболеваний. Пусть некоторая упрощённая модель качественного описания БЗ имеет вид,

приведенный на рисунке 2.12. Эта модель соответствует следующему набору медицинских знаний:

Одышка [o] может быть вследствие туберкулёза [t], рака лёгких [r] или бронхита [b], а также вследствие ни одного из перечисленных заболеваний или более, чем одного.

Визит в Азию [a] повышает шансы туберкулёза [t].

Курение [k] – фактор риска как для рака [r], так и бронхита [b].

Результаты рентгена, определяя затемнённость в лёгких, не позволяют различить рак [r] и туберкулёз [t], так же как не определяет факт наличия или отсутствия одышки [o].

Последний факт представляется в графе промежуточной переменной (событием) [tr]. Эта переменная соответствует логической функции «или» для двух родителей ([t] и [r]) и она означает наличие либо одной, либо двух болезней или их отсутствие.

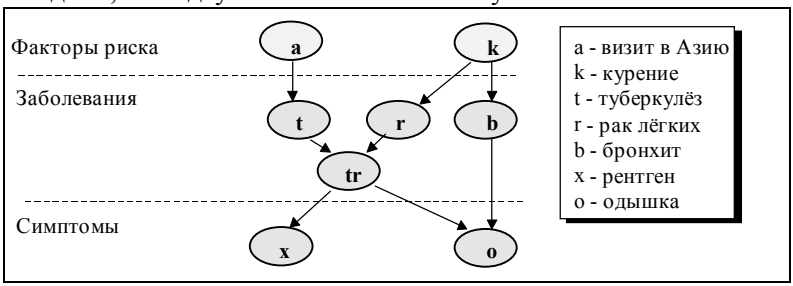


Рисунок 2.12. Представление фрагмента модели медицинской БЗ в виде БСД.

Важное понятие байесовской сети доверия – это условная независимость случайных переменных, соответствующих вершинам графа. Две переменные A и B являются *условно независимыми* при данной третьей вершине C, если при известном значении C, значение B не увеличивает информативность о значениях A, то есть

$$p\,(A \mid B,\, C) = p\,(A \mid C)\,.$$

Если имеется факт, что пациент курит, то мы устанавливаем наши доверия относительно рака и бронхита. Однако наши доверия относительно туберкулёза не изменяются. То есть [t] условно не зависит, от [k] при данном пустом множестве переменных:

$$p\,(t \mid k) = 0$$

Поступления положительного результата рентгена пациента повышают наши доверия относительно туберкулёза и рака, но не отно-

сительно бронхита. То есть [b] – условно не зависит от [x] при данном k

$$p\,(b \mid x,\, k\,) = p\,(b \mid k\,)$$

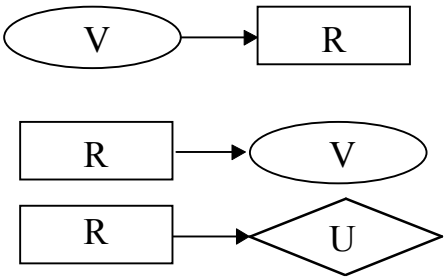
Однако, если бы знали также, что у пациента учащённое дыхание [o], то рентгеновские результаты также имели бы воздействие на наше доверие относительно бронхита. То есть [b] условно зависит от [x] при данных o и k. Таким образом, логический вывод в БСД означает вычисление условных вероятностей для одних переменных при наличии информации (свидетельств) о других. При этом для распространения вероятностей используется теорема Байеса.

2.10. Диаграммы влияния

Назначение и основные компоненты диаграмм влияния

Диаграммы влияния используются для принятия решений. Фактически диаграммы влияния – это байесовские сети доверия расширенные понятиями пользы (utility) и решения (decisions). Если байесовские сети доверия содержали только один тип вершин, которые мы назовём вершинами шансов, и которые соответствовали состоянию случайных переменных, то в диаграммах влияния используются ещё, как минимум, два типа вершин: вершины решения, обозначаемые в диаграммах влияния прямоугольниками и вершины пользы, обозначаемые в диаграммах влияния в виде ромба.

Вершины решения, а точнее сказать указания, содержащиеся в них, определяют временное старшинство:



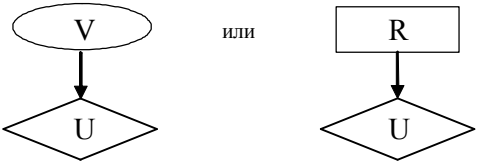
стрелка от случайной переменной (вершины шансов) к переменной решения (вершине решения) указывает, что значение случайной переменной известно на момент принятия решения;

стрелка от переменной решения к какой-либо другой переменной указывает время, упорядоченное решением.

При этом сеть должна оставаться ациклической и должен существовать непосредственный путь, содержащий все вершины решения в сети.

В процессе принятия решения важно не просто найти решение, а найти решение наилучшее в каком-то смысле. С этой целью в диаграммах влияния «вершины пользы» связываются с состоянием сети.

Каждая вершина пользы (полезности) содержит функцию полезности, которая связывает каждую конфигурацию состояния её родителей с полезностью. Вершины полезности не имеют наследников (а, следовательно, стрелка может быть направлена только к ним), т.е.



Принимая решение, мы исходим вероятности конфигурации сети. Поэтому можно вычислить ожидаемую полезность каждой альтернативы и выбрать альтернативу с наибольшей ожидаемой полезностью. Это *принцип максимальной ожидаемой полезности*. Диаграмма влияния может содержать несколько вершин полезности. При этом общая функция полезности представляет собой сумму всех локальных функций полезности:

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{k} f_i(x_1, \ldots, x_n)$$

Процесс принятия решения с использованием диаграмм влияния будет осуществляться в следующем порядке:

после наблюдения значений переменных, которые являются родителями первой вершины решения, мы хотим знать максимальную полезность для альтернатив;

ЭС вычислит эти полезности в предположении, что все будущие решения будут сделаны оптимально, используя все имеющиеся свидетельства в момент каждого решения.

Пример построения простейшей диаграммы влияния

Вернёмся к ранее рассмотренному примеру с плантацией «яблочного Джека». Оценив ранее состояние своего дерева, Джек ставит перед собой цель принятия решения об инвестировании материальных средств в лечение дерева.

Для решения этой задачи добавим к исходной байесовской сети доверия ещё три вершины шансов, полностью аналогичных тем, что

уже были в байесовской сети доверия. Новые вершины: “Болеет1”, “Засохло1» и “Облетело1” являются точно такими же как и их аналоги в предыдущей модели, но только отображают будущий момент времени (момент сбора урожая).

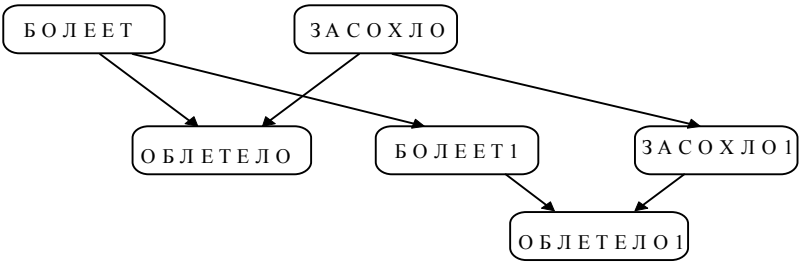


Рисунок 2.13. Расширенная модель БСД на момент уборки урожая.

Новые вершины имеют те же состояния, что и старые. В новой модели ожидаются зависимости от “Болеет» к “Болеет1” и от “Засохло” к “Засохло1”. Это связано с тем, что если дерево болеет сейчас, очень вероятно, что оно будет болеть и в будущем.

Конечно, сила зависимости зависит от того, как далеко в будущее мы хотим заглянуть. Можно было бы установить зависимость и от “Облетело» к “Облетело1”, но в данной модели, для ее упрощения, мы этого делать не будем. «Яблочный Джек «имеет возможность сделать что-либо для решения проблемы опадания листвы со своих плодовых деревьев и, тем самым, сохранить урожай:

Он может попытаться лечить дерево, проводя его опрыскивание, чтобы избавить от болезни.

С другой стороны, если он считает, что опадание листвы вызвано засухой, он может сохранить свои деньги и просто ждать дождя.

Действия, связанные с лечением дерева, могут быть добавлены в нашу модель в виде вершины решения, и при этом мы от байесовской сети доверия переходим к диаграмме влияния, которая будет иметь вид:

При этом случайная переменная решения, соответствующая вершине “Лечение” может иметь два состояния (“Лечение»=«да», “Лечение”=«нет»). Как видно из рисунка 2.14, диаграмма влияния смоделирована со стрелкой от “Лечение” к “Болеет1”. Это вызвано тем, что лечение повлияет на будущее здоровье дерева.

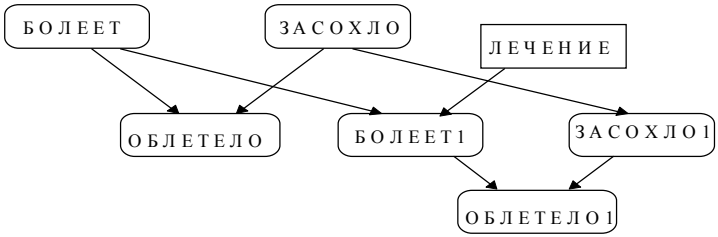


Рисунок 2.14. Преобразование БСД в диаграмму влияния добавлением в нее вершины решения.

Перед тем как завершить диаграмму влияния, необходимо определить функцию полезности, позволяющую вычислить пользу от принятия решения. Это делается добавлением к диаграмме влияния вершин полезности, каждая из которых определяет вклад в общую выгодность. При этом изменённая диаграмма влияния примет вид, приведённый на рисунок 2.15.

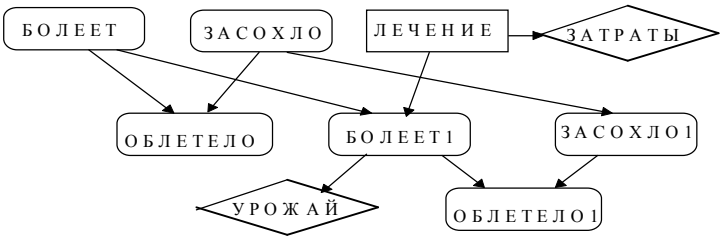


Рисунок 2.15. Диаграмма влияния с добавленными вершинами полезности.

Вершина “Затраты” содержит информацию о затратах на лечение деревьев, а вершина “Урожай» представляет собой доходы, полученные от сбора урожая. При этом естественно, что количество и качество урожая зависит от состояния деревьев. Поэтому вершина “Урожай» зависит от состояния вершины «Болеет1», указывая, что продукция зависит от здоровья дерева в момент сбора урожая.

Модель, представленная на рисунке 2.15, даёт законченное качественное представление диаграммы влияния. Для получения количественного представления необходимо построить таблицу условных вероятностей для каждой из вершин шансов и задать таблицы до-

ходности для каждой из вершин полезности. Вершины принятия решения не имеют таблиц условных вероятностей.

При этом таблицы условных вероятностей для *p*(“Болеет”), *p*(“Засохло”) и *p*(“Облетело» | “Болеет”, “Засохло”),будут иметь тот же вид, что и в примере предыдущей главы. Таблица же условных вероятностей для *p*(“Облетело1” | “Болеет1”, “Засохло1”), будет аналогична *p*(“Облетело» | “Болеет”, “Засохло”).

Таблицы условных вероятностей для всех остальных состояний должны быть получены из анализа предметной области и выявления знаний от экспертов. Для рассматриваемого примера они могут иметь вид аналогичный, приведенным в таблицах.

Таблица условных вероятностей *p*(“Болеет1” | “Болеет”, “Лечение”)

	“Лечение» = «да»		“Лечение» = «нет»	
	“Болеет” = «болеет»	“Болеет” = «нет»	“Болеет” = «болеет»	“Болеет” = «нет»
“Болеет1” = «да»	0,20	0,01	0,99	0,02
“Болеет1” = «нет»	0,80	0,99	0,01	0,98

Таблица условных вероятностей *p*(“Засохло1» | “Засохло”)

	“Засохло» = «засохло»	“Засохло» = «нет»
“Засохло1» = «да»	0,60	0,05
“Засохло1» = «нет»	0,40	0,95

Следующие таблицы показывают, как могут быть определены для рассматриваемого случая таблицы выгодности. В них функции полезности выражаются в виде стоимостных показателей и задаются в одних и тех же условных единицах.

Таблицы выгодности для вершин полезности

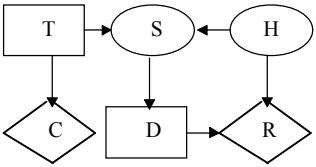
U(“Урожай”)			U(“Затраты”)	
“Болеет1» = «да»	“Болеет1» = «нет»		“Лечение” = «да»	“Лечение» = «нет»
3000	20000			

Цель диаграммы влияния – вычислить действие, связанное с вершиной “Лечение” для того чтобы получить наибольшую ожидаемую выгодность. Ответом для принятия решения об инвестировании лечения будет вычисление общей функции полезности при условии, что *p*(“Облетело”=«облетело»=1.

Диаграммы влияния с несколькими вершинами решения

Сложность построения и исследования диаграмм влияния в большей степени определяется не количеством вершин шансов, а сложностью их взаимосвязей как между собой, так и, особенно, взаимосвязями с вершинами решения и полезности.

Рассмотрим пример с небольшим числом переменных (вершин), но довольно-таки сложным взаимодействием между ними. Диаграмма влияния, моделирующая процесс принятия решения о бурении нефтяной скважины, будет иметь вид:



H – состояние нефтяного пласта (сухое, влажное, мокрое);
T – решение о проведении сейсморазведки;
C – затраты на сейсмотест;
D – решение бурить или не бурить скважину;
S – выход сейсмического теста;
R – доходы от бурения;

Рисунок 2.15. Диаграмма влияния для принятия решения о бурении нефтяной скважины.

