

Министерство образования Российской Федерации
Новосибирский государственный технический университет

А.В. ГАВРИЛОВ

ГИБРИДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Новосибирск
2002

А.В. Гаврилов. Гибридные интеллектуальные системы: Монография – Новосибирск: Изд-во НГТУ, 2002. – 142 с.

В монографии рассматриваются вопросы построения моделей и архитектур гибридных систем искусственного интеллекта (ИИ), совмещающих в себе разные методы представления и обработки знаний, в том числе, логические, эвристические и нейрокибернетические. Предлагаются такие модели и архитектуры. Приводятся примеры прикладных гибридных систем ИИ, разработанных автором. Некоторые результаты, представленные в монографии, публикуются впервые.

Монография предназначена для специалистов в области искусственного интеллекта и всех интересующихся им, для студентов и аспирантов, изучающих дисциплины, связанные с искусственным интеллектом.

Рис. 33, список лит. 332 назв.

Рецензенты: *В.В.Губарев*, академик РАЕН и МАИ,
Д.т.н., проф.
Е.В.Рабинович, д.т.н., проф.



Новосибирский государственный
технический университет,
Гаврилов А.В., 2002 г.,

ВВЕДЕНИЕ

*"С годами мозг мыслителя искусный
Мыслителя искусственно создаст".*

И.В. Гете. Фауст

В последние 20-30 лет исследования в области искусственного интеллекта заняли одно из ведущих мест в информатике и в развитии информационных технологий.

В истории человечества можно выделить несколько информационных революций, которые являлись переломными моментами в развитии цивилизации.

Первая была связана с появлением речи. Речь дала возможность передавать знания от одного индивидуума к другому и, следовательно, сохранять их, передавая из поколения в поколение.

Вторая информационная революция была связана с появлением письменности. Письменность позволила передавать знания между индивидуумами без непосредственного контакта и, следовательно, существенно увеличить доступность и надежность сохранения знаний.

Третья революция – появление книгопечатания – еще более увеличило доступность знаний и сделало возможным массовое распространение и сохранение знаний.

Четвертая революция была связана с появлением электросвязи в различных видах и средств записи знаний (информации) электрическим путем.

Пятая революция, которую мы переживаем сейчас, связана с появлением массовых ЭВМ, объединенных сетями, и обладающих мощными средствами для хранения, накопления и использования знаний. В рамках этой информационной революции можно выделить также ряд этапов, последним из которых является появление систем искусственного интеллекта или систем, основанных на знаниях. В целом пятая революция дала возможность сохранять и иметь быстрый доступ практически к неограниченным по объему знаниям. Кроме того, системы искусственного интеллекта дали возможность впервые связать непосредственно знания с материальным производством (или, в более общем случае, с окружающим материальным миром), исключив человека, как промежуточное звено. Более того, они могут вырабатывать новые знания. Для этого служат средства извлечения (или приобретения) знаний, например из баз данных путем выявления закономерностей и формулирования этих закономерностей в виде баз знаний. В компьютерном мире сейчас получили распространение исследования и разработки в области так называемых технологий "data mining" и "knowledge discovery" – извлечения полезных данных (знаний) из большого их количества, в том числе, из неструктурированных данных.

Современная информатика во многом обязана исследованиям в области искусственного интеллекта. Например, многие разделы исследования операций появились из разработанных в 50-х годах методов ограничения перебора вари-

антов при решении задач ИИ. На терминологию и методы построения компиляторов повлияли исследования в области машинного перевода. Средства распознавания и синтеза речи сейчас уже являются неотъемлемой частью некоторых специализированных информационных систем и претендуют на широкое использование в операционных системах. Системы распознавания текста являются обычной частью офисных программных систем (печатный текст) или карманных компьютеров (рукописный ввод). Объектно-ориентированное программирование выросло из представления знаний в виде фреймов, придуманных американским ученым Мински в конце 60-х годов. Нейросетевые технологии и технологии экспертных систем успешно применяются в системах экономического анализа и прогнозирования. Примеры можно было бы продолжить.

Существует много различных определений области информатики, называемой искусственным интеллектом (ИИ). Ниже приведены некоторые из них [1]:

"[Автоматизация] видов деятельности, которые мы ассоциируем с человеческим мышлением (human thinking), таких как принятие решений, решение проблем, обучение ..." (Belman, 1978);

"Прикладывание новых усилий для того, чтобы сделать думающие компьютеры, ... машины с мозгами в полном и дословном смысле" (Hougeland, 1985);

"Изучение ментальных способностей через использование вычислительных моделей" (Charniak, McDermott, 1985);

"Искусство создания машин, которые осуществляют функции, требующие интеллекта при реализации их человеком" (Kurzweil, 1990);

"Область науки, которая имеет дело с объяснением и воспроизведением интеллектуального поведения в терминах вычислительных процессов" (Schalkoff, 1990);

"Изучение того, как заставить компьютеры делать вещи, которые в настоящее время лучше делают люди" (Rich, Knight, 1991);

"Изучение вычислений, которые делают возможным распознавать, размышлять и действовать" (Winston, 1992);

"Область информатики, имеющая дело с автоматизацией интеллектуального поведения" (Luger, Stubblefield, 1993).

Такое разнообразие определений объясняется тем, что понятие "искусственный интеллект" может рассматриваться в различных контекстах. Оно может рассматриваться как наука, набор технологий, реализованная модель разума, раздел информатики, занимающийся изучением того, как работает мозг. Некоторые ученые склонны рассматривать "искусственный интеллект" как нечто постоянно ускользающее и недоступное (цель, всегда находящаяся за горизонтом). Такая точка зрения объясняется тем, что технологии и алгоритмы, разработанные в рамках ИИ, со временем становятся неотъемлемой частью информационных технологий и более не ассоциируются с ИИ. Если встать на эту точку зрения, то искусственным интеллектом можно назвать "еще не широко ис-

пользуемые или не открытые технологии, которые реализуют или моделируют процессы обработки информации в нервной системе".

В отличие от некоторых других достижений цивилизации, таких как автомобиль, самолет, кухонная бытовая техника, искусственный интеллект не имеет прототипов в народном творчестве – мифах, легендах и сказках. Мифы или мечты о нем возникают только с появлением вычислительной техники и автоматики (в 40-х годах 20-го века), за некоторыми исключениями (одно из них, может быть, единственное) приведено в эпиграфе. Что касается более ранней мифологии, наиболее близким легендарным персонажем является искусственный человек (андроид или гомункулус). Однако он всегда присутствует как бездумный послушный хозяину механизм. Можно вспомнить железного человека, выкованного Гефестом на горе Олимп, деревянного человека, сделанного итальянским мастером в Толедо и известного больше под именем "каменного гостя" А.С. Пушкина, Голема – слугу средневекового алхимика из г. Праги.

Они явились прототипом современного робота. Однако при создании реальных современных роботов возникла проблема общения с ними с целью формулирования им задач. При решении этой проблемы появилось понимание того, что если мы хотим иметь удобные и надежные средства общения с роботом (или в общем случае с некоторой сложной системой), мы должны обеспечить его средствами внутреннего представления и решения задач примерно такими же, как у человека. Иначе мы обречены либо на изучение искусственного языка общения или программирования либо на "непонимание" роботом того, что мы от него хотим. Впервые на проблему понимания интеллектуальной системой того, что от нее хотят, обратил внимание основатель кибернетики Н. Винер в 1948 году в своей книге "Кибернетика" [1], приведя в качестве примера притчу об обезьяньей лапке.

Особенное значение эта проблема взаимопонимания человека и искусственных интеллектуальных систем приобретает в связи с применением искусственного интеллекта в военной области, что является основой так называемой "стратегической компьютерной инициативы", провозглашенной президентом США Р. Рейганом еще в 1983 году.

Таким образом, основной целью научного направления, называемого искусственным интеллектом, является разработка механизмов человеческого мышления в искусственных технических системах.

В настоящее время можно выделить шесть основных бурно развивающихся и наиболее перспективных (по мнению автора) технологий ИИ:

1. Нейронные сети.
2. Мягкие вычисления за исключением нейронных сетей (включая сюда генетические алгоритмы и эволюционное программирование).
3. Интеллектуальные роботы.
4. Интеллектуальные агенты и мультиагентные системы.
5. Интеллектуальный анализ данных.
6. Обработка естественного языка (включая распознавание и синтез речи).

В таблице 0.1 приведена информация об основных научных школах, занимающихся разработкой этих технологий.

Из анализа существующих на рынке программных продуктов, использующих методы ИИ или разработанных для ИИ была предложена классификация прикладных задач, решаемых или способных решаться с использованием ИИ, приведенная на рис. 0.1.

В области искусственного интеллекта в СССР, России и за рубежом значительный вклад сделали следующие исследователи: Аверкин А.Н., Амосов Н.К., Батыршин И.З., Берштейн Л.С., Вагин В.М., Васильев В.И., Воронков Г.С., Гаазе-Рапопорт М.Г., Гаврилова Т.А., Галушкин А.И., Гладкий А.В., Гладун В.П., Горбань А.Н., Дорогов А.Ю., Дунин-Барковский В.Л., Ефимов Е.И., Жданов А.А., Загоруйко Н.Г., Загорулько Ю.А., Кандрашина Е.Ю., Короткий С., Крюков А.Ю., Кузнецов В.Е., Кузнецов И.П., Куссуль Э.М., Ловицкий В.А., Любарский Ю.А., Макаров И.М., Мартынов В.В., Мельчук И.А., Миркес Е.М., Нариньяни Н.С., Оныкий Б.Н., Осипов Г.С., Позин Н.В., Попов Э.В., Поспелов Г.С., Поспелов Д.А., Радченко А.Н., Рыбина Г.В., Рубашкин Б.Ш., Соломатин Н.М., Соколов Е.Н., Тимофеев А.В., Файн В.С., Финн В.К., Трахтенгерц Э.А., Хабаров В.И., Хорошевский В.Ф., Шумский С.А., Щетинин В.Г., Янковская А.Е., Bobrow D.G., Bundy A., Davis R., Feigenbaum E.A., Hayes-Roth B., Hinton J.E., Honavar V., Hopfield J.J., Hunt E.B., Kohen W.W., Kohonen T., Kovalski R.A., Lauriere J.L., Lenat D., McDermott J., McCarthy J., Minsky M., Newell A., Nil N.P., Nilsson N.J., Quillian M.R., Raphael B., Robinson J.A., Rosenblatt F., Rumelhart D.E., Sacerdoti E.D., Samuel A.L., Simon H.A., Schank R.C., Shannon C.E., Viner N., Wasserman F., Waterman D.A., Winograd T., Winston P.H., Yen J., Zadeh L.A. и др.

С самого начала развития вычислительной техники в области моделирования мыслительной деятельности параллельно развивались два подхода:

- логический, основанный на моделировании рассуждений или логического (вербального) мышления;
- нейрокибернетический, основанный на моделировании работы человеческого мозга как множества взаимодействующих нейронов.

Первый из них занимается моделированием поверхностного мышления, начиная с программирования эвристик в 50-х годах на примере решения главным образом игровых и математических задач [3] и кончая построением классических экспертных систем 80-х-90-х годов [4-10]. Этот подход реализуется в рамках технологий инженерии знаний. Но вся история развития этого направления связана с совершенствованием работы систем, основанных на знаниях, путем углубления этих знаний, и моделирования более глубоких мыслительных процессов, чем просто механизмов манипулирования символьной информацией. К этим попыткам можно отнести представление нечетких знаний с помощью лингвистических переменных [11], нечеткие [12-19] и псевдофизические логики [20] и т.п. Кроме того, в этом направлении в 80-90-годах наметилась тенденция объединения разных концепций представления и обработки

Таблица 0.1

Основные научные школы по перспективным технологиям ИИ

Нейронные сети (Neural Networks)	Нечеткие системы (Fuzzy Systems)	Интеллектуальные роботы (Robotics)	Мультиагентные сис- темы (Multi-agent Systems)	Интеллектуальный анализ данных (Intelligent Data Ana- lyzing)	Обработка естествен- ного языка (Natural Language Proc- essing)
University of Toronto Www.cs.toronto.edu/neuron/ Iowa State University http://www.cs.iastate.edu/~honoravar/ailab/ , Institute in Sheffield http://www.shef.ac.uk , Austrian Research Institute for AI www.ai.univie.ac.at , МГТУ им. Баумана http://www.chat.ru/~vlasov/ , ВЦ СО РАН (Красноярск) , Институт нейрокибернети- ки (Ростов-на-Дону), СпбГЭТУ , Институт кибернетики им. Глушкова (Киев), ХТУРЭ , МИФИ, Физический институт им. Лебедева http://canopus.lpi.msk.su/neurolab/ Институт системного про- граммирования http://www.ispras.ru/	Stanford University Http://cs.stanford.edu/Research , Iowa State University http://www.cs.iastate.edu/~honoravar/ailab/ , Carnegi-Mellon Uni- versity http://www.cs.cmu.edu , New University of Lis- boa Http://www.di.fct.unl.pt , Российский институт ИИ (Москва- Новосибирск) www.aha.ru/~artint/ , Институт программ- ных систем (Пере- славль-Залесский) www.botik.ru/PSI/	Carnegi-Mellon Univer- sity http://www.cs.cmu.edu , MIT http://www.ai.mit.edu , University of Melbourne http://www.cs.mu.oz.au , Stanford University Http://cs.stanford.edu/Research , Edinburg University Http://www.informatics.ed.ac.uk , Tokyo University, University of Sussex, Институт им. Глушкова (Киев), Институт FAW-Ulm (Германия)	Iowa State University http://www.cs.iastate.edu/~honoravar/ailab/ , Santa Fe Institute Http://alife.santafe.edu , University of Melbourne http://www.cs.mu.oz.au , Edinburg University Http://www.informatics.ed.ac.uk , Austrian Research Institute for AI www.ai.univie.ac.at , Российский институт ИИ www.aha.ru/~artint/	Iowa State University http://www.cs.iastate.edu/~honoravar/ailab/ , Stanford University Http://cs.stanford.edu/Research , Manchester University http://www.cs.man.ac.uk ,	Massachusets Institute of Technologies (MIT) Http://www.ai.mit.edu , Stanford University Http://cs.stanford.edu/Research , Edinburg University http://www.informatics.ed.ac.uk , New University of Lisboa http://www.di.fct.unl.pt , Harvard University http://www.eecs.harvard.edu/ai , Austrian Research Institute for AI www.ai.univie.ac.at , Университет в Пизе (Италия) Http://www.ilc.pi.cnr.it/ Институт FAW-Ulm (Германия) МГТУ им. Баумана, Российский институт ИИ www.aha.ru/~artint/

Это - попытка классификации всех задач информатизации, в которых применяются или могут применяться технологии ИИ. 0-й раздел - задачи, решаемые не для конечных пользователей, а для разработчиков интеллектуальных систем.

Таблица создана на основе анализа существующих на рынке программных продуктов.

- 0. Инструментарий
 - 0.1. Языки программирования для ИИ
 - 0.2. Средства для создания экспертных систем
 - 0.3. Средства для создания хранилищ данных и знаний
 - 0.4. Средства для создания нейронных сетей и включения их в приложения
 - 0.5. Средства для экспериментов
 - 0.5.1. С нейронными сетями
 - 0.5.2. С роботами

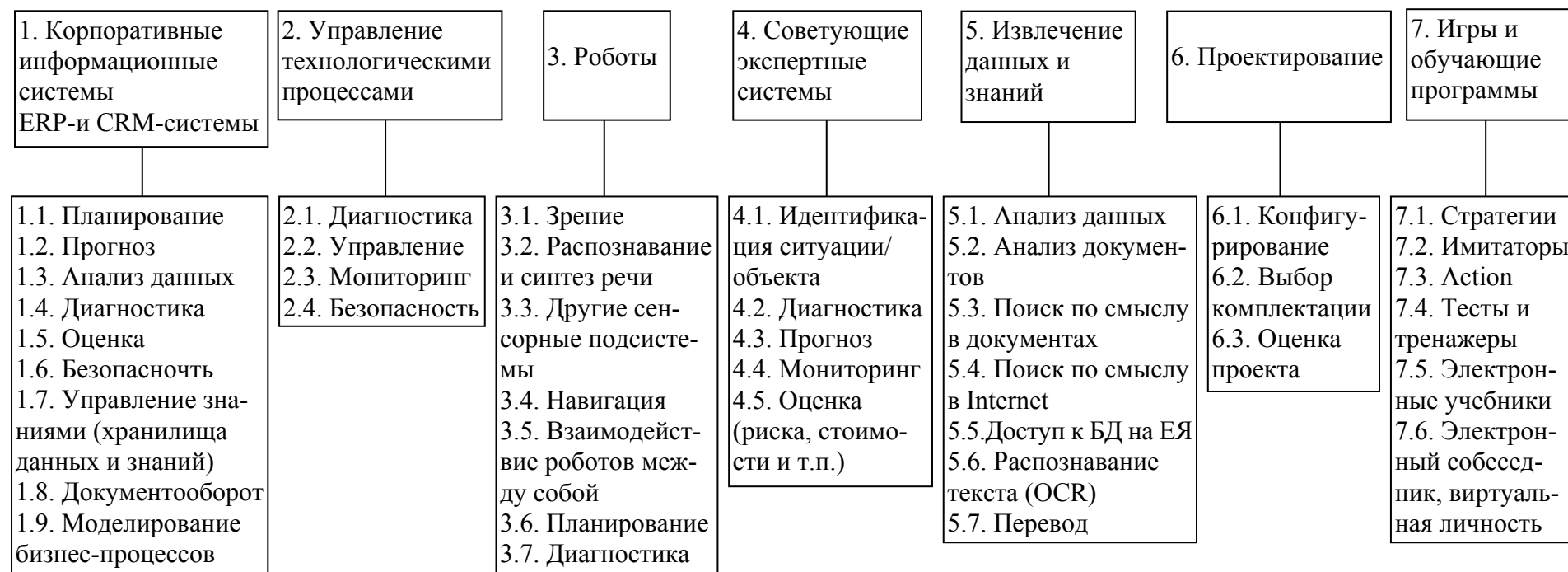


Рис. 0.1. Классификация задач, решаемых с использованием ИИ

знаний в одной системе ИИ [21-28], в отличие от ранних систем ИИ (например, ранних экспертных систем). Такие системы ИИ получили название гибридных интеллектуальных систем. Причем в последнее десятилетие появились новые парадигмы, такие как онтология [29-30], интеллектуальный агент [31-32], призванные дать возможность с одной стороны более структурировать базы знаний, с другой стороны, предоставить большее разнообразие точек зрения на сущности, участвующие в представлении и обработке знаний и, соответственно, большее разнообразие методов и алгоритмов обработки. Этот путь совершенствования моделирования мыслительной деятельности можно назвать моделированием "сверху вниз".

Достоинством этого направления является относительная легкость понимания и объяснения того, что происходит в интеллектуальной системе при решении задачи, т.к. объяснение основано на вербализации, естественно реализуемой в таких системах. Недостатками этого направления являются:

- поверхностность рассуждений, т.е. отсутствие в них творческой составляющей или инсайта;
- последовательный характер рассуждений, что затрудняет распараллеливание с целью ускорения решения задач;
- трудность и большая трудоемкость формализации знаний или обучения системы;
- трудность реализации таких перспективных моделей рассуждений как немонотонные рассуждения, поддержка множественности линий рассуждений, нечеткие рассуждения, ассоциативный поиск и т.п.

Нейрокибернетическое же направление [33-58] с самого начала занималось моделированием самых глубинных процессов, протекающих в мозгу человека или животного. Можно сказать, что в отличие от первого направления, моделирующего осознанное (на вербальном уровне) мышление, это направление занимается моделированием подсознания, на котором основана в основном творческая деятельность. Недостатками этого подхода являются:

- отсутствие вербализации в процессе решения задачи нейронной сетью и, следовательно, трудность понимания того, как задача решена и достаточно ли можно доверять полученному решению;
- трудность реализации диалога в процессе решения задач;
- отсутствие возможности работы с абстрактными понятиями и иерархическими структурами.

В направлении ликвидации этих недостатков в настоящее время ведутся исследования, в частности, по созданию ансамблевых нейронных сетей [59], в которых ансамбль нейронов ассоциируется с абстрактным понятием, которое поддается вербализации, или модульных нейронных сетей [60]. Это направление можно назвать моделированием мыслительной деятельности "снизу вверх".

В настоящее время (последние полтора десятилетия) наметилась тенденция комбинирования в одной гибридной интеллектуальной системе логического (в различных вариантах) и нейрокибернетического подходов [61-74]. Таким обра-

зом, исследователи пытаются компенсировать недостатки одного и другого подходов. В конечном итоге, такой путь может привести к созданию действительно "искусственного разума", рассуждающего так же как человек, и необходимого для того, чтобы обеспечивать такое же взаимопонимание между человеком и машиной, как и между людьми (при постановке и обсуждении задач, для решения которых создается прикладная система ИИ). В частности, автором в 1989 году [61] была предложена и далее развивается в [62-64, 71-72, 74] парадигма "двухполушарных экспертных систем", в основе которой лежит использование взаимодействующих между собой через доску объявлений (black board) классической экспертной системы и нейронной сети.

Таким образом, можно выделить две интерпретации понятия гибридной интеллектуальной системы – узкая, когда в ней объединяются разные парадигмы представления и обработки знаний и данных, но она остается в рамках инженерии знаний, и широкая – когда в ней объединяются модели инженерии знаний и нейроинформатики.

В настоящей работе представлены результаты исследований и разработок автора в направлении развития теории и практики гибридных интеллектуальных систем как в узком так и в широком смысле этого понятия.

1. ЭЛЕМЕНТЫ ТЕОРИИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

*"Для того, чтобы познать какую-либо вещь,
надо определить ее границы, выйти за их пределы,
и только тогда станет ясна истинная ее суть".*

Ф. Герберт. Капитул Дюны

1.1. Постановка задачи

В настоящее время существует много различных концепций, принципов и теорий, объясняющих те или иные аспекты процесса мышления и поведения человека. Они появились в рамках различных парадигм искусственного интеллекта и других разделов информатики, а также, других наук и научных дисциплин, таких как философия, психология, нейрофизиология, педагогика, лингвистика.

Эти концепции и теории можно разделить на следующие группы:

1. Модели для представления знаний;
2. Модели решения задач;
3. Модели приобретения знаний (обучения);
4. Модели общения и понимания естественного языка;
5. Модели зрительного восприятия;
6. Модели планирования поведения;
7. Глобальные модели мозга и разума.

Модели представления знаний и модели решения задач развиваются в рамках инженерии знаний и традиционно тесно связаны с появлением и развитием экспертных систем. Однако, автор считает, что в обученной нейронной сети также содержатся знания, и поэтому соответствующие модели представление знаний добавлены в перечень. Модели представления знаний традиционно делятся на логические и эвристические. К логическим относятся:

- логика высказываний (обычно не считается аппаратом инженерии знаний) [1];
- логика предикатов 1-го порядка [1];
- логика Хорна в языке Prolog и подобных ему системах [2];
- логика предикатов высших порядков;
- трехзначная логика;
- логика возможных миров [3];
- модальные логики [1, 3];
- теория нечетких множеств [4-6];
- нечеткая логика [4-6];
- лингвистические переменные [7].

К эвристическим моделям представления знаний можно отнести:

- деревья решений [8, 9];
- семантические сети [1, 10-12];

- фреймы и сети фреймов [11, 13];
- онтологии [14-16];
- объектно-ориентированное программирование [17];
- реляционная алгебра (обычно не рассматривается в рамках инженерии знаний) [18, 19];
- правила-продукции (эта модель является промежуточной между логическими и эвристическими и может в зависимости от особенностей интерпретации правил рассматриваться в рамках одной или другой группы) [11, 20-22];
- матрица весов связей обученной нейронной сети (коннекционистский подход) [23, 24];
- матрица весов связей и другие изменяемые параметры обученной нейронной сети, в том числе, в спайковых [25] или осцилляторных нейронных сетях [26].

Модели решения задач также можно разделить на логические и эвристические. К логическим можно отнести:

- метод резолюции [1];
- прямой, обратный и смешанный детерминированный дедуктивный логический вывод [20, 27];
- прямой, обратный и смешанный нечеткий дедуктивный логический вывод [20, 27];
- унификация в языке Prolog и подобных ему системах [1, 2, 3];
- немонотонный логический вывод [1];
- метод аргументации [28].

К эвристическим можно отнести следующие модели решения задач:

- табличный метод;
- различные методы оптимизации, рассматриваемые в рамках исследования операций (линейного, нелинейного и целочисленного программирования) [29];
- метод Монте-Карло и его разновидности [29];
- поиск решения в однородном пространстве состояний с использованием различных эвристических стратегий перебора вариантов [30];
- разбиение задач на подзадачи и решение последовательности подзадач [20, 31];
- разбиение пространства состояний на подпространства с разными уровнями абстракции и решение подзадач на этих уровнях абстракции [20];
- рассуждения по аналогии [32];
- вероятностный вывод [33, 34];
- сопоставление с образцом (четкое, нечеткое);
- генетические алгоритмы генерации и отбора решений [35];
- нейрноподобные модели решения задач, основанные на взвешенном суммировании факторов [36-48].

К моделям приобретения знаний или обучения можно отнести следующие:

- индуктивный логический вывод в рамках логических моделей представления знаний [27];
- индуктивный вывод (обобщение) в рамках эвристических моделей представления знаний (к этому классу относится ДСМ-метод [49], растущие семантические сети [12]);
- обучение на одном примере путем запоминания ассоциаций (к этому классу относятся большинство моделей, реализованных в диалоговых системах, обучающихся поиску текстов по содержанию, например, [50, см. главу 4]);
- обучение на одном примере путем анализа примера;
- модель множественного учета аргументов (МГУА) [51];
- обучение на множестве примеров путем изменения весов связей нейронной сети (коннекционистский подход) [41-48];
- обучение на множестве примеров путем изменения весов связей и порогов нейронов нейронной сети [36-40];
- обучение на множестве примеров путем создания новых нейронов и связей в нейронной сети (конструктивные методы обучения) [52-54].

К моделям общения и понимания естественного языка (ЕЯ) относятся следующие модели:

- представление смысла в виде шаблонов как в программе Alice и ей подобных системах [30];
- грамматики Холмского (почти не рассматриваются в рамках искусственного интеллекта);
- синтаксически-ориентированные модели [55] анализа ЕЯ, в том числе, семантические грамматики для построения специализированных диалоговых систем со сравнительно узкой и статической предметной областью;
- модель "смысл-текст" [56];
- модель концептуальных зависимостей [57];
- сети расширенных переходов [58];
- модели активного диалога [59-60];
- модель анализа смысла как задачи распознавания [61];
- модели синтеза предложений на естественном языке;
- модели распознавания и синтеза речи [62].

К моделям зрительного восприятия относятся следующие [63-64]:

- модели борьбы с помехами на изображении, в том числе, увеличения контрастности изображения;
- модели распознавания особенностей и элементов изображений (линий, углов, точек, эллипсов);
- модели распознавания объектов (на основе сопоставления с образцом, основанных на знаниях о структуре объектов, на основе выявления особенностей);
- модели распознавания текстуры изображений;
- модели описания и распознавания сцен;

- модели для распознавания размеров объектов и расстояний.
- К моделям планирования поведения можно отнести следующие:
- модели условного рефлекса [65];
 - модели адаптивного поведения интеллектуальных систем (к этому классу относятся и модели управления вниманием, в частности, принцип доминанты) [66-68];
 - модели искусственной жизни [68];
 - модели планирования оптимального маршрута движения [69];
 - модели планирования маршрута движения в неопределенной или изменяющейся среде [70];
 - модели управления роботом-манипулятором (к этому классу относятся методы планирования движения манипулятора, в частности, с учетом динамических нагрузок и других помех [71];
 - модели поведения мобильных интеллектуальных роботов [72].
- К глобальным моделям разума можно отнести следующие:
- модель отражения материального мира марксизма-ленинизма;
 - модель первичности и объективности сознания (Гегеля и других школ идеализма и мистицизма);
 - гомеостатическая модель Эшби [73];
 - модель целеустремленных систем [74];
 - теория функциональных систем П.К. Анохина [75];
 - голографическая метафора К. Прибрама [76];
 - квантовая психология Р.А. Уилсона [77].

Большинство из этих моделей либо носит неконструктивный характер (особенно, последняя группа глобальных моделей), либо слишком специализировано на решение узкого класса задач при существенной ориентации на использование определенных технологических методов и средств, что затрудняет (или даже делает невозможным) комбинировать их в одной интеллектуальной системе.

В данной главе сделана попытка сформулировать концепции и модели, связанные с описанием информационных процессов, происходящих в мозгу, или подобных им, лишенные этих недостатков, и удовлетворяющие следующим требованиям:

- они должны быть достаточно конструктивными, чтобы быть основой для создания действующих программных или программно-аппаратных моделей;
- должны описывать как можно большее количество сторон функционирования интеллектуальных систем, т.е. включать в себя концепции и модели из всех перечисленных выше классов, но достаточно гармонично связанных между собой;
- должны быть открыты для дальнейшего развития – уточнения и добавления новых моделей на основе уже имеющихся.

Наличие таких концепций и моделей может служить основой для создания теории интеллектуальных систем (естественных и искусственных) и создания на ее основе действительно полномасштабного искусственного разума, называемого в зарубежных публикациях "Strong Artificial Intelligence". Необходимость его создания диктуется потребностью в Искусственном Интеллекте (реализованном в различных приложениях) как в полноценном партнере, понимающем человека на уровне человеческого понимания.

1.2. Модель интеллектуальной системы [78]

*"... в людях глубоко заложено стремление
все разложить по полочкам, все классифицировать,
всему присвоить ярлык".*

Ф. Герберт. Капитул Дюны

Любой метод представления знаний в естественной или искусственной интеллектуальной системе можно представить как

$$(K, D, F(K,D)), \quad (1)$$

где: K – множество денотатов – образов внешнего мира, представленных в виде состояний разного типа рецепторов (датчиков или сенсоров);

D – множество десигнатов – знаков;

$F(K, D)$ – функция отображения K на D .

При этом множество образов K можно разделить на статические и динамические и на гомогенные и гетерогенные образы. Строго говоря, статических образов вообще не существует в реальном мире. Однако, для упрощения представления и обработки знаний полезно рассматривать отдельно образы, которые не зависят существенно от времени и считать их статическими.

Динамические образы можно представить в виде множества состояний рецепторов $K = (k_1, k_2, \dots, k_i(t,K), k_{i+1}(t,K), \dots)$, где $k_{i+1}(t,K)$ – параметр, зависящий от времени, т.е. функция, определенная на множестве K . Гомогенные образы состоят из состояний сенсоров (сигналов) одного типа. Например, зрительные или слуховые образы. В отличие от них, гетерогенные представляют собой комбинации состояний разного типа сенсоров, т.е. их можно рассматривать как комбинации гомогенных образов.

В (1) можно отображение $F(K,D)$ разделить на два отображения: $F_R(K,D)$ – отображение образов на десигнаты (знаки) и $F_A(D,D)$ – отображение знаков на знаки. Последнее отождествляется с обычно рассматриваемыми в рамках инженерии знаний методами представлением знаний – семантическими сетями, сетями фреймов, правилами-продукциями, иерархическими структурами и т.п., с которыми имеют дело всевозможные методы логического вывода, в том числе, и нечеткого вывода. Отображение $F_R(K,D)$ можно отождествить с преобразованием образов в знаки (кодированием или распознаванием образов), производимым, в частности, в искусственных нейронных сетях.

Несколько менее очевиден тот факт, что можно выделить еще одну составляющую отношения $F(K,D)$, отношение, определяемое на множестве нечетких образов внешнего мира, $F_F(K,K)$ (F от слова "fuzzy"). Это отношение формируется в процессе обработки поступающей из внешнего мира через рецепторы информации и является описанием получающейся в результате этого ассоциативной памяти, основы для образного (ассоциативного) мышления.

Если мы имеем дело с интеллектуальной системой, проявляющей активность по отношению к внешней среде (живое существо, интеллектуальный робот), т.е. с целеустремленной (активной) системой [74], то в формулу (1) можно добавить соответствующие компоненты, и она примет вид

$$(K, D, A, F(K,D), f(K,A)), \quad (2)$$

где: A – множество действий, которые может производить интеллектуальная система (механические действия, звуки, вывод сообщений и т.п.);

$f(K,A)$ – отображение множества образов на множество действий (в простейшем случае это – взаимно-однозначное соответствие, имеющее место в естественных интеллектуальных системах в виде условных и безусловных рефлексов) (рис. 1.1). Осознанные действия (управляемые вербальным мышлением) здесь явно не отражены, т.к. связь между действиями и знаками существует опосредованная через образы (через подсознательное мышление).

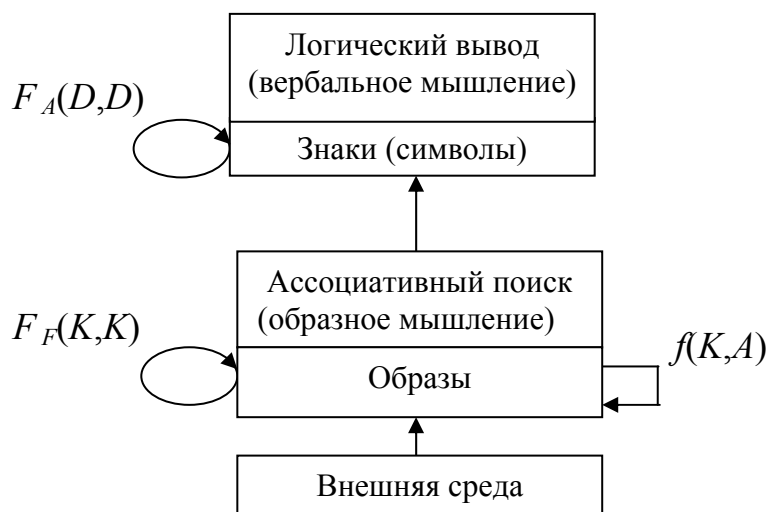


Рис. 1.1. Модель интеллектуальной системы

С другой стороны мозг можно представить (рис. 1.2) в виде ядра, ответственного за подсознательное мышление, и множества сенсорных анализаторов (зрительного, слухового, тактильного и т.п.), процесс мышления в которых осознан, т.к. непосредственно связан с активностью рецепторов. В определенном смысле можно сказать, что человек осознает себя и свои мысли через взаимодействие с окружающим миром.

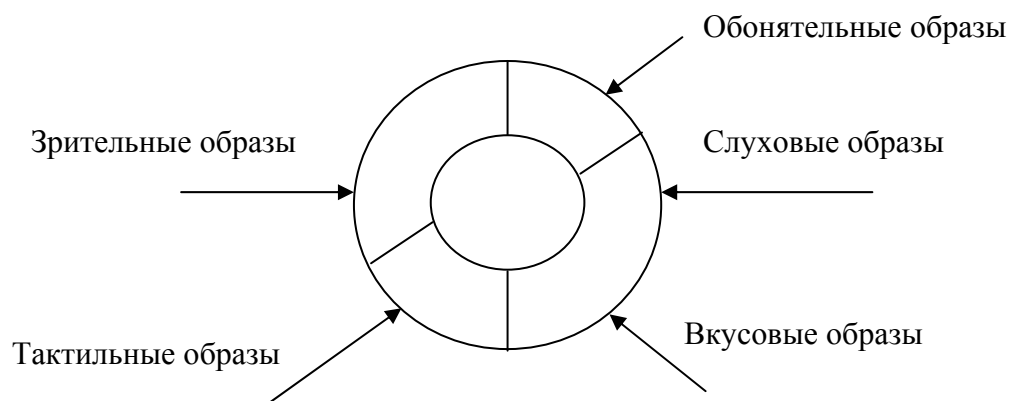


Рис. 1.2. Сознание и подсознание

Отображения $F_A(D,D)$ и $F_R(K,D)$ – основа для вербального мышления, относятся к области сознания, причем к небольшой ее части, связанной со слуховым анализатором. Это связано с тем, что в основе вербального мышления лежит последовательное проговаривание слов (знаков), развитием которого является последовательное логическое мышление. Все остальные области мозга занимаются образным мышлением (осознанным или подсознательным). В случае освоения человеком техники скорочтения проговаривание исключается из процесса восприятия текста и в этом случае при чтении имеет место параллельная обработка зрительных образов, распознавание которых может инициировать как образное, так и вербальное мышление. Известно, что при медитации [81] человек отключает сознание и органы чувств, т.е. процесс мышления сохраняется только в ядре. Это позволяет мозгу разбудить процесс поиска ассоциаций, свободный от ограничений, связанных с логическим мышлением и с ассоциациями, хранящимися в сенсорных анализаторах, описывающих конкретные образы (ситуации). В этом случае при возврате на сознательный уровень мышления возникает эффект "инсайта", т.е. знание в мозгу появляется как бы "само по себе". Подобные процессы могут иметь место в искусственных интеллектуальных системах, построенных в "двухполушарной" парадигме [80-87]. Пока же при реализации нейронных сетей мы имеем дело с моделированием подсознания (разработчики практически не продвинулись дальше реализации условных рефлексов и ассоциативной обработки данных на уровне зрительных и других анализаторов). При реализации систем, основанных на знаниях, моделируется сознание на уровне логического мышления (с элементами нечеткости как попыткой свести параллельные процессы ассоциативного поиска к последовательному логическому выводу).

1.3. Модель ассоциативного мышления [88, 89]

"Память никогда не восстанавливает реальность.

Память реконструирует.

Реконструкция же изменяет оригинал, становясь внешней формой, содержание которой неизбежно страдает изъятиями".

Ф. Герберт. Еретики Дюны

1.3.1. Постановка задачи

Главной задачей обработки знаний, решаемой в любой интеллектуальной системе, является задача поиска релевантных знаний или, другими словами, восстановление фрагмента знания (образа) по его неполному или зашумленному образцу.

В пролого-подобных системах поиск релевантного знания реализован в виде унификации. Во фреймовых системах задача поиска реализуется в виде сопоставления (matching) фреймов. В семантических сетях поиск осуществляется путем сопоставления фрагментов сети и графа-запроса.

Все эти методы поиска можно свести к одному, который и является, по-видимому, реализованным в естественных интеллектуальных системах, – ассоциативному нечеткому поиску знания по его фрагменту. Под знанием здесь подразумевается ассоциативная связь между образами

1.3.2. Ассоциации и ассоциативный поиск

Пусть задано:

- множество признаков $K = \{p_i\} \mid i = 1, N^p$, описывающих состояние внешнего мира и самой интеллектуальной системы в некоторый момент времени t , где N^p – количество признаков;

- множество комбинаций значений признаков на множестве K $P^0 = \{P_j\} \mid P_j = \{p_{ij}\} \mid j = 1, N^o, i = 1, N^p$, описывающих конкретные образы, с которыми сталкивается при своем функционировании интеллектуальная система, где N^o – количество образов;

- множество реальных (не с полным набором признаков) образов $P = \{P_{kj}\} \mid j = 1, N^o$ и k пробегает значения из подмножества натурального ряда чисел в интервале $(1, N^p)$;

- запрос (образ, инициирующий ассоциативный поиск) $P \in P$;

- образ-результат ассоциативного поиска $R \in P$.

Выделение множества "реальных" образов P в отличие от множества "идеальных" образов P^0 имеет принципиальное значение. Естественная интеллектуальная система практически всегда имеет дело с неполными данными о внешнем и внутреннем (своем) мире. Основываясь на них, она вспоминает (додумывает, довоображает) недостающие данные, а также, вспоминает связанные с ними ассоциативно другие образы (предшествующие или следующие во време-

ни). Этот процесс, основанный на ассоциативном воспоминании, и лежит в основе процесса мышления. При этом надо различать два разных процесса:

- процесс восстановления образа по частично заданным признакам (обычно именно этот процесс имитируется в различных моделях ассоциативной памяти, и он может иметь смысл восстановления искаженного помехами зрительного или слухового образа, воспоминания шума, услышанного одновременно с определенным зрительным образом и т.п.);

- процесс поиска связанных ассоциативно с данным образом других образов, привязанных к другим значениям времени (эти образы могут иметь смысл причины или следствия данного образа).

Если первый вариант реализуется в естественных интеллектуальных системах в рамках сенсорных отделов головного мозга, то второй – в коре головного мозга и является основой прогностической деятельности и всей мыслительной деятельности живого существа.

Далее будет рассматриваться модель, ориентированная на описание второго варианта, представляющего на взгляд автора наибольший интерес. До недавнего времени моделированию процесса ассоциативного мышления во времени не уделялось должного внимания, хотя в последнее время начали появляться модели нейронных сетей, работающих с динамическими образами [25, 26]. Из более ранних работ можно упомянуть [37].

В общем случае признаки из множества K являются числами, определенными на метрических шкалах. В частном случае признаки являются бинарными. Признаки могут быть внешними и внутренними. Внешние признаки связаны с восприятием объектов внешнего мира с помощью рецепторов или устройств, их заменяющих в искусственных интеллектуальных системах. Внутренние признаки являются состояниями элементов системы представления знаний (переменных, слотов, нейронов). Соответственно, образы, описываемые признаками, можно разделять на внешние и внутренние.

Определение 1. Назовем отношение между двумя образами P и R , обладающее свойствами симметричности, рефлексивности и транзитивности, ассоциацией $A(P,R)$ или (P,R) , предикат $\Xi(P^a, R^a, T^a)$, описывающий процесс восстановления $R^a \mid R^a \subseteq R$ по $P^a \mid P^a \subseteq P$, – ассоциативным поиском, P^a – начальным образом ассоциативного поиска и R^a – конечным образом ассоциативного поиска, T^a – временем ассоциативного поиска. Причем, для всех P^a и R^a справедливо, что $\Xi(P^a, R^a, T^a) = true$, если $P^a = P$ и $R^a = R$, $\Xi(P^a, R^a, T^a) = false$, если $P^a = \emptyset$. Если рассматривается интеллектуальная система, функционирующая в дискретном времени, то $T^a \in (I, N)$.

Определение 2. Множество ассоциаций $A = \{A_i(P_i, R_i)\} \mid i \in (I, M)$ образуют память или базу знаний интеллектуальной системы.

Определение 3. Ассоциативный поиск $\Xi(P^a, R^a, T^a)$ такой, что он использует только одну ассоциацию, содержащуюся в памяти интеллектуальной системы, $A = (P, R) \mid P^a \subseteq P, R^a \subseteq R$, называется элементарным ассоциативным поиском. Для случая интеллектуальной системы с дискретным временем элементар-

ному ассоциативному поиску соответствует время поиска $T^a=1$ (один шаг). В этом случае будем опускать время поиска и обозначать элементарный ассоциативный поиск как $\Xi(P^a, R^a)$.

Определение 4. Ассоциативный поиск, принимающий значение *true*, будем называть успешным, а принимающий значение *false*, – неудачным.