Нефтяники должны принять решение о бурении скважины. Предварительная экспертиза геологов выявила следующее распределение вероятности состояния нефтяного пласта:

P(H = «сухое») = 0,5 ; P(H = «влажное») = 0,3 ; P(H = «мокрое») = 0,2.

Однако решение о бурении может быть более точным, если предварительно провести дополнительную сейсморазведку, за которую надо затратить $10000. Её результатом будет геологическая структура участка: закрытая (хорошие запасы нефти), открытая (средние запасы), отсутствие (малые запасы нефти).

Разведанная структура, наряду с состоянием нефтяного пласта, определяет условные вероятности для результатов сейсмического теста по решению о бурении скважины:

Таблица условных вероятностей p(S | H, T)

	T = «тест_проведен»			T = «нет»
	H=« сухой»	H=« влажный»	H=« мокрый»	При всех H
S=« закрытая»	0,1	0,3	0,5	0,33
S=« открытая»	0,3	0,4	0,4	0,33
S=« нет»	0,6	0,3	0,1	0,33

Стоимость бурения $7000. Если принимается решение о бурении ожидаемый доход (то есть стоимость найденной нефти минус цена бурения) будет:

Таблица выгодности для вершин полезности R

	D = «бурить»			D = «не бурить»
	H=« сухой»	H=« влажный»	H=« мокрый»	При всех H
U(R)= =f(H,D)	-70000	50000	200000	0

На основе приведённых данных и диаграммы влияния ЭС вычислит полезность, связанную с сейсморазведкой – $ 22500 и полезность её не проведения – $20000. Таким образом, оптимальной стратегией является проведение разведки, а затем решение бурить или нет на основе полученных сейсмотестов.

2.11. Сети доверия с условными гауссовскими переменными

Непрерывные случайные величины

До сих пор мы предполагали, что каждое из событий Z характеризуется конечным множеством состояний (z_1, z_2,... z_n) и вероятностями пребывания в каждом из них:

$$Pz_1, Pz_2,.. ., Pz_n ; \sum_{i=1}^{n} Pz_i = 1$$

Однако во многих случаях события могут принимать любые состояния из некоторого диапазона. Так, например, доходность какого-либо мероприятия может характеризоваться любым числовым значением ожидаемой прибыли.

В этом случае Z будет являться непрерывной случайной величиной, пространством возможных состояний которой будет весь диапазон допустимых её значений:

$$Z = \{z \mid a \le z \le b\},$$

содержащий бесконечное множество точек. При этом уже нельзя говорить о вероятности отдельного состояния, так как при бесконечно большом их числе вес каждого будет равен нулю. Поэтому распределение вероятностей для непрерывной случайной величины определяется иначе, чем в дискретном случае и для их характеристики используются: функции распределения вероятностей; плотности распределения вероятностей.

Функция распределения вероятностей $F(x)$ определяет вероятность того, что значения случайной величины z не превзойдут некоторого x, то есть

$$F(x) = P\,(-\infty < z \le x\,)$$

Эта функция обладает такими свойствами, как: $F(x)$ – неубывающая функция, $F(-\infty) = 0$, $F(\infty) = 1$. Общий вид функции, удовлетворяющий отмеченным свойствам, графически можно представить в виде, аналогичном приведенному на рис.8.1. Зная функцию распределения вероятностей, можно вычислить вероятность того, что значение случайной величины z окажется внутри малого интервала от x до $x + \Delta x$

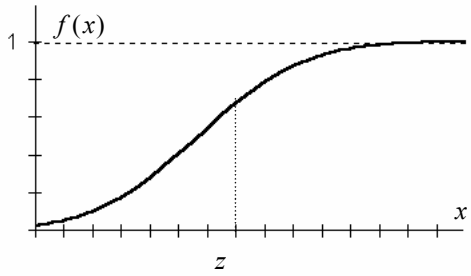


Рисунок 2.16. Функция распределения вероятностей

$$P(x \le z < x + \Delta x) = F(x + \Delta x) - F(x) == \frac{F(x + \Delta x) - F(x)}{\Delta x} \cdot \Delta x\,.$$

Первый сомножитель в правой части последнего выражения есть значение вероятности, приходящаяся на единицу длины участка Δx. Предел этого отношения при $\Delta x \to 0$ представляет собой производную функции распределения:

$$f(x) = \lim_{x \to 0} \frac{F(x + \Delta x) - F(x)}{\Delta x} = \frac{dF(x)}{dx}$$

и называется плотностью распределения вероятностей. Отметим основные свойства функции $f(x)$:

а). $\int_a^b f(x) \cdot dx = \int_{F(a)}^{F(b)} dF(x) = F(b) - F(a) = P(a \le z < b)$,

т.е. интеграл плотности распределения даёт вероятность того, что случайная величина z принимает значения, лежащие в интервале от a до b;

б). $\int_{-\infty}^{\infty} f(x) \cdot dx = 1$,

откуда следует, что площадь, ограниченная кривой $f(x)$ и осью абсцисс, всегда равна единице.

Непрерывные гауссовские переменные

Непрерывной гауссовской переменной будем называть случайную величину, подчиняющуюся нормальному (или гауссовскому) закону распределения, который характеризуется функцией распределения плотности вероятности вида:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{\frac{-(x-m)^2}{2\sigma^2}}, \text{ для } -\infty < x < \infty.$$

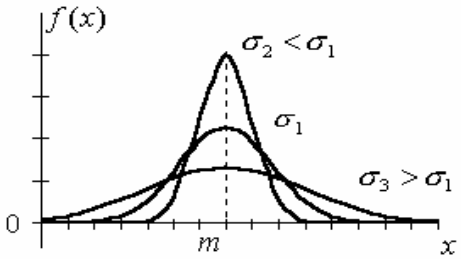


Рисунок 2.17. Графики плотности распределения вероятностей гауссовских переменных

Нормальное распределение: определяется параметрами m и σ, называемыми математическим ожиданием и среднеквадратическим отклонением; обычно обозначается $N(m,\sigma)$ и имеет график распределения вероятности вида, аналогичному приведенному на рисунке 2.17. В случае нормального распределения выражение вида:

$$P(a < x < b) = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{\frac{-(x-m)^2}{2\sigma^2}} \cdot dx$$

позволяет определить вероятность того, что случайная величина попадает на заданный интервал вещественной оси (a, b). Нормальный (или гауссовский) закон распределения является одним из наиболее важных и широко распространенных законов распределения случайных величин, так как он наиболее часто встречается на практике; является предельным законом, к которому приближается ряд других законов распределения при весьма часто встречающихся типовых условиях.

Числовые характеристики случайных величин

Функция распределения и плотность распределения вероятностей являются наиболее полными характеристиками случайных величин. Однако во многих задачах практики оказывается трудно или даже невозможно полностью описать функцию распределения вероятностей.

В то же время для решения многих задач достаточно знать лишь некоторые параметры, характеризующие случайную величину с той или иной точки зрения. Наиболее распространёнными числовыми характеристиками (или моментами) случайных величин являются математическое ожидание и дисперсия, которые определяются следующими математическими выражениями:

Для дискретных случайных переменных, когда пространство исходов эксперимента Z является конечным множеством	Для непрерывных случайных переменных, когда пространство исходов эксперимента Z содержит бесконечное множество точек
$M(z) = m_z = \bar{z} = \sum_{z \in Z} z \cdot p(z)$	$M(z) = m_z = \bar{z} = \int_a^b z \cdot f(x) \cdot dz$
$D(z) = \mu_2(z) = \sum_{z \in Z} (z - m_z)^2 \cdot p(z)$	$D(z) = \mu_2(z) = \int_{-\infty}^{\infty} (z - m_z)^2 \cdot f(x) \cdot dx$

Математическое ожидание является величиной, вокруг которой группируются значения случайной переменной. Дисперсия характеризует отклонение значений случайной величины от математического ожидания, то есть является характеристикой рассеивания случайной величины. Чем меньше дисперсия, тем более тесно группируются отдельные значения случайной величины вблизи математического ожидания.

Однако в ряде случаев дисперсия оказывается неудобной для практического использования, так как имеет размерность квадрата случайной величины. Поэтому в качестве характеристики рассеивания случайной величины часто используют корень квадратный из дисперсии, получивший название среднеквадратического отклонения:

$$\sigma_я = \sqrt{D(z)}$$

Отметим основные свойства дисперсии: дисперсия неслучайной величины равна нулю $D(c) = M[(c-c)^2] = 0$; неслучайную величину можно вынести за знак дисперсии, возведя её в квадрат, $D(cz)=M[(cz - cm_z)^2] = c^2 \cdot D(z)$.

Совместное использование дискретных и непрерывных переменных в байесовских сетях доверия

В настоящее время существует ряд программных реализаций оболочек ЭС на основе БСД, которые позволяют оперировать не только дискретными, но и непрерывными случайными переменными. К числу таких программных средств относится и Hugin. Однако при использовании БСД, содержащих как непрерывные, так и дискретные переменные существует ряд ограничений:

дискретные переменные не могут иметь непрерывных родителей;

непрерывные переменные должны иметь нормальный закон распределения, условный на значениях родителей;

распределение непрерывной переменной Y с дискретными родителями I и непрерывными родителями Z является нормальным распределением

$$P(Y \mid I = i, Z = z) = N(m_y(m_i, m_z), \sigma_y(\sigma_i)),$$

где m_y линейно зависит от непрерывных родителей, а σ_y вообще не зависит от непрерывных родителей. Однако, оба они (m_y и σ_y) зависят от дискретных родителей. Это ограничение гарантирует возможность точного вывода.

Рассмотрим пример построения БСД с непрерывными и дискретными вершинами шансов. Пусть требуется построить ЭС, позволяющую оценивать суммарные производственные затраты в зависимости от использования и загрузки трёх групп оборудования (например, трех пилорам). Такая ЭС поможет в выборе наиболее рациональной загрузки оборудования, в обоснованном решение об аренде необходимого оборудования и допустимых арендных платежах и многое другое, необходимое менеджеру или инженеру по деревообработке.

При анализе предметной ситуации эксперты установили, что в состав суммарных производственных затрат (без учёта зарплаты и начислений) входят:

прямые производственные затраты на каждую группу оборудования за исследуемый календарный период, которые зависят как от количества используемых групп оборудования, так и от времени работы каждой из групп в течение исследуемого периода времени, т.е. от степени загрузки каждой из групп;

расходы на амортизацию каждой из групп оборудования, зависящие как от её балансовой стоимости, так и установленных норм амортизации;

арендную плату за участок при каждой из групп оборудования, используемый для складирования сырья и продукции, которая зависит как от площади участка, так и от ставок арендной платы.

Построение любой модели БСД начинается с выделения основных объектов и событий предметной области, анализа возможных состояний каждого из событий и установления причинно-следственных связей между ними. Так, в нашем примере, исходя из мнения экспертов, можно заключить, что на суммарные производственные затраты оказывают влияние:

степень загрузки каждой из групп оборудования в течение исследуемого периода;

значения нормы амортизационных отчислений;

ставка арендной платы за участок, используемый определенной группой оборудованием.

При этом модель БСД будет иметь вид, приведенный на рисунке 2.18, где одинарные овалы соответствуют дискретным событиям, а двойные овалы – непрерывным событиям (гауссовским переменным). Для того чтобы данная качественная модель превратилась в полную БСД, необходимо определить ее количественные характеристики. Для этого необходимо тщательно проанализировать каждое из событий.

Так, вершина «Загрузка оборудования» соответствует дискретному событию, которое характеризуется тремя возможными состояниями. Вероятность пребывания в каждом из них определяется степенью загрузки каждой из групп оборудования, при условии, что суммарная загрузка всего оборудования равна единице. Если считать, что все группы оборудования загружены равномерно, то распределение вероятностей для этой вершины будет иметь вид, представленный в таблице.

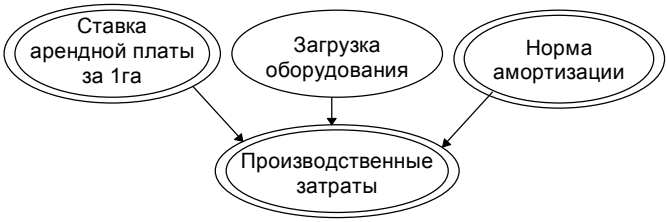


Рисунок 2.18. Модель БСД с непрерывными и дискретными событиями

Распределение вероятностей для вершины «Загрузка оборудования».

P(«Загрузка оборуд.» = ’Пилорама 1’	*P*(«Загрузка оборуд.» = ’Пилорама 2’	*P*(«Загрузка оборуд.» = ’Пилорама 3’
0,333	0,333	0,333

При этом следует отметить, что возможны и любые другие исходные распределения вероятностей, учитывающие различные варианты загрузки оборудования. Полученные от экспертов знания о том, что:

ставка аренды 1 га земли в среднем составляет 2500 у.е. и колеблется в пределах ±10%, т.е. принимает значения 2500 ± 250 у.е.;

а норма амортизации может находиться в пределах 5 - 10 % от балансовой стоимости, т.е. принимать значения 7,5 ± 2,5% (или 0,075 ± 0,025 относительных единиц);

позволяют определить параметры ещё двух вершин – «Ставка арендной платы» и «Норма амортизации». Предполагая, что эти вершины шансов являются непрерывными случайные переменными с гауссовским законом распределения, необходимо задать параметры этих законов для каждой из вершин.

Параметры законов распределение для непрерывных вершин

	« Ставка арендной платы»	« Норма амортизации»
математическое ожидание (m)	2500	0,075
дисперсия ($D = \sigma^2$)	62500 (=250^2)	0,000625 (=$0{,}025^2$)

Что касается вершины «Производственные затраты», то она характеризуется случайной переменной, условно нормальной на значениях родителей (т.е. на значениях трёх других вершин нашего примера). Следует отметить, что в общем случае распределение ве-

роятностей для вершин, аналогичных «Производственные затраты» является не просто нормальным, а смешанным нормальным распределением, т.е. представляет собой весовую сумму распределений, для каждого из которых должен быть задан список его параметров:

математические ожидания и дисперсии для распределений, описывающих степень влияния дискретных родителей;

весовые коэффициенты, учитывающие степень влияния на математическое ожидание непрерывных родителей.

Если теперь, используя экспертное оценивание, предположить, что:

балансовая стоимость каждой из пилорам составляет 50000, 40000 и 30000 у.е.;

площадь арендуемых участков, закрепляемая за ними равна 0,6; 0,5 и 0,4 га;

а оценка прямых затрат на поддержание нормальной работы каждой из пилорам в среднем составляет 3000, 3200 и 3500 у.е. и получена с 5% точностью;

то степень влияния родительских вершин на «Производственные затраты» можно представить таблицей следующего вида.

Параметры, определяющие распределение вероятностей для вершины «Производственные затраты»

« Загрузка оборудования»	Пилорама 1	Пилорама 2	Пилорама 3
математическое ожидание (m)	3000	3200	3500
дисперсия ($D = \sigma^2$)	22500 ($=150^2$)	25600 ($=160^2$)	30625 ($=175^2$)
« Норма амортизации»	50000	40000	30000
« Ставка аренды»	0,6	0,5	0,4

На основе проведенного экспертного оценивания предметной области можно теперь переходить к реализации БСД в системе Hugin и построения на ее базе ЭС, которая позволит получить оценку любых ее состояний, при любых произвольно задаваемых свидетельствах.

Логический вывод в байесовских сетях доверия с непрерывными и дискретными состояниями

Логический вывод в таких БСД заключается в распространении вероятностей и параметров гауссовских законов распределения по всей сети в зависимости от полученных свидетельств. В частности, для рассматриваемого примера для исходного набора данных будут получены оценки производственных затрат:

$m_{\text{п.з}} = 7483{,}33$,

$\sigma_{\text{п.з}} = 1208{,}1$.

Однако эти оценки могут быть пересчитаны для случая иной загрузки оборудования или получения новых свидетельств о ставках аренды или нормах амортизации. Наряду с этим возможен и обратный вывод в этой простейшей экспертной системе. Он, например, может заключаться в определении допустимых ставок арендных плат при возможном значении суммарных производственных затрат.

В основе процесса логического вывода лежат довольно сложные математические алгоритмы, которые мы рассмотрим на простейшей двухуровневой сети для случая прямого распространения распределения вероятностей.

Пусть независимые дискретные случайные величины $X_1,..., X_s$ и непрерывные случайные величины $Z_1,..., Z_r$ оказывают влияние на результирующую случайную величину Y.

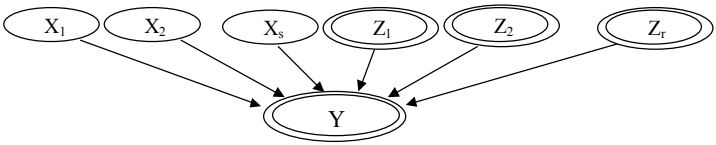


Рисунок 2.19. Двухуровневая БСД с непрерывными и дискретными переменными.

Каждая из дискретных случайных величин $X_j (j = \overline{1,s})$ имеет своими исходами значения $X_{ij} (i = \overline{1,n_j})$ с вероятностями P_{ij}, для которых $\sum_{i=1}^{n_j} P_{ij} = 1$. Совместное влияние дискретных случайных величин на Y характеризуется математическим ожиданием $(m_{i1},...,m_{is})$ и дисперсиями $(D_{i1},...,D_{is})$. Каждая из непрерывных случайных величин Z_λ имеет непрерывное нормальное распределение с параметрами (m_λ, D_λ), где $\lambda = \overline{1,r}$. Совместное влияние непрерывной случайной величины Z_λ и исходов дискретных величин

на результирующую случайную величину *Y* характеризуется весовыми коэффициентами $k_{\lambda,i1,\ldots,is}$ для $\lambda = \overline{1,r}$.

Тогда характеристики результирующей величины *Y* могут быть вычислены по следующим выражениям:

$$m = \sum_{i_1=1}^{n_1} \ldots \sum_{i_s=1}^{n_s} p_{1,i_1} \ldots p_{s,i_s} (m_{i_1,\ldots,i_s} + \sum_{\lambda=1}^{r} K_{\lambda,i_1,\ldots,i_s} \cdot m_\lambda)$$

$$D = \sum_{i_1=1}^{n_1} \ldots \sum_{i_s=1}^{n_s} p_{1,i_1} \ldots p_{s,i_s} ((m_{i_1,\ldots,i_s} + \sum_{\lambda=1}^{r} K_{\lambda,i_1,\ldots,i_s})^2 + D_{i_1,\ldots,i_s} + \sum_{\lambda=1}^{r} K^2_{\lambda,i_1,\ldots,i_s} \cdot D_\lambda) - m^2$$

В частности, для рассмотренного выше примера, содержащего две исходные непрерывные (r= 2) переменные и одну дискретную (s = 1) переменную, имеющую три исхода (n_1=3), числовые характеристики случайной переменной «Производственные затраты» будут

$$m = \sum_{i=1}^{3} p_i (m_i + \sum_{\lambda=1}^{2} k_{\lambda,i} \cdot m_\lambda) =$$

$$0{,}333 \cdot (3000 + 50000 \cdot 0{,}075 + 0{,}6 \cdot 2500) +$$
$$+ 0{,}333 \cdot (3200 + 40000 \cdot 0{,}075 + 0{,}5 \cdot 2500) +$$
$$+ 0{,}333 \cdot (3500 + 3000 \cdot 0{,}075 + 0{,}4 \cdot 2500) =$$
$$= 0{,}333 \cdot (8250 + 7450 + 6750) = 7483{,}33,$$

$$D = \sum_{i=1}^{3} p_i ((m_i + \sum_{\lambda=1}^{2} k_{\lambda,i} \cdot m_\lambda)^2 + D_i + \sum_{\lambda=1}^{r} k^2_{\lambda,i} \cdot D_\lambda) - m^2 = 1459505{,}6$$

$$\sigma = \sqrt{D} = \sqrt{1459505{,}6} = 1208{,}1.$$

Байесовы вероятностные методы обучения машин являются существенным шагом вперед, в сравнении с популярными моделями «черных ящиков». Они дают понятное объяснение своих выводов, допускают логическую интерпретацию и модификацию структуры отношений между переменными задачи, а также позволяют в явной форме учесть априорный опыт экспертов в соответствующей предметной области.

Благодаря удачному представлению в виде графов, байесовы сети весьма удобны в пользовательских приложениях.

Байесовы сети базируются на фундаментальных положениях и результатах теории вероятностей, разрабатываемых в течение нескольких сотен лет, что и лежит в основе их успеха в практической плоскости.

Байесова методология, в действительности, шире, чем семейство способов оперирования с условными вероятностями в ориентиро-

ванных графах. Она включает в себя также модели с симметричными связями (случайные поля и решетки), модели динамических процессов (марковские цепи), а также широкий класс моделей со скрытыми переменными, позволяющих решать вероятностные задачи классификации, распознавания образов и прогнозирования.

В ближайшем будущем предполагается значительно расширить применение Байесовых сетей доверия. Например, на одном из сайтов поисковиков конструируются байесовские сети для моделирования успешных запросов, поступающих от пользователей. Эти сети могут пополнять регистрационный файл поискового сервера назначаемыми категориями предполагаемых целей информации для обеспечения возможности предсказания модификаций запросов.

3. Прикладные программы, реализующие БСД

3.1. Производители программ

Байесовы сети — интенсивно развивающаяся научная область, многие результаты которой уже успели найти коммерческое применение. Приводится список популярных программ, спектр которых, отражает общую картину, возникшую в последние 20 лет.

AUAI — Ассоциация анализа неопределенности в искусственном интеллекте (URL: http://www.auai.org/)

Ассоциация анализа неопределенности в искусственном интеллекте (Association for Uncertainty in Artificial Intelligence — AUAI) — некоммерческая организация, главной целью которой является проведение ежегодной Конференции по неопределенности в искусственном интеллекте (UAI).

Конференция UAI-2002 прошла в начале августа 2002 года в Университете Альберты (Эдмонтон, Канада). Конференции UAI проходят ежегодно, начиная с 1985 года, обычно совместно с другими конференциями по смежным проблемам. Труды конференций издаются в виде книг, однако многие статьи доступны в сети.

NETICA (URL: http://www.norsys.com/index.html)

Norsys Software Corp. — частная компания, Расположенная в Ванкувере (Канада). Norsys специализируется в разработке программного обеспечения для байесовых сетей. Программа Netica – основное достижение компании разрабатывается с 1992 года, и стало коммерчески доступным в 1995 году. В настоящее время Netica является одним из наиболее широко используемых инструментов для разработки байесовых сетей.

Версия программы с ограниченной функциональностью свободно доступна на сайте фирмы Norsys.

Netica – мощная, удобная в работе программа для работы с графовыми вероятностными моделями. Она имеет интуитивный и приятный интерфейс пользователя для ввода топологии сети. Соотношения между переменными могут быть заданы как индивидуальные вероятности, в форме уравнений, или путем автоматического обучения из файлов данных (которые могут содержать пропуски).

Созданные сети могут быть использованы независимо и как фрагменты более крупных моделей, формируя тем самым библиотеку модулей. При создании сетевых моделей доступен широкий спектр функций и инструментов.

Многие операции могут быть сделаны несколькими щелчками мыши, что делает систему Netica весьма удобной для поисковых исследований, и для обучения и для простого просмотра, и для обучения модели байесовой сети. Система Netica постоянно развивается и совершенствуется.

Knowledge Industries (URL: http://www.kic.com/)

Knowledge Industries – ведущий поставщик программных инструментальных средств для разработки и внедрения комплексных диагностических систем. При проектировании сложных и дорогостоящих вариантов систем диагностик в компании используется байесовы сети собственной разработки.

Data Digest Corporation (URL: http://www.data-digest.com/home.html)

Data Digest Corporation является одним из лидеров в применении методов байесовых сетей к анализу данных.

BayesWare, Ltd (URL: http://www.bayesware.com/corporate/profile.html)

Компания BayesWare основана в 1999 году. Она производит и поддерживает программное обеспечение, поставляет изготовленные на заказ решения, предоставляет программы обучения, и предлагает услуги консультирования корпоративным заказчикам и общественным учреждениям. Одна из успешных разработок компании, Bayesware Discoverer, основана на моделях байесовых сетей.

HUGIN Expert (URL: http://www.hugin.com/)

Компания Hugin Expert была основана в 1989 году в Ольборге (Дания). Их основной продукт Hugin начал создаваться во время работ по проекту ESPRIT, в котором системы, основанные на знаниях, использовались для проблемы диагностирования нервно-мышечных заболеваний. Затем началась коммерциализация результатов проекта и основного инструмента — программы Hugin. К настоящему моменту Hugin адаптирована во многих исследовательских центрах компании в 25 различных странах, она используется в ряде различных областей, связанных с анализом решений, поддержкой принятия решений, предсказанием, диагностикой, управлением рисками и оценками безопасности технологий.

Hugin является программой реализацией системы принятия решений на основе байесовских сетей доверия. Имеет две версии Pro и Explorer. Функционирует в среде OS Windows'95, Windows NT, а также имеет версию UNIX. Эта система имеет развитый интерфейс и позволяет достаточно просто создавать базы знаний и фактов. Использует два основных режима работы:

режим редактирования и построения причинно-следственной сети, а также заполнения таблиц условных вероятностей, являющихся количественным описанием БЗ.

режим расчёта вероятностных оценок для принятия решения по всем событиям, входящим в причинно-следственную сеть. Расчёты могут осуществляться как на основе классической теории Байеса, так и на основе методов теории возможностей.

Hugin имеет возможность связи с основными наиболее распространёнными программными средствами фирмы Microsoft. Данная ЭС имеет все основные функции любой информационной системы, включая такие, как хранение данных, вывод на принтер всех элементов ЭС, диагностика ошибок в работе.

3.2. Основы построения простейшей байесовской сети доверия

Рассмотрим основы построения экспертных систем на основе байесовских сетей доверия (БСД), методы их реализации и приемы работы с ними в системе HUGIN на примере построения простейшей экспертной системы, описанной в параграфе 2.9.

ЕСЛИ	« *Дерево засыхает при недостатке влаги*»
ИЛИ	« *Дерево болеет*»
ТО	« *Листва опадает*».

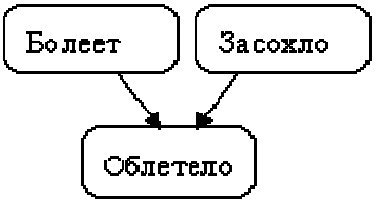


Рисунок 3.1. Пример БСД

Байесовская сеть доверия, содержит три вершины событий: “Болеет”, “Засохло” и “Облетело”. При этом каждое событие может принимать всего лишь одно из двух возможных состояний:

- «Болеет» – состояния ‘болеет’ или ‘нет’;
- «Засохло» – состояния ‘засохло’ или ‘нет’;
- «Облетело» – состояния облетело’ или ‘нет’.

Байесовская сеть доверия, приведённая на рисунке 3.1, моделирует тот факт, что имеется причинно-следственная зависимость от вершины “Болеет» к вершине “Облетело» и от вершины “Засохло” к вершине “Облетело”. Это отображено стрелками на байесовской сети доверия.