Каждой ассоциации $A = (P, R)$ соответствует в общем случае множество элементарных ассоциативных поисков, мощность которого равна мощности декартова произведения $P^s \times R^s$, где P^s – множество всех подмножеств множества P и R^s – множество всех подмножеств множества R .

Каждой ассоциации $A = (P, R)$ в общем случае соответствует множество успешных ассоциативных поисков $\Omega = \{\Xi_i(P_i^a, R_i^a, T^a)\}$, где $P_i^a \subseteq P$ и $R_i^a \subseteq R$. Формирование этого множества зависит от особенностей конкретной реализации ассоциативной памяти (а именно, сопоставления начального образа ассоциативного поиска P_i^a с образом P). Предикат-отношение, описывающий сопоставление двух образов (имеющий смысл примерного равенства), будем обозначать знаком \approx .

Определение 5. Ассоциация, которой соответствует множество успешных ассоциативных поисков Ω , мощность которого равна 1, называется детерминированной ассоциацией. Если мощность множества Ω больше 1, ассоциация называется нечеткой.

Определение 6. Назовем ассоциацию и ассоциативный поиск связанными, если на образы накладывается следующее ограничение:

$$P \subseteq R$$

Определение 7. Назовем ассоциацию и ассоциативный поиск несвязанными (или свободными), если никаких ограничений на множества P и R не накладывается.

Определение 8. Множество признаков K^0 называется ядром ассоциации, если

$$K^0 \subseteq P, K^0 \subset R$$

Отсюда

$$K^0 = P \cap R,$$

т.е. K^0 является пересечением множеств P и R .

Примером ассоциативного связанного поиска является функционирование нейронной сети Хопфилда [23] с "замораживанием" разрядов входного вектора, а примером свободного ассоциативного поиска – функционирование сети Хопфилда без "замораживания". Функционирование любой другой модели нейронных сетей, где входной и выходной векторы "разнесены в пространстве" (базируются на разных нейронах) и имеют разную интерпретацию, можно рассматривать как свободный ассоциативный поиск.

Множество из N ассоциаций $(P_i, R) / i \in (1, N)$, в котором в качестве множества признаков R выбирается один и тот же внутренний образ, является элемен-

тарным деревом и элементом иерархических структур, используемых для классификации образов.

Процесс мышления в определенных выше терминах ассоциаций и ассоциативного поиска можно представить как ассоциативный поиск (процесс вспоминания) образов, инициируемый начальным входным образом, состоящим из внешних признаков. В общем случае его можно представить как ассоциативный поиск, раскладываемый в цепочку или дерево элементарных ассоциативных поисков. Дерево получается, если на очередном шаге есть альтернативные для применения ассоциации. Возможно два вида цепочек:

1) цепочка с забыванием, когда все образы, восстановленные в результате предыдущих ассоциативных поисков, не учитываются в процессе выполнения текущего ассоциативного поиска;

2) цепочка с запоминанием, когда начальным образом для текущего ассоциативного поиска является образ, являющийся композицией (в простейшем случае, объединением) конечных образов, полученных на предыдущих шагах ассоциативного поиска, причем можно ввести параметр m – глубину запоминания, ограничивающий учет истории ассоциативного поиска (при $m = 0$ цепочка с запоминанием превращается в цепочку с забыванием).

При цепочке с запоминанием величина m ассоциируется с понятием емкости кратковременной памяти, используемым в психологии и нейрофизиологии. Считается, что объем кратковременной памяти составляет 5-7 образов.

Цепочку с забыванием можно представить в следующем виде:

$\Xi(P_1^a, P_n^a, T^a) = \Xi(P_1^a, P_2^a, T_1^a), \Xi(P_2^a, P_3^a, T_2^a), \dots, \Xi(P_i^a, P_{i+1}^a, T_i^a), \dots, \Xi(P_{n-1}^a, P_n^a, T_n^a)$,
где:

$$T^a = \sum_{i=1}^{i=n} T_i^a, \quad \Xi(P_i^a, P_{i+1}^a, T_i^a) = True, \forall i | i = \overline{1, n-1}$$

Графически процесс мышления в случае цепочки ассоциативных поисков с забыванием изображен на рис. 1.3. Здесь между начальным и конечным образами каждого элементарного ассоциативного поиска (жирные линии) показаны образы P и R ассоциации, использованной в нем (тонкие линии).

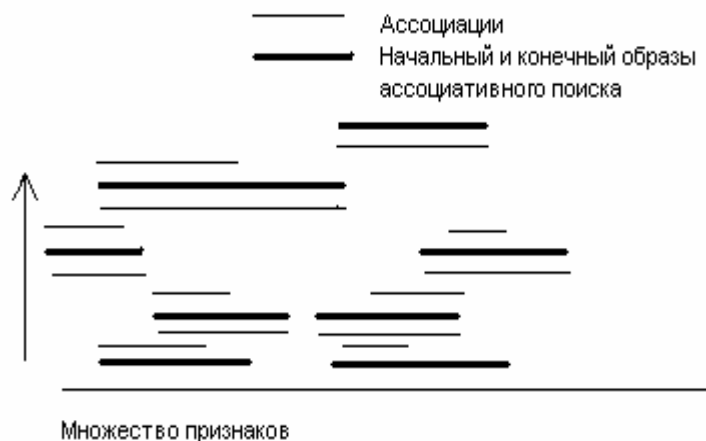


Рис. 1.3. Процесс мышления – цепочка элементарных ассоциативных поисков с забыванием

На рис. 1.4 изображен процесс ассоциативного поиска в случае цепочки с запоминанием с глубиной запоминания $m = 1$. Параметр m может интерпретироваться, по всей видимости, как аналог объема кратковременной памяти в естественной интеллектуальной системе.

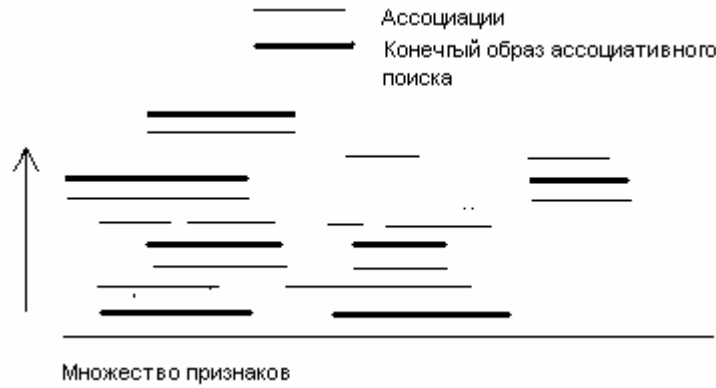


Рис. 1.4. Процесс мышления – цепочка элементарных ассоциативных поисков с запоминанием ($m = 1$)

Цепочку с запоминанием можно представить как

$$\Xi(P_1^a, P_n^a, T^a) = \Xi(P_1^a, P_2^a, T_1^a), \Xi(P_2^{a+}, P_3^a, T_2^a), \dots, \Xi(P_i^{a+}, P_{i+1}^a, T_i^a), \dots, \Xi(P_{n-1}^{a+}, P_n^a, T_n^a),$$

где:

$$T^a = \sum_{i=1}^{i=n} T_i^a;$$

$$\Xi(P_i^a, P_{i+1}^a, T_i^a) = True, \forall i | i = \overline{1, n-1};$$

$$P_i^{a+} = Comp(P_i^a, P_{i-1}^a, \dots, P_{i-m}^a);$$

$Comp$ – композиция образов;

m – глубина запоминания.

Под композицией в простейшем случае можно понимать объединение образов. В этом случае одинаковые признаки образов с разными значениями сосуществуют вместе. Однако более реальным и перспективным представляется вариант, когда разные значения образов одного и того же признака взаимодействуют между собой. Для упрощения формализации и дальнейшей реализации модели предлагается следующий вариант композиции образов с $(i-m)$ -го по i -й:

$$P_i^{a+} = \{p_{ik}\},$$

где:

$$p_{ik} = \frac{\sum_{j=i-m}^i p_{jk}}{m_k};$$

m_k – количество образов, в которых присутствует признак p_{ik} ;

$p_{jk} = 0$, если k -й признак отсутствует в j -ом образе.

Это по существу означает вычисление среднего арифметического среди присутствующих в образах значений k -го признака. При этом с учетом дискре-

тизации значений признаков необходимо округлять среднее арифметическое до ближайшего дискретного значения. В случае бинарного признака (со значениями из диапазона $\{0,1\}$) его значением при композиции образов необходимо считать последнее i -е значение.

Возможные варианты развития ассоциативного поиска в случае наличия альтернативных ассоциаций или множества альтернативных успешных ассоциативных поисков, соответствующих ассоциациям, можно представить в виде дерева, подобного дереву решений в классической теории искусственного интеллекта. Однако, это статическая картина, которая может применяться только для удобства понимания сторонним наблюдателем. В действительности, поступление внешних образов, инициирующих процесс ассоциативного поиска, происходит в произвольные моменты времени. Поэтому следует говорить о параллельных или квазипараллельных процессах ассоциативного поиска, рост количества которых ограничивается объемом кратковременной памяти (параметр m).

В естественных интеллектуальных системах процесс ассоциативного поиска реализован в виде цепочки применения ассоциаций с запоминанием, которая эквивалентна просмотру дерева, подобно тому, как существует последовательный алгоритм, эквивалентный итерационному. При моделировании этого процесса возможны три модели просмотра этого дерева:

- 1) детерминированная;
- 2) вероятностная;
- 3) параллельная.

Детерминированная модель эквивалентна просмотру дерева решений в классических (логических) системах искусственного интеллекта. Эта модель предполагает оценку правильности полученного решения (в данном случае – образа) и возможность возврата к альтернативным ассоциативным поискам, отвергнутым ранее. Очевидно, эта модель далека от того, что происходит в естественных интеллектуальных системах и плохо описывает реальный процесс мышления. Однако, она может применяться при построении искусственных интеллектуальных систем. При этом надо иметь в виду, что возникает проблема выбора стратегии упорядочивания альтернатив.

Вероятностная модель предполагает выбор альтернативной ассоциации с некоторой вероятностью, которая пропорциональна прочности ассоциации (ее подкрепления в процессе обучения) (см. 1.3.3).

Параллельная модель предполагает развитие параллельного процесса применения всех возможных ассоциаций, возможно, с уровнем прочности запоминания, превышающей некоторый порог. Вероятно, это модель наиболее близка к тому, что происходит в реальной естественной интеллектуальной системе. Кроме того, параллельность подразумевает поступление новых внешних образов, инициирующих новые процессы ассоциативного поиска, в произвольные моменты времени.

1.3.3. Формирование ассоциаций

В естественных интеллектуальных системах (в отличие от большинства моделей ИИ) процесс обучения не отделим от процесса рассуждений или ассоциативного поиска. Попытаемся в терминах, определенных выше, описать этот процесс.

Ассоциации в памяти возникают постоянно в процессе восприятия все новых образов внешнего мира. Но запоминаются не все из них, а только те, которые подкрепляются в процессе вызванного ими ассоциативного поиска. Можно сказать, что процесс запоминания управляется запомненной ранее информацией. В процессе ассоциативного вспоминания (описанного выше в виде цепочек элементарных ассоциативных поисков, которые могут протекать в виде дерева параллельных процессов) при срабатывании очередной ассоциации (P, R) она становится активной на время mT^a (в случае $T^a = 1$ эта величина может быть заменена на m – объем кратковременной памяти). Подкрепление ассоциации $A = (P_1, R_1)$ при срабатывании очередной ассоциации $B = (P_2, R_2)$ в процессе ассоциативного поиска происходит в случае, если удовлетворяются следующие условия:

- 1) ассоциация A активна;
- 2) $P_2 \approx R_1$.

Что скрывается под термином "подкрепление ассоциации", зависит от конкретной реализации ассоциативной памяти (так же как и отношение примерного равенства образов).

Если при поступлении некоторого образа из внешнего мира инициированный им ассоциативный поиск не приводит к успеху, образ запоминается в виде конечного образа R_1 ассоциации $A = (P_1, R_1)$, начальным образом P_1 которой является предыдущий образ. Эта ассоциация становится активной на время mT^a . Если за это время она не подкрепляется, то стирается из памяти. Так происходит при первичном формировании ассоциативной памяти. При этом на начальном этапе, когда еще память содержит немного ассоциаций, величина кратковременной памяти m может иметь большое значение и уменьшается по мере формирования ассоциативной памяти.

1.3.4. Нечеткое подобие

Понятие "ассоциация" тесно связано с понятием "подобие". Можно сказать, что если два понятия подобны, то между ними существует ассоциативная связь.

В инженерии знаний давно используются рассуждения по аналогии, основанные на применении отношений подобия (анalogии). Однако отношение подобия в инженерии знаний носит "четкий" характер, в то время как для определения ассоциативного поиска максимально, подобного происходящему в естественных интеллектуальных системах, требуется использовать категорию "нечеткость".

Пусть задано множество объектов A .

В дальнейшем будем отождествлять объекты с образами, если не оговорено иное.

Определение 9. На множестве образов (объектов) A определено отношение нечеткого подобия, если на нем определено отношение эквивалентности φ , и на декартовом произведении $A \times A$ определена мера истинности $\mu(\varphi) \in (0,1)$ отношения φ .

Определение 10. Два образа (объекта) a_i и a_j ($a_i, a_j \subset A$) являются нечетко подобными с уверенностью μ_{ij} , если на множестве A определено отношение нечеткого подобия φ , и паре объектов a_i и a_j соответствует значение меры истинности $\mu_{ij}(\varphi) \in [H_\varphi, 1]$, где H_φ – нижняя граница нечеткого подобия, причем $H_\varphi \in [0,1)$. Если нас не интересует мера уверенности нечеткого подобия, будем говорить, что образы a_i и a_j нечетко подобны (или неограниченно нечетко подобны), если $\mu_{ij}(\varphi) > 0$. Если задано значение $H_\varphi > 0$, будем говорить о нечетком подобии, ограниченном порогом H_φ .

Утверждение 1. Задание отношения нечеткого подобия на множестве образов (объектов) A равносильно заданию на этом множестве метрики расстояний между объектами.

Например, в нейронной сети Хопфилда с состояниями нейронов из интервала $(-1,1)$ и порогом $\theta = 0$ можно представить меру истинности подобия двух образов (векторов) в следующем виде:

$$\mu_{12}(\varphi) = \sum_i \text{Sign}(a_{1i}a_{2i}) \sum_j w_{ij},$$

где a_{1i} и a_{2i} – i -й разряд векторов (образов) a_1 и a_2 , соответственно;

функция $\text{Sign}(x)$ принимает значение 1, если $x \geq 0$, и 0 – в противном случае, w_{ij} – вес между i -м и j -м нейронами.

Необходимо только пронормировать ее, т.е. привести к диапазону значений $(0,1)$.

Как известно [23], при функционировании сети Хопфилда происходит минимизация энергетической функции

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j,$$

где x_i, x_j – состояния i -го и j -го нейронов.

Если величину $\mu_{12}(\varphi)$ инвертировать, то легко видеть связь ее с энергетической функцией. При этом надо иметь ввиду, что энергетическая функция E учитывает связи между как одинаковыми признаками (разрядами) образов a_1 и a_2 , так и между антагонистическими (имеющими разные значения), а функция $\mu_{12}(\varphi)$ учитывает только признаки, имеющие одинаковые значения. Можно представить формирование значения $\mu_{ij}(\varphi)$ двух образов a_i и a_j как взвешенной суммы разницы однотипных признаков:

$$\mu_{ij}(\varphi) = \frac{1}{n} \sum_k w_k \Delta(a_{ik}, a_{jk}),$$

где: $\Delta(a_{ik}, a_{jk})$ – разница k -х признаков i -го и j -го образов;

n – нормировочный коэффициент;

$a_i, a_j \in K$, где K – множество образов.

Вычисление разницы признаков Δ и нормировочного коэффициента n зависит от вида кодирования признаков. В случае кодирования образов в виде двоичных векторов разница вычисляется как сумма по модулю 2, а нормировочный коэффициент равен длине векторов.

Если сравниваемые векторы признаков, описывающих образы, имеют разную длину, надо выравнивать их длины по вектору с максимальной длиной и формировать в одном из векторов недостающие признаки. При этом можно использовать одну из трех стратегий:

1) оптимистическую, при которой недостающие признаки формируются так, чтобы для них разница была минимальна;

2) пессимистическую, при которой недостающие признаки формируются так, чтобы для них разница была максимальна;

3) умеренную, при которой недостающие признаки формируются так, чтобы для них разница принимала среднее значение из возможных значений.

Для более полного описания возможных ассоциативных связей между образами в интеллектуальных системах введем понятие нечеткого антиподобия.

Определение 11. На множестве образов (объектов) A определено отношение нечеткого антиподобия, если на нем определено отношение эквивалентности φ' , и на декартовом произведении $A \times A$ определена мера истинности $\mu(\varphi') \in (0,1)$ отношения φ' , причем, $\mu_{ij}(\varphi') \leq 1 - \mu_{ij}(\varphi)$ для всех пар объектов.

Определение 12. Два образа (объекта) a_i и a_j ($a_i, a_j \in A$) являются нечетко антиподобными с уверенностью μ_{ij} , если на множестве A определено отношение нечеткого антиподобия φ' , и паре образов a_i и a_j соответствует значение меры истинности $\mu_{ij} \in \mu(\varphi')$.

Два образа (объекта) могут одновременно быть нечетко подобны и нечетко антиподобны. Если подобие равносильно синонимии в естественном языке, то антиподобие – антонимии.

Отношение подобия описывает центростремительные связи между объектами (образами), а отношение антиподобия – центробежные. Первые обеспечивают группирование образов в кластер, соответствующий определенному отношению подобия, а вторые – объединение объектов в антагонистические пары или выявление противоположностей.

Если отношение подобия φ является отношением тождества, то отношение нечеткого подобия вырождается в отношение нечеткого равенства (тождества) объектов (образов). Это отношение моделирует возможности простых перцептронов по распознаванию образов, т.е. возможность отнесения похожих (близ-

ких в пространстве признаков) образов к одному и тому же классу (т.е. их отождествления).

Классификация образов в процессе их обработки интеллектуальной системой является следствием определения нечеткого подобия объектов. Нечеткое подобие является в свою очередь способом описания множества ассоциаций, существующих на множестве образов. Можно считать, что ассоциации первичны, а нечеткое подобие вторично.

Связь нечеткого подобия с ассоциациями, описанными выше, задается следующим утверждением.

Утверждение 2. Чтобы два образа A и B были нечетко подобными, необходимо и достаточно существование некоторого образа R и ассоциаций (A,R) и (B,R) .

Это утверждение 2 является основой для появления и определения процесса обобщения и (при его вербализации) классификации и понятия класса. Образ R можно отождествить с обобщенным образом или с классом.

Утверждение 3. Если существует пара ассоциаций (A,R) и (B,R) , то образы A и B являются нечетко подобными.

Утверждение 4. Если не существует пары ассоциаций (A,R) и (B,R) , то образы A и B не являются нечетко подобными.

Эти утверждения 3 и 4 являются следствиями из утверждения 2.

Связь концепции нечеткого подобия с классификацией задается следующим утверждением.

Утверждение 5. Образы, относящиеся к одному классу, являются нечетко подобными.

Теорема 1. Если два образа R_1 и R_2 являются нечетко подобными и существуют ассоциации (A, R_1) и (B, R_2) , то A и B нечетко подобны.

Доказательство. Предположим образы A и B не нечетко подобны. Так как R_1 и R_2 нечетко подобны, то из утверждения 2 существует образ R и ассоциации (R_1,R) и (R_2,R) . Так как ассоциация является транзитивным отношением (по определению 1), то из ассоциаций (A,R_1) и (R_1,R) следует ассоциация (A,R) , а из ассоциаций (A,R_2) и (R_2,R) следует ассоциация (B,R) . Из наличия ассоциаций (A,R) и (B,R) следует нечеткое подобие образов A и B . Но это противоречит первоначальному предположению. Следовательно, предположение о том, что образы A и B не являются нечетко подобными, не верно, что и следовало доказать.

Теорема 2. Если существуют ассоциации (A,R_1) и (A,R_2) , то образы R_1 и R_2 нечетко подобны.

Доказательство. В силу симметричности ассоциаций (определение 1) существуют ассоциации (R_1, A) и (R_2, A) . Из утверждения 3 следует, что образы R_1 и R_2 нечетко подобны, что и требовалось доказать.

Теорема 3. Если два образа A и B нечетко подобны и существуют ассоциации (A,R_1) и (B,R_2) , то образы R_1 и R_2 являются нечетко подобными.

Доказательство. Из нечеткого подобия A и B и утверждения 2 следует, что существует некоторый образ R и ассоциации (A, R) и (B, R) . Из свойства транзитивности ассоциаций (определение 1) следует, что существуют образы R_1' и R_2' и ассоциации (A, R_1') , (R_1', R) , (B, R_2') , (R_2', R) . Из утверждения 3 следует, что образы R_1' и R_2' нечетко подобны. Из существования ассоциаций (A, R_1') и (A, R_2') и (B, R_1') и (B, R_2') и из нечеткого подобия R_1' и R_2' по теореме 2 следует, что образы R_1 и R_2 нечетко подобны, что и требовалось доказать.

Теоремы 1, 2 и 3 справедливы для неограниченного нечеткого подобия, когда для образов a_i и a_j $\mu_{ij}(\varphi) > 0$. Если задано ограничение подобия (порог) $H_\varphi > 0$, то теоремы 1, 2 и 3 в общем случае не справедливы. Однако, справедливо следующее утверждение.

Утверждение 6. Для любого отношения подобия φ и для любых образов и ассоциаций существует такое значение $H_\varphi > 0$, при котором справедливы теоремы 1, 2 и 3.

1.4. Принципы организации функционирования интеллектуальных систем

1.4.1. Принцип обучения посредством формирования и последующего закрепления ассоциаций, происходящего в процессе ассоциативного вспоминания

Способность обучаться (адаптироваться, пополнять и корректировать базу знаний и т.п.) является неотъемлемой частью любой действительно интеллектуальной системы (те искусственные "интеллектуальные" системы, которые не обладают этим качеством, безусловно, не могут претендовать на звание "интеллектуальных").

Принцип обучения можно сформулировать следующим образом.

Интеллектуальная система в процессе взаимодействия с внешней средой запоминает ассоциации между разными образами (стимулами, сигналами, знаками и т.п.), которые используются в процессе планирования ее поведения посредством ассоциативного вспоминания образов по их фрагментам. При достаточном закреплении ассоциаций они могут превращаться в обозначения в форме отношений (атрибутивных, причинно-следственных, надежных и т.п.) между сущностями.

Более детальное описание этого принципа содержится в предыдущем разделе 1.3.

Принцип ассоциативного вспоминания является вариантом формулирования предложенной Прибрамом в [76] голографической метафоры работы мозга.

1.4.2. Принцип концентрации и экономии ресурсов

Принцип концентрации и экономии ресурсов можно сформулировать в общем виде следующим образом.

В интеллектуальной системе (естественной или искусственной) существует механизм выделения (распознавания) и активизации тех информационных ресурсов (нейронов, нейронных ансамблей, фреймов, правил и т.п.), которые существенны для решения текущей задачи интеллектуальной системой, и выключения (деактивации) тех ресурсов, которые для решения задачи не существенны.

Принцип концентрации и экономии на уровне сознательного мышления неразрывно связан с понятием целеполагания [76] и выражается в концентрации усилий (информационных и энергетических ресурсов) для достижения поставленной цели. На уровне подсознательного мышления принцип концентрации выражается, например, в решении задачи концентрации внимания (модель ART Гроссберга [90], называемая адаптивной резонансной теорией).

Этот принцип, также, аккумулирует в себе знания, накопленные в нейропсихологии в форме теории возбуждения и торможения.

В естественном мозге целеполагание или концентрация внимания реализованы в ретикулярной формации [76].

Этот принцип, по-видимому, связан с механизмом эмоций в интеллектуальных системах [91]. Можно сказать, что сила эмоций влияет на степень концентрации ресурсов, а характер эмоций – на выбор цели, для достижения которой концентрируются ресурсы. Можно также предположить, что эмоции связаны и с предыдущим принципом и сила и "знак" эмоций (положительные или отрицательные) влияет на закрепление ассоциаций, связанных с возникновением положительных эмоций.

1.4.3. Принцип неопределенности

"Изучение науки о мозге лучше подготовит вас к восприятию квантовой теории, чем изучение классической физики".

Р.А. Уилсон. Квантовая психология

Так же как в квантовой физике существует принцип неопределенности Гейзенберга, устанавливающий соотношение между точностью определения координат элементарной частицы и ее энергии, в теории интеллектуальных систем можно сформулировать принцип неопределенности, устанавливающий соотношение между точностью определения (распознавания) описания внутренней семантики объекта (синтаксиса или структуры образа) и его взаимосвязей с другими объектами (внешней или просто семантики образа). Он может быть сформулирован следующим образом.

Чем более точно распознается (интерпретируется) структура образа, т.е. его внутренняя семантика, тем менее точно распознается его взаимодействие с другими образами (его внешняя семантика) и наоборот.

Этот принцип неопределенности означает, что любая интеллектуальная система не может одинаково хорошо разбираться в синтаксисе образов и в их семантике или, другими словами, нельзя одновременно изучать глубоко структуру объекта и его взаимосвязи с другими объектами. Или, другими словами, в той степени, в которой разработчик (интерпретатор, пользователь) представления знаний в интеллектуальной системе уделяет большое внимание представлению и распознаванию деталей в структуре объектов (понятий), он вынужден обходить вниманием представление и распознавание взаимосвязей между ними. Этот принцип работает как при формализации знаний в процессе разработки интеллектуальной системы, так и при ее использовании в процессе интерпретации (применения) знаний.

Подобно тому, как погрешность определения координат и энергии элементарной частицы зависит от характеристик "наблюдателя" (средств измерений), погрешность обработки знаний зависит от объема памяти вычислительного средства и быстродействия логического вывода, используемых при интерпретации знаний.

Глубинной причиной принципа Гейзенберга так же как причиной выше сформулированного принципа является факт единства наблюдаемого и наблюдателя и необходимость их рассмотрения как единой системы. И применительно к интеллектуальной системе не суть важно кто (что) является наблюдателем знаний – интерпретатор знаний, являющийся ее частью или внешний наблюдатель (исследователь или разработчик).

Возможно, эта аналогия между образом и элементарной частицей не случайна. Не даром, в [76] была сформулирована аналогия между голограммой и образом в биологической памяти, и процесс вспоминания обладает свойствами, схожими со свойствами процесса восстановления голограммы.

Возможно, одно из причин появления в естественных интеллектуальных системах в ходе эволюции способности к абстрагированию является этот принцип, т.к. он ограничивает возможности системы устанавливать причинно-следственные связи, строить планы, используя работу только с конкретными образами внешнего мира.

При формализации базы знаний интеллектуальной системы (например, экспертной системы) этот принцип проявляется в том, что если разработчик базы знаний хочет описать как можно больше взаимосвязей некоторого понятия с другими понятиями, представление этого понятия "размывается". Его выделение как самостоятельного понятия (структуры данных) становится проблематичным. Оно все больше становится частью других понятий и теряет свое значение как отдельный объект со своей внутренней структурой.

Проявлением этого принципа в инженерии знаний является появление различных методов структуризации или масштабирования знаний, таких как, структурированные семантические сети, фреймы, онтологии, мультиагентные системы.

1.4.4. Принцип единства нечетких рассуждений и четких действий

В основе рассуждений, происходящих в естественной интеллектуальной системе, всегда лежит оперирование с нечеткими или искаженными образами, неточными понятиями, неполными описаниями, условными выводами при неполной информации и т.п. Причинами этого является ограниченность возможностей органов чувств, локальность восприятия ими внешней среды и недетерминированный характер внешней среды (мира), хотя последний фактор спорен, т.к. мы судим о внешнем мире посредством тех же самых ограниченных органов чувств, пусть даже усиленных всевозможными инструментами и приборами. С другой стороны действия, которые производит интеллектуальная система, носят точный характер, например, "хватание предмета", "бросание предмета", "поднятие руки", "рождение ребенка", "включение определенной мышцы" и т.п.

В простейшем случае при выработывании условного рефлекса происходит формирование ассоциативной связи между ситуацией и действием, которое необходимо произвести при ее возникновении. Отсюда, процесс рассуждений сводится к распознаванию ситуации и ассоциативному вспоминанию соответствующего ему действия. Кажется оправданным предположить, что попытка рассуждать в терминах такого "ситуативного" мышления привела к появлению абстрактного мышления, таксономии с классификацией и классической логики. Ведь по существу строгий логический вывод или решение задачи в четко описанной математически постановке сводится к цепочке принятия решений по принципу "ситуация-действие" (здесь под действием подразумевается операция над формулами).

Более творческий процесс рассуждений задействует образное мышление (см. п. 1.2), которое обеспечивает снижение риска потери информации о ситуации (задаче) в процессе принятия решений, т.к. использует на всех уровнях решения задачи (до самого принятия решения о действии) большое количество признаков, описывающих образ ситуации. Чем больший вектор признаков используется в цепочке ассоциативного вспоминания (рассуждения), тем большая часть мозга вовлечена в процесс рассуждений и тем больше возможность получения нетривиальных решений.

Принцип, указанный в заголовке, можно сформулировать следующим образом:

В основе рассуждений лежит оперирование с нечеткими образами посредством ассоциативного вспоминания образов (см. п. 1.3.2), в конце которого осуществляется выбор четкого действия (его вспоминание), с которым можно ассоциировать выбор достигаемой цели (решаемой

задачи), фокусировка внимания, запуск действия как программы срабатывания моторных нейронов и т.п. При этом выбранное действие в качестве признака вовлекается в дальнейший процесс рассуждений.

1.5. Количественная оценка знаний в сообщении

При передаче знаний между индивидуумами (пользователями знаний) возникает проблема понимания знания интеллектуальной системой-приемником. Только в случае понимания можно говорить о том, что переданное знание пополнило базу знаний, и количество знаний у приемника увеличилось. В противном случае, передаваемое знание содержит только информацию (меру неоднородности носителя информации, количество которой можно вычислить по известным формулам из теории информации). Проблема понимания возникает в силу двух причин:

- 1) использования для передачи знаний какой-либо знаковой системы и необходимости преобразования внутреннего представления знаний в представление в рамках этой знаковой системы;
- 2) разного онтогенеза при формировании баз знаний передатчика и приемника знаний, приводящего к разной интерпретации используемой знаковой системы.

Что же можно определить как количество знаний при передаче? Приведем несколько примеров знаний, выраженных на ЕЯ.

Пример 1. "Аист – птица".

Пример 2. "Аист умеет летать".

Пример 3. "Аист принес сына".

Если приемник имеет представление о том, что такое "птица" и что такое "аист" (например, однажды его видел и связал с его обликом слово "аист"), но не знает, что аист относится к разряду птиц, для него в первом примере содержится элемент знания. Это – отношение (или ассоциация) между понятиями "аист" и "птица". Если же у него уже промелькнула мысль о том, что "аист это птица", когда он его увидел (большинство людей способно на это), пример 1 не содержит для него никакого нового знания, т.е. количество знаний в нем равно нулю.

Если в примере 1 одно из понятий отсутствует в базе знаний приемника (например, "аист"), то новое знание присутствует в нем. Это отношение между понятием "птица" и некоторым новым понятием "аист", относительно которого приемник ничего не узнает нового, кроме того, что это нечто, ассоциативно связанное с понятием "птица". Таким образом, и в этом случае можно считать, что в примере 1 содержится элемент знания в виде отношения (ассоциации).

Рассмотрим пример 2. В случае, если в базе знаний приемника есть все три понятия, упоминаемые в нем, но без относительно друг к другу, в примере 2 содержатся два новых отношения: между "умеет" и "аист" и между "умеет" и "летать". Такая ситуация возможна, если мозг приемника обучался понятию "уметь" на других примерах умений (а не летать). Если же понятие "уметь" уже

связано с понятием "летать", в примере 2 будет всего одно новое отношение. Если же приемник уже однажды вывел для себя (используя для этого знание из примера 1 и знание о том, что птица умеет летать) знание о том, что "аист умеет летать", то в примере 2 для него нет нового знания, т.е. количество в нем знаний равно нулю.

В примере 3 имеют силу те же рассуждения, что и для примера 2. Кроме того, если даже в примере 3 не содержится нового знания, в случае, если человек знает соответствующую метафору, он получит знание о том, что у кого-то родился сын, т.е. в примере 3 для него будет неявно содержаться новое отношение между понятиями "родился" и "сын". Так как неизвестно у кого он родился, второго отношения в базу знаний приемника не прибавится.

Все эти рассуждения на примерах позволяют сформулировать следующее утверждение.

Утверждение 7. Количество знаний, содержащихся в сообщении, формализованном в какой-либо знаковой системе, передаваемом от интеллектуальной системы A интеллектуальной системе B , равно количеству бинарных отношений (ассоциаций) между понятиями, содержащимися в сообщении, удовлетворяющих следующим условиям: 1) эти отношения не известны системе B ; 2) хотя бы одно из пары понятий, связанных отношением, известно системе B .

Известность понятия системе означает наличие в памяти системы этого понятия, т.е. либо его образа, либо его знака.

Известность отношения системе означает присутствие его в памяти системы, т.е. наличие ассоциации между какими-либо понятиями или образами, соответствующее этому отношению.

Следует иметь в виду, что в этом утверждении не оценивается истинность/ложность и надежность/ненадежность получаемых из сообщения знаний. Эта задача решается в процессе рассуждений "приемника" над полученным знанием.

Пусть в сообщении $S = (\Omega^s, \Psi^s)$ содержится множество понятий (термов) $\Omega^s = \{\omega_i^s \mid i = (1, N_\omega^s)\}$ и множество бинарных отношений, $\Psi^s = \{\psi_j^s(\omega_k^s, \omega_l^s) \mid j = (1, N_\psi^s), k, l = (1, N_\omega^s)\}$ между ними. Пусть база знаний интеллектуальной системы-приемника $K = (\Omega^r, \Psi^r)$ содержит $\Omega^r = \{\omega_i^r \mid i = (1, N_\omega^r)\}$ и множество бинарных отношений $\Psi^r = \{\psi_j^r(\omega_k^r, \omega_l^r) \mid j = (1, N_\psi^r), k, l = (1, N_\omega^r)\}$ между ними. Тогда количество знаний в сообщении S равно

$$K = \sum_{\Psi^s} \text{Sign}(\psi_i^s(\omega_k^s, \omega_l^s) \notin \Psi^r \wedge (\omega_k^s \subset \omega_l^s \subset \Omega^r)),$$

где $\text{Sign}(\psi_i^s(\omega_k^s, \omega_l^s) \notin \Psi^r \wedge \omega_k^s \subset \omega_l^s \subset \Omega^r) = 1$, если отношение ψ_i^s не содержится в базе знаний приемника и понятия ω_k^s, ω_l^s содержатся в базе знаний приемника, и 0 – в противном случае.

1.6. Многомерные лингвистические переменные и иерархические нейронные сети

"Нет никаких фактов, есть лишь интерпретация".
Ф. Ницше

При разработке современных прикладных систем искусственного интеллекта одной из главных проблем является представление нечетких знаний и реализация рассуждений с их использованием. Эта проблема решается в зависимости от применяемого подхода (нейрокибернетического или логического) в основном с использованием механизма нейронных сетей [36-48] и лингвистической переменной [7], соответственно.

Недостатком применения нейронных сетей является трудность визуализации или вербализации результатов их работы, а также, перехода к логической обработке информации.

Этих недостатков лишен механизм лингвистической переменной. Однако, у классической лингвистической переменной есть свои недостатки:

- ее описание носит субъективный характер;
- ее описание зависит от контекста, в котором используется лингвистическая переменная, но этот контекст не описывается в рамках данного формализма.

механизма перехода от логической обработки информации (в рамках традиционных механизмов логического вывода) к ассоциативной обработке с использованием нейронных сетей и обратно.

Этих недостатков можно избежать, расширив понятие лингвистической переменной до т.н. многомерной лингвистической переменной.

Как известно, лингвистической переменной называется переменная, которая может принимать как символьные так и численные значения, связанные между собой отношением нечеткой принадлежности.

Другими словами, лингвистическая переменная описывается множеством символьных значений $S = [s_i] \mid i = (1, n)$, с каждым из которых связано нечеткое множество значений $V_i = v_j \mid j = (1, m_i)$ обычно, упорядоченных в виде метрической шкалы, на которой определена функция принадлежности $\mu_j^i = (0, 1)$. Если используется метрическая шкала, то значения имеют смысл чисел, если нет – эти значения также являются символьными.

Если рассматривать шкалу как размерность, то можно определить n -мерную лингвистическую переменную как множество символьных значений $S = [s_i]$, с каждой из которых связана функция принадлежности μ^i , определенная на множестве из n шкал, которые могут быть как метрическими так и топологическими.

Например, лингвистическая переменная "кровяное давление" может иметь значения "высокое", "нормальное", и "низкое", определенные на шкалах "возраст" и "вес". Это означает, что для возраста 20 лет будет одна кривая, описывающая функцию принадлежности, а для возраста 50 лет – другая. Более того,

лингвистическая переменная "кровенное давление" может рассматриваться как одно из значений лингвистической переменной "давление", в описание которой добавляется еще одна топологическая шкала, описывающая тип материала, давление которого описывается. Помимо значения "кровь" эта переменная может иметь другие, например, "вода в водопроводе", "вода в океане", "газ в трубопроводе" и т.п. Таким образом можно задавать контекст, в зависимости от которого кривая функции принадлежности может иметь разный вид.

Другой пример – лингвистическая переменная "напряжение" (электрическое). Она может иметь множество символьных значений: "высокое", "нормальное", "низкое". Каждое из них определяется на метрической шкале. Но семантика "высокого напряжения в энергетической системе" отличается от семантики "высокого напряжения в бытовой электротехнике". Чтобы отразить эту разницу введем топологическую шкалу "область применения", на которой зададим несколько значений: "энергетика", "бытовая электротехника", "телевизоры и мониторы", "компьютеры" (рис. 1.5).

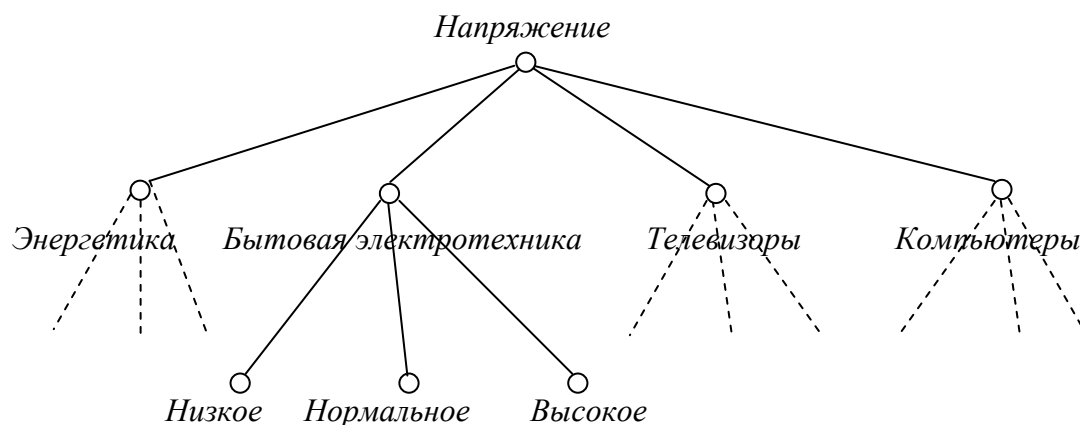


Рис. 1.5. Иерархия понятий и связь их с лингвистическими переменными

Такое представление лингвистической переменной можно использовать для его реализации в виде нейронной сети. При этом на основе нейронной сети реализуется ассоциативная память, в которой в процессе обучения формируются ассоциативные связи между численными и символьными значениями лингвистической переменной. При этом можно использовать любую модель, в которой реализуется ассоциативная память, например, модель Хопфилда или многослойный перцептрон.

При использовании многослойного перцептрона можно естественным образом получать описание новых лингвистических переменных на каждом новом слое многослойного перцептрона. При этом метрическая шкала (или шкалы), является входом очередного слоя перцептрона, а значения лингвистической переменной – его выходами. Если для описания лингвистической переменной требуется несколько шкал, зависящих от контекста, то шкалы-контексты добавляются как входы соответствующих уровней контекста слоев нейронной сети.

Кроме того, входами очередного слоя являются и выходы предыдущего слоя, т.е. значения лингвистической переменной, формируемой в нем. Такая структура нейронной сети напоминает структуру растущих семантических сетей, предложенных В.П. Гладуном в [12].

Таким образом, концепция "*n*-мерная лингвистическая переменная" имеет следующие полезные свойства, отличающие его от классического понятия:

- позволяет естественным образом строить иерархию лингвистических переменных, включая на каждом из следующих верхних уровней имя (обозначение) лингвистической переменной нижележащего уровня в качестве дополнительной размерности;

- позволяет естественным образом связать описание лингвистической переменной с описанием нейронной сети, считая гиперповерхность, описывающую функцию принадлежности функцией, обратной энергетической функцией нейронной сети;

- позволяет иерархию описаний лингвистических переменных рассматривать, как ассоциативную память, которая может быть реализована в виде нейронной сети, а процесс поиска решения как ассоциативное восстановление вектора состояния нейронной сети по его фрагменту, при котором происходит минимизация энергетической функции с возможностью "замораживания" некоторых состояний нейронов (известных с абсолютной уверенностью факторов) перехода от представления и обработки признаков и чисел в нейронной сети к представлению и обработке символьной информации в виде значений лингвистических переменных.

1.7. Моделирование эмоций

Эмоции без сомнения являются очень важным механизмом функционирования живых существ и неразрывно связаны с планированием поведения. Давно установлено [92] наличие в мозгу центров удовольствия и неудовольствия, ответственных за поведение живых систем. В связи с этим логично все многообразие эмоций свести к двум основным классам – положительным и отрицательным, в основе которых лежит активность соответствующих центров в мозгу, и из которых в зависимости от контекста происходит формирование всех остальных эмоций, имеющих свои наименьшие значения. Но положительные эмоции используются для закрепления ассоциаций в процессе обучения достижению цели, а отрицательные – в процессе обучения избеганию цели. Кроме того, сила эмоций управляет концентрацией усилий на решении той или иной задачи интеллектуальной системой.

Утверждение 8. Причиной положительных эмоций является совпадение ожидаемых (планируемых) образов с воспринимаемыми при решении задачи (достижении цели), а отрицательных – несовпадение.

Следует иметь в виду, что это утверждение справедливо только для прогнозирования образов в процессе планирования поведения системы. Если про-

гнозирование происходит безотносительно к планированию поведения (например, ожидание продолжения рассказа), оно может вызвать совсем другие эмоции. Например, можно предположить, что причиной положительных эмоций (вызывающих улыбку или хохот) при рассказе анекдота является не соответствие продолжения с ожидаемыми выводами из ранее сказанного (при условии, если у слушающего не возникает ассоциаций между персонажем анекдота и своей персоной).