Когда есть причинно-следственная зависимость от вершины A к другой вершине B, то мы ожидаем, что когда A находится в некотором определённом состоянии, это оказывает влияние на состояние B. Следует быть внимательным, когда моделируется зависимость в БСД. Иногда совсем не очевидно, какое направление должна иметь стрелка. Например, в рассматриваемом примере, мы говорим, что имеется зависимость от “Болеет» к “Облетело”, так как когда дерево болеет, это может вызывать опадание его листвы. Опадание листвы является следствием болезни, а не болезнь – следствием опадания листвы.

На рисунке 3.1 дано графическое представление БСД. Однако это только качественное представление. Перед тем, как назвать это полностью байесовской сетью доверия, необходимо определить количественное представление, то есть множество таблиц априорных и условных вероятностей.

Таблица 3.1.

P(Болеет)	
Болеет=’болеет’	Болеет=’нет’
0.1	0.9

Таблица 3.2.

P(Засохло)	
Засохло=’засохло’	Засохло=’нет’
0.1	0.9

Таблица 3.3.

P(Облетело | Болеет, Засохло)				
	Засохло=’засохло’		Засохло=’нет’	
	Болет=’болеет’	Болет=’нет’	Болеет=’болеет’	Болеет=’нет’
Облетело=’да’	0.95	0.85	0.90	0.02
Облетло=’нет’	0.05	0.15	0.10	0.98

Заметим, что все три таблицы показывают вероятность пребывания некоторой вершины в определённом состоянии, обусловленным состоянием её родительских вершин. Но так как вершины “Болеет» и “Засохло” не имеют родительских вершин, то их вероятности являются маргинальными, т.е. не зависят (не обусловлены) ни от чего. На данном примере мы рассмотрели, что и как описывается очень простой байесовской сетью доверия.

Современные программные средства (такие как MSBN, HUGIN и др.) обеспечивают инструментарий для построения таких сетей, а также возможность использования БСД для введения новых свидетельств и получения решения (вывода) за счёт пересчёта вероятностей во всех вершинах, соответствующих этим свидетельствам.

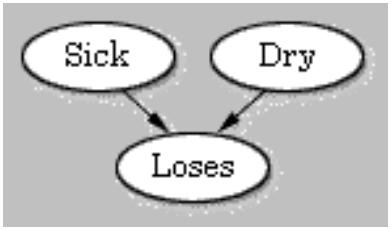


Рисунок 3.2

При реализации, описанной нами, БСД в среде HUGIN следует иметь в виду, что данная система не во всех режимах поддерживает работу с русскими шрифтами. Поэтому, в дальнейшем, будем пользоваться обозначениями вершин на английском языке. При этом исходная БСД примет вид, приведенный на рисунке 3.2.

Основные приемы работы с HUGIN при построении новой БСД

После запуска HUGIN Runtime откроется окно системы HUGIN, которое содержит панель меню, панель инструментов *(tool bar pane*), а также панель редактирования вершин (*node edit pane*) и панель графического отображения сети (network pane). Новая пустая сеть доверия получает имя «unnamed1» и автоматически открывается в окне сети (network window).

После запуска системы HUGIN (рисунок 3.3) она автоматически устанавливается в режим редактирования (edit mode), что позволяет немедленно приступить к построению новой БСД и определению состояния ее вершин.

Другим важным режимом работы системы является режим выполнения (run mode), который позволяет использовать БСД для получения требуемых результатов.

Добавление новых вершин в проектируемую БСД

Построение БСД начинается с определения отдельных вершин, входящих в проектируемую БСД. При этом для создания вершин с

дискретными состояниями, определения их свойств и задания причинно-следственных связей между вершинами используются, приведенные на рис.4, инструменты интерфейса системы, которые активируются путем нажатия левой кнопки мышки на их пиктограммах.

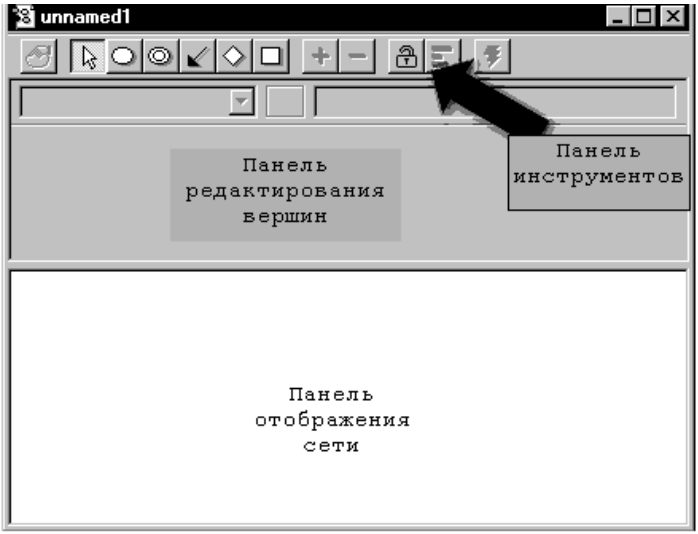


Рисунок 3.3

Для построения рассматриваемого примера БСД, первое, что необходимо сделать – это создать вершину ***Sick*** (“Болеет”). Для этого необходимо:

установить режим добавления вершин с дискретными состояниями, выбрав соответствующую пиктограмму в панели инструментов (рисунок 3.4);

	- вершины с дескретными состояниями *[discrete chance tool]*
	- свойства вершин *[node properties tool]*
	- добавление связей *[casual arrow tool]*

Рисунок 3.4

щелкнуть мышкой в любом месте панели отображения сети (рисунок 3.3), где предполагается размещения добавляемой вершины.

После щелчка в панели отображения сети должна отобразиться в виде овала вершина, название которой по умолчанию будет **C1**. В соответствии с нашей моделью, необходимо этой вершине присвоить название ***Sick***. Для этого:

выделите курсором мыши нужную вершину;

установите режим определения свойств вершины [*Node Properties*], щелкнув на соответствующей пиктограмме панели инструментов (рисунок 3.4);

измените содержимое полей **Name** и **Label** на ***Sick***;

нажмите «*OK*».

Name – это внутреннее имя вершины, пока **Label** является названием вершины. Если поле **Label** не задано (как это было до того момента, пока мы не изменили поле **Label**), то его значение соответствует внутреннему имени. Внутреннее имя может содержать символы 'a' - 'z' и 'A' - 'Z', цифры '0' - '9', а также знак подчеркивания '_'.

Вершины ***Dry*** (“Засохло”) и ***Loses*** (“Облетело”) добавляются в БСД таким же образом. Можно добавлять вершины, не нажимая каждый раз на пиктограмму вершин с дискретными состояниями, а однажды выбрав режим добавления и удерживая клавишу *SHIFT,* щелкнуть мышью в окне отображения сети столько раз, сколько вершин с необходимо добавить в БСД.

Установление причинно-следственных связей между вершинами проектируемой БСД

На текущий момент проектируемая БСД имеет вид, похожий на тот, что приведен на рисунке 3.5. Следующий этап проектирования БСД состоит в установлении причинно-следственных связей между событиями. Эти связи в модели БСД отображаются в виде стрелок, соединяющих между собой вершины БСД. Для добавления стрелок от вершины ***Sick*** к вершины ***Loses*** и от ***Dry*** к ***Loses*** необходимо сделать следующее:

нажмите иконку добавления связей (рис. 3.4);

протяните мышью стрелку от ***Sick*** к ***Loses***, нажав левую кнопку и удерживая нажатой клавишу *SHIFT*;

протяните мышью стрелку от ***Dry*** к ***Loses,*** нажав левую клавишу.

Теперь Вы имеете полное качественное представление, подобное изображенному на рисунке 3.2. Следующий шаг – это задание состояний и таблиц условных вероятностей для каждой вершины.

Рисунок 3.5

Определение всех возможных состояний каждой из вершин БСД

Ранее, каждую из вершин мы определили так, что каждая из них может находиться в одном из двух состояний: вершина ***Sick*** в состояниях ‘sick’ (’болеет’) и ‘not’ (нет), вершина *Dry* в состояниях ‘dry’ (’засохло’) и ‘not’ (нет), а вершина *Loses* в состояниях - ‘yes’(да) и ‘no’(нет). Как задаются состояния каждой из вершин БСД в системе HUGIN, рассмотрим на примере определения состояний вершины ***Sick***. Для этого необходимо:

сделать активной вершину Sick, выбрав ее в раскрывающемся списке на панели инструментов или дважды щелкнув на ней мышью;

нажать пиктограмму добавления состояний на панели инструментов (рисунок 3.6).

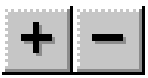


Рисунок 3.6

Щелкнув мышью, перевести курсор в ТУВ на поле, содержащем текст State 0 и ввести в него текст ‘sick’, задавая одно из состояний вершины. Затем State 1 заменить на ‘not’, задавая второе состояние вершины.

Самостоятельно проделайте аналогичные операции для вершины Dry, определяя ее состояния ‘dry’ и ‘not’, и вершины Loses, определяя ее состояния ‘yes’ и ‘no’. Обратите внимание, что при определении вершины Loses, таблица ее условных вероятностей будет отличаться от ТУВ вершин Sick и Dry. Это связано с тем, что вершина Loses имеет родительские вершины, которых Sick и Dry не имеют.

Задание значений таблиц условных вероятностей каждой из вершин БСД

Следующий шаг проектирования БСД – это корректное задание значений ТУВ каждой из вершин. По умолчанию система HUGIN для всех вершин устанавливает равномерное распределение. Для рассматриваемого примера значения ТУВ были заданы в таблицах 1, 2, 3 и в новых обозначениях будут иметь вид:

Таблица 4

P (Sick)	
Sick=’sick’	Sick=’not’
0.1	0.9

Таблица 5.

P (Dry)	
Dry=’dry’	Dry=’not’
0.1	0.9

Таблица 6.

P (Loses | Sick, Dry)				
	Dry=’dry’		Dry=’not’	
	Sick=’sick’	Sick=’not’	Sick=’sick’	Sick=’not’
Loses=’yes’	0.95	0.85	0.90	0.02
Loses=’no’	0.05	0.15	0.10	0.98

Процесс заполнения таблиц условных вероятностей рассмотрим на примере заполнения ТУВ для вершины ***Sick***.

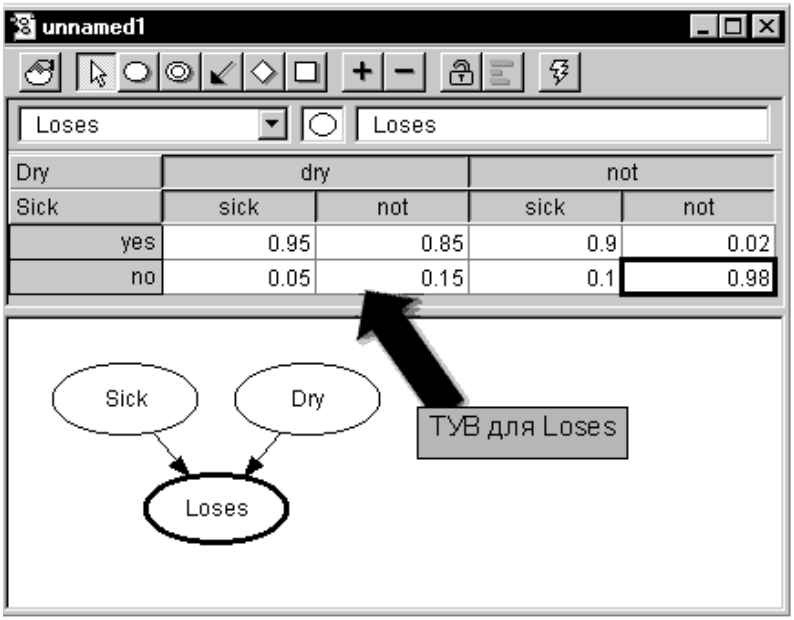


Рисунок 3.7

Выберите вершину ***Sick***.

Щелкните на поле содержащем ***Sick*** = '**sick'**.

Введите значение 0.1, соответствующее таблице 4.

Щелкните на поле содержащем ***Sick*** = '**not'**.

Введите значение 0.9, соответствующее таблице 4.

Таким же образом задайте значения для ***Dry*** и ***Loses***. Когда будите вводить значения для ***Loses***, будьте внимательны, чтобы значения соответствовали тем, что даны в таблице 6.

После задания таблиц условных вероятностей для всех вершин байесовской сети доверия окно отображения сети будет выглядеть так, как показано на рисунке 3.7.

На этом конструирование проектируемой БСД заканчивается.

Теперь необходимо сохранить результаты работы. Это можно сделать следующим образом:

выберите «Save As» в меню «File»;

введите имя файла (например apple.hkb или lab1.hkb);

нажмите «Save».

Компилирование спроектированной БСД и работа с ней

После того как БСД спроектирована и определена в системе HUGIN, ее необходимо скомпилировать и посмотреть, как она работает. Для этого необходимо перевести систему в режим вычислений. С этой целью щелкните мышкой по пиктограмме режима вычислений [run mode tool button] в панели инструментов (рисунок 3.8).

Рисунок 3.8

Если Вы четко следовали описанию, то ошибок компиляции не будет. Компиляция такой малой БСД закончится быстро, и система перейдет в режим вычислений [« run» mode].

В этом режиме окно сети поделено на две вертикальные секции (рисунок 3.9). Левая секция представляет собой панель списка вершин [*node list pane*], а правая – панель отображения сети [*network pane*].

В левой части окна отображаются вершины БСД, все возможные состояния каждой из вершин, а также вероятности пребывания вершины в каждом из состояний. В секции списка вершин [*node list pane*]. Посмотреть эти вероятности можно, дважды щелкнув на имени вершины в списке вершин. Теперь раскроем вершины ***Loses*** и ***Sick***. Для этого дважды щелкните на имени вершины ***Loses***, а затем дважды щелкните на ***Sick***. Вы также можете просмотреть вероятно-

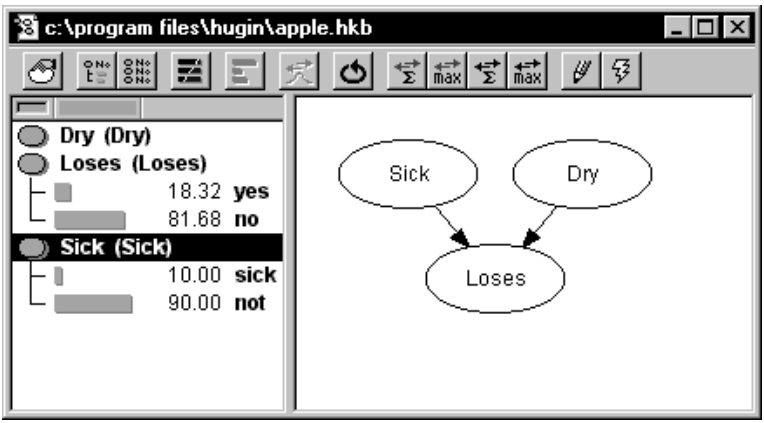


Рисунок 3.9

сти пребывания всех вершин во всех возможных состояниях, нажав пиктограмму раскрытия списка вершин [*expand node list tool*], вторую слева на панели инструментов.

Распространение вероятностей в БСД при поступлении новых свидетельств

Теперь, представим себе, что Вы хотите использовать БСД для нахождения вероятности болезни яблочного дерева, имея информацию о потери им листьев. Другими словами, в базу знаний экспертной системы добавляется новый факт о потере деревом листьев и нас интересует результат вывода ЭС о болезни яблочного дерева. Реализации такого типа запроса к экспертной системе на основе БСД выполняется следующим образом:

раскройте список всех вершин, нажав [*expand node list tool*];

введите факт потери листьев, дважды щелкнув на состоянии '**yes'** для вершины ***Loses***;

Рисунок 3.10

нажмите пиктограмму распространения (рисунок 3.10) на панели инструментов для распространения данного факта на всю БСД.

Посмотрите вероятность пребывания события ***Sick*** ("Болеет") в состоянии '**sick'** ('болеет'). Результаты распространения вероятностей по байесовской сети доверия приведены на рисунке 3.11.

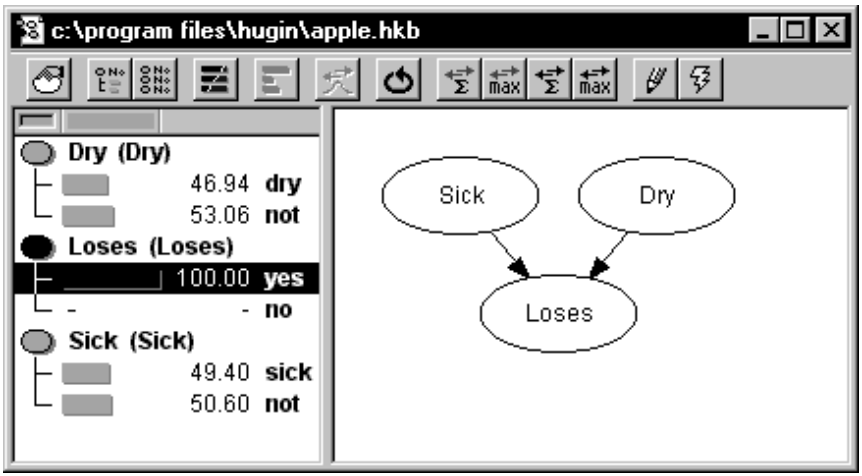


Рисунок 3.11

Вероятность того, что дерево болеет при учете причинно-следственных связей, отмеченных в БСД, и наличие факта опадания листвы составит 0.49.

Если Ваше значение не совпадает с представленным выше, возможно Вы что-то неправильно ввели, когда заполняли таблицы условных вероятностей. Проверьте таблицы для всех вершин.

Окна контроля

В последнем параграфе панель списка вершин использовалась для ввода фактов и нахождения вероятностей. Тоже самое Вы можете сделать, используя окно контроля [*monitor windows*]. Окна контроля показывают ту же информацию, что и панель списка вершин, но появляется возможность располагать эти окна рядом с соответствующими узлами БСД в панели отображения сети. Вы можете открыть такое окно для любой вершины, но лучший вариант – это открывать лишь те окна, которые представляют особый интерес. Иначе, эти окна будут занимать слишком много места на экране дисплея.

Теперь откроем такие окна для вершин ***Sick*** и ***Loses*** и повторим расчет. Сначала, инициализируем БСД, т.е. приведем ее к исходному состоянию. Для этого:

нажмите пиктограмму начальной инициализации в виде кольца со стрелкой [*initialize tool button*], расположенную слева от пиктограммы распространения.

После того, как сеть инициализирована, Вы можете открыть окна контроля для вершин Sick и Loses. С этой целью:

выберите Sick и Loses (удерживайте нажатой клавишу SHIFT для выбора более чем одной вершины).

Выберите «Show Monitor Windows» (Показывать окна контроля) из пункта меню «View» (Вид).

Наиболее вероятные комбинации состояний

На основе анализа поступившего факта опадания листвы можно с вероятностью 0,47 сделать предположения, что дерево засыхает и с вероятностью 0,49, что дерево болеет. При этом как для события *Sick*, так и для события *Dry* более вероятными являются состояние ‘not’. Это может породить уверенность, что наилучшая комбинация состояний, когда *Sick* и *Dry* находятся в состоянии ‘not’. Однако это неверный вывод.

Для того чтобы найти наиболее вероятную комбинацию состояний всех вершин, необходимо вместо распространения сумм использовать распространение максимумов. Пиктограмма распространения максимумов [*Max propagation tool*] находится на панели инструментов справа от пиктограммы распространения сумм.

Нажмите на эту пиктограмму, и Вы получите новое распространение в окнах отображения сети и вершин. При этом каждое из состояний вершин, имеющее значение 100,00, будет принадлежит к наиболее вероятной комбинации состояний. Для рассматриваемого примера будет получена одна уникальную комбинацию, при которой наиболее вероятным является, что событие *Sick(Болеет)* находится в состоянии ‘sick’, а событие *Dry(Засохло)* – в состоянии ‘not’.

Из полученного результата следует, что несмотря на то, что состояние *Sick*=’sick’ менее вероятно, чем состояние *Sick*=’not’, вместе с тем именно это состояние *Sick*=’sick’ входит в наиболее вероятную комбинацию состояний вершин БСД при поступлении факта об опадании листвы, в то время как состояние *Sick*=’not’ в нее не входит. Это говорит о том, что необходимо быть очень осторожными в своих выводах о результатах распространения вероятностей.

После получения вывода о наиболее вероятной комбинации состояний вершин БСД может появиться желание узнать вероятность наступления этой (или любой другой) комбинаций состояний при условии наступления введенных в БСД свидетельств.

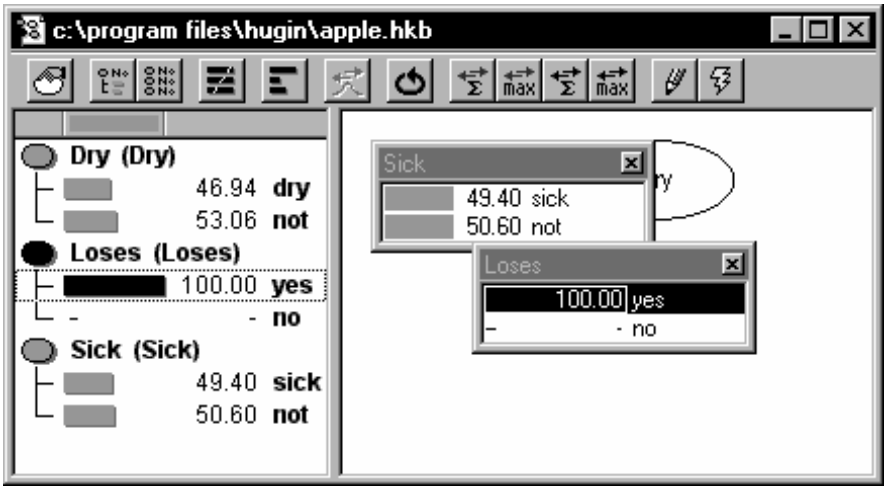


Рисунок 3.12

Расчет вероятности комбинаций состояний

Здесь мы расскажем о технике расчета вероятности наиболее вероятной комбинации состояний, полученных при наличии факта, что яблочное дерево теряет свои листья. Эта вероятность записывается в виде:

P (*Sick*=’sick’, *Dry*=’not’ | *Loses*=’yes’)

и для ее определения можно воспользоваться известной формулой из теории вероятности:

$$P(A, B) = P(A \mid B)\, P(B),$$

переписав ее в виде:

$$P(A \mid B) = P(A, B) / P(B),$$

Каждый раз при выполнении расчета распространения сумм для БСД, в нижнем левом углу HUGIN будет отображаться вероятность (значение P(All)) одновременного наступления нескольких событий, которую можно записать следующим образом:

$$P(A_1, A_2, ..., A_n)$$

Если теперь мы заменим A на *Sick*=’yes’, *Dry*=’not’ и B на *Loses*=’yes’, то вышеприведенная формула для определения условной вероятности может быть переписана в виде:

P(*Sick*=’yes’, *Dry*=’not’ | *Loses*=’yes’) = P(*Sick*=’sick’, *Dry*=’not’, *Loses*=’yes’) / P(*Loses*=’yes’).

Если выбрать состояние ‘yes’ для вершины *Loses* и сделать расчет распространения сумм, то можно увидеть вероятность пребывания вершины *Loses* в состоянии ‘yes’. Ее значение будет 0.1832. (т.е. P(Loses=’yes’)=0.1832). Определив, чему равно P(*Loses*=’yes’), нам осталось определить, чему равно P(*Sick*=’sick’, *Dry*=’not’, *Loses*=’yes’). Это делается следующим образом:

введите *Sick*=’sick’, *Dry*=’not’, и *Loses*=’yes’ в БСД.

Выполните расчет распространение сумм (нажмите *sum propagation tool*).

Посмотрите P(*Sick*=’sick’, *Dry*=’not’, *Loses*=’yes’) как значение P(All) в нижнем левом углу.

Это значение должно быть равно 0.081. Теперь мы готовы посчитать требуемую нам вероятность, которая будет равна

P(*Sick*=’yes’, *Dry*=’not’ | *Loses*=’yes’) = 0.081 / 0.1832 = 0.442

Таким образом, на основе поступившего факта опадания листвы экспертная система делает вывод о том, что наиболее вероятной является ситуация, связанная с тем, что дерево заболело и не засохло. При этом вероятность такой ситуации равна 0.442.

Основные компоненты байесовских сетей доверия с непрерывными и дискретными состояниями

Система Hugin позволяет оперировать не только дискретными, но и непрерывными случайными событиями. Однако при этом существует ряд ограничений:

в таких сетях нельзя использовать вершины решения и полезности, т.е. строить ДВ;

дискретные вершины шансов не могут иметь непрерывных родителей;

непрерывные вершины шансов должны иметь нормальный закон распределения, условный на значениях родителей;

распределение непрерывной переменной Y с дискретными родителями I и непрерывными родителями Z является нормальным распределением:

$$P(Y \mid I = i,\ Z = z\ ) = N(m_y\ (m_i,\ m_z\ ),\ s_{\ y}\ (s_i)).$$

При этом m_y зависит как от дискретных, так и непрерывных родителей, причем зависимость от непрерывных родителей линейна, а $s_{\ y}$ зависит только от дискретных родителей. Это ограничение гарантирует возможность точного вывода.

3.3. Проектирования простейшей ЭС на основе БСД с непрерывными и дискретными событиями

Описание приемов работы проведем на примере, изложенном в параграфе 2.10.

Анализируемая модель БСД имеет вид, приведенный на рисунке 3.13, где одинарные овалы соответствуют дискретным событиям, а двойные овалы – непрерывным событиям (гауссовским переменным).

Для того чтобы данная качественная модель превратилась в полную БСД, необходимо определить ее количественные характеристики. Вершина “Загрузка оборудования” соответствует дискретному событию, которое характеризуется тремя возможными состояниями. Вероятность пребывания в каждом из них определяется степенью за-

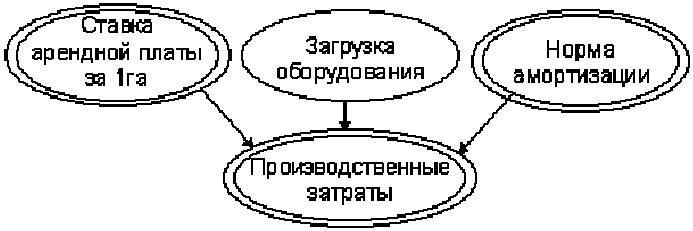


Рисунок 3.13

грузки каждой из групп оборудования, при условии, что суммарная загрузка всего оборудования равна единице. Если считать, что все группы оборудования загружены равномерно, то распределение вероятностей для этой вершины будет иметь вид:

Таблица 7.