Положительные эмоции в искусственных интеллектуальных системах можно использовать для подкрепления ассоциаций, используемых для планирования поведения (решения задачи, достижения цели). Отрицательные – для подкрепления ассоциаций с понятиями (образами, действиями), которые необходимо избегать.

1.8. Выводы

В главе предложены следующие концепции и модели:

- 1) модель интеллектуальной системы, в которой отражены понятия вербального и образного мышления, сознания и подсознания;
- 2) модель ассоциативного мышления, как цепочки ассоциативных поисков на множестве пар образов, называемых ассоциациями;
- 3) концепция нечеткого подобия образов, тесно связанная с моделью ассоциативного мышления;
- 4) сформулированы три основных принципа организации функционирования интеллектуальных систем:
 - принцип обучения посредством формирования и последующего закрепления ассоциаций, происходящего в процессе ассоциативного вспоминания;
 - принцип концентрации и экономии ресурсов;
 - принцип неопределенности.
- 5) предложены подходы к измерению количества знаний, содержащихся в сообщении на языке, естественном или близком к естественному.
- 6) предложена модель многомерной лингвистической переменной, как механизма, который может быть связующим звеном между нейронными сетями и символьной обработкой информации в искусственных интеллектуальных системах.
- 7) сформулированы причины и роль эмоций в функционировании интеллектуальных систем.

Предложенные в главе концепции и модели можно использовать как основу для дальнейшего развития теории интеллектуальных систем. Ее развитие позволит разрабатывать экспертные системы и интеллектуальные роботы, процесс рассуждений в которых более приближен к тому, который имеет место в мозгу человека.

При этом ключевой проблемой, которую следует решить, является моделирование процесса перехода от сигнального представления образа к его знако-

вому представлению (в виде термина). При этом запоминание ассоциаций на определенном уровне обработки знаний превращается в формирование отношений, и обратно. Это может происходить в форме взаимодействия между нейронными сетями, обрабатывающими разные подмножества образов, или между нейронными сетями и классическими механизмами поиска решений в экспертных системах.

Некоторые концепции, описанные в этой главе, были использованы при разработке архитектур и программных продуктов, описанных в следующих главах, в частности:

- 1) модель интеллектуальной системы (см. 1.2) и парадигма ассоциативного поиска лежат в основе "двухполушарной" архитектуры интеллектуальной системы, описанной в разделе 3.6;
- 2) парадигма нечеткого подобия используется в архитектурах программного обеспечения, описанных в главе 4 и предназначенных для решения разных задач, связанных с обработкой естественного языка.

Кроме того, концепции и модели, описанные в этой главе, являются обобщением опыта, накопленного при разработке разных систем искусственного интеллекта, описанных в главах 2, 3 и 4.

2. КОМБИНИРОВАНИЕ РАЗНЫХ МЕТОДОВ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ЗНАНИЙ В ГИБРИДНЫХ ЭКСПЕРТНЫХ СИСТЕМАХ

*"Никогда не становись экспертом,
это подавляет способность рассуждать".*

Ф. Герберт. Капитул Дюны

2.1. Методы представления знаний в гибридных экспертных системах

Обобщенная архитектура экспертной системы [1-11] показана на рис. 2.1.

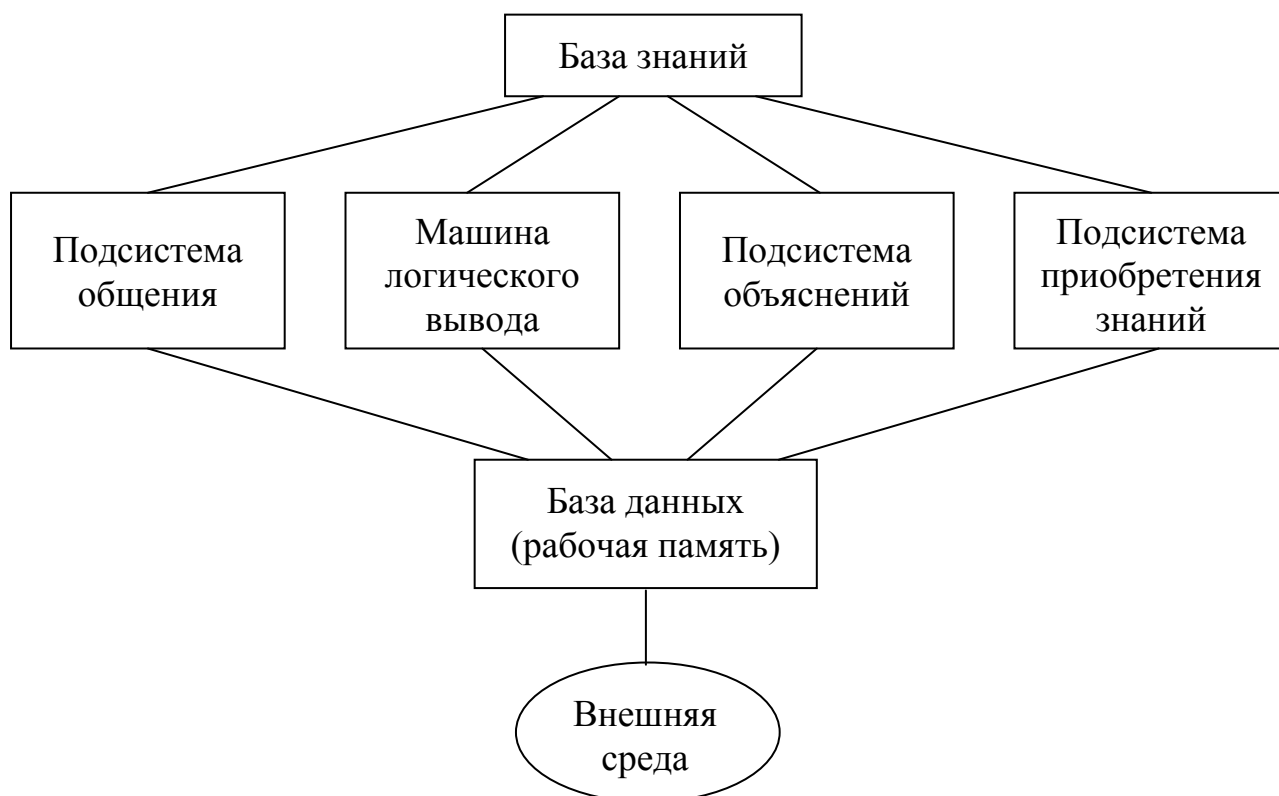


Рис. 2.1. Обобщенная структура экспертной системы

База знаний предназначена для хранения экспертных знаний о предметной области, используемых при решении задач экспертной системой.

База данных предназначена для временного хранения фактов или гипотез, являющихся промежуточными решениями или результатом общения системы с внешней средой, в качестве которой обычно выступает человек, ведущий диалог с экспертной системой.

Машина логического вывода – механизм рассуждений, оперирующий знаниями и данными с целью получения новых данных из знаний и других данных, имеющихся в рабочей памяти. Для этого обычно используется программно реализованный механизм дедуктивного логического вывода (какая-либо его

разновидность) или механизм поиска решения в сети фреймов или семантической сети (поиска релевантных знаний).

Подсистема общения служит для ведения диалога с пользователем, в ходе которого ЭС запрашивает у пользователя необходимые факты для процесса рассуждения, а также, дающая возможность пользователю в какой-то степени контролировать и корректировать ход рассуждений экспертной системы.

Подсистема объяснений необходима для того, чтобы дать возможность пользователю контролировать ход рассуждений и, может быть, учиться у экспертной системы. Если нет этой подсистемы, экспертная система выглядит для пользователя как "вещь в себе", решениям которой можно либо верить, либо нет. Обычно пользователь выбирает последнее, и такая ЭС не имеет перспектив для использования.

Подсистема приобретения знаний служит для корректировки и пополнения базы знаний. В простейшем случае это – интеллектуальный редактор базы знаний, в более сложных экспертных системах – средства для извлечения знаний из баз данных, неструктурированного текста, графической информации и т.д.

Архитектура экспертных систем характеризуется следующими особенностями ее построения:

- применяемый метод (методы) представления знаний (в том числе, нечетких);
- применяемый метод (методы) решения задач (интерпретации базы знаний);
- применяемый (или нет) метод (методы) приобретения знаний;
- особенности и глубина реализации подсистемы объяснений;
- особенности реализации подсистемы общения (взаимодействия с внешним миром).

Особенности архитектуры экспертных систем можно рассматривать в контексте трех основных классов характеристик (рис. 2.2): быстродействие, гибкость и понятность.

Быстродействие характеризует скорость решения задач или реакции на входную информацию. В контексте быстродействия различают статические и динамические экспертные системы [10]. Среди последних, в свою очередь, выделяются экспертные системы реального времени.

В контексте гибкости можно выделить следующие характеристики:

- поддержка разных парадигм представления знаний и решения задач;
- возможность интеграции с другими программными системами;
- легкость наращивания базы знаний, в частности, наличие возможностей автоматизации процесса приобретения новых знаний, т.е. обучения;
- легкость модификации программ;
- уровень языка программирования, на котором реализована экспертная система.

В контексте понятности можно рассматривать следующие характеристики:

- структуризация базы знаний;

- масштабируемость работы с базой знаний;
- использование графики в пользовательском интерфейсе;
- использование естественного языка в пользовательском интерфейсе;
- адекватность пользовательского интерфейса особенностям предметной области.

Треугольник иллюстрирует тот факт, что каждая пара из этих трех классов находится в отношении конкуренции друг с другом. Так, например, увеличивая гибкость, разработчики, как правило, тем самым уменьшают быстродействие и понятность, и наоборот. Увеличение понятности (это обычно связано с увеличением специализации экспертной системы), как правило, приводит к уменьшению гибкости и уменьшению быстродействия. Последнее может быть связано с упрощением структуры, а следовательно, с уменьшением возможности оптимизации поиска вариантов на разных этапах решения задачи, либо с увеличением доли графики и анимации, в том числе, многомерной в пользовательском интерфейсе.

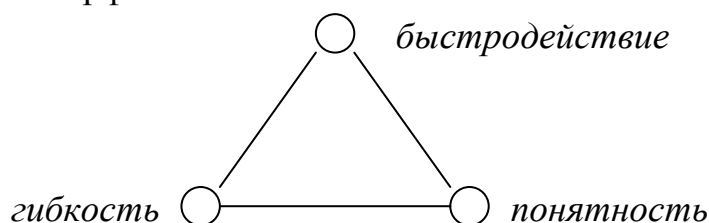


Рис. 2.2. Классы характеристик экспертных систем

Понятие гибридного интеллекта появилось впервые в работе [12] применительно к человеко-машинным системам (ЧМС), в которых под интеллектом понимался как искусственный интеллект, так и естественный интеллект пользователя. Разновидность экспертных систем, в которых существенная роль при решении задач отводится пользователю, получила название партнерских систем [13, 14]. Но в последнее время наметилась тенденция применения этого термина к искусственным интеллектуальным системам, не зависимо от участия в них человеческого интеллекта. В этой работе автор придерживается такого подхода.

Обычно под гибридными экспертными системами понимаются экспертные системы, в которых реализуются разные парадигмы (методы, модели) представления и интерпретации знаний, а также, подсистема взаимодействия с внешним миром не сводится только к пользовательскому интерфейсу.

В настоящее время существует большое разнообразие методов представления знаний, которые можно разделить на пять групп:

- логические;
- продукционные;
- фреймовые;
- семантические сети;
- нейронные сети.

Методы обработки (интерпретации знаний) обычно трудно (а иногда, невозможно) отделить от их представления и поэтому в дальнейшем будем говорить о методах представления и обработки знаний.

Каждая из перечисленных групп имеет свои преимущества, недостатки, условия и цели применимости.

Логические методы [15-17] представления знаний характеризуются хорошим математическим обоснованием. Цель их использования – дать базис для построения других методов представления знаний или для реализации языков программирования искусственного интеллекта, таких как Prolog [18-20] или SmallTalk [21]. Особое место среди них занимают методы для представления нечетких знаний: нечеткая логика [22], псевдофизические логики [23], лингвистические переменные [24] и т.п. Наиболее развитая из этих моделей – псевдофизические логики, в которых аккумулируются концепции нечетких множеств, лингвистической переменной и логики предикатов 1-го порядка.

Продукционные методы [25-28] дают возможность создавать и визуализировать базу знаний в привычной для человека нотации "ЕСЛИ ТО". С другой стороны они базируются на хорошо обоснованных логических методах доказательства и, в то же время, дают возможность достаточно вольно интерпретировать правила в зависимости от особенностей решаемых задач.

Цель фреймовых моделей [27, 29, 30] – дать удобную форму визуализации текстовой информации, описывающей понятия, с инкапсуляцией и наследованием свойств и ориентацией на представление иерархических структур знаний. К ним можно отнести и объектно-ориентированную модель, появившуюся позднее на основе, по всей видимости, концепции фреймов.

Семантические сети [31, 32] ориентированы на визуализацию знаний в виде графов. Они появились раньше фреймов как средство описания смысла, заключенного в предложениях естественного языка, и концепцию фреймов можно рассматривать, как попытку структурировать семантические сети и дать им лучшую обозримость путем комбинации текстового и графического представления информации.

Нейронные сети [33-60] до недавнего времени рассматривались отдельно от всех других выше перечисленных методов (вне инженерии знаний). Это связано с тем, что в них изначально речь шла об обработке сигналов, а не каким-либо образом формализованных данных или знаний. Сейчас, когда нейронные сети все больше применяются для обработки текстовой информации и извлечения знаний из данных, и наметилась тенденция использовать нейронные сети как основу построения экспертных систем, их можно выделить в отдельную группу методов представления знаний. Цель применения нейронных сетей как метода представления и обработки знаний – дать возможность наиболее естественным образом представлять нечеткие знания, реализовать процесс обучения на примерах на основе ассоциативного запоминания/вспоминания информации (см. 1.3 и 1.4.1).

В последнее десятилетие появились новые парадигмы для представления знаний, такие как онтологии [61, 62] и интеллектуальные агенты [63, 64].

Онтологии служат для структуризации больших баз знаний, построенных с использованием других методов представления (фреймов, правил, семантических сетей) и являются промежуточной категорией между понятием базы знаний (в целом) и единицы знаний (фрейма, узла семантической сети, отношения и т.п.).

Интеллектуальные агенты служат для той же цели, что и онтологии. Только их применение направлено на декомпозицию не только базы знаний, а интеллектуальной системы в целом, как сложной программной или программно-аппаратной системы, с целью обеспечения возможности построения распределенных интеллектуальных систем [65-66].

В рамках гибридных экспертных систем можно использовать разные комбинации методов представления знаний. В экспертных системах и инструментальных средствах для построения экспертных систем наиболее часто применяются комбинации правил-продукций с фреймами (XCON[5], CENTAUR [67], СПЭЙС [68], ESWin [69-72, см. раздел 2.2], BABYLON, Flex, AION, Eclipse, Arity Expert, Art*Enterprise) и семантическими сетями (PROSPECTOR [73], ДЕКЛАР [74]).

Во всех методах представления из выше перечисленных (кроме нейронных сетей) реализуется в том или ином виде логический вывод (или другими словами моделируется вербальное или логическое мышление). В нейронных сетях реализуется ассоциативная обработка информации (или моделируется ассоциативное мышление) (см. 1.2).

В последнее время наметилась тенденция объединять в одной системе ассоциативную и логическую обработку информации (см. главы 3, 4). К таким системам относится, например, экспертная система для управления прокатным станом [75], диагностическая экспертная система для подводного аппарата [76], экспертная система "Investor" для выдачи рекомендаций об инвестициях [77], экспертная система для диагностики аварийных ситуаций в работе энергосистемы [78], экспертная система для профориентации "PROFEXOR" [79, см. раздел 3.4].

2.2. Архитектура инструментального программного обеспечения ESWin для создания гибридных экспертных систем [69-72]

"Всякий порядок произволен".
Ф. Герберт. Капитул Дюны

2.2.1. Состав и назначение ПО

ESWin — инструментальное программное обеспечение для создания экспертных систем, ориентированное на решение задач диагностики, идентификации и классификации. В его состав входят пять программ (рис. 2.3): программ-

The diagram illustrates the architecture of an expert system shell, divided into two main user groups: the developer (Разработчик экспертной системы) and the end user (Конечный пользователь).

Developer Side (Left):

- База знаний (Knowledge Base):** The central component for the developer.
- Редактор баз знаний EdKB (Knowledge Base Editor):** Used by the developer to interact with the knowledge base.
- База фактов (Fact Base):** A component that receives data from the knowledge base and external programs.
- Программа KBView (Knowledge Base Viewer):** A tool for viewing the knowledge base.

User Side (Right):

- Базы данных (Databases):** Data sources for the end user.
- Внешние программы (External Programs):** Programs that interact with the system.
- Оболочка ESWin (ESWin Shell):** The main interface for the end user.
- Оболочка ESWinK (ESWinK Shell):** A specialized interface component.

Interactions:

- The **Редактор баз знаний EdKB** interacts with the **База знаний**.
- The **База знаний** interacts with the **База фактов**, **Базы данных**, and **Внешние программы**.
- The **База фактов** interacts with the **Базы данных** and **Внешние программы**.
- The **Базы данных** and **Внешние программы** interact with the **Оболочка ESWin** and **Оболочка ESWinK**.
- The **Оболочка ESWin** and **Оболочка ESWinK** interact with each other.

В пакете ESWin реализовано представление знаний в виде правил-продукций, фреймов и лингвистических переменных. Для решения задач в ней реализован нечеткий обратный логический вывод. Кроме того, оболочка ESWin позволяет запускать из базы знаний внешние программы и получать в процессе логического вывода факты из внешних реляционных баз данных.

2.2.2. База знаний

[illegible]

```

ENDF
RULE 1                                // правило-продукция
    <описание условий правила>
DO
    <описание заключений правила>
ENDR
...
RULE n                                // правило-продукция
    <описание условий правила>
DO
    <описание заключений правила>
ENDR

```

Пример базы знаний:

```

TITLE = для выбора метода представления знаний
FRAME = Цель
    Метод представления знаний: ()
ENDF
FRAME = Тип
    Решаемые задачи: (диагностика; проектирование)
ENDF
FRAME = Область
    Применение [Какова область применения?]: (медицина; вычислитель-
        ная техника)
ENDF
FRAME = Количество
    Число правил в базе знаний (численный): ()
    Число объектов в базе знаний (численный): ()
ENDF
FRAME = Действие
    Сообщение: ()
ENDF
RULE 1
    > (Количество.Число правил в базе знаний; 50)
    < (Количество.Число правил в базе знаний; 100)
    < (Количество.Число объектов в базе знаний; 30)
DO
    = (Тип.Решаемые задачи; диагностика) 100
ENDR
RULE 2
    > (Количество.Число правил в базе знаний; 100)
    > (Количество.Число объектов в базе знаний; 30)
DO

```

```

      = (Тип.Решаемые задачи; проектирование) 100
ENDR
RULE 3
      = (Область.Применение; медицина)
      = (Тип.Решаемые задачи; диагностика)
DO
      = (Метод представления знаний; Правила-продукции с представлени-
        ем нечетких знаний) 90
ENDR
RULE 4
      = (Область.Применение; вычислительная техника)
      = (Тип.Решаемые задачи; проектирование)
DO
      = (Метод представления знаний; Фреймы) 100
      = (Метод представления знаний; Правила-продукции с представлени-
        ем нечетких знаний) 70
      = (Метод представления знаний; Семантические сети) 70
      MS(Действие.Сообщение; Доказано правило 4)
ENDR

```

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

До начала работы с экспертной оболочкой база знаний находится в текстовом файле. В файле с расширением *.klb (KnowLedge Base) хранятся фреймы и правила-продукции (база знаний). Этот файл создается пользователем с помощью специального редактора EdKB или вручную с помощью какого-либо стандартного текстового редактора (например, "Блокнот" или WordPad). Кроме файла *.klb в базу знаний может входить файл *.lvd, содержащий описания лингвистических переменных (см. раздел 6), если они используются в базе знаний. В файле с расширением *.dtb (DaTa Base) хранятся факты, полученные в процессе логического вывода (база данных). При начале работы с программной оболочкой наличие данного файла необязательно. Файл с базой данных создается программной оболочкой в процессе логического вывода.

При работе с программной оболочкой (после загрузки в виртуальную память базы знаний) фреймы и правила-продукции, находившиеся в файле с расширением *.klb, остаются неизменными. Факты, находившиеся в файле с расширением *.dtb, могут изменяться в процессе логического вывода (появляться, удаляться или менять свое значение в результате срабатывания правил-продукций или диалога с пользователем).

2.2.3. Фреймы

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

```
FRAME (⟨тип фрейма⟩) = ⟨имя фрейма⟩
  PARENT: ⟨имя фрейма-родителя⟩
  OWNER: <имя фрейма-владельца>
  ⟨имя слота 1⟩ (⟨тип слота⟩) [⟨вопрос слота⟩?]
    {⟨комментарий слота⟩}:
      (⟨значение 1⟩; ⟨значение 2⟩; ...; ⟨значение k⟩)
  ⟨имя слота 2⟩ (⟨тип слота⟩) [⟨вопрос слота⟩?]
    {⟨комментарий слота⟩}:
      (⟨значение 1⟩; ⟨значение 2⟩; ...; ⟨значение l⟩)
  ...
  ⟨имя слота n⟩ (⟨тип слота⟩) [⟨вопрос слота⟩?]
    {⟨комментарий слота⟩}:
      (⟨значение 1⟩; ⟨значение 2⟩; ...; ⟨значение m⟩)
ENDF
```

Фрейм может быть одного из трех типов: фрейм-класс, фрейм-шаблон, фрейм-экземпляр. В базе знаний содержатся фреймы-классы и фреймы-шаблоны. Фреймы-экземпляры создаются в базе фактов в процессе работы системы на основе информации из фрейма-класса и в базе знаний (постоянной части) отсутствуют. Однако, факты можно задавать при создании базы знаний в виде единственного значения слота во фреймах-классах.

Среди фреймов-классов выделяется специальный фрейм-класс "Цель", задающий перечень целей логического вывода (то есть наименований задач, решаемых экспертной системой).

База данных содержит только фреймы-экземпляры. Фрейм-экземпляр порождается фреймом-классом, при работе с которым появился факт (т.е. слоту из фрейма-класса было присвоено значение), и имеет то же имя, что и породивший его фрейм-класс. Но существует механизм, позволяющий создавать фреймы-экземпляры с именами, задаваемыми в процессе диалога и соответствующими значениям определенных слотов.

Имя фрейма, фрейма-родителя, фрейма-владельца, слота – последовательность символов (кириллические и/или латинские буквы, цифры, пробелы, знаки подчеркивания).

Если перед именем слота стоит символ *, это означает, что слот является множественным, т.е. слотов с таким именем во фрейме-экземпляре может быть несколько (неограниченное число) с разными значениями. При этом при про-

верке условия в правиле на основе этого факта, факт никогда не будет найден, а будет запрашиваться значение соответствующего слота.

Тип слота – символьный, численный или лингвистическая переменная. Обязательным является описание численного типа слота (описывается зарезервированным словом "численный") и лингвистической переменной (описывается зарезервированным словом "лп"). Слот без описания типа по умолчанию интерпретируется как символьный.

Символьный слот может принимать значение в виде цепочки символов как и имена фреймов и слотов. Численный слот может иметь целое или вещественное значение. Лингвистической переменной в описании правил может соответствовать то и другое значение.

Вопрос слота – любая последовательность символов, заключенная в квадратные скобки []. Он используется в процессе диалога с системой для задания системой вопроса пользователю. Вопрос слота не является обязательным. При отсутствии вопроса будет использована формулировка: "Выберите значение" или "Введите значение".

Комментарий слота – имя текстового (*.txt) или графического файла (*.gif, *.avi, *.htm), заключенное в фигурные скобки {}. Комментарий слота не является обязательным. Комментарий используется для пояснения вопроса, задаваемого системой пользователю. При этом, графический комментарий выводится автоматически вместе с вопросом, а текстовый (файл типа *.txt) может быть вызван при необходимости с помощью соответствующей кнопки.

Значение слота – любая последовательность символов. Несколько значений слота разделяются символом "точка с запятой" (;). Список значений слота необязателен. Слот фрейма-экземпляра имеет единственное значение, слот фрейма-класса и фрейма-шаблона имеет неограниченное число значений. Если значений слота несколько (в случае списка значений), список значений описывается в круглых скобках. Список значений слота имеет только одну функциональную нагрузку - во фрейме-классе он используется в процессе диалога в качестве меню возможных значений символьного слота или лингвистической переменной.

2.2.4. Правила-продукции

Правила-продукции описывают отношения между объектами, событиями, ситуациями и прочими понятиями. На основе проверки и задания отношений, задаваемых в правилах, выполняется логический вывод (решение выбранной задачи). При этом используется обратный логический вывод, т.е. правила используются начиная с того, которое приводит к цели (задает значение целевого слота). В условиях и заключениях правил присутствуют ссылки на фреймы и их слоты. Формат внешнего представления правил:

RULE <номер правила>
 <условие 1>


```

    <условие 2>
    ...
    <условие m>
DO
    <заключение 1>
    <заключение 2>
    ...
    <заключение n>
ENDR

```

Предполагается, что условия в посылке правила образуют конъюнкцию условий, заключения - последовательность заключений.

Номер правила – целое число. Начало и порядок нумерации правил произвольный, предпочтительнее правила нумеровать по порядку и начинать с единицы.

Формат записи условий и заключений одинаков и имеет следующий вид:
 <отношение> (<имя слота>; <значение слота>) <коэффициент достоверности>

Отношения в условиях могут быть:

```

EQ    |=   равно;
GT    |>   больше;
LT    |<   меньше;
NE| <> не равно;
IN     два фрейма связаны отношением "часть-целое" (имеется связь через слот
OWNER).

```

В случае символьных слотов может использоваться только условия EQ и NE. В случае лингвистической переменной – все арифметические отношения (EQ, GT, LT, NE). Если значением слота является слово "any", это означает, что данное условие всегда истинно и используется только для запроса значения слота.

Отношения в заключениях могут быть:

```

EQ    |=   равно (создание факта – слота во фрейме-экземпляре);
IN     включение во фрейм-владелец (создание связи – слота OWNER во
фрейме-экземпляре);
DL     удаление слота во фрейме-экземпляре;
EX     запуск внешней программы;
FR     вывод фрейма-экземпляра;
GO     запуск правила;
MS     вывод на экран сообщения;
GR     вывод на экран графического файла (форматов *.gif, *.avi или *.htm).

```

Имя слота в условии или в заключении может быть относительным или абсолютным. Относительное имя слота соответствует имени слота в некотором текущем фрейме. Абсолютное имя слота содержит имя фрейма и имя слота в нем, разделенные точкой. При использовании относительного имени имя фрейма берется системой из контекста.

Контекст включает в себя стек имен текущих фреймов. При начале решения задачи в него помещается имя фрейма "Цель". При начале интерпретации некоторого правила в него помещается имя фрейма, значение слота в котором правило пытается определить в цепочке логического вывода. При удачном или не удачном завершении интерпретации правила это имя удаляется из стека. Кроме того, при начале интерпретации условий имени фреймов, указываемые в именах слотов, задают контекст, в котором происходит дальнейшая интерпретация последовательности условий данного правила. При интерпретации заключений можно также менять имя текущего фрейма, находящегося в верхушке стека, задавая явно имена фреймов. Контекст (имя текущего фрейма) может меняться и при явном задании имени фрейма при адресации к слотам в арифметических выражениях.

Имя слота может совпадать с именем некоторого фрейма-класса (назовем его виртуальным фреймом), слот PARENT которого содержит имя данного фрейма. В этом случае при использовании данного слота в условии правила формируется фрейм-экземпляр с именем, задаваемым при вводе значения слота, и устанавливается ассоциация между именем слота и именем нового фрейма-экземпляра (введенным значением слота). С этого момента имя виртуального фрейма, используемое в правилах, заменяется каждый раз на имя, ассоциативно связанное с ним.

Пример связи слота с виртуальным фреймом:

```
Frame=Человек
    Имя [Фамилия И.О.?]:
EndF

Frame=Имя
    parent: Человек
    Начислено [Какую зарплату начислить (в руб.)?] (численный):
EndF
```

В этом примере можно использовать виртуальный фрейм "Имя" для создания фрейма-экземпляра с именем, совпадающим с введенной фамилией (и И.О.) при определении значения слота "Имя".

Значение слота – строка, число или арифметическое выражение (определяется типом слота). Если в качестве значения слота используется имя фрейма-шаблона, то в процессе логического вывода выполняется одновременное определение значений для всех слотов данного фрейма.

Арифметическое выражение начинается с символа # и конструируется из имен слотов (см. 4.5), четырех арифметических операций (+, -, *, /) и круглых скобок, задающих порядок выполнения операций. Если слот, встретившийся в арифметическом выражении, не имеет еще значения, это вызывает поиск его значения с использованием всех возможных средств оболочки (вызов других

правил, которые его определяют, задавание вопроса пользователю, формирование SQL-запроса к базе данных).

Пример правила с арифметическим выражением в заключении:

Rule 2

>(Доходы.Фонд зарплаты; 4999)

=(Человек.Имя; any)

Do

=(Расходы на зарплату.Начисленная зарплата; #Расходы на зарплату.Начисленная зарплата + Имя.Начислено) 60

FR(Фрейм; Расходы на зарплату)

EndR

В этом примере используется виртуальный фрейм "Имя", описанный выше (он создается при проверке условия "=", всегда истинного, т.к. в нем используется значение слота "any"), и вывод содержимого фрейма при срабатывании правила.

Коэффициент достоверности, это число от 0 до 100, имеющее смысл уверенности в процентах. Коэффициент достоверности в заключении используется при формировании значения слота фрейма-экземпляра при срабатывании правила. Коэффициент достоверности в условии используется в качестве порога – значения достоверности, ниже которого факты игнорируются при проверке данного условия. По умолчанию коэффициент достоверности в заключении принимает значение 100, в условии – 0.

2.2.5. Связь с внешними базами данных

База знаний может содержать специальную конструкцию SOURCE фреймоподобного типа:

SOURCE = <имя конструкции>

PARENT: <имя фрейма с описанием внешней базы данных>

<имя слота 1> [<арифметическое выражение>]: (<имя поля 1 в БД>

<имя слота 2> [<арифметическое выражение >]: (<имя поля 2 в БД>

...

<имя слота n> [<арифметическое выражение >]: (<имя поля n в БД>

ENDS

Конструкция SOURCE используется для связи базы знаний с какой-либо стандартной базой данных. На ее основе автоматически формируется SQL-запрос. В нем задается отображение структуры одноименного фрейма на поля базы знаний. Имя внешней базы данных определяется во фрейме, имеющем имя, совпадающее с именем базы данных (именем ALIAS). Слот PARENT конструкции SOURCE ссылается на фрейм, имеющий имя, совпадающее с именем базы данных. Другие слоты этого фрейма определяют имена полей таблицы.

Если слот, предназначенный для задания SQL-запроса, не имеет значения, в поле вопроса такого слота можно использовать вычисляемое выражение, которое и будет использоваться в качестве значения слота. Слоты фрейма-родителя SOURCE могут использоваться для описания базы данных. Единственный слот, имеющий функциональную нагрузку в интерфейсе с базой данных, имеет имя TABLE и содержит в качестве значения имя таблицы или запроса базы данных, являющегося источником данных.

Использование конструкции SOURCE и фрейма с описанием внешней базы данных позволяет в процессе логического вывода получать знания из внешней базы данных с помощью SQL-запроса.

Конструкция SOURCE должна удовлетворять следующим требованиям:

- 1) значение слота Parent должно соответствовать ALIAS в BDE для доступа к базе данных;
- 2) количество слотов должно соответствовать количеству слотов в одноименном с SOURCE фрейме и их имена тоже;
- 3) значения слотов соответствуют внутренним именам полей в базе данных (таблице), в качестве таблицы может выступать запрос в ACCESS;
- 4) если надо использовать вычисляемое поле, значение слота должно быть пусто, а в качестве вопроса к слоту (в квадратных скобках) пишется выражение для значения поля;
- 5) перечень слотов для описания структуры SOURCE и одноименного с ним фрейма выбирается так, чтобы он однозначно определял запись в таблице или запросе, к которому происходит обращение.

Кроме структуры SOURCE в БЗ должен быть фрейм с именем, соответствующим ALIAS, в котором должен быть обязательно слот с именем Table и значением-именем таблицы в базе данных (или запроса в ACCESS). Могут быть и другие слоты, описывающие базу данных и носящие информативный характер или участвующие в диалоге (на усмотрение автора БЗ).

Во время логического вывода при обращении к слоту, содержащемуся во фрейме, связанном с внешней базой данных (одноименном с конструкцией SOURCE), происходит открытие формы для автоматизированного формирования SQL-запроса и чтения всех слотов фрейма из базы данных. При этом предполагается, что SQL-запрос (а, следовательно, и его описывающая конструкция SOURCE) должен однозначно идентифицировать одну запись базы данных.

Пример конструкции SOURCE описания доступа к базе данных с помощью конструкции SOURCE:

```
Frame=Demo для ESWin
Parent:
Table: ("Запрос1")
СУБД: (Access 97)
База данных: (Борей)
EndF
```

```

Frame=Товар
Parent:
Тип:
Марка:
Цена(численный): ()
На складе(численный): ()
EndF

Source=Товар
Parent: Демо для ESWin
Тип: (Категория)
Марка: (Марка)
Цена(численный): (Цена)
На складе(численный): (НаСкладе)
EndS

```

Этот пример описывает доступ к демонстрационной базе данных БОРЕЙ, поставляемой вместе с Access в рамках Office'97 Professional. Здесь слоты "СУБД" и "База данных" носят информативный характер, хотя могут использоваться в правилах для проверки условий.

На рис. 2.4 показана форма, открывшаяся для доступа к базе данных БОРЕЙ. Она открывается, если произошло обращение к любому из слотов, описанных во фрейме "Товар", и этот слот в данный момент еще не определен. При этом предполагается, что "марка" однозначно идентифицирует товар, т.е. запись в запросе "Запрос1".

The screenshot shows a Windows-style dialog box titled "Загрузка фрейма из базы данных". It is divided into two main panes. The left pane, titled "Товар", contains a label "Выберите значение параметра" above a text box labeled "Марка". Below this is a list box titled "из списка значений" containing the following items: Guarana Fantastica, Laughing Lumberjack Lager, Outback Lager, Rhonbrau Klosterbier, Lakkalikoori, Sasquatch Ale, Steeleye Stout, Cote de Blaye, Chartreuse verte, Ipoh Coffee, Chai, and Chang. The right pane contains four labels: "Тип : = Напитки", "Марка :", "Цена :", and "На складе :". At the bottom of the dialog, there is a text box containing the SQL query: "SELECT 'Марка' FROM 'Запрос1' WHERE 'Категория'='Напитки'". Below the query box are four buttons: "Читать параметры из БД", "Прочитано:" followed by a text box containing "12", "ОК", "Отмена", and "Заккрыть".

Рис. 2.4. Форма для доступа к внешней базе данных

Следует иметь в виду, что этот механизм доступа к базам данных, реализованный в настоящий момент в пакете ESWin, не поддерживает создание SQL-запроса на основе нескольких таблиц. Если это необходимо, такая возможность должна обеспечиваться средствами самой СУБД. В данном примере источником фактов является запрос "Запрос1", созданный предварительно в СУБД Access.

2.2.6. Лингвистические переменные

При формировании базы знаний для описания нечетких понятий используются лингвистические переменные в качестве слотов. Лингвистическая переменная позволяет при логическом выводе задавать как символьное, так и численное значение слота.

Лингвистическая переменная имеет одно или несколько символьных значений. Каждому символьному значению поставлена в соответствие функция принадлежности, которая определяет отношение между численным значением лингвистической переменной и коэффициентом достоверности для данного численного значения (соответствующего символьному значению). Для каждого символьного значения лингвистической переменной существует собственная функция принадлежности. Функция принадлежности определяется на отрезке метрической шкалы, одним и тем же для всех символьных значений лингвистической переменной.

Описание лингвистических переменных хранится в текстовом файле (*.lvd – Linguistic Variable Description). Первая часть имени файла должна соответствовать именам файлов, содержащих базу знаний и базу данных (*.klb и *.dtb). Формат внешнего представления лингвистической переменной:

```

<число лингвистических переменных>
<имя лингвистической переменной 1>
<нижнее значение границы метрической шкалы>
<верхнее значение границы метрической шкалы>
<шаг метрической шкалы>
<число символьных значений лингвистической переменной 1>
<символьное значение 1>
    <значение функции принадлежности 1>
    <значение функции принадлежности 2>
    ...
    <значение функции принадлежности m>
<символьное значение 2>
    <значение функции принадлежности 1>
    <значение функции принадлежности 2>
    ...
    <значение функции принадлежности m>
...
```

«символьное значение n»
 «значение функции принадлежности 1»
 «значение функции принадлежности 2»
 ...
 «значение функции принадлежности m»
...

2.2.7. Интерпретация правил-продукций

Решение задачи с помощью оболочки ESWin и загруженной в нее базы знаний начинается с выбора цели логического вывода или задачи. В качестве цели логического вывода используется один из целевых слотов, содержащихся во фрейме-классе со специальным именем «Цель».

Для решения задачи в системе ESWin используется обратный логический вывод. Он начинается с поиска правила, в заключении которого присутствует выбранный целевой слот.

После нахождения правила начинается его интерпретация (перебор и проверка условий). При проверке условия ищется (определяется) значение указанного в условии слота.

Определение значения слота производится в следующем порядке:

- 1) ищется факт со значением этого слота в базе фактов;
- 2) ищется значение слота в одноименном фрейме-классе (если это значение в нем единственное);
- 3) ищется правило, задающее при срабатывании значение данного слота, и запускается его интерпретация;
- 4) ищется одноименная структура SOURCE для формирования SQL-запроса к внешней базе данных;
- 5) ищется слот во фрейме-классе с описанием всей необходимой информации для запроса значения слота у пользователя.

К каждому из этих шагов оболочка переходит в случае неудачи предыдущего шага.

При вводе пользователем значения слота лингвистического типа, формируется численное значение с коэффициентом достоверности равным 100, если пользователь ввел число. Если пользователь выбрал символьное значение, формируется также численное значение, равное значению на шкале лингвистической переменной с максимальным значением коэффициента достоверности для данного символьного значения. Если значение слота в правиле было символьным, а пользователем было введено численное значение, то коэффициент достоверности формируется как значение функции принадлежности лингвистической переменной при заданном символьном и численном значениях.

Следует иметь в виду, что для проведения вычислений над коэффициентами достоверности они переводятся к нормированному виду в диапазоне от 0 до 1.

Коэффициент достоверности набора (последовательности) условий вычисляется как коэффициент достоверности конъюнкции в нечеткой логике Л. Заде (минимальное значение из значений коэффициентов достоверности условий).

Коэффициент достоверности слота фрейма-экземпляра, формируемого на основе заключения, вычисляется как произведение коэффициента достоверности набора условий и коэффициента достоверности заключения. Если такой слот во фрейме-экземпляре уже есть, то его коэффициент достоверности меняется на новое значение, вычисляемое по формуле:

$$\text{КД}_{\text{результатирующий}} = \text{КД}_{\text{исходного слота}} + \text{КД}_{\text{набора условий}} * (1 - \text{КД}_{\text{исходного слота}})$$

При проверке условия в правиле в случае, если коэффициент достоверности меньше определенной пороговой величины, заданной в условии, то условие считается не выполненным.

На рис. 2.5 показаны взаимосвязи между разными методами представления знаний и источниками данных, реализованные в оболочке ESWin.

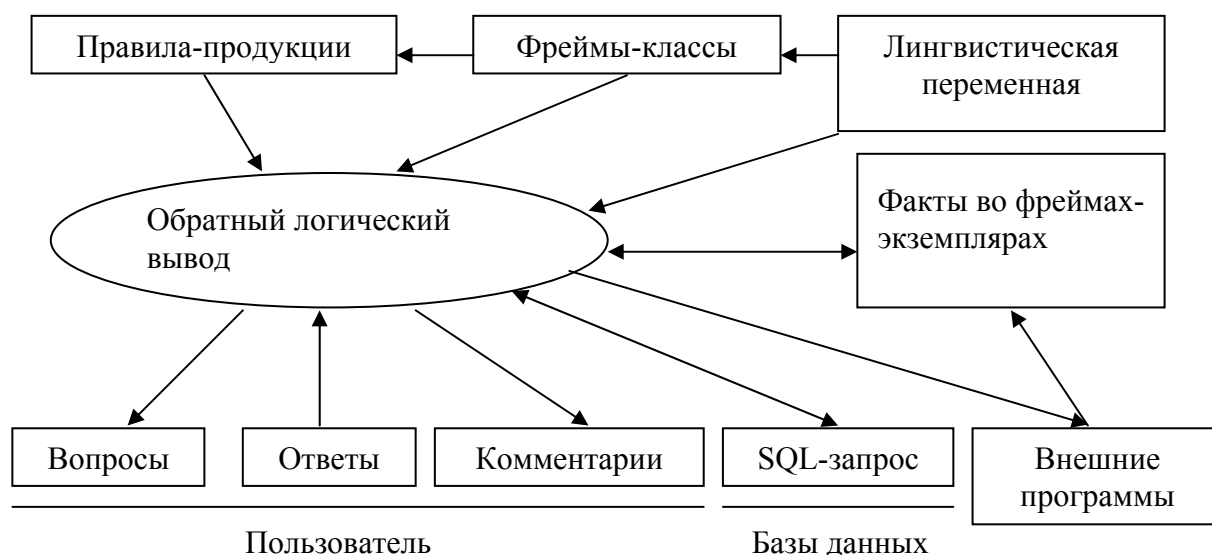


Рис. 2.5. Взаимосвязи между разными методами представления знаний в оболочке ESWin

Общий вид основного окна программной оболочки ESWin представлен на рис. 2.6.

Ниже приведены вид главного меню основной формы редактора EdKB (рис. 2.7), форма для редактирования фрейма (рис. 2.8), форма для редактирования лингвистической переменной (рис. 2.9).

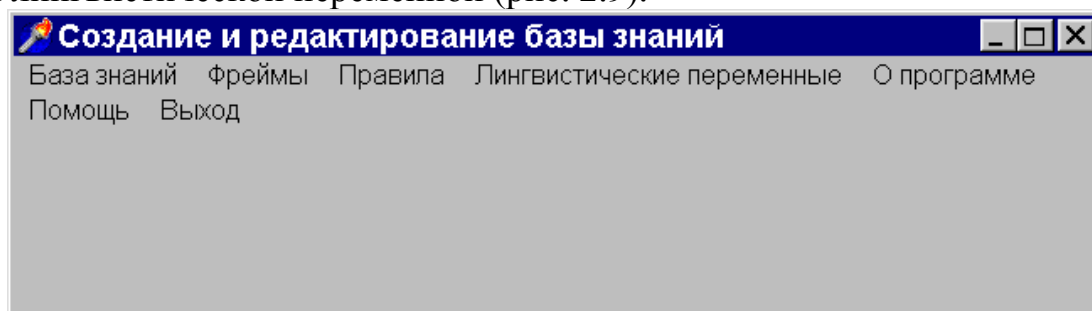


Рис. 2.6. Основное окно с результатами логического вывода

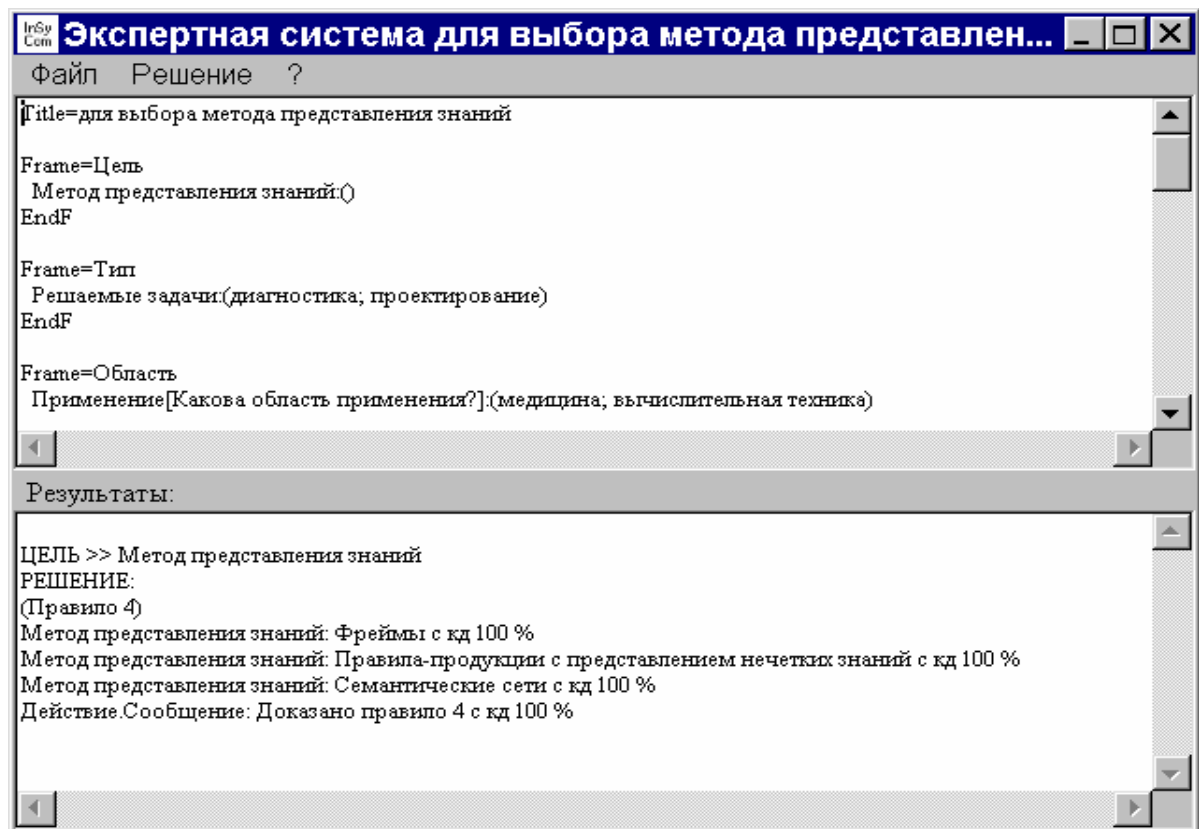


Рис. 2.7. Основное меню редактора баз знаний EdKB

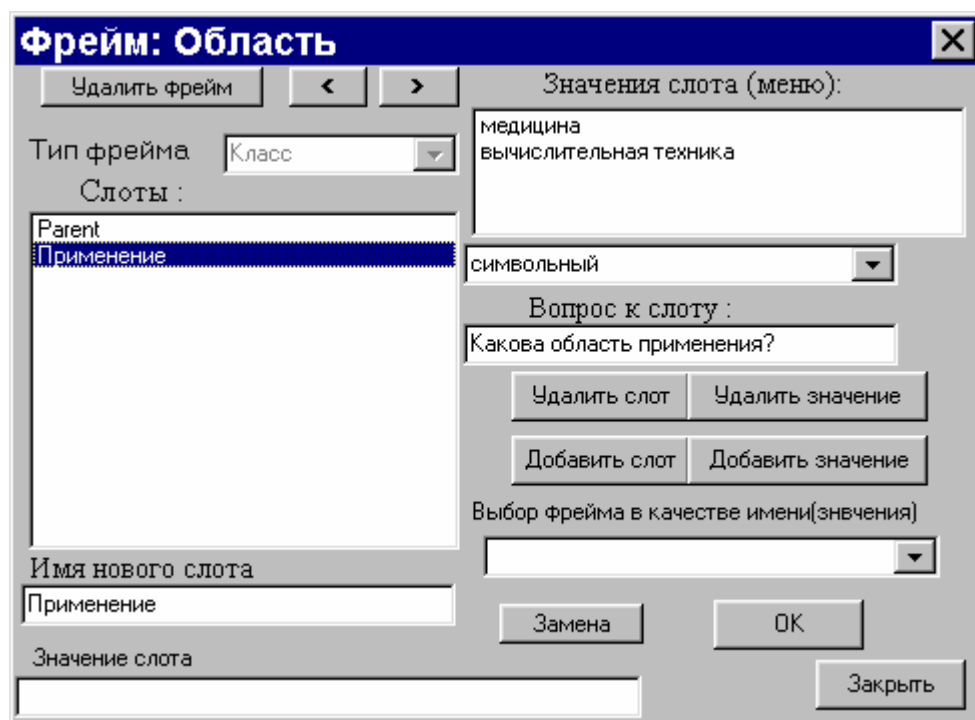


Рис. 2.8. Форма для редактирования фрейма

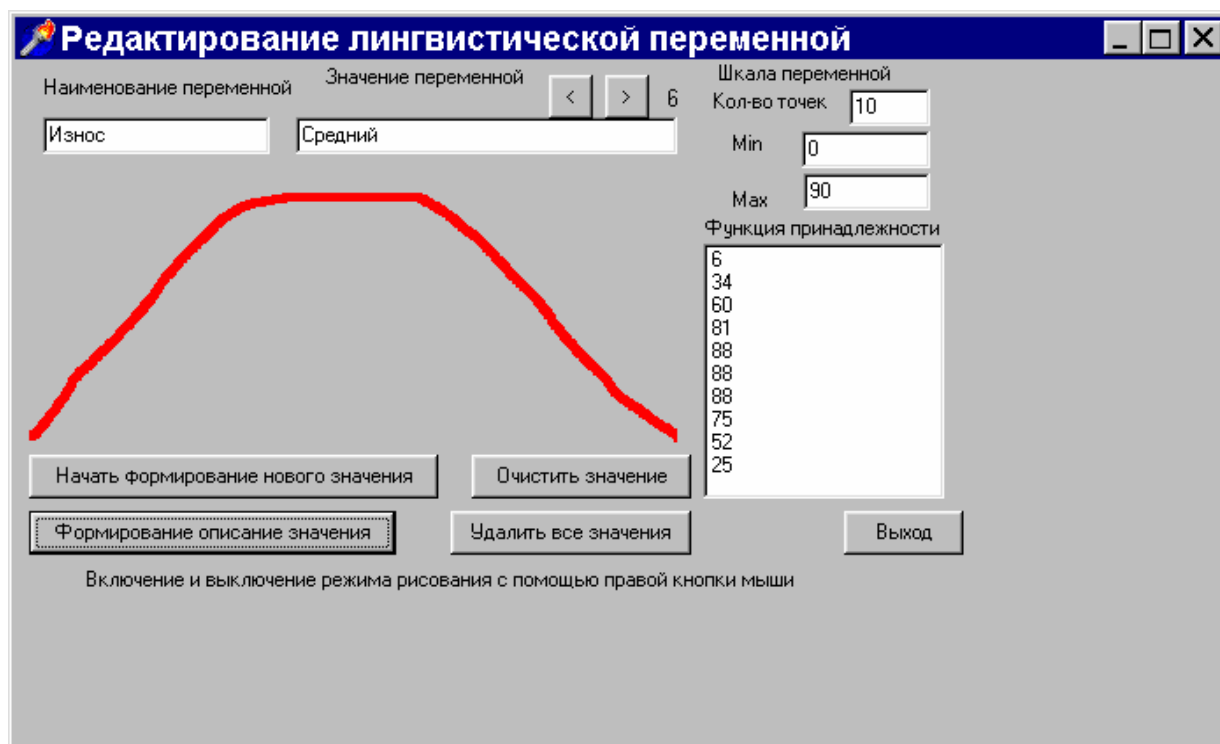


Рис. 2.9. Форма для редактирования лингвистической переменной

2.3. Выводы

В главе рассмотрены разные комбинации методов представления знаний в пределах инженерии знаний, т.е. всех методов представления кроме нейронных сетей, предложена и реализована архитектура гибридных экспертных систем.

В предложенной архитектуре реализованы следующие методы представления знаний:

- 1) правила-продукции с нечетким обратным логическим выводом;
- 2) фреймы;
- 3) лингвистические переменные в качестве одного из типов слотов фрейма.

В качестве источников данных (фактов в процессе обратного логического вывода) в этой архитектуре используются:

- 1) пользователь,
- 2) базы данных,
- 3) внешние программы.

В разработанное в соответствии с данной архитектурой ПО входит экспертная оболочка ESWin (сейчас разработана версия 2.0), редактор баз знаний EDKB, программа для просмотра и диагностики целостности баз знаний KBView и программа для оптимизации баз знаний KBOptim.

Экспертная оболочка ESWin (базовая версия 1.0) была реализована Ю.В. Новицкой, программа KBView – Г. Ладыгиным, KBOptim – студентами, которым автор выражает глубокую признательность.

Предложенная архитектура апробирована при построении ряда учебных и исследовательских экспертных систем на факультете АВТФ НГТУ (на кафедрах вычислительной техники и автоматизированных систем управления) и в Институте реабилитации при НГТУ, в частности, экспертной системы для формирования набора моделей для прогнозирования одномерных временных рядов в рамках метода вариативного моделирования [80]. Кроме того, ПО ESWin используется при разработке экспертной системы для диагностики и ликвидации аварийных ситуаций в энергосистеме по заказу ОАО "Новосибирскэнерго". В приложении приведены примеры демонстрационных баз знаний.

В настоящее время ведется работа по развитию предложенной архитектуры в направлении:

- 1) включения в нее процедурных знаний в виде процедур (последовательностей операторов-шагов) с возможностью программирования в них арифметической обработки фреймов-экземпляров;
- 2) добавления возможности присоединения процедур и подмножеств правил-продукций к фреймам и привязки их к событиям, связанным с фреймом;
- 3) включения в нее средств взаимодействия с нейронными сетями в рамках концепции "двухполушарных" интеллектуальных систем (см. 3.6).

3. ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ В ИСКУССТВЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

3.1. Варианты использования нейронных сетей в современных интеллектуальных системах

В настоящее время можно выделить следующие варианты использования нейронных сетей при построении интеллектуальных систем:

- 1) встраивание нейронной модели в прикладное программное обеспечение как инструмента для обработки событий, сигналов;
- 2) использование нейронной сети как основы для построения экспертной советующей системы без использования в системе обычных средств, моделирующих рассуждения (мышления на вербальном уровне);
- 3) использование нейросетевых моделей в системах обработки и поиска текстовой информации, сводящейся, как правило, к решению задач классификации или кластеризации;
- 4) использование нейросетевых моделей в системах обработки данных (Data Mining) для извлечения данных и знаний, а также, для прогнозирования;
- 5) использование нейронных сетей для распознавания и синтеза речи;
- 6) использование нейросетевых алгоритмов для повышения возможностей экспертных систем, основанных на традиционной архитектуре;
- 7) совместное использование нейронных сетей и традиционных методов представления и обработки знаний в одной интеллектуальной системе.

Вариант 1 обычно реализуется в системах управления технологическими процессами [1,2] и в соответствующем инструментальном программном обеспечении для автоматизации их построения (например, библиотека компонентов для программирования на Delphi NeuralBase [3]), а также, в системах обработки изображений [4, 5] и геофизической информации [6, 7].

Вариант 2 реализуется в виде экспертных систем в области медицинской и технической диагностики [8] и поддерживается множеством существующих инструментальных оболочек для работы с нейронными сетями, таких как Brain-Maker, NeuroShell, NeuroSolution, NEUFrame, Ecanse, Neural Bench (см. таблицу 0.2 в приложении 1).

Вариант 3 реализуется в системах для распознавания рукописных текстов [9], классификации и рубрикации текстов [10, 11], поиска информации по смыслу [12].

Вариант 4 получил в последнее время широкое распространение в рамках систем Data mining и KDD [13-18 и см. раздел 3.5], а также, в программном обеспечении для решения задачи прогнозирования [19, 20], в частности, в финансовом анализе.

С вариантом 5 [21] связан бум нейроинформатики 80-х годов. Этот вариант применения похож на вариант 1. Только на выходе нейронной сети информация визуализируется в вербальном виде.

Вариант 6 реализуется в виде либо экспертной системы с традиционной архитектурой, дополненной нейросетевыми алгоритмами интерпретации правил-продукций ([22] и см. раздел 3.4) либо в программе поиска документов по смыслу в виде алгоритмов взвешенного суммирования, используемых для оценки семантической близости понятий-узлов семантической сети ([23] и см. раздел 4.5).

Вариант 7, по-видимому, является самым перспективным для использования нейронных сетей в интеллектуальных системах, т.к. позволяет совместить возможности нейронных сетей по ассоциативному запоминанию-вспоминанию информации и большие наработки в области создания логистических интеллектуальных систем (систем, основанных на знаниях) с их хорошими возможностями визуализации и вербализации информации. Одним из направлений развития этого варианта является создание "двухполушарных" интеллектуальных систем (см. раздел 3.6), архитектура которых была предложена автором впервые в 1989 году [24]. Этот вариант применения нейронных сетей в архитектуре интеллектуальных систем является, по-видимому, наиболее адекватным тому, как устроены естественные интеллектуальные системы, и соответствует моделям, предложенным в главе 1.

3.2. Обработка символьной информации в нейронных сетях [25]

При практическом использовании нейронных сетей при построении экспертных систем [8], для обработки текстовой информации [9-12] или для анализа баз данных [13-18] одной из трудностей является ориентация нейронных сетей на обработку сигналов, а не символьной информации. В то же время в выше перечисленных примерах нейронной сети требуется обрабатывать символьную информацию и в качестве результата обработки предъявлять пользователю так же символьную информацию. Во многих моделях искусственных нейронных сетей (ИНС) входная информация представляется в виде двоичного вектора. Таким образом, при использовании таких моделей необходимо решать задачу кодирования входной информации ИНС и декодирования выходного вектора ИНС.

Если кодировать символьную информацию на входе ИНС "беспорядочно", т.е. не заботиться о корреляции между значениями двоичных векторов и соответствующими им символьными значениями, то близкие по семантике символьные значения могут кодироваться совершенно разными двоичными векторами, отстоящими друг от друга на очень большое расстояние в пространстве состояний нейронной сети. Это затрудняет обучение нейронной сети и может приводить к ошибкам при функционировании обученной ИНС. К такому же эффекту могут приводить и орфографические ошибки во входной информации, когда искаженное слово воспринимается как новое, а также, выход из строя нейроподобных элементов в случае аппаратной реализации ИНС. Кроме того, при использовании неполносвязных моделей ИНС, где ненулевая вероятность ошибки является особенностью архитектуры ИНС, желательно свести к мини-

муму эффект этой ошибки. Например, наверное, допустимо, если вместо решения "старый" на выходе ИНС появится семантически близкое значение "пожилой", но совершенно не допустимо, если ИНС сформирует решение "молодой".

Для исключения этих недостатков при использовании нейронных сетей для обработки символьной информации предлагается использовать следующие принципы:

- разбиение входного вектора на подвекторы, кодирующие разные компоненты символьной информации, поступающей на нейронную сеть (например, разные поля реляционной базы данных или разные аспекты контекста обрабатываемого нейронной сетью текста), при этом для кодирования подвекторов необходимо использовать тезаурусы с фиксированным количеством слов в каждом из них;
- использование представления лингвистической переменной для кодирования семантически близких значений, которые могут быть связаны с метрической шкалой;
- использование классификации понятий и определение семантических шкал для них (задание отношений частичного порядка на множестве понятий, семантически близких в определенном контексте, задаваемом классом и признаком классификации).

При кодировании символьной информации на входе ИНС необходимо использовать фиксированный тезаурус, свой для каждого подвектора входного вектора ИНС. Конечно, можно кодировать входные слова как произвольные последовательности символов. В этом случае набор используемых слов ничем не ограничен, и в процессе функционирования системы могут появляться новые, ранее не использованные слова. Но в этом случае они могут восприниматься нейронной сетью только как сигналы, и ни о каком использовании семантической близости понятий при работе ИНС не может быть речи. Вся тяжесть построения разделяющей гиперповерхности для трудно разделяемых входных векторов в пространстве признаков ложится в этом случае на нейронную сеть. К тому же, в этом случае теряются основания для разбиения входного вектора на подвекторы, т.к. заранее не известны их длины.

При использовании значений лингвистической переменной (ЛП) в качестве входной информации для нейронной сети с бинарными входами целесообразно кодировать значения ЛП так, чтобы расстояние между максимальными значениями функции принадлежности на метрической шкале взаимно однозначно соответствовало расстоянию Хэмминга между соответствующими двоичными входными векторами ИНС, и отношение частичного порядка на множестве этих максимальных значений сохранялось на множестве соответствующих расстояний Хэмминга. В этом случае можно предположить, что вероятность ошибок при распознавании значений ЛП будет минимальной. Естественно, что сохранение семантической близости двоичных векторов при таком кодировании приводит к избыточности разрядов. При кодировании "в лоб" достаточно $\text{int}(\log_2 n)$ двоичных разрядов, где n – количество значений ЛП, int – ок-

ругление до большего целого. При кодировании с сохранением семантической близости при строгом подходе требуется $(n - 1)$ двоичных разрядов.

Например, пусть нейронная сеть должна обрабатывать значения ЛП "возраст", принимающую значения "дитя", "ребенок", "юный", "молодой", "зрелый", "пожилой", "старый", "очень старый". Для кодирования "в лоб" (порядке перечисления слов) достаточно 8 двоичных разрядов и код для "дитя" будет 000, а для "зрелый" – 100. Если нейронная сеть ошибется в одном (2-м) разряде, это приведет к тому, что вместо "дитя" мы получим "зрелый" или наоборот. При кодировании с сохранением семантической близости можно использовать следующие коды:

"дитя"	– 0000000;
"ребенок"	– 0000001;
"юный"	– 0000011;
"молодой"	– 0000111;
"зрелый"	– 0001111;
"пожилой"	– 0011111;
"старый"	– 0111111;
"очень старый"	– 1111111.

Алгоритм кодирования, используемый здесь, очевиден.

Можно уменьшить избыточность, сняв требование строгого соответствия между расстоянием Хэмминга и расстоянием на метрической шкале:

"дитя"	– 00000;
"ребенок"	– 00001;
"юный"	– 00011;
"молодой"	– 00111;
"зрелый"	– 01111;
"пожилой"	– 11111;
"старый"	– 11110;
"очень старый"	– 11100.

При этом способе кодирования сначала расстояние Хэмминга от первого значения до текущего кодируемого растет, а с некоторого значения начинает падать. Уровень избыточности можно задавать ограничением на расстояние Хэмминга между крайними на шкале значениями ЛП. Любопытно, что в этом случае расстояние между значениями "дитя" и "очень старый" меньше, чем между "дитя" и "зрелый".

В случае использования в качестве входной информации произвольных символьных значений, которые не возможно представить в виде значений лингвистической переменной, можно использовать разбиение их на классы и определение для каждого класса своей семантической шкалы в контексте признака классификации и, может быть, признака, по которому оценивается семантическая близость между представителями заданного класса. На семантической шкале определяется отношение частичного порядка между значениями, принадлежащими данному классу, и семантическое расстояние между двумя зна-

чениями, равное количеству значений, находящихся между ними на шкале, увеличенному на 1. Например, класс "мебель" можно представить следующими значениями в порядке их расположения на семантической шкале: "кровать", "диван", "кресло", "стул", "журнальный столик", "письменный стол", "обеденный стол", "кухонный стол", "буфет", "шкаф". Семантическое расстояние между понятиями "кровать" и "диван" равно 1, между "кровать" и "стул" – 3, между "кровать" и "обеденный стол" – 6. Наименование класса и признака классификации кодируются отдельно. В этом случае классификация может производиться другой нейронной сетью, что может быть реализовано в известных ансамблевых моделях ИНС [26].

На выходе нейронной сети полученный в результате работы ИНС двоичный вектор необходимо декодировать, т.е. преобразовать его в одно или несколько символьных значений. При этом также можно использовать его разбиение на подвекторы, каждому из которых соответствует компонент решения со своим множеством возможных символьных значений.

При декодировании выходного вектора надо учитывать одно из возможных требований, которые могут предъявляться к решению, получаемому нейронной сетью:

- исключить или уменьшить вероятность ложного (неверного) решения при сохранении возможности не получить никакого;
- исключить отсутствие какого-либо решения, может быть, в ущерб качеству, при этом предполагается, что отсутствие ошибки гарантируется качеством обучения и особенностями архитектуры нейронной сети.

В первом случае необходимо вводить избыточность в кодирование значений символьных решений и двоичные вектора, не соответствующие кодам значений из тезауруса решений, не декодировать (случай "отсутствия решения").

Во втором случае избыточность при кодировании не требуется, а если она используется (например, для обеспечения симметричности методов кодирования на входе и выходе сети), в случае несоответствия выходного вектора какому-либо из значений тезауруса выбирается и декодируется ближайший (по Хэммингу) код.

3.3. Модель нейронной сети "ключ-порог" [27-30]

Эта архитектура была предложена автором впервые в 1990 году [27]. Она предназначена для варианта 1 применения нейронных сетей (см. 3.1). Отличие этой модели нейронной сети от других заключается в том, что нейрон в ней выполняет функцию простого перцептрона, т.е. распознает определенный бинарный вектор с использованием расстояния Хэмминга. Нейрон в этой модели функционирует по следующему правилу:

$$y_i = \begin{cases} 1, & \text{если } f(a, k) > h_i \\ 0 & \text{в противном случае} \end{cases}$$

где: $k = (k_{ij})$, $i = 1, N$, $j = 1, M$ – вектор-ключ i -го нейрона;

$a = (a_{ij})$, $i = 1, N$, $j = 1, M$ – входной вектор i -го нейрона;

$f(a, k)$ – функция равная количеству равных разрядов в векторах a и k .

Нейроны связаны между собой так же как в многослойном перцептроне с полными или неполными связями между слоями.

Обучение нейронной сети обеспечивается наличием у каждого нейрона входа ϕ , сигнал на котором имеет смысл наказания, и запускает процесс изменения состояния нейрона, т.е. его порога h или ключа k .

Модель испытывалась на примере решения задачи обучения избеганию столкновений с движущимся объектом в ограниченном пространстве. В этой задаче робот R, управляемый моделью нейронной сети, и объект O моделируются точками на замкнутом поле, из 8×8 точек (рис. 3.1).

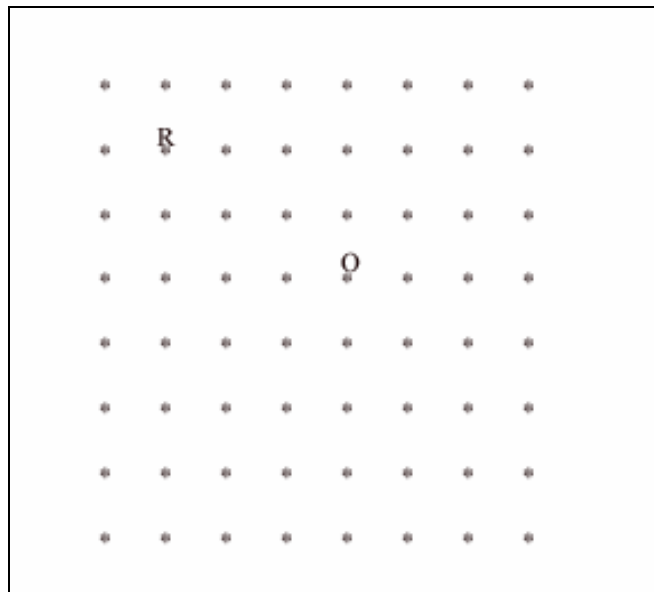


Рис. 3.1. Модель пространства для экспериментов с моделью нейронной сети "ключ-порог"

Робот может двигаться на шаг в одном из четырех направлений. При этом движение с 1 уровня вверх означает перемещение в соответствующую точку последнего уровня и наоборот, а движение с крайнего левого столбца влево – перемещение в соответствующую точку крайнего правого столбца и наоборот. По такому же принципу перемещается по полю объект. Он может перемещаться прямолинейно, делая за один такт перемещение на один шаг, в одном из трех направлений – вертикально вверх, горизонтально вправо и по диагонали слева направо и снизу вверх. Каждый нейрон сети имеет четыре входа. Нейронная сеть состоит из трех слоев – рецепторных нейронов, связанных по входам с "рецепторами", 4-х моторных нейронов, управляющими движением робота на один шаг в одном из четырех направлений, и промежуточных (скрытых) нейронов. В качестве "рецепторов" выступают 4 признака, определяющих направление, в котором находится объект по отношению к роботу, и принимающих значения 1 или 0 (модель L). Только один или два из них могут одновременно иметь значение 1. Кроме того, имеется "рецептор", имеющий смысл неудовле-

творенности (боли и т.п.), принимающий значение 1, если произошло столкновение робота с объектом (он оказался в непосредственной близости от него, другими словами в соседней точке пространства). Входы наказания нейронов могут быть связаны с этим рецептором или с выходом какого-либо другого нейрона.

Моделирование проводилось с разбиением времени моделирования на интервалы равной длительности. Если на текущем интервале не было столкновений, считалось, что робот обучился избегать столкновений с объектом, и объект в этом случае менял направление своего движения.

В качестве критерия для сравнения результатов экспериментов была выбрана частотность столкновений, среднее расстояние между роботом и объектом, а также, количество изменений направления перемещения объекта (адаптации робота).

Кроме выше описанной модели эксперименты проводились над ее разновидностью G, в которой в качестве рецепторов выступали признаки нахождения объекта в определенной точке пространства (64 признака) (эту модель можно назвать моделью с глобальным видением в отличие от локального в модели L) и над случайной моделью R, в которой робот принимал решения о шаге перемещения случайно вне зависимости от положения объекта (модель со "слепым роботом").

В табл. 3.1 приведены некоторые результаты экспериментов.

Частотность столкновений подсчитывалась только с учетом тех интервалов, на которых столкновения были (время пребывания на интервалах, где предположительно робот адаптировался к текущей траектории объекта, не учитывалось) по формуле:

$$P = \frac{1}{NT} \sum_{i=1}^N K_i,$$

где T – величина интервала в тактах;

K_i – количество столкновений за i -й интервал;

N – количество интервалов, на которых были столкновения

По результатам экспериментов с предложенной моделью были сделаны следующие выводы:

1) при соответствующем выборе количества рецепторных и скрытых нейронов система методом проб и ошибок находит траекторию движения, не пересекающуюся с движущимся объектом;

2) обучение не приводит к "устойчивой" траектории при большом количестве нейронов, возможны столкновения с объектом, хотя и редкие, т.е. подтверждается принцип "для решения простой задачи необходима простая нейронная сеть" (можно сказать, что при большом количестве нейронов сеть демонстрирует излишнюю "неуверенность в принятом решении" или "любопытность");

3) необходимо разработать алгоритм обучения, гарантирующий сходимость процесса обучения (может быть, при задании некоторых ограничений на

структуру связей между нейронами – задании регулярности структуры нейронной сети).

Таблица 3.1

Время моделирования – 1000 тактов,
Интервал моделирования – 100 тактов.

Мо- дель	Количество нейронов	Количество рецепторных нейронов	Доля наказа- ний от других нейронов	Частотность столкнове- ний	Среднее расстоя- ние	Количе- ство адаптаций
L	100	20	10	0.052		6
L	50	40	10	0.043		5
L	50	30	10	0.042		6
L	50	20	10	0.072		4
L	50	20	50	0.120		4
L	50	20	90	0.142		0
L	70	20	10	0.034		5
L	100	20	10	0.073		4
L	50	16	10	0.070		9
L	50	16	10	0.085		6
L	100	20	50	0.136		0
L	100	20	90	0.118		0
L	50	20	70	0.095		0
L	100	20	30	0.149		0
L	500	20	10	0.07		2
L	200	20	10	0.103		6
L	500	20	10	0.095		4
L	1000	20	10	0.18		2
G	100	20	10	0.051		3
G	100	48	10	0.044		4
G	100	48	90	0.124		1
G	100	30	10	0.097		2
G	70	20	10	0.080		3
G	50	20	0	0.044		4
G	50	20	10	0.025		4
G	50	30	90	0.140		0
G	50	40	90	0.089		1
R				0.096		0
R				0.046		0
R				0.080		0
R				0.082		0
R				0.071		0
R				0.080		0

3.4. Гибридная экспертная система для профориентации [22]

3.4.1. Введение

В разделе описана архитектура гибридной экспертной системы для профориентации для применения в центре занятости населения, разработанная автором в 1993-1994 годах. Нейронные сети в ней используются в соответствии с вариантом 6 (см. 3.1).

Система предназначена для решения двух основных задач:

- 1) выбора подходящей профессии;
- 2) диагностики пригодности к выбранной профессии.

Эти две задачи решаются в диалоге с клиентом-посетителем центра занятости населения. Но в дальнейшем клиент имеет возможность обратиться к психологу для разъяснений, если диалог с системой по каким-либо причинам его не удовлетворил.

Решение первой задачи предполагается для клиента, не имеющего определенного представления о том, какой профессией он может или хочет заниматься. Таким клиентом может быть, например, выпускник школы или человек, потерявший работу, и который готов переквалифицироваться на другой вид деятельности.

Вторая задача решается в случае, если клиент выбрал для себя одну или несколько профессий и хочет получить оценку своих возможностей при попытке заняться ими.

3.4.2. Структура экспертной системы

Экспертная система для профориентации представляет собой комплекс из четырех основных программ, программ для тестирования психофизиологических характеристик, набора модулей базы знаний, а также набора сервисных программ-утилит.

В состав основных программ входят следующие:

- программа PROF, ориентированная на диалог с клиентом и предназначенная для экспресс-консультирования его по вопросам профориентации и профдиагностики;
- программа PROFIN отличается от PROF отсутствием пользовательского интерфейса и может запускаться с передачей параметров из какой-либо другой программы, в частности, из банка данных о профессиях;
- программа PROFEXOR, ориентированная на специалиста по профориентации, позволяет просматривать и анализировать базы знаний, используемые в PROF и PROFIN, и пополнять их;
- программа EXTRACT, ориентированная на инженера по знаниям (достаточно квалифицированного разработчика баз знаний), более универсальна чем PROFEXOR, позволяет более глубоко анализировать базы знаний, содержащие-

ся в произвольных файлах, запускать их для тестирования и имеет более мощные средства редактирования баз знаний чем PROFEXOR.

Программа EXTRACT является, по существу, универсальной оболочкой экспертных систем, а PROFEXOR – специализированной оболочкой для повышения удобства использования в рамках системы PROFEX.

В состав программ-утилит входят:

- программа сравнения содержимого базы знаний и базы данных о профессиях COMPKD;
- программа манипулирования модулями, из которых состоит база знаний EDMODKB.

Программы тестирования психо-физиологических характеристик запускаются из базы знаний и включают в себя следующие тесты:

- оперативной памяти;
- концентрации и устойчивости внимания, переключения внимания;
- объема и распределения внимания;
- зрительно-моторной координации движений;
- умения читать чертежи, схемы.

База знаний содержит два главных модуля, соответственно для решения задач профессиональной диагностики и выбора профессии. Первый из них содержит правила:

- для тестирования типов профессий, подходящих для клиента, в основу которого положен дифференциально-диагностический опросник (ДДО);
- для тестирования некоторых психологических характеристик, в основу которого положен тест Кэттелла.

Опросник ДДО в процессе диалога определяет следующие пять типов профессий: "человек – человек" (например, преподаватель, тренер), "человек – знаковая система" (например, программист, бухгалтер), "человек – художественный образ" (например, музыкант, художник, маляр), "человек – техника" (например, токарь, шофер, диспетчер) и "человек-природа" (например, егерь, ветеринар).

Тест Кэттелла в процессе диалога определяет следующие психологические характеристики личности: переоценивание себя, адекватность самооценки, сообразительность, конкретно-логический тип мышления, общительность, эмоциональная уравновешенность, самоуверенность, решительность, положительная установка к деятельности, импульсивность, подвижность, ответственность, принципиальность, аккуратность, социальная смелость, социальная активность, мягкость, готовность помочь, артистичность, женственность, эгоцентризм, мнительность, ревнивость, самостоятельное творческое решение, развитое воображение, проницательность, дипломатичность, депрессивность, впечатлительность, ранимость, тревожность, гибкость, адаптивность поведения, склонность к критике, независимость в суждениях, самоконтроль, целенаправленность, исполнительность, повышенная мотивация, напряженность, беспокойство, тактичность, безропотность, зависимость, застенчивость, абстрактно-образный тип

мышления, утомляемость, раздражительность, подверженность чувствам, ипохондрия, замкнутость, холодность, осторожность, пессимизм, беспринципность, непостоянство, неорганизованность, суровость, жесткость, мужественность, практичность, сдержанность, робость, откровенность, доверчивость, благожелательность, исполнительность, наивность, прямолинейность, хладнокровие, безмятежность, спокойствие, консервативность, склонность к морализации, конформизм, недисциплинированность, вялость, низкая мотивация, невозмутимость.

Второй главный модуль содержит правила:

- реализующие тест ДДО, как и в первом модуле;
- реализующие тест Кэттела, как и в первом модуле;
- реализующие тест "Карта интересов".

В каждой из этих главных модулей есть ссылки на другие модули, содержащие знания о профессиях в виде правил-продукций.

Правила для профдиагностики по назначению делятся на три вида – для определения:

- пригодности к данной профессии;
- причины непригодности;
- родственных, рекомендуемых для дальнейшего тестирования профессий.

Структурная схема решения задачи профессиональной диагностики показана на рисунке 3.2.

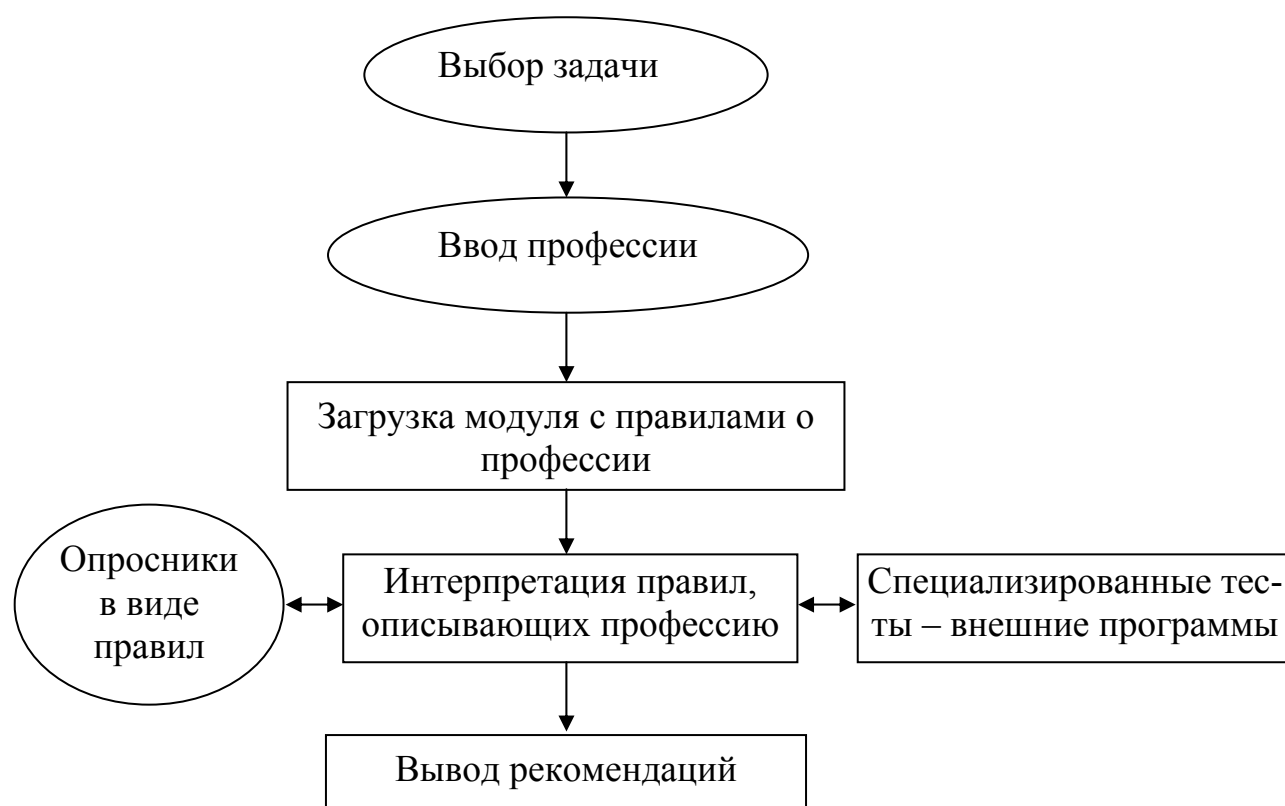


Рис. 3.2. Работа ЭС при решении задачи профессиональной диагностики

Правила для выбора профессии используют рекомендуемые виды деятельности, а также классификацию профессий не только по типам, но и по классам, отделам.

3.4.3. Представление знаний

В системе PROFEX используется представление знаний в виде правил-продукций с коэффициентами достоверности в процентах для представления нечеткости. Элементарные условия так же как, и заключения, сопровождаются коэффициентом достоверности. Правило имеет вид:

ПРАВИЛО номер правила :
ЕСЛИ
 элементарное условие 1,

 элементарное условие n ,
ТО
 заключение 1 КД = число

 заключение n КД = число.

Причем правила делятся на два вида в зависимости от формы интерпретации левой части:

- как конъюнкции элементарных условий с возможностью "отсеивания правила по порогу достоверности";
- как набора признаков с весовыми коэффициентами для вычисления взвешенной суммы их достоверностей для присвоения ее значения коэффициенту достоверности результата.

"Отсеивание правила по порогу достоверности" означает, что коэффициент при элементарном условии задает нижнюю границу достоверности факта, при которой еще имеет смысл далее интерпретировать данное правило. Если факт, соответствующий данному элементарному условию, имеет достоверность ниже пороговой, интерпретация правила прекращается (правило отсеивается).

Примеры правил такого вида для оценки пригодности к профессии "брокер" приведены ниже:

ПРАВИЛО 1: ЕСЛИ

Профессия=брокер КД=100% (Какую профессию Вы выбираете?) И
Тип=Человек-человек КД=60% (Какой тип профессии подходит для Вас?) И
Тип = Человек – знаковая система КД=50% (Какой тип профессии подходит для Вас?) И

Концентрация и устойчивость внимания КД=90% (Можете ли Вы сконцентрировать и длительно удерживать свое внимание на чем-либо?) И

Объем и распределение внимания КД=90% (Вы способны легко следить сразу за несколькими объектами?) И

Гибкость, адаптивность поведения КД=90% (Вы хорошо приспосабливаетесь к изменяющимся условиям и поведению окружающих ?) И

Переключение внимания КД=90% (Можете ли Вы быстро переключать свое внимание?) И

Долговременная память КД=95% (У Вас хорошая долговременная память?) И

Ответственность КД=90% (Вы ответственный человек?) И

Общительность КД=90% (Вы общительный человек?) И

Хорошая дикция КД=90% (Вы обладаете хорошей дикцией?) И

Дизъюнкт=/*мед. противопоказания для брокера КД=100%:

органов дыхания или

сердечно-сосудистой системы. И

Оперативная память КД=90% (У Вас хорошая оперативная память?) И

Эмоциональная уравновешенность КД=80% (Вы эмоционально устойчивы?)

ТО

Можно=да КД=100 И

Отказ=нет КД=100

Пример правила для определения причины непригодности к профессии "брокер":

ПРАВИЛО 2: ЕСЛИ

Профессия=брокер КД=100% (Какую профессию Вы выбираете?) И

Тип=/*Человек-человек КД=40% (Какой тип профессии подходит для Вас?)

ТО

Можно=нет КД=100. И

Отказ=Вам не подходит тип профессии "человек-человек" КД=100.

Использование второго вида интерпретации правил позволяет естественным образом вставлять в базу знаний обычно применяемые психологами опросники такие, как ДДО (дифференциально-диагностический опросник), тест Кэттелла для оценки психических характеристик и т.п., основанных на подсчете баллов при ответе на вопросы.

Пример правила второго вида интерпретации из опросника ДДО:

ПРАВИЛО 41: ЕСЛИ

2d=Составлять таблицы, схемы, программы для вычислительных машин КД=100% (Из двух возможностей что бы Вы предпочли?) И

5d=Обсуждать научно-популярные книги, статьи КД=100% (Из двух возможностей что бы Вы предпочли?) И

9d=Искать и исправлять ошибки в текстах, таблицах, рисунках КД=100% (Из двух возможностей что бы Вы предпочли?) И

10d=Выполнять вычисления, расчеты КД=100% (Из двух возможностей что бы Вы предпочли?) И

12d=Разбираться в чертежах, схемах, таблицах (проверять, уточнять) КД=100% (Из двух возможностей что бы Вы предпочли?) И

15d=Составлять точные описания-отчёты о наблюдаемых явлениях КД=100% (Из двух возможностей что бы Вы предпочли?) И

19d=Заниматься черчением, копировать чертежи, карты КД=100% (Из двух возможностей что бы Вы предпочли?) И

20d=Работать на клавишных машинах (пишущей машинке, телетайпе) КД=100% (Из двух возможностей что бы Вы предпочли?)

ТО

Тип=Человек – знаковая система КД=100.

Здесь идентификаторы типа "20d" – условные обозначения объектов, представляющих собой ответы на соответствующий вопрос. Так, например, объект "20d" имеет кроме указанного в правиле второе возможное значение "вести борьбу с болезнями растений, вредителями леса, сада".