P(“Загрузка оборудования” = ’Пилорама 1’	P(“Загрузка оборудования” = ’Пилорама 2’	P(“Загрузка оборудования” = ’Пилорама 3’
0.333	0.333	0.333

При этом следует отметить, что возможны и любые другие исходные распределения вероятностей, учитывающие различные варианты загрузки оборудования. Полученные от экспертов знания о том, что ставка аренды 1 га земли в среднем составляет 2500 у.е. и колеблется в пределах ± 10%, т.е. принимает значения 2500 ± 250 у.е.,

а норма амортизации может находиться в пределах 5 - 10 % от балансовой стоимости, т.е. принимать значения 7,5 ± 2,5% (или 0,075 ± 0,025 относительных единиц)

позволяют определить параметры ещё двух вершин – “Ставка арендной платы» и “Норма амортизации”. Предполагая, что эти вершины шансов являются непрерывными случайные переменными с гауссовским законом распределения, необходимо задать параметры этих законов для каждой из вершин, которые будут иметь вид:

Таблица 8

	“Ставка арендной платы”	“Норма амортизации”
математическое ожидание (m)	2500	0,075
дисперсия ($D = s^2$)	62500 (250^2)	0,000625 ($0,025^2$)

Что касается вершины “Производственные затраты”, то она характеризуется случайной переменной, условно нормальной на значениях родителей (т.е. на значениях трёх других вершин нашего примера). Следует отметить, что в общем случае распределение ве-

роятностей для вершин, аналогичных “Производственные затраты» является не просто нормальным, а смешанным нормальным распределением, т.е. представляет собой весовую сумму распределений, для каждого из которых должен быть задан список его параметров:

математические ожидания и дисперсии для распределений, описывающих степень влияния дискретных родителей;

весовые коэффициенты, учитывающие степень влияния на математическое ожидание непрерывных родителей.

Если теперь, используя экспертное оценивание, предположить, что:

балансовая стоимость каждой из пилорам составляет 50000, 40000 и 30000 у.е.,

площадь арендуемых участков, закрепляемая за ними равна 0,6; 0,5 и 0,4 га,

а оценка прямых затрат на поддержание нормальной работы каждой из пилорам в среднем составляет 3000, 3200 и 3500 у.е. и получена с 5% точностью,

то степень влияния родительских вершин на “Производственные затраты» можно представить таблицей вида:

Таблица 9

Параметры, определяющие распределение вероятностей для вершины “Производственные затраты”			
“Загрузка оборудования”	Пилорама 1	Пилорама 2	Пилорама 3
- математическое ожидание (m)	3000	3200	3500
- дисперсия ($D = s^2$)	22500 ($=150^2$)	25600 ($=160^2$)	30625 ($=175^2$)
“Норма амортизации”	50000	40000	30000
“Ставка аренды”	0,6	0,5	0,4

На основе проведенного экспертного оценивания предметной области можно теперь переходить к реализации БСД в системе Hugin и построению на ее базе ЭС, которая позволит получить оценку любых ее состояний, при любых произвольно задаваемых свидетельствах.

Формирование проектируемой модели БСД в системе Hugin

Реализация модели БСД с непрерывными и дискретными состояниями включает в себя несколько этапов, первым из которых является этап построения вершин шансов и установления связей между

ними. Для того чтобы включить в состав БСД новые дискретные или непрерывные вершины шансов, следует:

на панели инструментов выбрать кнопку с пиктограммой в виде одинарного («*Discrete Chance Tool*») или двойного эллипса («*Continuous Chance Tool*»), соответственно (рисунок 3.14);

подвести указатель мыши к тому месту на панели отображения сети, где будет располагаться вершина, и нажать левую клавишу мыши.

Рисунок 3.14

После того, как все необходимые вершины будут включены в модель БСД, необходимо определить причинно-следственные связи между событиями, соответствующими вершинам БСД. Для этого надо выбрать на панели инструментов пиктограмму со стрелочкой («*Link Tool*»), подвести курсор к вершине С1 и, удерживая кнопку мыши нажатой, провести линию к С3. Затем проделать то же самое и с другими вершинами (рисунок 3.15).

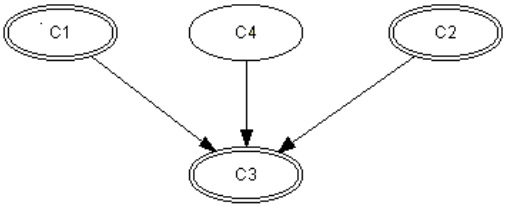


Рисунок 3.15

В тех случаях, когда необходимо добавить несколько однотипных элементов БСД, можно на панели инструментов выбрать нужный компонент, затем переместить курсор на рабочую область, нажать клавишу Shift и, удерживая ее нажатой, построить необходимое количество элементов

Вторым этапом процесса реализации модели БСД в системе Hugin является присвоение вершинам шансов имен и ярлыков. С этой целью необходимо выполнить следующую последовательность действий:

выделить курсором мыши нужную вершину;

выбрать на панели инструментов пиктограмму «*Node Properties*» или, нажав правую кнопку мыши, выбрать эту команду в контекстном меню;

в форме свойств (рисунок 3.16) в полях *Name* и *Label* задать новое имя и метку вершины (наименования только латинскими буквами

без пробелов, метки – в зависимости от подгруженных шрифтов, но желательно латинскими).

Если заданная метка вершины слишком длинная и не помещается в стандартные границы овалов вершин шансов, то имеется возможность изменить их параметры. Для этого:

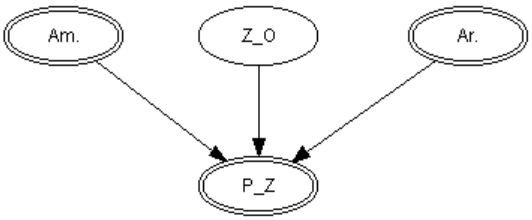


Рисунок 3.17

в пункте «*Network*» главного меню выберите опцию «*Network Properties*» (или нажмите сочетание клавиш Alt-Enter);

затем в свойствах «*Display*» окна «*Node Size*» задайте новые размеры узлов сети.

Окончательный вариант модели БСД с непрерывными и дискретными вершинами шансов для нашей задачи будет иметь вид, приведенный на рисунке 3.17, где Am.– амортизация; Ar.– арендная плата; Z_O – загрузка оборудования; P_Z – производственные затраты.

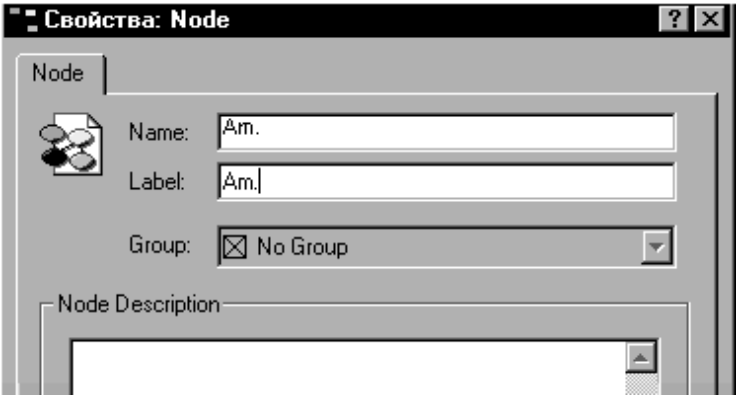


Рисунок 3.16

На третьем этапе реализации БСД необходимо дать количественное описание сети и всех ее элементов, т.е. определить все возможные состояния каждой из вершин сети и задать их вероятностные характеристики.

Для задания количественных параметров дискретной вершины шансов необходимо выполнить следующую последовательность действий:

двойным щелчком выделите вершину, соответствующую загрузке оборудования (Z_O);

на панели редактирования появится таблица состояний (*State*) дискретной вершины (рисунок 3.18);

кнопкой «*Add State*» добавьте нужное число состояний;

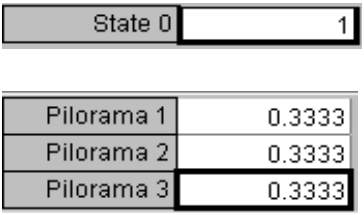


Рисунок 3.18

для переименования состояния выделите двойным щелчком левой клавиши мыши редактируемую область, и введите новое название состояния;

проставьте соответствующие априорные вероятности возможных состояний загрузки оборудования (рисунок 3.18).

Для задания количественных параметров непрерывных вершин шансов выполните следующую последовательность действий:

двойным щелчком выделите вершину, соответствующую ставке арендной платы за 1 га (Ar);

задайте значения математического ожидания (*Mean*) и дисперсии (*Variance*) нормального закона распределения вероятностей для непрерывного случайного события, соответствующего значению арендной ставки (рисунок 3.19);

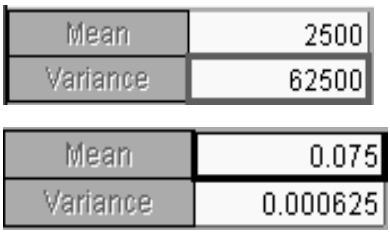


Рисунок 3.19

аналогичную операцию проделайте с вершиной (Am), определяющей случайное событие, соответствующее значению нормы амортизационных отчислений (рисунок 3.19);

двойным щелчком левой клавиши мыши выберите вершину шансов, соответствующую суммарным производственным затратам (P_Z);

используя данные таблицы 9, заполните таблицу условных вероятностей для вершины производственных затрат (рисунок 3.20).

На этом формирование БСД для рассматриваемого примера заканчивается и можно переходить к ее проверке, исследованию и эксплуатации в качестве экспертной системы.

Прежде, чем перейти к выполнению следующих этапов, не забудьте сохранить Вашу модель. Для этого войдите в меню File, выберите команду Save As ... и задайте имя для Вашей БСД.

Технология работы с БСД, реализованной в системе Hugin

Для тестирования и исследования сформированной модели необходимо ее откомпилировать и запустить на выполнение, т.е. из режима редактирования перейти в режим вычислений. Для этого нажмите кнопку «*Run*» на панели инструментов. В результате выпол-

Z_O	Pilorama 1	Pilorama 2	Pilorama 3
Mean	3000	3200	3500
Variance	22500	25600	30625
Am.	50000	40000	30000
Ar.	0.6	0.5	0.4

Рисунок 3.20

нения этой команды на экране появится модель БСД, представленная в двух видах: в виде списка вершин (« *Node List*») и в виде графического отображения сети.

Когда все состояния случайных событий описаны и модель переведена в режим вычислений, Вы можете исследовать влияние одних компонентов модели на другие. При этом надо владеть технологией работы со списком вершин и графическим отображением их состояний на модели БСД.

При переходе в режим вычислений в левой части экрана отображается список всех вершин исследуемой БСД, который будет иметь вид, аналогичный приведенному на рисунке 3.21.

Для того чтобы вызвать на экран состояния той или иной вершины (рисунок 3.21), надо щелкнуть левой клавишей мыши по ее изображению в списке вершин или дважды щелкнуть левой клавишей мыши на овале вершины в панели графического отображения сети.

Повтор этой операции вернет список в исходное состояние (рисунок 3.21). Эту процедуру можно проделывать и с отдельными (выборочными) вершинами, и со всеми одновременно.

Для одновременного вывода на экран всех состояний всех вершин модели следует воспользоваться кнопкой «*Expand Node List*» (раскрыть список вершин) на панели инструментов. На экране появится результат, аналогичный рисунок 3.22.

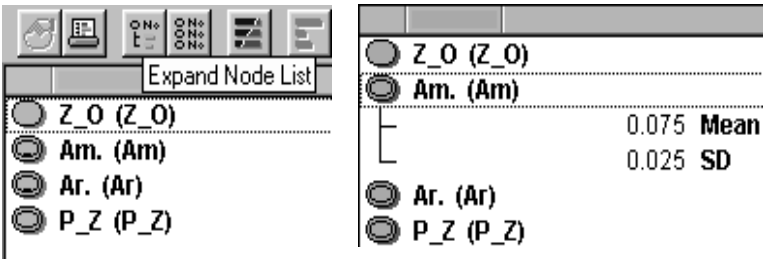


Рисунок 3.21

Для сворачивания раскрытого списка используйте кнопку «*Collapse Node List*». При этом будут свернуты все вершины модели независимо от того, использовали ли Вы инструмент «*Expand Node List*» или вручную раскрывали выборочные вершины.

С правой частью окна БСД, где представлено графического отображения сети, можно работать так же, как и со списком вершин, просматривая как числовые параметры, так и графики плотностей распределения каждой из вершин БСД. Рассмотрим технологию работы с графическим отображением параметров вершин:

выберите вершину, состояние которой хотите показать на экране, и выделите ее щелчком левой клавиши мыши;

нажатием правой клавиши мыши вызовите на экран контекстное меню и выберите опцию *Show Monitor*. На экране появится графическое окно с состояниями вершины;

нажмите кнопку с изображением графика (это возможно только для непрерывной случайной величины);

Для обновления графика плотности распределения вероятностей непрерывной случайной величины нажмите кнопку «*Update Monitor Graphs*» на панели инструментов или в нижнем правом углу графического окна вершины.

Для того чтобы изменить интервал построения графика, можно установить нужный диапазон значений среднеквадратических отклонений, отображаемых влево и вправо от математического ожида-

ния. Эти значения выбираются из двух выпадающих списков окна графического отображения вершины (рисунок 3.23).

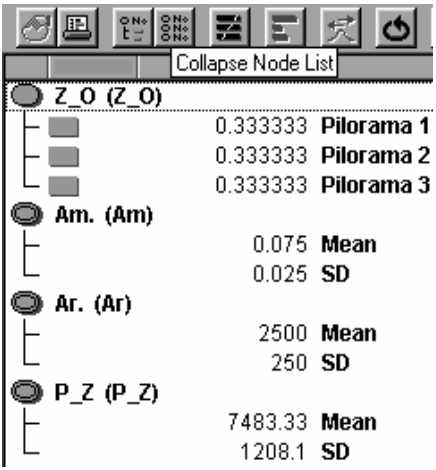


Рисунок 3.22

После любого изменения параметров графической области необходимо нажимать кнопку «*Update Monitor Graph*» *s*.