Другой пример правила из теста Кэттела:

ПРАВИЛО 35: ЕСЛИ

15=нет КД=200% (Когда я планирую что-нибудь, я предпочитаю делать это самостоятельно, без чьей-либо помощи?) И

15=иногда КД=100% (Когда я планирую что-нибудь, я предпочитаю делать это самостоятельно, без чьей-либо помощи?) И

32=да КД=200% (Мне нравятся общительные, компанейские люди?) И

32=не знаю КД=100% (Мне нравятся общительные, компанейские люди?) И

49=возглавляя группу в туристском лагере КД=200% (Я бы скорее предпочел провести два летних месяца?) И

49=затрудняюсь сказать КД=100% (Я бы скорее предпочел провести два летних месяца?) И

66=неверно КД=200% (Я избегаю общественной работы и связанной с этим ответственностью?) И

66=иногда КД=100% (Я избегаю общественной работы и связанной с этим ответственностью?) И

83=да КД=200% (Я получаю большое удовольствие, рассказывая местные новости?) И

83=иногда КД=100% (Я получаю большое удовольствие, рассказывая местные новости?) И

100=где надо играть в команде или иметь партнера КД=200% (Я предпочитаю игры?) И

100=не знаю КД=100% (Я предпочитаю игры?)

ТО

Конформизм КД=100.

Здесь в отличие от предыдущего примера результатом работы правила является не значение объекта, а атомарное утверждение "Конформизм", имеющее смысл признака или психофизиологической характеристики личности.

В условиях правила числами обозначены также объекты-ответы на вопросы, так, например, объект "100" может иметь одно из трех значений:

Я предпочитаю игры (это – вопрос)

- 1) где каждый играет за себя;
- 2) не знаю;
- 3) где надо играть в команде или иметь партнера.

Повторение двух разных значений в правиле означает, что каждое из них прибавляет веса результирующему утверждению.

Комбинация второго вида интерпретации правил и отсеивания правил по порогу достоверности может рассматриваться как моделирование в данной архитектуре синтаксическими средствами правил-продукций искусственной нейронной сети. В настоящее время такого рода представление знаний получило название нечетких или гибридных нейронных сетей [31]

Факты могут быть двух видов:

- объект, значение, коэффициент достоверности;
- утверждение, коэффициент достоверности.

Экспертная система может работать со строковыми и численными значениями объектов. В качестве элементарных условий в левой части правила могут использоваться следующие отношения:

$>$, $<$, $=$ для численных значений объектов;

$=$, \neq для текстовых значений объектов;

"истинно", "ложно" для утверждений, "истинно" кодируется самим утверждением, а "ложно" его отрицанием с частицей "не" перед ним.

Для интерпретации правил используется обратный логический вывод.

Во внутреннем представлении правила и факты выглядят как предикаты пролога. Правило кодируется предикатом

$\text{rule}(\text{N}, \text{List1}, \text{List2}),$

где N – номер (идентификатор) правила;

List1 – список элементарных условий;

List2 – список заключений.

Элементарное условие и заключение кодируется структурой

$X(\text{Object}, \text{Value}, \text{KD})$ или $Y(\text{Sentence}, \text{KD}),$

где X, Y – двух- или трехбуквенное обозначение типа условия или заключения;

Object – название объекта;

Value – значение объекта;

Sentence – атомарное утверждение;

KD – коэффициент достоверности.

Факт кодируется предикатом вида $\text{fact}(X(\text{Object}, \text{Value}, \text{KD}))$ (факт-объект) или $\text{fact}(Y(\text{Sentence}, \text{KD}))$ (факт-утверждение).

Кроме этих основных предикатов база знаний во внутреннем представлении содержит предикаты для связи утверждений с внешними программами-тестами, запускаемыми, если факта-утверждения или его отрицания нет еще в базе данных $\text{test}(\text{Sentence}, \text{NameProg})$, предикаты для связи главного модуля с модулями базы знаний, описывающими конкретные профессии $\text{mod}(\text{NameProf}, \text{NameMod})$, предикаты для описания возможных значений объектов $\text{obj}(\text{NameObj}, \text{ListValue})$ и вопросов, которые система задает пользователю при определении значения объекта $\text{quest}(\text{NameObj}, \text{Text})$ или при проверке утверждения, $\text{quest}(\text{Sentence}, \text{Text})$ и некоторые другие вспомогательные предикаты.

3.4.4. Заключение

Разработанные программы были опробованы в Городском центре занятости г. Новосибирска. Была создана экспериментальная база знаний примерно о двадцати профессиях. Работа с системой PROFEX показала ее работоспособность и эффективность принципов представления и обработки знаний, положенных в ее основу.

Программы системы PROFEX реализованы в 1994 году на языках PDC-Prolog и Turbo-Pascal 7.0 в среде MS DOS.

3.5. Архитектура программы AnalDB для анализа баз данных с помощью нейронных сетей [15-18]

3.5.1. Назначение и функции программы AnalDB

Под анализом базы данных в данной работе понимается комплекс мероприятий, направленных на выявление скрытых закономерностей в базах данных. Рассматривается использование аппарата искусственных нейронных сетей для решения задачи, которая в рамках задач автоматизированного интеллектуального анализа данных может трактоваться следующим образом.

1. Качественное решение задачи определения значений полей БД, ассоциативно связанных с известными значениями других полей.
2. Решение задачи прогнозирования значений набора полей БД при заданных значениях другого набора полей.

Имея систему, схожую с ассоциативной памятью, а также формальные правила переноса информационного содержимого анализируемой базы данных в эту память, можно решить поставленную задачу. При этом используется ассоциативность такой системы и по значению некоторого фактора – "ключа" извлекаются ассоциативно близкие значения других факторов.

Один из возможных путей реализации ассоциативной памяти состоит в том, чтобы построить распределенную динамическую систему или сеть из дискретных элементов, аттракторами которой являются типичные картины-образы (в настоящее время принято сопоставлять различным образам, запомненным

нейронной сетью, установившиеся режимы или аттракторы соответствующей динамической системы). Каждая такая картина будет обладать своей областью притяжения, и всякое начальное условие, представляющее собой какую-то допустимую картину, обязано попасть в одну из ее областей притяжения. С течением времени в ходе эволюции эта начальная структура трансформируется в наиболее близкую из хранящихся в памяти структур-аттракторов, а именно в ту, области притяжения которой она принадлежала. Следовательно, подавая на вход в качестве начального условия для такой распределенной системы некоторую структуру, мы будем осуществлять ее автоматическое распознавание, которое будет параллельным. В роли такой распределенной динамической системы предлагается использовать искусственную нейронную сеть.

Предлагаемое решение в общем виде выглядит следующим образом. Искусственная нейронная сеть обучается на записях анализируемой реляционной базы данных. В процессе обучения нейронная сеть становится гносеологической моделью обучающей БД. Полученную таким образом модель пользователь может использовать для прогнозирования и исследования ассоциативных связей, скрытых в базе данных.

В качестве такой модели предлагается использовать сеть, представляющую собой конкретную реализацию динамической системы с ассоциативной памятью, предложенную Хопфилдом [31].

Поле факторов для данной задачи является содержание реляционной базы данных. В представлении совокупности факторов в виде содержания реляционной базы данных заключается первичная подготовка данных для решения поставленной задачи. Тем самым задаются структурные отношения и задача сводится к определению семантической близости содержимого полей заданной структуры. Так как бинарные искусственные нейронные сети оперируют двоичными векторами, то требуется дополнительная обработка исходных данных с целью приведения к виду, представляемому таким вектором. Необходимый для искусственной нейронной сети вектор получается конкатенацией двоичных векторов, представляющих поля реляционной базы данных (рис. 3.3).

Каждое поле реляционной базы данных обобщенно может быть отнесено к одному из двух типов: числовое, символьное. Значение символьного поля заносится в соответствующий каждому полю словарь. С тем, чтобы повысить эффективность процедур поиска значений в словаре и их кодирование, содержимое словарей ранжируется, повторяющиеся значения исключаются. Значения числовых полей представляются иначе. Диапазон значений каждого из числовых полей (к числовым полям относятся и те, которые содержат даты) разбивается на интервалы, а каждое конкретное значение представляется интервалом, к которому оно принадлежит. Точку на числовой оси можно представить вырожденным интервалом. Замена числовых значений интервалами обеспечивает большую гибкость представления. Аналогом словарей для числовых полей является перечень интервалов.

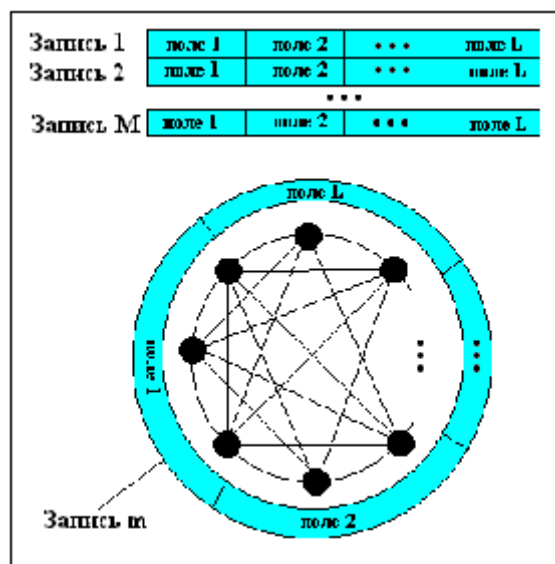


Рис. 3.3. Схематическое изображение представления структуры базы данных нейронной сетью

После того, как были сформированы словари и наборы интервалов, они подвергаются кодированию. То есть каждому значению в словаре или интервалу в наборе ставится в соответствие двоичный код, представляющий впоследствии это значение или интервал в нейронной сети. Размерность кода (длина кода) для каждого словаря и набора (другими словами для каждого поля) определяется мощностью множества разнообразных значений в словаре или наборе, а также способом кодирования. Различные варианты кодирования могут существенно сказываться на качестве работы ассоциативной памяти, представленной нейронной сетью Хопфилда, что подтверждается результатами макетного моделирования.

Формирование словарей и их кодирование является подготовительным этапом обучения нейронной сети. Каждая запись базы данных представляется в виде двоичного вектора и участвует в формировании весовых коэффициентов связей между формальными нейронами сети. Само обучение осуществляется в соответствии с известным правилом Хебба.

Из обучения желательно удалить повторяющиеся записи базы данных. Так как повторная запись вектора в ассоциативную память, представленную нейронной сетью неблагоприятно сказывается на энергетическом ландшафте этой сети. А именно, удваивается глубина энергетического минимума соответствующего такому вектору [32]. Более глубокий минимум обычно имеет более широкую область притяжения, а это уменьшает области притяжения соседних минимумов и искажает соответствующие притягивающие векторы. В предельном случае это может привести к тому, что в системе останется всего один очень глубокий минимум энергии, область притяжения которого захватывает все возможные спиновые конфигурации. У системы формируется как бы "навязчивая идея", что, конечно же, крайне нежелательно.

Существуют различные версии сетей Хопфилда сходные в структуре, но несколько отличные в функционировании. В данной архитектуре были использованы бинарные (возможно два состояния нейрона: +1 и -1) нейросети Хопфилда с дискретным функционированием во времени. Спецификой объединяющей такие сети является их асинхронность функционирования. То есть в отличие от синхронного функционирования, при котором состояние всех нейронов сети определяется одновременно, асинхронное функционирование подразумевает в каждый конкретный момент времени возможность переключения только одного нейрона.

В модели Хопфилда могут быть реализованы детерминированный и/или стохастический (с имитацией "отжига") алгоритмы. Процесс функционирования нейронной сети является многоитерационным. Каждая итерация включает в себя два шага: выбор нейрона-кандидата, формирование состояния выбранного нейрона-кандидата.

Отличие стохастического функционирования от детерминированного заключается в методе выбора нейрона-кандидата. При стохастическом кандидате на переключение является нейрон, выбранный случайно с помощью датчика случайных чисел. Таким образом, возможны такие ситуации, что в процессе функционирования состояние некоторых нейронов не анализировалось. При детерминированном функционировании нейроны становятся кандидатами на изменение своего состояния в порядке следования своих номеров.

В алгоритмах с имитацией "отжига" изменение состояния нейрона-кандидата носит вероятностный характер. Если в результате изменения состояния на противоположное полная энергия сети понизится, то состояние нейрона меняется на противоположное. Иначе состояние нейрона меняется на противоположное с определенной вероятностью, зависящей от параметра, называемого "температурой" сети, имитирующей уровень теплового шума. На первой итерации функционирования нейросети устанавливается начальное значение температуры (максимальное значение), затем постепенно через некоторое количество итераций значение температуры уменьшается. Итерационный процесс прекращается, когда температура достигает конечного значения (минимальное значение).

Если сравнивать между собой алгоритмы без "отжига" и с имитацией "отжига", то у первых множество преимуществ перед вторыми (они проще, требуют меньшее количество итераций, обладают большей точностью при условии, если энергетический ландшафт вблизи глобального минимума не изрезан локальными минимумами). Но, если энергетический ландшафт содержит множество локальных минимумов, то алгоритмы с имитацией "отжига" ведут себя лучше.

3.5.2. Этапы решения задач с помощью программы AnalDB

Использование нейронной сети Хопфилда для анализа базы данных включает в себя следующие этапы:

1. Формирование SQL-запроса для работы с нейронной сетью.

Ниже показана форма, используемая на этом этапе (рис. 3.4).

Формирование SQL-запроса

База данных: STANDARD1 Выбор базы данных

Путь к базе данных: D:\DATABASE

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG
▶	258	235	222	534	4440	3800	3440	
	360	325	309	1960	6240	4460	2990	
	305	275	275	420	3540	2660	1750	
	355	314	301	1380	4050	3960	4050	
	388	338	318	852	4620	5750	4970	
	422	364	334	1780	4970	2940	2470	
	209	190	190	327	3030	1580	1190	

SELECT JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DECEM
FROM "NGES22.DBF";

Выбор таблицы

Выбор полей

Dos <→ Win Проверить запрос Готово Отмена Заккрыть

Рис. 3.4. Форма для формирования SQL-запроса

2. Формирование словаря содержимого полей SQL-запроса.

Ниже приведена форма, используемая для редактирования словаря (рис. 3.5).

Просмотр и редактирование словаря

Выделенные в запросе поля Значения поля JUN

APR	1500-2000
MAY	2000-2500
JUN	2500-3000
JUL	3000-3500
AUG	3500-4000
SEP	4000-4500
OCT	4500-5000
	5000-5500
	5500-6000
	6000-6500
	6500-7000
	7000-7500

105 записей 12 значений Передать словарь

Шаг: 500 Max: 7500 Min: 1500 Закреть

Рис. 3.5. Форма для редактирования словаря

3. Создание и обучение нейронной сети.

Параметры нейронной сети устанавливаются в форме, приведенной ниже (рис.3.6). На этом же рисунке показано главное меню программы.

Анализ баз данных - AnalDB 3.1 Copyright 2001 (c) ООО "ИНСИКОМ", кафедра ВТ НГТУ
Проект Нейронная сеть Анализ БД Помощь Выход

Параметры нейронной сети (модели Хопфилда)

Кол-во итераций: 50 ☒ Детерминированный алгоритм
☐ Сотжигом

Кол-во нейронов: 0 (определяется при создании сети)

Коэффициент разрежения вектора: 1 ☒ Замораживать состояния нейронов

Дополнительные нейроны: 0 %

Начальная температура: 1

Экспоненциальный показатель изменения температуры: 0.1

Кол-во итераций на одно значение температуры: 10

INSYCOM Ltd

Библиотека функций сети Хопфилда
Copyright (c) 2000 г. ООО "ИНСИКОМ"

OK

Рис. 3.6. Форма для задания параметров нейронной сети Хопфилда

4. Поиск ассоциаций или прогноз значений полей БД.
На этом этапе используются формы, показанные ниже (рис. 3.7, 3.8).

Рис. 3.7. Форма для решения задачи прогнозирования

DISTRICT	SALL	SVIT	SKUH	TEL	BAL	LO
Железнодорожн	48	25	10	True	False	Fa
Ленинский	23.3	19	11.3	True	False	Tr
Ленинский	41.6	22.4	8	True	False	Tr
Центральный	81.2	50	14	True		Tr
Железнодорожн	74.4	48	14	True	False	Tr
Калининский	54	32	9	True		Tr
Октябрьский	54	32	9	True	False	Tr

Рис. 3.8. Форма для решения задачи поиска ассоциативных связей между полями SQL-запроса

Последний этап в свою очередь включает в себя следующие этапы.

1. Формирование запроса. На этом этапе пользователь выбирает часть полей БД и придает им некоторые значения (используя соответствующие этим полям словари значений и наборы интервалов). В процессе функционирования система будет прогнозировать значения других полей (отличных от задаваемых пользователем) подобно ассоциативной памяти.

2. Подготовка входного вектора сети. На этом этапе для нейронной сети формируется входной вектор. Он "собирается" из подвекторов, соответствующих полям базы данных. При формировании вектора нулевые разряды подвекторов заменяются на (-1) . Те подвектора, чьи значения задает пользователь, называются "замороженными". Значения других подвекторов формируются с использованием датчика случайных чисел.

3. Циклы функционирования нейронной сети. Этот этап включает в себя ряд циклов функционирования нейронной сети. Перед каждым циклом на нейронную сеть подается входной вектор, сформированный на предыдущем этапе функционирования. Нейронная сеть функционирует в соответствии с одним из четырех выше названных алгоритмов. Каждый цикл функционирования нейронной сети состоит из многократных итераций, при этом каждая итерация в свою очередь включает два вышеописанных этапа (выбор нейрона-кандидата, формирование состояния выбранного нейрона-кандидата). Следует заметить, что нейроны соответствующие "замороженным" подвекторам, не могут быть выбраны в качестве нейрона-кандидата, и, следовательно, они не меняют своего состояния в течении всего функционирования. По завершению первого цикла полученный нейронной сетью вектор и значение энергии сети запоминается в специальном буфере. В конце каждого последующего цикла значение энергии сети сравнивается с запомненным значением из буфера. Если текущее значение энергии меньше запомненного в буфере, то текущее состояние сети (состояние сети представляет собой вектор в качестве компонент которого выступают состояния нейронов) замещает в буфере ранее запомненный вектор.

4. Формирование выходного вектора. Это чисто формальный этап, когда вектор, сохраненный в буфере, выдается системой как вектор-ответ, восстановленный ассоциативной памятью по заданному пользователем фрагменту.

На рисунке 3.9 показана форма, которую можно использовать для того, чтобы просмотреть и анализировать вклад отдельных нейронов и полей в решение задачи нейронной сетью. Это делается с помощью визуализации весов связей между нейронами.

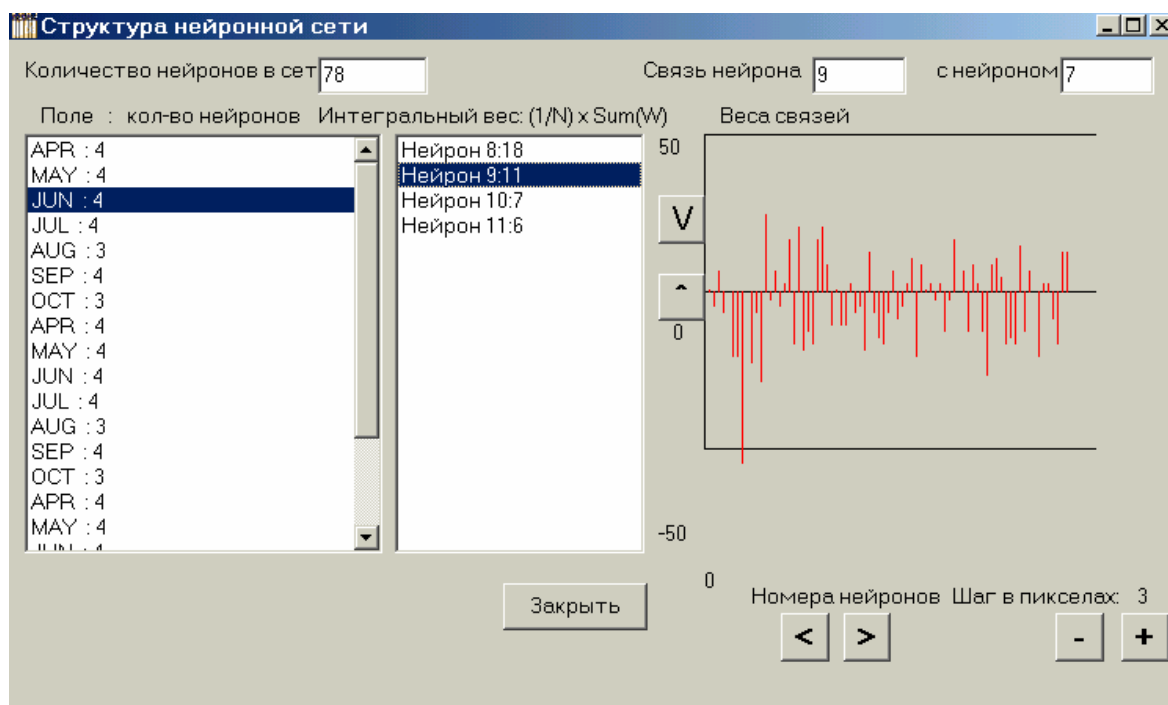


Рис. 3.9. Форма для просмотра и анализа весов связей между нейронами, характеризующих из вклад в решение задачи

3.5.3. Эксперименты по применению программы *AnalDB* для решения задачи прогнозирования притока реки Обь [19, 20, 34]

3.5.3.1. Постановка задачи

Для решения задачи оптимального управления наполнением и сработкой водохранилища, а также, стратегического планирования использования ресурсов водохранилища, необходимо иметь возможность прогнозирования притока воды в реку на ближайший месяц, квартал или год. В частности, такая задача была поставлена АО "Новосибирскэнерго" применительно к Новосибирскому водохранилищу. Исходными данными для ее решения были данные о расходе воды в реке Обь за последние 104 года (с 1894 по 1997 года).

Особенностью этих исходных данных является их существенная случайность. Обычными методами статистического анализа, выполненными с помощью пакета Statistica, в них не было обнаружено каких-либо существенных регулярностей (в частности, тренда) или закономерностей, кроме сезонных закономерностей [34, 35]. Поэтому в качестве альтернативного метода анализа было предложено использовать нейронные сети, а именно, модель Хопфилда [32] с обучением по правилу Хебба. Применение этой модели вместо традиционного многослойного перцептрона с обратным распространением ошибки дает возможность снизить затраты времени на обучение сети. Однако, следует ожидать большие значения ошибок обучения и прогноза в отличие от перцептрона. Но можно предположить, что при существенной случайности процесса высокая

точность не требуется, т.к. не достоверна, и длительный процесс обучения обратным распространением ошибки не окупается.

В качестве инструмента исследования была использована программа AnalDB 3.0 [15-18], предназначенная для анализа баз данных с использованием нейронных сетей.

Данные о притоке по месяцам кодировались в виде номера интервала, в который входит численное значение притока в куб.м/сек. Таким образом, множество полей одной или нескольких записей кодируются двоичным вектором, являющимся конкатенацией двоичных подвекторов. Выбираемая величина интервала при кодировании численного поля определяет точность прогнозирования, с одной стороны, и требования к информационной емкости сети, с другой стороны.

3.5.3.2. Пример эксперимента

Сеть обучалась на базе данных, состоящих из записей о притоке воды с 1894 года по 1997 год. В каждой записи содержатся данные о притоке по месяцам (12 полей) за один год. Записи упорядочены по возрастанию лет. В качестве источника информации при обучении и прогнозе используется SQL-запрос, содержащий следующие месяцы: апрель, май, июнь, июль, август, сентябрь, октябрь.

В качестве обучающего примера используется последовательность из 3 записей (3 соседних года), т.е. прогноз делается по предыдущим 2-м годам. Причем все поля последней записи, соответствующие семи месяцам, являются прогнозируемыми. Семь месяцев (апрель, май, июнь, июль, август, сентябрь, октябрь) были выбраны, как наиболее информативные и наиболее интересные с точки зрения прогноза. Прогноз дается на 1998 год. Результаты эксперимента приведены ниже в табл. 3.2.

Таблица 3.2

Месяц	Фактическое значение	Прогноз (интервал значений)	Ошибка в % (абсол.)	Процент правильных прогнозов на обучающей выборке
Апрель	1527,3	1500-1800	8 (122.7)	21
Май	3376,5	3000-3500	3.75 (126.5)	26
Июнь	3600	3000-3500	9.72 (350)	32
Июль	3396,8	3500-4000	10.4 (353.2)	29
Август	1431,6	1500-2000	22.24 (318.4)	40
Сентябрь	952,3	1200-1500	41.76 (397.7)	35
Октябрь	742,9	900-1200	41.34 (307.1)	40

В этой таблице последняя графа характеризует ошибку обучения, а предпоследняя – ошибку прогнозирования. При этом ошибка прогнозирования в процентах вычислялась по формуле $100(|V_{\phi} - (V_{\text{п}}^1 + V_{\text{п}}^2)/2| / V_{\phi})$ где V_{ϕ} – фактиче-

ское значение, $V_{\text{п}}^1$ и $V_{\text{п}}^2$ – нижняя и верхняя границы прогнозного интервала, соответственно.

Из таблицы видно, что ошибка прогнозирования имеет тенденцию к росту внутри прогнозируемого года. Отсюда можно сделать предположение о том, что приток в некотором месяце значительно сильнее зависит от притока в предыдущие месяцы текущего года, чем от поведения притока в предыдущие годы. Это предположение подтверждается и результатами статистической обработки.

В табл. 3.3 приводятся результаты аналогичного эксперимента, в котором проводилось прогнозирование притока на 1999 год.

Таблица 3.3

Месяц	Фактическое значение	Прогноз (интервал значений)	Ошибка в % (абсол.)	Процент правильных прогнозов на обучающей выборке
Апрель	1518,7	600-900	50.6 (768.7)	22
Май	2744	3000-3500	18.44 (506)	26
Июнь	2573.8	3000-3500	26.27 (676.2)	31
Июль	1817.1	1500-2000	3.69 (67.1)	28
Август	1913.7	1500-2000	8.55 (163.7)	38

3.5.3.3. Результаты экспериментов

Аналогичные эксперименты применительно к 1998 году проводились для разных длин последовательностей записей L , на которых обучалась сеть ($L=2, 4, 5, 6, 12$). В табл. 3.4 приведены ошибки прогноза, полученные в этих экспериментах.

Таблица 3.4

Интервал	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь
2	47.32	3.75	18.06	48.48	22.44	41.76	41.34
3	8	3.75	9.72	10.4	22.44	41.76	41.34
4	8	3.75	9.72	48.48	22.44	41.76	41.34
5	47.32	3.75	9.72	10.4	22.44	41.76	41.34
6	47.32	3.75	9.72	48.48	22.44	41.76	41.34
12	31.25	3.75	9.72	25.12	22.44	41.76	41.34

Отсюда видно, что ошибка прогноза практически не зависит от длины обучающего примера (даже, наоборот, при $L=3$ наблюдаются лучшие результаты за счет отсутствия грубых ошибок, что однако может являться случайностью, например, при прогнозе на 1999 год большая ошибка прогноза наблюдается для апреля).

В табл. 3.5 приведены ошибки обучения, полученные при этих же экспериментах.

Таблица 3.5

Интервал	Апрель	Май	Июнь	Июль	Август	Сентябрь	Октябрь
2	92	78	84	79	64	65	64
3	79	74	68	71	60	65	60
4	72.44	77.2	77.45	58.59	55.45	65.35	57.43
5	74.49	73	64.65	58.34	49	64	61
6	70.1	71.7	60.61	56.25	44.44	61.62	55.56
12	60	75	64	34	48	57	49

Отсюда видно, что ошибка падает с увеличением L . Это вряд ли можно объяснить увеличением точности прогноза при учете более длинной истории процесса, т.к. статистические методы не подтверждают наличия каких-либо существенных закономерностей (трендов) между годами. Это уменьшение ошибки можно объяснить уменьшением требуемой при обучении информационной емкости сети.

Первое впечатление от полученных результатов – а не слишком ли много ошибок дает прогноз с использованием данной модели сети даже при замене численных значений интервальными значениями (т.е. снижая точность прогноза)? Но надо иметь ввиду, что ошибка прогноза притока на 1998 год вполне приемлема. Однако, для уточнения прогноза и для оценки достоверности прогноза желательно использовать статистические методы.

Кроме того, в приведенных экспериментах применялся простейший алгоритм кодирования значений, при котором не сохраняется семантическая близость между значениями. Можно ожидать, что применение соответствующих алгоритмов [25] даст лучшие результаты.

3.5.3.4. Выводы

Из результатов проведенных экспериментов и анализа особенностей имеющихся в наличии данных о притоке реки Обь можно сделать следующие выводы.

1. В условиях сильной случайности процесса, когда традиционные методы статистического анализа не выявляют существенных корреляционных зависимостей, можно для прогноза использовать модель Хопфилда. В отличие от применения многослойного перцептрона с обратным распространением ошибки это дает возможность в итерационном режиме (методом проб и ошибок) без существенных затрат времени подбирать наиболее подходящие параметры модели и кодирования данных.

2. При кодировании численных значений интервалами значений следует выбирать величину интервала так, чтобы, с одной стороны не слишком "угрубить" прогнозирование, а с другой, – не снизить качество обучения сети за счет ее ограниченной информационной емкости.

3. Следует использовать для прогноза с применением модели Хопфилда не длинные обучающие примеры. Если во временном ряду наблюдается корреляция между удаленными значениями, модель Хопфилда не применима.

4. Следует дополнить применение нейронных сетей применением статистических методов для оценки достоверности решения, выдаваемого сетью, выбора начальных значений параметров модели для экспериментов, а также, выявления тренда с целью исключения его влияния на снижение качества обучения при прогнозе абсолютных значений.

Обученная с помощью программы AnalDB нейронная сеть была встроена в программное обеспечение NGES для управления сработкой-наполнением Новосибирского водохранилища. Это ПО предназначено для учета притока реки и других гидротехнических характеристик водохранилища, прогноза притока (методами статистического анализа, разработанных и описанных в [32, 33], и с использованием модели Хопфилда), расчета сработки-наполнения водохранилища.

3.5.5. Заключение

К основным достоинствам предлагаемого подхода для анализа данных с помощью нейронной сети можно отнести следующие:

- органичное объединение решений таких задач как прогнозирование и обнаружение ассоциаций в рамках единого подхода;
- обучение нейронной сети с целью последующего моделирования требует всего двух сплошных проходов по моделируемой базе данных. В то время как при традиционном подходе, а именно при накапливании статистик о комбинациях значений, встречающихся в БД, требуются многоитерационные сложные процедуры поиска;
- решение задачи в тех ситуациях, когда традиционный подход оказывается статистически несостоятельным. Например, когда анализируемая БД содержит небольшое число похожих (нетождественных) записей;
- универсальность нейронной сети Хопфилда, используемой в качестве основы построения ассоциативной памяти;
- простота программной реализации модели Хопфилда.

Предложенный подход реализован в программе для анализа баз данных AnalDB. Исходными данными для работы с нейронной сетью в ней служит SQL-запрос. Программа реализована на Delphi 5.

В настоящее время находится в стадии разработки и тестирования следующая версия программы AnalDB [45], которая будет лишена многих недостатков, в частности, она ориентирована на решения множества задач анализа баз данных (поиска ассоциаций, прогноза, классификации, кластеризации, распознавания) с использованием разных моделей нейронных сетей (модели Хопфилда, многослойного перцептрона и карт Кохонена).

3.6. Архитектура двухполушарных экспертных систем [24, 36-41]

Предлагаемая архитектура (впервые ее концепция была предложена автором в 1989 году в [24]) так называемой "двухполушарной" экспертной системы базируется на комбинации традиционной экспертной системы (в дальнейшем просто экспертной системы ЭС), осуществляющей логическую обработку и искусственной нейронной сети, реализующей ассоциативную обработку. Взаимодействие между этими двумя компонентами осуществляется посредством простых фактов или гипотез, которые имеют смысл наличия некоторых событий, свойств или отношений между сущностями. Для этого служит механизм "доски объявлений" (black board), впервые предложенный применительно к построению экспертной системы для распознавания речи HEARSAY-III [42]. На рисунке 3.10 показана архитектура "двухполушарной" экспертной системы.

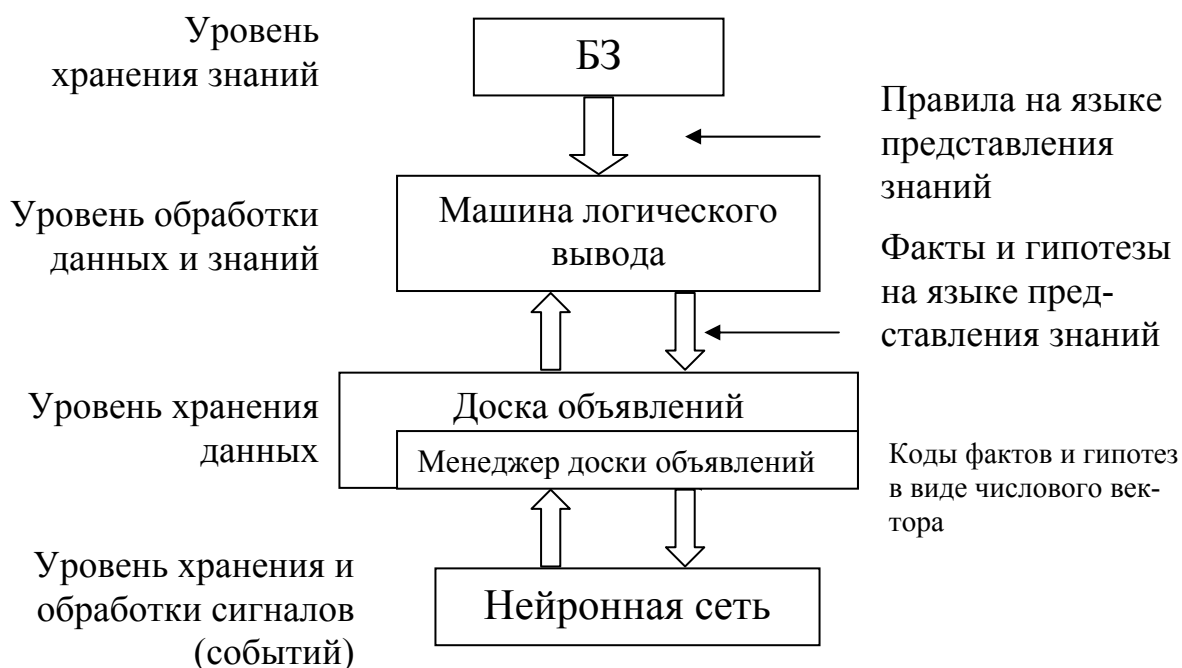


Рис. 3.10. Взаимодействие логической ЭС и нейронной сети

Очевидно, что ЭС и нейронная сеть должны функционировать относительно независимо друг от друга, т.е. параллельно или квазипараллельно. Поэтому можно сформулировать следующие требования к ЭС (механизму логического вывода), используя терминологию и классификацию архитектур ЭС из [42]:

- ориентация на параллельную асинхронную обработку фактов и гипотез;
- множественность линий рассуждений;
- возможность оценки успешности рассуждений;
- способность отменять полученные выводы (немонотонность рассуждений);
- возможность работы с ненадежными фактами и выводами;
- возможность работы в реальном времени;

– возможность работы с временными понятиями (моменты и интервалы времени, отношения "раньше", "позже", "одновременно" и т.п.).

Искусственная нейронная сеть должна удовлетворять следующим требованиям:

- иметь двоичные входы и выходы (для кодирования фактов и гипотез);
- иметь возможность управления со стороны логической ЭС при оценке ее поведения как верного или неверного.

Нейронная сеть такой "двухполушарной" экспертной системы может выполнять следующие функции:

1) порождение гипотез на основе текущего набора фактов, которое по существу подменяет собой прямой логический вывод в классических экспертных системах;

2) предварительная обработка (распознавание – классификация или кластеризация, контрастирование и т.п.) сигнальной информации или данных, поступающих на систему, и преобразование их в факты или гипотезы;

3) формирование ассоциативных связей между фактами взамен дерева решения, получающегося в ходе логического вывода, с целью ускорения принятия решения в "стандартных" (часто встречающихся ситуациях) или с целью переноса обученной нейронной сети в другую прикладную интеллектуальную систему (в этом случае экспертная система с ее базой знаний предназначена для обучения нейронной сети).

На верхнем уровне (рис. 3.10) обеспечивается хранение и обработка знаний с использованием традиционных методов инженерии знаний. Для этого используется продукционный метод представления знаний и обратный нечеткий логический вывод.

Данные для логического вывода (факты и гипотезы) берутся с доски объявлений, являющейся рабочей памятью экспертной системы. Помещенные на доску объявлений факты и гипотезы побуждают менеджера доски объявлений активизировать тот или иной фрагмент базы знаний. Менеджер доски объявлений может выполнять следующие функции:

- генерация фактов;
- генерация гипотез;
- замена факта;
- замена гипотезы;
- увеличение коэффициента достоверности гипотезы;
- уменьшение коэффициента достоверности гипотезы;
- наказание нейронной сети;
- поощрение нейронной сети;
- установка режима работы нейронной сети;
- активизация нейронной сети.

Все, кроме последней функции, можно задавать в заключениях правил-продукций. Последняя функция (активизация нейронной сети) выполняется ав-

томатически каждый раз при появлении на доске объявлений нового факта или гипотезы, сгенерированных при срабатывании некоторого правила.

Факты и гипотезы имеют следующую структуру:

<имя объекта><имя атрибута><значение атрибута>.

Доска объявлений делится на уровни, соответствующие уровням иерархии объектов предметной области (иерархии классов или/и иерархии вложенности).

В соответствии со структурой фактов и гипотез каждый уровень доски объявлений делится на следующие подуровни (рис. 3.11):

- подуровень имени объекта;
- подуровень имени атрибута;
- подуровень значения.

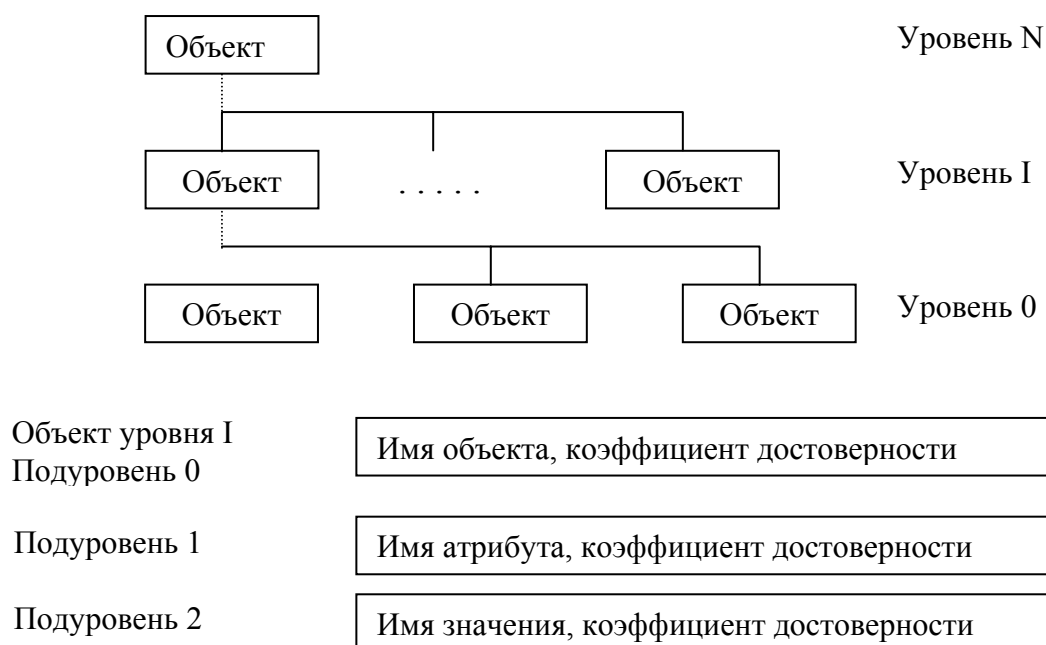


Рис. 3.12. Уровни доски объявлений

На каждом подуровне данные кодируются термом и сопровождающим его коэффициентом достоверности.

Номер уровня и подуровня составляют контекст, в котором протекает процесс логического вывода в экспертной системе и ассоциативной обработки в нейронной сети.

Коды имени на текущем подуровне и коды контекста являются входами нейронной сети. Выходом нейронной сети является код имени, который интерпретируется как один из вариантов в зависимости от установленного режима работы нейронной сети:

- код имени на текущем подуровне;
- код имени на более низком подуровне;
- код имени на более высоком подуровне.

Подуровень 2 не имеет более низкого подуровня, а для подуровня 0 более высоким подуровнем считается подуровень 0 следующего по иерархии уровня.

В качестве нейронной сети можно использовать модель "ключ-порог", предложенную автором [27-30] и описанную в разделе 3.3, или модель ассоциативного мышления, предложенную в разделе 1.3.

Данная архитектура была реализована в 1995 году в виде макетного образца в двух вариантах: на языке PDC-Prolog в среде MS DOS и на языках FoxPro и Borland C++ в среде MS Windows.

3.7. Выводы

В данной главе систематизированы и рассмотрены различные варианты использования искусственных нейронных сетей в интеллектуальных системах, большая часть из которых апробирована автором при разработке различных интеллектуальных систем, описанных в данной главе и в следующей главе 4. Кроме того, в данной главе описаны предложенные автором принципы построения двухполушарных интеллектуальных систем, объединяющих в себе классические методы представления и обработки знаний и методы ассоциативной обработки на основе нейронной сети.

В концепции двухполушарной интеллектуальной системы в полной мере отражены концепции и модели интеллектуальных систем, предложенные и описанные в главе 1. Эта парадигма может явиться основой для создания действительно интеллектуальных систем, рассуждающих так, как это делает человек.

4. ФОРМИРОВАНИЕ И РАСПОЗНАВАНИЕ СМЫСЛА В ЕСТЕСТВЕННОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ ГИБРИДНОГО ПОДХОДА

"... истина всегда несет на себе отпечаток двойственности слов, которыми ее пытаются выразить."

Ф. Герберт. Бог-император Дюны

4.1. Постановка задачи

Реализация диалога с компьютером на естественном языке (ЕЯ) является одной из главных проблем, решаемых в разделе информатики, называемой обычно "Искусственный интеллект". Недаром тест Тьюринга, целью которого является оценка "качества" искусственной интеллектуальной системы, базируется на диалоге с компьютером.

Главной же проблемой при создании диалоговых систем с использованием ЕЯ является проблема распознавания смысла (семантики) предложений ЕЯ. Кроме того, решение этой проблемы актуально и для решения чисто прикладных задач, таких как

- обработка документов (поиск по смыслу, реферирование, рубрикация);
- разработка поисковых серверов для поиска информации в Internet с использованием запросов на естественном языке;
- разработка средств для обработки запросов на естественном языке к базам данных;
- разработка средств тестирования знаний с использованием вопросов с открытым ответом, в частности, в системах дистанционного образования.

Решением проблемы формализации и распознавания смысла предложений на естественном языке занимаются давно и зарубежные и отечественные ученые. Из зарубежных известны работы П.Линдсея, Д.Нормана [1], Т.Винограда [2], Р.Шэнка [3], , Yang, P. Pai, V. Honavar, L. Miller [4], T.Honkela, S.Kaski, K.Lagus, T.Kohonen [5], S. Lowrence, C.L. Giles, S.J.Fong [6], S.W.McRoy, S.M.Haller, S.S.Ali [7], S.Tong, D.Koller [8], из отечественных (советских и российских) известны работы И.А.Мельчука [9], В.С.Файна [10], В.А.Ловицкого [11], В.В.Мартынова [12], Д.Я.Левина и А.С.Нариньяни [13], А.С. Нариньяни и А.Е. Кибрика [14], В.Ш. Рубашкина [15], Ю.А.Любарского [16], Н.М. Соломатина [17], Э.В. Попова [18, 19], Е.В.Золотова и И.П.Кузнецова [20], И.П.Кузнецова [21, 22], Осипова Г.С. [23], С.А.Шумского, А.В.Ярового, О.Л.Зорина [24], Н.Н.Жигирева, В.В.Коржа, Б.Н. Оныкого [25, 26], З.В.Дударя и Д.Е.Шуклина [27], Д.Е.Шуклина [28], А.Н.Флоренсова [29, 30] и др.

Существует два подхода к анализу предложения на ЕЯ.

1. Синтаксически ориентированный или на основе детального синтаксического разбора предложения. Средствами синтаксического анализа выделяются связанные понятия, которые объединяются в так называемые атомы смысла (АС). Создание АС идет только на основе данного предложения,

ла (АС). Создание АС идет только на основе данного предложения, определение связанных понятий идет только на основе синтаксических правил.

2. Семантически ориентированный, в рамках которого задача анализа предложения ЕЯ рассматривается как задача распознавания смысла предложения на основе использования базы знаний (о языке, об участниках общения, о предметной области и т.п.).

Разбор предложения идет путем вычленения связанных понятий (атомов смысла) с помощью базы знаний. На втором плане стоит синтаксический анализ, с помощью которого вычленяются дополнительные АС, те, которые не были сгенерированы с помощью БЗ.

Таким образом, после анализа предложения на ЕЯ имеется некоторый базовый набор атомов смысла, с помощью которого система начинает / продолжает работать с базой знаний.

Синтаксически ориентированный анализ ЕЯ использует в том или ином виде описание грамматики языка (структуры языковых конструкций). При этом распознавание языковых конструкций и их структуры является основной решаемой задачей, а после ее решения распознается семантика распознанных конструкций. Для структурного описания языка применяются контекстно-независимые и контекстно-зависимые грамматики Холмского, расширенные сети переходов.

Синтаксически ориентированный анализ ЕЯ имеет существенные недостатки, ограничивающие возможности его применения, главными из которых являются следующие:

- в нем трудно отразить неоднозначность понимания смысла языковых конструкций (в зависимости от различных контекстов – предметной области, хода диалога, ролей участников и т.п.);

- в нем трудно отразить "живость" языка, т.е. его терпимость к искажениям и появлению новых словоформ и синтаксических конструкций.

Попытки борьбы с этими недостатками в рамках синтаксически ориентированного анализа приводят к появлению очень громоздких, медленных и не поддающихся сопровождению программ.

Поэтому, наиболее перспективным и развивающимся направлением обработки ЕЯ в настоящее время является семантически ориентированный анализ [13].

В рамках семантически ориентированного анализа наиболее перспективным и адекватным предложенным в данной работе (см. главу 1) моделям является парадигма распознавания смысла предложений на естественном языке, предложенная В.С. Файном в [10].

В представлении знаний для семантически ориентированного анализа предложений ЕЯ можно выделить следующие варианты знаний.

1. Набор синонимов. Использование синонимов позволяет при ответе на запрос учитывать не только те термины, которые непосредственно указаны в запросе, но и все другие слова, близкие к ним по значениям. Здесь в качестве АС используются слова и нет иерархии АС.

2. Иерархия терминов и понятий, создаваемая человеком.

3. Лингвистические правила. Разработчик создает систему лингвистических правил, которые используются для анализа или грамматического разбора текстовой базы данных. Этот метод анализа определяет ключевые слова и понятия, объединяющиеся в базу знаний, которая отражает содержание конкретной базы данных. Затем база знаний используется для поиска и ранжирования групп родственных документов. Процесс грамматического разбора и создания базы знаний должен проводиться для каждой предметной области.

4. Использование ссылочных документов, в том числе обычных словарей и словарей терминов. Этот подход основан на смысловых значениях слов и называется семантической сетью. Как и словарь, семантическая сеть содержит атомы смысла, инкапсулирующие множество определений для каждого хранимого слова. Эти атомы смысла связаны между собой отношениями, при анализе предложения новые АС добавляются в семантическую сеть. Значения слов, наиболее подходящие для данного поиска, могут быть выбраны самим пользователем с целью повышения точности этого поиска. Подход на основе семантических сетей реально объединяет статистический поиск и поиск на основе базы знаний. При этом используются смысловые значения слов для определения и классификации отношений.

При семантически ориентированном анализе ЕЯ используются следующие методы:

- шаблоны;
- семантические грамматики;
- падежные фреймы;
- концептуальные зависимости.

Анализ ЕЯ с помощью шаблонов является очень грубым и поверхностным методом, но для практических применений при реализации диалоговых систем на ограниченном естественном языке с узкой предметной областью его применение может быть весьма эффективным. Примером использования шаблонов является одна из первых программ диалога на ЕЯ – программа Alice – и ее "потомки" [32].

Анализ ЕЯ, основанный на использовании семантических грамматик, очень похож на синтаксический, с той разницей, что вместо синтаксических категорий используются семантические. Естественно, семантические грамматики работают в узких предметных областях.

Недостатки семантических грамматик состоят в том, что, во-первых, необходима разработка отдельной грамматики для каждой предметной области, а во-вторых, они очень быстро увеличиваются в размерах. Способы исправления этих недостатков – использование синтаксического анализа перед семантическим, применение семантических грамматик только в рамках реляционных баз данных с исключением решения общезыковых проблем и комбинация нескольких методов (включая собственно семантическую грамматику).

С появлением падежных фреймов [33] связан большой скачок в развитии обработки естественного языка. На сегодняшний день падежные фреймы – один из наиболее часто используемых методов анализа ЕЯ, т.к. он является наиболее компьютерно-эффективным при анализе как снизу вверх (от составляющих к целому), так и сверху вниз (от целого к составляющим).

Падежный фрейм состоит из заголовка и набора ролей (падежей), связанных определенным образом с заголовком. Фрейм для компьютерного анализа отличается от обычного фрейма тем, что отношения между заголовком и ролями определяется семантически, а не синтаксически, т.к. в принципе одному и тому же слову могут приписываться разные роли (падежи), например, существительное может быть как инструментом действия, так и его объектом.

Использование метода падежных фреймов имеет следующие достоинства:

- совмещение двух стратегий анализа (сверху вниз и снизу вверх);
- комбинирование синтаксиса и семантики;
- удобство при использовании модульных программ.

Хотя существует большое количество моделей и методов анализа ЕЯ и программных реализаций, до сих пор нет достаточно полной модели представления знаний, содержащихся в предложении на ЕЯ, и эффективных принципов и алгоритмов обработки текстов на ЕЯ, объединяющих в себе достоинства методов инженерии знаний (например, семантических сетей) и преимущества нейронных сетей.

В данной главе предлагается такая модель и такие принципы и алгоритмы.

4.2. Обучаемые системы представления и распознавания смысла в ЕЯ

В системах для обработки естественного языка, обучаемых с использованием текстов или диалога на том же естественном языке, могут использоваться два подхода:

- количественный;
- качественный.

Количественный подход использует частотность появления слов в тексте. Он часто используется в системах поиска документов по содержанию, в частности, в Internet. Этот подход не использует и не синтезирует представление семантики текста. Поэтому здесь он далее рассматриваться не будет.

Качественный подход предполагает на основе обрабатываемого текста (документа или вводимого в процессе диалога сообщения) формирование базы знаний о смысле текста, которая может затем использоваться для решения каких-либо задач (для поиска документов по смыслу, ответа на вопросы и т.п.). При этом такие системы могут сильно отличаться друг от друга в зависимости от объема и глубины знаний, заложенных в нее априорно в виде базы знаний и/или алгоритмов обработки предложений естественного языка (его синтаксиса и семантики).

Наиболее перспективным вариантом качественного подхода на наш взгляд является использование априорных знаний только в том объеме, который компенсирует отсутствие в системе других источников информации, имеющих в наличии у обучаемой естественной интеллектуальной системы, в частности, источников зрительной информации и информации о двигательной активности. Например, при обучении ребенка новому понятию ему обычно показывают предмет и называют его. Если это слово для него еще не знакомо, он запоминает ассоциативную связь образа предмета с его обозначением (словом). Если же это слово уже было ранее связано с другим предметом, происходит процесс обучения на множестве примеров, сопровождаемый обобщением и формированием типичного представителя класса предметов, связанных с данным словом, т.е. понятия (например, понятия "стул" или "книга"). Если же надо сформировать понятие более высокого уровня, являющееся классом по отношению к сформированным ранее понятиям (например, "мебель"), ребенку показывают тот же стул и говорят "Это мебель", а затем, это же проделывают и для других видов мебели. Этот процесс показа можно смоделировать использованием в синтаксисе языка знака "-" (тире) и введением специальных алгоритмов обработки этого знака, формирующих ассоциативную связь между словами или словосочетаниями (или понятиями, с ними связанными), находящимися слева и справа от него.

Таким образом, автор исходит из того, что при обучении языковым понятиям осуществляется запоминание двух основных типов ассоциаций:

- неявно задаваемые ассоциации между словами, встречающимися рядом в тексте, формирующие понятия, связанные со словосочетаниями;
- явно задаваемые ассоциации между группами слов, задающие смысл новых слов и словосочетаний в контексте того, что уже известно интеллектуальной системе.

Далее ассоциативными (горизонтальными) связями будем называть только ассоциации последнего типа, ассоциации первого типа будем использовать для создания структур данных, описывающих понятия.

В основе представления и распознавания смысла ЕЯ предлагается использовать следующие принципы:

- ориентация на распознавание семантики с минимальным использованием знаний о синтаксисе языка;
- формирование иерархических структур из понятий, как результат обработки предложений ЕЯ, с горизонтальными (ассоциативными) связями между узлами этих иерархических структур;
- распознавание смысла (интерпретация) предложения на ЕЯ как процесс, состоящий из трех этапов:
 - 1) движения снизу вверх по иерархии понятий (распознавание понятий);
 - 2) движение по горизонтальным связям (поиск ассоциаций);
 - 3) движение сверху вниз по иерархии понятий (генерация отклика-выхода системы);

- распознавание слов, словосочетаний и понятий по максимальной схожести с использованием нейрореподобных алгоритмов.

Первый из этих принципов (семантически-ориентированный анализ) был сформулирован и активно развивался в работах Д.Я. Левина, А.С. Нариньяни [13] и В.С. Файна [10]. Первый и частично последний принципы были использованы автором в середине 90-х годов при создании экспертной системы для контроля знаний с ответом на естественном языке (см. 4.4). Последние три принципа были сформулированы автором и использованы им при создании системы программирования транспортных роботов на естественном языке [34-35] в рамках разработки системного ПО гибкого автоматизированного производства конденсаторов [36-38] для в конце 80-х годов (см. 4.3) и исследовательского прототипа программного обеспечения для поиска документов по смыслу в 2000 году [39] (см. 4.5). Также, они используются в настоящее время при развитии системы тестирования знаний Интертест [40] добавлением в нее возможности использования ответов на естественном языке при обработке тестовых заданий с открытым ответом.

4.3. Программное обеспечение для программирования роботов на естественном языке

4.3.1. Постановка задачи

В настоящее время одной из актуальных задач в области робототехники является создание адаптивных и интеллектуальных роботов с развитыми средствами программирования на естественном или ограниченном естественном языке, привычном технологу-программисту. В этом разделе описаны принципы построения системы подготовки программ (ДИСПОР), являющейся частью системного программного обеспечения транспортного робота-манипулятора (СПОР) с элементами искусственного интеллекта. ПО СПОР разрабатывалось в рамках создания гибкого автоматизированного производства конденсаторов [39,40,44] на ПО "ОКСИД" г. Новосибирска в конце 80-х годов. На рис. 4.1 показана структура СПОР [36].

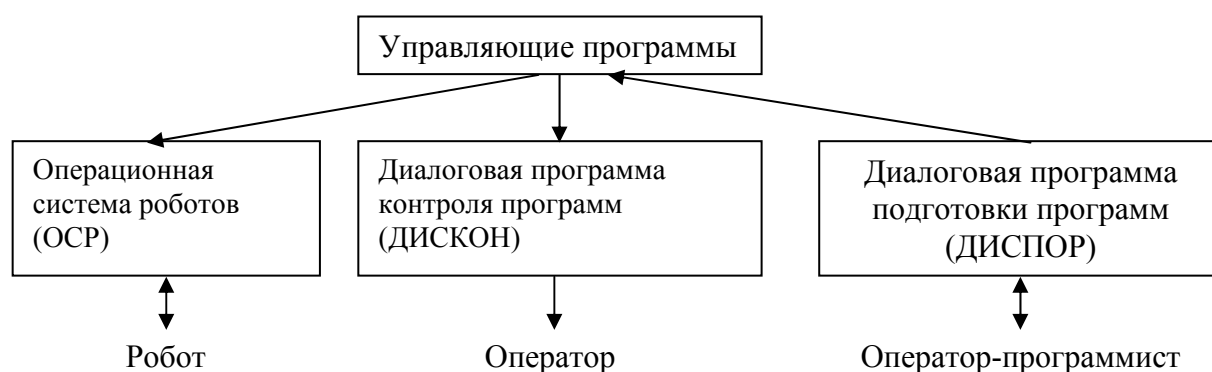


Рис. 4.1. Структура системного программного обеспечения транспортных роботов (СПОР)

ДИСПОР в процессе диалога с технологом программистом на языке, близком к естественному, синтезирует управляющие программы для роботов и размещает их в виде файлов на внешнем носителе (гибком магнитном диске). Управляющая программа кодируется на промежуточном базовом языке программирования транспортных роботов (БАЯР). После подготовки комплекса управляющих программ они могут пройти дополнительный контроль и оптимизацию с помощью диалоговой системы контроля и оптимизации управляющих программ (ДИСКОН). После этого, если ошибок не обнаружено, носитель с управляющими программами переносится на систему управления роботом (роботами), где управляющие программы загружаются и интерпретируются операционной системой робота (ОСР).

4.3.2. Принципы построения ДИСПОР

При разработке ДИСПОР была сделана попытка смоделировать процесс обучения понятиям предметной области без обучения грамматике языка. Суть такого обучения заключается в показе соответствия между тем, что демонстрируется системе (новыми словами) и тем, что она уже знает (ранее определенными словами и операторами языка программирования роботов БАЯР). При этом на грамматику языка (падежные окончания, порядок слов и т.п.) не обращается внимание. Распознавание слов и словосочетаний происходит по их максимальной похожести. При необходимости система задает уточняющие вопросы. Такое распознавание позволяет игнорировать в какой-то степени искажения слов, что дает возможность использовать в ДИСПОР речевой ввод.

Для сравнения слов в системе ДИСПОР был разработан следующий алгоритм:

```
// i – индекс буквы в первом слове str1
// j – индекс буквы во втором слове str2
// len1 – длина первого слова
// len2 – длина второго слова
// r – количество совпавших букв (результат)
i:=0;
j:=0;
r:=0;
while (i<len1) and (j<len2) do
  begin
    i:=i+1;
    j:=j+1;
    if str1[i]=str2[j] then
      inc(r)
    else
      begin
        if (j<=len2) and (str1[i]=(str2[j+1])) then
```

```

begin
  j:=j+1;
  r:=r+1;
end
else
begin
  if (i<=len1) and (str1[i+1]=str2[j]) then
    begin
      i:=i+1;
      r:=r+1;
    end;
  end;
end;
i:=i+1;
j:=j+1;
end; // while

```

Результат работы алгоритма (параметр r) используется для сравнения слов по максимальной похожести или для распознавания слова с использованием порогового значения этого параметра. Этот алгоритм был использован во всех разработках автора, связанных с распознаванием смысла естественного языка, приведенных далее в этой главе.

При разработке ДИСПОР мы исходили из стремления как можно меньше закладывать в нее априорной информации о языке и предметной области. В ДИСПОР используется три типа входных предложений, которые распознаются по используемым в них знакам препинания:

- 1) повествовательные, включающие в себя тире или двоеточие и заканчивающиеся точкой;
- 2) повелительные, не включающие тире или двоеточие и заканчивающиеся точкой или восклицательным знаком;
- 3) вопросительные, заканчивающиеся знаком вопроса.

Повествовательные предложения служат для обучения системы понятиям предметной области. Обучение в конечном итоге сводится к формированию в базе знаний ДИСПОР соответствий между словами или словосочетаниями естественного языка и операторами или последовательностями языка БАЯР.

Повелительные предложения служат для синтеза управляющих программ роботов. При их интерпретации ДИСПОР заменяет слова и словосочетания естественного языка на соответствующие им операторы языка БАЯР.

С помощью вопросительных предложений пользователь может спросить систему об известных ей понятиях, т.е. контролировать содержание базы знаний.

Кроме предложений на естественном языке с вкрапленными в них операторами языка БАЯР, ДИСПОР распознает и выполняет команды. Они подразделяются на две группы: команды, управляющие процессом диалога (начина-

ются с символа "&"), и команды, управляющие процессом формирования управляющих программ (начинаются с символа "@").

Первая группа команд включает в себя команды:

- начала диалога (&Н), при выполнении которой ДИСПОР осуществляет идентификацию пользователя с использованием пароля и загрузку базы знаний;
- изменение паролей (&П), которая доступна только привилегированным пользователям;
- просмотра содержимого словаря базы знаний (&С);
- удаления слова из база знаний (&У);
- конца диалога (&К), при выполнении которой ДИСПОР осуществляет сохранение базы знаний и завершение работы.

Вторая группа команд включает в себя команды:

- начала формирования управляющей программы (ØН);
- просмотра содержимого уже сформированной управляющей программы (ØП);
- отказа от формирования управляющей программы (ØО);
- возвращения назад в процессе формирования управляющей программы на заданное количество символов (ØВ);
- конца формирования управляющей программы (ØК).

Базовый язык программирования транспортных роботов – манипуляторов БАЯР.

Язык программирования роботов БАЯР разрабатывался, исходя из предположения о том, что он должен:

- быть достаточно простым в понимании и интерпретации операционной системой роботов;
- обладать полнотой для описания всех основных команд, которые могут выполнять транспортные роботы-манипуляторы, быть пригодным для программирования роботов с достаточно произвольной кинематической схемой и позиционной системой управления;
- иметь контекстно-зависимую грамматику, которая позволяет при максимальной простоте операторов и небольшом их количестве достичь требуемой гибкости;
- иметь средства для управления контекстом, в котором происходит интерпретация управляющих программ, написанных на нем;
- иметь возможность расширения с целью включения в него новых операторов или подмножеств операторов.

Язык БАЯР включает в себя следующие группы операторов:

- 1) задания контекста;
- 2) управления движением робота;
- 3) управления выполнением программ;
- 4) описания условий;
- 5) описания точек в зоне обслуживания робота;

- 6) описания состояния робота и технологического оборудования;
- 7) сервисные операторы.

Оператор языка БАЯР в общем случае имеет следующий синтаксис:

<оператор> :: = <имя> | <имя> <параметр>

<параметр> :: = <целое число> | <знак операции> <целое число>

Имя оператора кодируется русской или латинской буквой, знак операции – одним из следующих символов:

D – признак ссылки на объект определенный ранее, номер которого задан параметром оператора,

& – признак ссылки на стандартный объект (не требующий определения в программе), номер которого задан параметром,

+, * – операции сложения и умножения, выполняемые над параметром, к которому адресуется оператор.

Объект, над которым могут выполняться операции сложения и умножения является одной из так называемых контекстных переменных: номер робота, номер звена робота, параметр перемещения звена, номер технологического оборудования, номер ячейки обработки в технологическом оборудовании и универсальное контекстное число. Операторы для задания соответствующих контекстных переменных имеют имена R, Z, P, O, Y, C.

В группу операторов управления движением робота входят операторы без параметра: аварийного останова (A), перехода в исходное (базисное) состояние (B) и выполнения элементарного перемещения (действия) (D). Параметры для элементарного перемещения при выполнении оператора D (номер звена, номер робота, величина перемещения) равны значениям соответствующих контекстных переменных. Если контекстная переменная "параметр перемещения" содержит ссылку на точку в зоне обслуживания робота, оператор D вызывает перемещение в эту точку. Точка может быть описана оператором T, после которого следуют операторы X, Y, Z, задающие ее координаты в системе координат робота. Если присутствует только один из этих операторов, оператор T определяет плоскость, если два – прямую, параллельную какой-либо оси координат. Описание точки (плоскости, прямой) заканчивается оператором K или символом #. Можно говорить о том, что оператор T объявляет контекст описания точки, а оператор K или # отменяет этот контекст.

С помощью операторов U и E объявляется контекст описания условия. Оператор U имеет параметр – номер (идентификатор) условия, определяемого далее (в контексте этого оператора). В дальнейшем на это условие можно ссылаться с помощью оператора E, проверяющего его.

Оператор E может не иметь параметра (ссылки на условие). В этом случае условие описывается после оператора E. После выхода из контекста, объявленного оператором E, должен следовать оператор B (выполнить) который объявляет контекст ветви, выполняемой при условии.

В контексте описания условия с помощью операторов описания состояния (C) и оборудования (G), а также с помощью логических связок И, Л (или) и опе-

ратора Н (не) можно описать логические условия. Кроме того, в этом контексте можно определять новые локальные значения контекстных переменных, а через них определять объекты, состояния которых проверяются. В операторах С и Г состояние описывается в виде целого числа-параметра оператора.

В группе операторов управления выполнением программ, кроме операторов Е и В, относятся оператор Ц задания контекста циклического участка и оператор Ж, обеспечивающий ожидание выполнения условия или конца интервала времени. Оператор Ц обеспечивает выполнение циклического участка пока истинен предикат, описываемый с помощью оператора Е до оператора Ц или после него, или заданное в параметре оператора Ц количество итераций.

К сервисным операторам относятся некоторые операторы, помогающие отлаживать управляющие программы.

Все операторы, которые могут иметь параметр, интерпретируются без него. В этом случае значение параметра берется равным универсальному контекстному числу.

4.3.3. Представление знаний

В основу представления знаний в ДИСПОР положены следующие принципы:

- декларативное представление знаний на основе семантической сети;
- принцип максимальной однородности семантической сети.

База знаний ДИСПОР состоит из словаря, списка операторов языка БАЯР и семантической сети. Словарь и список операторов представляют из себя списки, динамически изменяемые и пополняемые в процессе обучения.

Семантическая сеть состоит из вершин – понятий и ребер семантических отношений. Понятия могут быть двух типов: терминальные Т и абстрактные А. Терминальным понятиям соответствуют слова или словосочетания на естественном языке, а также операторы языка БАЯР или фрагменты программы на языке БАЯР. В первом случае терминальное понятие назовем понятием-сенсором, во втором – понятием-актором. Абстрактным понятиям соответствует класс понятий. Семантические отношения могут быть двух типов; типа вхождения в понятие и типа эквивалентности.

Семантические отношения типа вхождения в понятие служат для отображения состава понятий обоих типов. Они не симметричны и состоят из двух семантических отношений: вхождения в понятие (I) и включения в состав (C). Семантические отношения типа эквивалентности Е служат для отображения в базе знаний соответствий между понятиями и являются симметричными. Таким образом каждому k -ому узлу семантической сети $G = \{g_i\}$ соответствует подмножество узлов $G^{(I)}$ таких, что $g_k I g_i \forall g_i \in G^{(I)}$, подмножество узлов $G^{(C)}$ таких, что, $g_k C g_i, g_i I g_k \forall g_i \in G^{(C)}$, подмножество узлов $G^{(E)}$ таких, что $g_k E g_i \forall g_i \in G^{(E)}$.

На рис. 4.2 даны примеры семантических сетей, соответствующих трем повествовательным предложениям (рис. 4.2, а, б, в) и получающейся результирующей семантической сети (рис. 4.2, г). При программировании робота с использованием этой сети понятию "поднять" соответствует перемещение звена 2 (в вертикальном направлении) на 500 единиц вверх, понятию "иди к цели" – перемещение транспортного робота к точке 1.

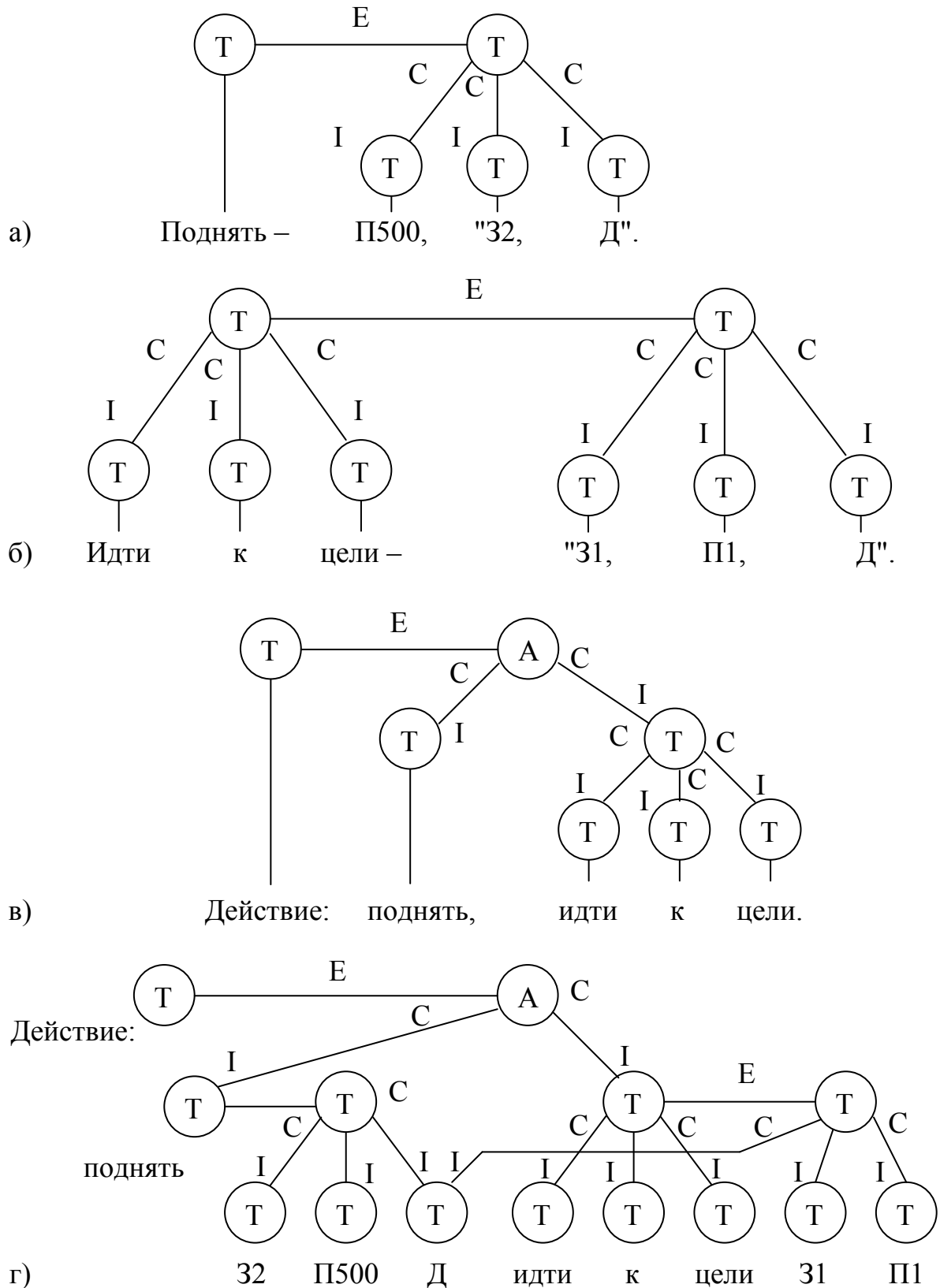


Рис. 4.2. Примеры представления знаний

Более сложные семантические отношения, необходимые для моделирования внешней среды и состояния робота конструируются из отношений вхождения в понятие и эквивалентности и понятий, связанных с некоторыми словами, с помощью которых идентифицируются понятия. К таким понятиям относятся: объект, агент, действие (агента на объект), место, время, определение, число (количество объектов в другом объекте), результат (действия) в т.п.

4.3.4. Формирование управляющей программы на языке БАЯР

Формирование управляющей программы начинается с выполнения команды $\emptyset N$. При этом ДИСПОР готовит для формируемой программы буфер, в который в дальнейшем по мере анализа вводимых повелительных предложений записываются в символьном виде операторы языка БАЯР.

Анализ повелительного предложения программой сводится к выбору терминального понятия $g_k \in G$, связанного отношением эквивалентности с терминальным понятием $g_l \in G$. Понятия g_k и g_l должны удовлетворять следующим условиям:

g_l – является отображением в семантической сети некоторой последовательности операторов языка БАЯР;

g_k – является отображением в семантической сети словосочетания на естественном языке, включающего в себя максимальное количество слов из анализируемого повелительного предложения.

Соответствующая понятию g_l последовательность операторов помещается в буфер, т.е. в управляющую программу.

При выполнении команды $\emptyset K$ ДИСПОР сохраняет управляющую программу в заданном пользователем файле.

4.3.5. Заключение

Диалоговая система подготовки программ (исследовательская версия) была разработана в конце 80-х годов 20-го века на языке ПАСКАЛЬ в операционной системе RT-11 на машинах типа ДВК в рамках создания гибкого автоматизированного производства конденсаторов наряду с созданием операционной системы транспортного робота и локальной сети для управления гибким автоматизированным модулем [36-38].

Эксперименты с этим ПО показали перспективность и результативность семантически-ориентированного анализа предложений на естественном языке, а также, возможность использования естественного языка для программирования роботов, эффективность применения при распознавании смысла ЕЯ семантических сетей и нейросетевых алгоритмов (пороговой логики при распознавании слов и понятий).

К сожалению, ПО для транспортного робота по известным причинам (это был конец 80-х и начало 90-х годов) так и не удалось внедрить в производство.

4.4. Программное обеспечение для тестирования знаний с использованием ответов на естественном языке

4.4.1. Назначение

Система для контроля знаний (экспертная система ЭСКИЗ) предназначена для контроля знаний студента в некоторой предметной области в режиме само-проверки с использованием естественного языка для ответа на вопросы, задаваемые системой. При этом система проверяет правильность и полноту ответа, не обращая внимание на возможные орфографические или стилистические ошибки. Преподаватель имеет удобные средства для создания и редактирования базы знаний о предметной области, а также, для описания сценария диалога, т.е. последовательности вопросов, реакции на качество ответов, степени "привередливости" системы по отношению к ответам.

Система ЭСКИЗ была реализована в виде исследовательского прототипа на языке PDC-Prolog.

4.4.2. Структура системы

ЭС состоит из следующих компонент:

- основная программа (оболочка экспертной системы), реализующая сценарий диалога и распознавание предложений на ЕЯ – экспертная система для контроля знаний ESLAN;
- программа для создания и редактирования сценария диалога SCENARIO;
- программа для создания и редактирования базы знаний EDKB;
- сценарий диалога;
- база знаний о предметной области, содержащая:
- словарь ЕЯ;
- файл с сетью фреймов, описывающей понятия предметной области и взаимосвязи между ними.

Структура основной программы показана на рис. 4.3.

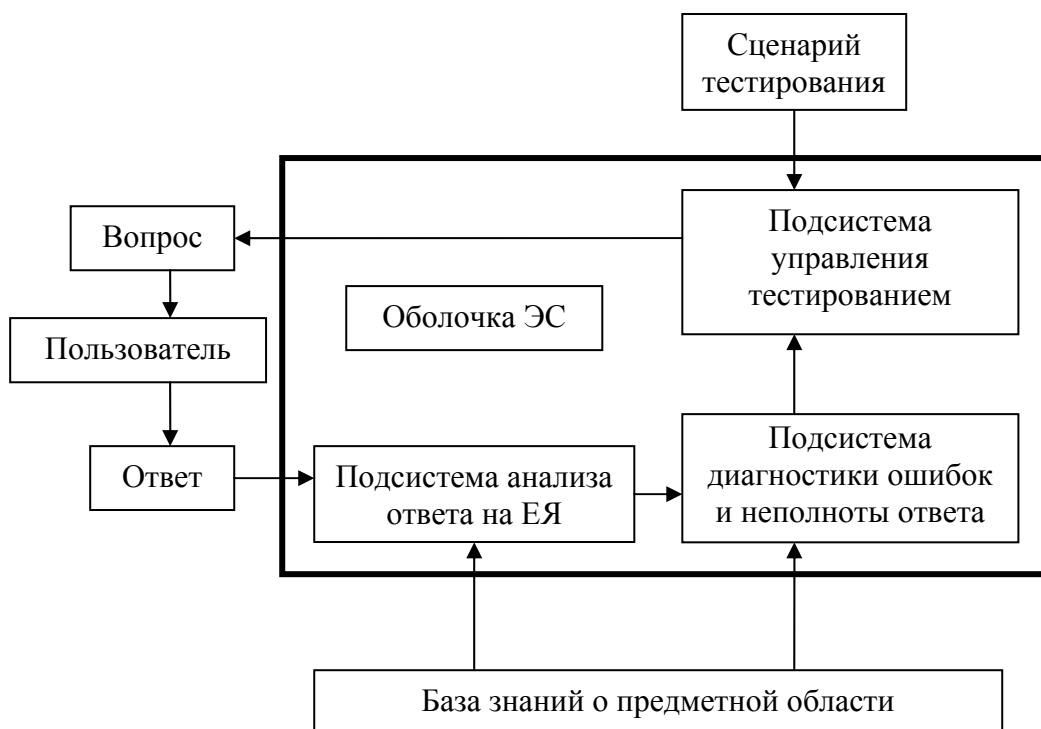


Рис. 4.3. Укрупненная структура оболочки экспертной системы

4.4.3. Типы вопросов, задаваемых системой

Система "ЭСКИЗ" задает следующие типы вопросов, относящиеся к некоторому понятию X:

- 1) Что такое X?
- 2) Перечислите из чего состоит X?
- 3) Какие бывают X? (перечислить элементы данного класса)
- 4) Что является причиной X?
- 5) К какому классу принадлежит X?
- 6) В состав чего входит X?
- 7) Когда было (будет) X?
- 8) Где находится (находился) X?

Каждому из типовых вопросов соответствует своя специализированная процедура сопоставления предложения-ответа и фрейма-понятия, о котором задавался вопрос, и оценки качества ответа. Качество ответа оценивается по трем критериям:

- правильность;
- полнота;
- неизбыточность.

Эти критерии оцениваются в процентах, которые затем учитываются при выставлении оценки – результата тестирования.

4.4.4. Структура базы знаний

Фрейм состоит из имени фрейма (слова или последовательности слов ЕЯ), и набора слотов. Слот состоит из имени слота и значения слота. Слот служит для описания какого-либо свойства (особенности, элемента определения и т.п.) понятия, описываемого данным фреймом. Слот может быть ссылкой на другой фрейм. Имена слотов могут кодироваться любыми словами ЕЯ из словаря. Кроме того, существуют следующие стандартные для данной системы имена слотов:

kind_of	– класс, к которому принадлежит понятие-фрейм;
part_of	– понятие, в состав которого входит понятие-фрейм;
include	– понятие, входящее в состав данного понятия;
obj	– объект, участвующий в описании понятия;
act	– действие, участвующее в описании понятия;
mode	– модальность или свойство понятия;
when	– когда произошло (произойдет) событие;
where	– где находится (был, будет) объект.

В описании объекта может быть несколько слотов с одинаковыми именами и разными значениями. Во внутреннем представлении фрейм представляется предикатом языка PROLOG, а слот является списком из двух элементов – имени и значения слота. Кроме выше перечисленных, существует слот, в котором задаются все возможные значения другого какого-либо слота. Он имеет структуру, отличную от структур других слотов, а именно, является списком произвольной длины, первым элементом которого является слово "menuval", вторым – имя слота, к которому он относится, а следующими – возможные его значения.

Например, фрейм, описывающий понятие "компьютер" может быть описан с помощью набора следующих слотов:

Kind_of:	оборудование;
Kind_of:	инструмент;
Obj:	устройство;
Act:	обработка;
Obj:	информация.

Такой фрейм обеспечивает правильность распознавания ответа на вопрос "что такое компьютер?" в следующем виде – "устройство для обработки информации". В этом ответе вместо слова "устройство" может быть использовано слово "оборудование" или "инструмент". В то же время в базе знаний может присутствовать и фрейм-синоним "компьютер" с другим составом слотов:

Kind_of:	оборудование;
Kind_of:	средство;
Obj:	устройство;
Act:	решение;
Obj:	задачи.

Этому фрейму соответствует правильность ответа "устройство для решения задач".

Фреймы могут быть четырех типов:

- фреймы-примеры, описывающие конкретные понятия или сущности;
- фреймы-прототипы, описывающие абстрактные понятия или классы сущностей, в состав которых могут входить слоты, для которых задаются возможные значения; эти фреймы используются для создания фреймов-примеров, а также, для задания общих свойств, для всех фреймов, классом для которых является данный фрейм;
- фреймы, описывающие словосочетания, обозначающие некоторое понятие, эти фреймы не видны для пользователя;
- фреймы, описывающие абстрактное понятие, не имеющее словесного обозначения и являющееся классом по отношению к набору понятий – синонимов; эти фреймы также не видимы для пользователей.

Перед работой основная программа ЭС загружает фреймы, участвующие в вопросе, в оперативную память и преобразует их в семантическую сеть в виде предикатов Пролога.

Сеть фреймов строится на основе ссылок между ними, которые есть имена фреймов, используемые в качестве значений слотов.

Правильность ответа определяется соответствием между набором распознанных слов и их ролей в ответе и набором слотов (ролей) во фрейме-понятии, о котором был задан вопрос.

Полнота определяется количеством набором совпавших распознанных в ответе слов и слотов во фрейме. Причем в зависимости от типа вопроса разное подмножество слотов используется для определения этого критерия (например, при использовании вопроса "из чего состоит X?" учитываются только слоты "include", при вопросе "что такое X?" – слоты "obj", "act", "mode").

Неизбыточность определяется количеством лишних в ответе слов, которым нет соответствующих ролей в понятии (слотов во фрейме), о котором задавался вопрос.

4.4.5. Словарь ЕЯ

Словарь содержит слова в каноническом виде (т.е. существительные и прилагательные в единственном числе и именительном падеже, а глаголы – в неопределенной форме) и соответствующие им отношения, имена которых соответствуют стандартным именам слотов.

Поиск в словаре слова осуществляется с использованием процедуры, оценивающей меру нечеткого подобия слов, которая обеспечивает игнорирование пропуска или искажения отдельных букв, разницу в окончаниях, приставках или суффиксах.

4.4.6. Программирование сценария диалога

Диалог (порядок задавания вопросов системой и реакция на ответы) программируется с помощью программы SCENARIO, разработанной Ю.В. Новицкой.

Сценарий состоит из шагов, идентифицируемых номером шага NStep. Описание шага сценария состоит из набора предикатов-дескрипторов, каждый из которых имеет одним из параметров соответствующий номер шага (один и тот же для всех дескрипторов данного шага). Дескрипторы могут быть следующих типов:

- вопроса (предикат quest);
- реакции на ответ (предикат ansq);
- подготовки к вопросу (предикат preq);
- подготовки к реакции (предикат preans).

Структура предиката quest – quest(NStep, тип_вопроса, понятие, текст вопроса).

Тип вопроса может иметь следующие значения:

- "what" (что такое);
- "kind_of" (к какому классу принадлежит);
- "include" (из чего состоит);
- "from" (в состав чего входит);
- "reason" (что является причиной);
- "kinds" (какие бывают виды).

Если понятие – пустая строка, то вопрос задается о понятии, о котором задавался предыдущий вопрос (об активном понятии). Если нет – понятие является именем фрейма, который надо загрузить в память. Текст вопроса используется тогда, когда текст типового вопроса не подходит для выбранного понятия, в противном случае текст вопроса="".

Структура предиката ansq – ansq(NStep, условие_реакции, параметр_условия, тип_реакции, параметр_1, параметр_2).

Условие реакции может иметь следующие значения:

- "notfull" (ответ не полный);
- "notright" (ответ не верный);
- "notexact" (ответ не точный или избыточный);
- "full" (ответ полный);
- "right" (ответ верный);
- "exact" (ответ точный или не избыточный).

Параметр условия уточняет в процентах условие реакции. Таким образом, условие может интерпретироваться как " ответ полный на 80 % " – ("full", 80).

Тип реакции может иметь следующие значения:

- "mes" (вывод сообщения);
- "pict" (вывод картинки из файла);
- "go" (перейти на заданный шаг сценария);

- "rep" (повторить данный шаг сценария);
- "kb" (смена файла с базой знаний);
- "clear" (очистка памяти из-под БЗ);
- "dict" (смена файла со словарем);
- "inc" (увеличение требований к ответам);
- "dec" (уменьшение требований к ответу).

Параметр_1 представляет собой число – параметр реакции, например, номер шага.

Параметр_2 представляет собой строковую константу – параметр реакции, например, имя файла, содержащего картинку или сообщение, выводимое на экран.

4.4.7. Заключение

Программное обеспечение ЭСКИЗ для создания и использования учебных тестов было реализовано в середине 90-х годов на языке PDC-Prolog в среде MS DOS. Эксперименты с ним, проведенные на кафедре вычислительной техники НГТУ, показали возможность и перспективность применения естественного языка для тестирования знаний, эффективность использования для распознавания смысла ответа на ЕЯ комбинации фреймового представления знаний и нейросетевых алгоритмов (использование пороговой логики при распознавании слов и при сравнении набора распознанных понятий с набором понятий, описанных в фрейме).

В приложении 2 приводится исходный текст программы ESLAN на языке PDC-Prolog и в приложении 3 – пример базы знаний, сценария диалога и протокола диалога.

Большая часть работы по реализации оболочки ESLAN была проведена А.Петренко в рамках дипломного проектирования, программа для редактирования сценариев SCENARIO была реализована Ю.В.Новицкой.