Кроме того, система Hugin позволяет одновременно для всех вершин сети отобразить их параметры в виде графических окон (рисунок 3.24). Для этого необходимо:

выбрать все вершины на панели отображения сети, что возможно двумя способами:

либо мышью при нажатой клавише Shift;

либо вызовом на экран контекстного меню (нажав правую клавишу мыши) и выбором в нем опции «Select All»;

вызвать на экран контекстное меню и выбрать в нем опцию «Show Monitor»;

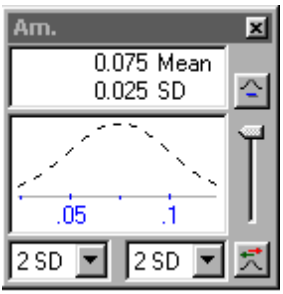


Рисунок 3.23

нажать на пиктограмму с символом добавления графика и затем, когда появится окно графика, нажать кнопку «*Update Monitor Graphs*» на панели инструментов (в случае работы со всеми вершинами сети этот способ более рационален);

если графики недостаточно наглядные, изменить интервалы отклонений от математических ожиданий и нажать кнопку «*Update Monitor Graphs*».

После знакомства с основами технологии работы с БСД, можно переходить к эксплуатации спроектированной БСД в качестве экспертной системы.

Задание свидетельств

Процесс эксплуатации БСД как экспертной системы заключается в автоматическом распространении вероятностей (логическом выводе), который осуществляется в системе при поступлении в нее свидетельств о наступлении тех или иных событий или вероятностях их возникновения. Для того чтобы задать в системе какое-либо свидетельство о событии, надо:

выбрать требуемое событие в списке вершин, либо в соответствующем окне графического отображения вершины;

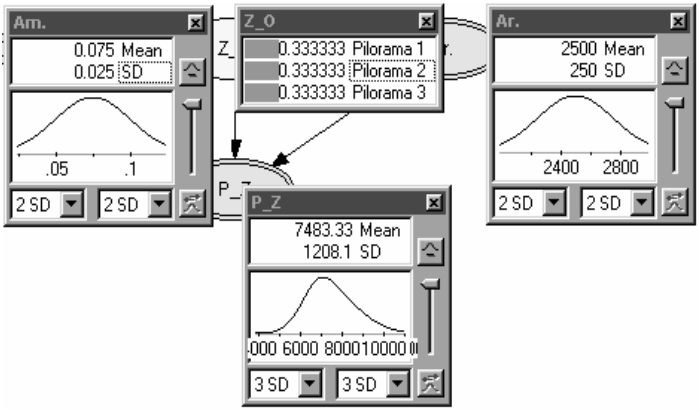


Рисунок 3.24

если это дискретное событие и свидетельство соответствует наступлению какого-либо одного из его состояний, то необходимо двойным щелчком левой клавиши мыши зафиксировать одно из его состояний (вероятность его наступления станет равна 1);

если это дискретное событие и свидетельство соответствует возможности наступления нескольких его состояний с заданной вероятностью, то необходимо нажать кнопку «*Enter Likelihood»* на панели инструментов (рисунок 3.25), и в появившемся на экране окне «*Likelihood»* задать распределение вероятностей для всех состояний задаваемого свидетельства;

Рисунок 3.25

если это непрерывное событие, то двойной щелчок левой клавиши мыши на его графическом изображении вызовет на экран окно «*Continuous*

Finding» (рисунок 3.26), где необходимо в строке *Evidence/Value* задать новое числовое значение математического ожидания этого события, соответствующее задаваемому свидетельству;

после задания одного или группы свидетельств, следует провести распространение сумм вероятностей на все составляющие БСД, нажав кнопку «*Prop Sum Normal «*панели инструментов;

для просмотра изменений на графиках используйте кнопкой «*Update Monitor Graphs»*;

для того чтобы привести систему в исходное состояние (т.е. отменить введенные свидетельства), надо использовать кнопку «*Initialize»* на панели инструментов.

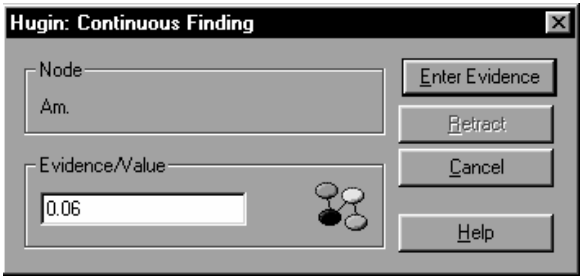


Рисунок 3.26

Пример эксплуатации ЭС для решения простейшей задачи

Пусть нам требуется оценить возможное изменение суммарных производственных затрат при загрузке только одной из групп оборудования при допустимом небольшом снижении норм амортизации.

Для ответа на такого типа запрос необходимо в исходную БСД ввести два новых свидетельства:

первым является свидетельство о том, что норма амортизации оборудования в рассматриваемом случае может составлять 6% (т.е. математическое ожидание данного непрерывного свидетельства будет равна 0.06, а дисперсия равна 0);

вторым является свидетельство об использовании только одной, а именно 1-ой группы оборудования (т.е. *P(Pilorama1)=1, P(Pilorama2)=0, P(Pilorama3)=0*).

Проведя распространение распределения вероятностей и параметров гауссовских законов распределения на остальные составляющие БСД, мы видим, что производственные затраты увеличились по сравнению с первоначальным результатом (рисунок 3.27).

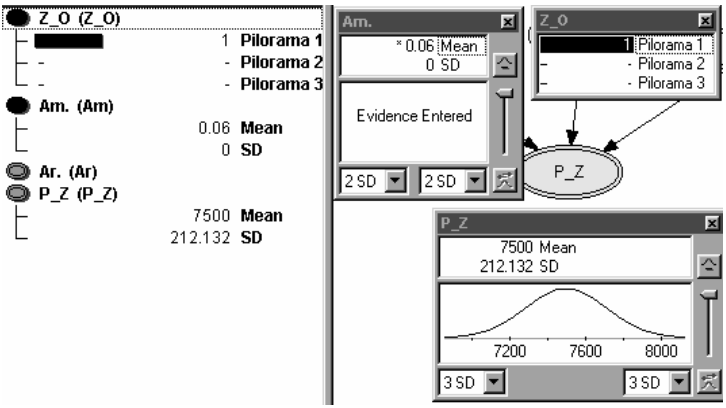


Рисунок 3.27

Проанализировав подобным образом, использование других пилорам, можно сделать вывод о том, что при заданном значении нормы амортизации и известных параметрах другой непрерывной случайной величины – арендной платы, наиболее эффективным является использование третьей пилорамы.

Однако Вы можете задавать свидетельства не только вершинам, влияющим на конечный результат, но и самому конечному результату. Это позволит Вам исследовать обратный процесс и узнать, какие состояния должны иметь исходные вершины, чтобы получился заданный результат.

3.4. Основы проектирования диаграмм влияния и методы работы с ними в системе Hugin

Пример построения простейшей диаграммы влияния

Описание примера приведено в параграфе 2.9.

Используя правила построения графических моделей, изложенные в предыдущих параграфах, нетрудно перейти от описательной модели к модели в системе Hugin (рисунок 3.28).

Модель, представленная на рисунке 3.28, даёт законченное качественное представление ДВ. Для получения количественного представления необходимо построить таблицы условных вероятностей для каждой из вершин шансов и задать таблицы доходности для ка-

ждой из вершин полезности. Вершины принятия решения не имеют таблиц условных вероятностей.

При этом таблицы условных вероятностей для P(“Болеет”), P(“Засохло”) и P(“Облетело”|“Болеет”,“Засохло”) будут иметь тот же вид, что и в параграфе 3.2. ТУВ для P(“Облетело1”|“Болеет1”,“Засохло1”) будет аналогична P(“Облетело”|“Болеет”, “Засохло”). ТУВ для остальных состояний должны быть заданы на основе знаний эксперта.

Цель диаграммы влияния – принятие решения, связанного с вершиной “Лечение” для того, чтобы получить наибольшую ожидаемую выгодность. Даже в таком простом примере ручной расчёт достаточно сложен и поэтому возникает необходимость в использовании инструмента для разработки ЭС, такого, например, каким является Hugin. Ответом для принятия решения о затратах на лечение

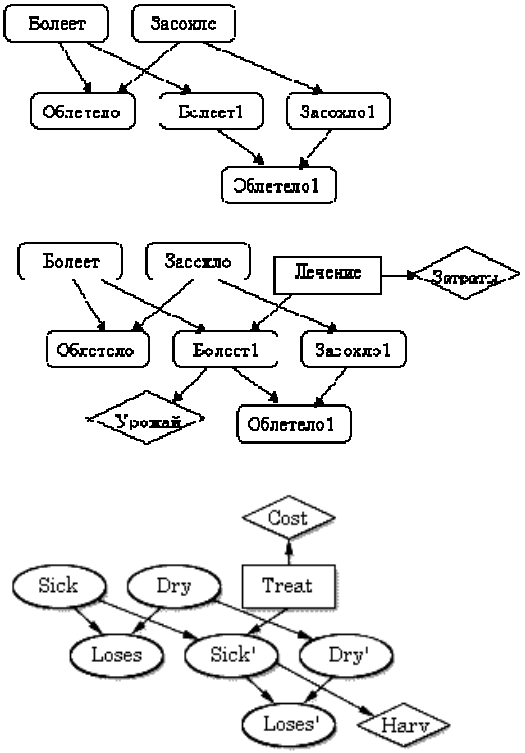


Рисунок 3.28

будет вычисление общей функции полезности при условии, что P(“Облетело”=’облетело’)= 1, а реализуемая модель имеет вид, аналогичный приведенному на рисунке 3.28.

Загрузка байесовской сети доверия для редактирования>

Для того чтобы построить требуемую ДВ на основе уже спроектированной БСД, необходимо открыть БСД, созданную, на основе параграфа 3.2. Это можно сделать следующим образом:

выберите опцию «*Open»* из пункта меню «*File»*;

введите название файла вашей БСД (например, *apple.hkb*) или выберите его из списка;

нажмите «*Enter».*

На рисунке 3.29 показана открытая БСД, находящаяся в режиме редактирования. Если при открытии (или в процессе работы) система находится в режиме вычислений (*run mode*), необходимо перейти в режим редактирования (*edit mode*), выбрав мышкой кнопку с пиктограммой в виде карандаша.

Копирование вершин при разработке модели БСД

В диаграмме влияния (рисунок 3.28) есть три узла аналогичные тем, какие уже имеются в модели. Вы можете скопировать эту группу следующим образом:

растяните прямоугольник вокруг всех трех узлов (двигайте мышь, удерживая левую клавишу нажатой);

выберите опцию «*Copy»* (копировать) из пункта меню «*Edit»;*

затем выберите опцию «*Paste»* (вставить) в том же пункте «*Edit»;*

переместите группу в нужное место.

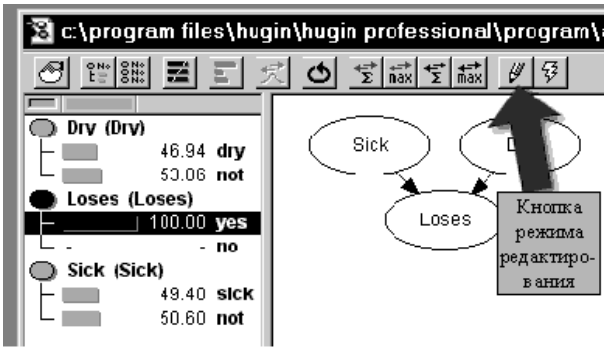


Рисунок 3.29

Система Hugin сгенерирует новые имена и метки для новых вершин. Можно сохранить имена или изменить их. Однако нельзя использовать имена *Sick', Dry'* или *Loses'*, так как они содержат управляющий символ “'”, который запрещено использовать в именах. Метки могут использовать произвольный набор символов и их можно изменить следующим образом:

выделите вершину курсором мыши;

откройте окно свойств вершины (узла). Для этого нажмите в панели инструментов кнопку «*Node Properties»* или выберите этот же режим из меню, вызываемого нажатием правой кнопки мыши;

измените значение поля «*Label»;*

нажмите «*OK*».

После выполнения этих действий для всех трех новых вершин, БСД будет выглядеть так, как показано на рисунке 3.30.

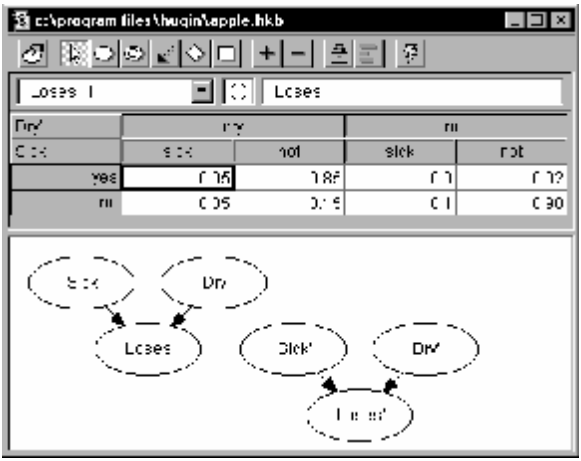


Рисунок 3.30

Следующим шагом необходимо добавить причинно-следственные связи (стрелки) от *Sick* к *Sick'* и от *Dry* к *Dry'*. Это можно сделать следующим образом (на примере вершин *Sick* и *Sick'*):

нажмите в панели инструментов кнопку с пиктограммой стрелки (*« Link Tool»*);

протяните стрелку от *Sick* к *Sick'* удерживая левую клавишу мыши в нажатом состоянии;

Если при этом удерживать нажатой клавишу *Shift*, то это даст возможность создавать несколько связей, не нажимая каждый раз на кнопку «*Link Tool»* панели инструментов.

Переход от БСД к ДВ путем добавление вершины полезности

На текущий момент построенная нами сеть пока еще является байесовской сетью доверия. Произведем первое изменение, которое приблизит нас к диаграмме влияния. Это изменение заключается в добавлении вершины полезности. Эта вершина будет характеризовать доход (полезность) от полученного урожая (вершина *Harv*). Для построения этой вершины необходимо выполнить следующие действия:

нажать кнопку «*Utility Tool»,* расположенную в панели инструментов справа от пиктограммы со стрелкой;

щелкнуть мышью в том месте рабочей области, где предполагается размещение добавляемой вершины (лучше справа от вершины *Loses'*);

изменить имя и метку новой вершины на *Harv*.