В настоящее время на кафедре вычислительной техники совместно с компанией "ИНСИКОМ" (Интеллектуальные Системы и Комплексы) под руководством автора ведется работа по применению архитектуры, разработанной в ПО ЭСКИЗ, для реализации механизмов создания и использования учебных тестов с открытым ответом в ПО "ИнтерТест" [40] и ПО Tester (<http://insycom.chat.ru>). В настоящее время реализована демонстрационная версия программного обеспечения "ЭСКИЗ-2" [41] в среде Delphi 5.0.

4.5. Программное обеспечение для поиска документов по смыслу

4.5.1. Постановка задачи

В настоящее время весьма бурно развивается направление информатики, связанное с разработкой систем, облегчающих работу с большим количеством разных документов (текстовых и графических). Главной проблемой при этом является проблема облегчения поиска нужного по содержанию документа. Она решается при разработке систем документооборота на предприятии и поиска информации в сети Internet.

Ниже описаны разработанные автором архитектура программного обеспечения и исследовательский прототип ПО для решения этой задачи на основе принципов распознавания смысла, предложенных выше (см. 4.2).

4.5.2. Архитектура программного обеспечения для обучения на основе обработки документов и для поиска документов по смыслу [39]

На рис. 4.4 показана функциональная структура системы обработки запросов к документам по их содержанию, которая взята за основу при разработке демонстрационной версии программы.

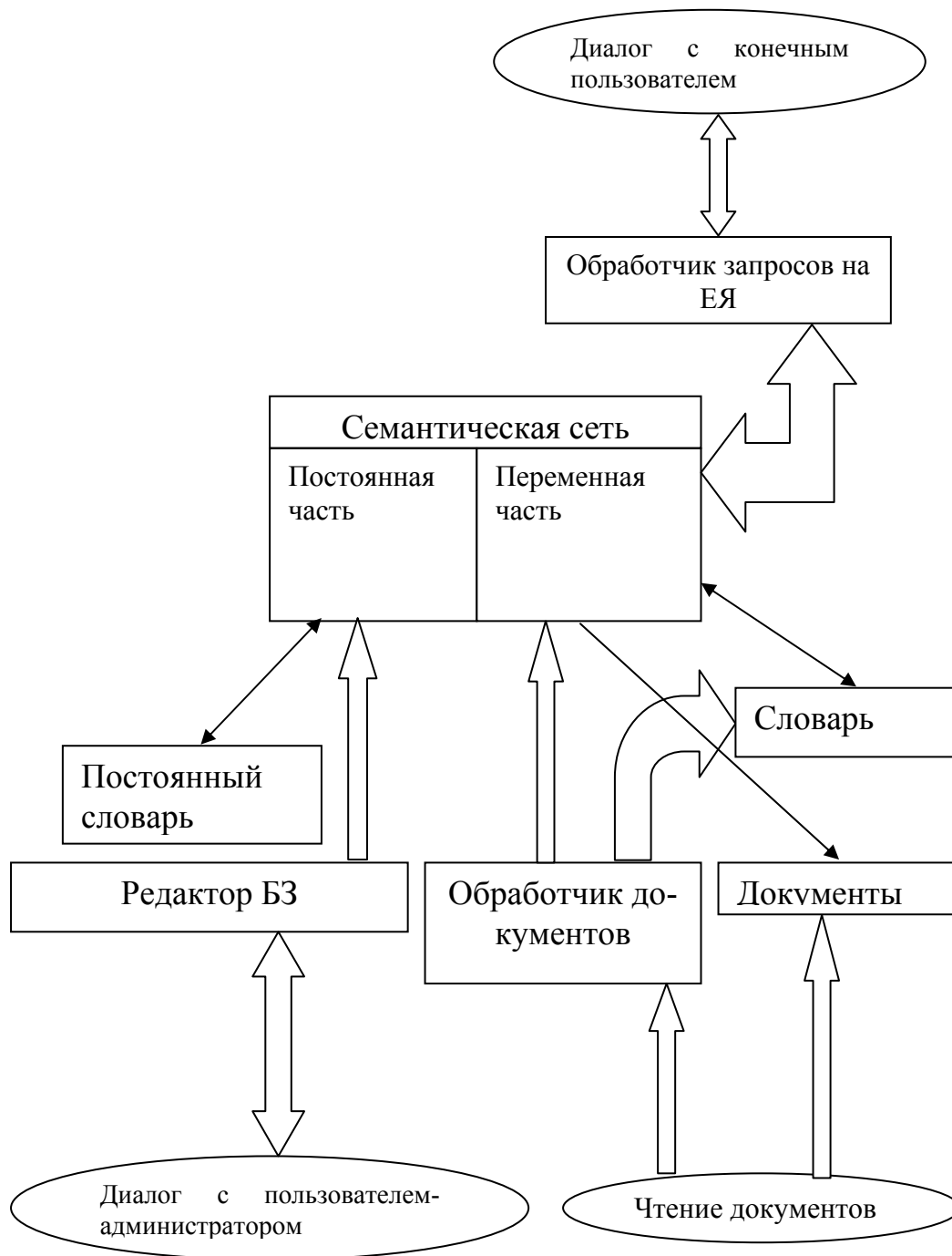


Рис. 4.4. Функциональная структура системы, ориентированной на поиск документов по запросу на ЕЯ.

База знаний (БЗ) представляет собой семантическую сеть фреймов. Правда, фреймами узлы сети названы условно, т.к. они являются их вырожденным случаем и напоминают узлы семантической сети

Условно в базе знаний можно выделить постоянную часть, описывающую предметную область, и переменную часть, описывающую содержимое документов.

Переменная часть формируется в результате обработки документов под управлением постоянной части.

С фреймом может ассоциироваться высказывание или набор высказываний с одинаковым смыслом, а также, слово из словаря или документ. Соответственно, одной из характеристик фрейма является его уровень в сети:

0 – фрейм, связанный непосредственно со словом или документом (фрейм-слово или фрейм-документ),

1 – фрейм, с которым ассоциируется словосочетание (составной фрейм),

2 – фрейм-понятие, включающее в себя ссылки на нескольких других фреймов, играющих определенные роли в этом понятии,

3 – фрейм-рубрика, описывающий понятие, которое является «описанием» (ссылкой, классом) всех понятий и документов, связанные с этой рубрикой,

Особая разновидность фреймов уровня 0 – фреймы для связи с процедурами, используемыми при анализе предложений ЕЯ. Такие фреймы содержат знак препинания или слово, которое обрабатывается особым образом (например, слово - «не», а знаки препинания – тире, двоеточие, запятая, и точка с запятой).

Фрейм состоит из слотов. Каждый слот – это его имя и значение.

Значением слота является ссылка на другой фрейм, на слово в словаре или на документ.

Во фрейме существуют (но не все из них используются в конкретном фрейме) следующие слоты:

- Parent – ссылка на фрейм-родитель или класс,
- Owner – список ссылок на фреймы-понятия или составные фреймы, в состав которых входит данный фрейм,
- Obj – объект, участвующий в понятии,
- Subject – субъект (или главный объект), участвующий в понятии,
- Act – действие, участвующее в понятии,
- Prop – свойство, участвующее в понятии,
- Equal – список понятий-синонимов, ассоциативно связанных с понятием, описанным в данном фрейме,
- Unequal – список понятий-антонимов, ассоциативно связанных с понятием, описанным в данном фрейме,
- Place – место действия,
- Time – время действия,
- Include – список ссылок на фреймы, включенные в данное понятие (являющиеся его составной частью),

Кроме того, фрейм включает в себя следующие параметры:

- Level – уровень фрейма,
- DocName – имя файла (путь) с документом, связанным с фреймом,

- IndWord – индекс слова в словаре, связанном с фреймом,
- Н – порог срабатывания фрейма, как нейрона при распознавании понятия,
- Role - роль фрейма в понятии, в которое он входит или может входить (А-действие, О-объект, S-субъект, Р-свойство, U-неопределена или D – действие при анализе (вызывается процедура),
- NO – признак инверсии (отрицания) фрейма.

В системе используется два словаря:

- базовый, в котором хранятся слова с их ролями (сущность, действие или свойство, другими словами – существительное, глагол или прилагательное,
- пополняемый (динамический) словарь, включающий в себя слова, не распознанные в базовом словаре, т.е. либо отсутствующие в нем (например, имена собственные) либо присутствующие в предложении в слишком искаженном виде или в другой форме, сильно отличающейся от формы в словаре.

Кроме того, некоторые слова (назовем их специальными) из этих двух словарей могут быть связаны (отождествлены) с символами-разделителями (пробел, тире, запятая). Они заменяются на соответствующие знаки при предварительной обработке предложения. Это позволяет фильтровать ненужные с точки зрения анализа смысла (отождествляя их с пробелами) и структурировать длинные словосочетания и процесс их анализа путем замены таких слов как «и» или «или» на запятые, «это» или «равно» - на тире, «состоит» – на двоеточие и т.п.

Ниже приводится описание структуры фрейма в виде класса в Delphi.

```
Tframe=Class(TObject)
    Parent, Act, Prop, Subject,
    Obj, Time, Place: TLink;
    IndWord: integer;    //ссылка на слово из словаря
                        //если -1 - нет ссылки на
                        // слово в словаре
                        // если > 1000000, то это
                        // индекс слова в VocUniq
    NameDoc: String;    // имя файла, содержащего документ
                        //если '' - нет ссылки на документ
                        // и при Level=0 это признак того,
                        // что это фрейм,
                        //      связанный со словом
    Level:shortint;    // уровень узла,
                        // 0 - связан непосредственно со
                        // словом или документом
                        // 1 - составной фрейм (из слов)
                        // 2 - фрейм-понятие
                        // 3 - фрейм-рубрика (пока не ис-
пользуется)
                        // 4 - фрейм-база знаний (пока не
используется)
```

```

    Owner: TListLink;    // список владельцев (вхождений)
    Equal: TListLink;    // список синонимов
    Include: TListLink;  // список включенных в данный
фрейм других фреймов
    H:shortint;          // порог
    OnSet:boolean;       // включен-выключен
    Mark:boolean;
    Role:TRole;          // Роль фрейма в понятии
                        // U - не определена
                        // A - действие
                        // S - субъект
                        // P - свойство
                        // D - действие при анализе (процедура)

    constructor Create(L:integer); virtual;
    destructor Destroy; override;
end;

constructor TFrame.Create(L:integer);
begin
    H:=3;
    OnSet:=False;
    Level:=L;
    Equal:=TListLink.Create;
    Owner:=TListLink.Create;
    Include:=TListLink.Create;
    Parent.Adr:=-1;
    Time.Adr:=-1;
    Place.Adr:=-1;
    Act.Adr:=-1;
    Obj.Adr:=-1;
    Prop.Adr:=-1;
    Subject.Adr:=-1;
    IndWord:=-1;
    NameDoc:='';
    Role:=U;
end;

destructor TFrame.Destroy;
begin
    Owner.Destroy;
    Equal.Destroy;
    Include.Destroy;
    inherited Destroy;
end;

```

4.5.3. Алгоритмы

Обработка предложения в режиме обучения состоит из следующих этапов:

- 1) выделение слов (используя знаки препинания и пробелы),
- 2) распознавание слов по максимальной схожести со словами в словаре, при этом если подходящее слово не находится в основном словаре, происходит его поиск в пополняемом словаре, и, если и там его нет, происходит его добавление в этот словарь.
- 3) создание фреймов, связанных с распознанными словами и знаками препинания (уровня 0), результатом распознавания предложения является объект-предложение, описывающий предложение и представляющий собой список фреймов
- 4) замена в этом объекте специальных слов знаками-разделителями,
- 5) обработка объекта-предложения процедурой распознавания-порождения фреймов.

Процедура распознавания-порождения фреймов рекурсивно обрабатывает объект-предложение, заменяя словосочетания между знаками препинания на распознанные в базе знаний или вновь созданные фреймы. В конце обработки этой процедурой список фреймов в объекте-предложении включает только один или два фрейма (в последнем случае в предложении присутствовал хотя бы один знак препинания "-" или двоеточие).

Фрейм-понятие создается на основе атрибутов (из словаря) в распознанных словах (это возможно при наличии, троек «субъект»-«действие»-«объект» или «объект-действие-объект» (в последнем случае во фрейме-понятии первый объект объявляется субъектом) или пар «объект»-«свойство», «объект-действие».

Фрейм-понятие или составной фрейм создаются только, если соответствующий фрейм не найден в базе знаний (операции 6 и 7).

При окончании обработки в случае, если в предложении есть тире, происходит создание связей типа Equal между фреймами - двумя частями предложения.

Если тире в предложении нет, создается составной фрейм из всех фреймов предложения. Последний созданный или распознанный фрейм запоминается как контекст. Если предложение начинается с тире, фрейм, соответствующий ему (распознанный или созданный) связывается связью Equal с фреймом-контекстом.

При создании составного фрейма или фрейма-понятия создается фрейм-документ и связь типа equal между ним и созданным фреймом.

При обработке фрейма-действия со словом «не» происходит инверсия следующего за ним фрейма-слова или в случае, если «не» встретилось в начале предложения или сразу после тире, происходит инверсия соответствующего составного фрейма или фрейма-понятия.

Фрейм-рубрика должен задаваться перед обработкой документа.

Обработка предложения в режиме обработки запроса состоит из следующих этапов:

- 1) выделение слов (используя знаки препинания и пробелы),
- 2) распознавание типа предложения по последнему знаку препинания: точка – повествовательное, ! – восклицательное, ? – вопросительное (результат распознавания пока не используется),
- 3) распознавание слов по максимальной схожести со словами в словаре, при этом если подходящее слово не находится в базовом словаре, происходит его поиск в пополняемом словаре, и, если и там его нет, происходит его добавление в этот словарь. В последнем случае система задает вопрос “что такое <введенное новое слово>?” для уточнения его смысла и увязывания его с другими словами или понятиями. Ответ пользователя обрабатывается в режиме обучения.
- 4) создание фреймов, связанных с распознанными словами и знаками препинания (уровня 0), результатом распознавания предложения является объект-предложение со списком фреймов,
- 5) распознавание фреймов уровня 1 – словосочетаний в базе знаний, максимально похожих на распознанный запрос (фразу) и фреймов-понятий уровня 2 (здесь используется нейросетевой алгоритм, т.е. взвешенное суммирование сигналов от входящих во фрейм слов или фреймов и сравнение с порогом),
- 6) поиск ассоциативно связанных ссылками Equal с распознанными фразами фреймов (уровня 0), связанных с документами,
- 7) поиск фреймов-документов из найденных фреймов по связям типа include, act, obj, subject, prop сверху вниз, при этом, если документов очень много, система выдает сообщение с просьбой переформулировать запрос.
- 8) вывод найденных имен документов или слов, входящих в состав найденных фреймов.

Фреймы-рубрики (уровня 3) планируется использовать для сокращения количества найденных фреймов. При этом выбираются для последующих операций только те фреймы, которые связаны через ссылку Parent с соответствующим фреймом-рубрикой, или у которых эта ссылка не определена. Фрейм-рубрика может задаваться в программе поиска в меню выбора рубрики или при нахождении соответствующего слова (словосочетания) в распознаваемом предложении-запросе.

4.5.4. Обучение

Рекомендуется обучать программу (создавать базу знаний) в следующей последовательности:

Первоначальное обучение распознаванию структуры предложения заданием предложений в виде <слово> - @<символ>. После этого программа будет перед обработкой предложения при последующем обучении заменять заданное слово на заданный символ-разделитель. В качестве символа-разделителя могут быть пробел, тире, двоеточие, запятая, точка с запятой. Пробел озна-

чает, что данное слово будет исключаться из анализа семантики предложения. При этом обучении создается словарь специальных слов.

Начальное обучение. При этом задаются основные понятия из житейской практики или предметной области в виде предложений типа «деньги – средство платежа», «мораль – правила поведения», «виды деятельности : торговля, производство, услуги» и т.п.

Базовое обучение. При этом программе «скармливается» толковый словарь предметной области, где объясняются понятия в предложениях вида <понятие> - <объяснение, что это или для чего, какие бывают и т.п.>.

Информационное наполнение. При этом программе "скармливаются" документы, в которых она должна искать ссылки при обработке запросов.

Этапы 1-3 необходимы для лучшей структуризации базы знаний и обеспечения достаточно большого процента распознаваемых понятий при обработке документов на этапе 4. Иначе, при обработке запроса программа часто будет не находить в запросе какого-либо известного ей понятия, выполнять поиск по словам, входящим в запрос.

После каждого этапа можно сохранять полученную базу знаний и для продолжения обучения загружать сохраненную ранее.

4.5.5. Исследовательский прототип программного обеспечения.

В настоящее время разработана исследовательская версия программного обеспечения, состоящая из двух программ:

- Программы Alang для создания и отладки базы знаний о содержимом документов,
- Программы Finder для поиска документов по смыслу конечным пользователем.

Пример запроса, который может обрабатываться разработанной системой поиска документов по смыслу.

Запрос: «Разработчики экспертных систем».

Возможные варианты содержимого документов, удовлетворяющих данному запросу.

Содержимое документа 1: «Наши разработки: ... экспертная оболочка ESWin»

Содержание документа 2: «Наша фирма разработала советующую систему на основе базе знаний ...»

Содержание документа 3: «Вы можете заказать у нас разработку экспертной системы для ...»

Содержание документа 4: «Фирма занимается разработкой программного обеспечения..... Наши продукты -инструментальная экспертная система ESWin»

Содержание документа 5: «Наши продукты.... Программа экономического анализа D ... знания о рынке представлены в виде семантической сети (или правил, фреймов и т.п.)....».

Для того, чтобы система могла обрабатывать запрос, как описано выше, в режиме обучения ей достаточно встретить в документах или обработать в диалоге следующие предложения:

- 1) экспертная система - советующая система,
- 2) методы представления знаний в экспертных системах: фреймы, семантические сети, правила, лингвистические переменные.
- 3) Экспертная оболочка это инструмент для создания и отладки экспертных систем.

Исследовательская версия разработана в среде Delphi 5, требует RAM не менее 64 Мб, скорость процессора не менее 300 Мгц. База знаний в ней хранится в текстовом виде и загружается целиком при запуске программы.

Программное обеспечение было испытано на компьютере с процессором AMD K6-2 350 Мгц и RAM 64 Мб. При этом база знаний была создана на основе предварительного диалога из порядка 40 обучающих предложений и обработки следующих документов в формате .txt:

- толковый словарь по экономике (880 Кб),
- гражданский кодекс (550 Кб),
- уголовный кодекс (360 Кб),
- набор документов, регулирующих авторские права (80 Кб),
- правила дорожного движения (70 Кб),
- набор файлов со статьями из газеты "Коммерсант" (100Кб),
- набор файлов со статьями о нейронных сетях (около 100 Кб).

Созданная БЗ на носителе занимала около 8Мб (в текстовом виде) и содержала 87513 фреймов и 14715 слов в пополняемом словаре. Обработка в режиме обучения всех этих документов на указанном компьютере заняла около двух часов времени.

Ниже приведены главные формы программ Alang и Finder.

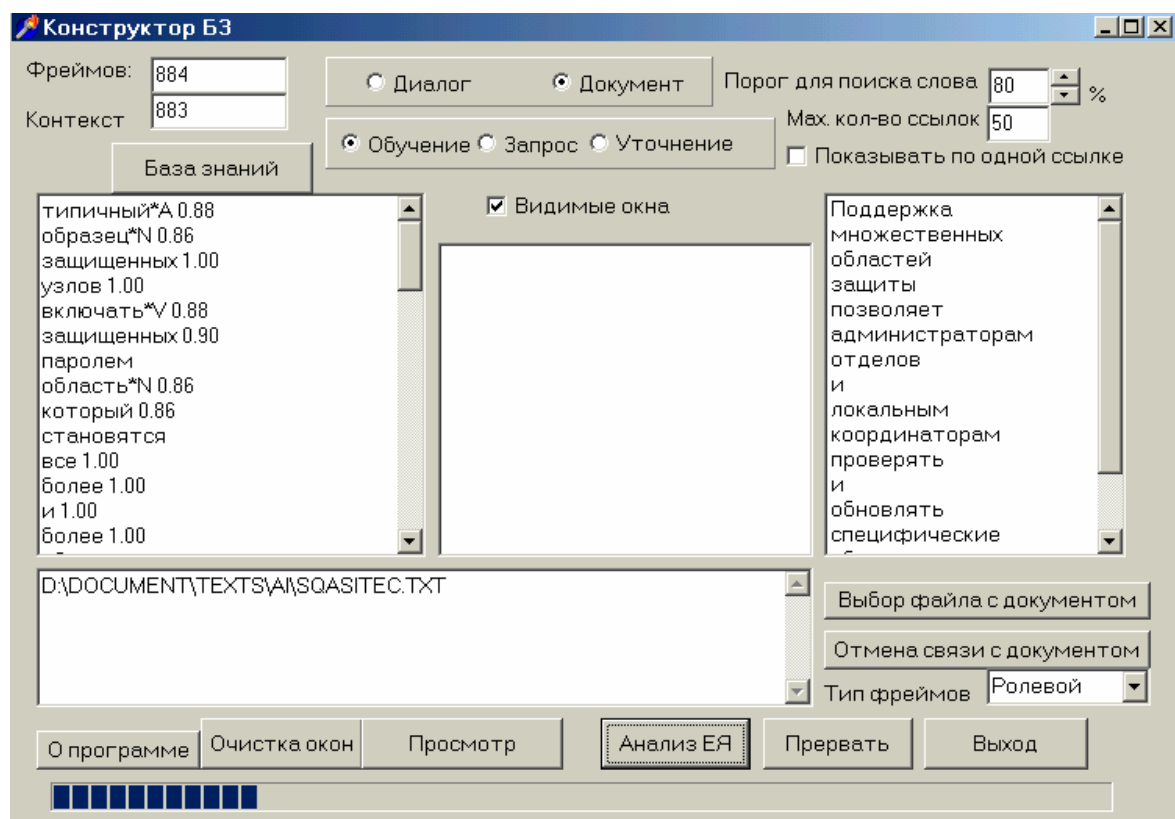


Рис. 4.5. Программа Alang в процессе обработки документа.

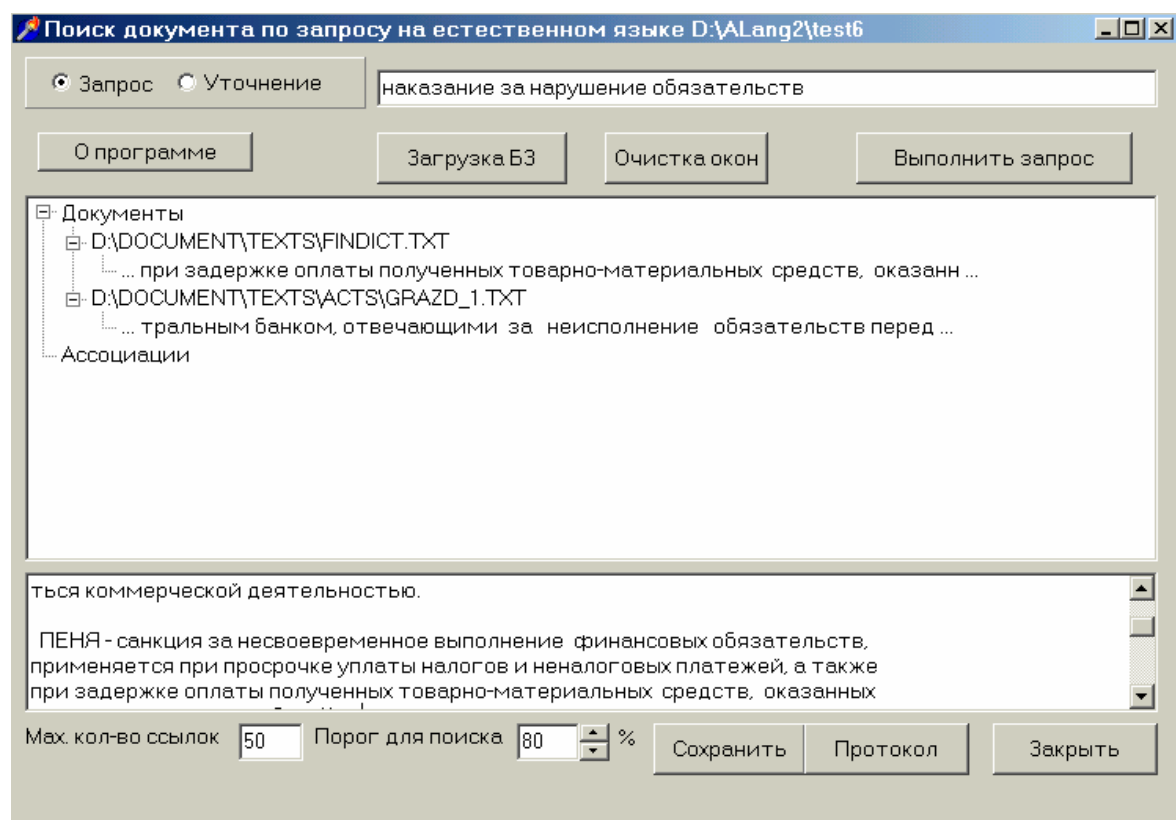


Рис. 4.6. Программа Finder.

На последнем рисунке показан результат обработки запроса "наказание за нарушение обязательств". При этом было найдено две ссылки - на статью "Пеня" в толковом словаре и на фрагмент 1-й части гражданского кодекса.

4.5.6. Заключение

Предварительные испытания показали, что предложенные принципы и алгоритмы позволили построить систему, осуществляющую поиск документов и их фрагментов в ответ на запрос на естественном языке с довольно не тривиальными результатами. В найденных фрагментах могли отсутствовать слова запроса. Причем качество ответов сильно зависит от качества обучающего материала на этапах 1, 2 и 3 (см. выше).

Результаты испытаний позволили наметить пути дальнейшего усовершенствования предложенной архитектуры:

- ввести обработку типовых вопросов о содержимом базы знаний, отражающей содержание документов,
- ввести средства синтеза ответов на естественном языке, используя фреймовое описание понятий,
- использование какой-либо объектно-ориентированной СУБД для хранения базы знаний, например, СУБД ESF, разработанной компанией NooLab (<http://www.noolab.ru>).

В настоящее время разработанная архитектура применяется в компании "ИНСИКОМ" (Интеллектуальные Системы и Комплексы) (<http://insycom.chat.ru>) при разработке ПО для поддержки работы с документами.

4.6. Выводы

В главе формулируются принципы распознавания смысла предложений на естественном языке, объединяющие в себе использование семантических сетей и нейроподобных алгоритмов.

Приводится описание ряда прикладных систем, разработанных автором, в которых разработаны и реализованы оригинальные алгоритмы распознавания слов и предложений естественного языка, основанные на этих принципах:

- диалоговой системы программирования транспортных роботов на естественном языке,
- системы тестирования знаний с ответами на естественном языке,
- системы для формирования базы знаний о содержимом документов и для поиска документов по запросу на естественном языке.

Эксперименты с разработанными системами показали эффективность и перспективность использования предложенных принципов и алгоритмов.

ЗАКЛЮЧЕНИЕ

В настоящее время процесс "гибридизации" прикладных систем ИИ, которому посвящена настоящая монография, является, пожалуй, основной тенденцией в развитии искусственного интеллекта. В монографии рассматриваются вопросы комбинирования разных методов представления и обработки знаний в гибридных интеллектуальных системах, решаемые в рамках исследований и разработок автора за последние примерно 25 лет.

В главе 1 предлагаются некоторые теоретические концепции, которые могут лечь в основу теории интеллектуальных систем, а именно:

- модель процесса мышления, объединяющая в себе вербальное и образное, осознанное и бессознательное мышление, которая является отражением "гибридности" естественных интеллектуальных систем,
- модель ассоциативного мышления и связанную с ним концепцию нечеткого подобия,
- принципы организации функционирования интеллектуальных систем,
- подход к количественной оценке объема знаний, содержащихся в сообщении, передаваемом между интеллектуальными системами,
- связь предлагаемых моделей с моделированием эмоций.

В главе 2 рассмотрены некоторые вопросы построения гибридных экспертных систем и описана инструментальное ПО ESWin для создания гибридных экспертных систем.

В главе 3 рассмотрены различные варианты использования нейронных сетей и нейроподобных алгоритмов в прикладных системах ИИ и примеры архитектур таких систем, разработанных автором.

В главе 4 предлагаются принципы создания гибридных интеллектуальных систем, понимающих естественный язык, и описаны примеры систем, построенных на этих принципах.

Таким образом, в монографии представлены разработками автора почти все основные классы прикладных систем искусственного интеллекта. Однако, за пределами рассмотрения оказались некоторые популярные и бурно развивающиеся направления ИИ, такие как генетические алгоритмы и эволюционное моделирование, мобильные автономные агенты (в том числе роботы) и распределенный интеллект. Эти направления остались не освещенными в монографии по той причине, что у автора нет достойных внимания разработок в этих областях, хотя они тоже имеют определенное отношение к гибридным интеллектуальным системам. Так, например, вариантами гибридных интеллектуальных систем можно считать применение генетических алгоритмов для обучения нейронных сетей и мобильных роботов, а при создании распределенного интеллекта гибридный подход напрашивается сам собой в силу относительной автономности составляющих его компонент и их возможной специализацией на выпол-

нение определенных функций и, соответственно, обладающих особенностями парадигм искусственного интеллекта, используемых в них.

Предложенные автором в монографии концепции и варианты реализации парадигм гибридных интеллектуальных систем могут быть использованы при построении так называемого "настоящего искусственного интеллекта" (или искусственного разума), объективная необходимость в котором по мнению автора существует и будет только усиливаться в процессе развития информационных технологий. К этому есть две объективные предпосылки:

- 1) ИИ должен мыслить как человек для того, чтобы человек получил полноценного помощника, с которым он бы мог общаться как с человеком,
- 2) ИИ должен иметь возможность неограниченно обучаться и совершенствоваться, так как сложность создания и совершенствования информационных технологий с каждым годом возрастает, и требуется все больше специалистов, имеющих все большую квалификацию, и быстро (в соответствии с темпами развития информационных технологий), чего не может и вряд ли сможет обеспечить система образования.

Здесь не рассматриваются этические и прочие проблемы будущего развития человечества, связанные с появлением такого ИИ. Эта тема для отдельной монографии с философским и футурологическим уклоном.

ЛИТЕРАТУРА

Литература к введению

1. *Russell S.J., Norvig P.* Artificial Intelligence: A modern approach. Prentice Hall, 1995.
2. *Винер Н.* Кибернетика или управление и связь в животном и машине. – М.: Советское радио, 1968.
3. *Лорьер Ж.-Л.* Системы искусственного интеллекта. – М: Мир, 1991.
4. *Попов Э.В.* Экспертные системы. – М.: Наука, 1987.
5. Построение экспертных систем. Под ред. Ф. Хейес-Рота, Д. Уотермена, Д. Лената. – М.: Мир, 1987.
6. Экспертные системы. Принципы работы и примеры. Под ред. Р. Форсайта. – М.: Радио и связь, 1987.
7. Экспертные системы для персональных компьютеров: методы, средства, реализации: Справочное пособие/ В.С. Крисевич, Л.А. Кузьмич и др. – Мн.: Выш. шк., 1990.
8. Элти Дж., Кумбс М. Экспертные системы. Концепции и примеры. – М.: Финансы и статистика, 1987.
9. Статические и динамические экспертные системы. Э.В., Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. – М.: Финансы и статистика, 1996.
10. *Джексон П.* Введение в экспертные системы. – М., СПб., Киев: Вильямс, 2001.
11. *Заде Л.* Понятие о лингвистической переменной и его применение к принятию решений. – М.: Мир, 1976.
12. *Аверкин А.Н., Батыршин И.З., Блишун А.Ф. и др.* Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д.А. Поспелова. – М.: Наука, 1986.
13. *Нариньяни А.С.* Недоопределенность в системах представления и обработки знаний // Изв. АН СССР. Техн. кибернетика. – 1986. – №5. – С. 3-28.
14. Нечеткие множества и теория возможностей. Под ред. Р. Ягера. – М.: Радио и связь, 1986.
15. *Дюбуа Д., Прад А.* Теория возможностей. Приложения к представлению знаний в информатике. – М.: Радио и связь, 1990.
16. Мелихов А.Н., Бернштейн Л.С., Коровин С.Я. Ситуационные советующие системы с нечеткой логикой. – М.: Наука, 1990.
17. *Малышев Н.Г., Берштейн Л.С., Боженик А.В.* Нечеткие модели для экспертных систем в САПР. – М.: Энергоатомиздат, 1991.
18. Обработка нечеткой информации в системах принятия решений / А.Н. Борисов, А.В. Алексеев, и др. – М.: Радио и связь, 1989.
19. *Трахтенгерц Э.А.* Неопределенность в моделях компьютерных систем поддержки принятия решений. Ч. 2 – Новости искусственного интеллекта, 2002, № 1. – С. 14-20.
20. *Кандрашина Е.Ю., Литвинцева А.В., Поспелов Д.А.* Представление знаний о времени и пространстве в интеллектуальных системах. – М.: Наука, 1989.
21. *Эйкинс Я.С.* Знание, организованное в виде прототипов, для экспертных систем. – / В сб. "Кибернетический сборник", вып. 22. – М.: Мир, 1985. – С. 221- 277.
22. *Petrov VV; Pavlova N.V.* Multi-method organization in Hybrid Expert Systems / Doklady Akademii Nauk. 1996, V. 350, № 4, Oct. - p. 465-466.
23. *Tabachneck-Schijf H.J.M., Leonardo A.M., Simon H.A.* CaMeRa: A Computational model of Multiple Representations. – Carnegie Mellon University, Pittsburgh.
24. *Гаврилов А.В., Новицкая Ю.В.* Инструментальное программное обеспечение для создания гибридных экспертных систем. – Доклады Межд. науч.-техн. конф. "Информационные системы и технологии" ИСТ-2000, Новосибирск, НГТУ, 2000. – Т. 3, с. 488-490.

25. Гаврилов А.В., Новицкая Ю.В. Архитектура гибридной экспертной системы. – Межд. симп. "ИНПРИМ-2000" – Новосибирск, 2000.
26. Гаврилов А.В., Новицкая Ю.В. Инструментальное программное обеспечение для создания гибридных экспертных систем – Рег. научно-практич. конф. "АГРОИНФО-2000". – Новосибирск, 2000. – С.142.
27. Nikitenko A.V., Grundspenkis J.A. The kernel of hybrid intelligent system based on inductive, deductive and case based reasoning / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение", СПб, 2001. – Т.2, с. 496-500.
28. Gavrilov A.V., Novickaja J.V. The Toolkit for development of Hybrid Expert Systems. – 5-th Int. Symp. "KORUS-2001". – Tomsk: TPU, 2001. – Proceedings. – Vol.1. – Pp. 73-75.
29. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб: Питер, 2000.
30. Смирнов А.В., Пашкин М.П., Шилов Н.Г., Левашова Т.В. Онтологии искусственного интеллекта: способы построения и организации. – Новости искусственного интеллекта, 2002, № 1. – С. 3-13.
31. Yang J., Pai P., Honavar V., Miller L. Mobile Intelligent Agents for Document Classification and Retrieval: A Machine Learning Approach.
32. Загоруйко Ю.А., Попов И.Г., Костов Ю.В., Сергеев И.П. Общая концепция агентов в системе моделирования Semp-A1 / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение". – СПб, 2001. – Т.1, с. 259-267.
33. Hebb D.O. The organization of behaviour. – N.Y.: Wiley, 1949.
34. Розенблатт Ф. Принципы нейродинамики (перцептрон и теория механизмов мозга) // М.: "Мир", 1965.
35. Радченко А.Н. Моделирование основных механизмов мозга. – Л.: Наука, 1968.
36. Позин Н.В. Моделирование нейронных структур. – М.: Наука, 1970.
37. Арбиб М. Метафорический мозг. – М.: Мир, 1976.
38. Кохонен Т. Ассоциативная память. – М.: Мир, 1980.
39. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities// Proc. Nat. Acad. Sci. USA. – 1982. – 79. – p. 2554-2558.
40. Васильев В.И. Распознающие системы. Справочник. – Киев: Наукова думка, 1983.
41. Honavar V., Uhr L. Brain-structured connectionist networks that perceive and learn // Connection Science, 1989, 1. – Pp. 139-159.
42. Горбань А.Н. Обучение нейронных сетей. – М.: СП Парграф, 1990.
43. Gavrilov A.V. An Architecture of Neurocomputer for Image Recognition / – Neural Network World, N.1, 1991. – Pp. 59-60.
44. Гаврилов А.В. Модель нейроподобной системы – В сб. "Локальные вычислительные сети" под ред. А.А. Малявко, Новосибирск: НЭТИ, 1991.
45. Carpenter G., A., Grossberg S. Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MA, MIT Press, 1991.
46. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. – М.: Мир. 1992.
47. Борисюк Г.Н. и др. Осцилляторные нейронные сети. Математические результаты и приложения. – Математическое моделирование, 1992, т. 4, №1.
48. Горбань А.Н., Россиев Д.А.. Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996.
49. Гроссберг С. Внимательный мозг. – Открытые системы, 1997, № 4.
50. Галушкин А. И. Современные направления развития нейрокомпьютерных технологий в России. – Открытые системы, 1997, № 4. – С. 25-28.
51. Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей. – Сибирский журнал вычислительной математики. – 1998. – Т. 1, № 1. – С. 12-24.

52. Нейроинформатика. / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. – Новосибирск: Наука. Сибирское предприятие РАН, 1998. – 296 с.
53. *Zhdanov A.A.*. The Mathematical Models of Neuron and Neural Network in Autonomous Adaptive Control Methodology. – / WCCI'98/IJCNN'98 Proceedings, IEEE World Congress on Computational Intelligence, Anchorage, Alaska, May 4-9, 1998. pp. 1042-1046.
54. *Галушкин А.И.* Теория нейронных сетей. – М.: ИПРЖР, 2000.
55. *Галушкин А.И.* Нейрокомпьютеры. – М.: ИПРЖР, 2000.
56. *Круглов В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. – М.: Горячая линия-Телеком, 2001.
57. *Каллан Р.* Основные концепции нейронных сетей. – М.: "Вильямс", 2001.
58. *Комарцова Л.Г., Максимов А.В.* Нейрокомпьютеры. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.
59. *Куссуль Э.М.* Ассоциативные нейроподобные структуры. – Киев, Наукова думка, 1990.
60. *Дорогов А.Ю.* Модальные категории модульных нейронных сетей. // Проблемы нейрокибернетики (материалы XII Межд. конф. по нейрокибернетике). – Ростов-на-Дону, 1999. – С. 137-141.
61. *Гаврилов А.В.* Об одной архитектуре экспертных систем / Вс. конф. "Освоение и концептуальное проектирование экспертных систем", ч. 2, М., 1989. – С. 98.
62. *Gavrilov A.V.* The Model of mind / Int. Symp. "BIOMOD-92", S.-Peterburg, 1992.
63. *Гаврилов А.В.* Архитектура "двухполушарной" экспертной системы. // В кн.: Системы искусственного интеллекта: Межвуз. сб. науч. трудов под ред. А.В. Гаврилова. – Новосибирск: НГТУ, 1993. – С.10-14.
64. *Гаврилов А.В.* Архитектура экспертной системы для работы в реальном времени. – 2-й Межд. сем. "Интеллектуализация баз данных". – Киев, 1993.
65. *Honavar V.* Toward Learning Systems That Use Multiple Strategies and Representations. In: Artificial Intelligence and Neural Networks: Steps Toward Principled Integration. Honavar, V. and Uhr, L. (Ed.) New York: Academic Press, 1994. – Pp. 615-644.
66. *Honavar V.* Symbolic Artificial Intelligence and Numeric Artificial Neural Networks: Toward a Resolution of the Dichotomy. Invited chapter. In: Computational Architectures Integrating Symbolic and Neural Processes. Sun, R. and Bookman, L. (Ed.) N. Y.: Kruwer, 1994. – Pp. 351-385.
67. *Gavrilov A.V., Novickaja J.V.* The Expert Shell based on the Artifitial Neural Networks / Int. Conf. NITS'94. – Penza, 1994.
68. *Funobashi M., Moeda A., Morooka. Y., Mori K.* Fuzzy and Neural Hybrid Expert Systems: Sinergetic AI. – AI in Japan, IEEE, 1995, august. – Pp. 33-40.
69. *Kandel A., Schneider M.* Fuzzy intelligent hybrid systems and their applications. – 1995 IEEE. – Pp. 2275-2280.
70. *Shastri L., Wendelken C.* Seeking coherent explanations --- a fusion of structured connectionism, temporal synchrony, and evidential reasoning. – Proceedings of Cognitive Science 2000. Philadelphia, PA, August 2000.
71. *Гаврилов А.В., Новицкая Ю.В.* Архитектура "двухполушарной" экспертной системы / В межвуз. сб. "Кибернетика и ВУЗ. Интеллектуальные информационные технологии". Вып. 28. – Томск, 1994.
72. *Гаврилов А.В., Новицкая Ю.В.* Вопросы построения экспертных систем на основе нейронных сетей / 3-й Сибирский конгресс "ИНПРИМ-98", Ч. 5, Новосибирск, 1998. – С. 73.
73. *Рыбина Г.В.* Современные экспертные системы: тенденции к интеграции и гибридизации. – Приборы и системы. Управление, контроль, диагностика. – 2001, № 8. – С. 18-21.
74. *Gavrilov A.V., Novitskaya J.V.* The Architecture of the Hybrid Expert System / The 6-th Russian-Korean International Symposium on Science and Technology. Materials. – Novosibirsk, 2002. – Vol. 3. – P. 70.

Литература к главе 1

1. Логический подход к искусственному интеллекту. – М.: Мир, 1990.
2. Ковальски Р. Логика в решении проблем. – М.: Наука, 1990.
3. Модальные и интенциональные логики и их применение к проблемам методологии науки. Под ред. В.А. Смирнова. – М.: Наука, 1984.
4. Нечеткие множества в моделях управления и искусственного интеллекта/А.Н. Аверкин, И.З. Батыршин, А.Ф. Блишун, В.Б. Силов, В.Б. Тарасов. Под ред. Д.А. Поспелова. – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 312 с.
5. Нечеткие множества и теория возможностей. Под ред. Р. Ягера. – М.: Радио и связь, 1986.
6. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний в информатике. – М.: Радио и связь, 1990.
7. Заде Л. Понятие о лингвистической переменной и его применение к принятию решений. – М.: Мир, 1976.
8. Нильсон Н. Принципы искусственного интеллекта. – М.: Радио и связь, 1985.
9. Ефимов Е.И. Решатели интеллектуальных задач. – М.: Наука, 1982.
10. Линдсей П., Норман Д. Переработка информации у человека. – М.: Мир, 1974.
11. Представление и использование знаний / Под ред. Х. Уэно, М. Исидзука. – М.: Мир, 1989.
12. Гладун В.П. Планирование решений. – Киев: Наукова думка, 1987.
13. Минский М. Фреймы для представления знаний. – М.: Энергия, 1979.
14. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб: Питер, 2000.
15. Гаврилова Т.А. Онтологии для изучения инженерии знаний / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение". – СПб, 2001. – Т.1, с. 131-135.
16. Смирнов А.В., Пашкин М.П., Шилов Н.Г., Левашова Т.В. Онтологии искусственного интеллекта: способы построения и организации. – Новости искусственного интеллекта, 2002, №1. – С. 3-13.
17. Страуструп Б. Язык программирования C++. – Киев: Диасофт, 2001.
18. Codd E.F. A rational model of data for large shared data banks. – Comm. ACM, 1970, 13. – Pp. 377-387.
19. Грэй П. Логика, алгебра и базы данных. – М.: Машиностроение, 1989.
20. Построение экспертных систем. Под ред. Ф. Хейес-Рота, Д. Уотермена, Д. Лената. – М.: Мир, 1987.
21. Кузнецов В.Е. Представление в ЭВМ неформальных процедур. – М.: Наука, 1989.
22. Джексон П. Введение в экспертные системы. – М., СПб., Киев: Вильямс, 2001.
23. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proc. Nat. Acad. Sci. USA. – 1982. – 79. – p. 2554-2558.
24. Kosko B. Bidirectional associative memories. – IEEE Transactions on Systems, Man and Cybernetics, Vol. 18, No. 1, 1988. – Pp. 49-60.
25. Thorpe S.J., Delorme A., VanRullen R. Spike-based strategies for rapid processing. – Centre de Recherche Cerveau and Cognition UMR 5549.
26. Борисюк Г.Н. и др. Осцилляторные нейронные сети. Математические результаты и приложения. – Математическое моделирование, 1992, т. 4, № 1.
27. Вагин В.М. Дедукция и обобщение в системах принятия решений. – М.: Наука, 1988. – 384 с.
28. Вагин В.Н., Загорянская А.А. Использование теории аргументации для выполнения абдуктивного вывода в логическом программировании. / Труды конф. КИИ-2002, М.: Физматгиз, 2002. – Т. 1. – С. 42-51.
29. Зайченко. Исследование операций.
30. Лорьер Ж.-Л. Системы искусственного интеллекта. – М.: Мир, 1991.

31. *Элти Дж., Кумбс М.* Экспертные системы. Концепции и примеры. – М.: Финансы и статистика, 1987.
32. Приобретение знаний / Под ред. С. Осуги, Ю. Саэки. – М: Мир, 1990.
33. *Wendelken C., Shastri L.* Probabilistic inference and learning in a connectionist causal network. – Proc. of the Second Int. Symp. On Neural Computation, Berlin, 2000.
34. *Lam W., Serge A.M.* A distributed learning algorithm for bayesian inference networks. – IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 1, 2002. – Pp. 93-105.
35. *Goldberg D.E.* Genetic algorithms in search, optimization and machine learning. – Addison-Wesley, 1989.
36. *Розенблатт Ф.* Принципы нейродинамики (перцептрон и теория механизмов мозга) // М.: Мир, 1965.
37. *Радченко А.Н.* Моделирование основных механизмов мозга. – Л.: Наука, 1968.
38. *Позин Н.В.* Моделирование нейронных структур. – М.: Наука, 1970.
39. *Арбиб М.* Метафорический мозг. – М.: Мир, 1976.
40. *Васильев В.И.* Распознающие системы. Справочник. – Киев, Наукова думка, 1983.
41. *Уоссермен Ф.* Нейрокомпьютерная техника: Теория и практика. – М.: Мир. 1992.
42. *Горбань А.Н., Россиев Д.А.* Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996.
43. Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. – Новосибирск: Наука. Сибирское предприятие РАН, 1998. – 296 с.
44. *Галушкин А.И.* Теория нейронных сетей. – М.: ИПРЖР, 2000.
45. *Галушкин А.И.* Нейрокомпьютеры. – М.: ИПРЖР, 2000.
46. *Круглов В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. – М.: Горячая линия-Телеком, 2001.
47. *Каллан Р.* Основные концепции нейронных сетей. – М.: "Вильямс", 2001.
48. *Комарцова Л.Г., Максимов А.В.* Нейрокомпьютеры. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.
49. *Финн В.К.* Индуктивные модел. // В сб.: Представление знаний в человеко-машинных и робототехнических системах. Т. А. 0 М.:ВИНИТИ, 1984.
50. *Гаврилов А.В.* Архитектура программного обеспечения для поиска документов по запросу на естественном языке / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение", СПб, 2001. – Т. 1, с. 124-130.
51. *Ивахненко А.Г., Зайченко Ю.П., Димитров В.Д.* Принятие решений на основе самоорганизации. – М.: Сов. Радио, 1976.
52. *Carpenter G., A., Grossberg S.* Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MA, MIT Press, 1991.
53. *Castellano G., Fanelli A.M.* Feature selection: a neural approach. – IEEE, 1999. – Pp. 3156-3160.
54. *Parekh R., Yang J., Honavar V.* Constructive neural network learning algorithms for multi-category pattern classification. – IEEE Transactions on Neural Networks, Vol. 11, No. 2, 2000. – Pp. 436-451.
55. *Гладкий А.В.* Синтаксические структуры естественного языка в автоматизированных системах общения. – М.: Наука, 1985.
56. *Мельчук И.А.* Опыт теории лингвистических моделей "смысл-текст". – М.: Наука, 1974.
57. *Шенк Р.* Обработка концептуальной информации. – М.: Энергия, 1980.
58. *Вудс В.А.* Сетевые грамматики для анализа естественных языков // Кибернетический сборник. Новая серия, вып. 13. – М.: Мир, 1976. – С. 120-158.
59. *Золотов Е.В., Кузнецов И.П.* Расширенные системы активного диалога. – М.: Наука, 1982.
60. *Попов Э.В.* Общение с ЭВМ на естественном языке. – М.: Наука, 1986.

61. *Файн В.С.* Распознавание образов и машинное понимание естественного языка. – М.: Наука, 1987.
62. Искусственный интеллект. Справочник в 3-х томах. Книга 2: Модели и методы. – М.: Радио и связь, 1990.
63. *Завалишин Н.В., Мучник И.Б.* Модели зрительного восприятия и алгоритмы анализа изображений. – М.: Наука, 1974.
64. *Фу К.* Структурные методы в распознавании образов. М.: Мир, 1977.
65. *Шеррингтон Ч.* Интегративная деятельность нервной системы. – Л.: Наука, 1969.
66. *Гаазе-Рапопорт М.Г., Поспелов Д.А.* От амебы до робота: модели поведения. – М.: Наука, 1987.
67. *Zhdanov A.A.* The Mathematical Models of Neuron and Neural Network in Autonomous Adaptive Control Methodology / WCCI'98/IJCNN'98 Proceedings, IEEE World Congress on Computational Intelligence, Anchorage, Alaska, May 4-9, 1998. pp. 1042-1046.
68. *Редько В.Г.* Модели искусственной жизни и адаптивного поведения. / Труды конф. КИИ-2002, М.: Физматгиз, 2002, том 1. – С. 24-32.
69. *Мики Д.* Формирование и выполнение планов вычислительной машиной. – В кн.: Интегральные роботы, под ред. Г.Е. Поздняка, М.: Мир, 1975. выпуск 2. – С. 378-405.
70. *Мансон Дж.* Робот планирует, выполняет и контролирует в неопределенной среде. – В кн.: Интегральные роботы, под ред. Г.Е. Поздняка, М.: Мир, 1973. – С. 355-381.
71. *Vidal T.* Planning and executing plans in a dynamic and uncertain world. – Electronic Transactions on Artificial Intelligence, vol. 4 (2000), Section A: 1-2. – URL: <http://www.ep.liu.se/ej/etai/2000/008/>
72. *Нильсон Н.* Мобильный автомат, построенный с использованием принципов искусственного интеллекта. – В кн.: Интегральные роботы, под ред. Г.Е. Поздняка, М.: Мир, 1973. – С. 21-40.
73. *Эшби У.Р.* Конструкция мозга. Происхождение адаптивного поведения // М.: Издательство иностранной литературы, 1962.
74. *Акофф Р., Эмери Ф.* О целеустремленных системах. – М.: Советское радио, 1974.
75. *Анохин П.К.* Принципы системной организации функций. – М.: Наука, 1973.
76. *Прибрам К.* Языки мозга. – М.: Прогресс, 1975.
77. *Уилсон Р.А.* Квантовая психология. – К.: "Янус", 2001.
78. *Gavrilov A.V.* About knowledge representation and processing in intelligent systems // Int. Symp. KORUS'2000. – Ulsan, 2000. – Pp. 84-87.
79. *Анагарика Говинда.* Творческая медитация и многомерное сознание. – М.: Единство, 1993.
80. *Гаврилов А.В.* Об одной архитектуре экспертных систем / Вс. конф. "Освоение и концептуальное проектирование экспертных систем", ч. 2, М., 1989. – С. 98.
81. *Gavrilov A.V.* The Model of mind / Int. Symp. "BIOMOD-92", S.-Peterburg, 1992.
82. *Гаврилов А.В.* Архитектура "двухполушарной" экспертной системы // В кн.: Системы искусственного интеллекта: Межвуз. сб. науч. трудов под ред. А.В. Гаврилова. – Новосибирск: НГТУ, 1993. – С. 10-14.
83. *Гаврилов А.В.* Архитектура экспертной системы для работы в реальном времени. – 2-й Межд. сем. "Интеллектуализация баз данных". – Киев, 1993.
84. *Gavrilov A.V., Novickaja J.V.* The Expert Shell based on the Artificial Neural Networks / Int. Conf. NITS'94, Penza, 1994.
85. *Гаврилов А.В., Новицкая Ю.В.* Архитектура "двухполушарной" экспертной системы / В межвуз. сб. "Кибернетика и ВУЗ. Интеллектуальные информационные технологии". Вып. 28, Томск, 1994.
86. *Гаврилов А.В., Новицкая Ю.В.* Вопросы построения экспертных систем на основе нейронных сетей / 3-й Сибирский конгресс "ИНПРИМ-98", Ч. 5, Новосибирск, 1998. – С. 73.

87. *Gavrilov A.V., Novitskaya J.V.* The Architecture of the Hybrid Expert System / The 6-th Russian-Korean International Symposium on Science and Technology. Materials. – Novosibirsk, 2002. – Vol. 3. – P. 70.
88. *Gavrilov A.V.* The model of associative memory of intelligent system / The 6-th Russian-Korean International Symposium on Science and Technology. Proceedings. – Novosibirsk, 2002. – Vol. 1. – Pp. 174-177.
89. *Гаврилов А.В.* Модель ассоциативного мышления / Труды конф. КИИ-2002, М.: Физматгиз, 2002. – Т. 2. – С. 464-472.
90. *Гроссберг С.* Внимательный мозг. – Открытые системы, 1997, № 4.
91. *Lewin D.I.* Why is that computer laughing? – IEEE, 2001. – Pp. 79-81.
92. *Окс С.* Основы нейрофизиологии. – М.: Мир, 1969.

Литература к главе 2

1. *Попов Э.В.* Экспертные системы. – М.: Наука, 1987.
2. Построение экспертных систем. Под ред. Ф. Хейес-Рота, Д. Уотермена, Д. Лената. – М.: Мир, 1987.
3. Экспертные системы. Принципы работы и примеры. Под ред. Р. Форсайта. – М.: Радио и связь, 1987.
4. Экспертные системы для персональных компьютеров: методы, средства, реализации: Справочное пособие / В.С. Крисевич, Л.А. Кузьмич и др. – Мн.: Выш. шк., 1990.
5. *Элти Дж., Кумбс М.* Экспертные системы. Концепции и примеры. – М.: Финансы и статистика, 1987.
6. *Уотерман Д.* Руководство по экспертным системам. – М.: Мир, 1989.
7. *Левин Р., Дранг Д., Эдельсон Б.* Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на Бейсике. – М.: Финансы и статистика, 1990.
8. *Сойер Б. Фостер Д.Л.* Программирование экспертных систем на Паскале. – М.: Финансы и статистика, 1990.
9. *Нейлор К.* Как построить свою экспертную систему. М.: Энергоатомиздат, 1991.
10. Статические и динамические экспертные системы. Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. – М.: Финансы и статистика, 1996.
11. *Джексон П.* Введение в экспертные системы. – М., СПб., Киев: Вильямс, 2001.
12. *Венда В.Ф.* Системы гибридного интеллекта. М.: Машиностроение, 1990.
13. *Загоруйко Н.Г.* Партнерские системы – // В кн.: Системы искусственного интеллекта: Межвуз. сб. науч. трудов под ред. А.В. Гаврилова. – Новосибирск, НГТУ, 1993. – С. 3-9.
14. *Гладун В.П.* Партнерство с компьютером. – Киев: "Port-Royal", 2000.
15. *Хоггер К.* Введение в логическое программирование. М.: Мир, 1988.
16. Логический подход к искусственному интеллекту. – М.: Мир, 1990.
17. *Ковальски Р.* Логика в решении проблем. – М.: Наука, 1990.
18. *Братко И.* Программирование на языке ПРОЛОГ для искусственного интеллекта. – М.: Мир, 1990.
19. *Гаврилов А.В.* Применение языка Пролог для создания экспертных систем. – Automatyka, v. 100, Glivice, 1990, p. 43-53.
20. *Гаврилов А.В.* Об использовании языка Пролог в проектировании и программировании робототехнических комплексов / II Вс. совещание "Робототехника и ГПС", Челябинск, 1988. – С. 172-173.
21. *Фути К., Судзуки Н.* Языки программирования и схемотехника СБИС. – М.: Мир, 1988.
22. Нечеткие множества в моделях управления и искусственного интеллекта / А.Н. Аверкин, И.З. Батыршин, А.Ф. Блишун, В.Б. Силов, В.Б. Тарасов. Под ред. Д.А. Поспелова. – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 312 с.

23. *Кандрашина Е.Ю., Литвинцева А.В., Поспелов Д.А.* Представление знаний о времени и пространстве в интеллектуальных системах. – М.: Наука, 1989.
24. *Заде Л.* Понятие о лингвистической переменной и его применение к принятию решений. – М.: Мир, 1976.
25. *Нильсон Н.* Принципы искусственного интеллекта. – М.: Радио и связь, 1985.
26. Построение экспертных систем. Под ред. Ф. Хейес-Рота, Д. Уотермена, Д. Лената. – М.: Мир, 1987.
27. Представление и использование знаний / Под ред. Х. Уэно, М. Исидзука. – М.: Мир, 1989.
28. *Кузнецов В.Е.* Представление в ЭВМ неформальных процедур. – М.: Наука, 1989.
29. *Минский М.* Фреймы для представления знаний. – М.: Энергия, 1979.
30. *Гаврилов А.В.* О представлении знаний в виде фреймов на языке Пролог / Тез. докл. Вс. конф. по искусственному интеллекту. – Переславль-Залесский, 1988. – Ч. 1, С. 111-112.
31. *Линдсей П., Норман Д.* Переработка информации у человека. – М.: Мир, 1974.
32. *Гладун В.П.* Планирование решений. – Киев: Наукова думка, 1987.
33. *Hebb D.O.* The organization of behaviour. – N.Y.: Wiley, 1949.
34. *Розенблатт Ф.* Принципы нейродинамики (перцептрон и теория механизмов мозга) // М.: Мир, 1965.
35. *Радченко А.Н.* Моделирование основных механизмов мозга. – Л.: Наука, 1968.
36. *Позин Н.В.* Моделирование нейронных структур. – М.: Наука, 1970.
37. *Арбиб М.* Метафорический мозг. – М.: Мир, 1976.
38. *Кохонен Т.* Ассоциативная память. – М.: Мир, 1980.
39. *Hopfield J.J.* Neural networks and physical systems with emergent collective computational abilities // Proc. Nat. Acad. Sci. USA. – 1982. – 79. – p.2554-2558.
40. *Васильев В.И.* Распознающие системы. Справочник. – Киев, Наукова думка, 1983.
41. *Hopavar V., Uhr L.* Brain-structured connectionist networks that perceive and learn // Connection Science, 1989, 1. – Pp. 139-159.
42. *Горбань А.Н.* Обучение нейронных сетей. – М.: СП Парграф, 1990.
43. *Gavrilov A.V.* An Architecture of Neurocomputer for Image Recognition / Neural Network World, N.1, 1991. – Pp. 59-60.
44. *Гаврилов А.В.* Модель нейроподобной системы. – В сб. "Локальные вычислительные сети" под ред. А.А. Малявко, Новосибирск: НЭТИ, 1991.
45. *Carpenter G.A., Grossberg S.* Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MA, MIT Press, 1991.
46. *Уоссермен Ф.* Нейрокомпьютерная техника: Теория и практика. – М.: Мир, 1992.
47. *Борисюк Г.Н. и др.* Осцилляторные нейронные сети. Математические результаты и приложения. – Математическое моделирование, 1992, т. 4, № 1.
48. *Горбань А.Н., Россиев Д.А.* Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996.
49. *Гроссберг С.* Внимательный мозг. – Открытые системы, 1997, № 4.
50. *Галушкин А. И.* Современные направления развития нейрокомпьютерных технологий в России. – Открытые системы, 1997, № 4. – С. 25-28.
51. *Горбань А.Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей. – Сибирский журнал вычислительной математики. – 1998. – Т. 1, № 1. – С. 12-24.
52. Нейроинформатика / А.Н. Горбань, В.Л. Дунин-Барковский, А.Н. Кирдин и др. – Новосибирск: Наука. Сибирское предприятие РАН, 1998. – 296с.
53. *Zhdanov A.A.* The Mathematical Models of Neuron and Neural Network in Autonomous Adaptive Control Methodology / WCCI'98/IJCNN'98 Proceedings, IEEE World Congress on Computational Intelligence, Anchorage, Alaska, May 4-9, 1998. pp. 1042-1046.
54. *Галушкин А.И.* Теория нейронных сетей. – М.: ИПРЖР, 2000.
55. *Галушкин А.И.* Нейрокомпьютеры. – М.: ИПРЖР, 2000.

56. *Круглов В.В., Борисов В.В.* Искусственные нейронные сети. Теория и практика. – М.: Горячая линия-Телеком, 2001.
57. *Каллан Р.* Основные концепции нейронных сетей. – М.: Вильямс, 2001.
58. *Комарцова Л.Г., Максимов А.В.* Нейрокомпьютеры. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.
59. *Куссуль Э.М.* Ассоциативные нейроподобные структуры. – Киев, Наукова думка, 1990.
60. *Дорогов А.Ю.* Модальные категории модульных нейронных сетей // Проблемы нейрокибернетики (материалы XII Межд. конф. по нейрокибернетике). – Ростов-на-Дону, 1999. – С. 137-141.
61. *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем. – СПб: Питер, 2000.
62. *Смирнов А.В., Пашкин М.П., Шилов Н.Г., Левашова Т.В.* Онтологии в искусственного интеллекта: способы построения и организации. – Новости искусственного интеллекта, 2002, № 1. – С. 3-13.
63. *Yang J., Pai P., Honavar V., Miller L.* Mobile Intelligent Agents for Document Classification and Retrieval: A Machine Learning Approach.
64. *Загоруйко Ю.А., Попов И.Г., Костов Ю.В., Сергеев И.П.* Общая концепция агентов в системе моделирования Semp-A1 / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение", СПб, 2001. – Т.1, С. 259-267.
65. *Барсуков А.В., Гаврилов А.В., Олейник Е.И.* Представление знаний в системе распределенных баз знаний "СОКРАТ" / Труды Междунар. конф. "Научные основы высоких технологий" НОВТ-97, Новосибирск: НГТУ, 1997. – Т. 2. – С. 212-217.
66. *Барсуков А.В., Гаврилов А.В.* Распределенная система баз знаний в среде Internet / 6-й Междунар. семинар "Распределенная обработка информации" (РОИ-98), Новосибирск, 1998. – С. 353-356.
67. *Эйкинс Я.С.* Знание, организованное в виде прототипов, для экспертных систем. – / В сб. "Кибернетический сборник", вып. 22. – М.: Мир, 1985. – С. 221-277.
68. *Ковригин О.В., Перфильев К.Г.* Гибридные средства представления знаний в системе СПЭИС / Вс. конф. по искусственному интеллекту. Тезисы докладов, т. 2. – Москва, 1988. – С. 490-495.
69. *Гаврилов А.В., Новицкая Ю.В.* Инструментальное программное обеспечение для создания гибридных экспертных систем. – Доклады Межд. науч.-техн. конф. "Информационные системы и технологии" ИСТ-2000, Новосибирск: НГТУ, 2000. – Т. 3, С. 488-490.
70. *Гаврилов А.В., Новицкая Ю.В.* Архитектура гибридной экспертной системы. – Межд. Симп. "ИНПРИМ-2000", Новосибирск, 2000.
71. *Гаврилов А.В., Новицкая Ю.В.* Инструментальное программное обеспечение для создания гибридных экспертных систем – Рег. научно-практич. конф. "АГРОИНФО-2000", Новосибирск, 2000. – С. 142.
72. *Gavrilov A.V., Novickaja J.V.* The Toolkit for development of Hybrid Expert Systems. – 5-th Int. Symp. "KORUS-2001". – Tomsk: TPU, 2001. – Proceedings. – Vol.1. – Pp. 73-75.
73. *Элти Дж., Кумбс М.* Экспертные системы. Концепции и примеры. – М.: Финансы и статистика, 1987.
74. Система обработки декларативных структур знаний "ДЕКЛАР-2". – Под общ. ред. И.П. Кузнецова. – Препринт, М.: ИПИАН, 1988.
75. *Funobashi M., Moeda A., Morooka, Y.Mori K.* Fuzzy and Neural Hybrid Expert Systems: Synergetic AI. – AI in Japan, IEEE, 1995, august. – Pp. 33-40.
76. *Ranganathan N., M. I. Patel, Sathuamurthy R.* An intelligent system for failure detection and control in an autonomous underwater vehicle / IEEE Transactions on systems, man and cybernetics. – Part A: Systems and Humans, Vol. 31, No. 6, November, 2001. – Pp. 762-767.
77. *Nikolopoulos C., Fellrath P.* A hybrid expert system for investment advising. – IEEE, 1994. – Pp. 1818-1820.

78. *Mori H., Aoyamu H., Yamanaka T., Urano S.* A hybrid intelligent system for fault detection in power systems. – IEEE, 2002. – Pp. 2138-2143.
79. *Гаврилов А.В.* Гибридная экспертная система для профориентации / В сб. научных трудов НГТУ, № 3(8), 1997. – С. 123-132.
80. *Альсова О.К.* Прогнозирование притока реки Обь в створе Новосибирской ГЭС на основе вариативного моделирования. Автореф. канд. дисс. – Новосибирск: НГТУ, 2002.

Литература к главе 3

1. *Власов А.И., Иванов И.П., Тимошкин А.Г.* Применение нейросетевых методов в управлении качеством при производстве современных электронных изделий / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 137-143.
2. *Еремин Д.М., Лохин В.М.* Многослойная нейронная сеть прямого распространения в системах прямого управления / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 1. – Москва, 2000. – С. 243-249.
3. Лаборатория "Base Group". Технологии анализа данных. – <http://www.basegroup.ru>
4. *Аршавский А.В., Чепин Е.В.* О подходе к обработке изображений с использованием нейронной сети / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 13-16.
5. *Назаров Л.Е., Букатова И.Л.* Применение многослойных нейронных сетей для классификации земных объектов на основе анализа данных дистанционного зондирования / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 107-112.
6. *Бархатов Н.А., Беллюстин Н.С., Сахаров С.Ю.* Предсказание геомагнитной активности методом искусственных нейронных сетей / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 17-23.
7. *Назимова Д.И., Царегородцев В.Г.* Нейросетевая идентификация зональных групп лесных формаций Сибири / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 112-119.
8. *Жернаков С.В.* Идентификация параметров ГТД гибридным ансамблем нейросетей / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 45-51.
9. *Касаткина Л.М., Кукуль Э.М., Байдык Т.Н.* Распознавание рукописных текстов на нейронных сетях / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 71-76.
10. *Гареев А.Ф.* Применение вероятностной нейронной сети для автоматического рубрицирования текстов / В сб. науч. трудов "Нейроинформатика-2000". Всероссийская науч.-техн. конф., ч. 3. – Москва, 1999. – С. 71-78.
11. *Жигирев Н.Н., Корж В.В., Оныкий Б.Н.* Самонастраивающийся словарь ключевых слов и нейронная сеть хопфилда для классификации текстов / Труды 2-й Всероссийской науч.-технич. конференции "Нейроинформатика-2000", ч. 2. – М., 2000. – С. 58-61.
12. *Шумский С.А., Яровой А.В., Зорин О.Л.* Ассоциативный поиск текстовой информации / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 101-109.
13. *Корнеев В.В., Гарев А.Ф., Васютин С.В., Райх В.В.* Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2000.
14. Нейронные сети. STATISTICA Neural Networks. – М.: Горячая линия – Телеком, 2000.
15. *Гаврилов А.В., Канглер В.М.* Использование искусственных нейронных сетей для обнаружения ассоциативных взаимосвязей между значениями полей в реляционных базах

- данных / Тез. докл. 6-й Всерос. семинара "Нейроинформатика и ее приложения", Красноярск, 1998. – С. 27.
16. Гаврилов А.В., Канглер В.М. Использование искусственных нейронных сетей для анализа данных // Сб. научн. трудов НГТУ. – Новосибирск: Изд-во НГТУ, 1999. – № 3 (16). – С. 56-63.
 17. Гаврилов А.В., Канглер В.М., Катомин М.Н., Коротенко А.И. Обнаружение ассоциативных взаимосвязей между полями в базах данных с использованием нейронной сети / Труды межд. н.-т. конф. "Научные основы высоких технологий", Том 2, Новосибирск, 1997. – С. 210-211.
 18. Gavrilov A.V., Kangler V.M. The use of Artificial Neural Networks for Data Analysis // The Third Russian-Korean International Symposium on Science and Technology. – Novosibirsk: NSTU, 1999. – Proceedings/ – Vol.1. – P.257-260; Abstracts. – Vol. 1. – P. 192.
 19. Гаврилов А.В., Губарев В.В. Применение модели Хопфилда для решения задачи прогнозирования на примере анализа притока реки Обь / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 33-38.
 20. Gubarev V.V., Alsova O.K., Belenky A.I., Gavrilov A.V., Golovansky A.P., Jatsko V.A. Research of Hydrological Series Laws of the Inflow Charge of the river Ob in the Novosibirsk Hydroelectric station Power Site // The Third Russian-Korean International Symposium on Science and Technology / Abstracts. – Novosibirsk: NSTU, 1999. – Vol.1. – P.295.
 21. Лалетин П.А., Ланкина Э.Г., Ланкин Ю.П. Использование сетей с самостоятельной адаптацией для распознавания слов человеческой речи / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – Москва, 2000. – С. 88-95.
 22. А.В. Гаврилов. Гибридная экспертная система для профориентации / В сб. научных трудов НГТУ, № 3 (8), 1997. – С. 123-132.
 23. Гаврилов А.В. Архитектура программного обеспечения для поиска документов по запросу на естественном языке / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение", СПб, 2001. – Т. 1, с. 124-130.
 24. Гаврилов А.В. Об одной архитектуре экспертных систем / Вс. конф. "Освоение и концептуальное проектирование экспертных систем", ч. 2, М., 1989. – С. 98.
 25. Гаврилов А.В. Проблемы обработки символьной информации в нейронных сетях / Международный конф. "Мягкие вычисления и измерения" (SCM-98), С.-Петербург, 1998.
 26. КуССуль Э.М. Ассоциативные нейроподобные структуры. – Киев: Наукова думка, 1990.
 27. Gavrilov. A.V. An Architecture of Neurocomputer for Image Recognition / Int. Conf. LSPIC'90, Vol. 2. – Riga, 1990. – Pp. 306-308
 28. Gavrilov A.V. An Architecture of Neurocomputer for Image Recognition / Neural Network World, N.1,1991. – Pp. 59-60.
 29. Гаврилов А.В. Модель нейроподобной системы. – В сб. "Локальные вычислительные сети" под ред. А.А. Малявко, Новосибирск: НЭТИ, 1991.
 30. Гаврилов А.В. Моделирование нейронной сети на основе нейрона "ключ-порог" / 3-й Сибирский конгресс "ИНПРИМ-98", Ч. 5, Новосибирск, 1998. – С. 73.
 31. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия-Телеком, 2001.
 32. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proc. Nat. Acad. Sci. USA. – 1982. – 79. – p. 2554-2558.
 33. Веденов А.А., Ежов А.А., Книжникова Л.А., Левченко Е.Б., Чернов Ю.Г. Нелинейные системы с памятью и моделирование функций нейронных ансамблей / В кн.: "Интеллектуальные процессы и их моделирование", М.: Наука, 1987. – С. 229-248.
 34. Губарев В.В., Альсова О.К., Беленький А.И., Гаврилов А.В., Голованский А.П., Давыдова Т.Н., Канглер В.М. Управление Новосибирским водохранилищем на основе прогнозирования притока / Водное хозяйство России. Проблемы, технологии, управление. – Екатеринбург: Изд-во РосНИИВХ, 2000, т. 2, № 5. – С. 484-499.

35. *Альсова О.К.* Прогнозирование притока реки Обь в створе Новосибирской ГЭС на основе вариативного моделирования. – Автореферат канд. дисс., Новосибирск: НГТУ, 2002.
36. *Губарев В.В.* Интеллектуальный анализ данных и вариативное моделирование в экспериментальных исследованиях / В кн.: "Информационные системы и технологии ИСТ'2000. Сб. научных трудов под ред. В.В. Губарева. – Новосибирск: НГТУ, 2001. – С. 5-25.
37. *Гаврилов А.В.* Архитектура "двухполушарной" экспертной системы // В кн.: Системы искусственного интеллекта: Межвуз. сб. науч. трудов под ред. А.В. Гаврилова. – Новосибирск: НГТУ, 1993. – С. 10-14.
38. *Гаврилов А.В.* Архитектура экспертной системы для работы в реальном времени. – 2-й Межд. Сем. "Интеллектуализация баз данных". – Киев, 1993.
39. *Гаврилов А.В., Новицкая Ю.В.* Архитектура "двухполушарной" экспертной системы / В межвуз. сб. "Кибернетика и ВУЗ. Интеллектуальные информационные технологии". Вып. 28, Томск, 1994.
40. *Gavrilov A.V.* The Model of mind / Int. Symp. "BIOMOD-92", S.-Peterburg, 1992.
41. *Gavrilov A.V., Novickaja J.V.* The Expert Shell based on the Artificial Neural Networks / Int. Conf. NITS'94, Penza, 1994.
42. *Гаврилов А.В., Новицкая Ю.В.* Вопросы построения экспертных систем на основе нейронных сетей / 3-й Сибирский конгресс "ИНПРИМ-98", Ч. 5, Новосибирск, 1998. – С. 73.
43. *Gavrilov A.V., Novitskaya J.V.* The Architecture of the Hybrid Expert System / The 6-th Russian-Korean International Symposium on Science and Technology. Materials. – Novosibirsk, 2002. – Vol. 3. – P. 70.
44. Построение экспертных систем. Под ред. Ф. Хейес-Рота, Д. Уотермена, Д. Лената. – М.: Мир, 1987.
45. *Gavrilov A.V., Kangler V.M., Zaitsev S.A.* Data Analysis Program by means of Neural Networks / The 6-th Russian-Korean International Symposium on Science and Technology. Materials. – Novosibirsk, 2002. – Vol. 3. – P. 71.

Литература к главе 4

1. *Линдсей П., Норман Д.* Переработка информации у человека. – М.: Мир, 1974.
2. *Виноград Т.* Программа, понимающая естественный язык. – М.: Мир, 1976.
3. *Шенк Р.* Обработка концептуальной информации. – М.: Энергия, 1980.
4. *Yang J., Pai P., Honavar V., Miller L.* Mobile Intelligent Agents for Document Classification and Retrieval: A Machine Learning Approach.
5. *Honkela T., Kaski S., Lagus K., Kohonen T.* Exploration of full-text databases with self-organizing maps. – Proc. of Int. Conf. On Neural Networks (ICNN'96), 1996.
6. *Lowrence S., Giles C.L., Fong S.* Natural language grammatical inference with recurrent neural networks. – IEEE Transactions on Knowledge and Data Engineering. – 2000, Vol. 12, N. 1, Pp. 126-140.
7. *McRoy S.W., Haller S.M., Ali S.S.* Mixed depth representations for dialog processing. – Proc. Of Cognitive Science '98, 1998. – Pp. 687-692.
8. *Tong S., Koller D.* Support vector machine active learning with applications to text classification. – Journal of Machine Learning Research, 2001. – Pp. 45-66.
9. *Мельчук И.А.* Опыт теории лингвистических моделей "смысл-текст". – М.: Наука, 1974.
10. *Файн В.С.* Распознавание образов и машинное понимание естественного языка. – М.: Наука, 1987.
11. *Ловицкий В.А.* Обучаемая естественно-языковая система автоматизированного синтеза алгоритмов / Вс. конф. по искусственному интеллекту. Тезисы докладов. – Москва, 1988. – Т. 3, С. 425-439.

12. Мартынов В.В. УСК-4, базы знаний и решение задач / Вс. конф. по искусственному интеллекту. Тезисы докладов. – Москва-Переславль-Залесский, 1988. – Т. 1, С. 213-218.
13. Левин Д.Я., Нариньяни А.С. Экспериментальный минипроцессор. Семантически ориентированный анализ // Взаимодействие с ЭВМ на естественном языке. – Новосибирск: ВЦ СО АН СССР, 1978. – С. 223-233.
14. Моделирование языковой деятельности в интеллектуальных системах. Под ред. А.Е. Кибрика и А.С. Нариньяни. – М.: Наука, 1987.
15. Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах. – М.: Наука, 1989.
16. Любарский Ю.Я. Интеллектуальные информационные системы. – М.: Наука, 1990.
17. Соломатин Н.М. Информационные семантические системы // В уч. пос. "Перспективы развития вычислительной техники" в 11 кн. Кн. 1. – М.: Высшая школа, 1989.
18. Попов Э.В. Общение с ЭВМ на естественном языке. – М.: Наука, 1986.
19. Попов Э.В. Общение с базами данных на ограниченном естественном языке: прошлое, настоящее и будущее. – Новости искусственного интеллекта, 2002, № 1. – С. 21-26.
20. Золотов Е.В., Кузнецов И.П. Расширенные системы активного диалога. – М.: Наука, 1982.
21. Кузнецов И.П. Кибернетические диалоговые системы. – М.: Наука, 1976.
22. Кузнецов И.П. Механизмы обработки семантической информации. – М.: Наука, 1978.
23. Osipov G.S. Semantic types of natural language statements. A method of representation. – Proc. Of 10th IEEE Int. Symp. on Intelligent Control, 1995.
24. Шумский С.А., Яровой А.В., Зорин О.Л. Ассоциативный поиск текстовой информации / Труды Всероссийской науч.-технич. конференции "Нейроинформатика-1999", ч. 3. – М., 1999. – С. 101-109.
25. Шумский С.А., Яровой А.В., Зорин О.Л. Ассоциативный поиск текстовой информации / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 2. – М., 2000. – С. 101-109.
26. Жигирев Н.Н., Корж В.В., Оныкий Б.Н. Самонастраивающийся словарь ключевых слов и нейронная сеть хопфилда для классификации текстов / Труды 2-й Всероссийской науч.-технич. конференции "Нейроинформатика-2000", ч. 2. – М., 2000. – С. 58-61.
27. Жигирев Н.Н., Корж В.В., Оныкий Б.Н. Использование асимметрии частотных свойств информационных признаков для построения автоматизированных систем классификации текстовых документов / В сб. науч. трудов "Нейроинформатика-2000". 2-я Всероссийская науч.-техн. конф., ч. 3. – Москва, 2000. – С. 83-91.
28. Дударь З.В., Шуклин Д.Е. Семантическая нейронная сеть как формальный язык описания и обработки смысла текстов на естественном языке. – Радиоэлектроника и информатика. – Харьков: Изд-во ХТУРЭ, 2000. – № 3, с. 72-76.
29. Шуклин Д.Е. Применение семантической нейронной сети в экспертной системе, преобразующей смысл текста на естественном языке. – Радиоэлектроника и информатика. – Харьков: Изд-во ХТУРЭ, 2001. – № 1.
30. Флоренсов А.Н. Построение семантического пространства для информационных систем. / Труды междунар. конф. ИСТ'2000, Новосибирск, НГТУ, 2000. – Т. 3, С. 532-535.
31. Флоренсов А.Н. Метризация знаний в системах информации / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение", СПб, 2001. – Т. 2, с. 608-614.
32. Лорьер Ж.-Л. Системы искусственного интеллекта. – М.: Мир, 1991.
33. Филмор Ч. Дело о падеже // В кн.: Новое в зарубежной лингвистике, Вып. X, Лингвистическая семантика. – М.: Прогресс, 1981. – С. 369-495.
34. Гаврилов А.В. Диалоговая система подготовки программ для роботов. – В сб. "Диалоговые системы в задачах управления". – Новосибирск: НЭТИ, 1987.
35. Гаврилов А.В. Диалоговая система подготовки программ для роботов. – В сб. Automatyka, v. 99, Glivice, 1988, p. 173-180.

36. *Гаврилов А.В., Щипцов В.А.* Системное программное обеспечение транспортного робота / Тез. докл. Вс. научно-техн. конф. "Микропроцессорные системы автоматизации технологических процессов", Новосибирск, 1987. – С. 230-231.
37. *Гаврилов А.В., Амелин Б.Г., Веприк В.Н., Ерофеев Ю.Ф.* Локальная сеть ячеистой топологии из микро-ЭВМ / В сб. "Информационно-измерительные системы (Теория и реализация). – Новосибирск: НЭТИ, 1987.
38. *Гаврилов А.В., Гоньиз Х.К., Гарсия М.Н., Медина В.Е.* Система управления распределенными базами данных для использования в гибких автоматизированных производствах / Тез. докл. 2-й Вс. научно-техн. конф. "Микропроцессорные системы автоматики", Новосибирск, 1990, ч. 1. – С. 99.
39. *Гаврилов А.В.* Архитектура программного обеспечения для поиска документов по запросу на естественном языке / Труды межд. конф. KDS-2001 "Знание-Диалог-Решение", СПб, 2001. – Т. 1, с. 124-130.
40. *Гаврилов А.В., Зайцев С.А., Макаревич Л.Г., Романов Е.Л.* Автоматизированная система тестирования знаний в среде Internet/Intranet. – Открытое и дистанционное образование, № 1 (3), 2001. – С. 49-51.
41. *Всеволодский С.Н., Гаврилов А.В.* Принципы построения интеллектуальной системы тестирования знаний с ответами на естественном языке / Материалы Межд. Конф. "Открытое и дистанционное образование: анализ опыта и перспективы развития", Барнаул, 2002. – С. 60-62.

ОГЛАВЛЕНИЕ

	Стр.
Предисловие	3
Введение.....	5
1. Элементы теории интеллектуальных систем	13
1.1. Постановка задачи	13
1.2. Модель интеллектуальной системы [78].....	17
1.3. Модель ассоциативного мышления [88, 89]	20
1.3.1. Постановка задачи	20
1.3.2. Ассоциации и ассоциативный поиск	20
1.3.3. Формирование ассоциаций	26
1.3.4. Нечеткое подобие	26
1.4. Принципы организации функционирования интеллектуальных систем	30
1.4.1. Принцип обучения посредством формирования и последующего закрепления ассоциаций, происходящего в процессе ассоциативного вспоминания	30
1.4.2. Принцип концентрации и экономии ресурсов	31
1.4.3. Принцип неопределенности.....	31
1.4.4. Принцип единства нечетких рассуждений и четких действий	33
1.5. Количественная оценка знаний в сообщении	34
1.6. Многомерные лингвистические переменные и иерархические нейронные сети	36
1.7. Моделирование эмоций	38
1.8. Выводы	39
2. Комбинирование разных методов представления и обработки знаний в гибридных экспертных системах.....	41
2.1. Методы представления знаний в гибридных экспертных системах	41
2.2. Архитектура инструментального программного обеспечения ESWin для создания гибридных экспертных систем [69-72].....	45
2.2.1. Состав и назначение ПО.....	45
2.2.2. База знаний	46
2.2.3. Фреймы	49
2.2.4. Правила-продукции	50
2.2.5. Связь с внешними базами данных	53
2.2.6. Лингвистические переменные	56
2.2.7. Интерпретация правил-продукций.....	57
2.3. Выводы	60
3. Использование нейросетевых моделей в искусственных интеллектуальных системах	62
3.1. Варианты использования нейронных сетей в современных интеллектуальных системах	62
3.2. Обработка символьной информации в нейронных сетях [25]	63
3.3. Модель нейронной сети "ключ-порог" [27-30].....	66
3.4. Гибридная экспертная система для профориентации [22]	70
3.4.1. Введение	70
3.4.2. Структура экспертной системы.....	70
3.4.3. Представление знаний	73
3.4.4. Заключение	77
3.5. Архитектура программы AnalDB для анализа баз данных с помощью нейронных сетей [15-18]	77
3.5.1. Назначение и функции программы AnalDB.....	77
3.5.2. Этапы решения задач с помощью программы AnalDB.....	81

3.5.3. Эксперименты по применению программы AnalDB для решения задачи прогнозирования притока реки Обь [19, 20, 34]	85
3.5.3.1. Постановка задачи	85
3.5.3.2. Пример эксперимента	86
3.5.3.3. Результаты экспериментов	87
3.5.3.4. Выводы	88
3.5.5. Заключение	89
3.6. Архитектура двухполушарных экспертных систем [24, 36-41]	90
3.7. Выводы	93
4. Формирование и распознавание смысла в естественном языке с использованием гибридного подхода	94
4.1. Постановка задачи	94
4.2. Обучаемые системы представления и распознавания смысла в ЕЯ	97
4.3. Программное обеспечение для программирования роботов на естественном языке	99
4.3.1. Постановка задачи	99
4.3.2. Принципы построения ДИСПОР	100
4.3.3. Представление знаний	104
4.3.4. Формирование управляющей программы на языке БАЯР	106
4.3.5. Заключение	106
4.4. Программное обеспечение для тестирования знаний с использованием ответов на естественном языке	107
4.4.1. Назначение	107
4.4.2. Структура системы	107
4.4.3. Типы вопросов, задаваемых системой	108
4.4.4. Структура базы знаний	109
4.4.5. Словарь ЕЯ	110
4.4.6. Программирование сценария диалога	111
4.4.7. Заключение	112
4.5. Программное обеспечение для поиска документов по смыслу	113
4.5.1. Постановка задачи	113
4.5.2. Архитектура программного обеспечения для обучения на основе обработки документов и для поиска документов по смыслу	113
4.5.3. Алгоритмы	118
4.5.4. Обучение	119
4.5.5. Исследовательский прототип программного обеспечения	120
4.5.6. Заключение	123
4.6. Выводы	123
Заключение	124
Литература	126
Литература к введению	126
Литература к главе 1	129
Литература к главе 2	132
Литература к главе 3	135
Литература к главе 4	137