Так как полезность непосредственно определяется качеством урожая, который в свою очередь зависит от состояния здоровья деревьев к моменту сбора урожая, то необходимо установить причинно-следственную связь (добавить стрелку) от *Sick'* к *Harv*. Для этого:

нажмите в панели инструментов кнопку «Link Tool»;

протяните стрелку от *Sick'* к *Harv*.

Следующим этапом надо определить значения полезности вершины *Harv* для всех возможных состояний, связанной с ней вершины *Sick'*. С этой целью:

выделите вершину *Harv* (из раскрывающегося списка или двойным щелчком мыши);

в таблицу определения вершины *Harv* введите значения из таблицы 10.

Таблица 10

U (Harv).	
Sick=’sick’	Sick=’not’
3000	20000

Эти значения определяют возможный доход (полезность) от собранного урожая в зависимости от состояния сада и могут быть заданы на основе экспертных оценок, например, в денежном эквиваленте.

Добавление в ДВ вершины решения и еще одной вершины полезности

Процесс принятия решения с использованием проектируемой ДВ заключается в ответе на запрос о целесообразности дополнительных затрат на лечение деревьев с целью повышения их урожайности, а, следовательно, и дохода от собранного урожая. С этой целью, в соответствием с рис.4., добавим в проектируемую ДВ вершину решения *Treat* (от англ. обработка). Делается это так же, как и в случае с вершинами шансов или вершинами полезности:

нажмите в панели инструментов кнопку «Decision Tool» с пиктограммой прямоугольника (справа от «Utility Tool»);

щелкните мышью в рабочей области (лучше правее вершины Dry);

измените имя и метку созданной вершины на *Treat*.

Возможные состояния вершины решения задаются точно так же, как и состояния для вершин шансов, а именно:

выделите вершину Treat (из раскрывающегося списка или двойным щелчком мыши);

нажмите кнопку «добавить состояние (« Add State»);

измените название возможных состояний на ‘treat’ и ‘not’.

Решение о текущей обработке деревьев против их заболевания окажет влияние на их заболеваемость в момент уборки урожая, т.е. вершина *Treat* имеет влияние на вершину *Sick'*. Для учета в диаграмме влияния этой причинно-следственной связи:

добавьте стрелку от *Treat* к *Sick'*.

Таблица 11

U (Cost).	
Treat=’treat’	Treat=’not’
-8000	0

Вместе с тем решение об обработке деревьев требует определенных материальных затрат. Это моделируется вершиной полезности *Cost*. Значения величины полезности вершины *Cost* для всех возможных состояний вершины *Treat*, выраженные в том же самом эквиваленте, что и для вершины *Harv,* представлены в таблице 11. Добавим вершину полезности *Cost* к проектируемой диаграмме влияния. Для этого:

вместе с тем решение об обработке деревьев требует определенных материальных затрат. Это моделируется вершиной полезности *Cost*. Значения величины полезности вершины *Cost* для всех возможных состояний вершины *Treat*, выраженные в том же самом эквиваленте, что и для вершины *Harv* представлены в таблице 11. До-

бавим вершину полезности *Cost* к проектируемой диаграмме влияния. Для этого:

добавьте новую вершину полезности (желательно справа от вершины *Treat*);

измените имя и метку вершины на *Cost*;

Добавьте стрелку от *Treat* к *Cost*;

Заполните состояния для вершины полезности *Cost* в соответствии с таблицей 11.

Заполнение таблиц условных вероятностей (ТУВ)

После того как при проектировании ДВ были скопированы вершины *Sick'* и *Dry'*, они унаследовали от вершин *Sick* и *Dry* свои таблицы условных вероятностей. Однако, так как *Sick'* и *Dry'* являются зависимыми не от тех же самых вершин, что и *Sick* и *Dry*, то их ТУВ не соответствуют действительности и требуют корректировки. Новые таблицы условных вероятностей для этих вершин, полученные на основе обработки знаний экспертов, могут иметь вид, аналогичный приведенному в таблицах 12 и 13.

Для окончательного определения проектируемой диаграммы влияния необходимо:

заполнить ТУВ для вершины Sick' данными из таблицы 12.

заполнить ТУВ для вершины *Dry'*данными из таблицы 13.

Таблица 12

	P (Sick' | Sick, Treat).			
	Treat=’treat’		Treat=’not’	
	Sick=’sick’	Sick=’not’	Sick=’sick’	Sick=’not’
Sick'=’sick’	0.20	0.01	0.99	0.02
Sick'=’not’	0.80	0.99	0.01	0.98

Таблица 13

	P(Dry' | Dry)	
	Dry=’dry’	Dry=’not’
Dry'=’dry’	0.60	0.05
Dry'=’not’	0.40	0.95

После того, как к исходной БСД были добавлены три вершины шансов, вершина решения, две вершины полезности, а также заполнены все необходимые ТУВ, построение диаграммы влияния можно считать законченным.

Если все было сделано верно, то ДВ должна быть похожа на ту, что приведена на рисунке 3.31.

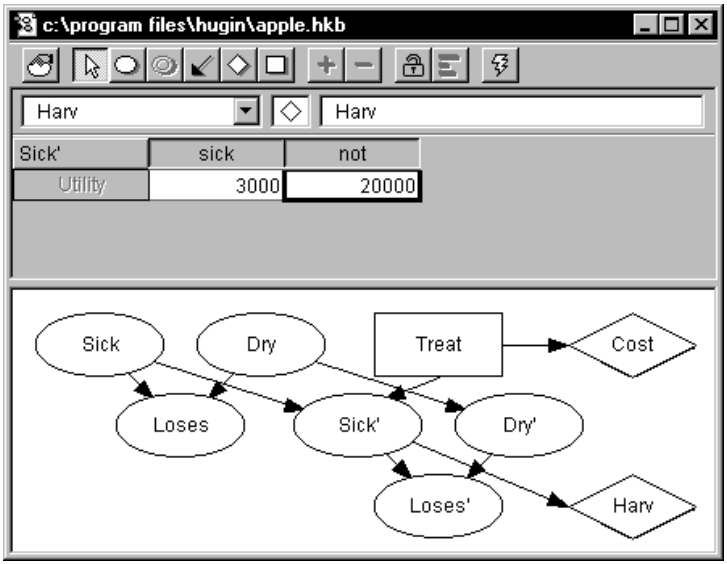


Рисунок 3.31

Процесс работы и принятия решения с использованием ЭС на основе ДВ

После завершения процесса моделирования знаний, экспертная система на основе ДВ готова к работе. Для работы с ней, прежде всего, необходимо ее откомпилировать. Для этого необходимо из режима редактирования перейти в режим в выполнения. С этой целью:

нажмите кнопку «*Run*» (самая правая на панели инструментов);

после успешной компиляции можно приступить к работе с ЭС и процессу принятия решения.

На начальном этапе, единственное, что знает садовод – это то, что его деревья теряют листву. Другими словами, единственным свидетельством, которое можно сообщить ЭС, является факт опадания листвы, т.е. тот факт, что событие *Loses* находится в состоянии ’yes’.

Какое решение лучше принять садоводу в данной ситуации? Для решения этой задачи:

отобразите в списке вершин левой части экрана допустимые состояния вершин Loses и Treat, щелкнув мышкой по их графическому обозначению в списке вершин;

установите в качестве абсолютно достоверного для вершины Loses состояние ‘yes’ (дважды щелкнув мышкой на этом состоянии);

проведите распространение сумм вероятностей, нажав кнопку «Prop Sum Normal»);

посмотрите на значения ‘treat’ и ‘not’ вершины *Treat*.

Если все проделано верно, то у Вас должно получиться что-то похожее на рисунке 3.32. Откуда видно, что значение суммарной полезности события Treat, определяемое на основе всех вершин полезности ДВ, принимает (для нашего примера в денежном эквиваленте) значения:

10234,40 – при условии проведения лечения деревьев;

11514,00 – в противном случае.

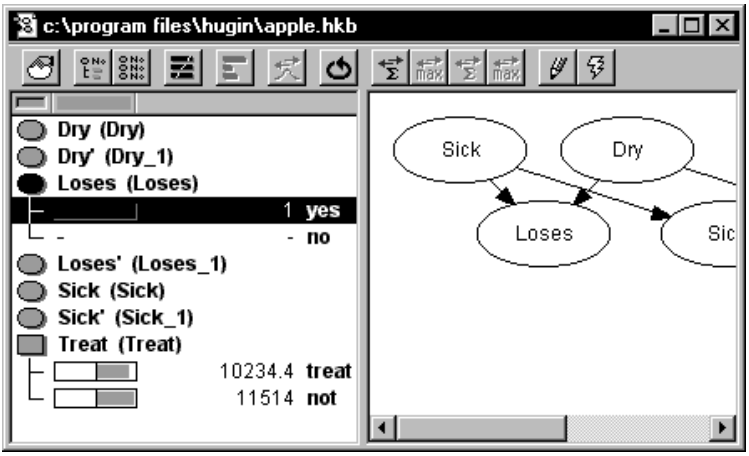


Рисунок 3.32

Это означает, что самое лучшее для садовода с точки зрения получения дохода от урожая – это ничего не делать, то есть не заниматься обработкой деревьев для их лечения, так как затраты на лечение не компенсируются увеличением урожая от выздоровления

деревьев. Если у Вас получились другие значения, значит были ошибки при задании ТУВ вершин и их необходимо проверить.

При выявлении новых фактов, мнение садовода относительно того, что делать может измениться. Рассмотрим тот случай, когда садовод имеет информацию о достаточном количестве осадков, выпавших за последнее время, при котором деревья никак не могут страдать от засухи. Введем в ЭС это новое дополнительное свидетельство, представляющее собой факт, что для вершины *Dry* абсолютно достоверным состояние ‘not’:

раскройте в списке вершин ДВ список состояний вершины *Dry*.

Установите абсолютно достоверным состояние ‘not’ для *Dry* (дважды щелкнув мышкой на состоянии ‘not’).

Проведите новое распространение вероятностей для диаграммы влияния (нажмите кнопку «*Prop Sum Normal*» на панели инструментов).

Посмотрите на значения суммарной полезности вершины *Treat* при ее различных состояниях, а именно ‘treat’ и ‘not’.

У Вас должно получиться:

9138.33 – при проведении каких-либо мероприятий;

5918.33 – при их отсутствии.

В этом случае, очевидно, что лучшим выходом из ситуации для садовода будет обработка деревьев. Причина различий этих двух ситуаций состоит в том, что в первом варианте учитывается степень нашего незнания о том, что страдает ли дерево от засухи. А, следовательно, издержки связанные с лечением не будут возмещены, в виду возможного допускаемого засыхания деревьев.

Аналогичным образом могут быть исследованы и любые другие факты и комбинации фактов с различной степенью доверия к каждому из них.

Литература

1. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы: Учебник. – М.: Финансы и статистика, 2006.

2. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – М.: ФИЗМАТЛИТ, 2003.

3. Джексон П. Введение в экспертные системы.: Пер. с англ.: Учеб. пос. – М.: Издательский дом "Вильямс", 2001.

4. Л.В. Путькина Л.В., Пискунова Т.Г. Интеллектуальные информационные системы: учеб. пособие. – СПб: Изд-во СПбГУП, 2008.

5. Матвеев М.Г. Модели и методы искусственного интеллекта. Применение в экономике: Учеб. пособие / М.Г. Матвеев, А.С. Свиридов, Н.А. Алейникова. – М.: Финансы и статистика; ИНФА-М, 2008.

6. Представление и использование знаний: Пер. с англ. / Под ред. Х. Уэно, М. Исидзука. – М.: Мир, 1989.

7. Смолин Д.В. Введение в искусственный интеллект: конспект лекций. – 2-е изд, перераб. – М.: М. ФИЗМАТЛИТ, 2007.

8. Ясницкий Л.Н. Введение в искусственный интеллект: Учеб. пособие для студ. высш. учеб. заведений – М.: Издательский центр "Академия", 2005.

Содержание

Введение ……..3

1. Введение в интеллектуальные информационные системы

1.1. Методологические основы теории искусственного интеллекта ……..5
1.2. Основные понятия и определения теории интеллектуальных информационных систем ……..11
1.3. Классификация интеллектуальных информационных систем….17
1.4. Данные и знания ……...20
1.5. Архитектура систем искусственного интеллекта ……32

2. Экспертные системы

2.1. Определение и классификация экспертных систем ……..40
2.2. Архитектура статических и динамических экспертных систем ……....51
2.3. Этапы разработки экспертных систем ……....57
2.4. Стратегии извлечения знаний ……68
2.5. Выявление знаний от экспертов ……....87
2.6. Обработка экспертных оценок ……..98
2.7. Экспертные системы с неопределенными знаниями ……109
2.8. Логический вывод на основе субъективной вероятности …….113
2.9. Байесовские сети доверия как средства разработки экспертных систем ……119
2.10. Диаграммы влияния ……126
2.11. Сети доверия с условными гауссовскими переменными ……132

3. Прикладные программы, реализующие БСД

3.1. Производители программ ……143
3.2. Основы построения простейшей байесовской сети доверия …145
3.3. Проектирования простейшей ЭС на основе БСД с непрерывными и дискретными событиями ……..158
3.4. Основы проектирования диаграмм влияния и методы работы с ними в системе Hugin ……169

Литература ……179

Редактор Т.Б. Кузнецова,
Компьютерная верстка П.Г. Немашкалов

Подписано в печать 09.12.09

Формат 60x84 1/16 | Усл.печ.л. 10,52 | Уч.-изд.л. 8,98
Бумага офсетная | Тираж 100 экз. | Заказ 243

Отпечатано в ООО «Бюро новостей».